



## UNIVERSIDAD NACIONAL DE COLOMBIA TPI (TALLER DE PROYECTOS INTERDISCIPLINARIOS)

MANUAL DE USO: LECTURA DE CÓDIGOS QR

*Fresneda H. Yulian*

*Gómez B. Johan*

*Antolinez Z. Stewart*

*Figueroa F. Juan*

*Mateus R. Dimar*

*Dueñas. S. Natalia*

---

## 1. INSTALACIÓN Y CONFIGURACIÓN INICIAL

---

### 1.1 Instalar Visual Studio Code

1. Descargar desde: <https://code.visualstudio.com/>
2. Ejecutar el instalador y seguir las instrucciones.
3. Al abrir VS Code por primera vez, instalar la extensión de Python.

### 1.2 Instalar Python

1. Descargar desde: <https://www.python.org/downloads/>
2. Seleccionar la opción "Add Python to PATH" al instalar.

### 1.3 Crear entorno de trabajo

- Crear una carpeta en la ruta deseada (por ejemplo: C:\Users\NombreUsuario\ProyectoQR).
- Abrir VS Code y seleccionar "Abrir carpeta" para trabajar en ese directorio.

### 1.4 Instalar librerías necesarias

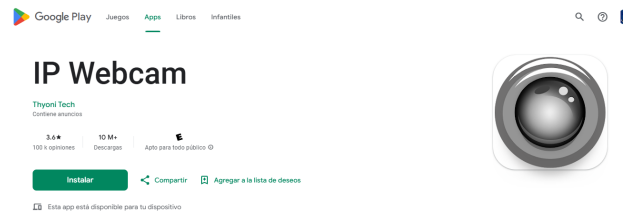
Desde el terminal integrado de VS Code ("Ver → Terminal"), ejecutar:

```
pip install opencv-python pyzbar imutils
```

## 2. CASO 1: CAPTURA DESDE CELULAR (CAM IP)

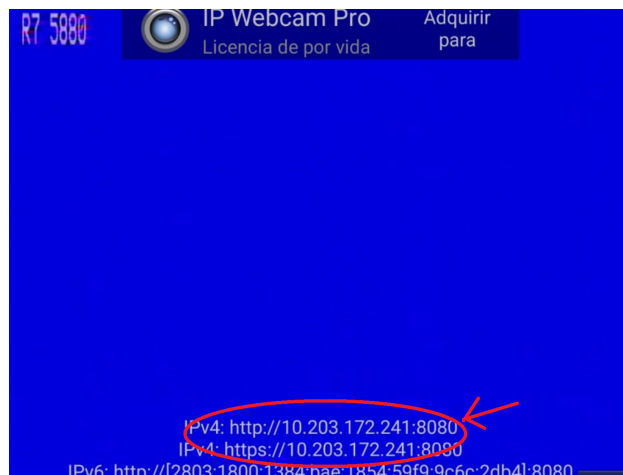
### 2.1 Preparación en el celular

1. Instalar la app **IP Webcam** desde la Play Store.



2. Conectar el celular y el PC a la misma red Wi-Fi.
3. Abrir la app y configurar resolución y calidad deseadas.
4. Iniciar el servidor en el celular. Se mostrará una URL como:

`http://192.168.0.101:8080/video`



### Modificación del código para lectura desde cámara IP (celular)

Una vez que la cámara del celular esté transmitiendo por medio de una aplicación de IP Webcam o similar , se debe identificar la dirección de transmisión del video. Esta dirección suele tener el siguiente formato:

`http://<IP_DEL_CELULAR>:8080/video`

Para utilizar esta transmisión en el código Python, se debe reemplazar la fuente del video. Localice la línea en el código que inicia el acceso a la cámara, similar a:

```
url='https://172.20.10.6:8080/video'
```

## Manual de Uso: Lectura de Códigos QR

Y cámbiela por:

```
url = ("http://<IP_DEL_CELULAR>:8080/video")
```

```
import cv2
import numpy as np
import os
from pyzbar.pyzbar import decode

print(cv2.__version__)
#URL de la camara que estamos viendo
url='https://172.20.10.6:8080/video'

Camara=cv2.VideoCapture(url)
Camara.set(3,640)
Camara.set(4,480)
Color=(0,0,255)

if not Camara.isOpened():
    print("Me perdi :(")
    exit()

while(True):
    ret, frame = Camara.read()
    if not ret:
        print("Error leyendo el frame")
        break
    barcodes = decode(frame)
```

**Importante:** Asegúrese de que tanto el celular como el computador estén conectados a la misma red Wi-Fi. Si no se carga el video, intente abrir la URL en un navegador para confirmar que el streaming esté activo.

### 3. CASO 2: CAPTURA DESDE DRON CON CÁMARA ANALÓGICA

#### 3.1 Componentes necesarios

- Dron con cámara analógica (5.8 GHz).
- Receptor de video analógico (ej. el utilizado **Slonwake 5.8GHz**) con salida USB OTG.
- Celular Android compatible con señal OTG.
- Aplicación para visualizar señal analógica (**GoFPV**, **WiFi AVIN**, etc.)
- Aplicación **IP Webcam** para compartir la señal al PC vía IP.

#### 3.2 Flujo de conexión

1. Conectar el receptor analógico al celular mediante OTG.
2. Prender el dron.
3. Visualizar la cámara del dron desde alguna app de visualizacion de camara otg. (ej. GoFPV).
4. Una vez la recepcion otg este correcta pasar a la app tipo **IP Webcam** para re-transmitir la pantalla del celular al PC:
5. Si usas **IP Webcam**, sigue la misma lógica que en el caso anterior.
6. Cambia la cámara en configuración en la app para que transmita el receptor otg.

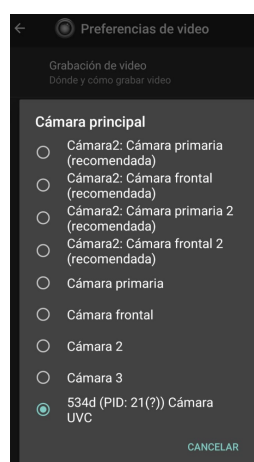


Figura 1: Enter Caption

7. El resto del código Python y su implementación es exactamente igual que en el caso anterior.

## Manual de Uso: Lectura de Códigos QR

---

### Enlace a pruebas realizadas

A continuación, se presentan los enlaces directos a las pruebas realizadas durante el desarrollo del proyecto:

- **Pruebas con dron y con camara:** <https://tpifibog.wixsite.com/g88cc1>

Estos recursos permiten observar el funcionamiento real del sistema en dos escenarios distintos: la detección de códigos desde la señal analógica del dron y la lectura directa desde una cámara conectada al celular. Ambas pruebas fueron fundamentales para validar el desempeño del sistema de detección y almacenamiento automático de códigos.