

# Empaquetando software para distribuciones basadas en Debian

Gabriel Saldaña    Gunnar Wolf

Facultad de Ingeniería, UNAM

7 de agosto de 2015

# Índice

1 ¿Paquetes?

2 Paquetes fuente

3 Estructura

# Paquetes: La unidad básica de una distribución

- 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 *distribución* es un conjunto de paquetes de determinadas versiones, que *están ampliamente probados en su conjunto*

# ¿Por qué empaquetar?

Discusión sin guión

Incluir referencia a «In defence of curl | sudo bash» (Daniel Silverstone)

# Infraestructura de manejo de paquetes en Debian

- apt (A Package Tool),  $\approx$  1998
  - apt-get, aptitude, synaptic, etc.
- Una de las innovaciones 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 orden correcto

## ¿Y por qué quiero manejar paquetes?

- Para agregar software nuevo a la distribución
- Para colaborar con el mantenimiento de paquetes *huérfanos*
- Para corregir bugs
- Para traducir las cadenas de un programa
- Para facilitar la distribución de mi software
  - Aunque sea localmente
- ...

# 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`<sup>1</sup>
  - 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*)

---

<sup>1</sup>Hoy en día pueden comprimirse en otros formatos, como `.tar.bz2` o `.tar.xz`

# Índice

1 ¿Paquetes?

2 Paquetes fuente

3 Estructura



## Desde fuera

El paquete fuente consta de (típicamente) tres archivos<sup>2</sup>, 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.

---

<sup>2</sup>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 presentación

## 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:

```
1 $ apt-get source unpaquete
2 (...)
3 $ ls unpaquete*
4 unpaquete-2.1 unpaquete_2.1-2.debian.tar.gz
5 unpaquete_2.1.orig.tar.gz unpaquete_2.1-2.dsc
6 $
```

- Para construirlo:

```
1 $ cd unpaquete-2.1
2 $ dpkg-buildpackage
```

- Y el resultado esperado, unpaquete\_2.1-1\_amd64.deb (o algo equivalente).

# Índice

1 ¿Paquetes?

2 Paquetes fuente

3 **Estructura**

# 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 subsiguientes) a los paquetes binarios.

```
1 Source: unpaquete
2 Section: interpreters
3 Priority: optional
4 Maintainer: Gunnar Wolf <gwolf@debian.org>
5 Uploaders: Fulano D. Tal <fulano@tal.org>
6 Build-Depends: libotropkg-dev
7 Standards-Version: 3.9.3
8 (...)
```

## debian/control (2)

```
1 Package: unpaquete
2 Architecture: any
3 Depends: otropkg
4 Recommends: unpaquete-data
5 Description: Implementa la lógica de un paquete
6     Explicamos acá más a detalle los detalles de por qué
7     este paquete binario puede resultar del interés de
8     alguien
9
10 Package: unpaquete-data
11 Architecture: all
12 Description: Datos útiles para unpaquete
13     Este paquete tiene los datos que típicamente usa
14     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

```
1  Format: http://www.debian.org/doc/packaging-manuals/  
2      copyright-format/1.0/  
3  Upstream-Name: unpaquete  
4  Source: ftp://ftp.example.com/pub/unpaquete  
5  
6  Files: *  
7  Copyright: Copyright 2012 Tal por Cual <tal@cual.com>  
8  License: GPL-2+  
9      This program is free software; you can redistribute it  
10     and/or modify it under the terms of the GNU General  
11     Public License as published by the Free Software  
12     Foundation; either version 2 of the License, or (at  
13     your option) any later version.  
14     (...)
```

# debian/changelog

```
1  unpaquete (0.1.2-1) unstable; urgency=low
2
3  * New upstream release
4  * Fixed a bug when foo and bar (Closes: #613263)
5
6  -- Gunnar Wolf <gwolf@debian.org>  Fri, 29 Jun 2012 10:07:57
   -0500
7
8  unpaquete (0.1-1) unstable; urgency=low
9
10 * Initial upload (Closes: #501535)
11
12 -- Gunnar Wolf <gwolf@debian.org>  Thu, 21 Jun 2012 18:30:01
   -0500
```

# 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:

```
1  #!/usr/bin/make -f
2  %:
3      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`):

```
1  override_dh_install:  
2      dh_install  
3      mv debian/unpaquete/var/lib/unpaquete/data \\  
4          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:  
*unpaquete.\** — `dirs`, `install`, `manpages`, `docs`, etc.
  - El nombre del archivo se toma de *cada paquete binario* a construir
- 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 ;-)

## 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/>