

Configuración de Repositorios e Instalación Paquetes

1.1 ¿Qué es un paquete?

Un paquete no es más que un archivero (también llamado 'tar') que contiene varios ficheros dentro como si de uno sólo se tratase. Este paquete normalmente suele traer una versión precompilada del software a instalar, aunque puede descargarse también las fuentes para compilarlo manualmente.

1.2 ¿Qué es una dependencia?

Una dependencia es aquel paquete/programa necesario por otro paquete para poder funcionar correctamente.

Según la filosofía de GNU/Linux, un paquete (programa) solamente tiene una única función en el sistema. Al tener una sola función puede ser utilizada en cualquier momento por otro paquete en cuanto le sea necesario. Esta modularización hace que un paquete dependa de otro para poder funcionar, y de esta forma no vamos a tener dos programas que realicen la misma función en el sistema sino un mismo programa accesible por todos los demás.

Como hemos dicho anteriormente, para que un paquete pueda ser instalado correctamente deben cumplirse todas sus dependencias.

1.3 ¿Qué es un gestor de paquetes?

El gestor de paquetes es un software que se encarga de la instalación/desinstalación de paquetes del sistema, resolviendo a su vez todas las dependencias de un determinado paquete.

En Debian el gestor de paquetes se llama APT (Advanced Packaging Tool)

1.4 Funcionamiento de APT

Como hemos comentado anteriormente, en los inicios de GNU/Linux los usuarios tenían que compilar sus programas para poder hacer uso de ellos y resolver las dependencias de estos de forma manual.

Cuando apareció Debian, esta contaba con un programa llamado 'dpkg' encargado de manejar los paquetes en el sistema. De esta forma, el usuario ya no tenía que compilar sus programas si no que se adquiría un paquete con extensión .deb ya precompilado (lo cual no quiere decir que no se siguiese utilizando la compilación de programas).

Pero dpkg tenía un problema, y es que no resuelve las dependencias de un paquete, es decir, dpkg solo instala, desinstala y muestra información acerca de paquetes. Si un paquete tiene una dependencia tendremos que adquirir dicha dependencia (descargarla manualmente) e instalarla para poder continuar con la instalación del paquete.

Para facilitar, mejorar y conseguir una gestión de paquetes mucho más eficiente Debian comenzó a utilizar una herramienta llamada APT (Advanced Packaging Tool = Herramienta Avanzada de Empaquetamiento). Esta herramienta innovó en la gestión de paquetes debido a que resuelve las dependencias de un determinado paquete y se basa en repositorios para descargar los paquetes y dependencias a instalar.

Debido a que dpkg es una herramienta de bajo nivel, se necesitaba una aplicación de alto nivel que se encargase de resolver las dependencias y descargar los paquetes de lugares remotos.

1.4.1 ¿Qué es un repositorio?

Un repositorio es un servidor remoto donde se encuentran alojados todos los paquetes de una determinada distribución GNU/Linux. Normalmente estos repositorios permiten la descarga de paquetes mediante varios protocolos como http o ftp.

Un repositorio no tiene por qué contener solamente los paquetes de una distribución, es más normalmente contienen los paquetes de varias distribuciones.

Además de contener paquetes, en estos repositorios podemos encontrar imágenes iso de la distribución listas para descargar a través de http, ftp, bitTorrent, jigdo, etc.

El funcionamiento de APT (como hemos dicho anteriormente) se basa en el uso de una lista de repositorios desde los cuales va a poder descargar los paquetes y dependencias. Esta lista de repositorios se configura en el fichero /etc/apt/sources.list.

Ejemplo de fichero sources.list

```
#Repositorio CICA
```

```
deb http://ftp.cica.es/debian/ etch main contrib non-free
```

```
deb-src http://ftp.cica.es/debian/ etch main contrib non-free
```

#Repositorios de seguridad

deb http://security.debian.org/ etch/updates main contrib

deb-src http://security.debian.org/ etch/updates main contrib

APT se descarga el índice de paquetes de los repositorios que tenemos configurados para poder realizar búsquedas de paquetes. Cuando decidimos instalar un paquete, APT se lo descarga de los repositorios junto con sus dependencias y los almacena en el directorio `/var/cache/apt/archives`. Una vez descargado el paquete y sus dependencias, APT le pasa el turno a `dpkg` el cual se encarga de instalarlos en el sistema.

Resumiendo, APT y `dpkg` trabajan en equipo y por turnos; APT se encarga de resolver las dependencias y descargar los paquetes y `dpkg` los instala en el sistema

comando apt-get

apt-get es la herramienta que utiliza Debian y sus derivadas (Ubuntu incluida), para gestionar los paquetes instalables disponibles en los repositorios . Existen a nuestra disposición herramientas gráficas.

Comandos

`apt-get update`

Actualiza el listado de paquetes disponibles.

`apt-get check`

Comprueba que todo ha ido bien tras la utilización de `apt-get update`.

`apt-get install lista_de_paquetes`

Instala los programas deseados.

`apt-get upgrade`

Instala las actualizaciones.

`apt-get remove lista_de_paquetes`

Desinstala un paquete.

`apt-get clean`

Para limpiar los paquetes descargados e instalados.

`apt-get autoclean`

Para limpiar los paquetes viejos que ya no se usan.

`apt-cache search nombre paquete`

Para buscar un paquete determinado.

`apt-show-versions (-u)`

Para conocer que paquetes hay instalados.

`apt-cache show paquete`

Obtener más información de un paquete específico.

`apt-cache depends paquete`

Para saber de que paquete depende.

`apt-file search filemane`

Para encontrar el nombre de un paquete desde un archivo.

`apt-file list paquete`

Lista el contenido de un paquete.

`apt-cache pkgnames -generate`

Muestra una lista de todo lo que tenemos instalado en el sistema.

Opciones

`apt-get -s`

Simula una acción.

`apt-get -d`

Sólo descarga.

`apt-get -y`

No pregunta y asume que si a todo.

`apt-get -u`

Muestra paquetes actualizados.

`apt-get -h`

Muestra texto de ayuda.

`apt-get -q`

Salida registrable – sin indicador de progreso.

`apt-get -qq`

Sin salida, excepto si hay errores.

`apt-get -f`

Intenta continuar si la comprobación de integridad falla.

`apt-get -m`

Intenta continuar si los archivos no son localizables.

`apt-get -b`

Construye el paquete fuente después de obtenerlo .

`apt-get -V`

Muestra números de versión detallados.

`apt-get -c=?`

Lee este archivo de configuración.

DIFERENCIA ENTRE LOS COMANDOS **apt-get update** y **apt-get upgrade**

apt-get update
apt-get upgrade

- **apt-get update:** actualiza la lista de paquetes disponibles y sus versiones, pero no instala o actualiza ningún paquete.

Esta lista la coge de los servidores con repositorios que tenemos definidos en el `sources.list`.

- **apt-get upgrade:** una vez el comando anterior ha descargado la lista de software disponible y la versión en la que se encuentra, podemos actualizar dichos paquetes usando este comando: [apt-get upgrade](#).

Instalará las nuevas versiones respetando la configuración del software cuando sea posible (esta es la maravilla de este tipo de sistemas).

En resumen: *el update lista los paquetes de los repositorios y el upgrade instala las actualizaciones.*

Sources.list es el archivo situado en [/etc/apt/sources.list](#) en distribuciones GNU/Linux derivadas de Debian GNU/Linux.

En él se **listan** las "fuentes" o "repositorios" disponibles de los paquetes de software candidatos a ser: *actualizados, instalados, removidos, buscados, etc.*

apt-get install

`apt-get install`

Comando para instalar paquetes

Ejemplos: INSTALACIÓN DEL comando man EN ESPAÑOL

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Y ahora sí:

```
$ sudo apt-get install manpages-es manpages-es-extra
$ sudo dpkg-reconfigure locales
```

Hay comandos que no están traducidos al español y aunque se instale el paquete no aparecen.

COMO SABER SI UN PAQUETE SE ENCUENTRA DISPONIBLE /INSTALADO:

Para saber si un paquete se encuentra disponible en los repositorios,

- Arch Linux: `pacman -Ss paquete`
- Fedora: `yum search paquete`
- Debian/Ubuntu: `apt-cache search paquete`
- OpenSUSE: `zypper se paquete`
- Gentoo: `emerge -S paquete`

Para saber si un paquete se encuentra instalado en nuestra máquina:

- Arch Linux: `pacman -Qs paquete`
- Fedora: `rpm -qa | grep paquete`
- Debian/Ubuntu: `dpkg -l | grep paquete`
- OpenSUSE: `zypper se -i paquete`
- Gentoo: `emerge -pv paquete`

Añadir repositorios PPA

¿**Por qué** agregar **repositorios PPA** si ya disponemos de miles de programas utilizando los repositorios oficiales de Ubuntu?

Los archivos de paquetes personales (**P**ersonal **P**ackage **A**rchive, en inglés), **permiten a los desarrolladores distribuir software y sus respectivas actualizaciones en forma directa** a los usuarios de Ubuntu sin tener que esperar a que se actualicen los propios repositorios de Ubuntu.

Launchpad, el sitio que alberga la mayor parte de los PPAs disponibles, construye los binarios y los almacena en un repositorio específico. Esto significa que los usuarios de Ubuntu pueden instalar estos paquetes del mismo modo en que están acostumbrados a instalar el resto de las aplicaciones en Ubuntu, con la ventaja extra de que dispondrán de las últimas actualizaciones de estos programas e incluso podrán encontrar programas que no están disponibles en los repositorios oficiales.

Cómo instalar repositorios PPA

Pongamos un ejemplo práctico. Supongamos que queremos instalar Shutter. Lo primero que tenemos que saber es el nombre identificador del PPA que queremos instalar. En la página del PPA de Shutter queda claro que para agregar este repositorio hace falta tomar nota de la línea *ppa:shutter/ppa*.

Opción 1: desde la línea de comandos

Todo lo que hay que hacer es abrir un terminal e ingresar los comandos apropiados para agregar el PPA, actualizar la lista de paquetes e instalar el programa deseado (en nuestro ejemplo, Shutter).

```
sudo add-apt-repository ppa:shutter/ppa
sudo apt-get update
sudo apt-get install shutter
```

Opción 2: desde el Centro de Software

- 1.- Abrir el Centro de Software de Ubuntu.
- 2.- Editar > Orígenes del Software
- 3.- Luego, en la pestaña Otro software, clic en Agregar e ingresar la línea del PPA. En nuestro ejemplo: *ppa:shutter/ppa* y clic en Aceptar.

Cómo quitar repositorios PPA

Opción 1: quitar el PPA desde la línea de comandos

Siguiendo nuestro ejemplo de Shutter:

```
sudo add-apt-repository --remove ppa:shutter/ppa
```

Obviamente, habrá que reemplazar la línea ppa:shutter/ppa por lo que corresponda en cada caso.

Opción 2: desde el Centro de Software

- 1.- Abrir el Centro de Software de Ubuntu.
- 2.- Editar > Orígenes del Software
- 3.- Luego, en la pestaña Otro software, clic en Quitar y clic en Aceptar.