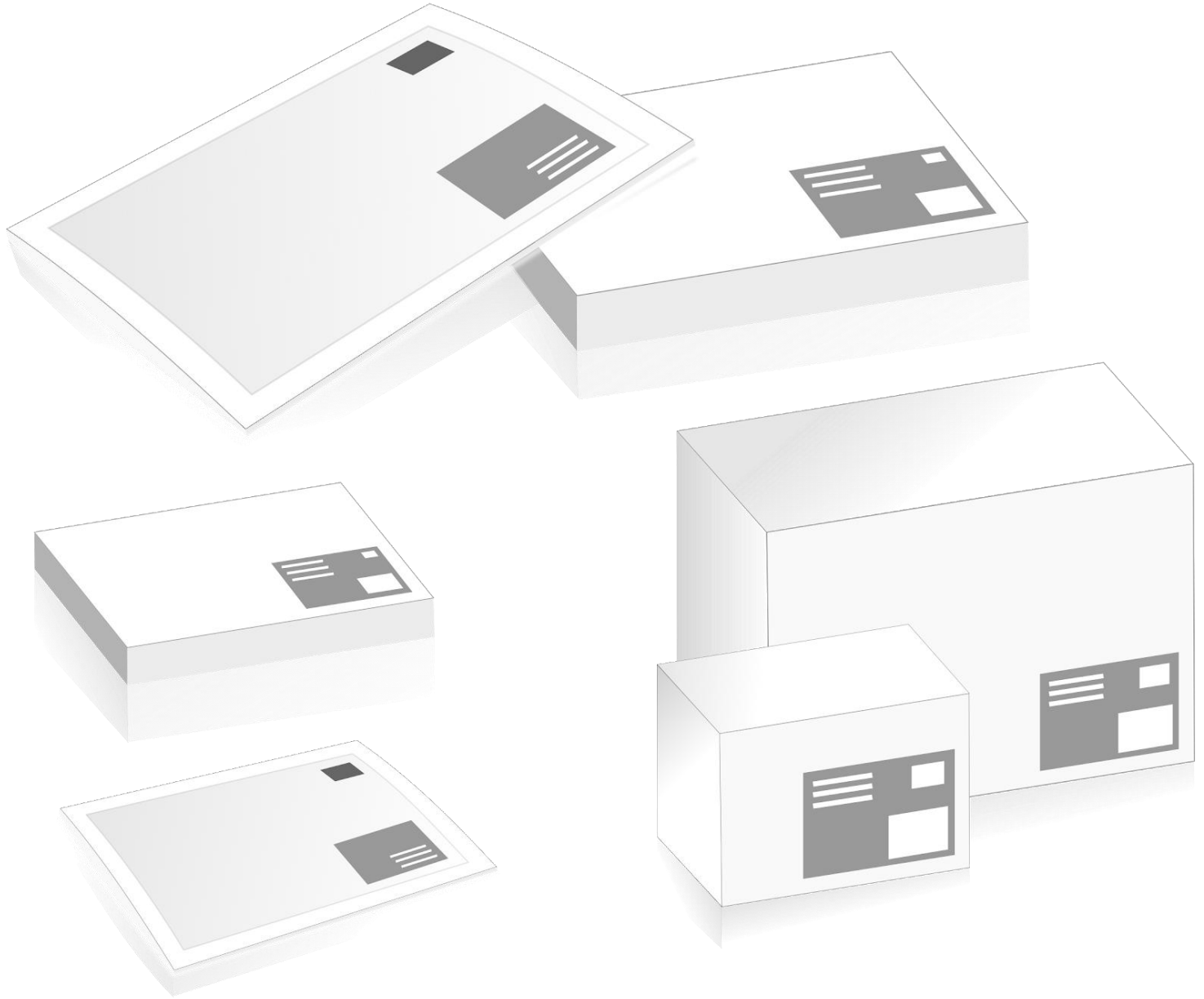


Librerías y paquetes



Información sobre LPI

Tema Cubierto

104.3: Administración de librerías compartidas

Peso

1 (El peso indica la cantidad de preguntas relativas al tema en cuestión, es decir cuánto más grande el número, es más su indicencia en el resultado del examen. Su valor puede ir de 1 a 6) [Descripción](#)

Los alumnos deberán ser capaces de determinar las librerías compartidas de las que dependen los programas ejecutables e instalarlas cuando sea necesario. [Áreas](#)

Claves de Conocimiento

- Identificar las librerías compartidas
 - Identificar las ubicaciones típicas de las librerías compartidas
 - Cargar librerías compartidas
- ### Términos y herramientas

- ldd
- ldconfig
- /etc/ld.so.conf
- LD_LIBRARY_PATH

ADMINISTRANDO BIBLIOTECAS COMPARTIDAS

¿Qué Es Una Librería?

Una librería es un archivo que contiene uno o más programas llamados funciones.

Cuando un programa es compilado (pasado de código fuente a código máquina) bajo linux, muchas de las funciones que se necesitan provienen de otras funciones ya existentes como las de manejo de disco, memoria, entre otras, por esta razón no se van a compilar nuevamente, puesto que ya están instaladas. Debido a esto, los programas van a compartir librerías.

Para evitar problemas relacionados con la memoria, estos programas apuntan a las librerías que necesitan, estando así dinámicamente enlazados.

Si un programa detecta que no puede acceder a estas librerías, lo más probable es que falle en su ejecución, teniendo que cumplir primero con este requerimiento para poder seguir su funcionamiento.

Administrando Librerías Compartidas

ldd

Muestra las librerías compartidas que requiere el programa.

Devuelve el nombre de la librería y dónde se supone que debe estar.

```
[root@restauracion gens-2.15.5]# ldd /usr/bin/nmap
linux-gate.so.1 => (0x00a96000)
libpcrc.so.0 => /lib/libpcrc.so.0
(0x0045f000)
libpcap.so.0.9.4 => /usr/lib/libpcap.so.0.9.4 (0x00252000)
libssl.so.6 => /lib/libssl.so.6 (0x00966000)
libcrypto.so.6 => /lib/libcrypto.so.6 (0x0074f000)
libdl.so.2 => /lib/libdl.so.2 (0x00458000) libstdc++.so.6
=> /usr/lib/libstdc++.so.6 (0x00e96000) libm.so.6 =>
/lib/libm.so.6 (0x00b35000)
...
```

Librerías del Sistema

Ubicaciones

Las librerías suelen encontrarse en los siguientes directorios:

- /lib

- /lib64 (librerías para arquitectura amd64 en distribuciones estilo Red Hat)
- /usr/lib
- /usr/lib64 (librerías para arquitectura amd64 en distribuciones estilo Red Hat)
- /usr/lib/x86_64-linux (librerías para arquitectura amd64 para distribuciones basadas en Debian)
- /usr/lib/i386-linux (librerías para arquitectura 32 bit para distribuciones basadas en Debian)

El enlazador dinámico es la herramienta que busca las librerías de acuerdo a las opciones que se usaron para compilar un programa. Uno de los primeros lugares que buscará está determinado por la variable **LD_LIBRARY_PATH**. Generalmente no suele asignarse algún valor a esa variable, ya que es preferible que utilice las librerías predeterminadas. Solamente cuando se está desarrollando una aplicación tendría sentido declararla, por ejemplo:

```
# export LD_LIBRARY_PATH=/home/sergio/mylibs
```

ldconfig

Este comando configura los enlaces en tiempo de ejecución de los distintos programas. Dicha tarea es realizada por el enlazador dinámico.

Cuando se instala una librería en el sistema es necesario actualizar esos enlaces para que los programas dependientes puedan funcionar bien. Eso se hace sencillamente ejecutando **ldconfig**.

```
# ldconfig
```

En el siguiente ejemplo el comando muestra las librerías de la caché.

```
# ldconfig -p
1054 libs found in cache `/etc/ld.so.cache'
lib3dkit.so.1 (libc6) => /usr/lib/lib3dkit.so.1
libzvbi.so.0 (libc6) => /usr/lib/libzvbi.so.0
libzvbi-chains.so.0 (libc6) => /usr/lib/libzvbi-chains.so.0
libz.so.1 (libc6) => /usr/lib/libz.so.1          libz.so
(libc6) => /usr/lib/libz.so          libx264.so.60 (libc6)
=> /usr/lib/libx264.so.60          libx11globalcomm.so.1
(libc6) => /usr/lib/libx11globalcomm.so.1
.....
```

Buscar por una en especial:

```
[root@restauracion ld.so.conf.d]# ldconfig -p |grep
ncurses libncursesw.so.5 (libc6) =>
/usr/lib/libncursesw.so.5 libncursesw.so (libc6) =>
/usr/lib/libncursesw.so libncurses.so.5 (libc6) =>
/usr/lib/libncurses.so.5 libncurses.so (libc6) =>
/usr/lib/libncurses.so
```

Librerías Locales

Reconstruir la caché:

```
# ldconfig
```

Luego que /usr/local/lib fue agregado, ld.so.conf debería verse así:

```
/usr/lib
/usr/i486-linux-libc5/lib
/usr/X11R6/lib
/usr/local/lib
```

Es recomendable ejecutar ldconfig luego de cualquier cambio en las librerías, para asegurarnos que fueron cargadas. Dicho archivo que guarda todo en memoria es :

```
[root@villadalmine zod]# ls -l /etc/ld.so.cache
```

-rw-r--r-- 1 root root 31332 Oct 30 08:44 /etc/ld.so.cache

Variable LD_LIBRARY_PATH

Esta variable se utiliza para definir temporalmente una ubicación diferente de las librerías. Esto se utiliza únicamente en ambientes de desarrollo.

```
# LD_LIBRARY_PATH=/usr/local/desarrollo/lib
# export LD_LIBRARY_PATH
```

BIBLIOGRAFÍA

Libros

LPI Linux Certification in a Nutshell, Third Edition, June 2010 —> ⚠️ Capítulo 5 Objetivo 102.3
LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009 —> ⚠️ Capítulo 2 Objetivo 102.3

INSTALANDO Y ADMINISTRANDO PAQUETES DEB

Peso	3
Tópico Cubierto	102.4 Usar el administrador de paquetes Debian
Descripción	Los alumnos deberán ser capaces de realizar la administración de paquetes, usando las herramientas de paquetes de Debian
Temas	<ul style="list-style-type: none">• Encontrar paquetes que contengan archivos específicos o bibliotecas, las cuales pueden estar o no instaladas.• Obtener información de un paquete como la versión, contenido, dependencias, integridad del paquete y estado de instalación (esté o no esté
	instalado el paquete).
Ejemplos	<ul style="list-style-type: none">• /etc/apt/sources.list• dpkg• dpkg-reconfigure• apt-get

- apt

Peso: Indica el valor de importancia que tiene este tópico en la certificación.

Tópico Cubierto: Indica, según el programa de certificación LPI, qué tópico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se deben tener en cuenta.

Distribuciones Basadas en Debian

El sistema de gestión de paquetes APT (Advanced Packaging Tool), fue creado por el proyecto Debian y se utiliza para la instalación y eliminación de programas en sistemas GNU/Linux. APT fue rápidamente utilizado para funcionar con paquetes **.deb**. Entre las distribuciones más conocidas que utilizan esta herramienta, se destacan Debian, Ubuntu y Mint.

Nomenclatura de un Paquete

Los nombres de los paquetes siguen la siguiente nomenclatura:

`<nombre>_<Nro.Version>-<NúmeroDeRevisión>_<arquitectura>.deb`

<nombre> Es el nombre del paquete

<Nro.Version> Es el número de versión establecido por el desarrollador, no sigue un estándar.

<NúmeroDeRevisión> Es el número de revisión que normalmente implica cambios en el archivo de control, los scripts de instalación o desinstalación, o en los archivos de configuración usados en el paquete.

<arquitectura> Indica para qué arquitectura es el paquete.

Ejemplo: El siguiente paquete “mysql-server_5.1.49-3_all.deb” indica que:

nombre: mysql-server versión: 5.1.49 número de revisión: 3 arquitectura:
all

Características de los Paquetes

Los paquetes en sistemas Debian/Ubuntu tienen la extensión “.deb”, y son manejados por un programa llamado “**dpkg**”. Este es una herramienta para instalar, eliminar y manipular de paquetes de Debian/Ubuntu. La herramienta principal de Debian para gestionar paquetes es apt-get.

Los paquetes “deb” contienen tres archivos:

debian-binary - Contiene el número de versión del paquete .deb. **control.tar.gz** - Contiene la información de control del paquete en una serie de ficheros de texto, por ejemplo, dependencias del paquete, prioridad, mantenedor, arquitectura, conflictos, versión, md5sum,...

data.tar - Contiene todos los archivos que se instalarán, con sus rutas de destino.

Instalando Paquetes

Dpkg

Mantiene información sobre los paquetes en el directorio `/var/lib/dpkg`.

Hay dos archivos que son los más relevantes:

available contiene la lista de todos los paquetes disponibles e información sobre los mismos. **status** contiene los atributos de los paquetes como el Status (instalado o marcado para eliminar). Estos archivos también son modificados por *apt-get* y *dselect*

Sintaxis

`dpkg [opciones] acción paquete`

Opciones frecuentes

`--configure` Configura un paquete que está desempaquetado pero no configurado.

`-i` (*-install*) Instala el paquete. Si se especifica la opción `--recursive` o `-R`, el fichero-paquete debe ser un directorio.

`-l` (*-list*) Lista todos los paquetes. Si se le agrega `-l [patron]` lista el o los paquetes que coinciden con el patrón.

`-L paquete` Lista los archivos instalados por el paquete.

`-s paquete` Obtiene información del paquete, como el estado, version, dependencias, etc. **Ejemplos**

Instalar el programa bzip2

```
# dpkg -i bzip2_1.0.5-6_i386.deb
```

Desinstala el programa quota

```
# dpkg -r quota
```

Para desinstalar un paquete y borrar incluso sus archivos de configuración:

```
# dpkg -P exim
```

Lista información del paquete

```
# dpkg -s bzip2
```

```
Package: bzip2
Status: purge ok installed
Priority: optional
Section: utils
Installed-Size: 156
Maintainer: Anibal Monsalve Salazar <anibal@debian.org>
Architecture: i386
Version: 1.0.5-6
Replaces: libbz2 (<< 0.9.5d-3)
Depends: libbz2-1.0 (= 1.0.5-6), libc6 (>= 2.3)
Suggests: bzip2-doc
Description: high-quality block-sorting file compressor - utilities
 bzip2 is a freely available, patent free, high-quality data compressor.
```

Ver información de un paquete no instalado

```
# dpkg -I mc_3%3a4.8.18-1_amd64.deb
```

```
.....
```

Repositorios

sources.list

El archivo `/etc/apt/sources.list` indica desde dónde deberán ser descargados los paquetes. Este archivo contiene una lista con repositorios.

La lista de repositorios está diseñada para gestionar cualquier número de fuentes y distintas procedencias de paquetes.

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

```
deb http://security.debian.org/ stretch/updates main
deb-src http://security.debian.org/ stretch/updates main
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
```

Descripción deb: repositorio con los paquetes

compilados deb-src: repositorio con el código fuente de los paquetes

En la línea se hace mención a los parámetros **stretch**, **main**, **contrib** y **non-free**.

Stretch es una de las ramas de la distribución, solo buscará paquetes y actualizaciones para dicha versión. Si quisiera tener la posibilidad de poder hacer un upgrade a futuras versiones, hay que cambiar dicho parámetro a **stable** o al nombre que corresponde al de la nueva versión.

Ramas de la distribución

Es importante recordar que Debian tiene 3 ramas en el desarrollo de la distribución:

Rama	Nombre en código	Características
stable	Stretch	Indicada para usar en producción
testing	Buster	Sirve para probar paquetes más nuevos pero aun no aceptados en 'stable'
unstable	Sid	Sirve para tener los paquetes más actualizados, pero tiene la desventaja de no estar testeados suficientemente

🚩 **Nota**: Los nombres en código de las versiones son correspondientes a octubre de 2018.

Se puede verificar el nombre de la última versión estable en la página <https://www.debian.org/releases/>.

Los paquetes se dividen en tres secciones, **main** (libres y predeterminado), **contrib** (programas libres, pero con dependencias no libres) y **non-free** (paquetes no libres)

Apt-get

apt-get no trabaja directamente con los paquetes .deb como lo hace **dpkg**, sino que utiliza los nombres de los paquetes, por ejemplo en

dpkg utilizo "**dpkg -i bzip2_1.0.5-6_i386.deb**", mientras que con apt-get utilizo "**apt-get install bzip2**" apt-get tiene una base de datos con

información que le permite a la herramienta actualizar automáticamente paquetes y sus dependencias, como también instalar nuevos paquetes disponibles.

Sintaxis

apt-get [opciones] [comando] [nombre_paquete ...]

Opciones frecuentes

- d Descarga los archivos, pero no los instala.
- s No realiza ninguna acción, simula lo que hubiese ocurrido pero sin hacer cambios en el sistema.
- y Responde que si(yes), a todas las preguntas que nos realiza la herramienta.

Comandos frecuentes

install Instala o actualiza uno o más paquetes **remove** Remueve los paquetes seleccionados **update** Sincroniza el listado de paquetes disponibles en los repositorios (configurados en el archivo *sources.list*) **upgrade** Realiza una actualización de todos los paquetes.

dist-upgrade Realiza la misma acción que **upgrade**, pero además, puede llegar a quitar paquetes si las nuevas condiciones de los repositorios lo requieren (por ejemplo, hay paquetes que no existían antes, o paquetes que ya no existen más). Se puede usar para actualizar de una versión estable a una más nueva del mismo tipo. Para hay que modificar el archivo **sources.list** como se indicó anteriormente.

clean Borra los paquetes de instalación descargados (/var/cache/apt/archives)

Ejemplos

Actualizar base de datos contra los repositorios

```
# apt-get update
```

Instalar el programa **mc**, sin confirmación

```
# apt-get -y install mc
```

Leyendo lista de paquetes... Hecho Creando árbol de dependencias Leyendo la información de estado... Hecho El paquete indicado a continuación se instaló de forma automática y ya no es necesarios.

libpq5

Utilice «apt-get autoremove» para eliminarlos. Paquetes sugeridos:

zip unzip arj xpdf-reader pdf-viewer dbview odt2txt gv catdvi djvulibre-bin imagemagick python-boto python-tz

Se instalarán los siguientes paquetes NUEVOS: mc

0 actualizados, 1 se instalarán, 0 para eliminar y 28 no actualizados. Necesito descargar 2,173 kB de archivos. Se utilizarán 6,603 kB de espacio de disco adicional después de esta operación.

Eliminar el programa **mc**

```
# apt-get remove mc
```

Eliminar el programa **mc** y toda su configuración

```
# apt-get --purge remove mc
```

Simular una operación

En este caso se utiliza la opción **--dry-run**:

```
# apt-get --dry-run upgrade
```

Apt-cache

Esta herramienta se utiliza para consultar la caché local de la base de datos paquetes de Debian.

Comandos frecuentes

showpkg Muestra información acerca del paquete y sus dependencias entre otras cosas. **show**

Muestra descripción acerca del paquete y paquetes sugeridos.

search Busca un paquete por su nombre o descripción.

Ejemplos

Buscar paquete que contenga en su nombre o descripción la palabra midnight

```
# apt-cache search commander gnome-commander - nice
and fast file manager for the GNOME desktop mc -
Midnight Commander - a powerful file manager
```

Además, podemos buscar para que coincida solamente con el nombre el paquete:


```
# apt-cache search --names-only music
```

Apt

Además, existe la herramienta **apt** que es una herramienta más sencilla equivalente a apt-get y apt-search combinadas.

Ejemplos

```
# apt search mc
# apt install mc
```

BIBLIOGRAFIA

Libros

LPI Linux Certification in a Nutshell, Third Edition, June 2010 → 🚨 Capítulo 1 Objetivo 102.4

LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009 → 🚨 Capítulo 3 Objetivo 102.4

INSTALANDO Y ADMINISTRANDO PAQUETES RPM

Peso	3
Descripción	Los alumnos deberían ser capaces de realizar la gestión de paquetes usando las herramientas RPM y YUM
Tópico Cubierto	Usar los gestores de paquetes RPM y YUM
Temas	<ul style="list-style-type: none">• Instalar, reinstalar, actualizar y eliminar paquetes usando RPM, YUM y Zypper• Obtener la información sobre paquetes RPM tales como la versión, status, dependencias, integridad y firmas.• Determinar qué archivos proporciona un paquete, así como también encontrar a cuál paquete corresponde un archivo específico• Saber qué es dnf
Ejemplos	<ul style="list-style-type: none">• rpm• rpm2cpio• /etc/yum.conf• /etc/yum.repos.d/• yum• yumdownloader• zypper

Peso: Indica el valor de importancia que tiene este tópico en la certificación.

Tópico Cubierto: Indica, según el programa de certificación LPI, qué tópico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se deben tener en cuenta.

Distribuciones Que Usan RPM

RPM es un acrónimo recursivo que significa Red Hat Package Manager. Este sistema de paquetes, que viene desde Red Hat, es utilizado por varias distribuciones (Fedora, CentOS, SUSE, etc) Con estos paquetes vamos a poder instalar nuestros programas de manera fácil.

Nomenclatura de un Paquete

Un paquete RPM consta de cuatro componentes, al igual que Debian:

Un nombre corto y descriptivo del paquete, que puede estar separado por -, _ , etc.

firefox-3.6.9-2.el5

Nombre: Nombre del paquete (ejemplo firefox)

Versión: Cada paquete contiene una versión. La forma que se usa es (major.minor.patchlevel).

Revisión: Se trata de la versión release del paquete, dado que, cuando tienen mismo número de versión, se determina cuál es la más nueva por este campo; ya que, algunas veces, los cambios hechos en los paquetes son menores.

Architecture: Muestra cuál es la arquitectura que determina el paquete (i386, i586, i686, ia64, sparc, ppc, noarch)

Ejemplo:

Installed Packages

Name: firefox

Arch: i386

Version: 3.6.9

Release: 2.el5

Gestionando Paquetes RPM

Repositorios

Archivos de configuración:

/etc/yum.conf → a través de este archivo se configura yum

Configuración /etc/yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=0 debuglevel=2
logfile=/var/log/yum.log
distroverpkg=redhat-release
tolerant=1 exactarch=1 obsoletes=1
gpgcheck=1 plugins=1 # Note: yum-
```

```
RHN-plugin doesn't honor this.
metadata_expire=1h

# Default.
# installonly_limit = 3

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

/etc/yum.repos.d → Aquí podemos poner nuestros propios repositorios.

```
[root@restauracion /etc/yum.repos.d]# ls
adobe-linux-i386.repo  rhel-debuginfo.repo  convirt.repo
rpmforge.repo         dsmerror.log         rpmforge-testing.repo
dsmwebcl.log          rpmfusion-free-updates.repo
epel.repo             rpmfusion-free-updates-testing.repo
epel-testing.repo     rpmfusion-nonfree-updates.repo
google.repo           rpmfusion-nonfree-updates-testing.repo
mirrors-rpmforge     yum.conf              openfusion.repo.bkp
```

El Comando rpm

Este comando nos permite instalar, remover, actualizar y verificar paquetes entre otras cosas.

Sintaxis: **rpm**
opciones

Instalar y actualizar paquetes

Opciones:

-i → instala el paquete propiamente dicho.

-e → desinstala el paquete

-U → hace un `//update//` del paquete a su versión más nueva. (si el paquete no existe en el sistema, se comporta como un `-i`)

-F → o `freshen` con este actualiza solo aquellos paquetes que están instalados en el sistema, pero en este caso, no va actualizar las dependencias de ese paquete, aunque esté mencionado en la línea de comandos.

--force fuerza la actualización de los paquetes ya sea viejo por nuevo o viceversa (también se puede usar con `--replacepkgs`, `--replacefiles`, `--oldpackage`)

-h (también `--hash`) marca un string de progreso de la tarea a realizar.

--nodeps no busca las dependencias del paquete; esto puede generar problemas en la base de dependencias.

--test hace una simulación de la instalación o de la tarea que fuera; no puede ser utilizado con `-h` pero sí con `-vv -v` muestra más información

-vv muestra información más ampliada que solamente con `-v`

Preguntar por paquetes usando -q

-a Nos devuelve todos los paquetes que tenemos instalados, podemos filtrarlos con **grep** para buscar alguno en particular

```
[root@restauracion ~]# rpm -qa |grep grub
grub-0.97-13.5
```

Preguntar por palabra

Con la opción **-f** vamos a poder preguntar qué paquete está usando el fichero que demos como argumento.*

```
[root@restauracion ~]# rpm -qf /etc/passwd setup-2.5.58-7.el5
```

Preguntar información

Vamos a preguntar por un paquete con la opción **-p** pero si ya tenemos el paquete ^{instalado,} omitimos esa opción. Le vamos a agregar **-i** para que nos de más información.

```
[root@restauracion ~]# rpm -qi grub-0.97-13.5
Name           : grub                      Relocations: (not relocatable)
Version        : 0.97                     Vendor: Red Hat, Inc.
Release        : 13.5                     Build Date: Tue 30 Jun 2009 02:34:35 PM ART
Install Date: Thu 29 Apr 2010 04:37:57 PM ART Build Host: hs20-bc1-2.build.redhat.com
Group          : System Environment/Base    Source RPM: grub-0.97-13.5.src.rpm
Size           : 1057883                   License: GPL
Signature      : DSA/SHA1, Wed 15 Jul 2009 05:53:04 AM ART, Key ID 5326810137017186
Packager       : Red Hat, Inc.
URL            : http://www.gnu.org/software/grub/
Summary        : GRUB - the Grand Unified Boot Loader.
Description    : GRUB (Grand Unified Boot Loader) is an
experimental boot loader capable of booting into most
free operating systems - Linux, FreeBSD, NetBSD, GNU
Mach, and others as well as most commercial operating
systems.
```

Preguntar por paquete y su información

Con las opciones **-qpi**

```
[root@restauracion opt]# rpm -qpi shutter-0.85.1-1.fc10.noarch.rpm
warning: shutter-0.85.1-1.fc10.noarch.rpm: Header V3 DSA signature: NOKEY, key
ID 4ebfc273 Name           : shutter                      Relocations: (not
relocatable)
Version          : 0.85.1                               Vendor: Fedora Project
Release         : 1.fc10                                 Build Date: Mon 07 Dec 2009 10:23:15 AM ART
Install Date: (not installed) Build Host: x86-6.fedora.phx.redhat.com
Group           : Applications/Multimedia                Source RPM: shutter-0.85.1-1.fc10.src.rpm
Size            : 9531857                                 License: GPL+ or Artistic
Signature       : DSA/SHA1, Mon 07 Dec 2009 12:54:07 PM ART, Key ID bf226fcc4ebfc273
Packager        : Fedora Project
URL             : http://shutter-project.org
Summary         : GTK+ 2.0 screenshot application written in perl
Description    :
Shutter is a GTK+ 2.0 screenshot application
written in perl. Shutter covers all features of
common command line tools like scrot or import and
adds reasonable new features combined with a
comfortable GUI using the GTK+ 2.0 framework
```

Preguntar por archivos de configuración

Algunos usos más frecuentes de opciones para consultar.

-c Muestra solo los archivos de configuración:

```
[root@restauracion opt]# rpm -qc at
/etc/at.deny
/etc/pam.d/atd
/etc/rc.d/init.d/atd
```

Preguntar por documentación

-d Muestra solo los archivos de documentación del paquete:

```
[root@restauracion opt]# rpm -qd at
/usr/share/doc/at-3.1.8/ChangeLog
/usr/share/doc/at-3.1.8/Copyright
/usr/share/doc/at-3.1.8/Problems
/usr/share/doc/at-3.1.8/README
/usr/share/doc/at-3.1.8/timespec
/usr/share/man/man1/at.1.gz
/usr/share/man/man1/atq.1.gz
/usr/share/man/man1/atrm.1.gz
....
```

Preguntar archivos instalados

-l Muestra todos los archivos involucrados en un paquete

```
[root@restauracion opt]# rpm -ql at
/etc/at.deny
/etc/pam.d/atd
/etc/rc.d/init.d/atd
/etc/sysconfig/atd
/usr/bin/at
/usr/bin/atq
/usr/bin/atrm
.....
```

Preguntar por dependencias

-R Muestra los paquetes que éste necesita (dependencias):

```
[root@restauracion opt]# rpm -qR at
/bin/bash
/bin/sh
/etc/init.d
chkconfig config(at)
= 3.1.8-84.el5
fileutils libc.so.6
libc.so.6(GLIBC_2.0)
.....
```

Preguntar por archivos de un paquete

Para preguntar por los archivos de un paquete;

```
[root@restauracion opt]# rpm -qlp shutter-0.85.1-1.fc10.noarch.rpm
/usr/bin/shutter
/usr/share/applications/shutter.desktop
/usr/share/doc/shutter-0.85.1
/usr/share/doc/shutter-0.85.1/COPYING
/usr/share/icons/hicolor/16x16/apps/shutter.png
.....
```

Verificación de paquetes

Una opción muy importante es la de verificar los paquetes para poder saber qué fue cambiado Se realiza con la opción **-V** (o **--verify**)

Opciones que se pueden usar con **-V**

--nofiles ignora archivos perdidos

--nomd5 ignora los chequeos de error de md5

--nopgp ignora los chequeos de error de pgp

```
[root@restauracion opt]# rpm --verify * package
expect-5.43.0-9.fc8.i386.rpm is not installed
package expect-5.44.1.15-2.el6.i686.rpm is not
installed package graphviz-2.20.3-
5.el5_89_1.1.i686.rpm is not installed
```

Reparación base de datos de paquetes

Si apareciera un error con la base de datos se puede regenerar

```
rpm --rebuilddb [options] rpm --initdb [options]
```

yum

El gestor de paquetes YUM (YellowDog Updater Modified) nos ofrece una manera rápida de instalar paquetes. Podemos actualizar, instalar y remover paquetes. Tiene funciones muy similares a la de rpm, pero con la particularidad que puede administrar toda la resolución e instalación de dependencias de paquetes.

Además, YUM, nos permite cargar múltiples repositorios de paquetes de manera muy sencilla.

Primeros Pasos

Opciones más usadas con yum **install** →

instala un paquete o paquetes **erase** →

remueve un paquete o paquetes **update** →

actualiza un paquete o paquetes **list** →

lista paquetes o un grupo de paquetes

search → busca paquetes **check-update**

→ lista los paquetes a actualizar

Instalando un paquete

```
# yum install postgresql.x86_64
Resolving Dependencies
Install      2 Package(s)
Is this ok [y/N]: y

Package(s) data still to download: 3.0 M
(1/2): postgresql-9.0.4-5.fc15.x86_64.rpm      | 2.8 MB      00:11
(2/2): postgresql-libs-9.0.4-5.fc15.x86_64.rpm  | 203 kB      00:00
-----
Total                                           241 kB/s | 3.0 MB      00:12

Running Transaction
  Installing : postgresql-libs-9.0.4-5.fc15.x86_64
1/2  Installing : postgresql-9.0.4-5.fc15.x86_64
2/2

Complete!
```

De manera predeterminada, **yum** pedirá confirmación de la instalación, se puede evitar con el parámetro **-y**

```
# yum -y install paquete
```

Desinstalar un paquete con yum

Para desinstalar un paquete incluyendo todas sus dependencias

```
# yum remove postgresql.x86_64
Resolving Dependencies ---> Package
postgresql.x86_64 0:9.0.4-5.fc15 will be erased

Is this ok [y/N]: y

Running Transaction
  Erasing      : postgresql-9.0.4-5.fc15.x86_64      1/1
  Removed:
postgresql.x86_64 0:9.0.4-5.fc15
Complete!
```

Actualizar un paquete ya instalado

Si tenemos una versión vieja de un paquete, podemos utilizar “yum update paquete” para actualizarlo a la última versión. Esto también busca y identifica todas las dependencias requeridas.

```
# yum update postgresql.x86_64
```

Buscar un paquete para instalarlo con yum

Si no sabemos exactamente el nombre del paquete a instalar, podemos usar el comando “yum search palabra”, que realizará una búsqueda de todos los paquetes que coincidan con la palabra indicada.

En el siguiente ejemplo, la búsqueda que realizaremos devolverá como resultado todos los paquetes disponibles que coincidan con la palabra “firefox”.

```
# yum search firefox
Loaded plugins: langpacks, presto, refresh-packagekit
===== N/S Matched: firefox =====
firefox.x86_64 : Mozilla Firefox Web browser gnome-do-plugins-
firefox.x86_64 : gnome-do-plugins for firefox mozilla-firetray-
firefox.x86_64 : System tray extension for firefox mozilla-
adblockplus.noarch : Adblocking extension for Mozilla Firefox
mozilla-noscript.noarch : JavaScript white list extension for
Mozilla Firefox
```

Solo busca en los campos nombre y resumen; si quisiéramos que busque por todos los campos tendríamos que usar la opción “search all”

Mostrar información adicional acerca de un paquete

Una vez que busquemos y poseemos el nombre del paquete con “yum search”, podremos obtener información adicional acerca del paquete utilizando el siguiente comando: “yum info paquete”

```
# yum info samba-common.i686
Loaded plugins: langpacks, presto, refresh-packagekit
Available Packages
Name           : samba-common
Arch           : i686
Epoch         : 1
Version        : 3.5.11
Release        : 71.fc15.1
Size           : 9.9 M
Repo           : updates
Summary        : Files used by both Samba servers and clients
URL            : http://www.samba.org/
License        : GPLv3+ and LGPLv3+
Description    : Samba-common provides files necessary for both the server and client packages of Samba.
```

Ver todos los paquetes disponibles

El siguiente comando nos mostrará una lista de todos los paquetes disponibles que hay en nuestra base de datos yum.

```
# yum list | less
```

Listar paquetes instalados

Muestra una lista de todos los paquetes instalados de nuestro sistema.

```
# yum list installed
```


Visualizar a qué paquete le pertenece un archivo

Utilizar el comando “**yum provides**” para identificar a qué paquete le corresponde determinado archivo.

```
# yum provides /etc/sysconfig/nfs
Loaded plugins: langpacks, presto, refresh-packagekit
1:nfs-utils-1.2.3-10.fc15.x86_64 : NFS utilities and supporting clients and
                                : daemons for the kernel NFS server

Repo           : fedora
Matched from:
Filename       : /etc/sysconfig/nfs
.....
```

Listar los grupos de programas instalables

En **yum**, una gran cantidad de paquetes están agrupados. En lugar de tener que buscar paquetes individualmente, con esta opción se instalarán los que correspondan a un grupo específico.

Para poder ver los grupo de paquetes disponibles, usamos el comando “**yum grouplist**”. La salida de este comando mostrará los grupos disponibles y los instalados.

```
# yum grouplist

Installed Groups:
  Administration Tools
  Base
  Design Suite
  ....

Installed Language Groups:
  Arabic Support [ar]
  Armenian Support [hy]
  Bengali Support [bn]
  ....

Available Groups:
  Authoring and Publishing
  Books and Guides
  Clustering
  DNS Name Server
  Development Libraries
  Development Tools
  .....
```

Instalar un grupo de programas específicos

Para instalar un grupo específico de programas, utilizamos la opción “**groupinstall**”. Se instalará el grupo “DNS Name Server”, el cual trae por dependencia el paquete bind-chroot.

```
# yum groupinstall 'DNS Name Server'
```

```
Dependencies Resolved
Install      2 Package(s)
Is this ok [y/N]: y
```

```
Package(s) data still to download: 3.6 M
(1/2): bind-9.8.0-9.P4.fc15.x86_64.rpm      | 3.6 MB      00:15
(2/2): bind-chroot-9.8.0-9.P4.fc15.x86_64.rpm | 69 kB      00:00 -----
-----
Total                235 kB/s | 3.6 MB      00:15
  Installed:  bind-chroot.x86_64
32:9.8.0-9.P4.fc15  Dependency
Installed:
  bind.x86_64 32:9.8.0-
9.P4.fc15
Complete!
```

Actualizar un grupo instalado

Si ya disponemos de un grupo de programas instalado, podemos realizar una actualización de este con el comando “**yum groupupdate**”.

```
# yum groupupdate 'Graphical Internet'
```

```
Dependencies Resolved
Upgrade      5 Package(s)
Is this ok [y/N]: y
```

```
Running Transaction
  Updating   : evolution-data-server-3.0.2-1.fc15.x86_64      1/10
  Updating   : evolution-3.0.2-3.fc15.x86_64                 2/10
  Updating   : evolution-NetworkManager-3.0.2-3.fc15.x86_64  3/10
  Updating   : evolution-help-3.0.2-3.fc15.noarch             4/10
  Updating   : empathy-3.0.2-3.fc15.x86_64                   5/10
  Cleanup    : evolution-NetworkManager-3.0.1-1.fc15.x86_64  6/10
  Cleanup    : evolution-help-3.0.1-1.fc15.noarch             7/10
  Cleanup    : evolution-3.0.1-1.fc15.x86_64                 8/10
  Cleanup    : empathy-3.0.1-3.fc15.x86_64
9/10 Cleanup : evolution-data-server-3.0.1-
1.fc15.x86_64      10/10
```

```
Complete!
```

Desinstalar un grupo de programas

Para borrar un grupo de programas instalados, se utiliza el comando “**yum groupremove**”.

```
# yum groupremove 'DNS Name Server'
```

```
Dependencies Resolved
Remove      2 Package(s)
Is this ok [y/N]: y
```

```
Running Transaction
```

```
Erasing      : 32:bind-chroot-9.8.0-9.P4.fc15.x86_64
1/2 Erasing   : 32:bind-9.8.0-9.P4.fc15.x86_64
2/2
```

Complete!

Mostrar los repositorios instalados

Para poder ver los repositorios que tenemos configurados en nuestro sistema se usa el comando “**yum repolist**”. El siguiente comando muestra solamente los repositorios activos.

```
# yum repolist
repo id      repo name
status fedora      Fedora 15 - x86_64
24,085 updates  Fedora 15 - x86_64 - Updates
5,612
```

Para mostrar todos los repositorios (habilitados y deshabilitados)

```
# yum repolist all
repo id      repo name
status fedora      Fedora 15 - x86_64
enabled: 24,085 fedora-debuginfo      Fedora 15 - x86_64 - Debug
disabled fedora-source      Fedora 15 - Source
disabled rawhide-debuginfo    Fedora - Rawhide - Debug
disabled rawhide-source      Fedora - Rawhide - Source
disabled updates              Fedora 15 - x86_64 - Updates
enabled: 5,612 updates-debuginfo      Fedora 15 - x86_64 -
Updates - Debug      disabled updates-source      Fedora 15 -
Updates Source      disabled updates-testing
Fedora 15 - x86_64 - Test Updates      disabled updates-testing-
debuginfo Fedora 15 - x86_64 - Test Updates Debug      disabled updates-
testing-source      Fedora 15 - Test Updates Source      disabled
```

Limpiar la caché

Yum puede guardar en una caché:

- Paquetes descargados antes de instalarlos
- Encabezados
- Metadatos de paquetes
- Metadatos de la caché sqlite
- Datos de la base local de RPM

Por ejemplo para borrar toda la caché se realiza lo siguiente:

```
# yum clean all
```

Esta tipo de operación se suele realizar cuando alguno de los elementos de la caché se ha dañado.

DNF

Es un gestor de paquetes que utiliza las librerías `libsolv` y `libdnf`.

En versiones recientes de Fedora **dnf** ha reemplazado a yum como herramienta predeterminada para administrar paquetes. La sintaxis de `dnf` y `yum` son muy similares. Se puede encontrar más información en la página del manual `yum2dnf`.

rpm2cpio

Convierte un archivo **rpm** a un archivo **cpio**.

Aquí veremos el contenido del archivo rpm

```
[root@restauracion opt]# rpm2cpio nxclient-3.4.0-7.i386.rpm |cpio -t
./etc/profile.d/nx.csh
./etc/profile.d/nx.sh
./usr/NX
./usr/NX/bin
./usr/NX/bin/nxclient
.....
```

Ahora descargo lo que necesito:

```
[root@restauracion opt]# rpm2cpio nxclient-3.4.0-7.i386.rpm |cpio -ivd ./usr/NX/share/images/wizard.png
./usr/NX/share/images/wizard.png
g 19913 blocks
```

zypper

Zypper es el gestor de paquetes predeterminado de la distribución SUSE. Está basado en la librería libzypp. Algunas de sus características son:

- Archivos .repo compatibles con YUM
- Trata las dependencias entre los paquetes como un problema de satisfacibilidad booleana •

Usa metalink al igual que yum para mejorar las descargas

Instalar paquetes

```
# zypper install ranger
```

Reinstalar paquetes

```
# zypper install -f ranger
```

Actualizar todos los paquetes

```
# zypper update
```

Actualizar un paquete

```
# zypper update ranger
```

Obtener información de un paquete

```
# zypper info --provides --requires ranger
```

Archivo de configuración de zypper

El archivo de configuración principal de zypper es **/etc/zypp/zypp.conf**, uno de los parámetros más importantes es:

```
repo.refresh.delay = 10
```

En el ejemplo de arriba significa que cada 10 minutos zypper actualizará la información que posee de los repositorios.

Archivos de repositorios

Los archivos de repositorios están en el directorio `/etc/zypp/repos.d`, cada uno de ellos puede tener un parámetro **autorefresh**, si está puesto en **1**, zypper actualizará la información del repositorio de acuerdo a la configuración dada por `repo.refresh.delay` en el archivo principal. Por ejemplo:

```
[openSUSE-Leap-15.0-1] name=openSUSE-Leap-15.0-1
enabled=1
autorefresh=1
baseurl=http://download.opensuse.org/distribution/leap/15.0/repo/oss path=/ type=rpm-md keeppackages=0
```

BIBLIOGRAFÍA

Libros

- [CompTIA Linux+ / LPIC-1 Cert Guide: (Exams LX0-103 & LX0-104/101-400 & 102-400)](<https://www.lpimarketplace.com/CompTIA-Linux-LPIC-1-Cert-Guiden-p/978-0-78-975455-4.htm>)
- [LPIC-1 Certification All-in-One Exam Guide Exams LPIC-1 book | LPI Marketplace](<https://www.lpimarketplace.com/LPIC-1-Certification-All-in-One-Exam-Guide-p/978-0-071-84168-9.htm>)

Páginas

- [Preparando LPIC-1 102.5 Usar el administrador de paquetes RPM y YUM](#)
- [15 Linux Yum Command Examples – Install, Uninstall, Update Packages](#)
- [Armar Repositorio Local](#)
- [Reparar base RPM](#)

Páginas

- [Preparando LPIC-1 102.3 Administrar bibliotecas compartidas](#)
- [Introduccion a las Librerias Compartidas GeekStuff](#)
- [Compilando mi libreria en c GeekStuff](#)
- [\[access.redhat.com/articles/yum-cheat-sheet\]\(https://access.redhat.com/articles/yum-cheat-sheet\)](https://access.redhat.com/articles/yum-cheat-sheet)

MATERIAL ANEXO

aptitude

Es un gestor de paquetes para sistemas Debian y su utilización es muy similar al **apt**, la diferencia está en su interfaz en modo texto para el manejo del sistema de paquetes, y, además, utiliza un algoritmo distinto para manejar dependencias, por lo que debe usarse con precaución.

Estos son algunos de los archivos más importantes de aptitude:

/var/lib/aptitude/pkgstates: Se almacenan el conjunto de estados y algunas banderas de los paquetes

/etc/apt/apt.conf, */etc/apt/apt.conf.d/**, *~/.aptitude/config*: Son los archivos de configuración de aptitude.

/var/log/aptitude: Logs de aptitude

Tanto **apt** como **aptitude** comparten el mismo archivo **sources.list**

Sintaxis

aptitude [opciones] [comando] [paquetes]

Ejemplos

```
# aptitude search mc # (igual que apt-cache search)
# aptitude update
# aptitude install mc
```

Para acceder a la interfaz de texto hay que ejecutar **aptitude** sin ningún parámetro adicional.

Alien

Alien es un programa que convierte principalmente paquetes **rpm** a **deb** o paquetes **deb** a **rpm**. Si se desea utilizar un paquete de una distribución GNU/Linux diferente a la que está instalada en nuestro sistema, alien, puede convertir el paquete de acuerdo a la distribución que tengamos en nuestro sistema.

Sintaxis

alien [-to-deb] [-to-rpm] [options] paquete

Opciones frecuentes

- i Instala automáticamente el paquete generado y elimina el paquete convertido.
- r Convierte un paquete a RPM.
- t Convierte un paquete a un archivo tar gzip

Ejemplos

Convierte un paquete RPM a un paquete DEB # **alien -to-deb package.rpm**

Convierte un paquete DEB a un paquete RPM # **alien -to-rpm package.deb**

Convierte un paquete RPM a un paquete DEB e instala el paquete generado (Por default, se convierte a **.deb**, no es necesario especificarlo).
alien -i package.rpm

Convierte los tres paquetes *foo.deb bar.rpm baz.tgz* a los demas formatos # **alien -to-deb -to-rpm -to-tgz -to-slp foo.deb bar.rpm baz.tgz**

Es decir, que el paquete *foo.deb* es convertido a *afoo.rpm*, *foo.tgz* y *foo.slp*

dselect

Funciona como una interfaz de **dpkg**, la cual ofrece una pantalla completa de selección de paquetes con un solucionador de dependencias y conflictos. Permite instalar, actualizar y eliminar paquetes. Habitualmente, dselect se invoca sin parámetros. Se presenta un menú interactivo que ofrece al usuario una lista de acciones. Si se da una acción como argumento se ejecutará inmediatamente.

yumdownloader

Este agregado sirve para descargar paquetes para uso posterior, junto con sus dependencias.

Algunas veces, quizás necesitemos descargar paquetes con sus dependencias pero no para instalarlos, sino para luego copiarlos e instalarlos en otra computadora.

Yumdownloader (que viene con el paquete yum-utils) es muy útil para eso, porque nos sirve construir grupos de paquetes reutilizables para un grupo de computadoras idénticas y luego distribuirlos.

Descargar Paquete Fuente

```
# yumdownloader --source packagename
```

Descargar un paquete y dependencias

```
# yumdownloader --resolve packagename
```

Descargar un paquete y sus dependencias para una diferente instalación Ejemplo

de instalación con chroot

```
# yumdownloader --resolve packagename --root /path/to/chroot
```

Un truco muy interesante de yumdownloader con chroot, copiando el directorio /var/lib/rpm , puedes hacer uso de un perfil de rpm para descargar paquetes para la computadora de un amigo y pasarlo a un cd.

Operaciones Avanzadas Con yum

Instalar desde un repositorio deshabilitado

Yum solo instala desde los repositorios habilitados. Si por alguna razón deseamos instalar algún paquete desde un repositorio deshabilitado, deberemos usar el comando “**yum install**” con las opciones que se muestran a continuación:

```
# yum --enablerepo=fedora-source install vim-X11.x86_64
```

```
Dependencies Resolved
```

```
Install      1 Package(s)
```

```
Is this ok [y/N]: y
```

```
Running Transaction
```

```
Installing : 2:vim-X11-7.3.138-1.fc15.x86_64 1/1
```

```
Complete!
```

Ejecutar comandos interactivamente

Yum también posee un método interactivo para correr sus comandos:

```
# yum shell

Setting up Yum Shell
> info samba.x86_64
Available Packages
Name           : samba
Arch           : x86_64
Epoch         : 1
Version        : 3.5.11
Release        : 71.fc15.1
Size           : 4.6 M
Repo           : updates
Summary        : Server and Client software to interoperate with Windows machines
URL            : http://www.samba.org/
License        : GPLv3+ and LGPLv3+
Description    :
                : Samba is the suite of programs by which a lot of PC-related
                : machines share files, printers, and other
information (such as      : lists of available files and
printers). The Windows NT, OS/2, and
...
```

También podemos leer los comandos desde un archivo y ejecutarlos uno por uno. Esto es muy útil cuando tenemos múltiples sistemas, porque en lugar de ejecutar los mismos comandos en cada equipo, se ejecutará con “**yum shell**” como muestra a continuación.

```
# cat yum_cmd.txt
repolist
info nfs-utils-lib.x86_64

# yum shell yum_cmd.txt
repo id      repo name
status fedora      Fedora 15 - x86_64
24,085 updates      Fedora 15 - x86_64 -
Updates      5,612
Available Packages
Name         : nfs-utils-lib
Arch         : x86_64
Version      : 1.1.5
Release      : 5.fc15
Size         : 61 k
Repo         : fedora
Summary      : Network File System Support Library
URL          : http://www.citi.umich.edu/projects/nfsv4/linux/
License      : BSD
Description  : Support libraries that are needed by the commands and
                : daemons the nfs-utils rpm.

Leaving Shell
```

Armar Repositorio Local

Crear el directorio que va a contener el repositorio local


```
# mkdir -p /var/pub/os
# mkdir -p
/var/pub/updates
```

Copiamos el contenido de los discos de instalación

```
#mount -t iso9660 -o ro /dev/cdrom /cdrom (el directorio /cdrom se ha creado
previamente) #cp -Rf /cdrom/* /var/pub/os
```

Instalamos el siguiente paquete:

```
# yum -y install createrepo
```

Ahora solo hay que ejecutar 'createrepo' sobre el directorio, a fin de generar el repositorio para yum:

```
# createrepo /var/pub/os
```

Crear el siguiente archivo:

```
# vi /etc/yum.repos.d/fedora-local.repo
```

```
[local]
name=Fedora $releasever -
$basearch
baseurl=file:///var/mirepo/
enabled=1 gpgcheck=0 :wq
```

Luego limpiamos y actualizamos.

```
# yum clean all
# yum check-
update
```

Podemos comprobar qué repositorios está usando yum :

```
# yum repolist
```

Debe listar el repositorio añadido anteriormente.

Instalando Plugins

yum-plugin-fastestmirror

Este es el primer plugin de yum que se suele instalar.

Este plugin hace un chequeo del tiempo de todos los servidores de repositorios que tenemos configurados y escoge el que conteste más rápido.

```
# yum install yum-plugin-fastestmirror
```

El plugin, luego de chequear cuál es el servidor más rápido de todos, almacena metadatos en el directorio /var/cache/yum/timedhosts.txt; estos metadatos expiran luego de algún tiempo y por el contrario si los tiempos dejan de ser los más rápidos se puede forzar la búsqueda de nuevo borrando el archivo.

yum-presto

Si tenemos una conexión a Internet muy limitada, este paquete nos servirá. El plugin descarga “deltarpms”, que solo cuentan con los cambios de las versiones instaladas en nuestro equipo y en Internet, haciendo que los tiempos de descarga sean menores.

```
# yum install yum-presto
```

yum-security

Una de las alternativas es solo actualizar en nuestro sistema los paquetes de seguridad; esto sucede debido a que Fedora actualiza constantemente sus repositorios, por lo cual muchos paquetes quedan obsoletos rápidamente. Por esta razón, algunas personas son temerosas de actualizar sus sistemas con los últimos paquetes de actualización, principalmente porque algunas veces, esas actualizaciones, repentinamente “rompen” cosas.

Las actualizaciones de Fedora no son “backports” de correcciones, sino que más bien se tratan de versiones nuevas. Una manera de resolver esto es solo actualizando el sistema con actualizaciones de seguridad. Para esto, debemos usar el siguiente plugin

```
# yum install yum-security
```

Para utilizarlo

```
# yum --security update
```

```
# yum-skipbroken
```

Una vez que empezamos a instalar repositorios de terceros, romper dependencias es un riesgo que necesitamos tomar; esto usualmente ocurre cuando los paquetes del repositorio base y los repositorios de terceros no están sincronizados entre sí.

Existe un plugin de yum que nos ayuda para que de forma automática se eviten estos problemas.

```
# yum install yum-skipbroken
```

Para utilizarlo se usa la opción `--skip-broken` en la línea de comandos.

```
# yum --skip-broken update
```

yum-versionlock

Si quisiéramos usar “drivers” propietarios, tendríamos que asegurar la versión de Xorg y del Kernel. Este plugin es muy útil para usuarios que estén usando drivers propietarios o módulos del kernel en donde les gustaría que ciertos paquetes se mantuvieran como están para evitar “romper” los drivers.

```
# yum install yum-versionlock
```

La documentación acerca de su uso está disponible en `/usr/share/doc/yum-versionlock-%{version}/README`

yum-utils

Para poder solucionar problemas con paquetes, tenemos el comando `package-cleanup`, presente en yum-utils.

Alguna veces, puede suceder que la instalación es interrumpida o ocurre algo con la base de datos de paquetes instalados, y aparecen paquetes duplicados o dependencias rotas. Esta herramienta ayuda a rastrear este tipo de problemas: **package-cleanup**.

```
# yum install yum-
```

```
utils
```

Varios comandos comunes de yum-utils :

```
# package-cleanup --dupes # lista duplicados
# package-cleanup --cleandupes # limpia duplicados #
package-cleanup --problems # lista paquetes con
dependencias rotas
```

Instalando Repositorios de terceros

```
# rpm -ivh http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-
stable.noarch.rpm # rpm -ivh
http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-
stable.noarch.rpm
```