

# MEMORIA 2

Una vez convertido nuestro código del puzzle 1 en una clase Rfid con sus respectivas funciones ; innit y readuid. Ahora debemos convertirlo en librería para poder utilizar estas mismas funciones importadas en el código del puzzle 1.

Primero hice un directorio llamado “librería” donde guardé el archivo **setup.py** el cual sirve como archivo de configuración para la instalación de la librería. Para ello me descargue la librería setuptools mediante pip.

```
home > pi > librerias > setup.py
1  from setuptools import setup, find_packages
2
3  setup(
4      name="puz1",
5      version='0.1',
6      packages= find_packages(),
7      install_requires=['adafruit-pn532'],
8  )
```

Después cree otra carpeta dentro de “libreria” llamada “libreria2” donde guardé el código del puzzle1 “puz1.py” y el archivo **\_\_innit\_\_.py** que tiene como función hacer que la carpeta sea un paquete de python.

```
1  |
2  from puz1 import __init__, read_uid
```

Para acabar, dentro del directorio “libreria” ejecutamos el comando “pip install .”. De esta forma hemos instalado nuestra librería en nuestro entorno de Python y podemos importar sus funciones a cualquier script.

```
from puz1 import Rfid
```

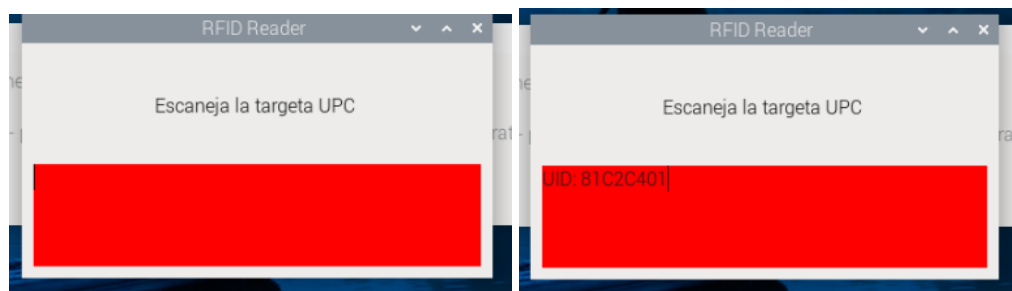
Para el puzzle 2 se nos indica utilizar Pygobject para las bibliotecas GTK y GObject, tuve bastantes problemas con estas bibliotecas así que probé instalando usando pip con: “pip install pygobject” dentro del entorno y usando el gestor de paquetes apt con: “sudo apt install python3-gi”. Para aprender las funciones de estas bibliotecas investigue en la página web de pygobject :<https://pygobject.gnome.org/tutorials/gtk4/display-widgets/label.html>.

A partir de esto desarrollé un código básico para el lector del puzzle 1 con una interfaz gráfica presentando labels en la pantalla. Sin embargo, tuve bastantes problemas con la biblioteca GTK, parte del problema se solucionó escribiendo en el código la versión requerida de GTK : `gi.require_version('Gtk', '3.0')`

También me daba otros problemas donde tenía que generar manualmente el GI\_TYPELIB\_PATH al lugar donde se encontraba la librería del girepository-1.0 porque de otra forma no la detectaba. Utilicé el siguiente método de exportación después de encontrar la librería y su path : `export GI_TYPELIB_PATH=/usr/lib/aarch64-linux-gnu/girepository-1.0/`

Así que finalmente opté por utilizar una alternativa a Pygobject que utilizaba GTK que me resultaba problemático. Buscando encontré la alternativa tkinter, lo descargué mediante pip e investigué sus funciones. Para esta alternativa elaboré un código más completo con uso de funciones y obtuve un resultado que me complacía más.

Ambos códigos los probé mediante el REAL VNC viewer y funcionaron correctamente. A partir de ahí, modifiqué la interfaz a mi gusto comparando los resultados.  
Con pygobject: (menos trabajado)



Con tkinter (más trabajado, funciones, colores, botón...)

