

Distribuyendo tus aplicaciones Python en PyPI

Si desarrollaste un módulo o aplicación Python, distribuirlo en el *Python Package Index* (PyPI) y dejarlo disponible a solo un “pip install” de cualquier usuario, es una tarea sumamente sencilla que en unos pocos minutos puedes aprender. Entérate cómo distribuir las leyendo este artículo.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)



Eugenia es **Arquitecta de Software**, **docente** instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y eXtreme Programming. Miembro de la **Free Software Foundation** e integrante del equipo de **Debian Hackers**.

Webs:

Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

El *Python Package Index*, más conocido por sus siglas **PyPI**, es el repositorio oficial de aplicaciones desarrolladas en Python, que permite a programadores de todo el mundo distribuir sus creaciones de forma eficiente y muy simple.

Las aplicaciones disponibles en <http://pypi.python.org>, pueden ser instaladas por los usuarios, mediante un simple `pip install nombre-del-paquete` y para que puedas distribuir tus aplicaciones a través de PyPI, solo es necesario prepararlas siguiendo unos pocos pasos como veremos a continuación.

Preparar tu aplicación

Lo más importante es contar con una carpeta para alojar los módulos y paquetes de la aplicación. Esta carpeta, contará con algunos archivos estándar para su distribución. Una estructura de archivos y directorios apropiada, podría ser como la siguiente:

MiAplicacion/	Directorio de la aplicación
_ paquete_1/	Paquete (parte de la app)
_ _ __init__.py	
_ _ ...	
_ paquete_2/	
_ _ __init__.py	
_ _ ...	
_ docs/	Documentación
_ _ ...	
_ tests/	Test Unitarios
_ _ ...	
_ README.txt	Descripción de la aplicación
_ LICENSE.txt	Copia de la licencia http://www.gnu.org/licenses/gpl.txt
_ MANIFEST.in	Lista de archivos que formarán parte del paquete
_ setup.py	Archivo de instalación

El archivo README.txt

En principio, aclarar que la extensión .txt se utiliza en los paquetes para otorgar compatibilidad con sistemas operativos privativos no basados en *nix, pero en sistemas GNU/Linux, dicha extensión, carece de significado. Sin embargo, es un estándar a la hora de distribuir paquetes Python a través de PyPI.

En este archivo se puede colocar cualquier información relativa a la aplicación, preguntas frecuentes, modos de uso e implementación, etc. La misma será aquella que PyPI, muestre en la página del paquete:

pyprintr 0.3

Module that allows to emulate the `print_r()` PHP function

Package: admin | releases | view | edit | files | PKG-INFO

Download
pyprintr-0.3.tar.gz

Welcome
login.launchpad.net_90

Your details (Logout)

Your packages:

pyprintr

Python-printr

Python-printr for own objects of a class instance Python-printr is a module that allows to emulate the `print_r()` function of PHP by printing the objects properties of a class instance and its internal structure. If you need to use `print_r()` function for list, tuples and dicts, please, don't use `print_r()` function. You can choose `pprint` module. `print_r()` function is only for objects of your own class instances.

Developed by Eugenia Bahit. Distributed under a GPL v3.0 licence. <http://www.python-printr.org/> Texto del archivo README

Este archivo, así como la documentación que se incluya en docs/*.txt puede escribirse aplicando el formato *reST*.

Una guía rápida de *reST* puede obtenerse en:

<http://docutils.sourceforge.net/docs/user/rst/quickstart.html>

El archivo LICENSE.txt

Básicamente se trata de una copia del texto de la licencia con la que se distribuye el Software o paquete. Para distribuir el paquete como Software Libre, recomiendo utilizar la licencia GPL en su versión 3.0 cuyo texto completo puede obtenerse en <http://www.gnu.org/licenses/gpl.txt>

Recordad que la licencia **GPL v 3.0**, es la recomendada por la **Free Software Foundation**¹ para la distribución de **Software Libre**, sea éste de uso comercial o no.

El archivo MANIFEST.in

Éste, es el que almacena el listado de archivos que componen el paquete. El listado de archivos puede colocarse “a mano” (es decir, archivo por archivo) o se pueden utilizar instrucciones como `include expresión`.

Para un paquete con la siguiente estructura:

```
PythonPrintr/  
|_ printr/  
|_ |__init__.py    -- módulo que contiene la función printr()  
|_ docs/  
|_ tests/  
|_ README.txt  
|_ LICENSE.txt  
|_ MANIFEST.in  
|_ setup.py
```

El archivo MANIFEST.in podría verse como el siguiente:

```
include docs/*.txt  
include *.txt  
include printr/*.py
```

Algunas instrucciones frecuentes que podrían indicarse en un archivo MANIFEST.in, son:

```
include  
Archivos a incluir en el paquete  
  
exclude  
Archivos a ignorar en el paquete  
  
recursive-include  
Incluye archivos de forma recursiva  
  
recursive-exclude  
Ignora archivos de forma recursiva
```

Otras instrucciones como `global-include`, `global-exclude`, `graft` y `prune` también se encuentran disponibles.

1 <http://www.fsf.org/>

El archivo setup.py

Este es el archivo clave. Contendrá toda la información sobre el paquete que deseas distribuir.

Un ejemplo de formato de archivo setup.py se verá como el siguiente:

```
from distutils.core import setup

setup(
    name="PythonPrintr",           # Nombre de la app (CamelCase)
    version="0.3",                # Versión
    author="Eugenia Bahit",       # Nombre del Autor
    author_email="ebahit@member.fsf.org", # E-mail del autor
    packages=["printr"],          # Paquetes a incluir
    url="http://www.python-printr.org", # URL de la app
    license="GPL v3.0",           # Licencia

    # Descripción corta
    description="Module that allows to emulate the print_r() PHP function",

    long_description=open('README.txt').read(), # Descripción larga

    # Selección de categorías que pueden encontrarse en:
    # https://pypi.python.org/pypi?action=list_classifiers
    classifiers=[
        "Development Status :: 5 - Production/Stable",
        "Environment :: Console",
        "Intended Audience :: Developers",
        "License :: OSI Approved :: GNU General Public License v3 (GPLv3)",
        "Natural Language :: English",
        "Operating System :: OS Independent",
        "Programming Language :: Python :: 2.7",
        "Programming Language :: Python :: 2 :: Only",
        "Topic :: Software Development",
        "Topic :: Software Development :: Libraries :: Python Modules"
    ]
)
```

Cada uno de los parámetros que se pasan a la función setup(), son opcionales. Sin embargo, los tres primeros (name, version y author) siempre deberían estar presentes.

Distribuyendo la aplicación

Ya hemos preparado todos los archivos. Es hora de distribuir la aplicación. Para ello, lo primero que haremos es registrar nuestro proyecto en el PyPI. Para ello, desde la terminal, navegamos hasta el directorio de la aplicación y ejecutamos:

```
python setup.py register
```

La primera vez que registremos nuestro proyecto, antes de completar la acción nos

preguntará si deseamos crear una cuenta o utilizar una existente, mostrando el siguiente menú:

```
running register
We need to know who you are, so please choose either:
  1. use your existing login,
  2. register as a new user,
  3. have the server generate a new password for you (and email it to you), or
  4. quit
Your selection [default 1]:
```

Si no dispones de una cuenta en PyPI, la opción 2 será la más apropiada. Eligiendo dicha opción, te pedirá ingresar un e-mail, elegir y confirmar una contraseña alfanumérica que contenga al menos una letra mayúscula.

Una vez aportes dichos datos, te preguntará si deseas guardar los mismos para que no te los pida nuevamente en el futuro.

ADVERTENCIA: La contraseña será almacenada en texto plano. Utilizar esta función con suma precaución.

En caso de aceptar esta opción, tus datos serán guardados en el archivo `~/.pypirc` con el siguiente formato:

```
[distutils]
index-servers =
    pypi

[pypi]
username:tu_usuario
password:tu_clave
```

Este archivo, es posible crearlo de forma manual.

Finalizado el registro de la cuenta, `distutil` continuará registrando el proyecto:

```
running register
running check
Registering PythonPrintr to http://pypi.python.org/pypi
Server response (200): OK
```

Una vez finalizado, crearemos el *tarball* de distribución y lo subiremos a PyPI en un solo paso:

```
python setup.py sdist --format=gztar upload
```

Al finalizar, un directorio `dist` será creado en la carpeta de la aplicación incluyendo el `.tar.gz`:

```
eugenia@cococha-gnucita:~/webprojects/PythonPrintr$ tree
```

```
├── dist
│   └── pyprintr-0.3.tar.gz
├── docs
│   └── howto.txt
├── printr
│   └── __init__.py
├── tests
├── LICENSE.txt
├── MANIFEST          -- archivo creado por el sdist
├── MANIFEST.in
├── README.txt
└── setup.py
```

Y esto, es todo lo que necesitamos para distribuir nuestra aplicación.

Instalar tu aplicación

Para visualizar tu aplicación en el repositorio PyPI, deberás ingresar en: <https://pypi.python.org/pypi/NombreDeTuApp>. Luego, para que otros usuarios puedan instalarla, deberán ejecutar:

```
sudo pip install nombre-de-tu-app
```



¿Quieres codear tu app conmigo?

Si me cuentas la aplicación que tienes en mente la **podemos codear juntos en Python o PHP** y lograr un desarrollo verdaderamente profesional.

Si tienes ganas de **codear tu app como un verdadero experto**, te propongo un curso individual y personalizado de:

Análisis, Organización y Desarrollo de Requerimientos

Más información

Curso individual a distancia. Fecha de inicio a elección del alumno. Sin duración fija. Costo: USD 150 por mes (USD 120 después del tercer mes de curso). Más información en <http://cursos.eugeniabahit.com/analisis>