



Gestores de paquetes en GNU/Linux Actualización del sistema

Alfredo Abad

ISO-02-12_Ext_InstAplicUbuntu-GestoresPaquetes.pptx

UA: 17-ago-2019

<http://blog.desdelinux.net/estudio-de-los-paquetes-de-debian-parte/>

1

Paquetes de software GNU/Linux

- Un **paquete de software** en un Sistema Operativo GNU/Linux es generalmente un archivo comprimido que:
 - Posee una estructura interna predefinida,
 - Que facilita y permite que el mismo sea manipulado por Herramientas de Gestión de Software (Gestores de Paquetes)
 - Para lograr su compilación y / o instalación, actualización y / o eliminación sobre el Sistema Operativo, de forma cómoda, segura, estable y centralizada
- Un paquete es
 - **Compilable** si su instalación se basa en su código fuente directamente (Ejm. *.tar.gz) o
 - **Instalable** si lo hace en binarios compilados ya para una determinada arquitectura o plataforma (Ejm. *.deb)

2

Composición del paquete y formatos más comunes

- La mayoría de los paquetes vienen con su documentación incluida, sus scripts de pre y post instalación, sus archivos de configuración inicial, sus archivos de recursos, y sus binarios o el código fuente con todo lo necesario si está destinado a ser compilado
- Los formatos de paquetes más conocidos son los
 - **deb** creado para la distro **DEBIAN** y todas sus derivadas,
 - Y los **rpm** creados por **Red Hat** para su propia distro y derivadas como Fedora y Open SUSE

3

Gestión de paquetes en una distro

- Cada distro mantiene almacenada su paquetería en Repositorios, tanto en medios como CDs / DVDs como en Servidores remotos
 - Lo que permite actualizar e instalar por red (Internet) todo o parte del Sistema Operativo desde una localización segura y confiable (Repositorios oficiales)
 - Para no tener que andar buscando servidores desconocidos (e inseguros) a menos que fuese estrictamente necesario

4

Tipos de paquetes por su contenido y gestión de dependencias

- Cada distro suele aportar sus propios paquetes (parches) de seguridad y de mejoras (actualizaciones), para así lograr poner a disposición de sus comunidades de usuarios gran cantidad de software perfectamente funcional e integrado en el Sistema Operativo
- En cuanto a las dependencias de cada paquete, suelen gestionarse de forma automática para evitar posibles problemas a los usuarios menos expertos

5

¿Qué es mejor: compilar o instalar?

- Lo bueno de compilar frente a instalar es la posibilidad de especificar opciones de compilación para tu sistema y software usado que permiten aprovechar mejor los recursos y ajustarse a las preferencias del usuario/administrador, y lo malo lo lento y complicado que puede tornarse este proceso
- Ya que por lo general, instalar un paquete (Ej.: *.deb) es muy rápido y sencillo, pero puede no estar suficientemente actualizado o ajustado a la distro de nuestro uso o recursos de nuestro equipo

6

Repositorios

- Los Repositorios son grandes servidores (externos o internos a la corporación) que actúan como bancos de datos que alojan las aplicaciones (paquetes) que necesitan los sistemas GNU/Linux, que se instalan mediante un Gestor de paquetes
- Su finalidad es mantener al sistema operativo actualizado, sobre todo en materia de seguridad
- Los Repositorios pueden ser de dos tipos:
 - Oficiales
 - No oficiales

7

Localización de la configuración de repositorios en una Distro

- Los Sistemas Operativos (distros) basados en Linux tienen por lo general un archivo en el que se guarda la lista de los repositorios (oficiales o no) a los que podemos acceder mediante el gestor de paquetes para su descarga, instalación, actualización o eliminación
- Este fichero se sitúa generalmente en la ubicación `/etc/nombre_gestor_paquete/` donde “nombre_gestor_paquete” normalmente suele ser el nombre del gestor de paquetes de la Distro
 - Por ejemplo en DEBIAN estaría en `/etc/apt/sources.list`

8

Estructura de un repositorio oficial

- Los repositorios oficiales almacenan los paquetes de las aplicaciones que nuestra distro soporta
- Vienen divididos en una estructura (de ramas y versiones) que, dependiendo de las políticas de sus creadores,
 - garantizan bajo un protocolo de revisión muy riguroso que todos los paquetes que contienen se encuentran en estado óptimo
 - y no representan riesgos de seguridad o estabilidad para el sistema
- Los paquetes nuevos o en desarrollo se alojan en ramas separadas para que puedan ser alcanzables por aquellos usuarios más avanzados o experimentados

9

Las ramas del repositorio DEBIAN

- **Main (Principal):**
 - Rama que almacena todos los paquetes incluidos en la distribución oficial de Debian que son libres de acuerdo a las directrices de software libre de Debian
 - La distribución oficial de Debian se constituye totalmente de esta rama
- **Contrib (Contribución):**
 - Rama que almacena los paquetes con licencia libre, pero con dependencias de otros programas que no son libres
- **Non-Free (No Libres):**
 - Rama que almacena los paquetes que tienen alguna condición de licencia onerosa que restringe su uso o redistribución

10

Versiones de los repositorios DEBIAN

(supone versión actual Jessie)

- **OldStable (Vieja Estable):**
 - Versión que almacena los paquetes pertenecientes a la antigua versión estable de DEBIAN
 - Actualmente esta pertenece a la versión Wheezy
- **Stable (Estable):**
 - Versión que almacena los paquetes pertenecientes a la actual versión estable de DEBIAN
 - Actualmente esta pertenece a la versión Jessie
- **Testing (Prueba):**
 - Versión que almacena los paquetes pertenecientes a la futura versión estable de DEBIAN
 - Actualmente esta pertenece a la versión Stretch
- **Unstable (Inestable):**
 - Versión que almacena los paquetes pertenecientes a los futuros paquetes que continuamente están en desarrollo y pruebas, que con el tiempo podrían pertenecer a la versión **Testing (Prueba)** de DEBIAN
 - Esta siempre pertenece a la Versión SID.

11

Nota sobre sufijos de versiones

- Muchas veces el nombre de la versión suele ir acompañado del prefijo “**-updates**” o “**-proposed-updates**” para resaltar los **paquetes** ahí almacenados, aunque pertenecen a dicha versión, suelen estar **más actualizados**, ya que provienen (más recientemente) de la **versión inmediatamente superior**
- En otras ocasiones, cuando se trata del **Repositorio de Seguridad**, el prefijo suele ser “**/updates**”

12

Dos tipos de paquetes DEBIAN de acuerdo con sus contenidos

- **deb:**
 - Repositorios que solo contendrán paquetes compilados
- **deb-src:**
 - Repositorios que solo contendrán los códigos fuentes de los paquetes compilados disponibles

13

Ejemplo de repositorios DEBIAN

```
#####
# REPOSITARIOS OFICIALES DE LINUX DEBIAN 8 (JESSIE)
deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main contrib non-free
deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
# deb http://ftp.us.debian.org/debian/ jessie-proposed-updates main contrib non-free
# deb http://ftp.us.debian.org/debian/ jessie-backports main contrib non-free
# deb http://www.deb-multimedia.org jessie main non-free
# aptitude install deb-multimedia-keyring
# #####
```

- Nota: Aquellas líneas que se inician con el carácter " # " se encuentran deshabilitados del Gestor de Paquetes. Dicho carácter también se utiliza para insertar comentarios tales como descripciones de las líneas de repositorios o la línea de comando que debe ser utilizada para agregar las llaves (keys) del repositorio descrito

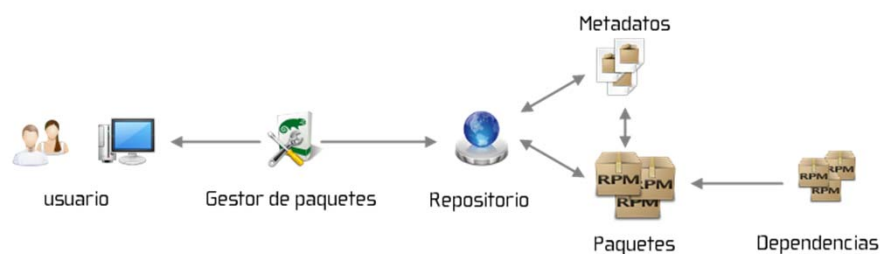
14

Gestores de paquetes

- Con estas herramientas de gestión de paquetes fácilmente se puede conocer, descargar, instalar, actualizar y eliminar cualquier paquete
- En nuestro caso, para DEBIAN, nos concentraremos en los gestores de paquetes **apt-get, aptitude, apt y dpkg**
 - Que usan tanto DEBIAN como las distros que derivan de DEBIAN (como, por ejemplo, Ubuntu)
- El gestor de paquetes puede ser de línea de comandos (habitual), pero también los hay que proporcionan una interfaz gráfica

15

Ejemplo de interacciones entre agentes en la gestión del software mediante repositorios



16

Comandos apt-get (Ayuda: man apt-get)

- Actualizar Listas: apt-get update
- Chequear actualización de Listas: apt-get check
- Instalar paquete: apt-get install nombre_paquete
- Reinstalar paquete: apt-get install --reinstall nombre_paquete
- Actualizar Distro: apt-get upgrade / apt-get dist-upgrade / apt-get full-upgrade
- Actualizar paquete: apt-get upgrade nombre_paquete
- Actualizar paquetes usando dselect: apt-get dselect-upgrade
- Eliminar paquetes: apt-get remove / apt-get autoremove
- Purgar paquetes: apt-get purge
- Conocer paquete: apt-cache show nombre_paquete / apt-cache showpkg nombre_paquete
- Listar paquetes: apt-cache search nombre_paquete
- Listar dependencias de un paquete: apt-cache depends nombre_paquete
- Listar paquetes instalados: apt-cache pkgnames --generate / apt-show-versions
- Validar dependencias incumplidas de un paquete: apt-cache unmet nombre_paquete
- Configurar dependencias de un paquete: apt-get build-dep nombre_paquete
- Descargar paquetes: apt-get source nombre_paquete
- Corregir problemas post-instalación de paquetes: apt-get install -f
- Forzar ejecución de orden de comando: apt-get comando -y
- Eliminar descargas de paquetes: apt-get clean
- Eliminar paquetes obsoletos y sin usos: apt-get autoclean
- Otros importantes: apt-file update / apt-file search nombre_paquete / apt-file list nombre_paquete

17

Comandos aptitude (Ayuda: man aptitude)

- Actualizar Listas: aptitude update
- Instalar paquete: aptitude install nombre_paquete
- Reinstalar paquete: aptitude reinstall nombre_paquete
- Actualizar Distro: aptitude upgrade / aptitude safe-upgrade / aptitude full-upgrade
- Actualizar paquete: aptitude upgrade nombre_paquete
- Eliminar paquetes: aptitude remove
- Purgar paquetes: aptitude purge
- Listar paquetes: aptitude search nombre_paquete
- Listar paquetes instalados / rotos: apt search [*] | grep "^i" / apt search [*] | grep "^B"
- Configurar dependencias de un paquete: aptitude build-dep nombre_paquete
- Descargar paquetes: aptitude download nombre_paquete
- Corregir problemas post-instalación de paquetes: aptitude install -f
- Forzar ejecución de orden de comando: aptitude comando -y
- Eliminar descargas de paquetes: aptitude clean
- Eliminar paquetes obsoletos y sin usos: aptitude autoclean
- Otros importantes: aptitude (un)hold, aptitude (un)markauto, why, why-not
- Conocer paquete:
 - aptitude show nombre_paquete
 - aptitude show "?installed ?section(fonts)" | egrep '(Paquete|Estado|Versión)'
 - aptitude show "?not(?installed) ?section(fonts)" | egrep '(Paquete|Estado|Versión)'
 - aptitude show "?section(fonts)" | egrep '(Paquete|Estado|Versión)'

18

Comandos apt (Ayuda: man apt)

- Actualizar Listas: apt update
- Instalar paquete: apt install nombre_paquete
- Reinstalar paquete: apt install --reinstall nombre_paquete
- Actualizar Distro: apt upgrade / apt full-upgrade
- Actualizar paquete: apt upgrade nombre_paquete
- Eliminar paquetes: apt remove / apt autoremove
- Purgar paquetes: apt purge
- Conocer paquete: apt show nombre_paquete
- Listar paquetes: apt search nombre_paquete
- Listar paquetes instalados / actualizables: apt list --installed / apt list --upgradeable
- Corregir problemas post-instalación de paquetes: apt install -f
- Forzar ejecución de orden de comando: apt comando -y
- Eliminar descargas de paquetes: apt clean
- Eliminar paquetes obsoletos y sin usos: apt autoclean
- Otros importantes: apt edit-sources

19

Comandos dpkg

- Instalar paquete:
 - dpkg -i nombre_paquete
- Eliminar paquete:
 - dpkg -r nombre_paquete / dpkg --force -r nombre_paquete / dpkg --purge -r nombre_paquete
- Purgar paquete:
 - dpkg -P nombre_paquete
- Descomprimir paquete:
 - dpkg --unpack nombre_paquete
- Conocer paquete:
 - dpkg -c nombre_paquete / dpkg --info nombre_paquete / dpkg -L nombre_paquete
- Buscar archivos de paquetes instalados:
 - dpkg -S nombre_archivo
- Configurar paquetes:
 - dpkg --configure nombre_paquete / dpkg --configure --pending / dpkg --configure -a
- Listar paquetes:
 - dpkg -l patrón_búsqueda / dpkg --get-selections nombre_paquete / dpkg --get-selections | grep -v deinstall > lista-paquetes-actuales.txt

20



<http://geekland.eu/retener-paquetes-en-debian-y-derivadas/>

La retención o bloqueo de un paquete implica su exclusión de los procedimientos de actualización, con objeto de conservar la versión anterior retenida

BLOQUEO Y DESBLOQUEO DE PAQUETES DE INSTALACIÓN

21

Bloqueo/desbloqueo de paquetes mediante dpkg (ejemplo de paquete caffeine)

- **sudo echo **caffeine** hold | sudo dpkg --set-selections**
 - La instalación quedará bloqueada para apt-get aptitude o el front-end GUI de actualización del sistema
 - Synaptic no bloqueará el paquete porque usa otros ficheros de configuración específicos
- Comprobar estado de paquete
 - **sudo dpkg --get-selections caffeine**
- Desbloquear paquete
 - **echo **caffeine** install | sudo dpkg --set-selections**

```
Terminal - joan@debian: ~
Archivo Editar Ver Terminal Pestañas Ayuda
joan@debian:~$ echo caffeine hold | sudo dpkg --set-selections
joan@debian:~$ sudo dpkg --get-selections caffeine
caffeine
hold
joan@debian:~$
```

22

Al intentar actualizar...

```
Terminal - joan@debian: ~
Archivo Editar Ver Terminal Pestañas Ayuda
joan@debian:~$ sudo apt-get dist-upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Los paquetes indicados a continuación se instalarán de forma automática y ya no son necesarios.
  gnome-power-manager gnome-screensaver libbit-vector-perl libcarp-clan-perl
  libdate-calc-perl libdate-calc-xs-perl libdiscid0 libgee2
  librexp-common-perl libsnappy1 python-musicbrainz2 python3-xlib tsocks
Utilice «apt-get autoremove» para eliminarlos.
Listo
Se instalarán los siguientes paquetes NUEVOS:
  linux-headers-4.1.0-1-amd64 linux-headers-4.1.0-1-common
  linux-image-4.1.0-1-amd64 linux-kbuild-4.1
Los siguientes paquetes se han retenido:
  caffeine db5.1-util
Se actualizarán los siguientes paquetes:
  dnsmasq dnsmasq-base libarchive-zip-perl libgsf-1-114 libgsf-1-common
  libpcsc-lite1 libsqlite3-0 libsqlite3-0:i386 linux-compiler-gcc-4.9-x86
  linux-headers-amd64 linux-image-amd64 linux-libc-dev python-qt4
13 actualizados, 4 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 45,9 MB de archivos.
Se utilizarán 202 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

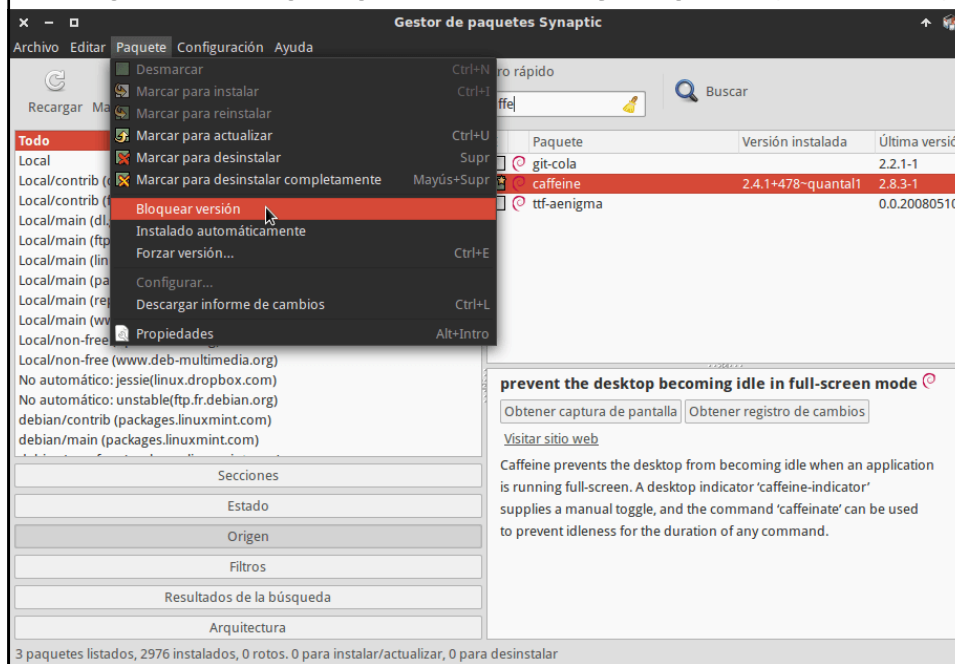
23

Bloqueo/desbloqueo con aptitude

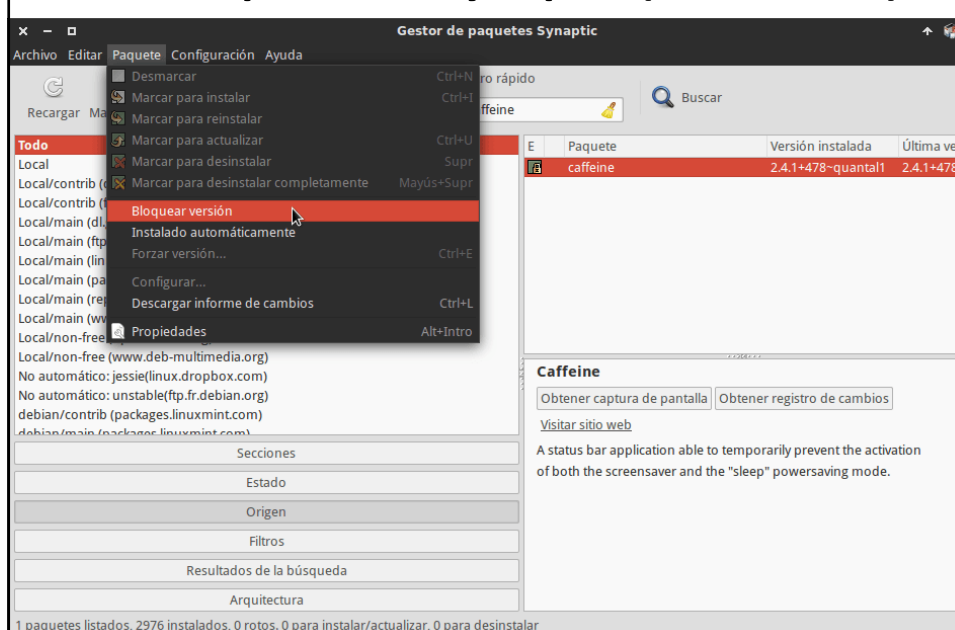
- Bloqueo
 - `aptitude hold <paquete>`
- Desbloqueo
 - `aptitude unhold <paquete>`

24

Bloqueo de paquete con synaptic (marcar)



Desbloqueo con synaptic (desmarcar)



Purgar kernels antiguos

- Buscar los paquetes con Synaptic o similar y eliminarlos con sus dependencia
- O también, utilizar herramientas específicas como purge-old-kernels (Ubuntu a partir de 16.04, desktop o server):
 - **sudo apt-get install byobu**
 - **sudo purge-old-kernels**

27

Modo tradicional de eliminación de kernels antiguos

- Para saber cuál es nuestro kernel actual escribimos:
 - **uname -r**
- Si queremos ver una lista de todos los kernels instalados:
 - **dpkg --get-selections | grep linux-image**
- Para eliminar uno de los kernels escribimos modificando "x" con el número de versión del mismo:
 - **sudo apt-get purge linux-image-x.x.x.x-generic**
- Actualizamos el gestor de inicio del sistema GRUB2:
 - **sudo update-grub2**
- Finalmente reiniciamos el sistema:
 - **sudo shutdown -r now**

28

Otra forma tradicional

- Buscamos kernels instalados
 - **dpkg --get-selections | grep linux-image**
- Eliminamos kernels antiguos
 - Lo recomendable es eliminar los kernels más antiguos que tengas, dejando intactos los 2 o 3 instalados recientemente. Para ello se hace uso del comando:
 - **sudo apt-get remove --purge linux-image-xxxx-generic**
 - Donde el linux-image-xxxx-generic es el kernel que se desea eliminar
- Después se hace algo similar con los headers antiguos:
 - **dpkg --get-selections | grep linux-headers**
 - **sudo apt-get remove --purge linux-headers-xxxx linux-headers-xxxx-generic**
 - linux-headers-xxxx hace referencia al primer header del kernel eliminado
 - linux-headers-xxxx-generic hace referencia al segundo header del kernel eliminado
- Finalmente, se actualiza la configuración de GRUB2
 - **sudo update-grub2**
- Se reinicia el sistema

29



<https://ubuntuforums.org/showthread.php?t=1409558>

<http://www.elmundoenbits.com/2013/02/crear-un-repositorio-local-para.html>

<http://blog.desdelinux.net/usar-repositorio-local-ubuntu/>

CREAR Y UTILIZAR UN REPOSITORIO LOCAL DE PAQUETES DEB EN UBUNTU/DEBIAN

30

Creación del medio de almacenamiento de paquetes

- Debemos crear un directorio que contenga todos los paquetes (.deb) que queramos alojar en el repositorio que vamos a crear
 - Por ejemplo: **mkdir /media/mi-disco/deb/paquetes**
 - Debemos cuidar los permisos de lectura/escritura
- Copiamos a ese directorio todos los ficheros deb
 - **cp /media/mi-disco/deb/*.deb /media/mi-disco/paquetes**
- Creamos el fichero Packages.gz que indica a apt cuál es el contenido del repositorio
 - **sudo dpkg-scanpackages paquetes /dev/null | gzip -c > paquetes/Packages.gz**

31

Uso del nuevo repositorio local

- Editar el archivo sources.list, con:
 - **sudo gedit /etc/apt/sources.list**
 - Añadimos al final de dicho archivo:
 - **deb file:/media/mi-disco/paquetes/**
 - Opcionalmente se pueden añadir los nombres de las ramas del repositorio (por ejemplo, **main universe multiverse restricted**), pero no es imprescindible
 - Ejemplo: **deb file:///media/externo/repositorio-ubuntu lucid main universe multiverse restricted**
 - Guardamos los cambios
- El sistema ya está listo para recargar índices desde Synaptic o hacer **apt-get update**
 - Después ya se podrá hacer **apt-get install**
- Para tener actualizado el índice, cada vez que se modifique el contenido del directorio habrá que reconstruir el fichero Packages.gz

32



<http://www.gigainside.com/instalar-la-misma-lista-paquetes-varias-computadoras/>

INSTALAR LA MISMA LISTA DE PAQUETES EN VARIOS SISTEMAS LINUX

33

Justificación del escenario

- Sirve para ejecutar la misma instalación de paquetes en diversos sistemas o para reinstalar un sistema operativo con el mismo estado que tenía un tiempo atrás
- Instalamos el paquete dpkg-dev:
 - root@sysadmin:~# apt install dpkg-dev
- Obtenemos la lista de todos los paquetes instalados en el equipo tomado como “Plantilla”; o en nuestro propio equipo mediante el comando:
 - root@sysadmin:~# **dpkg --get-selections > lista-de-paquetes**
 - Visualizamos la lista de paquetes con:
 - root@sysadmin:~# **less lista-de-paquetes**

34

Operación

- Guardamos en un medio seguro el archivo lista-de-paquetes, pues lo utilizaremos mas adelante
 - Si estamos recuperando nuestro equipo y dejamos a ese archivo en la carpeta /root, lo mas probable es que al formatear la partición “/” lo perdamos
- Hacemos una instalación básica del sistema operativo sin entorno de escritorio o manejador de ventanas
 - El procedimiento a continuación se encargará de instalar todos los paquetes indicados en el archivo /root/lista-de-paquetes
 - Si entre los paquetes listados en ese archivo se encuentra un entorno de escritorio completo, también se instalará
- En el nuevo equipo Iniciamos sesión como root
- Copiamos el archivo lista-de-paquetes a la computadora que deseamos clonarle los programas instalados en el equipo Plantilla (o hacia el equipo que queremos tener como estaba anteriormente)
- Declaramos los repositorios en el archivo /etc/apt/sources.list y actualizamos la lista de paquetes:
 - root@nuevoequipo:~# **apt update**

35

Operación en el nuevo equipo

- En el nuevo equipo, instalamos el paquete dpkg-dev:
 - root@nuevoequipo:~# **apt install dpkg-dev**
- Registramos la lista de paquetes que vamos a instalar:
 - root@nuevoequipo:~# **dpkg --set-selections < lista-de-paquetes**
- Instalamos los paquetes registrados:
 - root@nuevoequipo:~# **apt-get dselect-upgrade**
- Y esperamos pacientemente a que se instale todo el software
 - Si estamos recuperando la configuración inicial de nuestra máquina y no hemos borrado en nuestra carpeta /home/usuario los archivos y carpetas de configuración cuyos nombres generalmente comienzan con “.”, es muy probable que recuperemos hasta la configuración personalizada del ambiente de escritorio

36

Crear un gráfico de dependencias entre paquetes

- Deben instalarse los paquetes debtree y graphviz
 - **apt-get -y install debtree graphviz**
- Para después ejecutar:
 - **debtree --max-depth 1 linex-xubuntu-core | dot -T png -o linex-xubuntu-core.png**
 - Con debtree generamos la representación del gráfico y con la herramienta dot, el gráfico en formato png.
 - En el fichero png se dibujará el gráfico solicitado



37



Cuando se instala un nuevo sistema operativo, se pueden transportar las aplicaciones y ppa's del sistema antiguo al nuevo mediante la aplicación aptik

Consultar: <http://www.howtogeek.com/206454/how-to-backup-and-restore-your-apps-and-ppas-in-ubuntu-using-aptik/>

BACKUP Y RESTORE DE LAS APPS Y PPA DE UBUNTU USANDO APTIK

38



Es la aplicación equivalente al "System Restore" de Windows, que permite restaurar un sistema a su estado anterior a partir de un snapshot

Consultar: <http://www.howtogeek.com/206491/how-to-restore-your-ubuntu-linux-system-to-its-previous-state/>

RESTAURAR UN SISTEMA UBUNTU A UN ESTADO ANTERIOR CON TIMESHIFT

39



<http://www.ubuntizando.com/instalar-los-ultimos-drivers-nvidia-ubuntu-16-04/>

EJEMPLO: INSTALAR LOS ÚLTIMOS DRIVERS DE NVIDIA EN UBUNTU

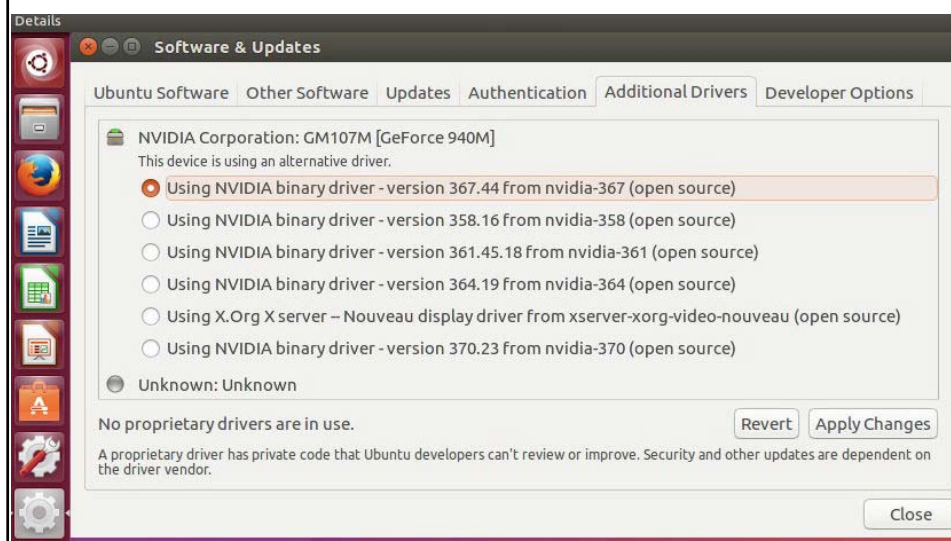
40

Agregación de repositorio e instalación por línea de comandos

- Ubuntu instalará los drivers libres para NVIDIA, pero NVIDIA también tiene una versión open source de los drivers: hay que elegir entre unos (libres, dentro de los repositorios de Ubuntu) u otros (open source de NVIDIA, descargables por ppa)
- Agregamos repositorio privado:
 - **sudo add-apt-repository ppa:graphics-drivers/ppa**
 - **sudo apt update**
- Por línea de comandos
 - Buscar drivers
 - **apt-cache search nvidia**
 - Instalación
 - **sudo apt install nvidia-VERSION**
 - Por ejemplo: **sudo apt install nvidia-370**

41

Instalación mediante gráficos



42

Verificación de la instalación y posibles conflictos con los controladores libres, que provocan “pantallas negras”

- Ejecutamos **lsmod | grep nvidia**
 - Si no hay salida puede ser que haya fallado la instalación o que el controlador no esté disponible en la base de datos de controladores del sistema
- Para ver si lo que estamos ejecutando es el controlador libre, ejecutamos
 - **lsmod | grep nouveau**
 - Si la salida es negativa para nouveau, la instalación de NVIDIA es correcta
 - En caso contrario deberemos prohibir que se ejecuten los controladores libres (diapo siguiente)

43

Añadiendo los controladores libres a una lista negra

- Para crear esta lista negra, vamos a ejecutar el siguiente comando:
 - **nano /etc/modprobe.d/blacklist-nouveau.conf**
- Y en ella vamos a añadir lo siguiente:
 - **blacklist nouveau**
 - **blacklist lbm-nouveau**
 - **options nouveau modeset=0**
 - **alias nouveau off**
 - **alias lbm-nouveau off**
- Reiniciamos el sistema
- Otra de las causantes es por la instalación automática de actualizaciones de versiones menores
 - Esto suele suceder cuando ejecutamos un **apt upgrade**
- Para evitar esto solo hay que ejecutar el siguiente comando, que dejará retenida la actualización del paquete:
 - **sudo apt-mark hold nvidia-370**

44

Desinstalación de los drivers de NVIDIA

- Solo debemos de ejecutar el comando:
 - **sudo apt-get purge nvidia ***
- Y eliminar de la lista negra antes descrita los drivers Nouveau y ejecutamos:
 - **sudo apt-get install nouveau-firmware**
 - **sudo dpkg-reconfigure xserver-xorg**
- Reiniciamos para que los cambios surtan efecto y con ello regresamos nuevamente a los drivers libres

45



ACTUALIZACIONES DEL SISTEMA OPERATIVO Y APLICACIONES DE REPOSITORIOS

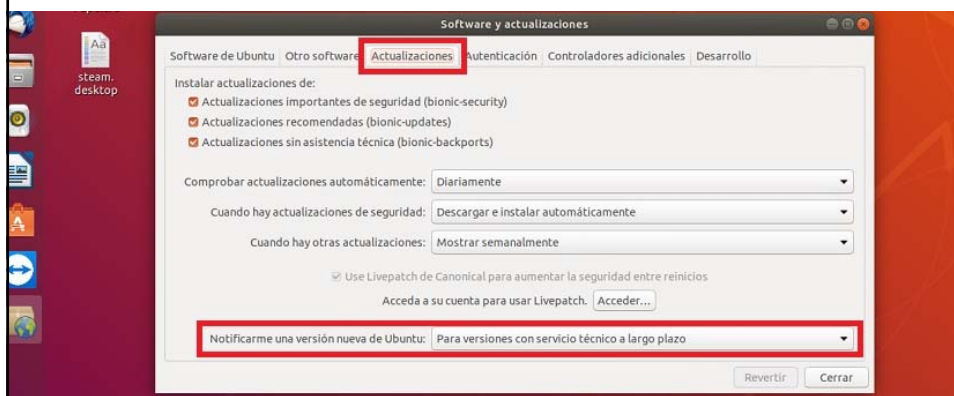
46

Actualizaciones del mismo sistema

- **apt-get update**
 - Actualiza índices de repositorios, pero no descarga software
- **apt-get upgrade**
 - Actualiza todos los paquetes instalados, pero no instala ningún paquete alternativo ni desinstala ninguno de los ya instalados
- **apt-get dist-upgrade**
 - Además de actualizar todos los paquetes instalados, gestiona las dependencias por si la actualización requiere la instalación de nuevos paquetes y la desinstalación de algunos de los ya instalados
- Si vamos a hacer un upgrade del sistema a una versión posterior
 - IMPORTANTE: salvar datos de usuario (\$HOME)

47

Ubuntu gráfico

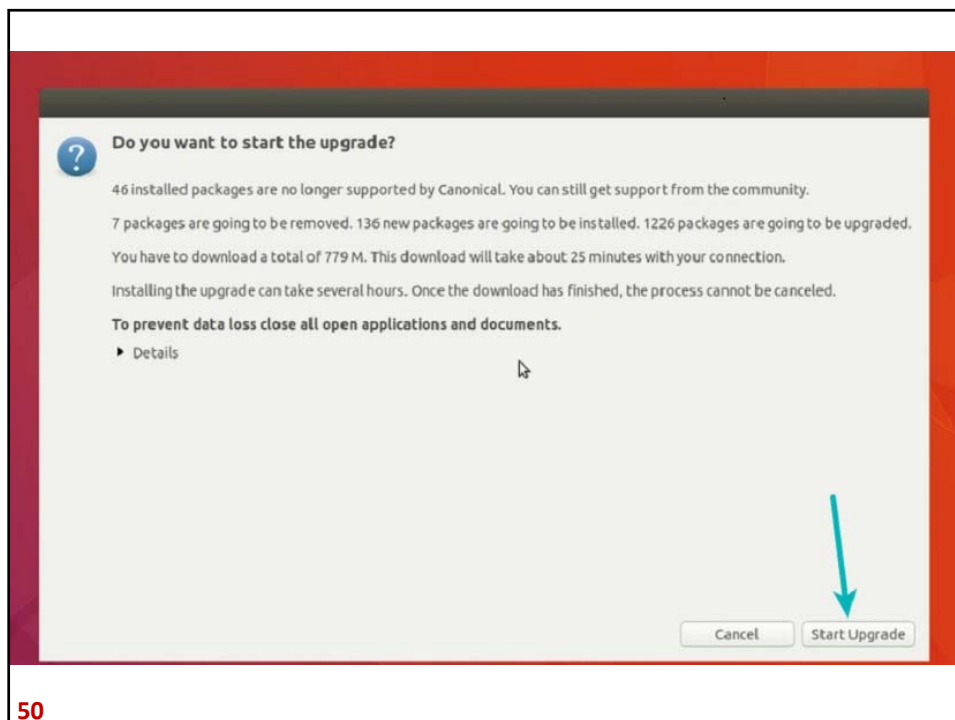


48

Actualización con gestor de actualizaciones (modo gráfico)

- Debemos tener instalado update-manager
 - **sudo apt install update-manager-core**
- Antes de avanzar, actualizamos sistema
 - **sudo apt update**
 - **sudo apt upgrade**
- Ejecutamos el gestor de actualizaciones con:
 - **sudo update-manager -d**
 - Esto abrirá el Actualizador de software y les notificará la disponibilidad de la siguiente versión y haremos clic en el botón de “actualizar” (diapo siguiente)

49



50

Actualizar mediante la línea de comandos

- Para empezar con nuestra actualización a Ubuntu 18.04 LTS debemos de abrir una terminal y ejecutamos los comandos de actualización de los paquetes:
 - **sudo apt update && sudo apt dist-upgrade**
- Puede demorar un poco, si solicita que debe reiniciarse, debe hacerse. Finalizado esto se ejecuta el comando para actualizar a la nueva versión:
 - **sudo do-release-upgrade**
- Si al ejecutar este comando les muestra la siguiente leyenda:
 - *Checking for a new Ubuntu release*
 - *No new release found.*
 - Podemos añadir el siguiente parámetro para actualizar el sistema (versión de desarrollo):
 - **sudo do-release-upgrade -d**

51

Más información

- Tanto gráficamente como por línea de comandos:
 - <https://www.howtogeek.com/351360/how-to-upgrade-to-the-latest-version-of-ubuntu/>

52



<https://www.administradortotal.net/sistemas/linux/desactivar-las-actualizaciones-automaticas-ubuntu/>

BLOQUEAR UPDATES Y UPGRADES AUTOMÁTICOS EN UBUNTU

53

Comandos de bloqueo y desbloqueo

- Bloqueo de updates automáticas:
 - Se edita el fichero `/etc/apt/apt.conf.d/10periodic`
 - **APT::Periodic::Update-Package-Lists "0"**
 - Se desbloquea con:
 - **APT::Periodic::Update-Package-Lists "1"**
- Bloqueo de upgrades automáticas
 - Se edita el fichero `/etc/apt/apt.conf.d/20auto-upgrades`
 - **APT::Periodic::Unattended-Upgrade "0"**
 - Se desbloquea con:
 - **APT::Periodic::Unattended-Upgrade "1"**

54



<https://geekland.eu/reparar-paquetes-rotos-linux/>

GESTIÓN DE PAQUETES ROTOS

55

¿Qué es un paquete roto?

- En el momento de instalar un paquete puede darse el caso que no se instale correctamente o que no sea compatible con el resto de paquetes
 - Si esto pasa es posible que no podamos actualizar, instalar ni desinstalar ningún paquete de nuestro sistema operativo
 - El paquete responsable de que no podamos instalar y desinstalar paquetes es un paquete roto
- Los paquetes acostumbran a romperse en los siguientes casos:
 - Al instalar programas externos a los repositorios oficiales de nuestra distribución
 - Si se interrumpe accidentalmente la instalación de un programa
 - Si se interrumpe accidentalmente la actualización del sistema operativo
 - Etc.

56

¿Qué son las dependencias incumplidas?

- En el momento que un paquete requiere una dependencia que no está presente en los repositorios de nuestra distribución se generará una dependencia incumplida y por lo tanto no podremos instalar y usar el programa que estamos instalando
- En el caso que tengamos paquetes rotos también se generarán dependencias incumplidas o errores de dependencias

57

Reparar paquetes rotos con dpkg y el gestor de paquetes apt

- Lo primero que haremos es intentar configurar paquetes que han sido desempquetados, pero no configurados. Para ello ejecutaremos el siguiente comando en la terminal:
 - `sudo dpkg --configure -a`
- Si no se soluciona el problema limpiaremos la cache local de paquetes .deb ejecutando los siguiente comando en la terminal:
 - `sudo apt-get clean && sudo apt-get autoclean`
- Acto seguido regeneraremos la cache de los repositorios con el parámetro `--fix-missing`. De este modo intentaremos detectar y reparar problemas de dependencias en nuestro sistema operativo. Por lo tanto en la terminal ejecutamos el siguiente comando:
 - `sudo apt-get update --fix-missing`
- Seguidamente usaremos apt para intentar corregir las dependencias rotas. Para ello ejecutaremos el siguiente comando en la terminal:
 - `sudo apt-get install -f`
 - Después de ejecutar el comando se intentará buscar una solución para solucionar nuestro problema de dependencias incumplidas y paquetes rotos.
- En el caso que sigan los problemas borraremos las lock files de apt y dpkg. Para ello ejecutaremos el siguiente comando en la terminal:
 - `sudo rm /var/lib/apt/lists/lock && sudo rm /var/cache/apt/archives/lock && sudo rm /var/lib/dpkg/lock`
 - Nota: Las lock files tienen la función de evitar que 2 o más procesos accedan a los mismos datos.
- Después de ejecutar este último comando se vuelve a repetir la totalidad de comandos citados en este apartado.

58

Reparar paquetes rotos y dependencias con el terminal y dpkg

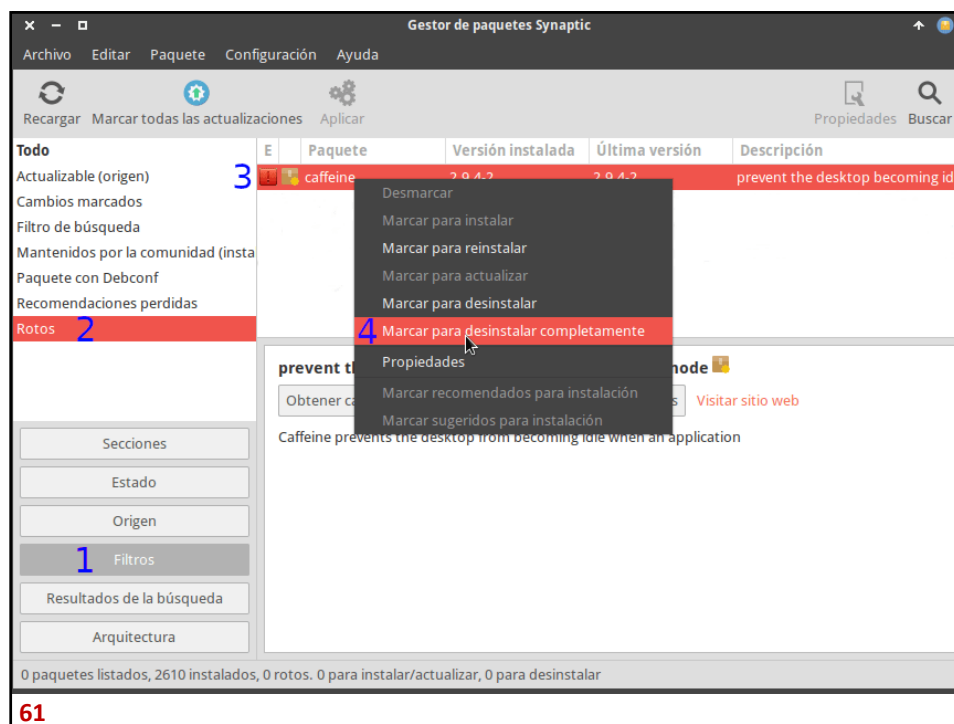
- En el caso que la solución anterior no haya funcionado intentaremos eliminar el paquete roto que está generando problemas. Para ello ejecutaremos el siguiente comando:
 - **sudo apt-get remove**
- En el caso que no tengamos éxito averiguaremos el paquete que está generando problemas ejecutando el siguiente comando:
 - **sudo dpkg -l | grep ^..r**
- Acto seguido ejecutaremos el siguiente comando para intentar forzar la desinstalación del paquete que presenta problemas:
 - **sudo dpkg --remove --force-remove--reinstreq nombre_paquete**
 - Nota: Deben remplazar nombre_paquete por el paquete roto o que está generando problemas.
- A continuación limpiaremos la cache local de nuestro repositorios ejecutando el siguiente comando:
 - **sudo apt-get clean && sudo apt-get autoclean**
- Finalmente intentaremos actualizar nuestro sistema operativo ejecutando el siguiente comando en la terminal:
 - **sudo apt-get update && sudo apt-get dist-upgrade**
- Si el sistema se actualiza sin problema habremos resuelto el problema de dependencias y paquetes rotos.

59

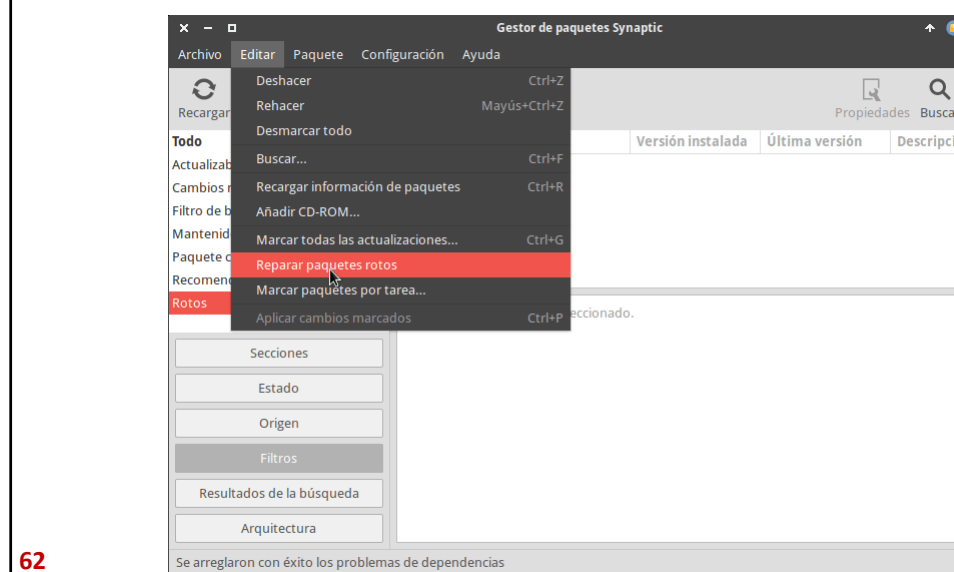
Reparar paquetes rotos con Synaptic package manager

- Otra opción para intentar solucionar el problema es mediante el programa Synaptic
 - Para instalarlo tan solo tienen que ejecutar el siguiente comando en la terminal:
 - **sudo apt-get install synaptic**
- Acto seguido abran el programa y sigan las siguientes instrucciones (imagen en diapo siguiente):
1. Clic en el botón Filtros.
 2. A continuación seleccionar la opción Rotos.
 3. Se selecciona el paquete que está dañado.
 4. Acto seguido clic sobre el botón derecho del ratón y cuando aparezca el menú contextual cliquen sobre la opción Marcar para desinstalar completamente. Acto seguido se intentará desinstalar el paquete que está generando todos los problemas.

60



- En el caso que sigan con los mismos problemas pueden intentar lo siguiente. Justo al abrir el gestor de paquetes Synaptic entren dentro del menú Editar y cliquen sobre la opción Reparar paquete roto. Acto seguido Synaptic interará arreglar los paquetes todos de nuestro sistema operativo.



Último recurso para intentar solucionar problemas de paquetes rotos y dependencias

- Si después de aplicar la totalidad de opciones siguen con problemas se puede intentar lo siguiente.
- Editar el fichero `/var/lib/dpkg/status` ejecutando el siguiente comando en la terminal:
 - **`sudo nano /var/lib/dpkg/status`**
 - Nota: El fichero `/var/lib/dpkg/status` contiene información del estado de los paquetes de nuestro sistema operativo
- Cuando se abra el editor de texto debe buscarse el bloque de información que hace referencia al paquete roto que genera problemas. Una vez encontrado el bloque debe borrarse.
- En mi caso, el paquete que genera problemas es `caffeine`, por lo tanto tendré que borrar todo el bloque que está dentro del rectángulo rojo (ver diapo siguiente).

63

```

x - □ Terminal - joan@debian: ~
GNU nano 3.2 /var/lib/dpkg/status

Package: caffeine
Status: purge ok installed
Priority: optional
Section: misc
Installed-Size: 794
Maintainer: Isaiah Heyer <freshapplepy@gmail.com>
Architecture: all
Version: 2.4.1+478-quantal1
Depends: dconf-gsettings-backend | gsettings-backend, python ($
Description: Caffeine
 A status bar application able to temporarily prevent the acti$
 of both the screensaver and the "sleep" powersaving mode.
Homepage: https://launchpad.net/caffeine
Python-Version: >= 2.6

^G Ver ayuda^O Guardar ^W Buscar ^K Cortar tx^U Justificar
^X Salir ^R Leer fich^N Reemplaza^U Pegar txt^T Ortografía
  
```

- Una vez borrado el contenido guardamos los cambios y cerramos el fichero. Acto seguido podemos intentar actualizar nuestro sistema operativo ejecutando el siguiente comando:
 - **`sudo apt-get clean && sudo apt-get update && sudo apt-get dist-upgrade`**
- Después de seguir los pasos del artículo deberían tener su sistema operativo plenamente operativo. En caso contrario detallen su problema de forma adecuada en los comentarios del artículo dando la información necesaria para que alguien les pueda ayudar.

64



<https://geekland.eu/comprobar-paquetes-soporte-de-seguridad/>

CICLO DE VIDA DE UN PAQUETE

65

El problema de la obsolescencia de los paquetes de una distribución

- A medida que las versiones de Linux (por ejemplo, Debian) se van haciendo obsoletas existen paquetes que dejan de tener soporte de seguridad
- Algunas de las razones por las que esto pasa son las siguientes:
 - En la rama LTS de Debian es posible que algunos de los paquetes dejen de tener soporte
 - Hay que recordar que la rama LTS no es mantenida por el equipo de Debian y en ocasiones hay falta de recursos
 - El desarrollador o el mantenedor del software puede abandonar su creación
 - En el caso que un paquete o programa esté sin soporte durante mucho tiempo será retirado de los repositorios de Debian

66

¿Por qué es importante que el software tenga soporte de seguridad?

- Es importante que los programas y paquetes que usamos en nuestro equipo tengan soporte de seguridad por los siguientes motivos:
 - A diario se descubren bugs y fallos de seguridad que el desarrollador de un paquete o programa debe solucionar
 - Por tanto, para nuestra seguridad es importante que los paquetes tengan soporte
 - Para que los programas puedan ir mejorando de forma progresiva día tras día
 - Siempre es posible optimizar el funcionamiento de un programa
 - Las tecnologías y los sistemas operativos no paran de evolucionar
 - Por tanto, para que un programa funcione adecuadamente es necesario que se vaya actualizando periódicamente
- Una vez somos conscientes de la importancia que nuestros paquetes reciban soporte de seguridad pasaremos a ver un método para saber si nuestro software dispone de soporte y por lo tanto recibe actualizaciones de seguridad

67

Instalación de debian-security-support para detección de paquetes sin seguridad

- Mediante la utilidad **debian-security-support** comprobaremos los paquetes instalados que han dejado de tener soporte de seguridad
- Para instalar **debian-security-support** tenemos que ejecutar el siguiente comando:
 - **sudo apt-get install debian-security-support**
- Una vez instalado el paquete ya podemos pasar a realizar las comprobaciones oportunas

68

Comprobar los paquetes sin soporte con check-support-status

- Para comprobar los paquetes instalados que han dejado de tener soporte tenemos que ejecutar el siguiente comando:
 - **check-support-status**

```

Terminal - joan@jessie: ~
Archivo Editar Ver Terminal Pestañas Ayuda
joan@jessie:~$ check-support-status
Soporte de seguridad limitado para uno o más paquetes

Desafortunadamente, ha sido necesario limitar el soporte de seguridad
para algunos paquetes.

Los siguientes paquetes de este sistema se ven afectados:

* Fuente:kde4libs
Detalles: khtml has no security support upstream, only for use on
ent
Paquetes binarios afectados:
- kdelibs-bin (versión instalada: 4:4.14.2-5+deb8u2)
- kdelibs5-data (versión instalada: 4:4.14.2-5+deb8u2)
- kdelibs5-plugins (versión instalada: 4:4.14.2-5+deb8u2)
- kdoctools (versión instalada: 4:4.14.2-5+deb8u2)
- libkcmutils4 (versión instalada: 4:4.14.2-5+deb8u2)
- libkde3support4 (versión instalada: 4:4.14.2-5+deb8u2)
- libkdeclarative5 (versión instalada: 4:4.14.2-5+deb8u2)
- libkdecore5 (versión instalada: 4:4.14.2-5+deb8u2)
- libkdesu5 (versión instalada: 4:4.14.2-5+deb8u2)
- libkdeui5 (versión instalada: 4:4.14.2-5+deb8u2)
  
```

69

¿Qué hacer con los paquetes que carecen de seguridad?

- Una vez detectados paquetes sin soporte podemos tomar las siguientes decisiones:
 - Puede darse el caso que no haya más remedio que usar los paquetes obsoletos
 - En esto caso podemos tomar la decisión de mantener los paquetes o servicios obsoletos instalados en nuestro sistema operativo
 - En caso que tomemos esta decisión se recomienda ejecutar los programas o servicios en un entorno de ejecución seguro: de este modo evitaremos problemas mayores
 - Reemplazar los paquetes obsoletos por otros que realizan la misma función
 - De esta forma podremos seguir utilizando nuestros servicios y software con mayor seguridad y eficiencia
 - Si los paquetes no son necesarios y no los usamos, lo mejor que podemos realizar es desinstalarlos

70

Desinstalar: una decisión peligrosa

- En el caso de decidamos desinstalar los paquetes hay que ir con cuidado
 - En el momento de desinstalar paquetes es posible que ciertos programas o servicios dejen de funcionar
 - Por tanto es importante conocer la funcionalidad que tienen los paquetes que estamos desinstalando
- Una forma útil para averiguar la utilidad de un paquete es ver sus dependencias y dependencias inversas
 - Si por ejemplo queremos intentar averiguar los programas que usan el paquete libqtwebkit4 podríamos ejecutar el siguiente comando en la terminal:
 - `apt-cache showpkg libqtwebkit4`
- Al aplicar este comando y observando su salida podrán ver que es un paquete necesario para que por ejemplo funcione el programa Skype, Amarok, etc.
 - Por lo tanto al desinstalar libqtwebkit4 es posible que alguna de las funcionalidades de los programas mencionados deje de funcionar

71

```
// Automatically upgrade packages from these (origin:archive) pairs
Unattended-Upgrade::Allowed-Origins {
//   "${distro_id}:${distro_codename}";
//   "${distro_id}:${distro_codename}-security";
//   // Extended Security Maintenance; doesn't necessarily exist for
//   // every release and this system may not have it installed, but if
//   // available, the policy for updates is such that unattended-upgrades
//   // should also install from here by default.
//   "${distro_id}ESM:${distro_codename}";
//   "${distro_id}:${distro_codename}-updates";
//   "${distro_id}:${distro_codename}-proposed";
//   "${distro_id}:${distro_codename}-backports";
};
```

Información de configuración en:

<https://www.howtoforge.com/tutorial/how-to-setup-automatic-security-updates-on-ubuntu-1604/>

Se configuran en /etc/apt/apt.conf.d/50unattended-upgrades

AVANZADO:
¿CONFIGURAR ACTUALIZACIONES
AUTOMÁTICAS EN UBUNTU?

72



Estudiar el documento:

2019_How to Upgrade to Ubuntu 19.04 (Disco Dingo) Server Today.pdf

HOW TO UPGRADE TO UBUNTU 19.04 (DISCO DINGO) SERVER?

73



F-Droid

<https://geekland.eu/instalar-tienda-aplicaciones-f-droid/>

INSTALACIÓN DEL REPOSITORIO F-DROID EN ANDROID

74

¿Qué tiene de especial la tienda de aplicaciones F-Droid

- F-droid es diferente al resto de tiendas porque todas las aplicaciones que contiene son de Software Libre u Open Source
- Dentro de la tienda podremos encontrar el siguiente contenido:
 1. El archivo binario para instalar la aplicación que necesitemos
 2. El código fuente de la aplicación para que lo podamos consultar
 3. Incidencias que los usuarios han reportado al creador de la App
 4. Historial de cambios de la aplicación
 5. Información para poder contactar con el desarrollador la aplicación
 6. Los permisos que otorgamos a las aplicaciones

75

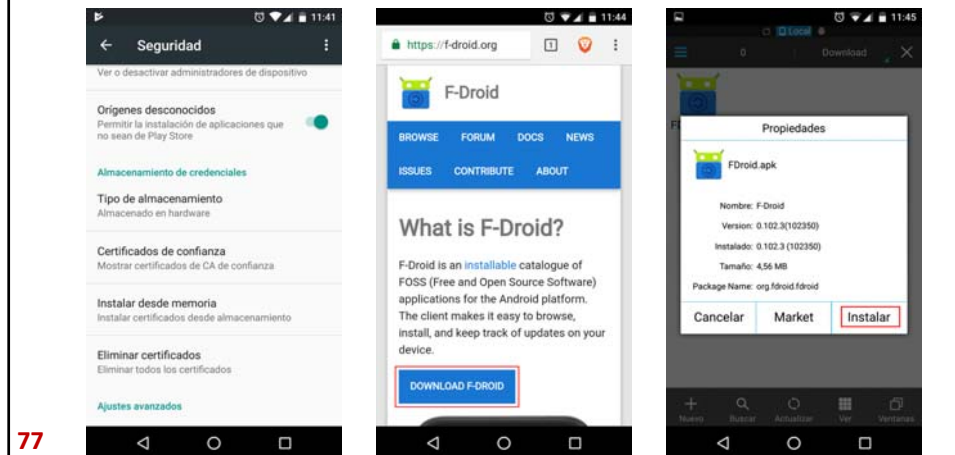
¿Por qué es interesante usar la tienda de aplicaciones de F-Droid?

- Obviamente los grandes atractivos de esta tienda son los siguientes:
 - Podemos consultar el código fuente de todas las aplicaciones. Esto que yo sepa no se puede hacer en otras tiendas como Google Play o la Amazon Store
 - Tenemos la capacidad de ver todas las incidencias y bugs que los usuarios han reportado al desarrollador
 - Podemos contactar fácilmente con los desarrolladores para reportar las incidencias que queramos
 - Podremos encontrar aplicaciones que no están disponibles en tiendas como la Google Play Store
- Todo este conjunto de características hace que las aplicaciones de esta tienda sean seguras, respetuosas con la privacidad y libres de malware

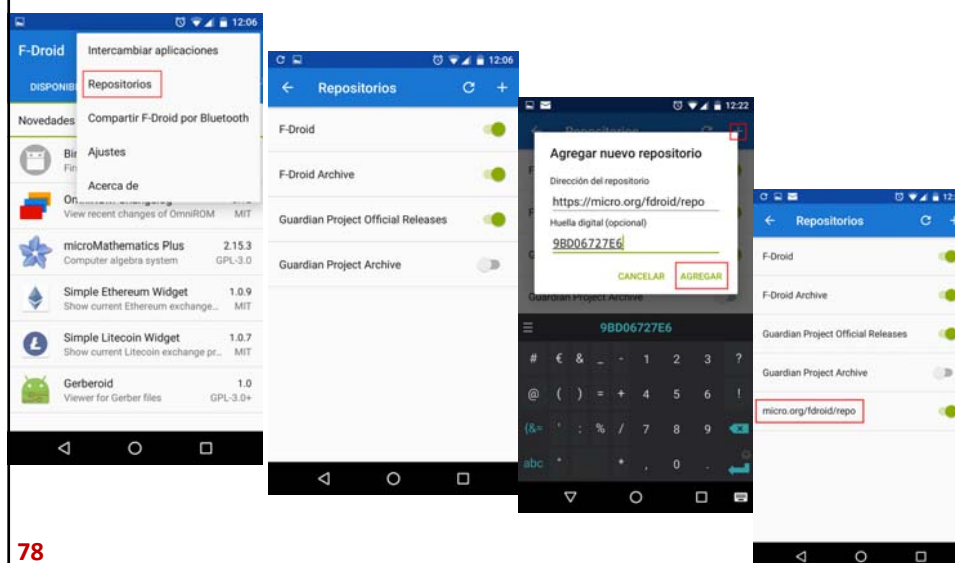
76

Instalación de F-Droid en Android

- Activamos la instalación de aplicaciones de orígenes desconocidos e instalación de la app de la tienda (<https://f-droid.org/>)



Configuración de la tienda y activación de repositorios adicionales (estándar u otros)



**AppImage**

Linux apps that run everywhere

<https://www.atareao.es/opinion/appimage-snap-flatpak/>

NUEVAS TECNOLOGÍAS EN LA GESTIÓN DE PAQUETES: APPIMAGE, SNAP Y FLATPAK

79

Fragmentación de distribuciones y repositorios de software

- Uno de los problemas actuales en el mundo GNU/Linux es la fragmentación
 - Fragmentación no solo en lo que a distribuciones se refiere, si no también a entornos de escritorio y por supuesto a la distribución de aplicaciones
- Así existen diferentes formatos en los que se distribuyen las aplicaciones dependiendo de la distribución que se utilice
 - Así por ejemplo, Debian y derivados utilizan .deb, mientras que Red Hat utiliza .rpm
 - No son los únicos formatos, hay más, por supuesto...
- Para resolver este problema surgen diferentes iniciativas, entre las que se encuentran, en orden cronológico, que no de importancia:
 - AppImage (no soportada por ninguna empresa)
 - Snap (abanderada por Ubuntu)
 - Flatpak (patrocinada por Red Hat)

80

APPIMAGE

- AppImage es un formato que surge para distribuir aplicaciones en Linux sin la necesidad de instalarlas, y sin necesidad de tener permisos de administrador para su ejecución
 - Pero no solo esto, sino que se trata de permitir su funcionamiento con independencia de la distribución Linux en la que se ejecute
- Esta forma de distribuir aplicaciones se lanzó por primera vez bajo el nombre de klik, para ser renombrada en 2.011 a PortableLinuxApps y en 2.013 a AppImage que es como lo conocemos ahora
- Una de las características mas significativas de AppImage con respecto a los otros dos formatos, es que la aplicación no se instala
 - El archivo AppImage es una imagen de la aplicación, de forma que cuando la ejecutamos se monta en el sistema de archivos en el espacio de usuario
 - Tan solo tienes que dar permisos de ejecución y hacer doble clic sobre él
- AppImage, no tiene una tienda de aplicaciones desde donde descargarlas
 - Se puede descargar desde cualquier sitio, darle permisos de ejecución, y ponerla en marcha
 - Sin embargo, existe un lugar donde encontrar y descargar unos cientos de aplicaciones, se trata de AppImageHub (<https://appimage.github.io/apps/>)

81

APPIMAGE:

Appimaged y AppImageUpdate

- Tal como sucede con los otros formatos, se trata de una solución completamente autónoma
 - Es decir, incluye todas las librerías necesarias para que funcione por si solo
- Uno de los problemas con los que podemos encontrarnos con AppImage es que no se integra con el entorno de escritorio, lo cual es una ventaja o un inconveniente
 - Sin embargo, opcionalmente, podemos utilizar un demonio, appimaged, que permite de forma sencilla integrar las aplicaciones AppImage con el sistema
 - Este demonio, se encarga de poner las aplicaciones AppImage en los menús, registrar los tipos MIME, iconos, etc, y todo ello al vuelo
 - Lo único que se necesita es descargarlo, pero es totalmente opcional
- Por otro lado, para la actualización de aplicaciones, también se necesita otro AppImage: AppImageUpdate
 - Se descarga desde <https://github.com/AppImage/AppImageUpdate>
 - Esta herramienta se encarga de revisar las AppImage que tienes instaladas y actualizarlas en el caso de que exista una actualización

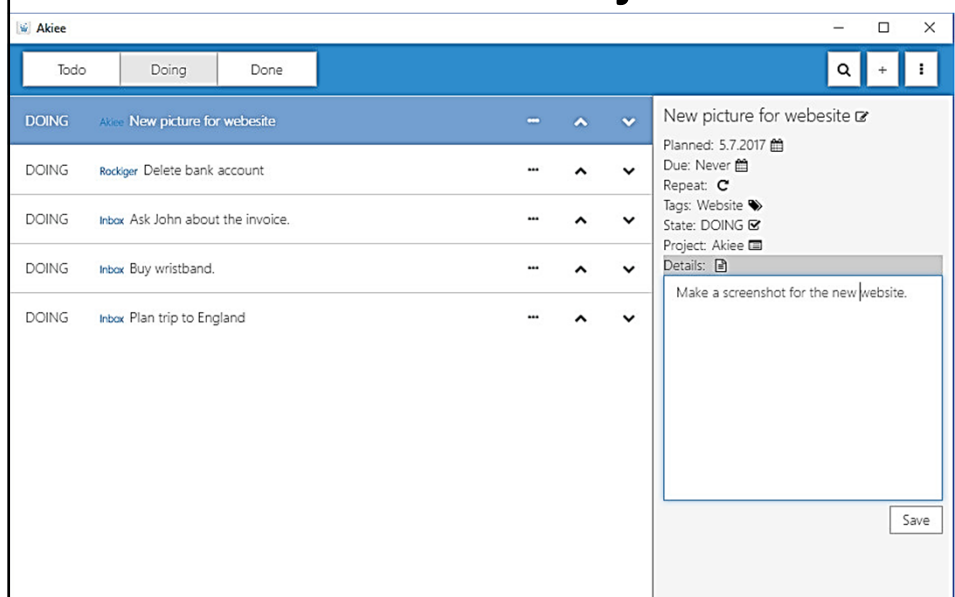
82

Akiee (administrador de tareas): Ejemplo de app instalada con AppImage

- <https://blog.desdelinux.net/akiee-un-administrador-de-tareas-basado-en-markdown/>
- Al estar construida en Electron, la aplicación se distribuye en appimage, con la cual la podremos instalar en cualquier distribución de Linux
- Para instalar este administrador de tareas en nuestro sistema debemos dirigirnos a su web oficial y en su sección descargas encontramos el enlace para obtener el instalador de la aplicación:
 - <https://rockiger.com/en/akiee/>
- Primero debemos de saber que arquitectura es nuestro sistema, si no lo sabes puedes ejecutar este comando:
 - **uname -m**
- Hecha la descarga debemos darle permisos de ejecución al archivo con este comando:
 - **sudo chmod +x Akiee*.appimage**
- Al ejecutar el archivo por primera vez, se les preguntará si desea integrar el programa con el sistema. Si seleccionan que, si desean la integración, el iniciador del programa se agregará al menú de aplicaciones y los iconos de instalación. Si eligen "No", siempre tendrá que iniciarlo haciendo doble clic en el archivo AppImage

83

Vista de Akiee en ejecución



SNAP

- Al igual que sucede con Flatpak, los paquetes Snap son gestionados por una herramienta
 - En este caso, la herramienta es snapd, y es la encargada del despliegue y gestión de paquetes y aplicaciones
 - Esta herramienta está disponible en varias distribuciones (con mas o menos suerte) y por tanto permite el uso de estos paquetes en estas distribuciones
- En diciembre de 2.014 Mark Shuttleworth anuncia snappy, cuya primera versión se libera esa mismo mes
- Los snaps no depende de ninguna tienda de aplicaciones. Se pueden obtener de cualquier fuente, descargados e instalados
- A diferencia de Flatpak, con Snap, no solo tienes aplicaciones de escritorio sino también aplicaciones de línea de comandos y servicios
- 85 • Ubuntu soporta la tecnología SNAP desde la versión 16.04

Ubuntu Software Center: apps de snap



FLATPAK

- Flatpak surge como idea en 2.013 y es en 2.014 cuando se publica un artículo sobre este formato, que originalmente se llamó xdg-app
 - Sin embargo, la primera versión se liberó en Julio de 2.016
 - Se desarrolló como parte del proyecto freedesktop.org.
- Flatpak es una herramienta que nos permite desplegar, gestionar y virtualizar aplicaciones en entornos de escritorio Linux
 - Flatpak está diseñado para funcionar en sesiones de escritorio, en tanto en cuanto utiliza servicios proporcionados por la sesión, como puede ser dbus
 - Proporciona un entorno aislado del sistema, y donde las aplicaciones que utilizan Flatpak necesitan aprobación de permisos para utilizar determinado hardware o acceder a determinados archivos
- Aunque inicialmente, Flatpak fue implementada por desarrolladores unidos a la comunidad GNOME, la realidad, es que este formato no está atado a ningún escritorio en particular
 - De hecho, el objetivo principal, de este formato es el de facilitar que el desarrollador pueda construir sus aplicaciones utilizando cualquier librería y lenguaje de programación que quiera
- En febrero de 2.017 algunas de las aplicaciones mas populares en el universo Linux, ya estaban disponibles en este formato, como pueden ser Blender, Gimp o LibreOffice
 - Actualmente puedes encontrar cientos de aplicaciones disponibles para instalar, utilizando esta tecnología en FlatHub
- Al igual que sucede con el resto de formatos, la ventaja de Flatpak respecto a la forma tradicional de distribuir paquetes, es que con un solo formato, se puede instalar la aplicación en cualquier distribución
 - A esto, le tenemos que unir la ventaja de que el desarrollo y prueba de una aplicación es independiente de la distribución e idéntico al que el usuario tendrá
- Eso sí, Flatpak, instala una serie de runtime para que los puedan utilizar las aplicaciones
 - Y en el caso de que alguna dependencia no se encuentre en esos runtime, siempre se pueden empaquetar en la aplicación

87

Versiones de los gestores en diferentes distribuciones

	AppImage	Flatpak	Snap
Arch Linux	No disponible	0.99.2	No disponible
Debian	No disponible	0.99.2	2.30
Fedora	No disponible	0.99.2	2.33.1
Gentoo	No disponible	0.10.4	2.15.2
Mageia	No disponible	0.10.3	No disponible
openSUSE	9	0.11.8	2.33.1
RHEL/CentOS	No disponible	0.8.8	No disponible
Ubuntu	No disponible	0.11.7	2.32.9

88

Comparativa de características entre Applmage, Snap y Flatpak

General

Característica	Applmage	Snap	Flatpak
Aplicaciones de escritorio	Si	Si	Si
Herramientas de terminal	Si	Si	Si
Servicios	Si	Si	No
Aplicación correcta de temas	Si	No	No
Librerías y dependencias	En la propia imagen	En la propia imagen o con complementos	Uso de runtimes de las librerías principales
Soporte	Ninguno	Canonical	Red Hat y otros

89

Confinamiento

Característica	Applmage	Snap	Flatpak
Sin confinamiento	Si	Si	No
Puede utilizar diferentes confinamientos	Si	No (solo AppArmor)	No (solo Bubblewrap)
Instalación o ejecución			
Característica	Applmage	Snap	Flatpak
Ejecutable	Si	No . Necesita instalación	No . Necesita instalación
Sin root	Si	No. Necesita root para instalar.	No. Necesita root para instalar.
Ejecutable desde comprimido	Si	Si	No

90

Distribución de aplicaciones			
Característica	AppImage	Snap	Flatpak
Repositorio central	AppImageHub	SnapCraft	FlatHub
Necesita repositorio	No	No	No
Repositorios individuales	No	Si	Si
Múltiples versiones en paralelo	Si	Si	Si
Actualizaciones			
Característica	AppImage	Snap	Flatpak
Mecanismo de actualización	AppImageUpdate	Repositorio	Repositorio
Actualizaciones incrementales	Si	Si	Si
Autoactualizaciones	Si	No	No

91

Característica	AppImage	Snap	Flatpak
Aplicación en disco comprimida	Si	Si	No
LibreOffice 6.0.0	200 MB	200 MB	659 MB

92



<https://www.atareao.es/ubuntu/que-son-los-paquetes-snap/>

SNAP

93

SNAP y las dependencias de software

- Seguro que te ha pasado que estás utilizando una aplicación, y al cambiar de versión de Ubuntu, esta aplicación deja de funcionar, o que ni siquiera la puedas instalar: ¿A que se debe esto?
 - A las dependencias
- Esto sucede, porque con los cambios de versiones en Ubuntu, se incorporan nuevas versiones de librerías, que es posible que no sean compatibles con esa aplicación
- Lo que normalmente hacen los desarrolladores es aprovechar las librerías que llevan las distribuciones, con dos objetivos; por un lado para implementar aplicaciones mas rápido, y por el otro para que las aplicaciones no tengan un peso o tamaño excesivo
- El asunto es que actualmente, si una aplicación ocupa algo mas, no es muy relevante, incluso cuando ese algo mas es mucho más
- Los paquetes snap son paquetes que contienen todas las dependencias que necesitan para funcionar correctamente
 - Así, no te tienes que preocupar de que le falta una dependencia o que la dependencia que necesita está obsoleta

94

Características de los paquetes SNAP

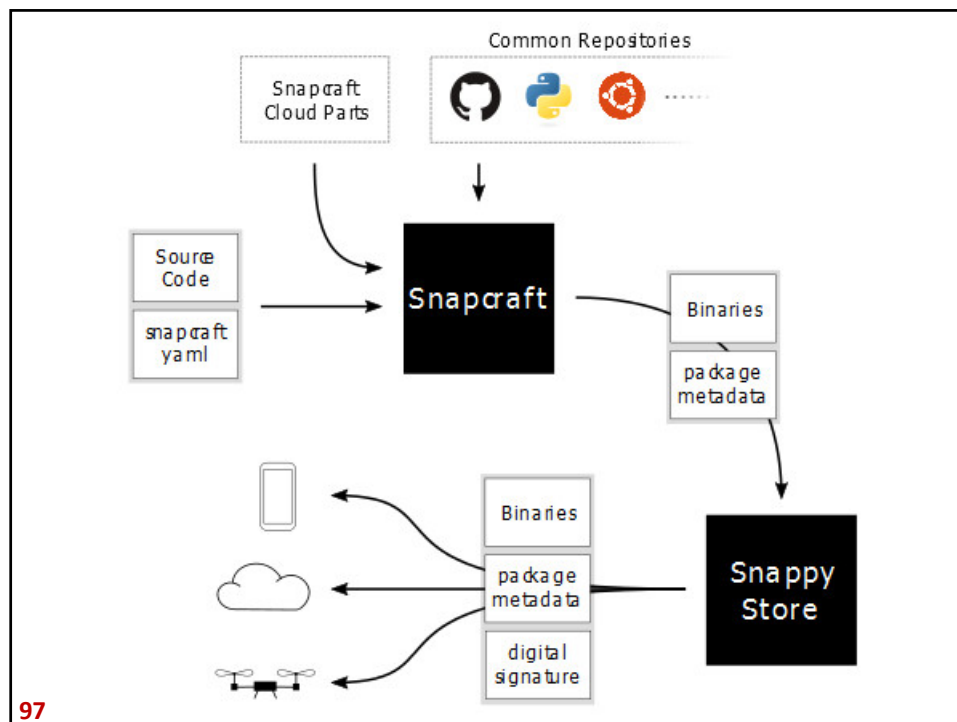
- Los paquetes snap tienen una serie de características que los hacen realmente interesantes:
 - Los paquetes snap son más rápidos de instalar, más fáciles de crear y mas seguros desde el punto de vista de la ejecución
 - Se actualizan de forma automática, de forma que la aplicación que contienen esos paquetes snap siempre está actualizada
 - Los paquetes snap, funcionan en cualquier distribución y dispositivo soportados por snap
 - Desde el internet de las cosas, a servidores, ordenadores de sobremesa y dispositivos móviles

95

¿Qué es un paquete SNAP?

- Los paquetes snap son sistemas de archivos comprimidos de solo lectura squashFS, que contienen el código de tu aplicación y un archivo snap.yaml que contiene información específica del paquete
 - Se trata de sistemas de archivos de solo lectura, que una vez instalados disponen de un área que se puede escribir
- Se trata de paquete autocontenido
 - Esto quiere decir que lleva la mayor parte de las librerías que necesita para su funcionamiento
 - Estos paquetes pueden ser actualizados sin que esto afecte al resto del sistema
- Otro aspecto interesante de los paquetes snap es que están confinados
 - Es decir, la interacción con el sistema operativo o con otras aplicaciones, se hace a través de políticas y permisos controlados por el usuario

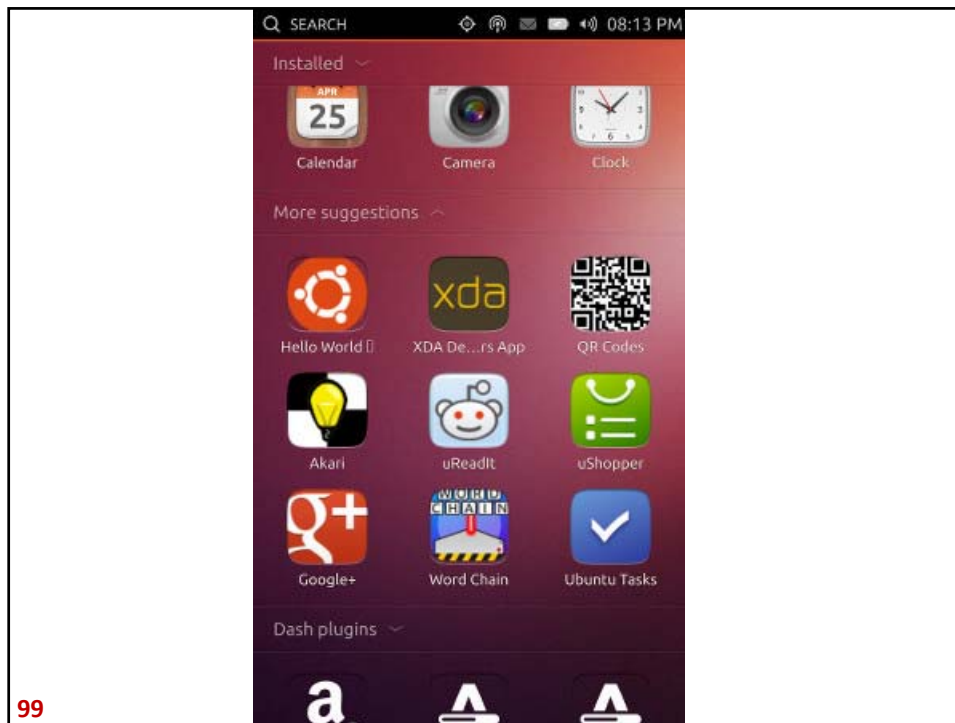
96



Tiendas de paquetes SNAP

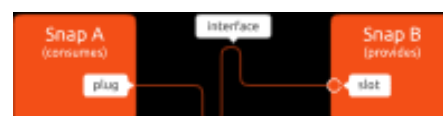
- Existen diferentes maneras de distribuir paquetes snap, y no están asociadas a una distribución concreta
 - Sin embargo, la forma principal para distribuir paquetes snap es desde el Ubuntu Store
 - Allí, cualquier desarrollador puede personalizar los datos e información de sus paquetes snap
- Para los usuarios, la forma mas sencilla de encontrar un snap es desde el uApp Explorer (<https://uappexplorer.com/>)





Interfaces de comunicación entre aplicaciones SNAP y otros recursos

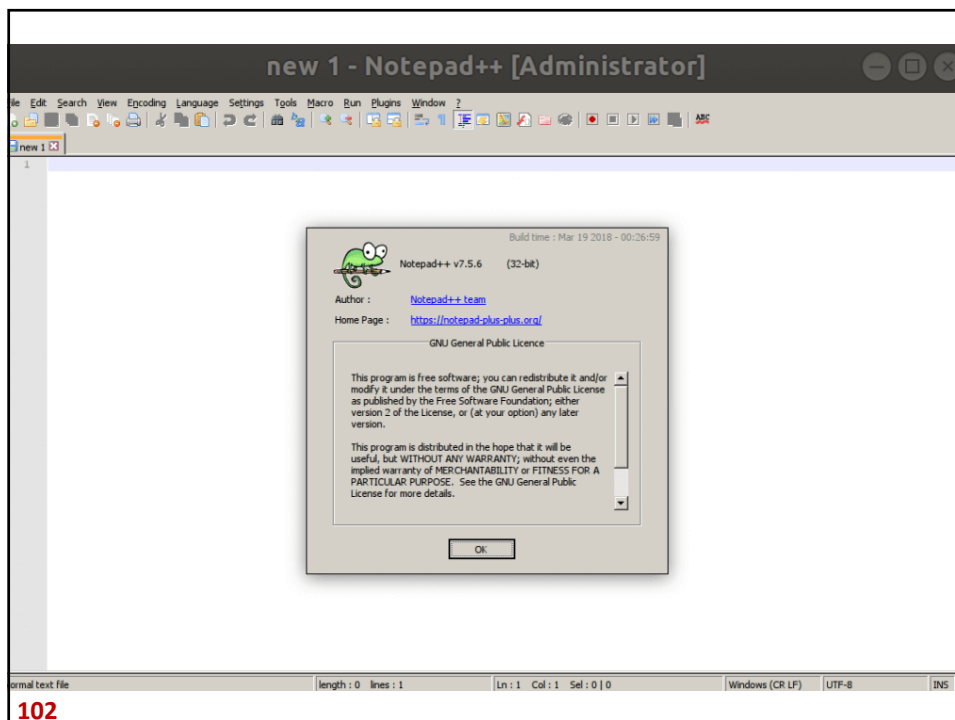
- Lo siguiente a destacar son los permisos, o interfaces
 - Las interfaces, permite a los snaps comunicarse entre sí o compartir recursos conforme al protocolo establecido por la interfaz en cuestión
- Por ejemplo, cuando un snap utiliza la cámara, debe declarar que requiere la interfaz camera
 - Así, en la otra parte de la conexión, el núcleo snap proporciona acceso a la cámara
 - Cuando el interfaz está conectada, el snap tiene acceso de lectura a /dev/video*
- Varios snap se pueden conectar a un mismo slot
 - Por ejemplo todas las aplicaciones que necesitan acceso a la red, deben declarar la interfaz network



Ejemplo: Instalación de Notepad++ vía snap en Ubuntu

- Instalar snap, si no está ya instalado
 - **sudo apt-get install snapd snapd-xdg-open**
- Instalar Notepad++
 - **sudo snap install notepad-plus-plus**
- Instalar plugin obligatorio
 - **sudo snap connect notepad-plus-plus:process-control**
- Instalar plugins opcionales
 - **sudo snap connect notepad-plus-plus:removable-media**
 - **sudo snap connect notepad-plus-plus:hardware-observe**
 - **sudo snap connect notepad-plus-plus:cups-control**
- Ahora desde “Activity Overview”, se puede buscar e iniciar Notepad++

101



102

Best Linux Software Applications



<https://blog.desdelinux.net/listado-aplicaciones-gnu-linux/>

LISTADO DE APLICACIONES ESENCIALES E IMPORTANTES PARA LINUX (2018/2019)

103

Aplicaciones Linux (I)

- **Editores sencillos**
 - [Gedit](#)
 - [Kate](#)
 - [Kwrite](#)
 - [Leafpad](#)
 - [Mousepad](#)
 - [Pluma](#)
- **Editores avanzados**
 - [Atom](#)
 - [Bluefish](#)
 - [BlueGriffon](#)
 - [Brackets](#)
 - [Geany](#)
 - [Glade](#)
 - [Google Web Designer](#)
 - [Kompozer](#)
 - [Lime](#)
 - [Light Table](#)
 - [Notepadqq](#)
 - [Scribes](#)
 - [Sublime Text](#)
- **Editores Mixtos (Terminal/Gráficos)**
 - [Emacs](#)
 - [Vim](#)
- **Entorno de Programación Integrado (IDE)**
 - [Ajusta DeveStudio](#)
 - [Aptana](#)
 - [Arduino IDE](#)
 - [Code::Blocks](#)
 - [Codelite](#)
 - [Eclipse](#)
 - [Gambas](#)
 - [GNAT Programming Studio](#)
 - [JetBrains Suite](#)
 - [KDevelop](#)
 - [Lazarus](#)
 - [NetBeans](#)
 - [Ninja IDE](#)
 - [Python Idle](#)
 - [Postman](#)
 - [QT Creator](#)
 - [Simply Fortran](#)
 - [Visual Studio Code](#)
 - [Wing Python IDE](#)
- **Kit de Desarrollo de Software (SDK)**
 - [.NET Core SDK](#)
 - [Android SDK](#)
 - [Java JDK](#)

104

Aplicaciones Linux (II)

- **Sistemas de Control de Versiones**
- [Bazaar](#)
- [CVS](#)
- [Git](#) / [Clientes Git](#)
- [LibreSource](#)
- [Mercurial](#)
- [Monotone](#)
- [Subversion](#)
- **Entretenimiento**
- **Emuladores de Aplicaciones y Juegos de MS Windows**
- [Crossver](#)
- [Playonlinux](#)
- [Q4wine](#)
- [Wine](#)
- [Winetricks](#)

105

Aplicaciones Linux (III)

- **Emuladores de Juegos de Videoconsolas**
- [Advanced MAME](#)
- [Atari 800](#)
- [Desmume](#)
- [Dolphin](#)
- [DosBox](#)
- [DosEmu](#)
- [ePSXe](#)
- [Fceux](#)
- [Fs-uae](#)
- [GNOME Video Arcade](#)
- [Hatari](#)
- [Higan](#)
- [Kega Fusion](#)
- [Mame](#)
- [Mednafen](#)
- [Nemu](#)
- [Nestopia](#)
- [Pcsxr](#)
- [Pcsxr-df](#)
- [Playonlinux](#)
- [Project 64](#)
- [PPSSPP](#)
- [RPCS3](#)
- [Stella](#)
- [VisualBoy Advance](#)
- [Virtual Jaguar](#)
- [Wine HQ](#)
- [Yabause](#)
- [ZSnes](#)
- **Gestores de Juegos**
- [Game Jolt](#)
- [Itch.io](#)
- [Lutris](#)
- [Steam](#)
- **Juegos**
- [0. A.D.](#)
- [Alien Arena: Warriors of Mars](#)
- [AssaultCube](#)
- [Battle for Wesnoth](#)
- [FlightGear Flight Simulator](#)
- [Freeciv](#)
- [Hedgewars](#)
- [MegaGlest](#)
- [Minetest](#)
- [OpenTTD](#)
- [Red Eclipse](#)
- [Supertux](#)
- [SuperTuxKart](#)
- [Tales of Maj'Eval](#)
- [The Dark Mod](#)
- [Voxelands](#)
- [Warsow](#)
- [Xonotic](#)

106

Aplicaciones Linux (IV)

- **Multimedia**
- **Administración de Sonido del Sistema**
- [Alsa Tools GUI](#)
- [Alsa Mixer GUI](#)
- [Jack](#)
- [Pavucontrol](#)
- [Pulse Audio](#)
- [Pulse Audio Manager](#)
- **Animación 2D/3D**
- [Art of Illusion](#)
- [Blender](#)
- [K-3D](#)
- [Misfit Model 3D](#)
- [Pencil2D](#)
- [Synfig Studio](#)
- [Wings 3D](#)
- **Centros Multimedia**
- [Kodi](#)
- [Penguintv](#)
- [MythTV](#)
- [TVTime](#)
- **Creación de Vídeo con Imágenes y Sonidos**
- [Imagination](#)
- [PhotoFilmStrip](#)
- **Digitalización de Imágenes/Documentos**
- [GScan2PDF](#)
- [SimpleScan](#)
- [Skanlite](#)
- [XSane](#)
- **Diseño CAD**
- [Antimony](#)
- [BricsCAD](#)
- [BRL-CAD](#)
- [CyCAS](#)
- [Draftsight](#)
- [FreeCAD](#)
- [gCAD3D](#)
- [HeeksCAD](#)
- [LibreCAD](#)
- [OpenCascade](#)
- [QCAD](#)
- [SagCAD](#)
- [SolveSpace](#)

107

Aplicaciones Linux (V)

- **Edición de Imágenes**
- [Agave](#)
- [Darktable](#)
- [F-spot](#)
- [Figma](#)
- [Fotobox](#)
- [GIMP](#)
- [Gravit Designer](#)
- [GTKRawGallery](#)
- [ImageMagick](#)
- [Inkscape](#)
- [Krita](#)
- [Kolourpaint](#)
- [LightZone](#)
- [Mypaint](#)
- [Photivo](#)
- [Pinta](#)
- [Pixeluvo](#)
- [Polarr Photo Editor](#)
- [Rawtherapee](#)
- [ShowFOTO](#)
- [UFRaw](#)
- **Edición de Sonido**
- [Ardour](#)
- [Audacity](#)
- [Cecilia](#)
- [Frinika](#)
- [Guitarix](#)
- [Hydrogen Drum](#)
- [LMMS](#)
- [MixXX](#)
- [Openmpt123](#)
- [Qtractor](#)
- [Rosegarden](#)
- [Tracktion](#)
- [Traverso](#)
- [Wavesurfer](#)

108

Aplicaciones Linux (VI)

- **Edición de Vídeo**
 - [Cinelerra](#)
 - [DaVinci Resolve](#)
 - [Flowblade](#)
 - [Fusion](#)
 - [HandBrake](#)
 - [Jokosher](#)
 - [Kdenlive](#)
 - [Lightworks](#)
 - [MKVToolNix](#)
 - [Natron](#)
 - [OBS](#)
 - [Pitivi](#)
 - [OpenShot](#)
 - [Shotcut](#)
 - [VidCutter](#)
- **Gestión de Videocámaras**
 - [Cheese](#)
 - [Webcamoid](#)
- **Gestión de Imágenes de CD/DVD**
 - [Brasero](#)
 - [K3B](#)
 - [Xfburn](#)
- **Maquetadores**
 - [Akira](#)
 - [Alva](#)
 - [Pencil](#)
 - [Scribus](#)
 - [Vectr](#)
- **Reproducción Multimedia**
 - [Atunes](#)
 - [Amarok](#)
 - [Audacious](#)
 - [Banshee](#)
 - [Clementine](#)
 - [Dragón Player](#)
 - [Deepin Music](#)
 - [Exaile](#)
 - [Google Play Music](#)
 - [Harmony](#)
 - [Helix Player](#)
 - [Juk](#)
 - [Kaffeine](#)
 - [Lollypop](#)
 - [Mellow Player](#)
 - [Miro](#)
 - [Mplayer](#)
 - [MPV](#)
 - [Museek](#)
 - [Ncmpcpp](#)
 - [Nightingale](#)
 - [Nuvola Player](#)
 - [Parole](#)
 - [Qmmp](#)
 - [Rhythmbox](#)
 - [Sayonara Player](#)
 - [SMPlayer](#)
 - [Sound Juicer](#)
 - [Tomahawk](#)
 - [Tótem](#)
 - [UMPlayer](#)
 - [VLC](#)

109

Aplicaciones Linux (VII)

- **Gestor de tamaños de Imágenes**
 - [Converseen](#)
 - [Mirage](#)
- **Visualizadores de Imágenes**
 - [DigiKam](#)
 - [Eye of GNOME](#)
 - [Fototox](#)
 - [Gwenview](#)
 - [gThumb](#)
 - [imgSeek](#)
 - [Photonic](#)
 - [PhotoQt](#)
 - [Ristretto](#)
 - [Shotwell](#)
- **Subtitulado de Vídeos**
 - [Aegisub](#)
 - [SubDownloader](#)
 - [Subliminal](#)
- **Ofimática (Hogar y Oficina)**
 - **Administradores de Archivos**
 - [Caia](#)
 - [Double Commander](#)
 - [Dolphin](#)
 - [Konqueror](#)
 - [Krusader](#)
 - [Nautilus](#)
 - [Nemo](#)
 - [PCManFM](#)
 - [Polo](#)
 - [SpaceFM](#)
 - [Thunar](#)
 - [XFE](#)
 - **Administradores de Descarga**
 - [Flareget](#)
 - [JDownloader2](#)
 - [MultiGet](#)
 - [Persepolis](#)
 - [Steadyflow](#)
 - [UGet](#)
 - [XDM](#)

110

Aplicaciones Linux (VIII)

- **Calendarizadores**
 - [California](#)
 - [Day Planner](#)
 - [Evolution](#)
 - [Kalendar](#)
 - [Korganizer](#)
 - [Lightning](#)
 - [Rainlendar](#)
- **Capturadores de Pantalla**
 - [Gnome Screenshot](#)
 - [Flameshot](#)
 - [Kazam](#)
 - [ScreenCloud](#)
 - [Shutter](#)
- **Capturadores de Vídeo de Escritorio**
 - [Byzanz](#)
 - [Kazam Screencaster](#)
 - [RecordMyDesktop](#)
 - [ScreenStudio](#)
 - [SimpleScreenRecorder](#)
 - [Vokoscreen](#)
- **Clientes de Correo Electrónico**
 - [Claws](#)
 - [Evolution](#)
 - [Geary](#)
 - [KMail](#)
 - [Thunderbird](#)
 - [Trojita](#)
 - [Sylpheed](#)
- **Comunicación Personal por Chat**
 - [Chatty](#)
 - [Caprine](#)
 - [Cutegram](#)
 - [Franz](#)
 - [Ghetto Skype](#)
 - [HexChat](#)
 - [Irssi](#)
 - [Keybase](#)
 - [Konversation](#)
 - [Manageryum](#)
 - [Pidging](#)
 - [Quassel](#)
 - [Rambot](#)
 - [ScudCloud](#)
 - [Telegram](#)
 - [Viber](#)
 - [Weechat](#)
 - [YakYak](#)
 - [XChat](#)

111

Aplicaciones Linux (IX)

- **Comunicación Personal por Videoconferencia**
 - [Discord](#)
 - [Jitsi](#)
 - [Empathy](#)
 - [Retrosshare](#)
 - [Ring](#)
 - [Skype](#)
 - [Slack](#)
 - [Viber](#)
 - [Wire](#)
- **Navegadores de Internet**
 - [Brave](#)
 - [Chrome](#)
 - [Chromium](#)
 - [Dillo](#)
 - [Epiphany](#)
 - [Falcon Browser](#)
 - [Firefox](#)
 - [Iron Browser](#)
 - [Konqueror](#)
 - [Maxton](#)
 - [Midori](#)
 - [NetSurf](#)
 - [Opera](#)
 - [Palemoon](#)
 - [SeaMonkey](#)
 - [Tor Browser](#)
 - [Yandex Browser](#)
 - [Vivaldi](#)
- **Gestores de Documentos (Suite Ofimáticas)**
 - [Apache OpenOffice](#)
 - [Calligra](#)
 - [FreeOffice](#)
 - [LibreOffice](#)
 - [OnlyOffice](#)
 - [OxigenOffice](#)
 - [Softmaker](#)
 - [WPS](#)
- **Gestores de Finanzas Personales**
 - [GnuCash](#)
 - [HomeBank](#)
 - [KMyMoney](#)
 - [Skrooge](#)
- **Visores de Documentos PDF**
 - [Evince](#)
 - [Foxit](#)
 - [GNU GV](#)
 - [Llpp](#)
 - [PDFJS](#)
 - [Qpdfview](#)
 - [MuPDF](#)
 - [Okular](#)
 - [XPDF](#)

112

Aplicaciones Linux (X)

- **Notas**
- [Everpad](#)
- [ForeverNote](#)
- [Medleytext](#)
- [Nixnote](#)
- [QOwnNotes](#)
- [Simplenote](#)
- [Standard Notes](#)
- [Whatever](#)
- **Portapapeles**
- [Clipboard Anywhere](#)
- [Clipman](#)
- [Clipit](#)
- [CopyQ](#)
- [Diodon](#)
- [Glipper](#)
- [GPaste](#)
- [Indicator Bulletin](#)
- [Keepboard](#)
- [Klipper](#)
- [Pasio](#)
- [Parcellite](#)
- **Torrents**
- [Deluge](#)
- [KTorrent](#)
- [qBittorrent](#)
- [Tixati](#)
- [Transmission](#)
- [uTorrent](#)
- [Vuze](#)
- [WebTorrent Desktop](#)
- **Seguridad**
- **Antivirus**
- [BitDefender](#)
- [ClamAV – ClamTk](#)
- [ChkrootKit](#)
- [Comodo](#)
- [F-PROT](#)
- [Nod 32](#)
- [RootKit Hunter](#)
- [Sophos](#)
- **Protección Web**
- [GuFW](#)
- **Tecnología de Empaquetamiento de Aplicaciones**
- [Appimages](#)
- [Flatpak](#)
- [Winepak](#)
- [Snap](#)
- **Tiendas de Aplicaciones**
- [AppCenter](#)
- [Appimages](#)
- [Flathub](#)
- [GetDeb](#)
- [Openstore](#)
- [SnapCraft](#)

113

Aplicaciones Linux (XI)

- **Utilitarios de Terminal/Consola**
- **Terminales**
- [Gnome-Terminal](#)
- [Guake](#)
- [Konsole](#)
- [LilyTerm](#)
- [LXTerminal](#)
- [ROXTerm](#)
- [Rxtv](#)
- [Sakura](#)
- [ST](#)
- [Terminator](#)
- [Terminology](#)
- [TermKit](#)
- [WTerm](#)
- [XTerm](#)
- [XTerminal](#)
- [Yakuake](#)
- **Administradores de Archivo**
- [Midnight Commander](#)
- [Nnn](#)
- [Ranger](#)
- [Vifm](#)
- **Administradores de Descarga/Transferencias**
- [AxeI](#)
- [Curl](#)
- [Wget](#)
- **Calendarizadores**
- [Calcurse](#)
- **Clientes de Correo Electrónico**
- [Vmail](#)
- **Editores de Archivos**
- [Diakonos](#)
- [Jed](#)
- [JOE](#)
- [Kakoune](#)
- [LE](#)
- [Micro](#)
- [Mined](#)
- [Nano](#)
- [Nice Editor](#)
- [Pico](#)
- [SETEdit](#)
- **Reproductores Multimedia**
- [Cmus](#)
- [Ffmpeg](#)
- [MOC](#)
- [Mp3blaster](#)
- **Visualizadores de Imágenes**
- [Feh](#)
- **Navegadores de Internet**
- [Links](#)
- [Lynx](#)
- [W3m](#)
- **Gestores de Correo Electrónicos**
- [Alpine](#)
- [Sup](#)
- [Not Much Mail](#)
- [Sendmail](#)
- [Postfix](#)
- [Exim](#)
- [Qmail](#)
- [Mutt](#)
- **Torrents**
- [Aria2](#)

114

Paquetes AppImage

- Son similares a las aplicaciones portables de Windows
- Todos y cada uno de los paquetes AppImage tienen las siguientes propiedades:
 - **Son archivos binarios**, pero en vez de tener la extensión .exe o .dmg tienen la extensión .AppImage.
 - Al ejecutarlos se automonta una imagen ISO comprimida que contiene un programa y la totalidad de librerías para que el programa funcione de forma independiente. Una vez montada la imagen se ejecutará el contenido de la imagen y por lo tanto se abrirá el programa.
 - Un archivo .AppImage **tan solo puede contener un programa**. Al hacer doble clic sobre el paquete se abrirá el programa.
 - Los programas encapsulados dentro de un paquete .AppImage **se pueden ejecutar en la totalidad de distribuciones Linux**.
 - Nos permiten instalar y usar un programa sin necesidad de otorgar permisos de administración.
 - **Contienen la totalidad de librerías para que un programa funcione de forma autónoma** y adecuada. Por lo tanto, los programas se pueden usar sin depender de las librerías que tenga nuestra distribución.
- Por lo tanto los programas encapsulados en paquetes AppImage nos permitirán distribuir y usar programas en cualquier distribución Linux de forma sencilla y segura.
- <https://geekland.eu/instalar-paquetes-appimage-ventajas/>

115

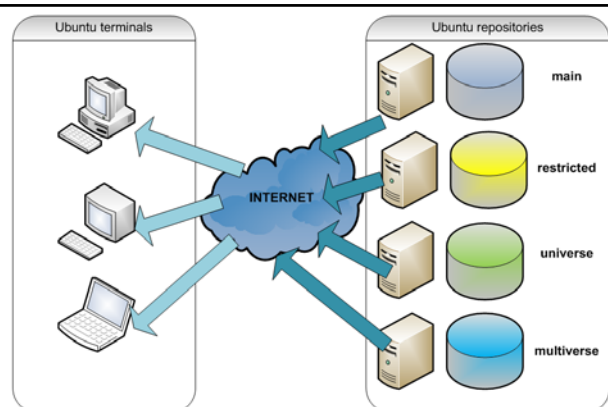


Estudiar los siguientes documentos:

- 2019_Instalar paquetes AppImage en Linux Ventajas, inconvenientes.pdf

INSTALAR PAQUETES APPIMAGE EN LINUX: VENTAJAS E INCONVENIENTES

116



<https://blog.desdelinux.net/repositorios-de-distribuciones-gnu-linux/>

MONOGRÁFICO SOBRE RESPOSITARIOS DE LINUX (BASADO EN DEBIAN)

117

Estructura de un repositorio

Por lo general, un Repositorio estándar tiene una ruta de acceso o configuración similar a la que se muestra a continuación:

```
FORMATO_PAQUETE PROTOCOLO://URL_SERVIDOR/DISTRO/ VERSIÓN RAMAS_PAQUETES
```

Ejemplo de línea de repositorio para DEBIAN Jessie (8):

```
deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
```

Un ejemplo más amplio de un archivo `sources.list` típico, es decir, el archivo de configuración predefinido para guardar las líneas de acceso y configuración de los Repositorios accesibles por una Distro, por ejemplo basada en DEBIAN Jessie (8) sería el siguiente:

```
#####
# REPOSITARIOS OFICIALES DE LINUX DEBIAN 8 (JESSIE)
#
# Repositorio base
deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
# Actualizaciones de seguridad
deb http://security.debian.org/ jessie/updates main contrib non-free
# Actualizaciones para la base estable
deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
# Futuras actualizaciones para la base estable
# deb http://ftp.us.debian.org/debian/ jessie-proposed-updates main contrib non-free
# Retroadaptaciones para la base estable
# deb http://ftp.us.debian.org/debian/ jessie-backports main contrib non-free
# Actualizaciones Multimedia no oficiales
# deb http://www.deb-multimedia.org jessie main non-free
# Llave del Repositorio Multimedia no oficial
# aptitude install deb-multimedia-keyring
#
#####
```

118

Significado de cada campo

• FORMATO_PAQUETE:

1. **deb**: Indica que el Repositorio solo contiene paquetes compilados, es decir, los paquetes de instalación (binarios)
2. **deb-src**: Indica que el Repositorio solo contiene los códigos fuentes de los paquetes compilados disponibles, es decir, los paquetes fuentes.

• PROTOCOLO:

1. **http://** – para indicar un origen disponible en un servidor web
2. **ftp://** – para un origen disponible en un servidor FTP
3. **cdrom://** – para instalaciones desde CD-ROM/DVD-ROM/Blue-ray
4. **file://** – para indicar un origen local instalado en la jerarquía de archivos del sistema

• URL_SERVIDOR:

1. **ftp.xx.debian.org** ==> xx corresponde a país de origen del servidor
2. **nombre_de_servidor** ==> puede ser cualquier otro que contenga DEBIAN.

• DISTRO:

1. **debian**: Aplicable para Sistemas Operativos basados en DEBIAN.
2. **nombre_distro**: Nombre disponible en el Servidor para indicar cualquier otra distro o tipo especial de paquetes que contenga.
3. **vacío**: Muchas veces no hay nada en esta posición, indicando que todo lo existente es para una sola Distro en especial.

119

• VERSIÓN:

En el caso de DEBIAN indica las versiones lanzadas al mercado, por ejemplo:

```
DEBIAN GNU/Linux X ("sid") versión de desarrollo actual (inestable) (sid / unstable).
DEBIAN GNU/Linux 10.0 ("buster") versión de prueba actual (prueba) (stretch / testing).
DEBIAN GNU/Linux 9.0 ("stretch") versión de prueba actual (estable) (stretch / stable).
DEBIAN GNU/Linux 8.0 ("jessie") versión estable actual (vieja estable) (jessie / oldstable)
DEBIAN GNU/Linux 7.0 ("wheezy") antigua versión estable.
DEBIAN GNU/Linux 6.0 ("squeeze") antigua versión estable.
DEBIAN GNU/Linux 5.0 ("lenny") antigua versión estable.
DEBIAN GNU/Linux 4.0 ("etch") antigua versión estable.
DEBIAN GNU/Linux 3.1 ("sarge") antigua versión estable.
DEBIAN GNU/Linux 3.0 ("woody") antigua versión estable.
DEBIAN GNU/Linux 2.2 ("potato") antigua versión estable.
DEBIAN GNU/Linux 2.1 ("slink") antigua versión estable.
DEBIAN GNU/Linux 2.0 ("hamm") antigua versión estable.
DEBIAN GNU/Linux 1.2 ("buzz") antigua versión estable.
DEBIAN GNU/Linux 1.1 ("rex") antigua versión estable.
DEBIAN GNU/Linux 1.0 ("bo") antigua versión estable.
```

<

>

120

Los Repositorios DEBIAN se dividen en Versiones:

1. **OldStable (Vieja Estable):** Versión que almacena los paquetes pertenecientes a la antigua Versión Estable de DEBIAN. Actualmente esta pertenece a la Versión Jessie.
2. **Stable (Estable):** Versión que almacena los paquetes pertenecientes a la actual Versión Estable de DEBIAN. Actualmente esta pertenece a la Versión Stretch.
3. **Testing (Prueba):** Versión que almacena los paquetes pertenecientes a la futura Versión Estable de DEBIAN. Actualmente esta pertenece a la Versión Buster.
4. **Unstable (Inestable):** Versión que almacena los paquetes pertenecientes a los futuros paquetes que continuamente están en desarrollo y pruebas, que con el tiempo podrían pertenecer a la Versión Testing (Prueba) de DEBIAN. Esta siempre pertenece a la Versión SID.

Nota: Muchas veces el nombre de la versión suele ir acompañado del prefijo “-updates” o “-proposed-updates” para resaltar que dichos paquetes ahí almacenados aunque pertenecen a dicha versión suelen ser mas actualizados, ya que provienen más recientemente de la versión inmediatamente superior. En otras oportunidades cuando se trata del Repositorio de Seguridad el prefijo suele ser “/updates”.

121

• RAMAS_PAQUETES:

En el caso de DEBIAN los Repositorios cuentan con 3 ramas:

1. **Main (Principal):** Rama que almacena todos los paquetes incluidos en la distribución oficial de DEBIAN que son libres de acuerdo a las Directrices de Software libre de DEBIAN. La distribución oficial de DEBIAN se constituye totalmente de esta Rama.
2. **Contrib (Contribución):** Rama que almacena los paquetes cuyos creadores les han dado licencia libre, pero los mismos poseen dependencias de otros programas que no son libres, es decir, software de código abierto que no puede funcionar sin elementos privativos. Estos elementos pueden ser software de la sección non-free o archivos privativos como ROMs de juegos, BIOS para consolas, etc.
3. **Non-Free (No Libres):** Rama que almacena los paquetes que tienen alguna condición de licencia onerosa que restringe su uso o redistribución, es decir, contiene software que no sigue (completamente) estos principios pero que aún pueden ser distribuidos sin restricciones.

Para conocer los de cada Distro, muy seguramente debemos consultar las paginas oficiales de cada una, donde seguramente nos ofrecerán datos sobre los mismos, tal como Ubuntu y Mint

122

Compatibilidad entre Repositorios

- Hay una correlación directa de compatibilidad entre el lanzamiento de las diferentes versiones de la Meta-distribución DEBIAN y las basadas o derivadas de ellas, tal como Ubuntu, Mint, MX-Linux, Canaima y MineroS (ver diapo siguiente)
- Esta coincidencia de compatibilidad se da, ya que a medida que la Madre de todas las Distros (DEBIAN) va sacando nuevas versiones con nuevos paquetes y aplicaciones, los mismos van siendo migrados e implementados a otras más pequeñas directamente o adaptados progresivamente a otras más grandes como Ubuntu y de ahí a sus derivadas
- En cada Meta-distribución o Distro Madre y sus derivadas o basadas en ellas habrá una propia y diferente Lista de Equivalencia de Repositorios

123

LISTADO DE EQUIVALENCIA (COMPATIBILIDAD) DE REPOSITORIOS DE DISTRIBUCIONES GNU/LINUX - BLOG DEL PROYECTO TIC - TAC / ING. JOSÉ ALBERT

#	DEBIAN	UBUNTU	MINT	LMDE	MX-LINUX	CANAIMA	MINER-OS
1	1.1 (Buzz)	-	-	PROYECTO TIC TAC	-	-	-
2	1.2 (Rex)	-	-	¡E! que más Aprenda es el que más Conoce.	-	-	-
3	1.3 (Bo)	-	-	-	-	-	-
4	2.0 (Hamm)	-	-	PROYECTO TIC TAC	-	-	-
5	2.1 (Slink)	-	-	-	-	-	-
6	2.2 (Potato)	-	-	-	-	-	-
7	3.0 (Woody)	4.10 (Warty Warthog - Jabali Verrugoso)	-	-	-	-	-
8	3.1 (Sarge)	5.04 (Hoary Hedgehog - Puerco-espín Anciano) 5.10 (Breezy Badger - Tejón Despreocupado)	-	-	-	-	-
9	4.0 (Etch)	6.06 LTS (Dapper Drake - Pato Culto) 6.10 (Edgy Eft - Tritón Crispado) 7.04 (Feisty Fawn - Cervatillo Animado) 7.10 (Gutsy Gibbon - Gibón Valiente)	01.0 - Ada 02.0 - Barbara 03.0 - Cassandra 04.0 - Daryna	PROYECTO TIC TAC	-	1.X (Canaima)	-
10	5.0 (Lenny)	8.04 LTS (Hardy Heron - Garza Robusta) 8.10 (Intrepid Ibex - Cabra Intrepida) 9.04 (Jaunty Jackalope - Jackalope Desenfadado) 9.10 (Karmic Koala - Koala Kármico)	05.0 - Elyssa LTS 06.0 - Felicia 07.0 - Gloria 08.0 - Helena	-	-	2.0 (Menu) 2.1 (Aponwao)	-
11	6.0 (Squeeze)	10.04 LTS (Lucid Lynx - Lince Lucido) 10.10 (Maverick Meerkat - Suricato Inconformista) 11.04 (Natty Narwhal - Narval Elegante) 11.10 (Oneiric Ocelot - Ocelote Onirico)	09.0 - Isadora LTS 10.0 - Julia 11.0 - Katya 12.0 - Lisa	LMDE DEBIAN Testing	14	3.0 (Roraima) 3.1 (Auyantepui)	-
12	7 (Wheezy)	12.04 LTS (Precise Pangolin - Pangolín Preciso) 12.10 (Quantal Quetzal - Quetzal Cuántico) 13.04 (Raring Ringtail - Lemur Ansioso) 13.10 (Saucy Salamander - Salamandra Descarada)	13.0 - Maya LTS 14.0 - Nadia 15.0 - Olivia 16.0 - Petra	LMDE DEBIAN Testing	15	4.0 (Kerepakupai) 4.1 (Kukenan)	-
13	8 (Jessie)	14.04 LTS (Trusty Tahr - Tauro Fiel) 14.10 (Utopic Unicorn - Unicornio Utopico) 15.04 (Vivid Vervet - Mono Araña Vivido) 15.10 (Wily Werewolf -)	17 - Qiana LTS 17.1 - Rebecca 17.2 - Rafaela 17.3 - Rosa	LMDE DEBIAN Testing LMDE2 DEBIAN Stable	16	5.0 (Chimanta) 5.1 (Chimanta)	-
14	9 (Stretch)	16.04 LTS (Xenial Xerus - Xerus Hospitalario) 16.10 (Yakkety Yak - Yak Cotorreador) 17.04 LTS (Zesty Zapus - Ratoncito Vigoroso) 17.10 (Artful Aardvark - Cerdo Hormiguero Ingenioso)	18 - Sarah LTS 18.1 - Serena 18.2 - Sonya 18.3 - Sylvia	LMDE3 DEBIAN Stable	17	6.0 (Kavac)	Beta 0.2
15	10 (Buster)	18.04 LTS (Bionic Beaver - Castor Biónico) 18.10 (JibialSepia Cósmica - Cosmic Cuttlefish)	19 - Tara LTS	LMDE3 DEBIAN Testing	18	7.0 (Sin Asignar)	Beta 0.3 1.0 (Petro)

124