Empaquetando software para Debian: Herramientas y procesos básicos

Gunnar Wolf

Desarrollador del Proyecto Debian

DebConf12 — 8 al 15 de julio, 2012

Índice

. . . Así que quieres participar en Debian

- La motivación básica para esta ponencia (y este track)
- . . . Puede ser difícil saber por dónde comenzar
- Hagas lo que hagas, comprender lo que son los paquetes probablemente será fundamental

Paquetes: La unidad básica

- El principal mecanismo de distribución de *binarios* (programas ya compilados)
- *Recetas* completas de compilación en paquetes fuente
- Información de *dependencias* que nos otorgan un sistema funcional, completo
- Una *suite* es un conjunto de paquetes de determinadas versiones, que *están ampliamente probados en su conjunto*
  - Se puede mezclar paquetes de diferentes *suites*, pero no se recomienda... Especialmente a usuarios finales
  - ¡Y recuerda que nuestra prioridad son los usuarios!

Infraestructura de manejo de paquetes

- `apt` (A Package Tool), ~1998
  - `apt-get`, `aptitude`, `synaptic`, etc.
- Una de las inovaciones que presentó Debian es la de una infraestructura que maneja *listas* de paquetes
- Entre sus principales funciones:
  - Resolución de dependencias y conflictos
  - Obtención de paquetes (de Internet, de CD)
  - Llamar a `dpkg` en el órden correcto

¿Y por qué quiero manejar paquetes?

- Para agregar software nuevo a Debian
- Para adoptar paquetes *huérfanos*
- Para corregir bugs
- Para traducir las cadenas de un programa
- ...

Paquetes fuente, paquetes binarios

- El *usuario* trabaja con *paquetes binarios*
  - Archivos `.deb`
  - Los que están compilados para su computadora (arquitectura) específica y listos para usar
- El *desarrollador* trabaja con *paquetes fuente*
  - Aquellos a partir de los cuales se compilan los paquetes binarios
  - Dos (típicamente tres) o más archivos: `*.orig.tar.gz`, `*.debian.tar.gz` y `*.dsc`
  - Para trabajar, *desempaquetamos* el paquete fuente y trabajamos en el árbol resultante (`dpkg-source -x`)
- Al construir el paquete, se generan ambos
  - El `.orig.tar.gz` se mantiene idéntico (*fuentes prístinas*)

Índice

Desde fuera

El paquete fuente consta de (típicamente) tres archivos[1], todos ellos iniciando con el nombre del paquete y su versión:

`.orig.tar.gz` Fuentes provenientes del autor. *No se modifican* (mantiene el mismo checksum, es verificable que no tuvo ninguna modificación)

`.debian.tar.gz` Un árbol parcial, que se desempaqueta dentro del directorio `debian/`.

`.dsc` Descripción autogenerada con las dependencias de construcción del paquete e información básica de sus mantenedores.
Lleva la firma GPG de quien la sube al archivo.

[1] Hay otras estructuras que pueden generar sólo dos, o más de tres archivos, o un archivo `.diff.gz`, pero quedan fuera del ámbito de esta plática

Dentro del paquete (1)

Toda la magia se hace dentro del directorio `debian/` dentro del directorio fuentes. Puede haber muchos archivos, pero los cuatro fundamentales son:

`control` Información básica del paquete fuente: Descripción, mantenedor, los paquetes binarios a generar, relaciones con otros paquetes.

`rules` Archivo interpretado por *make* con las instrucciones para construir el paquete binario. Típicamente usa a un *asistente de construcción* como `dh` (*debhelper*).

Dentro del paquete (2)

`copyright` Información *detallada* de los derechos de autor. Puede ser a texto corrido, pero estamos migrando a un formato estandarizado, analizable (DEP5)

`changelog` Bitácora con la historia del paquete, en un formato estandarizado.

Todas las modificaciones que hagamos al paquete deben estar dentro del directorio `debian/`.

¿Y cómo se come?

- Para bajar un paquete fuente: #+srcname baja$_{\text{paqfuente}}$

```
$ apt-get source unpaquete
(...)
$ ls unpaquete*
unpaquete-2.1  unpaquete_2.1-2.debian.tar.gz
unpaquete_2.1.orig.tar.gz  unpaquete_2.1-2.dsc
$
```

- Para construirlo: #+srcname construye$_{\text{paqfuente}}$

```
$ cd unpaquete-2.1
$ dpkg-buildpackage
```

- Y el resultado esperado, `unpaquete_2.1-1_i386.deb` (o algo equivalente).

Índice

debian/control (1)

- Es un archivo de formato RFC 822 (como los encabezados del correo — *campo*: *valor*)
- Está separado en varios *párrafos*; el primero describe al paquete fuente, y el segundo (y subsecuentes) a los paquetes binarios.

```
Source: unpaquete
Section: interpreters
Priority: optional
Maintainer: Gunnar Wolf <gwolf@debian.org>
Uploaders: Fulano D. Tal <fulano@tal.org>
Build-Depends: libotropkg-dev
Standards-Version: 3.9.3
(...)
```

debian/control (2)

```
Package: unpaquete
Architecture: any
Depends: otropkg
Recommends: unpaquete-data
Description: Implementa la lógica de un paquete
 Explicamos acá más a detalle los detalles de por qué
 este paquete binario puede resultar del interés de
 alguien

Package: unpaquete-data
Architecture: all
Description: Datos útiles para unpaquete
 Este paquete tiene los datos que típicamente usa
 unpaquete.
```

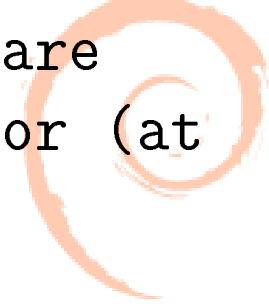
debian/copyright

- Hoy en día se recomienda seguir un formato estandarizado, *parseable* por computadora, que nos proporcionará la posibilidad de hacer verificaciones automatizadas.
  La descripción del formato está en http://dep.debian.net/deps/dep5/
- Es importante verificar la información de autoría de *todos los archivos* que forman parte del paquete fuente
  - Muchas veces los autores incluyen copias completas de otros paquetes (*bundling*). *Hay que documentar todo esto* en `debian/copyright`
  - Si alguna parte *no fundamental* del paquete no es DFSG-libre, podemos reempaquetar el `.orig.tar.gz` omitiéndolo; típicamente ajustamos el número de versión para explicitarlo (0.1.2 se vuelve 0.1.2+dfsg)

debian/copyright

```
Format: http://www.debian.org/doc/packaging-manuals/
            copyright-format/1.0/
Upstream-Name: unpaquete
Source: ftp://ftp.example.com/pub/unpaquete

Files: *
Copyright: Copyright 2012 Tal por Cual <tal@cual.com>
License: GPL-2+
 This program is free software; you can redistribute it
 and/or modify it under the terms of the GNU General
 Public License as published by the Free Software
 Foundation; either version 2 of the License, or (at
 your option) any later version.
 (...)
```



debian/changelog

```
unpaquete (0.1.2-1) unstable; urgency=low

  * New upstream release
  * Fixed a bug when foo and bar (Closes: #613263)

 -- Gunnar Wolf <gwolf@debian.org>  Fri, 29 Jun 2012 1(

unpaquete (0.1-1) unstable; urgency=low

  * Initial upload (Closes: #501535)

 -- Gunnar Wolf <gwolf@debian.org> Thu, 21 Jun 2012 18:
```

debian/rules (1)

- Es la lista de instrucciones para construir el paquete
- Es un archivo escrito en `GNU Make`, con varios *objetivos*, incluyendo `build`, `clean`, `binary`, y subobjetivos como `binary-arch` y `binary-indep`
- Lo más común (especialmente para los casos simples) es usar al asistente *DebHelper* (`dh`), que permite tener archivos `debian/rules` de tres líneas: `#+srcname rules`

```
#!/usr/bin/make -f
%:
	dh $@
```

debian/rules (2)

- Con versiones recientes de dh es común, más que escribir objetivos, escribir las porciones en que el proceso de construcción se *desvía* del estándar (overrides):

```
override_dh_install:
        dh_install
        mv debian/unpaquete/var/lib/unpaquete/data \
           debian/unpaquete-data/var/lib/unpaquete
```

Otros archivos

- Dentro del directorio `debian/` puede haber muchos archivos adicionales
- `dh` recibe muchos de sus parámetros de operación a través de algunos archivos adicionales, con listados simples en texto: *paqbin*.* — `dirs`, `install`, `manpages`, `docs`, etc.
- También están los scripts de mantenedor — Scripts en shell que se realizan antes o después de instalar o remover el paquete (`preinst`, `postinst`, `prerm`, `postrm`)

En resumen, necesitas saber...

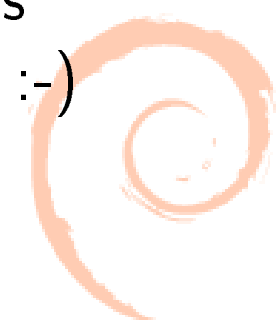
Para construir paquetes de Debian, tienes que saber:

1. Programación básica en shell Bourne (`bash` o `dash`)
2. Sintaxis básica de `GNU Make`
3. Estructurar información en un formato de texto plano
4. Familiarizarte con la familia de comandos de `dh` (tu mejor aliado: `man`)
5. ...Muchas convenciones internas que hemos desarrollado con el paso de los años ;-)

Índice

Empaquetar es sólo una de las actividades. . .

- Hay muchas maneras de participar en Debian
- Pero si entras a colaborar en serio, tarde o temprano terminarás metiendo las manos en un paquete
  - . . . Aunque sea para corregir un bug
  - O para hacer alguna traducción
  - O para tu uso local
  - O. . .
- Y si esto no es para tí, ¡no te preocupes! Hay muchas otras áreas en las que seguramente puedes participar :-)



Referencias útiles

- La guía del nuevo mantenedor: http://www.debian.org/doc/manuals/maint-guide/
- La Referencia del Desarrollador Debian: http://www.debian.org/doc/manuals/developers-reference/index.en.html
- Guía de creación de paquetes Debian: http://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.es.pdf
- Políticas de Debian: http://www.debian.org/doc/debian-policy/

¿Dudas?

¡Gracias!

¿Alguna duda?

gwolf@debian.org
http://gwolf.org/soft/pkg-debian