

# **Cómo Crear RPMs**

**Jorge Alberto García**

---

# Cómo Crear RPMs

por Jorge Alberto García

Copyright © 2004 Jorge Alberto García

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation. No hay Secciones Invariantes ni Textos de Portada o Contraportada. Puedes consultar una copia de la licencia en <http://www.gnu.org/copyleft/fdl.html>.

## Disclaimer

Usa esta información bajo tu propio riesgo.

Al usar esta información deslindas de cualquier responsabilidad a el autor, usala bajo tu propio riesgo y responsabilidad.

Todas las marcas son propiedad de sus dueños a menos que se exprese explícitamente de otra forma.

La mención de una marca en este documento es para fin ilustrativo y no debe verse como violación de derechos reservados por la empresa.

Te recomiendo que realices copias de respaldo ANTES de usar la información contenida en este documento y realizarlas regularmente para evitar pérdidas de tu información.

---

---

# Capítulo 1. Introducción

Existen diferentes distribuciones de GNU/Linux y cada una tiene su manejador de paquetes, es decir, la forma en que se instalarán nuevas aplicaciones.

Pueden ser instalados desde las fuentes de la aplicación, es decir compilarl el código fuente a "mano" e instalarlo; o pueden ser desde un paquete para la distribución que se tenga en uso.

En este documento se tratará de explicar como compilar paquetes RPM, esto no te convertirá en Dios, un Gurú ni nada por el estilo, solo te ayudará a hacer más fácil la instalación de alguna aplicación.

---

# Capítulo 2. Crear una llave GNUPG para firmar los paquetes generados.

En esta sección explicaremos como crear una llave GNUPG para firmar nuestros paquetes RPM.

## Nota

Si ya cuentas con una llave GNUPG puedes saltar esta sección

Lo primero que debes ver es si contamos con GNUPG en nuestro sistema, este documento asume que cuentas con el, sino lo tienes instalado, ve a <http://www.gnupg.org> [<http://www.gnupg.org/>] y descarga e instala una versión estable.

Primero vemos si lo tenemos, en una consola de comandos teclea lo siguiente.

```
$ rpm -qa | grep gnupg
```

Y deberás tener el siguiente resultado:

```
$ rpm -qa | grep gnupg
gnupg-x.x.x-x.x
```

Si tienes ese resultado cuentas con GNUPG en tu sistema, las X deberán ser números mostrando la versión que tienes instalada.

Ahora debes generar una llave, en una consola de comandos escribe:

```
$ gpg --gen-key
```

Generar una llave GNUPG es un proceso interactivo, debes responder algunas preguntas, como el tipo de llave, la duración, tu nombre y correo.

Deberás tener una salida similar a esta:

```
$ gpg --gen-key
gpg (GnuPG) 1.2.4; Copyright (C) 2003 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /home/jorge/.gnupg: directorio creado
gpg: creado un nuevo fichero de configuración /home/jorge/.gnupg/gpg.conf
gpg: AVISO: las opciones en /home/jorge/.gnupg/gpg.conf no están aún activas_
en esta ejecución
gpg: anillo /home/jorge/.gnupg/secring.gpg creado
gpg: anillo /home/jorge/.gnupg/pubring.gpg creado
Por favor seleccione tipo de clave deseado:
  (1) DSA y ElGamal (por defecto)
  (2) DSA (sólo firmar)
```

(4) RSA (sólo firmar)

```

Su elección: 1
El par de claves DSA tendrá 1024 bits.
Listo para generar un nuevo par de claves ELG-E.
    el tamaño mínimo es 768 bits
    el tamaño por defecto es 1024 bits
    el tamaño máximo recomendado es 2048 bits
¿De qué tamaño quiere la clave (1024)? 1024
El tamaño requerido es de 1024 bits
Por favor, especifique el periodo de validez de la clave.
    0 = la clave nunca caduca
    n = la clave caduca en n días
    n w = la clave caduca en n semanas
    n m = la clave caduca en n meses
    n y = la clave caduca en n años
¿Validez de la clave (0)? 0
Key nunca caduca
¿Es correcto (s/n)? s

```

Necesita un identificador de usuario para identificar su clave. El programa construye el identificador a partir del Nombre Real, Comentario y Dirección de Correo Electrónico de esta forma:

## Heinrich Heine (Der Dichter)

```
Nombre y apellidos: Jorge Garcia (Garsan)
Dirección de correo electrónico: garsan@garsan.ws
Comentario: Llave para firmar paquetes RPM y correo-e
Ha seleccionado este ID de usuario:
    Jorge Garcia (Garsan) (Llave para firmar paquetes RPM y correo-e)
```

¿Cambia (N)ombre, (C)omentario, (D)irección o (V)ale/(S)alir? v  
Necesita una frase contraseña para proteger su clave secreta.

Es necesario generar muchos bytes aleatorios. Es una buena idea realizar alguna otra tarea (trabajar en otra ventana/console, mover el ratón, usar la red y los discos) durante la generación de números primos. Esto da al generador de números aleatorios mayor oportunidad de recoger suficiente entropía.

+++++.+++++.+++++.+++++.

Es necesario generar muchos bytes aleatorios. Es una buena idea realizar alguna otra tarea (trabajar en otra ventana/console, mover el ratón, usar la red y los discos) durante la generación de números primos. Esto da al generador de números aleatorios mayor oportunidad de recoger suficiente entropía.

+++++ . ++++++ . ++++++ . .

```
gpg: /home/jorge/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
claves pública y secreta creadas y firmadas.
clave marcada como de confianza absoluta.
```

```
pub 1024D/XXXXXXXX 2004-05-20 Jorge Garcia (Garsan) (Llave para firmar
paquetes RPM y correo-e)
```

Huella de clave = XXXX XXXX XXXX XXXX XXXX    XXXX XXXX XXXX XXXX XXXX  
sub 1024g/XXXXXXXXXX 2004-05-20

Después de responder unas cuantas preguntas, ya tendremos nuestra llave lista.

Crear  
una llave GNUPG  
para firmar los pa-  
quetes generados.

---

Para listar y comprobar que se ha creado la llave puedes usar el siguiente comando:

```
$ gpg --list-keys
```

El cual dará el siguiente resultado:

```
$ gpg --list-keys
```

```
gpg: ATENCION ¡se está usando memoria insegura!
```

```
gpg: por favor, vea http://www.gnupg.org/faq.html para más información  
/home/jorge/.gnupg/pubring.gpg
```

```
-----
```

```
pub 1024D/XXXXXXXX 2004-06-22 Jorge Garcia (Garsan) <garsan@garsan.ws>  
sub 2048g/XXXXXXXX 2004-06-22
```

---

# Capítulo 3. Preparación del sistema

Antes de empezar a compilar paquetes, debes configurar tu sistema para realizar dichas compilaciones, y de eso trata este punto.

La primer regla de oro en la construcción de paquetes RPM es que NUNCA se deben de construir como root, siempre se deben construir como usuario normal y la segunda regla es que siempre instales mediante el manejador de paquetes de la distribución que uses, es decir que si usas paquetes RPM siempre instales mediante RPM, si usas fuentes siempre uses fuentes para instalar la aplicación y no mezclar RPM con fuentes, aunque los path de los archivos quedan en la misma ubicación, no suelen encontrarse entre ellos.

Nuestro primero paso es crear un archivo llamado `rpmmacros` en nuestro directorio HOME:

```
$ touch .rpmmacros
```

Lo editas con tu editor favorito:

```
$ nano .rpmmacros
```

Y pegas las siguientes líneas:

```
%_topdir %(echo $HOME)/rpmbuild
%debug_package %{nil}
%packager Garsan
%distribution BasicosFC
%vendor garsan.ws
%tmppath %{_topdir}/tmp
%builddir %{_tmppath}
%rpmtopdir %{_topdir}/%{name}
%_sourcedir %{_rpmtopdir}
%_specdir %{_rpmtopdir}
%_rpmdir %{_topdir}/RPMS
%_srcrpmdir %{_topdir}/SRPMS
%_rpmfilename %{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}.rpm
%_signature gpg
%_gpg_path ~/.gnupg
%_gpg_name Jorge Garcia (Garsan)
%_gpgbin /usr/bin/gpg
```

De estas líneas lo más conveniente cambiar a tu preferencia son:

```
%packager - Puedes poner tu nombre o nick
%distribution - Puedes poner el nombre de tu repositorio o la distribución,
para la cual estas compilando
%vendor - Puedes poner tu correo o URL
```

Con este archivo estamos especificando en que directorios queremos que coloque los temporales mientras compila, los RPM terminados y los SRPMS. Su función no termina ahí, puedes agregar más opciones como el tipo de

procesador en que se está compilando y varias más; en este documento solo manejaremos lo básico para que construyas un paquete básico.

Ya que tenemos el archivo .rpmmacros en nuestro HOME, copiamos la estructura que viene por default para root en nuestro HOME:

```
$ mkdir /home/tu_usuario/rpmbuild
$ cd /usr/src/redhat/
$ cp -r * /home/tu_usuario/rpmbuild/
$ cd /home/tu_usuario/
$ chmod 755 rpmbuild/* -R
```



---

# Capítulo 4. Compilación

Llegamos a la parte buena, la compilación de los paquetes.

Antes de que inicies, debes revisar que tu sistema cuente con las librerías de desarrollo (paquete-devel.rpm) que necesita la aplicación que quieres compilar.

Si quieres compilar una aplicación que no tiene un paquete SRPM tienes que hacer una compilación previa al empaquetado, en los siguientes pasos te explico por que tienes que hacer esto.

Los RPM contienen un archivo en el cual especificas el modo de compilación, que cambios haz hecho, los archivos que tiene la aplicación ya compilada, dependencias y otras opciones.

Un archivo .spec puede ser como el siguiente:

```
Summary : Breve descripción de la aplicación
Name    : El nombre de la aplicación
Version : La versión
Release : Sub-versión o numero de parche
License : La licencia
Group   : El grupo de aplicaciones al que pertenece
Source  : El nombre del archivo fuente
URL     : La URL de la aplicación
BuildRoot : %{_tmppath}/%{name}-root
BuildRequires : Librerías de desarrollo que requiere al momento de compilar
Requires : Librerías o aplicaciones de las cuales depende

%description
Aquí puedes poner una descripción un poco más larga,
en donde expliques que es lo que hace esta aplicación, por ejemplo.

%prep
%setup -q
%build
%configure
make
%makeinstall

%clean
rm -rf %{buildroot}

%files
/bin/aplicacion
/usr/lib/aplicacion/lo_que_contenga

%changelog
* Wed Feb 24 2004 Garsan <garsan@garsan.ws> - 2.4.2-3.garsan
- Construcción inicial para Fedora Core 1
```

Ya tienes tu HOME lista, tu archivo fuente comprimido y tu fichero .spec listo, pero... ¿Cómo sabes que archivos poner en %files? Pues para eso necesitas hacer una compilación previa como indiqué en pasos anteriores.

Primero tienes que descomprimir tus fuentes y hacer una compilación casi normal:

```
$ ./configure
```

```
$ make
```

Si no te marca ningún error ahora tienes que redirigir la instalación a un directorio:

```
$ make DESTDIR=/home/tu_usuario/rpmbuild/tmp/aplicacion install
```

Con esto hace una instalación dirigida a el directorio que le especificamos, de esta forma podemos ver que es lo que crea al momento de instalar, por ejemplo:

```
$ tmp/aplicacion/usr/lib/aplicacion/libreria1
$ tmp/aplicacion/usr/bin/aplicacion-bin
$ tmp/aplicacion/usr/share/man/aplicacion
```

Podemos incluir toda la lista de archivos o podemos usar comodines, por ejemplo:

```
%files
%{_libdir}/aplicacion/*
%{_bindir}/*
%{_mandir}/*
```

O sin comodines sería:

```
%files
%{_libdir}/aplicacion/libreria1
%{_libdir}/aplicacion/libreria2
%{_libdir}/aplicacion/libreria3
%{_bindir}/aplicacion-bin
%{_mandir}/aplicacion
```

¿Qué rayos son esas cosas que tienen %?

Son MACROS que se usan para especificar los directorios por defecto, para hacerlo un poco más uniforme y que tu paquete sea compatible con versiones anteriores o futuras, puedes poner el path completo hacia el archivo que quieres incluir o bien usar un MACRO, lo más recomendable es un MACRO. También puedes usar MACROS para copiar, mover, borrar, etc.

Ahora si, ya tienes tus fuentes, tu HOME preparado y tu .spec listo ¿Qué sigue?

!!!La compilación del RPM!!!

Dentro del directorio rpmbuild deberás hacer un directorio con el nombre de la aplicación, por ejemplo:

```
$ cd rpmbuild
$ mkdir aplicacion
```

Y dentro de este directorio colocar el .spec y la fuente de la aplicación que quieres compilar, por ejemplo:

```
$ cd rpmbuild/aplicacion
$ ls
aplicacion.tar.bz2 aplicacion.spec
```

Ya no te mareo tanto y vamos a la hora de la verdad, al momento en que verás si lo que haz hecho está bien. En una consola como usuario normal, escribes lo siguiente:

## NOTA

Recuerda que no debes hacer la compilación como root sino como usuario normal.

```
$ cd rpmbuild/aplicacion
$ rpmbuild -ba aplicacion.spec
```

```
$ cd rpmbuild/aplicacion
$ rpmbuild -ba --sign aplicacion.spec
```

Y empieza la preparación y compilación de tu aplicación, verás algo como esto:

```
$ rpmbuild -ba --sign xmule.spec
Enter pass phrase:
Pass phrase is good.
Executing(%prep): /bin/sh -e /home/jorge/rpmbuild/tmp/rpm-tmp.54455
+ umask 022
+ cd /home/jorge/rpmbuild/tmp
+ LANG=C
+ export LANG
+ unset DISPLAY
+ cd /home/jorge/rpmbuild/tmp
+ rm -rf xmule-1.7.4b
+ /usr/bin/bzip2 -dc /home/jorge/rpmbuild/xmule/xmule-1.7.4b.tar.bz2
+ tar -xf -
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd xmule-1.7.4b
++ /usr/bin/id -u
+ '[' 500 = 0 ']'
++ /usr/bin/id -u
+ '[' 500 = 0 ']'
+ /bin/chmod -Rf a+rX,g-w,o-w .
Executing(%build): /bin/sh -e /home/jorge/rpmbuild/tmp/rpm-tmp.24968
+ umask 022
+ cd /home/jorge/rpmbuild/tmp
+ cd xmule-1.7.4b
+ LANG=C
+ export LANG
+ unset DISPLAY
+ CFLAGS=-march=pentium4 -O3 -pipe -fomit-frame-pointer
```

```
+ export CFLAGS
+ CXXFLAGS=-march=pentium4 -O3 -pipe -fomit-frame-pointer
+ export CXXFLAGS
+ FFLAGS=-O2 -g -pipe -march=i386 -mcpu=i686
+ export FFLAGS
++ find . -name config.guess -o -name config.sub
++ basename ./config.guess
+ '[' -f /usr/lib/rpm/redhat/config.guess ']'
+ /bin/rm -f ./config.guess
++ basename ./config.guess
+ /bin/cp -fv /usr/lib/rpm/redhat/config.guess ./config.guess
+ /usr/lib/rpm/redhat/config.guess -> ./config.guess
++ basename ./config.sub
+ '[' -f /usr/lib/rpm/redhat/config.sub ']'
+ /bin/rm -f ./config.sub
++ basename ./config.sub
+ /bin/cp -fv /usr/lib/rpm/redhat/config.sub ./config.sub
+ /usr/lib/rpm/redhat/config.sub -> ./config.sub
+ ./configure --host=i686-pc-linux-gnu --build=i686-pc-linux-gnu
--target=i386-redhat-linux-gnu
--program-prefix= --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin
--sbindir=/usr/sbin
--sysconfdir=/etc --datadir=/usr/share --includedir=/usr/include
--libdir=/usr/lib
--libexecdir=/usr/libexec --localstatedir=/var --sharedstatedir=/usr/com
--mandir=/usr/share/man
--infodir=/usr/share/info
.....
```

Si no hay ningún error al momento de compilar, verás algo como esto:

```
Checking for unpackaged file(s): /usr/lib/rpm/check-files
/home/jorge/rpmbuild/tmp/xmule-1.7.4b-2.garsan-root
Generating signature: 1005
Wrote: /home/jorge/rpmbuild/SRPMS/xmule-1.7.4b-2.garsan.src.rpm
Generating signature: 1005
Wrote: /home/jorge/rpmbuild/RPMS/xmule-1.7.4b-2.garsan.i386.rpm
Executing(%clean): /bin/sh -e /home/jorge/rpmbuild/tmp/rpm-tmp.34717
+ umask 022
+ cd /home/jorge/rpmbuild/tmp
+ cd xmule-1.7.4b
+ rm -rf /home/jorge/rpmbuild/tmp/xmule-1.7.4b-2.garsan-root
+ exit 0
```

Si ves las líneas que dicen Wrote: xxxxxxxxxxxxxxxxxxxx.rpm.....

*¡¡¡FELICIDADES!!!* tu paquete RPM está listo, solo te queda instalarlo y probarlo durante un tiempo, sino tienes errores y funciona como debería de funcionar, pues puedes empezar a distribuirlo en repositorios, anunciarlo en listas o como tu quieras. Si quieres ver un poco de información de tu paquete puedes hacerlo así:

```
$ cd rpmbuild/RPMS
$ rpm -qip paquete.rpm
```

Y la salida sería como esta:

```
Name       : xmule                      Relocations: (not relocateable)
Version    : 1.7.4b                    Vendor: garsan.ws
Release    : 2.garsan                  Build Date: vie 05 mar 2004 23:31:08 CST
Install Date: (not installed)          Build Host: garsan.garsan.ws
Group      : Applications/Internet     Source RPM: ↵
xmule-1.7.4b-2.garsan.src.rpm
Size       : 3814899                    License: GPL
Signature  : DSA/SHA1, vie 05 mar 2004 23:31:14 CST, Key ID 947e16e7475c436e
Packager   : Garsan
URL        : http://www.xmule.org/
Summary    : Easy to use client for ED2K Peer-to-Peer Network based on eMule
Description:
xmule is an easy to use multi-platform client for ED2K Peer-to-Peer Network.
It is originally based on eMule, the popular windows-only client for the
same network.
```

Si quieres ver los archivos que tiene el RPM puedes hacerlo así:

```
$ rpm -qlp archivo.rpm
```

Y la salida sería:

```
/usr/bin/ed2k.xmule
/usr/bin/xmule
/usr/share/applications/xmule.desktop
/usr/share/doc/xmule-1.7.4b
/usr/share/doc/xmule-1.7.4b/AUTHORS
/usr/share/doc/xmule-1.7.4b/COPYING
/usr/share/doc/xmule-1.7.4b/ChangeLog
/usr/share/doc/xmule-1.7.4b/README
/usr/share/doc/xmule-1.7.4b/TODO
/usr/share/locale/de/LC_MESSAGES/xmule.mo
/usr/share/locale/ee/LC_MESSAGES/xmule.mo
/usr/share/locale/es/LC_MESSAGES/xmule.mo
/usr/share/locale/es_MX/LC_MESSAGES/xmule.mo
/usr/share/locale/fr/LC_MESSAGES/xmule.mo
/usr/share/locale/it/LC_MESSAGES/xmule.mo
/usr/share/locale/ko/LC_MESSAGES/xmule.mo
/usr/share/locale/pl/LC_MESSAGES/xmule.mo
/usr/share/locale/tr/LC_MESSAGES/xmule.mo
/usr/share/pixmaps/xmule.xpm
```

---

# Capítulo 5. Macros del Sistema

Los macros son muy útiles al momento de hacer un paquete RPM ya que nos ahorran trabajo.

Por ejemplo, si tenemos que escribir el nombre de la aplicación varias veces podemos usar el macro name:

```
%{name}
```

O podemos definir nuestros macros con variables que se usan varias veces en nuestro .spec; esto se hace en el principio de nuestro .spec

```
%define _kde_applnk    /usr/share/applnk

Name      :  amsn
Version   :  0_90
Release   :  1.garsan
Group     :  Internet
Summary   :  Clon de Messenger
Source    :  %{name}-%{version}.tar.gz
URL: http://amsn.sf.net
Copyright :  GPL
BuildRoot:  %{_tmppath}/%{name}-root
BuildRequires: tcl-devel imlib-devel tk-devel
Requires:  tcl tk imlib
```

Con las primeras líneas que tienen "%define" estamos indicando que macro definimos, por ejemplo estamos especificando que "\_kde\_applnk" corresponde a "/usr/share/applnk" y lo usamos como "%{\_kde\_applnk}"

```
mkdir -p $RPM_BUILD_ROOT%{_kde_applnk}
```

Algunas macros más usadas en el empaquetado de RPMs son las siguientes:

Macros para directorios del sistema

## MACRO DIRECTORIO

---

```
%{_sysconfdir} /etc/
%{_bindir}     /usr/bin/
%{_libdir}     /usr/lib/
%{_datadir}    /usr/share/
%{_mandir}     /usr/share/man/
%{_includedir} /usr/include/
%{_infodir}    /usr/share/info/
%{_localstatedir} /var/
%{_libexecdir} /usr/lib/exec/
%{_prefix}     /usr/
%{_sbindir}    /usr/sbin/
```

Macros para comandos del sistema

**MACRO COMANDO**

---

```
%{__cat} /bin/cat
%{__perl} /usr/bin/perl
%{__rm} /bin/rm
%{__make} /usr/bin/make
%{__install} /usr/bin/install
%{__mkdir} /bin/mkdir
%{__mv} /bin/mv
%{__cp} /bin/cp
%configure ./configure
%makeinstall make install
```

---

# Capítulo 6. Ejemplos de .spec de aplicaciones.

Para fines de ilustración incluyo un par de archivos .spec que te pueden servir para generar los tuyos.

flashplayer.spec - Para generar el plugging Flash de Macromedia para Mozilla

```
Summary: Macromedia Flash Player
Name: flashplayer
Version: 6.0r79
Release: 3.garsan
License: Macromedia Software End User License
Group: Applications/Internet
Source: install_flash_player_6_linux.tar.gz
URL: http://www.macromedia.com/software/shockwaveplayer/index.html
BuildRoot: %{_tmppath}/%{name}-%{version}-root
Requires: mozilla

%description
Macromedia Flash Player is the leading rich client for Internet content
and applications across the broadest range of platforms.

%prep

%setup -q -c %{name}-%{version}

%build

%install
rm -rf %{buildroot}
install -D -m 755 install_flash_player_6_linux/flashplayer.xpt \
    $RPM_BUILD_ROOT%{_libdir}/mozilla/plugins/flashplayer.xpt
install -m 755 install_flash_player_6_linux/libflashplayer.so \
    $RPM_BUILD_ROOT%{_libdir}/mozilla/plugins

%clean
rm -rf %{buildroot}

%files
%defattr(-,root,root)
%doc install_flash_player_6_linux/readme.htm
%doc install_flash_player_6_linux/readme.txt
%{_libdir}/mozilla/plugins/*

%changelog
* Tue Dec 23 2003 Garsan garsan_AT_garsan.ws
- reconstruido para Fedora Core 1
```

amsn.spec - Para generar un RPM de aMSN y compilar su Traydoc

```
%define _kde_applnk    /usr/share/applnk
%define _gnome_applnk  /etc/X11/applnk
%define _applnk_cat Internet
```



```
%define _pixmap          /usr/share/pixmaps/amsn

Name      :   amsn
Version   :   0_90
Release   :   1.garsan
Group     :   Internet
Summary   :   Clon de Messenger
Source    :   %{name}-%{version}.tar.gz
URL: http://amsn.sf.net
Copyright :   GPL
BuildRoot:   %{_tmppath}/%{name}-%{PACKAGE_VERSION}-root
BuildRequires: tcl-devel imlib-devel tk-devel
Requires:    tcl tk imlib

%description
This is a modified and multilanguage version of Compu'S Messenger (CCMSN).
It has many new features, but we're still working on it.

%prep
%setup -q -c msn
%install
rm -rf %{buildroot}
mkdir -p $RPM_BUILD_ROOT%{_libdir}/amsn \
$RPM_BUILD_ROOT%{_kde_applnk}/%{_applnk_cat}/ \
$RPM_BUILD_ROOT%{_gnome_applnk}/%{_applnk_cat}/ \
$RPM_BUILD_ROOT%{_pixmap}/
cd msn
cp -a * $RPM_BUILD_ROOT%{_libdir}/amsn
cp -a $RPM_BUILD_ROOT%{_libdir}/amsn/amsn.desktop \
$RPM_BUILD_ROOT%{_gnome_applnk}/%{_applnk_cat}/
cp -a $RPM_BUILD_ROOT%{_libdir}/amsn/amsn.desktop \
$RPM_BUILD_ROOT%{_kde_applnk}/%{_applnk_cat}/
cp -a $RPM_BUILD_ROOT%{_libdir}/amsn/icons/* $RPM_BUILD_ROOT%{_pixmap}/
cd $RPM_BUILD_ROOT%{_libdir}/amsn/plugins/traydock/
./configure
make

%clean
rm -rf %{buildroot}

%post
ln -s /usr/lib/amsn/amsn /usr/bin/amsn

%files
%defattr(755,root,root)
%doc msn/FAQ msn/GNUGPL msn/HELP msn/README msn/TODO
%{_libdir}/amsn/*
%{_gnome_applnk}/%{_applnk_cat}/*
%{_kde_applnk}/%{_applnk_cat}/*
%{_pixmap}/*

%changelog
* Tue Feb 17 2004 Garsan garsan_AT_garsan.ws
- Construido para Fedora Core 1
```

epiphany.spec - Para construir epiphany

```
%define mversion 38:1.6
Summary: Web browser based on the mozilla rendering engine.
Name: epiphany
Version: 1.0.8
Release: 1.garsan
License: GPL
Group: Applications/Internet
URL: http://www.gnome.org/projects/epiphany/
Source:␣
http://ftp.gnome.org/pub/GNOME/sources/epiphany/1.0/epiphany-%{version}.tar.bz2

BuildRoot: %{_tmppath}/root-%{name}-%{version}
Prefix: %{_prefix}
BuildRequires: mozilla-devel = %{mversion}, gtk2-devel, libbonoboui-devel␣
>= 2.1.1
BuildRequires: scrollkeeper, nautilus

Requires: mozilla = %{mversion}
Requires(post): scrollkeeper

%description
Epiphany is a GNOME web browser based on the mozilla rendering engine.
The name meaning: "An intuitive grasp of reality through something (as
an event) usually simple and striking".

%package devel
Summary: Header files, libraries and development documentation for %{name}.
Group: Development/Libraries
Requires: %{name} = %{version}-%{release}

%description devel
This package contains the header files, static libraries and development
documentation for %{name}. If you like to develop programs using %{name},
you will need to install %{name}-devel.

%prep
%setup

%build
%configure \
  --disable-dependency-tracking \
  --enable-nautilus-view="yes" \
  --enable-compile-warnings="no" \
  --disable-schemas-install
%{__make} %{?_smp_mflags}

%install
%{__rm} -rf %{buildroot}
export GCONF_DISABLE_MAKEFILE_SCHEMA_INSTALL="1"
%makeinstall
%find_lang %{name}

%{__rm} -rf %{buildroot}%{_localstatedir}/scrollkeeper/

%post
export GCONF_CONFIG_SOURCE="$(gconftool-2 --get-default-source)"
gconftool-2 --makefile-install-rule␣
```

```
%{_sysconfdir}/gconf/schemas/%{name}.schemas
scrollkeeper-update -q || :

%postun
scrollkeeper-update -q || :

%clean
%{__rm} -rf %{buildroot}

%files -f %{name}.lang
%defattr(-, root, root, 0755)
%doc AUTHORS ChangeLog COPYING* NEWS README TODO
%doc %{_datadir}/gnome/help/epiphany/
%config %{_sysconfdir}/gconf/schemas/*.schemas
%{_bindir}/*
%{_libdir}/bonobo/servers/*.server
%{_datadir}/application-registry/*.applications
%{_datadir}/applications/*.desktop
%{_datadir}/epiphany/
%{_datadir}/omf/epiphany/
%{_datadir}/pixmaps/*

%files devel
%defattr(-, root, root, 0755)
%{_libdir}/pkgconfig/*.pc
%{_includedir}/epiphany-1.0/

%changelog
* Sun Mar 21 2004 Garsan garsan_AT_garsan.ws
- Reconstruccion en Fedora Core 1
- Optimizado para i686
```

## Nota

Estos ejemplos pueden de ser de versiones muy viejas o descontinuadas, solo son presentados para fines de ilustración.

---

# Capítulo 7. Créditos

Jorge Alberto García <garsan\_AT\_garsan.ws>

Esta es el documento "Como construir paquetes RPM" y puede contener *errores*, el autor se *deslinda de cualquier responsabilidad* si tu sistema es dañado, tu cerebro colapsa o cualquier otra cosa mala que pueda suceder, usalo bajo tu responsabilidad. Si quieres hacer notar un error, agregar algo para futuras publicaciones o simplemente saludar puedes hacerlo al correo <garsan\_AT\_garsan.ws> y visitando el sitio [<http://www.garsan.ws/>]

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation. No hay Secciones Invariantes ni Textos de Portada o Contraportada. Puedes consultar una copia de la licencia en <http://www.gnu.org/copyleft/fdl.html>.