

PDI - TP 4 - A - RestAPI - IoT - MQTT

(Entrega 24/06/2024 o en Primera Fecha de Final de Julio)

Ejecuto y muestro el estado del servidor(mosquitto) que escucha en el puerto 1883(por defecto).

```
mint@mint:~$  
mint@mint:~$  
mint@mint:~$ sudo systemctl start mosquitto  
mint@mint:~$ sudo systemctl enable mosquitto  
Synchronizing state of mosquitto.service with SysV service script with /lib/systemd/systemd-sysv-i  
ninstall.  
Executing: /lib/systemd/systemd-sysv-install enable mosquitto  
mint@mint:~$  
mint@mint:~$ sudo systemctl status mosquitto  
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker  
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sat 2024-06-29 15:37:48 UTC; 5h 21min ago  
     Docs: man:mosquitto.conf(5)  
           man:mosquitto(8)  
  Main PID: 3258 (mosquitto)  
    Tasks: 3 (limit: 8779)  
   Memory: 1.9M  
    CGroup: /system.slice/mosquitto.service  
           └─3258 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf  
  
Jun 29 15:37:48 mint systemd[1]: Starting Mosquitto MQTT v3.1/v3.1.1 Broker...  
Jun 29 15:37:48 mint mosquitto[3258]: [ 700.446316]~DLT~ 3258~INFO ~FIFO /tmp/dlt cannot be  
Jun 29 15:37:48 mint systemd[1]: Started Mosquitto MQTT v3.1/v3.1.1 Broker.  
lines 1-14/14 (END)
```

Captura I

El cliente se suscribe en sitio1 y espera a recibir mensajes de ese tema. El mosquitto escucha el puerto MQTT, gestiona las suscripciones y publica mensajes a los clientes suscritos a dichos temas. A continuacion, se muestran los resultados del cliente que se suscribe al tema y recibe los mensajes especificos:

```
mint@mint:/etc/mosquitto$  
mint@mint:/etc/mosquitto$  
mint@mint:/etc/mosquitto$ mosquitto_sub -h localhost -t sitio1/temperatura  
Sitio1 Temp. = 22 C  
Sitio1 Temp. = 23 C
```

Captura II

- > -h localhost: Especifica el servidor MQTT al que me estoy conectando.
- > -t sitio1/temperatura: Especifica el tema al que quiero suscribirme.

A continuacion se publican los mensajes para conexiones MQTT:

A terminal window titled 'mint@mint: /etc/mosquitto' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a sequence of commands: 'cd /etc', 'cd mosquitto', and two 'mosquitto_pub' commands publishing temperature data to the 'sitiol/temperatura' topic. The first command publishes '22 C' and the second publishes '23 C'.

```
mint@mint:~$  
mint@mint:~$ cd /etc  
mint@mint:/etc$ cd mosquitto  
mint@mint:/etc/mosquitto$  
mint@mint:/etc/mosquitto$ mosquitto_pub -h localhost -t sitiol/temperatura -m "Sitiol Temp. = 22 C"  
mint@mint:/etc/mosquitto$ mosquitto_pub -h localhost -t sitiol/temperatura -m "Sitiol Temp. = 23 C"  
mint@mint:/etc/mosquitto$
```

Captura III

PDI - TP 4 - B - IoT MQTT (Entrega 24/06/2024 o en Primera Fecha de Final de Julio)

El publisher simula la generacion de sensores y los publica en mosquitto (broker MQTT). Los publica en el puerto 1883 y se deja ejecutando.

```
mint@mint: ~/Documents/TP4-Protocolos
File Edit View Search Terminal Help

mint@mint:~$ cd Documents/TP4-Protocolos
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$ sudo systemctl restart mosquitto
mint@mint:~/Documents/TP4-Protocolos$ sudo systemctl start mosquitto
mint@mint:~/Documents/TP4-Protocolos$ sudo systemctl enable mosquitto
Synchronizing state of mosquitto.service with SysV service script with /lib/systemd/systemd-sysv
-install.
Executing: /lib/systemd/systemd-sysv-install enable mosquitto
mint@mint:~/Documents/TP4-Protocolos$ sudo systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
   Active: active (running) since Sun 2024-06-30 04:39:31 UTC; 26s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Main PID: 15962 (mosquitto)
     Tasks: 3 (limit: 8779)
    Memory: 1.2M
    CGroup: /system.slice/mosquitto.service
            └─15962 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Jun 30 04:39:31 mint systemd[1]: Starting Mosquitto MQTT v3.1/v3.1.1 Broker...
Jun 30 04:39:31 mint mosquitto[15962]: [47603.276107]~DLT~15962~INFO ~FIFO >
Jun 30 04:39:31 mint systemd[1]: Started Mosquitto MQTT v3.1/v3.1.1 Broker.
lines 1-14/14 (END)
```

```
mint@mint: ~/Documents/TP4-Protocolos
File Edit View Search Terminal Help

mint@mint:~/Documents/TP4-Protocolos$ python3 mqtt_pub_r1.py
mqtt_pub_r1.py:16: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
2024-06-29 21:44:36,139 INFO: Iniciando publicador MQTT
2024-06-29 21:44:36,140 DEBUG: Datos publicados en el tema 'sensores/datos': {'co2': 400, 'temp': 25.5, 'hum': 50.2, 'fecha': '2024-06-29T12:00:00', 'lugar': 'Oficina', 'altura': 500, 'presion': 1013.25, 'presion_nm': 1010.5, 'temp_ext': 28.0}
2024-06-29 21:44:41,146 DEBUG: Datos publicados en el tema 'sensores/datos': {'co2': 400, 'temp': 25.5, 'hum': 50.2, 'fecha': '2024-06-29T12:00:00', 'lugar': 'Oficina', 'altura': 500, 'presion': 1013.25, 'presion_nm': 1010.5, 'temp_ext': 28.0}
2024-06-29 21:44:46,149 DEBUG: Datos publicados en el tema 'sensores/datos': {'co2': 400, 'temp': 25.5, 'hum': 50.2, 'fecha': '2024-06-29T12:00:00', 'lugar': 'Oficina', 'altura': 500, 'presion': 1013.25, 'presion_nm': 1010.5, 'temp_ext': 28.0}
2024-06-29 21:44:51,155 DEBUG: Datos publicados en el tema 'sensores/datos': {'co2': 400, 'temp': 25.5, 'hum': 50.2, 'fecha': '2024-06-29T12:00:00', 'lugar': 'Oficina', 'altura': 500, 'presion': 1013.25, 'presion_nm': 1010.5, 'temp_ext': 28.0}
2024-06-29 21:44:56,161 DEBUG: Datos publicados en el tema 'sensores/datos': {'co2': 400, 'temp': 25.5, 'hum': 50.2, 'fecha': '2024-06-29T12:00:00', 'lugar': 'Oficina', 'altura': 500, 'presion': 1013.25, 'presion_nm': 1010.5, 'temp_ext': 28.0}
```

Salida Suscriptor conectado al MQTT broker:

```
mint@mint: ~/Documents/TP4-Protocolos
File Edit View Search Terminal Help
mint@mint:~$ cd Documents
mint@mint:~/Documents$ cd TP4-Protocolos
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$ python3 mqtt_sub_r1.py
mqtt_sub_r1.py:21: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
2024-06-30 03:49:23,106 INFO: Conectando al broker MQTT
2024-06-30 03:49:23,106 INFO: Conectado al broker MQTT
2024-06-30 03:49:25,095 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:30,101 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:35,105 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:40,111 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:45,117 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:50,121 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:49:55,127 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:50:00,133 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-29T12:00:00", "lugar": "Oficina", "altura": 500, "presion": 1013.25, "presion_nm": 1010.5, "temp_ext": 28.0}
2024-06-30 03:50:05,137 DEBUG: Mensaje recibido en tópico sensores/datos: {"co2": 400, "temp": 25.5, "hum": 50.2, "fecha": "2024-06-
```

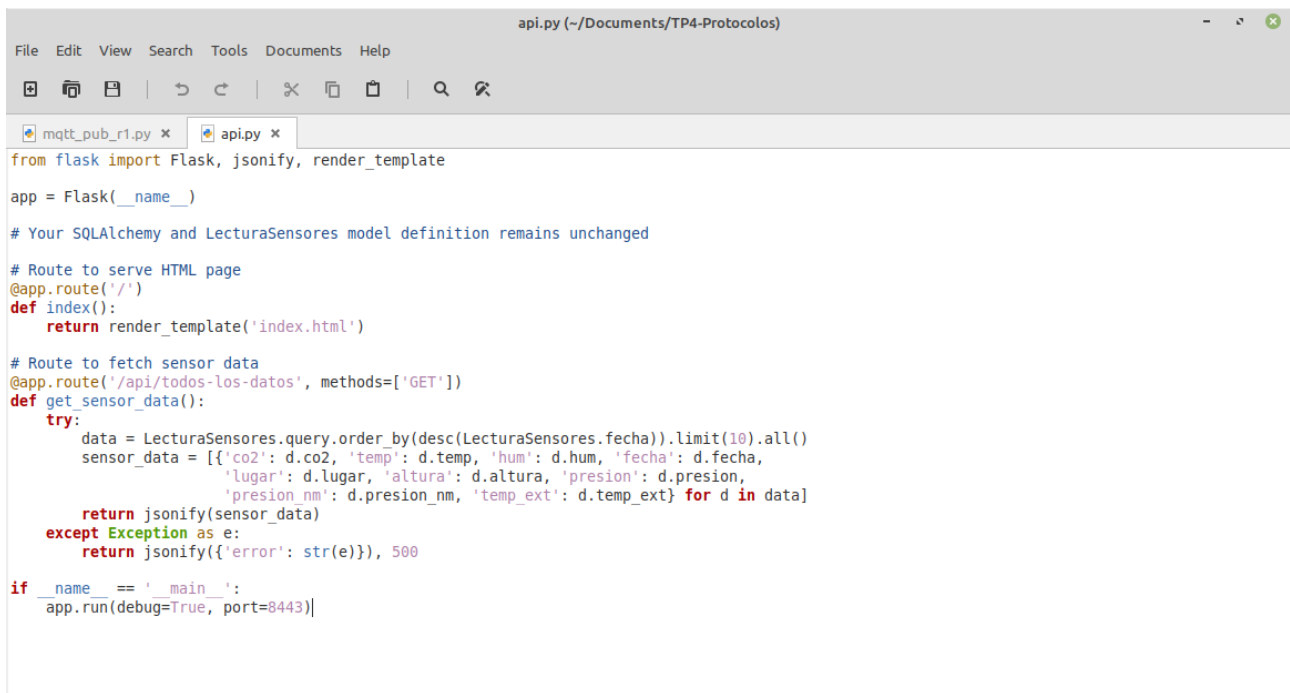
Ejecuto la REST API proporcionada para manejar los datos de los sensores. En el navegador web puedo visualizar los mismos.

Datos: desde la ruta <http://127.0.0.1:8443> puedo visualizar la pagina HTML con la tabla creada con las consultas POST desde api.py

```
mint@mint: ~/Documents/TP4-Protocolos
File Edit View Search Terminal Help
mint@mint:~/Documents/TP4-Protocolos$ sudo kill 15407
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$
mint@mint:~/Documents/TP4-Protocolos$ python3 api.py
* Serving Flask app 'api'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:8443
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 105-336-719
127.0.0.1 - - [30/Jun/2024 04:24:21] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2024 04:24:22] "GET /api/todos-los-datos?sort=&start=0&length=10 HTTP/1.1" 500 -
127.0.0.1 - - [30/Jun/2024 04:25:14] code 400, message Bad HTTP/0.9 request type ('\x10')
127.0.0.1 - - [30/Jun/2024 04:25:14] "\x10\x0c\x00\x04MQTT\x04\x02\x00<\x00\x00"
HTTPStatus.BAD_REQUEST -
127.0.0.1 - - [30/Jun/2024 04:25:24] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2024 04:25:24] "GET /api/todos-los-datos?sort=&start=0&length=10 HTTP/1.1" 500 -
```

Finalmente puedo ingresar a los datos
Codigo proporcionado:

api.py



```
api.py (~/Documents/TP4-Protocolos)

File Edit View Search Tools Documents Help

mqtt_pub_r1.py x api.py x

from flask import Flask, jsonify, render_template

app = Flask(__name__)

# Your SQLAlchemy and LecturaSensores model definition remains unchanged

# Route to serve HTML page
@app.route('/')
def index():
    return render_template('index.html')

# Route to fetch sensor data
@app.route('/api/todos-los-datos', methods=['GET'])
def get_sensor_data():
    try:
        data = LecturaSensores.query.order_by(desc(LecturaSensores.fecha)).limit(10).all()
        sensor_data = [{ 'co2': d.co2, 'temp': d.temp, 'hum': d.hum, 'fecha': d.fecha,
                          'lugar': d.lugar, 'altura': d.altura, 'presion': d.presion,
                          'presion_nm': d.presion_nm, 'temp_ext': d.temp_ext} for d in data]
        return jsonify(sensor_data)
    except Exception as e:
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True, port=8443)
```

mqtt_pub_r1.py

```
import json
import logging
import time
import paho.mqtt.client as mqtt

# Configuración del logging para depuración
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(levelname)s: %(message)s')

# Configuración de MQTT
MQTT_BROKER = "localhost"
MQTT_PORT = 8443
MQTT_TOPIC = "sensores/datos"

client = mqtt.Client()

def on_connect(client, userdata, flags, rc):
    if rc == 0:
        logging.info("Conectado al broker MQTT")
    else:
        logging.error(f"Conexión fallida con código de resultado: {rc}")

client.on_connect = on_connect

try:
    client.connect(MQTT_BROKER, MQTT_PORT, 60)
    logging.info("Conectando al broker MQTT")
```

```

while True:
    # Aquí simulamos la obtención de datos de sensores
    data = {
        'co2': 400,
        'temp': 25.5,
        'hum': 50.2,
        'fecha': '2024-06-29T12:00:00',
        'lugar': 'Oficina',
        'altura': 500,
        'presion': 1013.25,
        'presion_nm': 1010.5,
        'temp_ext': 28.0
    }

    client.publish(MQTT_TOPIC, json.dumps(data))
    logging.debug(f"Datos publicados en el tema '{MQTT_TOPIC}': {data}")
    time.sleep(5) # Publicar cada 5 segundos

except Exception as e:
    logging.error(f"Error al conectar al broker MQTT: {e}")

client.loop_forever()

```

mqtt_sub_r1.py

```

import logging
import paho.mqtt.client as mqtt

# Configuración del logging para depuración
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(levelname)s: %(message)s')

MQTT_BROKER = "localhost"
MQTT_PORT = 8443
MQTT_TOPIC = "sensores/datos"

def on_connect(client, userdata, flags, rc):
    if rc == 0:
        logging.info("Conectado al broker MQTT")
        client.subscribe(MQTT_TOPIC)
    else:
        logging.error(f"Conexión fallida con código de resultado: {rc}")

def on_message(client, userdata, msg):
    logging.debug(f"Mensaje recibido en tópico {msg.topic}: {msg.payload.decode()}")

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

```

```

try:
    client.connect(MQTT_BROKER, MQTT_PORT, 60)
    logging.info("Conectando al broker MQTT")
    client.loop_forever()
except Exception as e:
    logging.error(f"Error al conectar al broker MQTT: {e}")

```

index.html

```

<!DOCTYPE html>
<html>
<head>
  <title>Sensores de prueba</title>
  <link href="https://unpkg.com/gridjs/dist/theme/mermaid.min.css" rel="stylesheet" />
  <style>
    body {
      font-family: Sans-Serif;
    }
  </style>
</head>
<body>
  <div>
    <h1>Sensores de prueba</h1>
    <hr>
    <!-- Table Container -->
    <div id="table"></div>
  </div>
  <script src="https://unpkg.com/gridjs/dist/gridjs.umd.js"></script>
  <script>
    const tableDiv = document.getElementById('table');

    const updateUrl = (prev, query) => {
      return prev + (prev.indexOf('?') >= 0 ? '&' : '?') + new URLSearchParams(query).toString();
    };

    const editableCellAttributes = (data, row, col) => {
      if (row) {
        return {contentEditable: 'true', 'data-element-id': row.cells[0].data};
      }
      else {
        return {};
      }
    };

    new gridjs.Grid({
      columns: [
        { id: 'id', name: 'id' },
        { id: 'co2', name: 'CO2', attributes: editableCellAttributes },

```



```

    { id: 'co2_corregido', name: 'CO2 Norm ', attributes: editableCellAttributes },
    { id: 'temp', name: 'Temp', attributes: editableCellAttributes },
    { id: 'hum', name: 'Hum', attributes: editableCellAttributes },
    { id: 'fecha', name: 'Fecha', attributes: editableCellAttributes },
    { id: 'lugar', name: 'Lugar', attributes: editableCellAttributes },
    { id: 'altura', name: 'Altura', attributes: editableCellAttributes },
    { id: 'presion', name: 'Presion', attributes: editableCellAttributes },
    { id: 'presion_nm', name: 'Presion nm', attributes: editableCellAttributes },
    { id: 'temp_ext', name: 'Temp ext', attributes: editableCellAttributes },
    { id: 'temp_ref', name: 'Temp ref', attributes: editableCellAttributes },
  ],
  server: {
    url: '/api/todos-los-datos',
    then: data => data,
    total: data => data.total,
  },
  search: {
    enabled: true,
    server: {
      url: (prev, search) => {
        return updateUrl(prev, {search});
      },
    },
  },
  sort: {
    enabled: true,
    multiColumn: true,
    server: {
      url: (prev, columns) => {
        const columnIds = ['id', 'co2', 'co2_corregido', 'temp', 'hum', 'fecha', 'lugar', 'altura',
'presion', 'presion_nm', 'temp_ext', 'temp_ref'];
        const sort = columns.map(col => (col.direction === 1 ? '+' : '-') + columnIds[col.index]);
        return updateUrl(prev, {sort});
      },
    },
  },
  pagination: {
    enabled: true,
    server: {
      url: (prev, page, limit) => {
        return updateUrl(prev, {start: page * limit, length: limit});
      },
    },
  },
}).render(tableDiv);

let savedValue;

tableDiv.addEventListener('focusin', ev => {
  if (ev.target.tagName === 'TD') {
    savedValue = ev.target.textContent;
  }

```

```

});

tableDiv.addEventListener('focusout', ev => {
  if (ev.target.tagName === 'TD') {
    if (savedValue !== ev.target.textContent) {
      fetch('/api/data', {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify({
          id: ev.target.dataset.elementId,
          [ev.target.dataset.columnId]: ev.target.textContent
        }),
      });
    }
    savedValue = undefined;
  }
});

tableDiv.addEventListener('keydown', ev => {
  if (ev.target.tagName === 'TD') {
    if (ev.key === 'Escape') {
      ev.target.textContent = savedValue;
      ev.target.blur();
    }
    else if (ev.key === 'Enter') {
      ev.preventDefault();
      ev.target.blur();
    }
  }
});
</script>
</body>
</html>

```