By Cristopher Mejia

IOT DISTRIBUTED SYSTEM

with MQTT, Flask & SQLite

IoT Distributed System with MQTT, Flask & SQLite

1. Instalación en Ubuntu

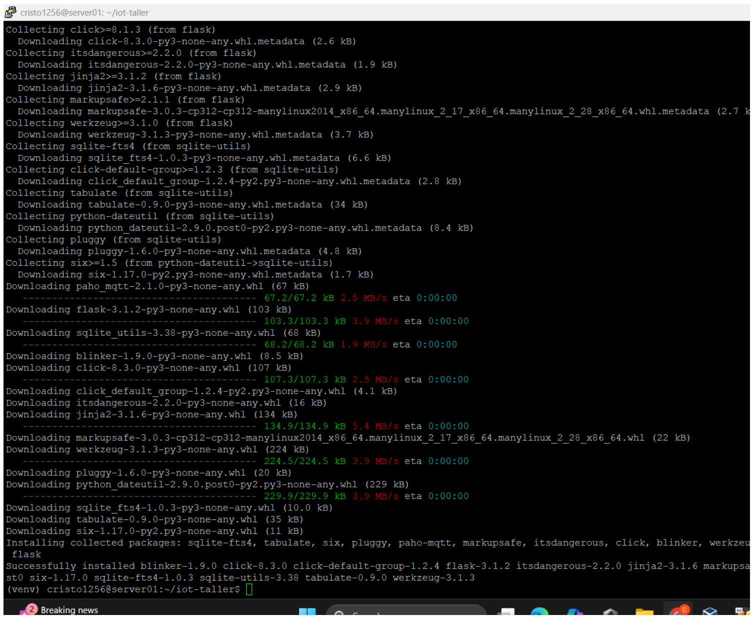
Actualizar sistema
sudo apt update && sudo apt upgrade -y

Instala el paquete python3-venv**:**
sudo apt update
sudo apt install -y python3.12-venv

Recrea el entorno virtual:
python3 -m venv venv

Activa el entorno virtual:
source venv/bin/activate
Verás que tu prompt cambia a algo como:
(venv) cristo1256@server01:~/iot-taller$

Instala las librerías necesarias dentro del entorno:
pip install paho-mqtt flask sqlite-utils



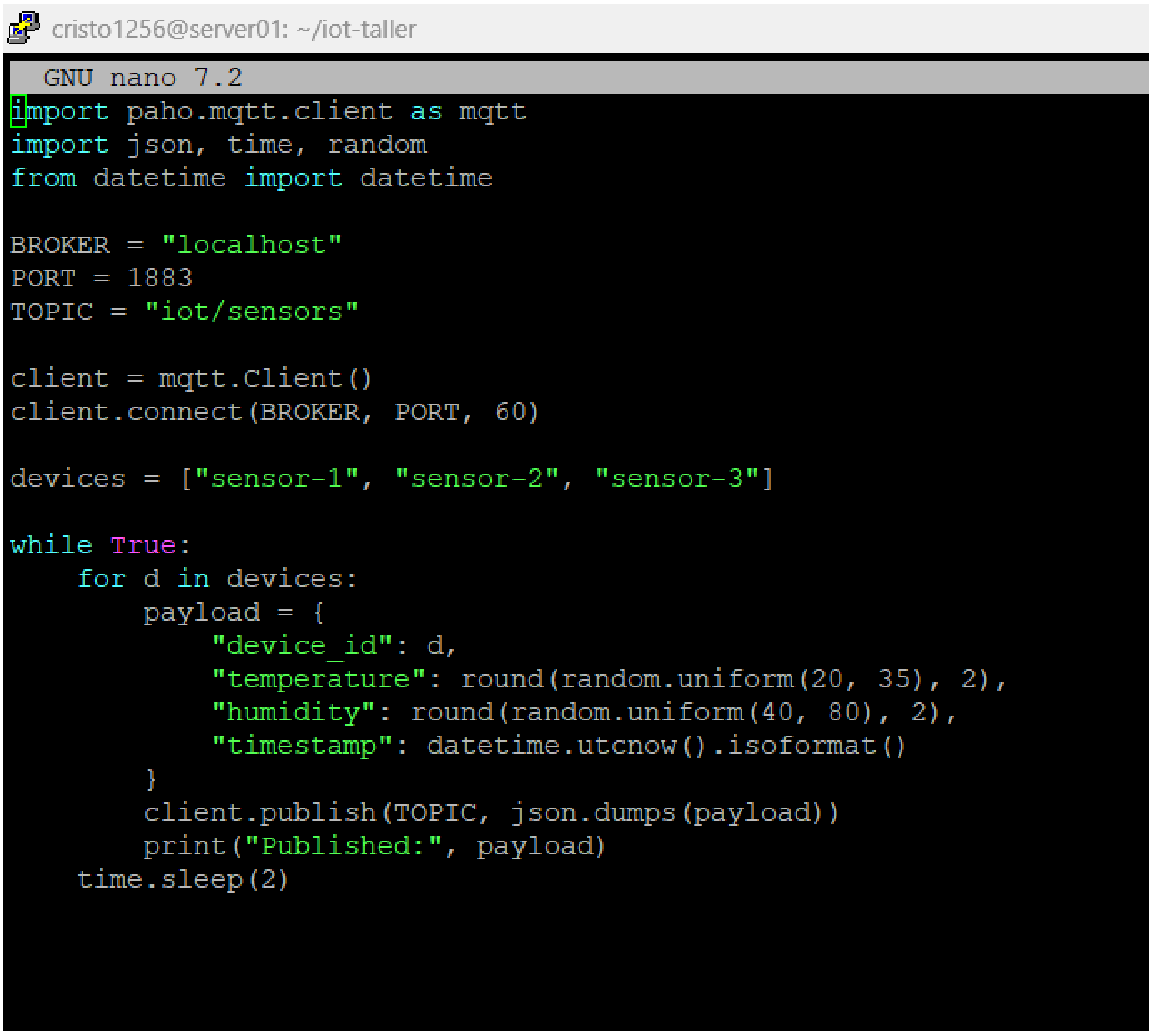
cristo1256@server01: ~/iot-taller

```
Collecting click>=8.1.3 (from flask)
  Downloading click-8.3.0-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask)
  Downloading markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (2.7 k
Collecting werkzeug>=3.1.0 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting sqlite-fts4 (from sqlite-utils)
  Downloading sqlite_fts4-1.0.3-py3-none-any.whl.metadata (6.6 kB)
Collecting click-default-group>=1.2.3 (from sqlite-utils)
  Downloading click_default_group-1.2.4-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting tabulate (from sqlite-utils)
  Downloading tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)
Collecting python-dateutil (from sqlite-utils)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pluggy (from sqlite-utils)
  Downloading pluggy-1.6.0-py3-none-any.whl.metadata (4.8 kB)
Collecting six>=1.5 (from python-dateutil->sqlite-utils)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
   ---------------------------------------- 67.2/67.2 kB 2.5 MB/s eta 0:00:00
Downloading flask-3.1.2-py3-none-any.whl (103 kB)
   ---------------------------------------- 103.3/103.3 kB 3.9 MB/s eta 0:00:00
Downloading sqlite_utils-3.38-py3-none-any.whl (68 kB)
   ---------------------------------------- 68.2/68.2 kB 1.9 MB/s eta 0:00:00
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.3.0-py3-none-any.whl (107 kB)
   ---------------------------------------- 107.3/107.3 kB 2.5 MB/s eta 0:00:00
Downloading click_default_group-1.2.4-py2.py3-none-any.whl (4.1 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
   ---------------------------------------- 134.9/134.9 kB 5.4 MB/s eta 0:00:00
Downloading markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (22 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
   ---------------------------------------- 224.5/224.5 kB 3.9 MB/s eta 0:00:00
Downloading pluggy-1.6.0-py3-none-any.whl (20 kB)
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
   ---------------------------------------- 229.9/229.9 kB 3.9 MB/s eta 0:00:00
Downloading sqlite_fts4-1.0.3-py3-none-any.whl (10.0 kB)
Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: sqlite-fts4, tabulate, six, pluggy, paho-mqtt, markupsafe, itsdangerous, click, blinker, werkzeu
 flask
Successfully installed blinker-1.9.0 click-8.3.0 click-default-group-1.2.4 flask-3.1.2 itsdangerous-2.2.0 jinja2-3.1.6 markupsa
st0 six-1.17.0 sqlite-fts4-1.0.3 sqlite-utils-3.38 tabulate-0.9.0 werkzeug-3.1.3
(venv) cristo1256@server01:~/iot-taller$
```

2. Simulación de Sensores IoT (Publishers)

Archivo: sensor.py

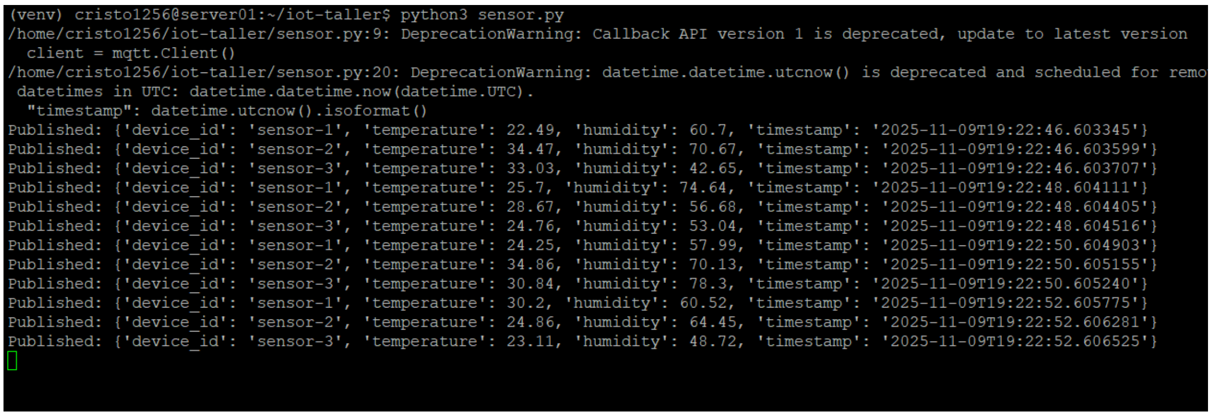
```
import paho.mqtt.client as mqtt
import json, time, random
from datetime import datetime

BROKER = "localhost"
PORT = 1883
TOPIC = "iot/sensors"

client = mqtt.Client()
client.connect(BROKER, PORT, 60)

devices = ["sensor-1", "sensor-2", "sensor-3"]

while True:
    for d in devices:
        payload = {
            "device_id": d,
            "temperature": round(random.uniform(20, 35), 2),
            "humidity": round(random.uniform(40, 80), 2),
            "timestamp": datetime.utcnow().isoformat()
        }
        client.publish(TOPIC, json.dumps(payload))
        print("Published:", payload)
    time.sleep(2)
```



```
cristo1256@server01: ~/iot-taller
  GNU nano 7.2
import paho.mqtt.client as mqtt
import json, time, random
from datetime import datetime

BROKER = "localhost"
PORT = 1883
TOPIC = "iot/sensors"

client = mqtt.Client()
client.connect(BROKER, PORT, 60)

devices = ["sensor-1", "sensor-2", "sensor-3"]

while True:
    for d in devices:
        payload = {
            "device_id": d,
            "temperature": round(random.uniform(20, 35), 2),
            "humidity": round(random.uniform(40, 80), 2),
            "timestamp": datetime.utcnow().isoformat()
        }
        client.publish(TOPIC, json.dumps(payload))
        print("Published:", payload)
    time.sleep(2)
```

Ejecutar:

python3 sensor.py

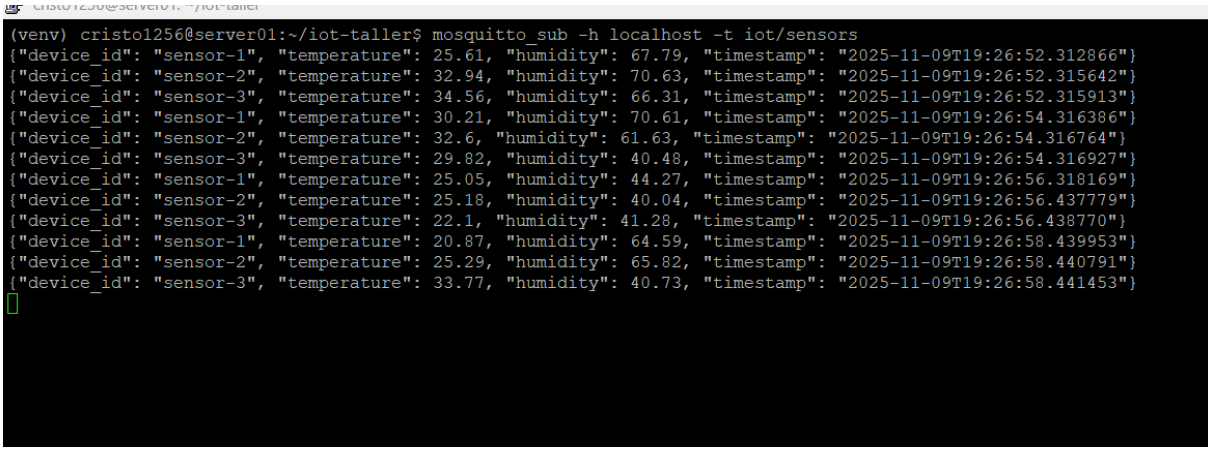


```
(venv) cristo1256@server01:~/iot-taller$ python3 sensor.py
/home/cristo1256/iot-taller/sensor.py:9: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
/home/cristo1256/iot-taller/sensor.py:20: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for remo
 datetimes in UTC: datetime.datetime.now(datetime.UTC).
  "timestamp": datetime.utcnow().isoformat()
Published: {'device_id': 'sensor-1', 'temperature': 22.49, 'humidity': 60.7, 'timestamp': '2025-11-09T19:22:46.603345'}
Published: {'device_id': 'sensor-2', 'temperature': 34.47, 'humidity': 70.67, 'timestamp': '2025-11-09T19:22:46.603599'}
Published: {'device_id': 'sensor-3', 'temperature': 33.03, 'humidity': 42.65, 'timestamp': '2025-11-09T19:22:46.603707'}
Published: {'device_id': 'sensor-1', 'temperature': 25.7, 'humidity': 74.64, 'timestamp': '2025-11-09T19:22:48.604111'}
Published: {'device_id': 'sensor-2', 'temperature': 28.67, 'humidity': 56.68, 'timestamp': '2025-11-09T19:22:48.604405'}
Published: {'device_id': 'sensor-3', 'temperature': 24.76, 'humidity': 53.04, 'timestamp': '2025-11-09T19:22:48.604516'}
Published: {'device_id': 'sensor-1', 'temperature': 24.25, 'humidity': 57.99, 'timestamp': '2025-11-09T19:22:50.604903'}
Published: {'device_id': 'sensor-2', 'temperature': 34.86, 'humidity': 70.13, 'timestamp': '2025-11-09T19:22:50.605155'}
Published: {'device_id': 'sensor-3', 'temperature': 30.84, 'humidity': 78.3, 'timestamp': '2025-11-09T19:22:50.605240'}
Published: {'device_id': 'sensor-1', 'temperature': 30.2, 'humidity': 60.52, 'timestamp': '2025-11-09T19:22:52.605775'}
Published: {'device_id': 'sensor-2', 'temperature': 24.86, 'humidity': 64.45, 'timestamp': '2025-11-09T19:22:52.606281'}
Published: {'device_id': 'sensor-3', 'temperature': 23.11, 'humidity': 48.72, 'timestamp': '2025-11-09T19:22:52.606525'}
```

3. Middleware de Comunicación (Broker MQTT)

Ya instalado con Mosquitto.
Puedes probar con:

```
mosquitto_sub -h localhost -t iot/sensors
```



```
(venv) cristo1256@server01:~/iot-taller$ mosquitto_sub -h localhost -t iot/sensors
{"device_id": "sensor-1", "temperature": 25.61, "humidity": 67.79, "timestamp": "2025-11-09T19:26:52.312866"}
{"device_id": "sensor-2", "temperature": 32.94, "humidity": 70.63, "timestamp": "2025-11-09T19:26:52.315642"}
{"device_id": "sensor-3", "temperature": 34.56, "humidity": 66.31, "timestamp": "2025-11-09T19:26:52.315913"}
{"device_id": "sensor-1", "temperature": 30.21, "humidity": 70.61, "timestamp": "2025-11-09T19:26:54.316386"}
{"device_id": "sensor-2", "temperature": 32.6, "humidity": 61.63, "timestamp": "2025-11-09T19:26:54.316764"}
{"device_id": "sensor-3", "temperature": 29.82, "humidity": 40.48, "timestamp": "2025-11-09T19:26:54.316927"}
{"device_id": "sensor-1", "temperature": 25.05, "humidity": 44.27, "timestamp": "2025-11-09T19:26:56.318169"}
{"device_id": "sensor-2", "temperature": 25.18, "humidity": 40.04, "timestamp": "2025-11-09T19:26:56.437779"}
{"device_id": "sensor-3", "temperature": 22.1, "humidity": 41.28, "timestamp": "2025-11-09T19:26:56.438770"}
{"device_id": "sensor-1", "temperature": 20.87, "humidity": 64.59, "timestamp": "2025-11-09T19:26:58.439953"}
{"device_id": "sensor-2", "temperature": 25.29, "humidity": 65.82, "timestamp": "2025-11-09T19:26:58.440791"}
{"device_id": "sensor-3", "temperature": 33.77, "humidity": 40.73, "timestamp": "2025-11-09T19:26:58.441453"}
```

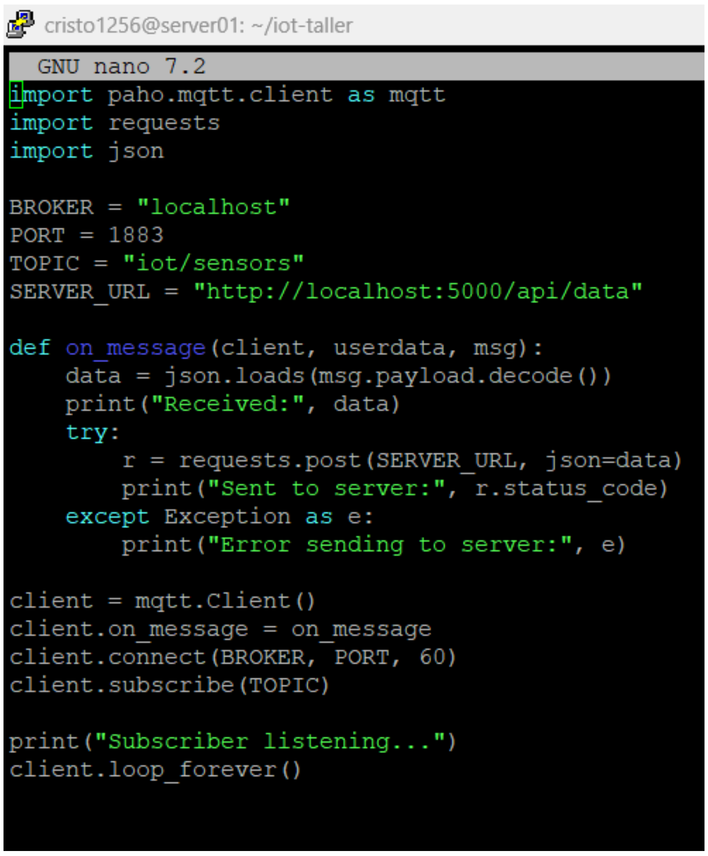
Deberías ver los mensajes JSON publicados por los sensores.

4. Subscriber que envía datos al Servidor Central

Archivo: subscriber.py

```
import paho.mqtt.client as mqtt
import requests
import json

BROKER = "localhost"
PORT = 1883
TOPIC = "iot/sensors"
SERVER_URL = "http://localhost:5000/api/data"

def on_message(client, userdata, msg):
    data = json.loads(msg.payload.decode())
    print("Received:", data)
    try:
        r = requests.post(SERVER_URL, json=data)
        print("Sent to server:", r.status_code)
    except Exception as e:
        print("Error sending to server:", e)

client = mqtt.Client()
client.on_message = on_message
client.connect(BROKER, PORT, 60)
client.subscribe(TOPIC)
```

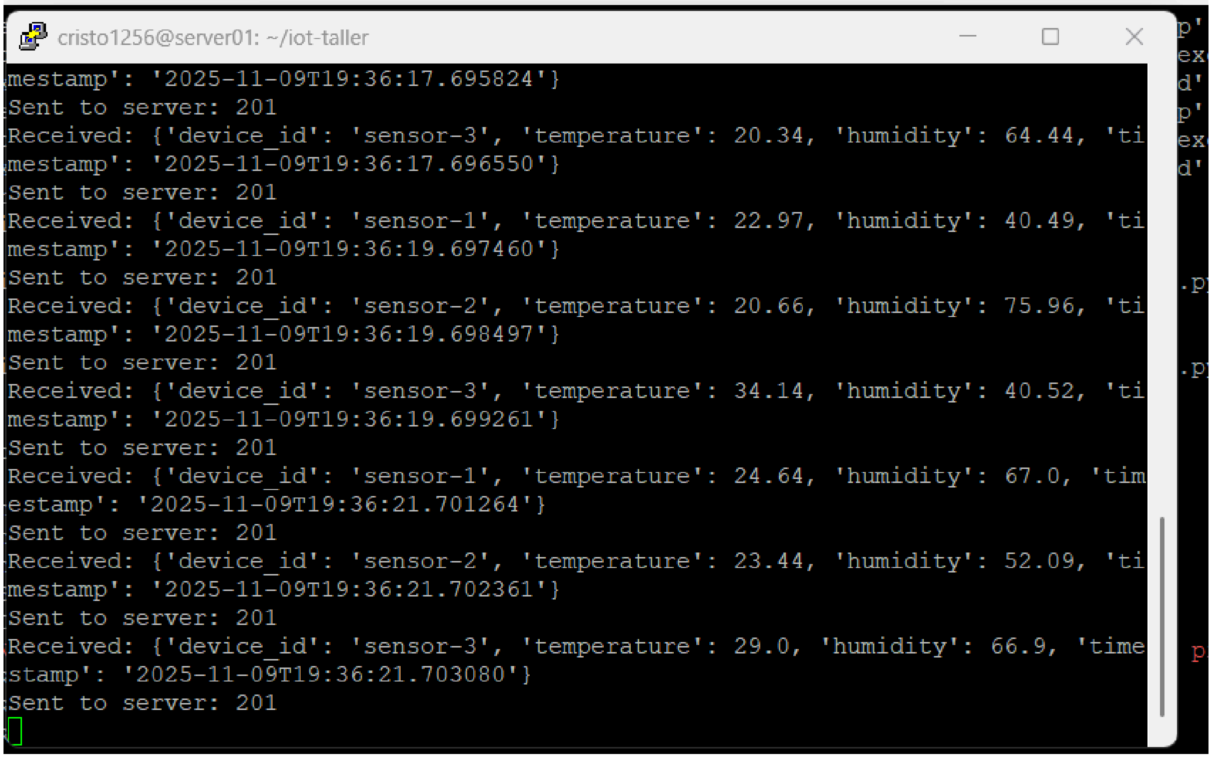
print("Subscriber listening...")
client.loop_forever()



```
cristo1256@server01: ~/iot-taller
  GNU nano 7.2
import paho.mqtt.client as mqtt
import requests
import json

BROKER = "localhost"
PORT = 1883
TOPIC = "iot/sensors"
SERVER_URL = "http://localhost:5000/api/data"

def on_message(client, userdata, msg):
    data = json.loads(msg.payload.decode())
    print("Received:", data)
    try:
        r = requests.post(SERVER_URL, json=data)
        print("Sent to server:", r.status_code)
    except Exception as e:
        print("Error sending to server:", e)

client = mqtt.Client()
client.on_message = on_message
client.connect(BROKER, PORT, 60)
client.subscribe(TOPIC)

print("Subscriber listening...")
client.loop_forever()
```

Ejecutar:

python3 subscriber.py



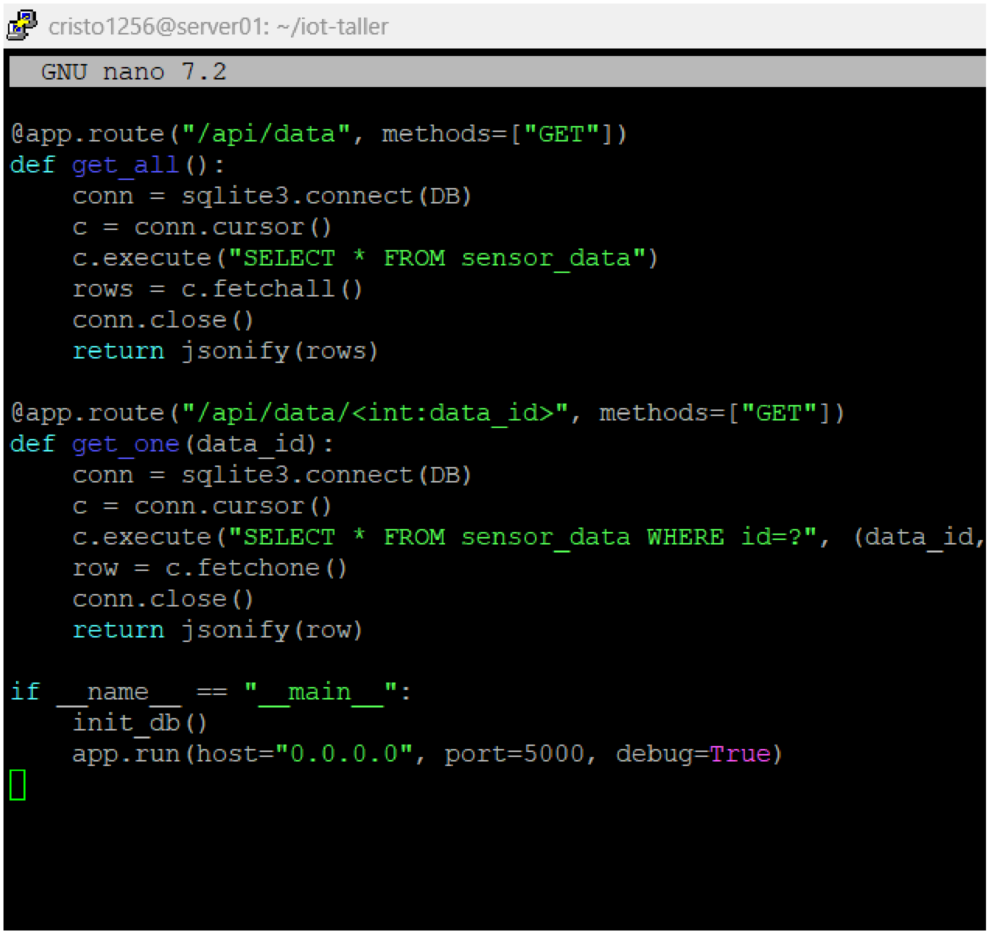
```
cristo1256@server01: ~/iot-taller
mestamp': '2025-11-09T19:36:17.695824'}
Sent to server: 201
Received: {'device_id': 'sensor-3', 'temperature': 20.34, 'humidity': 64.44, 'ti
mestamp': '2025-11-09T19:36:17.696550'}
Sent to server: 201
Received: {'device_id': 'sensor-1', 'temperature': 22.97, 'humidity': 40.49, 'ti
mestamp': '2025-11-09T19:36:19.697460'}
Sent to server: 201
Received: {'device_id': 'sensor-2', 'temperature': 20.66, 'humidity': 75.96, 'ti
mestamp': '2025-11-09T19:36:19.698497'}
Sent to server: 201
Received: {'device_id': 'sensor-3', 'temperature': 34.14, 'humidity': 40.52, 'ti
mestamp': '2025-11-09T19:36:19.699261'}
Sent to server: 201
Received: {'device_id': 'sensor-1', 'temperature': 24.64, 'humidity': 67.0, 'tim
estamp': '2025-11-09T19:36:21.701264'}
Sent to server: 201
Received: {'device_id': 'sensor-2', 'temperature': 23.44, 'humidity': 52.09, 'ti
mestamp': '2025-11-09T19:36:21.702361'}
Sent to server: 201
Received: {'device_id': 'sensor-3', 'temperature': 29.0, 'humidity': 66.9, 'time
stamp': '2025-11-09T19:36:21.703080'}
Sent to server: 201
```

5. Servidor Central con Flask + SQLite

Archivo: server.py

```
from flask import Flask, request, jsonify
import sqlite3

DB = "iot_data.db"

app = Flask(__name__)

# Crear tabla si no existe
def init_db():
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS sensor_data (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                device_id TEXT,
                temperature REAL,
                humidity REAL,
                timestamp TEXT
            )''')
    conn.commit()
    conn.close()

@app.route("/api/data", methods=["POST"])
def insert_data():
    data = request.get_json()
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute("INSERT INTO sensor_data (device_id, temperature, humidity,
timestamp) VALUES (?, ?, ?, ?)",
          (data["device_id"], data["temperature"], data["humidity"],
data["timestamp"]))
    conn.commit()
    conn.close()
    return jsonify({"status": "ok"}), 201

@app.route("/api/data", methods=["GET"])
```

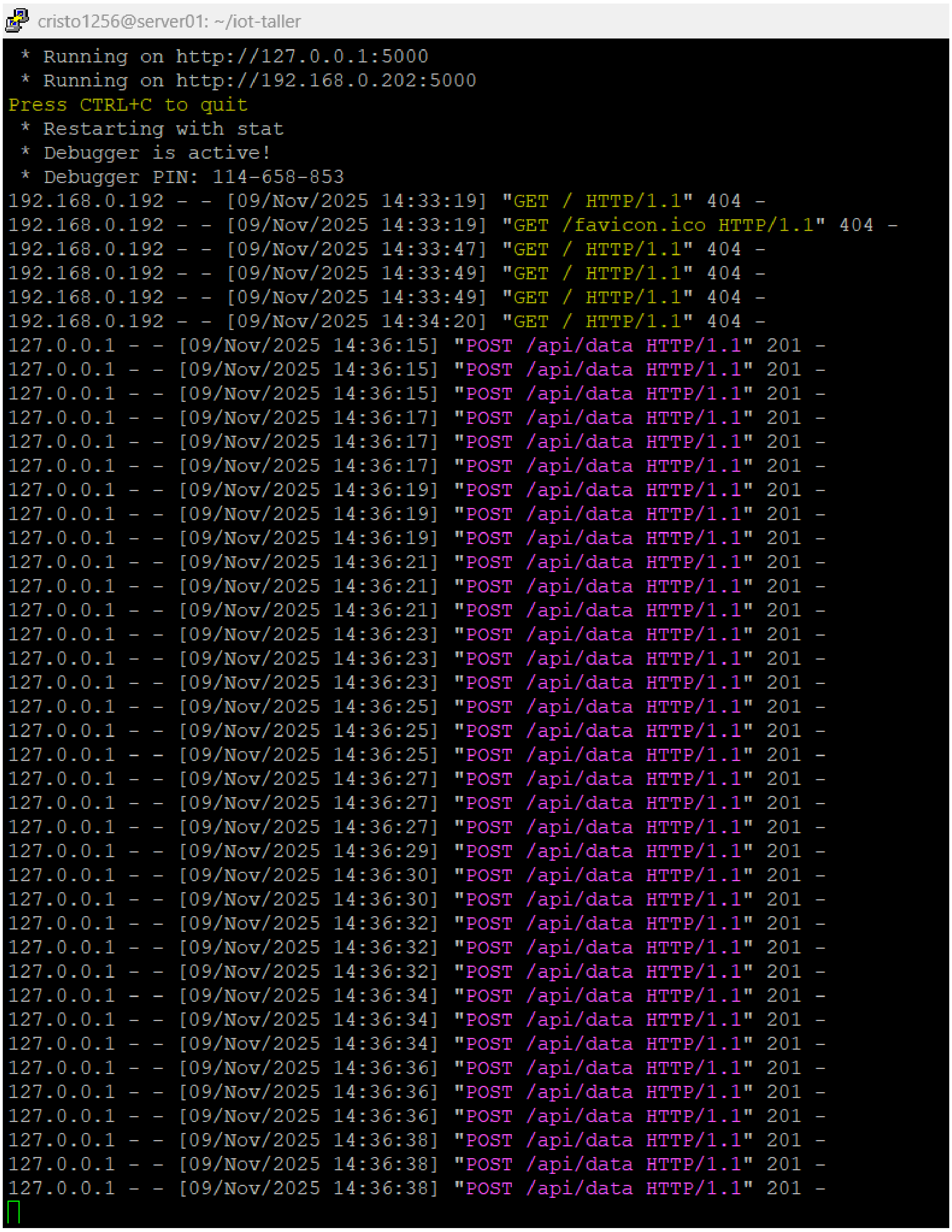
```
def get_all():
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute("SELECT * FROM sensor_data")
    rows = c.fetchall()
    conn.close()
    return jsonify(rows)

@app.route("/api/data/<int:data_id>", methods=["GET"])
def get_one(data_id):
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute("SELECT * FROM sensor_data WHERE id=?", (data_id,))
    row = c.fetchone()
    conn.close()
    return jsonify(row)

if __name__ == "__main__":
    init_db()
    app.run(host="0.0.0.0", port=5000, debug=True)
```



```
cristo1256@server01: ~/iot-taller
 GNU nano 7.2

@app.route("/api/data", methods=["GET"])
def get_all():
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute("SELECT * FROM sensor_data")
    rows = c.fetchall()
    conn.close()
    return jsonify(rows)

@app.route("/api/data/<int:data_id>", methods=["GET"])
def get_one(data_id):
    conn = sqlite3.connect(DB)
    c = conn.cursor()
    c.execute("SELECT * FROM sensor_data WHERE id=?", (data_id,
    row = c.fetchone()
    conn.close()
    return jsonify(row)

if __name__ == "__main__":
    init_db()
    app.run(host="0.0.0.0", port=5000, debug=True)
```

Ejecutar:

python3 server.py



cristo1256@server01: ~/iot-taller

```
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.0.202:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 114-658-853
192.168.0.192 - - [09/Nov/2025 14:33:19] "GET / HTTP/1.1" 404 -
192.168.0.192 - - [09/Nov/2025 14:33:19] "GET /favicon.ico HTTP/1.1" 404 -
192.168.0.192 - - [09/Nov/2025 14:33:47] "GET / HTTP/1.1" 404 -
192.168.0.192 - - [09/Nov/2025 14:33:49] "GET / HTTP/1.1" 404 -
192.168.0.192 - - [09/Nov/2025 14:33:49] "GET / HTTP/1.1" 404 -
192.168.0.192 - - [09/Nov/2025 14:34:20] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [09/Nov/2025 14:36:15] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:15] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:15] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:17] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:17] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:17] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:19] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:19] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:19] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:21] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:21] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:21] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:23] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:23] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:23] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:25] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:25] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:25] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:27] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:27] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:27] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:29] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:30] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:30] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:32] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:32] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:32] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:34] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:34] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:34] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:36] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:36] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:36] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:38] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:38] "POST /api/data HTTP/1.1" 201 -
127.0.0.1 - - [09/Nov/2025 14:36:38] "POST /api/data HTTP/1.1" 201 -
```

6. Pruebas de la API REST

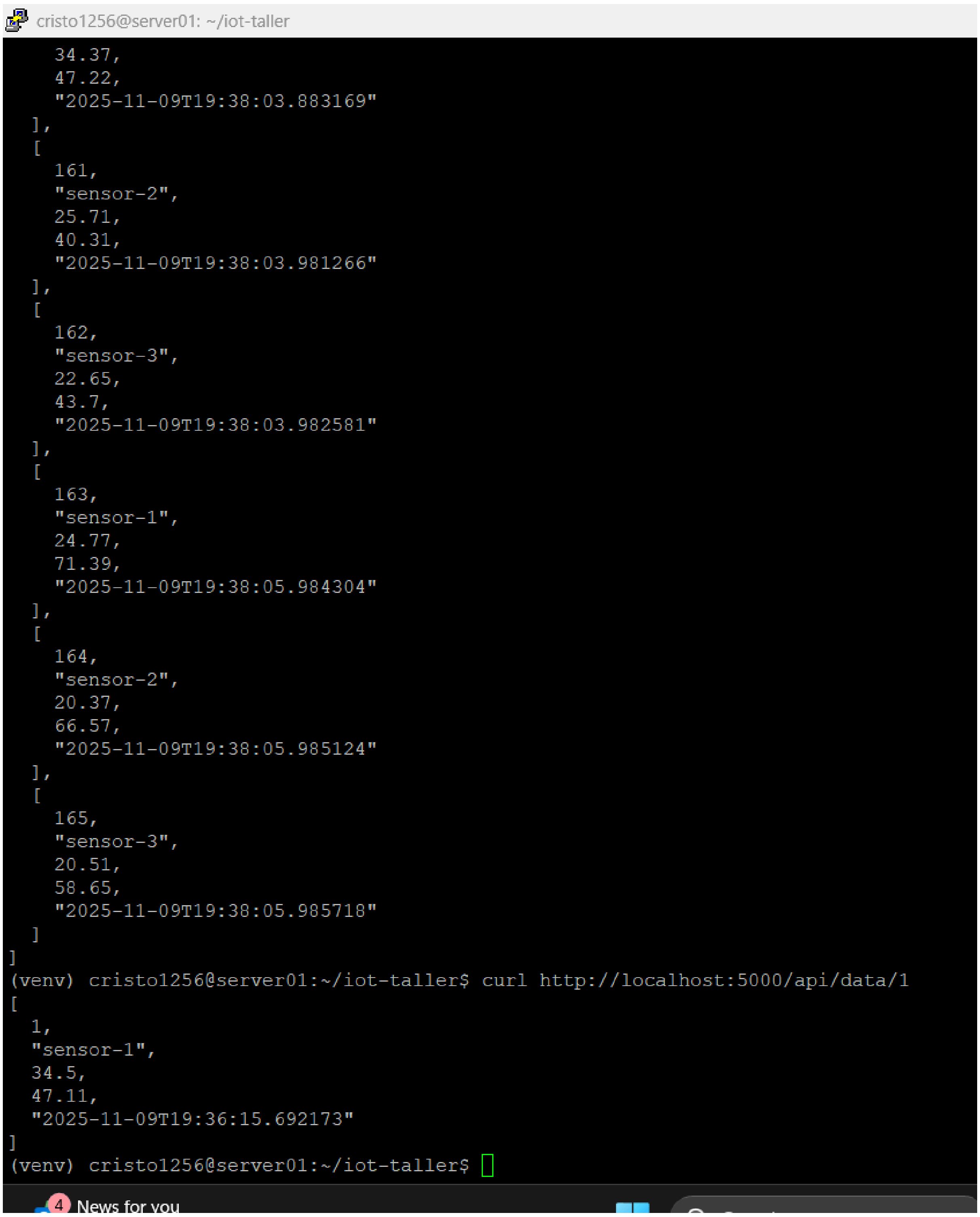
Insertar datos (ya lo hace el subscriber automáticamente).

Consultar todos los datos:

curl http://localhost:5000/api/data

Consultar un registro específico:

curl http://localhost:5000/api/data/1



cristo1256@server01: ~/iot-taller

```
    34.37,
    47.22,
    "2025-11-09T19:38:03.883169"
  ],
  [
    161,
    "sensor-2",
    25.71,
    40.31,
    "2025-11-09T19:38:03.981266"
  ],
  [
    162,
    "sensor-3",
    22.65,
    43.7,
    "2025-11-09T19:38:03.982581"
  ],
  [
    163,
    "sensor-1",
    24.77,
    71.39,
    "2025-11-09T19:38:05.984304"
  ],
  [
    164,
    "sensor-2",
    20.37,
    66.57,
    "2025-11-09T19:38:05.985124"
  ],
  [
    165,
    "sensor-3",
    20.51,
    58.65,
    "2025-11-09T19:38:05.985718"
  ]
]
(venv) cristo1256@server01:~/iot-taller$ curl http://localhost:5000/api/data/1
[
  1,
  "sensor-1",
  34.5,
  47.11,
  "2025-11-09T19:36:15.692173"
]
(venv) cristo1256@server01:~/iot-taller$
```

7. Resultados Esperados

- **Sensores** publican datos periódicos en MQTT.
- **Mosquitto** recibe y distribuye los mensajes.
- **Subscriber** escucha y reenvía al servidor central.
- **Servidor Flask** almacena en SQLite y expone API REST.
- **Consultas** vía navegador, cURL o Postman muestran los datos recolectados.

8. Conclusiones

- Se implementó un sistema IoT distribuido básico con sensores simulados, middleware MQTT y servidor central.
- Se cumplió con los objetivos del taller: simulación, comunicación y almacenamiento con API REST.
- La arquitectura es extensible: se pueden añadir más sensores, métricas o migrar a bases de datos más potentes.

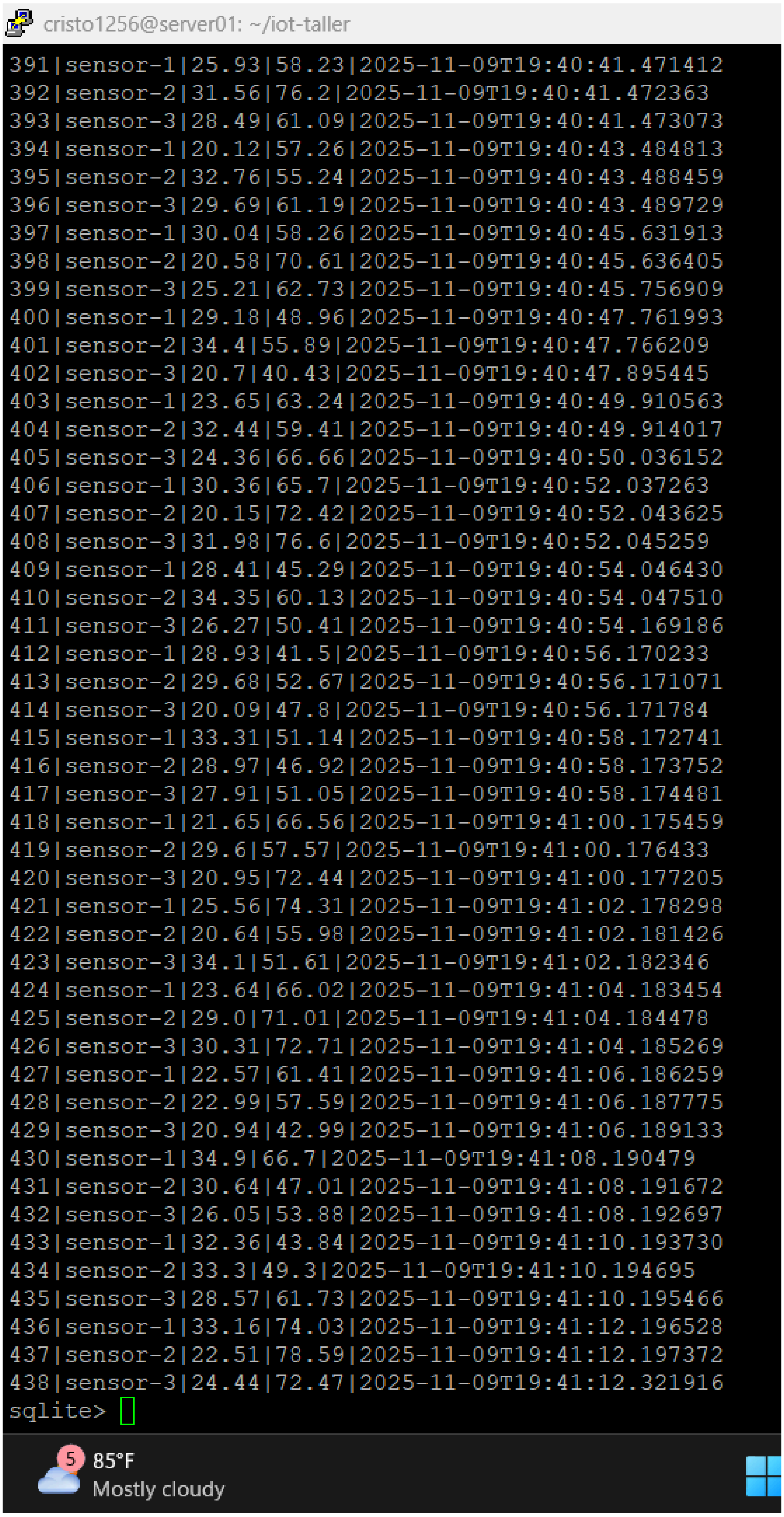
Anexo: Uso de SQLite en Ubuntu CLI

Instalar SQLite:

bash

```
sudo apt install sqlite3 -y
```

1. Abrir la base de datos:
   ```
   sqlite3 iot_data.db
   ```
2. Ver tablas:
   ```
   .tables
   ```
3. Consultar registros:
   ```
   SELECT * FROM sensor_data;
   ```
4. Salir:
   ```
   .exit
   ```
5. Ejemplo de salida:



```
391|sensor-1|25.93|58.23|2025-11-09T19:40:41.471412
392|sensor-2|31.56|76.2|2025-11-09T19:40:41.472363
393|sensor-3|28.49|61.09|2025-11-09T19:40:41.473073
394|sensor-1|20.12|57.26|2025-11-09T19:40:43.484813
395|sensor-2|32.76|55.24|2025-11-09T19:40:43.488459
396|sensor-3|29.69|61.19|2025-11-09T19:40:43.489729
397|sensor-1|30.04|58.26|2025-11-09T19:40:45.631913
398|sensor-2|20.58|70.61|2025-11-09T19:40:45.636405
399|sensor-3|25.21|62.73|2025-11-09T19:40:45.756909
400|sensor-1|29.18|48.96|2025-11-09T19:40:47.761993
401|sensor-2|34.4|55.89|2025-11-09T19:40:47.766209
402|sensor-3|20.7|40.43|2025-11-09T19:40:47.895445
403|sensor-1|23.65|63.24|2025-11-09T19:40:49.910563
404|sensor-2|32.44|59.41|2025-11-09T19:40:49.914017
405|sensor-3|24.36|66.66|2025-11-09T19:40:50.036152
406|sensor-1|30.36|65.7|2025-11-09T19:40:52.037263
407|sensor-2|20.15|72.42|2025-11-09T19:40:52.043625
408|sensor-3|31.98|76.6|2025-11-09T19:40:52.045259
409|sensor-1|28.41|45.29|2025-11-09T19:40:54.046430
410|sensor-2|34.35|60.13|2025-11-09T19:40:54.047510
411|sensor-3|26.27|50.41|2025-11-09T19:40:54.169186
412|sensor-1|28.93|41.5|2025-11-09T19:40:56.170233
413|sensor-2|29.68|52.67|2025-11-09T19:40:56.171071
414|sensor-3|20.09|47.8|2025-11-09T19:40:56.171784
415|sensor-1|33.31|51.14|2025-11-09T19:40:58.172741
416|sensor-2|28.97|46.92|2025-11-09T19:40:58.173752
417|sensor-3|27.91|51.05|2025-11-09T19:40:58.174481
418|sensor-1|21.65|66.56|2025-11-09T19:41:00.175459
419|sensor-2|29.6|57.57|2025-11-09T19:41:00.176433
420|sensor-3|20.95|72.44|2025-11-09T19:41:00.177205
421|sensor-1|25.56|74.31|2025-11-09T19:41:02.178298
422|sensor-2|20.64|55.98|2025-11-09T19:41:02.181426
423|sensor-3|34.1|51.61|2025-11-09T19:41:02.182346
424|sensor-1|23.64|66.02|2025-11-09T19:41:04.183454
425|sensor-2|29.0|71.01|2025-11-09T19:41:04.184478
426|sensor-3|30.31|72.71|2025-11-09T19:41:04.185269
427|sensor-1|22.57|61.41|2025-11-09T19:41:06.186259
428|sensor-2|22.99|57.59|2025-11-09T19:41:06.187775
429|sensor-3|20.94|42.99|2025-11-09T19:41:06.189133
430|sensor-1|34.9|66.7|2025-11-09T19:41:08.190479
431|sensor-2|30.64|47.01|2025-11-09T19:41:08.191672
432|sensor-3|26.05|53.88|2025-11-09T19:41:08.192697
433|sensor-1|32.36|43.84|2025-11-09T19:41:10.193730
434|sensor-2|33.3|49.3|2025-11-09T19:41:10.194695
435|sensor-3|28.57|61.73|2025-11-09T19:41:10.195466
436|sensor-1|33.16|74.03|2025-11-09T19:41:12.196528
437|sensor-2|22.51|78.59|2025-11-09T19:41:12.197372
438|sensor-3|24.44|72.47|2025-11-09T19:41:12.321916
sqlite>
```