Cómo Crear un Paquete Redistribuible en Python

Crear un paquete redistribuible en Python implica estructurar tu código, escribir los archivos de configuración necesarios y preparar el paquete para su distribución. Este documento describe los pasos para crear un paquete de Python utilizando setuptools, una herramienta popular para empaquetar proyectos Python.

# 1. Estructura del Proyecto

Comienza organizando tu proyecto de manera clara y estructurada. Una estructura típica para un paquete de Python se ve así:

* mi\_paquete/
* ├── mi\_paquete/
* │ ├── \_\_init\_\_.py
* │ ├── modulo1.py
* │ ├── modulo2.py
* ├── setup.py
* ├── README.md
* ├── LICENSE
* ├── requirements.txt

# 2. Creando setup.py

El archivo `setup.py` es esencial para definir cómo debe construirse tu paquete, qué dependencias tiene y qué metadatos deben incluirse. Aquí tienes un ejemplo de cómo podría verse `setup.py`:

from setuptools import setup, find\_packages  
  
setup(  
 name='mi\_paquete',  
 version='0.1',  
 packages=find\_packages(),  
 install\_requires=[  
   
 ],  
 description='Una descripción breve de tu paquete',  
 long\_description=open('README.md').read(),  
 long\_description\_content\_type='text/markdown',  
 author='Tu Nombre',  
 author\_email='tu.email@ejemplo.com',  
 url='https://github.com/tuusuario/mi\_paquete',  
 classifiers=[  
 ],  
)

# 4. Probando el Paquete

Es fundamental probar tu paquete antes de distribuirlo. Puedes crear archivos de prueba en el directorio `pruebas/`. Por ejemplo, `prueba\_modulo1.py` podría incluir pruebas unitarias para los módulos de tu paquete.

# 5. Construyendo y Distribuyendo el Paquete

Una vez que tu paquete esté listo y probado, puedes usar setuptools para construir y distribuir tu paquete. Usa los siguientes comandos para crear una distribución de origen y un archivo wheel:

Hacel ls o dir segun que consola usen y si esta setup.py

python setup.py sdist bdist\_wheel

Luego puedes subir tu paquete a PyPI usando Twine, de la siguiente manera:

twine upload dist/\*