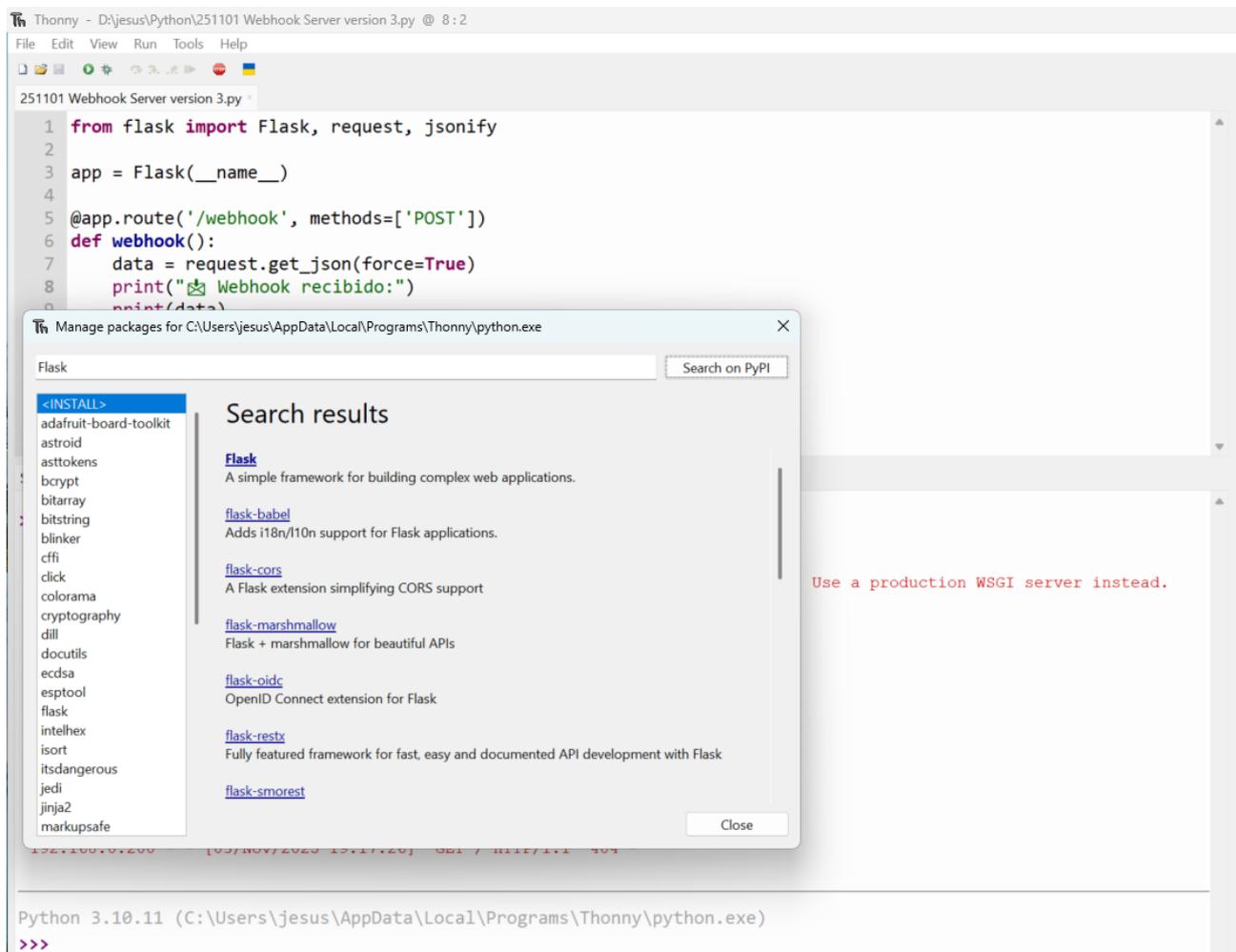


## Como hacer que Shelly se comunice con Python

1. Mediante el Administrador de paquetes de Thonny instalar Flask.

Thonny/Tools/Manage Packages...



Instalar Flask

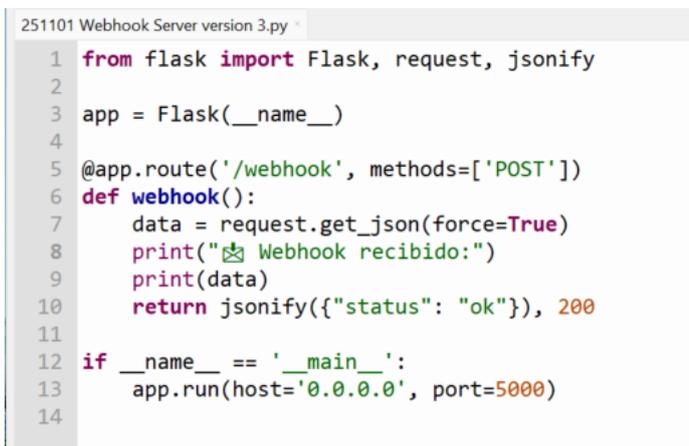
2. Escribir el siguiente programa en Thonny:

```
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/webhook', methods=['POST'])
def webhook():
    data = request.get_json(force=True)
    print("📩 Webhook recibido:")
    print(data)
    return jsonify({"status": "ok"}), 200

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

A screenshot of a code editor window titled "251101 Webhook Server version 3.py". The code is identical to the one shown in the previous block, but with syntax highlighting. Line numbers 1 through 14 are visible on the left side of the editor. The code defines a Flask application with a single POST endpoint at /webhook that returns a JSON response with status "ok".

```
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 @app.route('/webhook', methods=['POST'])
6 def webhook():
7     data = request.get_json(force=True)
8     print("📩 Webhook recibido:")
9     print(data)
10    return jsonify({"status": "ok"}), 200
11
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=5000)
14
```

El programa abre el puerto tcp 5000 en el ordenador y se queda a la espera de recibir mensajes http del tipo POST.

### 3. Ejecutar el programa.

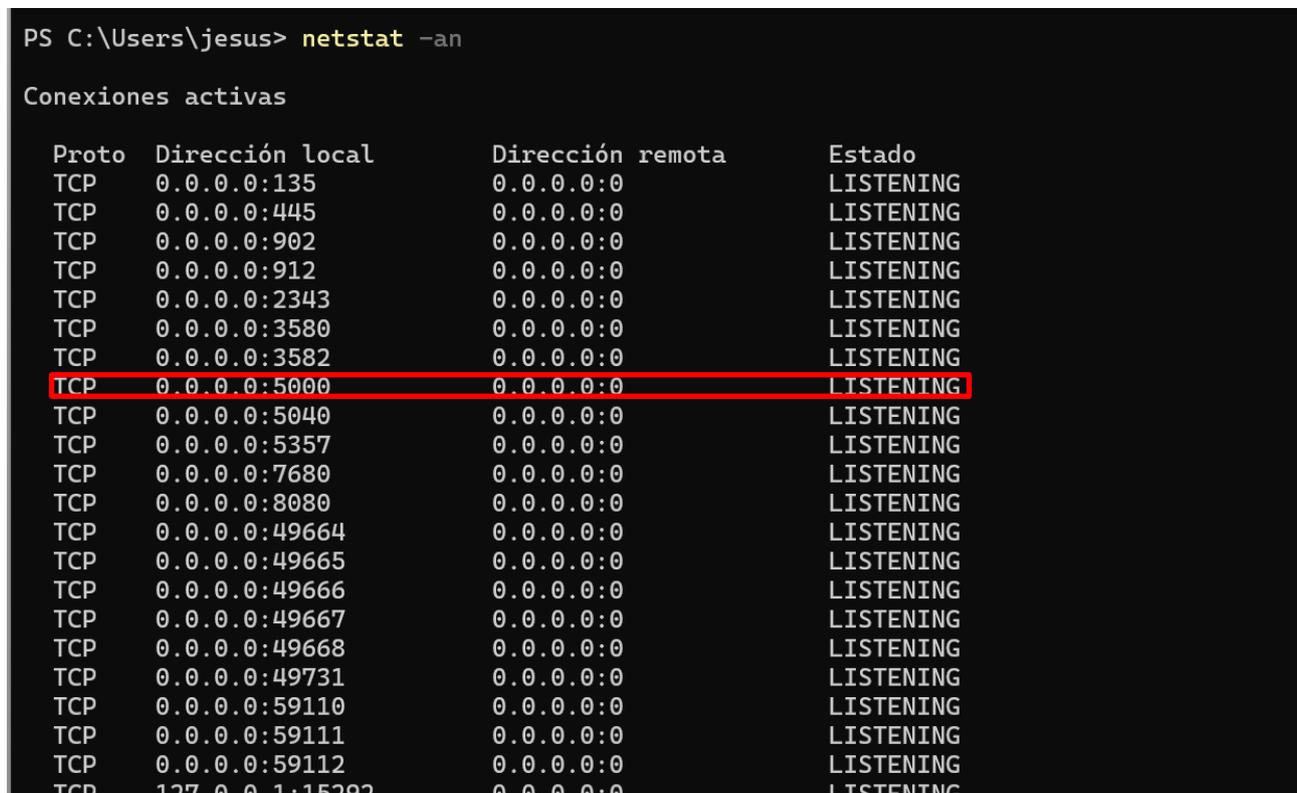


```
File Edit View Run Tools Help
251101 Webhook Server version 3.py
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 @app.route('/webhook', methods=['POST'])
6 def webhook():
7     data = request.get_json(force=True)
8     print("Webhook recibido:")
9     print(data)
10    return jsonify({"status": "ok"}), 200
11
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=5000)
14

Shell
>>> %Run '251101 Webhook Server version 3.py'
* Serving Flask app '251101 Webhook Server version 3'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.30:5000
Press CTRL-C to quit
```

Dirección IP en la que escucha. Tiene que ser la dirección a la que el Shelly manda el http POST.

En cmd con el comando netstat -an se puede ver el puerto tcp 5000 abierto.



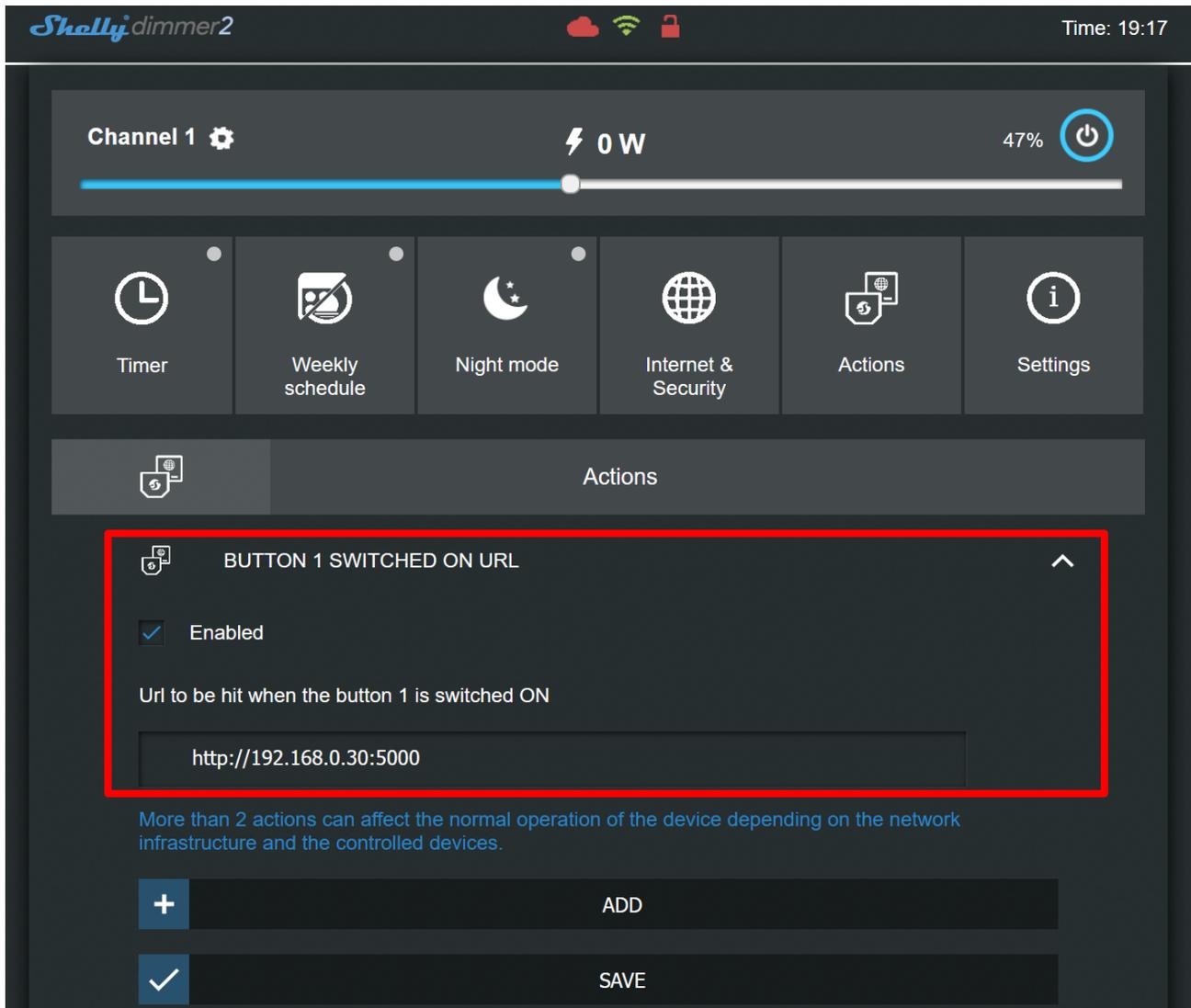
```
PS C:\Users\jesus> netstat -an

Conexiones activas

Proto  Dirección local      Dirección remota      Estado
TCP    0.0.0.0:135           0.0.0.0:0             LISTENING
TCP    0.0.0.0:445           0.0.0.0:0             LISTENING
TCP    0.0.0.0:902           0.0.0.0:0             LISTENING
TCP    0.0.0.0:912           0.0.0.0:0             LISTENING
TCP    0.0.0.0:2343          0.0.0.0:0             LISTENING
TCP    0.0.0.0:3580          0.0.0.0:0             LISTENING
TCP    0.0.0.0:3582          0.0.0.0:0             LISTENING
TCP    0.0.0.0:5000          0.0.0.0:0             LISTENING
TCP    0.0.0.0:5040          0.0.0.0:0             LISTENING
TCP    0.0.0.0:5357          0.0.0.0:0             LISTENING
TCP    0.0.0.0:7680          0.0.0.0:0             LISTENING
TCP    0.0.0.0:8080          0.0.0.0:0             LISTENING
TCP    0.0.0.0:49664         0.0.0.0:0             LISTENING
TCP    0.0.0.0:49665         0.0.0.0:0             LISTENING
TCP    0.0.0.0:49666         0.0.0.0:0             LISTENING
TCP    0.0.0.0:49667         0.0.0.0:0             LISTENING
TCP    0.0.0.0:49668         0.0.0.0:0             LISTENING
TCP    0.0.0.0:49731         0.0.0.0:0             LISTENING
TCP    0.0.0.0:59110         0.0.0.0:0             LISTENING
TCP    0.0.0.0:59111         0.0.0.0:0             LISTENING
TCP    0.0.0.0:59112         0.0.0.0:0             LISTENING
TCP    127.0.0.1:15292      0.0.0.0:0             LISTENING
```

#### 4. Programar el Shelly Dimmer 2.

Programar el Shelly Dimmer 2 para que cuando se active el pulsador ON, se mande un http tipo GET a la dirección del "servidor" HTTP.



!!!Atención!!!

Para la prueba, poner el Shelly Dimmer 2 y el "servidor" http de Shelly en la misma red.

5.

```
Thonny - D:\jesus\Python\251101 Webhook Server version 3.py @ 14: 1
File Edit View Run Tools Help
251101 Webhook Server version 3.py
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 @app.route('/webhook', methods=['POST'])
6 def webhook():
7     data = request.get_json(force=True)
8     print("Webhook recibido:")
9     print(data)
10    return jsonify({"status": "ok"}), 200
11
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=5000)
14

Shell
>>> %Run '251101 Webhook Server version 3.py'
* Serving Flask app '251101 Webhook Server version 3'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.30:5000
Press CTRL+C to quit
192.168.0.200 - - [03/Nov/2025 19:39:04] "GET / HTTP/1.1" 404 -
192.168.0.200 - - [03/Nov/2025 19:39:08] "GET / HTTP/1.1" 404 -
192.168.0.200 - - [03/Nov/2025 19:39:10] "GET / HTTP/1.1" 404 -
192.168.0.200 - - [03/Nov/2025 19:39:14] "GET / HTTP/1.1" 404 -
192.168.0.200 - - [03/Nov/2025 19:39:17] "GET / HTTP/1.1" 404 -
```

Paquetes HTTP recibidos al activar el pulsador (ON)