

Instalación de paquetes

- Instalación de paquetes
- 1. Noción de paquete
- 2. Gestor de paquetes DPKG
- Ejemplo: instalación de google chrome
- Gestion de dependencias
- 3. Programa de gestión de paquetes avanzada APT y YUM
- 4. Repositorios
 - Lista de repositorios
- 5. Prácticas con el gestor APT
- Mostrar que paquetes se pueden actualizar:
- Tareas
- 6. Cliente gráfico synaptic
- 7. Aptitude

1. Noción de paquete

En Linux no es habitual disponer de software proporcionado con un programa de instalación interactivo (install.exe). Con Linux es muy habitual disponer de herramientas, actualizaciones, etc. en forma de **paquetes** (packages).

Un **paquete** es un archivo que contiene un software a instalar y unas reglas. Suele tener una extensión .rpm, .deb, en función del tipo de paquete. Cada tipo de distribución puede funcionar con un tipo de paquetes diferentes. Los paquetes son incompatibles entre sí, pero se pueden convertir.

2. Gestor de paquetes DPKG

El comando **dpkg** permite instalar, crear, eliminar y gestionar paquetes debian. Estos paquetes llevan una extensión .deb. El paquete se tiene que descargar manualmente y si existen dependencias incumplidas, se deben resolver previamente.

Comandos comunes:

- Instalación: `dpkg -i mipaquete.deb`
- Listar paquetes instalados: `dpkg -l`
- Borrar paquetes: `dpkg -r`

Ejemplo: instalación de google chrome

Descarga del paquete:

```
wget -c https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`
```

Instalación del paquete

```
dpkg -i google-chrome-stable_current_amd64.deb
```

Borrar el paquete descargado (si queremos)

```
rm google-chrome-stable_current_amd64.deb
```

Gestion de dependencias

Los paquetes generalmente dependen de otros paquetes mas basicos para funcionar. Estos paquetes necesarios se llaman **dependencias**. Pueden suceder dos cosas:

- Que todas las dependencias que necesita el paquete esten instaladas en la maquina. En este caso el paquete se instalara correctamente.
- Que falte alguna dependencia. Deberemos instalar primero los paquetes necesarios previamente.
- Por ultimo, puede ser que el paquete necesario se encuentre instalado pero no en la version necesaria. Para ello deberemo actualizarlo primero.

3. Programa de gestión de paquetes avanzada APT y YUM

Tanto **dpkg** como **rpm** detectan dependencias de paquetes para autorizar o no la instalación, pero no las resuelven. Si una dependencia de un paquete no se encuentra, se deben instalar previamente.

YUM o **APT** son gestores de paquetes. Descargan los paquetes de un repositorio y gestionan las dependencias automáticamente. Las herramientas **DPKG** o **RPM** no resuelven las dependencias, si hay alguna detienen la instalación e informan de las dependencias que no están cubiertas.

- **YUM** se utiliza en distribuciones Red Hat, Fedora, etc y gestiona paquetes **.rpm**
- **APT** se utiliza en sistemas Debian y derivados y gestiona paquetes **.deb**

4. Repositorios

Un **repositorio** es un lugar que contiene un conjunto de paquetes. Los repositorios de paquetes pueden estar situados en diferentes sitios, como pueden ser:

- En un CD, DVD
- En un directorio local
- En internet (HTTP, FTP)

Cuando se instala un paquete desde un repositorio, se instalan también sus dependencias.

Para mostrar la lista de repositorios:

```
cat /etc/apt/sources.list
```

Para modificar el repositorio:

```
nano /etc/apt/sources.list
```

Lista de repositorios

```
deb http://ftp.cixug.es/mint/packages tessa main upstream import backport

deb http://mirror.tedra.es/ubuntu bionic main restricted universe multiverse
deb http://mirror.tedra.es/ubuntu bionic-updates main restricted universe
multiverse
deb http://mirror.tedra.es/ubuntu bionic-backports main restricted universe
multiverse

deb http://security.ubuntu.com/ubuntu/ bionic-security main restricted universe
multiverse
deb http://archive.canonical.com/ubuntu/ bionic partner
```

Cómo conocer la versión de SO

```
lsb_release -a
```

En mi caso, nos devuelve:

```
No LSB modules are available.
Distributor ID: LinuxMint
Description:    Linux Mint 19.1 Tessa
Release:       19.1
Codename:      tessa
```

5. Prácticas con el gestor APT

- Para buscar un paquete: `apt-cache search nombre`
- Actualizar lista de paquetes: `apt update`

Mostrar que paquetes se pueden actualizar:

Para mostrar los paquetes:

```
apt list --upgradable
```

```
firefox/tessa 66.0.3+linuxmint1+tessa amd64 [actualizable desde:
66.0.2+linuxmint1+tessa]
```

- Simulación de actualización: `apt upgrade (cancelar al final)`
- Buscar si un paquete está instalado o no: `apt list -installed`
- Mostrar información de un paquete: `apt-cache show apache2`
- Volver a mirar si está instalado: `apt list -installed | grep Apache`

- Comprobar que ha abierto el Puerto 80: `sudo netstat -ntpa`
- Desinstalar apache2: `apt remove apache2`
- Desinstalar borrando archivos de configuración: `apt purge`
- Borrar paquetes no necesarios: `sudo apt autoremove`

Tareas

Las **tareas** son grupos de paquetes que proporcionan un modo sencillo de seleccionar un conjunto predefinido de paquetes.

Por ejemplo, la tarea **LAMP server** instala los siguientes paquetes:

Buscar paquetes que comienzan con apache:

```
aptitude search ^apache
```

6. Cliente gráfico synaptic

Synaptic es un front-end o interfaz gráfica que llama a las funciones de APT.

7. Aptitude

Es un administrador de paquetes con una interfaz en modo texto.