



GENERADOR DE QR EN ANDROID

RAFAEL DELGADO GARCÍA-VALDECASAS

PABLO RIENDA SÁNCHEZ

GRADO EN INGENIERÍA INFORMÁTICA

Periféricos y dispositivos de interfaz humana

Curso 2023-2024

Índice

1. Introducción
2. Instalación y pasos previos
3. Código para el programa de generación de QR
4. Resultados y visualización
5. Bibliografía

1. Introducción

En este trabajo nuestro objetivo será crear un generador de QR que se pueda ejecutar desde Android y que sea fácilmente trasladable a otros dispositivos y sistemas operativos.

Para ello, utilizando el lenguaje de programación Python crearemos un código que aproveche las librerías ya existentes y que nos permita generar QR. Así tendremos un generador de QR universal para el que solo hará falta un intérprete de Python en nuestro sistema operativo.

2. Instalación y pasos previos

Para conseguir trasladar nuestra aplicación Python a Android buscamos en Android un intérprete de Python. Para ello y tras probar distintas aplicaciones encontradas en Google Play decidimos que la que mejor se adaptaba a nuestros requisitos era la aplicación gratuita Pydroid 3.

Una vez decidido que intérprete de Python vamos a utilizar podemos pasar a instalar las librerías y dependencias necesarias para la implementación de la funcionalidad requerida.

Instalaremos las dependencias OpenCV y qrcode con el comando:

```
pip3 install Image opencv-python qrcode
```

3. Código para el programa de generación de QR

Crearemos nuestro programa qrandroid.py en python.

Lo primero que tendremos que hacer es importar la librería a utilizar, en este caso qr code.

```
import qrcode
```

A continuación, pasamos a definir la función que se encarga de generar los qr.

La función necesitará dos parámetros de entrada, que corresponderán al enlace o texto que aparecerá al escanear el qr y al nombre de la imagen con el que queremos que se guarde la imagen qr.

```
def crear_qr_link (link: str, name: str = 'imagen'):  
    img=qrcode.make(link)  
    img.save(name + '.png')
```

Como vemos, hemos utilizado la función de qr code make, que se encarga de crear una imagen qr que muestre lo que se le pase por parámetro, en este caso un enlace a una web o un texto. También utilizamos la función save de img para almacenar en el dispositivo la imagen qr generada.

Finalmente con el código siguiente hacemos funcionar la función pidiendo al usuario el link o texto y el nombre de la imagen que desea almacenar.

```
cadena = input("Ingresa la cadena o link ")
```

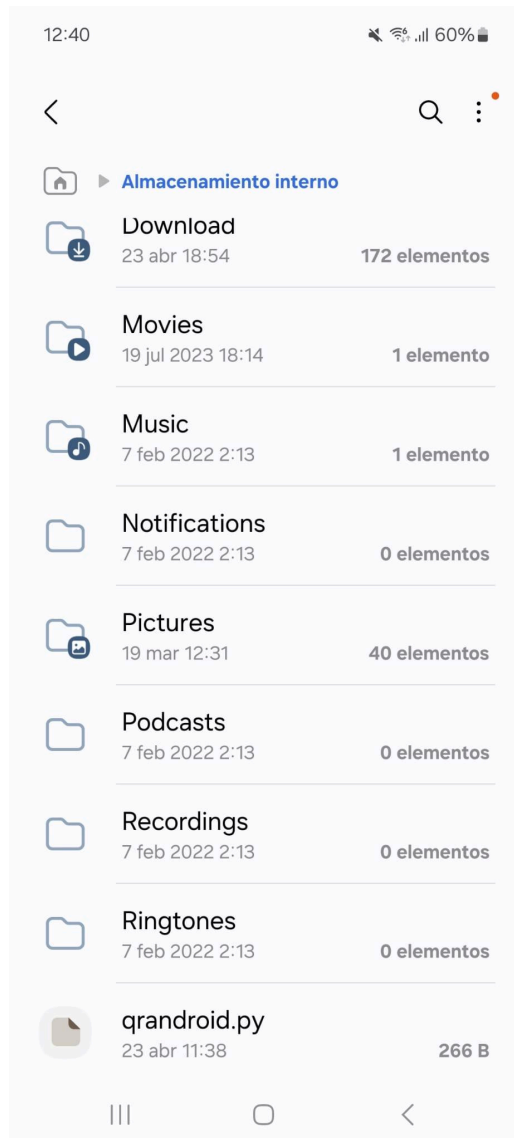
```
nombre_qr = input("Ingresa el nombre del archivo imagen ")
```

```
crear_qr_link(cadena, nombre_qr)
```

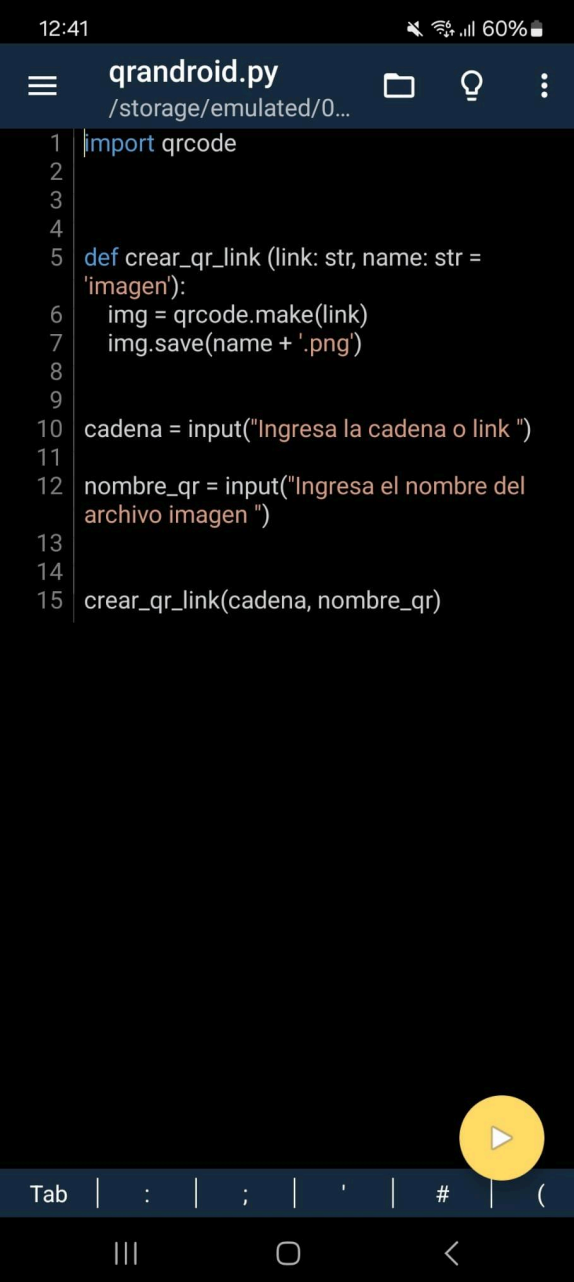
4. Resultados y visualización

En este apartado mostraremos cómo se ejecuta el programa en el intérprete elegido, así como los resultados que da, dónde se almacenan las imágenes generadas y demostraremos su correcto funcionamiento.

Lo primero que debemos conocer es donde se encuentra almacenado nuestro archivo .py. En concreto, vemos que la localización dentro de la estructura de archivos del teléfono Andorid utilizado de ejemplo es la siguiente:



Una vez dentro de la aplicación Pydroid 3 abriremos el archivo .py y pulsaremos la flechita amarilla para ejecutar el programa:

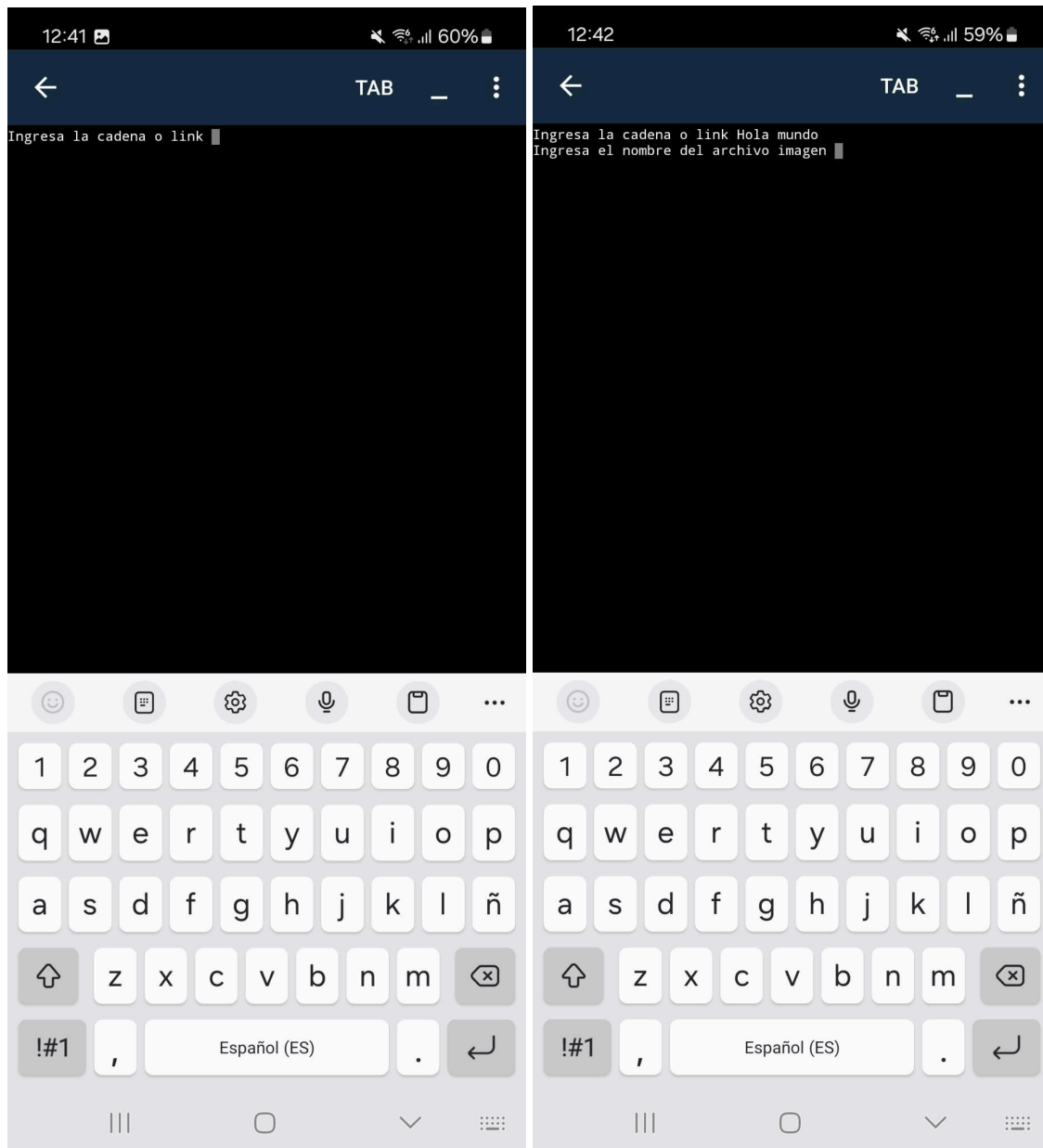


The screenshot shows the Pydroid 3 application interface. At the top, the status bar displays the time 12:41, signal strength, Wi-Fi, and 60% battery. The app's title bar is dark blue with the text 'pydroid.py' and the path '/storage/emulated/0...'. Below the title bar, the Python code is displayed on a black background with white and yellow syntax highlighting. The code is as follows:

```
1 import qrcode
2
3
4
5 def crear_qr_link (link: str, name: str =
  'imagen'):
6     img = qrcode.make(link)
7     img.save(name + '.png')
8
9
10 cadena = input("Ingresa la cadena o link ")
11
12 nombre_qr = input("Ingresa el nombre del
  archivo imagen ")
13
14
15 crear_qr_link(cadena, nombre_qr)
```

At the bottom right of the code editor, there is a yellow circular button with a white play icon, used to execute the program. Below the code editor, there is a dark blue bar with keyboard shortcuts: 'Tab', ':', ';', '|', '#', and '('. At the very bottom, there is a black bar with three white icons: a vertical ellipsis, a circle, and a left arrow.

Nos pedirá ingresar la cadena o link. Para este primer ejemplo le diremos que imprima la cadena “Hola mundo”. El nombre de la imagen será “hola”. Al enviarlo el intérprete nos avisará de que la ejecución del programa ha acabado:



12:43

5G 59%

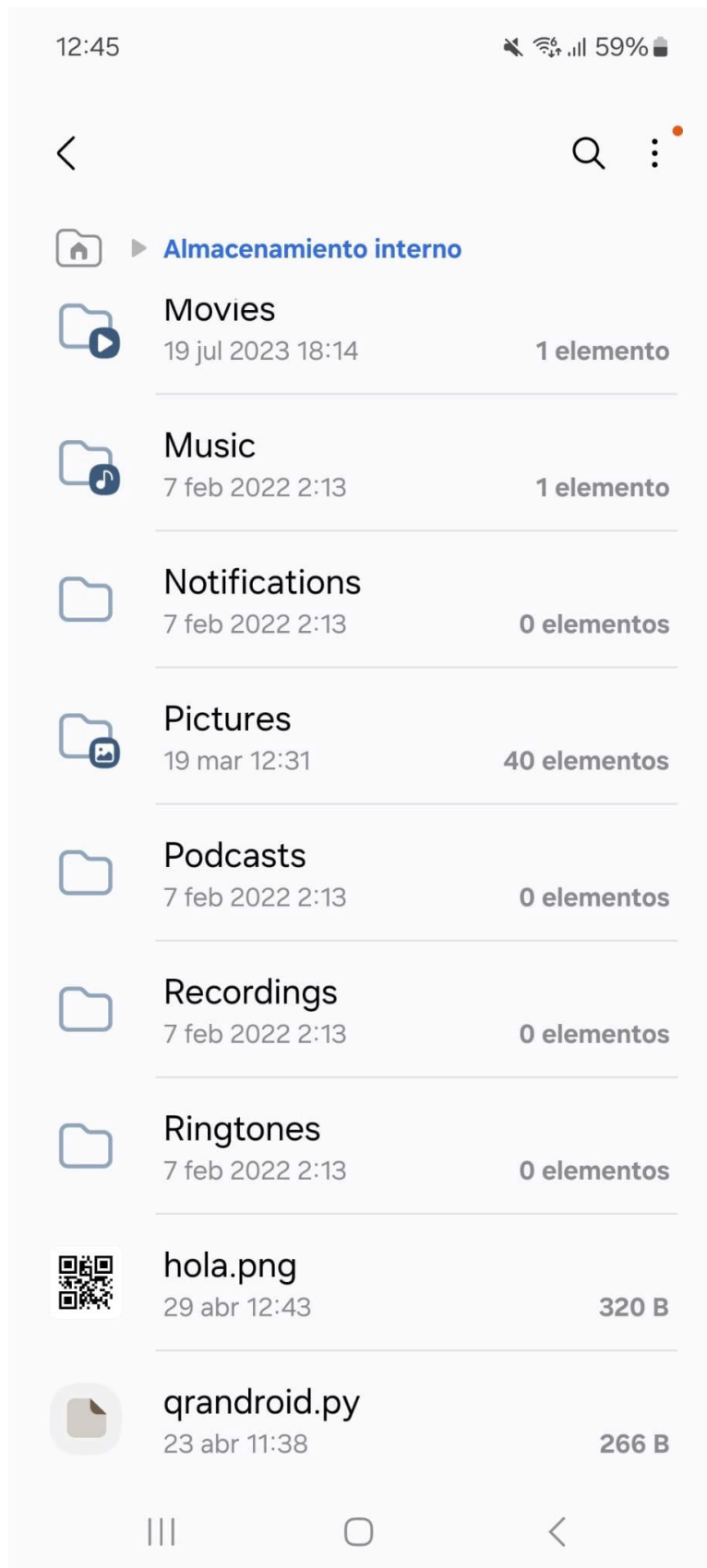


TAB



```
Ingresa la cadena o link Hola mundo
Ingresa el nombre del archivo imagen hola
[Program finished]
```


Para ver los resultados, simplemente debemos dirigirnos al directorio donde se encontraba nuestro `grandroid.py`. Ahí se nos debe haber guardado el archivo `hola.png` correspondiente al código qr generado por nuestro programa:

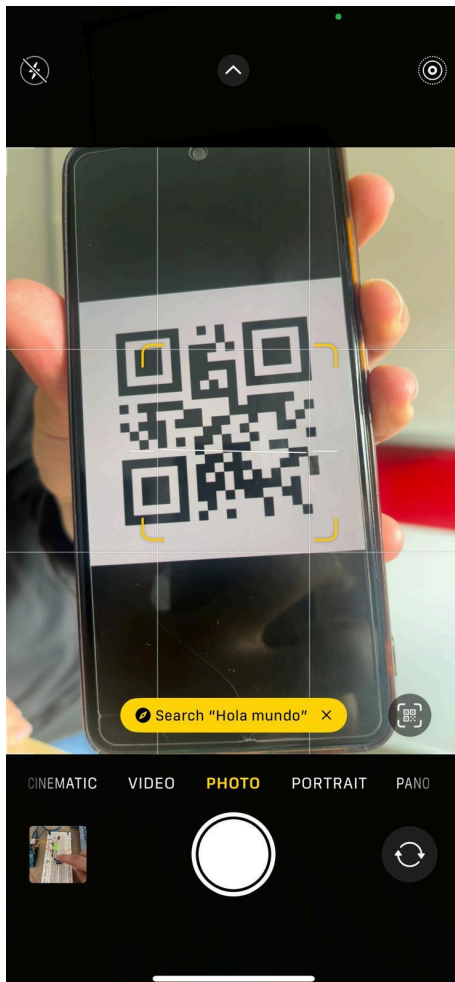


Al abrir la imagen veremos el qr generado:



Al escanear este qr podemos ver que funciona perfectamente e incluso podemos leer el qr desde un teléfono móvil diferente, lo cual es ideal.

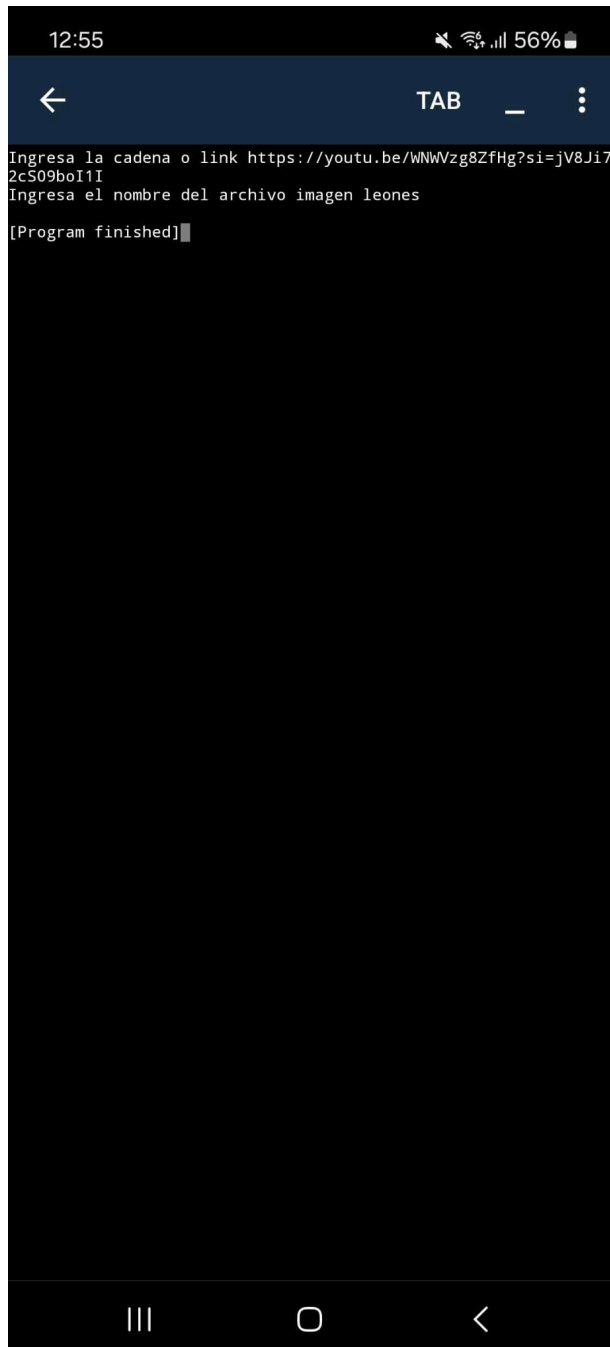
Este es el resultado de la lectura de la imagen qr:



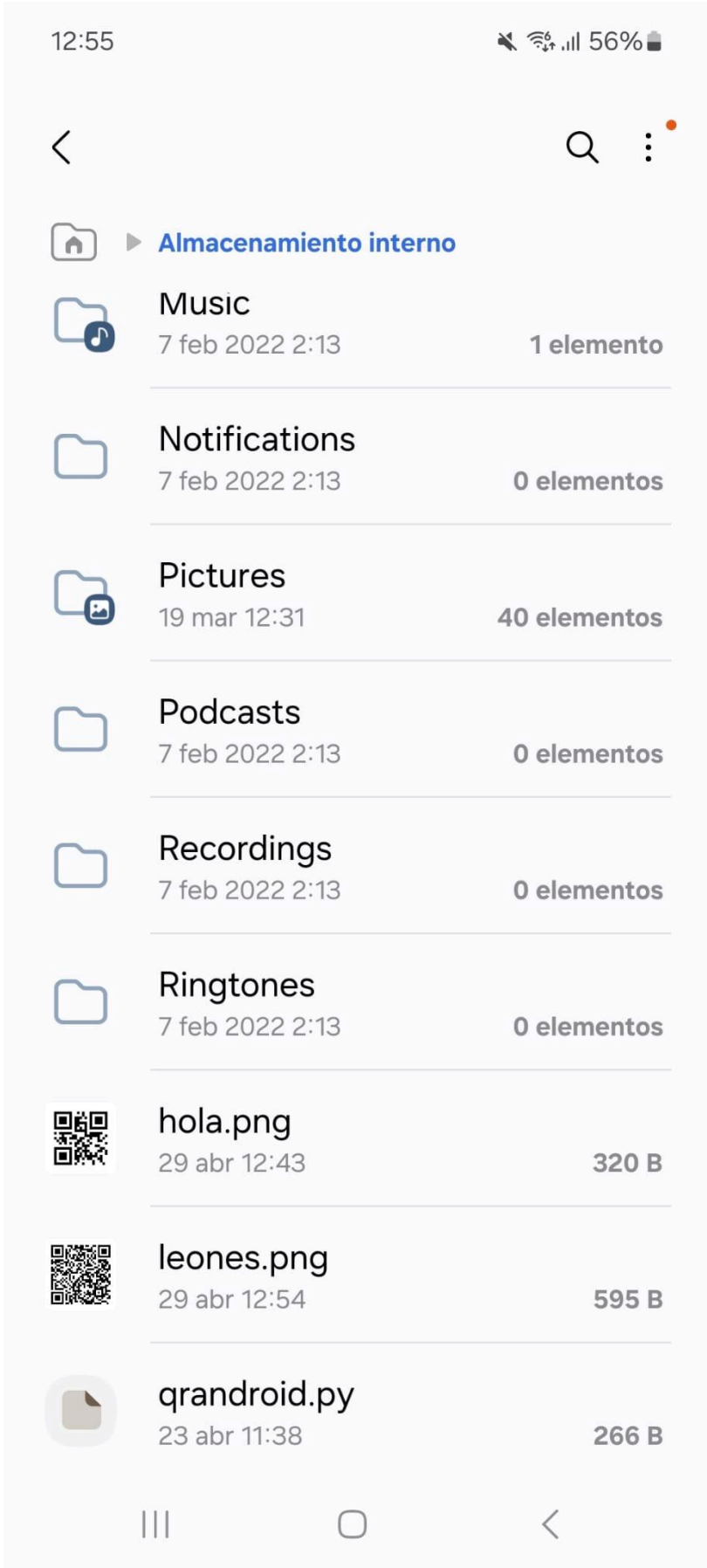
Ahora probaremos con un enlace. Para ello obtendremos el enlace de este vídeo de Youtube correspondiente a un vídeo sobre leones de National Geographic:

<https://youtu.be/WNWVzg8ZfHg?si=jV8Ji72cS09boI1I>

Introduciremos este enlace cuando se nos pida y llamaremos a la imagen leones:



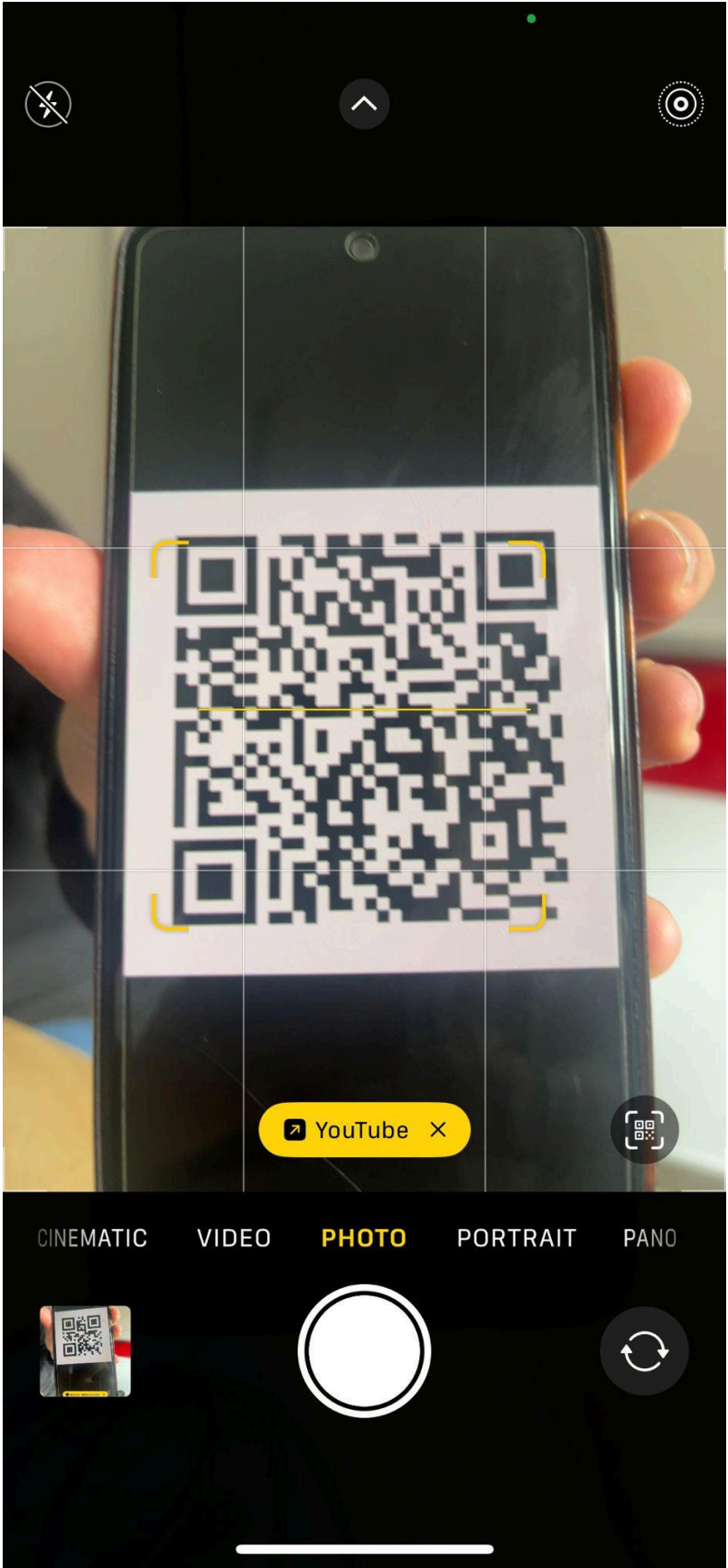
La imagen generada estará junto a la anterior:



El qr resultante es el siguiente:



Al escanearlo desde otro teléfono móvil aparecerá lo siguiente:



Y si seleccionamos el enlace se nos abrirá el vídeo en Youtube:

12:58

5G

◀ Camera

Batalla de REYES. Lucha entre LEONES | National Geographic en Español

6.5M views · 6 yr ago · ...more

National Geographic Esp... 674K [Subscribe](#)

32K | | [Share](#) | [Download](#) | [Clip](#)

Comments 1.1K

Mi hipopotamo le gana a 10 leones

✦ Paris ✦

5. Bibliografía

Para la realización del trabajo nos hemos ayudado principalmente del material aportado por el profesor, además, hemos consultado las distintas páginas de documentación oficial que hemos necesitado como:

<https://pypi.org/project/qrcode/>