# M8. Instalador DEB para programas Python[1]

- **Paso 0)** Primero es necesario tener instalados los siguientes paquetes *dpkg-dev* y *dh-make*, los cuales pueden instalarse de la siguiente forma:

```
$ sudo apt-get install dpkg-dev dh-make
```

Tomamos todo el contenido de la carpeta donde actualmente estaría nuestro programa que se ha desarrollado..

- **Paso 1)** Primero vamos a necesitar crear una carpeta donde trabajar:

```
$ mkdir deb  && cd deb
```

Luego una vez dentro de esa carpeta, procedemos a crear una carpeta con el nombre de nuestro programa y la versión, vamos a suponer que el programa consta de los siguientes archivos, *hola.py, hola.png* y la base de datos *hola.dat* :

```
$ mkdir hola-1.2
```

En este último directorio creado debemos copiar una versión empaquetada de nuestro programa, la cual podemos realizar de la siguiente forma:

(Accedemos a la carpeta de nuestro programa, supongamos que sea */home/alumno/hola*)

```
$ cd /home/alumno/hola

$ tar cfzv hola-1.2.tar.gz *

$ cp * /home/deb/hola-1.2  //copiamos todo dentro y entramos

$ cd /home/deb/hola-1.2
```

El empaquetado creado con "tar" debe llevar el nombre y versión del programa.

Los **fuentes (para entendernos el programa)** que fueron empaquetados dentro de "*hola-1.2.tar.gz*" **también es necesario copiarlos** dentro de la misma carpeta ("*/home/deb/hola-1.2*")

- **Paso 2)** Dentro de **nuestra carpeta de trabajo donde quedamos después del último comando** ejecutado, utilizamos el comando **dh_make** (el cual es uno de los programas instalados al iniciar) de la siguiente forma:

```
$ sudo dh_make -e diego.sarmentero@gmail.com  -f hola-1.2.tar.gz -c GPL
```

Este comando se encarga del proceso de "debianización" del directorio. Los parámetros

1 Extraído de http://www.diegosarmentero.com/2010/03/instalador-deb-para-programas-python.html

especificados son los siguientes:

- **-e micorreo@gmail.com**: es el e-mail del creador del programa
- **-f hola-1.2.tar.gz**: indica el nombre del paquete comprimido de nuestro programa
- **-c GPL**: indicamos la licencia del programa

Luego de haber ejecutado el último comando, podremos ver que **se ha creado una carpeta llamada "debia**n" en el directorio en el que nos encontramos.

**Paso 3)** Entramos dentro de la carpeta "debian" y nos encontramos con una serie de archivos, algunos de ellos no serán de utilidad para nuestro propósito, por lo que podemos eliminarlos:

```
$ sudo rm *.ex *.EX
```

Ahora procedemos a editar los demás archivos para nuestro programa:

- **CHANGELOG**
  Simplemente (si no tenemos historiales de cambios o cosas asi) es necesario colocar la versión del programa y el nombre del autor

```
hola (1.2) stable;

-- Mi Nombre Fri, 05 Feb 2015 17:50:57 -0300
```

- **COMPAT**
  No es necesario modificar este archivo, así que lo dejamos como está.

- **CONTROL**
  Este es uno de los mas importante, ya que establece las categorias, dependencias, y otras configuraciones del programa. Para *hola* la configuración quedo como sigue y se explica a continuación:

```
Source: hola

Section: utility

Priority: extra

Maintainer: Mi nombre

Build-Depends: debhelper (>= 7)

Standards-Version: 3.8.1

Homepage:
Package: hola

Architecture: any

Depends: python (>=2.5.2-1ubuntu1), python-qt4 (>=4.4.4-2), sqlite (>=2.8.17-6build1), python-beautifulsoup (>=3.1.0.1-2)Description: Hola  utility
```

Los detalles a configurar mas relevantes son:

- **Source:** (OPCIONAL) el nombre del paquete fuente del que proviene él paquete binario, en caso de que ambos sean distintos. En este caso se podría omitir
- **Section:** (OPCIONAL) define la categoría del paquete. Una forma simple de verlo es que dentro de esta categoría lo podremos encontrar en el menú de Aplicaciones. Otras categorías son: "games", "devel", "text", "video", "sound", etc.
- **Maintainer y Homepage:** Es la información del desarrollador (o el que mantenga) el programa y la web del mismo programa
- **Package:** (CAMPO NECESARIO) determina el nombre del paquete.
- **Architecture:** La arquitectura para la que se compiló el paquete. Arquitecturas comunes son ‘i386’, 'amd64', ‘m68k’, ‘sparc’, ‘alpha’, ‘powerpc’, etc. La opción "all" es para paquetes independientes de la arquitectura como lo son los scripts de Perl o la documentación.
- **Depends:** Este quizás sea el mas importante de configurar si nuestro programa necesita de alguna librería adicional o lo que fuera. Lo que necesitamos poner es el nombre del paquete con el que se tiene dependencia y a partir de que versión (según con la que hayamos trabajado para hacer el programa). Es importante respetar el formato que se muestra arriba. Los paquetes y sus versiones se pueden consultar en el Gestor De Paquetes.
- **Description:** en la documentación de figura como NECESARIO, así que por las dudas poner aunque sea una pequeña descripción del proyecto... no cuesta nada

- **COPYRIGHT**

Para este archivo solo es necesario el nombre del autor y la url de la página del proyecto.

- **DIRS**

Contiene la ruta de los directorios que el paquete necesitará para para instalar el programa. En el caso de *hola*:

```
usr/bin
usr/share/hola
usr/share/applications
```

- **DOCS**

Puede permanecer vacio, pero si hay alguna documentación referente al proyecto que se pueda agregar, ese podría ser un lugar donde colocarla.

- **README.Debian**

En él ponemos comentarios que consideremos importantes para quien vaya a utilizar el programa.

Este archivo no es obligatorio, agreguar simplemente lo siguiente:

```
hola for Debian/Ubuntu
----------------------
http://code.google.com/p/hola/        web con el código

-- My name  Fri, 05 Feb 2010517:50:57 -0300
```

- **RULES**

Este archivo tiene una configuración un tanto extensa, aunque no complicada.
La única sección que sería necesario modificar es: "**install: build**", cambiando donde debe figurar el nombre del programa *hola*. Además hice un par de modificaciones. En el **ANEXO I** dejo el fichero con alguna recomendación. Construidlo en este paso

- **Paso 4)** Luego de esto **dejamos la carpeta *debian*** y volvemos a la carpeta *hola-1.2* que contenía a *debian*. Un detalle importante es en cuanto al manejo de **imágenes y archivos** creados por el programa, generalmente iconos y base de datos que son accedidas mediante rutas relativas.

```
#Cargando imagen

pixmap = QtGui.QPixmap('hola.png')

#Accediendo a la Base de Datos

import sqlite3 as dbapi

db = dbapi.connect(".hola.dat")
```

Pero esto nos puede ocasionar un problema cuando la aplicación este instalada, por lo que una opción **recomendable seria darle a las rutas de los archivos una ruta absoluta,** como por ejemplo:

```
pixmap = QtGui.QPixmap('/usr/share/hola/hola.png')
```

Que es donde se copiarían en el caso de *hola* todos los archivos del programa según especificamos al configurar el archivo **DIRS** y **RULES** en la carpeta **DEBIAN** en el punto anterior. Otro detalle a tener en cuenta, como es en el caso de la Base de Datos que se crea en *hola* para ser usada con SQLite, es que todo lo que antes se creaba utilizando una ruta relativa en el directorio actual, ahora pasaría a crearse a partir del "*home*".

- **Paso 5)** Creando el archivo ".desktop"

El archivo ".desktop" cumple la función de ser una especie de acceso directo. Se trata de un archivo de texto que contiene información de nuestro programa, define un icono y permite que el usuario lance el programa haciendo doble clic sobre el. El archivo ***"hola.desktop"*** (el cual debe estar dentro de la carpeta *"hola-1.2*" en el caso de este ejemplo) contiene lo siguiente:

```
[Desktop Entry]
Encoding=UTF-8
Name=Hola
Icon=/usr/share/hola/hola.png
Comment=Hola es mi programa
Exec=/usr/bin/hola
Terminal=false
Type=Application
Categories=GNOME;Utility;
StartupNotify=False
```

Como podemos ver es básicamente información descriptiva acerca de la aplicación, donde lo mas relevante serían las rutas de la imagen y ejecutable del programa.

- **Paso 6)** Dentro de la carpeta *"hola-1.2*" en la que seguimos trabajando, creamos el archivo ***hola*** el cual contiene lo siguiente:

```
#!/bin/sh

python /usr/share/hola/hola.py
```

- **Paso 7)** Momento de crear el Paquete DEB

Lo único que debemos hacer es situarnos en la carpeta "**hola-1.2**"

```
$ cd /home/deb/hola-1.2
```

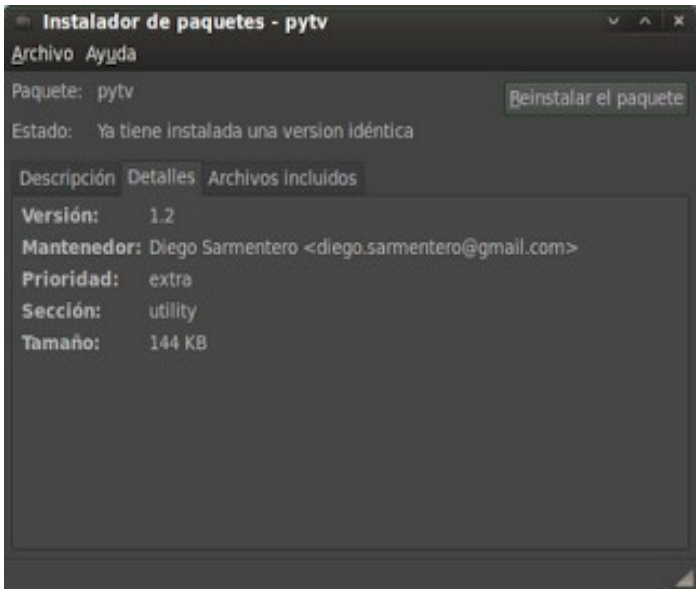
y luego ejecutar el comando:

```
$ sudo dpkg-buildpackage
```

Y ya deberíamos tener creado nuestro archivo "**.deb**" en la carpeta "*deb*". En esta carpeta van a aparecer otros archivos mas junto con el "**.deb**" pero no son de importancia, así que pueden ser eliminados sin problema.
Y ahora podemos proceder a instalar la aplicación, bien con el comando

```
$ dpkg -i hola-1.2.deb
```

O bien con herramientas gráficas




***Advertencia:*** Al generar el Paquete DEB lo crea por la configuración del archivo *CONTROL* para la arquitectura especifica de la maquina en la que se esta trabajando. En "amd64", no tocar nada, pero para hacer un instalador "i386" en una maquina virtual con otra arquitectura y correr el comando "**sudo** dpkg-buildpackage" dentro de la maquina virtual, y de paso probar en esa arquitectura si el instalador funcionaba correctamente. Pero no debería dar problemas.

## ANEXO I – FICHERO RULES

```
#!/usr/bin/make -f
# -*- makefile -*-
# Sample debian/rules that uses debhelper.
# This file was originally written by Joey Hess and Craig Small.
# As a special exception, when this file is copied by dh-make into a
# dh-make output file, you may use that output file without restriction.
# This special exception was added by Craig Small in version 0.37 of dh-make.
# Uncomment this to turn on verbose mode.
#export DH_VERBOSE=1

configure: configure-stamp
configure-stamp:
	dh_testdir
	# Add here commands to configure the package.
	touch configure-stamp


build: build-stamp

build-stamp: configure-stamp
	dh_testdir
	touch build-stamp

clean:
	dh_testdir
	dh_testroot
	rm -f build-stamp configure-stamp
	dh_clean

install: build
	dh_testdir
	dh_testroot
	dh_clean -k
	dh_installdirs
	mkdir -p ${CURDIR}/debian/hola
	mkdir -p ${CURDIR}/debian/hola/usr/share/hola
	cp hola.py ${CURDIR}/debian/hola/usr/bin/
	chmod a+x ${CURDIR}/debian/hola/usr/bin/hola.py
	cp *.py ${CURDIR}/debian/hola/usr/share/hola/
#	cp *.png ${CURDIR}/debian/hola/usr/share/hola/
	cp hola.desktop ${CURDIR}/debian/hola/usr/share/applications/

# Build architecture-independent files here.
binary-indep: build install


# We have nothing to do by default.


# Build architecture-dependent files here.
```

```
binary-arch: build install
	dh_testdir
	dh_testroot
	dh_installchangelogs
	dh_installdocs
	dh_installexamples
	dh_install
	dh_installmenu
#	dh_installdebconf
#	dh_installlogrotate
#	dh_installemacsen
#	dh_installpam
#	dh_installmime
#	dh_python
#	dh_installinit
#	dh_installcron
#	dh_installinfo
	dh_installman
	dh_link
	dh_strip
	dh_compress
	dh_fixperms
#	dh_perl
#	dh_makeshlibs
	dh_installdeb
	dh_shlibdeps
	dh_gencontrol
	dh_md5sums
	dh_builddeb

binary: binary-indep binary-arch
.PHONY: build clean binary-indep binary-arch binary install configure
```