



# **Paquetes de Software**

**Laboratorio de Sistemas Operativos II**

# PAQUETES

La mayoría de las distribuciones Linux usan programas o conjuntos de programas prefabricados denominados **paquetes**, que se obtienen listos para ser instalados en dicha distribución.

**Un paquete es un conjunto de archivos que contiene lo necesario para instalar, desinstalar y ejecutar un programa en particular.**



# Tipos de paquetes

- **Paquetes Binarios:** contienen los archivos para reconstruir una aplicación dentro del sistema. Es software precompilado. Se reconocen por ser **.deb (dpkg)** **.rpm (yum)**
- **Paquetes de Código Fuente:** permiten crear paquetes binarios. Son archivos comprimidos y se reconocen por ser **.tgz** **.tar.gz** **.tar.bz2**



# Contenido de paquetes

- Un programa puede estar contenido en un solo paquete, pero también es común que un programa consista de varios paquetes **relacionados** entre ellos.
- También es posible que varios programas pequeños y relacionados entre sí se encuentren en el mismo paquete: por ejemplo, el paquete **fileutils** que contiene varias órdenes de Unix, tales como ls, cp, etc.
- Algunos paquetes requieren de otros para funcionar. En Debian, algunos paquetes pueden depender de otro, recomendar, sugerir, romper, o entrar en conflicto con otros paquetes..

# Dependencia de paquetes

- Si un paquete A **depende** de otro paquete B, entonces B es necesario para que A funcione correctamente. Por ejemplo, el paquete gimp depende del paquete gimp-data para permitir que el editor gráfico GIMP pueda acceder a sus ficheros críticos de datos.
- Si un paquete A **recomienda** otro paquete B, entonces B ofrece una importante funcionalidad adicional para A que sería deseable en la mayoría de las circunstancias. Por ejemplo, el paquete mozilla-browser recomienda el paquete mozilla-psm, que añade la capacidad para la transferencia segura de datos al navegador web de Mozilla.
- Si un paquete A **sugiere** otro paquete B, entonces el paquete B ofrece a A una funcionalidad que puede que mejore A, pero que no es necesaria en la mayoría de los casos. Por ejemplo, el paquete kmail sugiere el paquete gnupg, el cual contiene software de cifrado que KMail puede emplear.
- Si un paquete A entra en **conflicto** con otro paquete B, los dos paquetes no se pueden instalar a la vez. Por ejemplo, fb-music-hi entra en conflicto con fb-music-low porque ofrecen conjuntos alternativos de sonidos para el juego Frozen Bubble.

# ¿Cómo gestionar Paquetes?

- ✓ Un **Gestor de Paquetes** asiste al usuario en la tarea de administrar el conjunto de paquetes
- ✓ Mantiene un registro del software que está instalado en el equipo, y permite instalar software nuevo, actualizarlo a versiones más recientes, o eliminar software de una manera sencilla.
- ✓ Es una colección de herramientas (comandos) que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software.
- ✓ Los **paquetes** incluyen, además del SW mismo, nombre completo, descripción de su funcionalidad, número de versión, y una lista de otros paquetes requeridos para el funcionamiento del SW. El Gestor de Paquetes utiliza esta información.



# Repositorios

- ✓ El software se pone a disposición de los usuarios en los **repositorios**, con el fin de proporcionar un sencillo control sobre los diferentes tipos de software que van a instalar en su sistema
- ✓ Cada distribución mantiene organizados los paquetes en repositorios, lo que permite actualizar e instalar por red todo el sistema desde una localización confiable (repositorios oficiales).
- ✓ Son directorios en servidores especiales que únicamente mantienen paquetes e información de estos, en una estructura similar a una base de datos.
- ✓ Podemos utilizar estos repositorios para actualizar los programas instalados en nuestro sistema, o para instalar programas que no se encuentren en nuestra distribución.
- ✓ Pueden ser de acceso público, o pueden estar protegidos y necesitar de una autenticación previa. Los depósitos más conocidos son los de carácter académico e institucional.



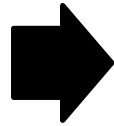
# Diferencias entre SGP e Instalador

SGP	Instalador
Forma parte del sistema operativo.	Cada producto viene unido a su propio instalador
Usa una única base de datos de instalación	Rastrea su propia instalación
Puede verificar y administrar todos los paquetes sobre el sistema	Sólo trabaja con su propio producto.
Un único formato de paquetes	Múltiples formatos de instalación

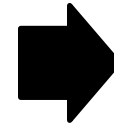


# Flujo de información en SGP

Usuario solicita  
instalación de  
paquete



SGP busca  
paquetes en las  
direcciones que  
tiene.



SGP encuentra  
paquetes  
necesarios



SGP baja los  
paquetes desde  
el repositorio



SGP instala,  
configura y hace  
sugerencias



Usuario puede  
usar software

# Archivo `/etc/apt/sources.list`

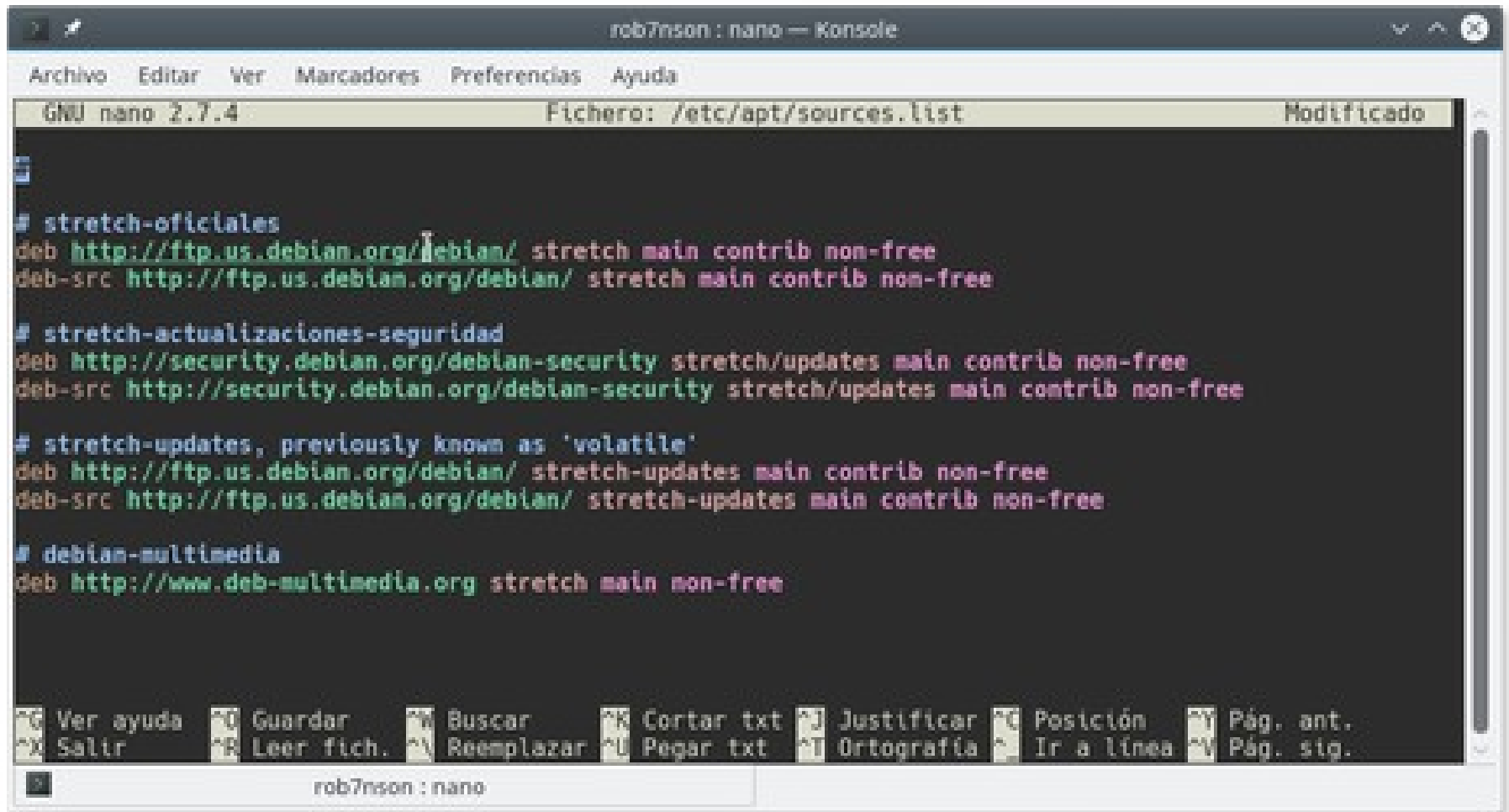
- Es un archivo que contiene una lista de direcciones en donde se encuentran los repositorios de paquetes de la distribución.
- Se puede editar y agregar otras fuentes diferentes a las cargadas durante la instalación.

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

- Luego de editar `sources.list`, siempre actualizar:

```
#apt-get update
```

# #nano /etc/apt/sources.list



The image shows a terminal window with the nano text editor open. The title bar reads "rob7nson : nano — Konsole". The menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Preferencias", and "Ayuda". The status bar at the top shows "GNU nano 2.7.4", "Fichero: /etc/apt/sources.list", and "Modificado". The main editing area contains the following text:

```
# stretch-oficiales
deb http://ftp.us.debian.org/debian/ stretch main contrib non-free
deb-src http://ftp.us.debian.org/debian/ stretch main contrib non-free

# stretch-actualizaciones-seguridad
deb http://security.debian.org/debian-security stretch/updates main contrib non-free
deb-src http://security.debian.org/debian-security stretch/updates main contrib non-free

# stretch-updates, previously known as 'volatile'
deb http://ftp.us.debian.org/debian/ stretch-updates main contrib non-free
deb-src http://ftp.us.debian.org/debian/ stretch-updates main contrib non-free

# debian-multimedia
deb http://www.debian-multimedia.org stretch main non-free
```

The bottom status bar displays various keyboard shortcuts for nano, such as "Ver ayuda", "Guardar", "Buscar", "Cortar txt", "Justificar", "Posición", "Pág. ant.", "Salir", "Leer fich.", "Reemplazar", "Pegar txt", "Ortografía", "Ir a línea", and "Pág. sig.". The window title at the bottom is "rob7nson : nano".

**deb** <http://deb.debian.org/debian> **stretch** **main**

**tipo**: Formato de los argumentos, pueden ser de dos tipos: deb (típico archivo de Debian de dos niveles, que son distribución y componente) y deb-src. (código fuente de la distribución, son necesarias si queremos descargar un índice de los paquetes que tienen el código fuente disponible, entonces de ésta forma obtendremos los códigos originales)

**uri**: Identificador Universal de Recursos, tipo de recurso de la cual se obtienen los paquetes.

- CD-ROM: El cdrom permite a APT usar la unidad de CD-ROM local. Se puede usar el programa apt-cdrom para añadir entradas de un cdrom al fichero sources.list de manera automática, en modo consola.
- FTP: Especifica un servidor FTP como archivo.
- HTTP: Especifica un servidor HTTP como archivo.
- FILE: Permite considerar como archivo a cualquier fichero en el sistema de ficheros. Esto es útil para particiones montadas mediante NFS (sistema de ficheros usado para montar particiones de sistemas remotos) y réplicas locales.

**distribución**: Distribución.

**componentes**: Los componentes son los tipos de repositorios clasificados según las licencias de los paquetes que contienen. Dentro de los componentes tenemos main, contrib y non-free, para usuarios Debian

# Componentes

## main

En este directorio se encuentran los paquetes 100% libres, esto quiere decir que cumplen o están de acuerdo con las directivas de Debian, en donde marcan cuando un paquete se le puede considerar que es 100% software libre.

## non-free

Aquí se encuentran paquetes que no pueden considerarse software libre según las directivas de Debian, por dar un ejemplo, hay software que puede ser distribuido e instalado, pero no se tiene acceso a su código fuente (No todos de esta sección son así hay software que si se proporciona su código fuente), simplemente por la licencia que trae el software de este paquete no cuadra con las directivas de Debian.

## contrib

En este directorio se pueden encontrar software libre, pero depende de alguna forma de un paquete que no es 100% libre

# SGP en Debian

- Debian cuenta con las herramientas:
  - **apt**
  - **aptitude**

# apt

- **A**dvanced **P**ackaging **T**ool (Herramienta Avanzada de Empaquetado). Es un SGP creado por el proyecto Debian.
- Es un grupo de programas que permite gestionar paquetes:
  - instalar o eliminar un paquete (manteniendo **dependencias**),
  - actualizar el sistema,
  - listar paquetes disponibles, etc.
- Es una herramienta potente de alto nivel para traer los paquetes desde lugares remotos.

# Comando apt

- ❏ Baja los paquetes de ordenadores remotos especialmente organizados para ello (**repositorios**), resolviendo e instalando automáticamente todas las dependencias de cada paquete a ser instalado, recomendando la instalación de otros posiblemente relacionados.
- ❏ Casi la totalidad del material que se encuentra en estos repositorios suele ser gratuito.



# Operaciones con paquetes

>> Instalar un paquete:

```
#apt-get install paquete
```

>> Eliminar paquetes y dependientes:

```
#apt-get remove paquete
```

>> Eliminar paquetes y archivos de configuración:

```
#apt-get remove --purge paquete
```

>> Buscar paquetes relacionados a lo que buscamos:

```
#apt-cache search expresion_regular
```

# apt-get

- "apt-cdrom add" Lo primero que hay que hacer si se instala un sistema Debian desde CD, añadir los paquetes de los CDs a la base de datos, pues apt sólo trabaja con lo que tiene en su base de datos.
- "apt-get install <paquete>" Baja el paquete si no esta, junto con todas sus dependencias y los instala o actualiza (según toque); los paquetes bajados se guardan en /var/cache/apt/archives.
- "apt-get remove [--purge] <paquete>" Borra el paquete y todos los que dependan de él. --purge especifica que los paquetes deberían ser purgados (ver dpkg --purge para mas información).
- "apt-get update" Actualiza la lista de paquetes de los mirrors de Debian. Debería usarse cada día si hay que instalar algún paquete aquél día, y cada vez que se modifique /etc/apt/sources.list.
- "apt-get upgrade [-u]" Actualiza todos los paquetes instalados a las últimas versiones disponibles según el último 'update'. No instalará nuevos paquetes ni borrará de viejos. Si un paquete cambia sus dependencias y requiere la instalación de otro paquete, no será actualizado, sino que será puesto en estado 'hold'. apt-get upgrade no actualizará ningún paquete puesto en 'hold'. También es útil la opción '-u' para saber qué paquetes van a ser actualizados.
- "apt-get autoclean" Borra los paquetes que hay en /var/cache/apt/archives cuya versión no está indicada en la base de datos de paquetes (útil para cuando se hace un update y se quieren borrar los paquetes anteriormente bajados cuya versión disponible es ahora superior).
- "apt-get clean" Borra todos los paquetes en /var/cache/apt/archives.
- "apt-cache search <patrón>" Busca un patrón en los nombres de paquetes y sus descripciones.
- "apt-cache show <paquete>" Muestra la descripción completa del paquete.

# APTITUDE

- Es una versión mejorada de "apt" y tiene otra forma de funcionar internamente a la hora de instalar los paquetes resolviendo dependencias. Es más limpia esta forma ya que va creando un registro de lo que vayas instalando y cuando desinstalas algo también elimina las dependencias que no estés usando con ningún otro programa lo que resulta en un sistema con menos paquetes huérfanos y por lo tanto más limpio.

# Sintaxis Aptitude

- Actualizar repositorios:

**# aptitude update**

- Actualizar todos los paquetes:

**# aptitude upgrade**

- Instalar paquete:

**# aptitude install paquete**

- Eliminar paquete:

**# aptitude remove paquete**

- Eliminar paquete con todos sus ficheros de configuración:

**# aptitude remove --purge paquete**

•

# Sintaxis Aptitude (cont.)

- Actualizar sistema con cuidado de dependencias y de forma agresiva:

**# aptitude full-upgrade**

- Bloquear paquete para que no se actualice o elimine:

**# aptitude hold paquete**

- Desbloquear paquete:

**# aptitude unhold paquete**

- Descripción de un paquete:

**# aptitude show paquete**

- Limpiar caché de paquetes:

**# aptitude clean**

- Descargar un paquete al directorio actual:

**# aptitude download paquete**

- Instalar solo las dependencias para un paquete:

**# aptitude build-deps paquete**

# Comando dpkg

- ✓ Es un programa que gestiona paquetes **.deb** , los extrae, analiza, descomprime y proporciona información sobre los paquetes instalados.
- ✓ A diferencia del apt-get, no instala automáticamente las dependencias. Se limita a indicarlo durante el proceso de instalación.

# dpkg

- **dpkg -l** Comprueba los paquetes instalados en la máquina y ofrece un listado completo. Si queremos información relacionada con un solo paquete, se puede utilizar `grep`. Ejemplo: `dpkg -l | grep tree`
- **dpkg -L NombrePaquete** Informa sobre el contenido (los ficheros) que forman un paquete.
- **dpkg -i NombrePaqueteCompleto** Para instalar paquetes que tenemos localmente y no necesitamos descargar
- **dpkg -r NombrePaqueteCompleto** Desinstala el paquete
- **dpkg -P NombrePaquete** Elimina el paquete y archivos de configuración
- **dpkg -s fichero** Informa de los paquetes que contienen ese fichero (a que paquetes pertenece)

# Significado de los caracteres

El primer carácter de cada línea indica el estado deseado del paquete:

- **u** desconocido                      **p** purgar
- **i** instalar                              **h** mantener
- **r** eliminar

El segundo carácter de cada línea indica el estado actual del paquete:

- **n** no instalado
- **i** instalado
- **c** Sólo los archivos de configuración del paquete existen en el sistema.
- **u** El paquete sea desempaquetado, pero no se ha configurado
- **f** fallo configuración.
- **h** Instalación del paquete se ha iniciado pero no terminado por alguna razón.
- **w** El paquete esta a la espera del procesamiento de otro paquete.
- **t** El paquete depende del procesamiento de otros paquetes.

El tercer carácter de cada línea indica el estado de error del paquete:

- **h** mantener.
- **r** requiere reinstalar.
- **x** ambos problemas.



# Uso de apt y dpkg

- En la mayor parte de las ocasiones utilizaremos apt-get o aptitude, dejando dpkg para desinstalar paquetes problemáticos o para instalar paquetes binarios que hemos bajado de Internet sin pasar por los repositorios (.deb) Por ejemplo Libreoffice.

# Paquetes de Código Fuente

- Contienen el código fuente del programa, y vienen con los archivos necesarios para compilar e instalar el programa manualmente.
- Suelen presentarse en formato .tar.gz o tar.bz2 (o sea compactado con tar y comprimido con gzip o bzip).
- Lo normal es que cada aplicación tenga la información en el fichero README o INSTALL de como instalarlo.

# Instalar con Código fuente

- En Debian y derivados, contar con el paquete **build-essential** instalado
- Descargar el **archivo.tar.gz** en un directorio (/opt)
- Situarse en el directorio y descomprimir el archivo: **#tar -xzf archivo.tar.gz**
- Configurar (prepara el equipo para compilar el programa):  
**./configure**
- Compilar el programa: **make** (compilación del código fuente para convertirlo en código de máquina)
- Instalación: **make install** (instala según especificaciones del creador)