

Universidad Católica Argentina
Facultad de Ingeniería y ciencias
agrarias

PROTOCOLOS DE INTERNET

Trabajo Práctico N°4

Cursada: 1^{er} Cuatrimestre 2024

Grupo: Monti Facundo, Villanueva Mateo, Lamela Pablo

Profesor: Ingeniero Javier A. Ouret.

Integrantes:

N°	Nombre	Mail	Legajo
1	Facundo Monti	facundommontiuca@uca.edu.ar	44.759.207
2	Mateo Villanueva	mateovillanueva@uca.edu.ar	152253149
3	Pablo Lamela	lamelapablo@uca.edu.ar	45.748.517

Corrección:

Entrega 1	Devolución 1	Entrega 2	Nota



TP 2: MIB Browser

Índice:

Objetivos:.....	3
Consigna:.....	3
Procedimiento:	4



Objetivos:

- Observar el funcionamiento de las “suscripciones” de un cliente a un servidor.
- Comprender la transición de uno a otro y el funcionamiento del “broker”.

Consigna:

Instalar Mosquitto.

Configurar Mosquitto.

Arrancar y verificar el servidor Mosquitto.

Publicar datos desde otra terminal para ver como el cliente recibe las notificaciones.

Una vez que tengan funcionando lo anterior deberán simular la publicación de datos de sensores tomando lo que está ingresado en la base datos. Por practicidad y seguridad se puede hacer una aplicación cliente servidor entre los sensores y una base de datos, para que luego un publicador mqtt tome los datos desde ahí.

Luego un suscriptor toma los datos desde el publicador y los pasa a otra base datos local y propia del publicador para su análisis.

Esto permite que varios suscriptores usen los mismos datos.

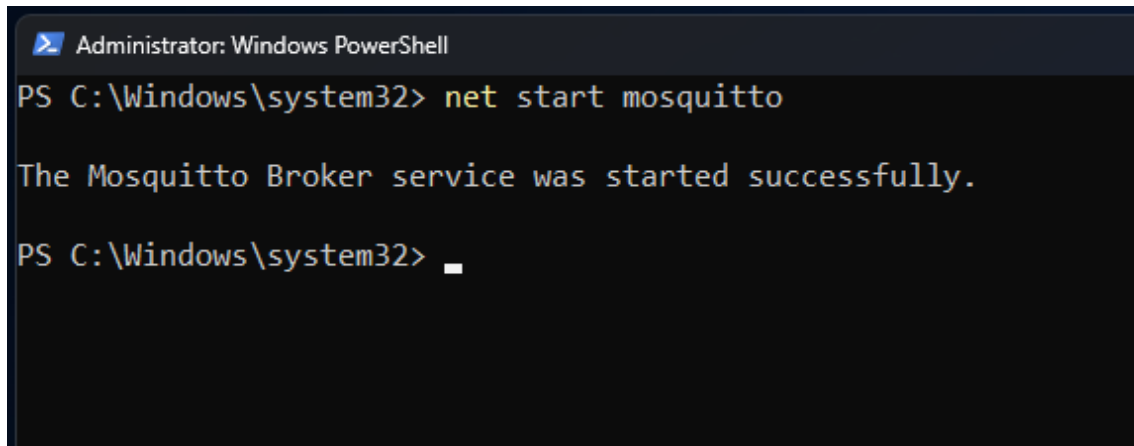
Realizar la modificación de las rutas RestAPI para mostrar los datos de los sensores suscriptos de esta forma.



Procedimiento:

Primero se realizó la instalación de Mosquitto según las indicaciones dadas. A continuación, se configuro el mismo con los comandos proporcionados.

Procedemos a conectar el servidor Mosquitto. Luego, suscribimos un cliente a la temática “temperaturas” y publicamos datos desde otra terminal en la misma. Finalmente vemos como se refleja en el lado del cliente.

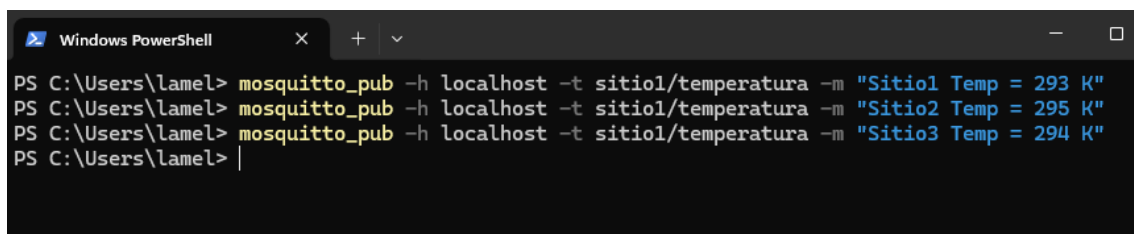


```
Administrator: Windows PowerShell
PS C:\Windows\system32> net start mosquitto

The Mosquitto Broker service was started successfully.

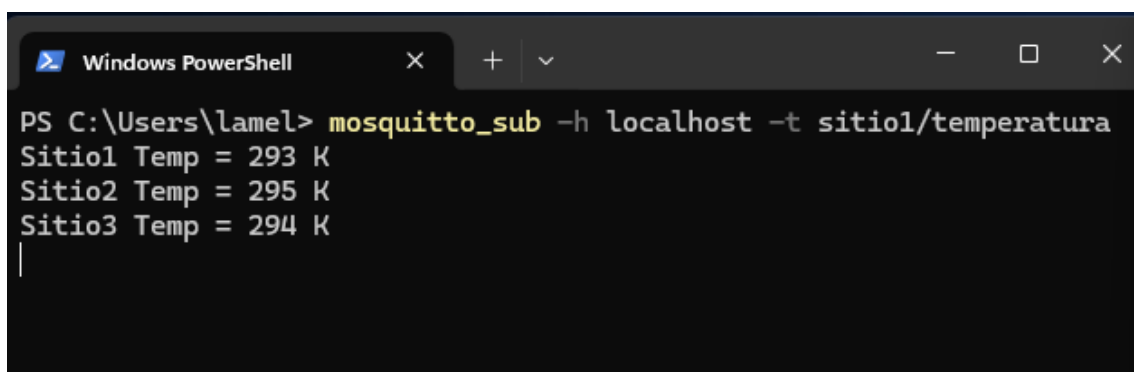
PS C:\Windows\system32> _
```

Imagen 1: Inicialización de Mosquitto



```
Windows PowerShell
PS C:\Users\lamel> mosquitto_pub -h localhost -t sitio1/temperatura -m "Sitio1 Temp = 293 K"
PS C:\Users\lamel> mosquitto_pub -h localhost -t sitio1/temperatura -m "Sitio2 Temp = 295 K"
PS C:\Users\lamel> mosquitto_pub -h localhost -t sitio1/temperatura -m "Sitio3 Temp = 294 K"
PS C:\Users\lamel> |
```

Imagen 2: Publicación del tópico “temperatura”



```
Windows PowerShell
PS C:\Users\lamel> mosquitto_sub -h localhost -t sitio1/temperatura
Sitio1 Temp = 293 K
Sitio2 Temp = 295 K
Sitio3 Temp = 294 K
|
```

Imagen 3: Suscripción y recepción del subscriptor



PARTE B:

En esta parte utilizamos el programa mqtt_pub_r1.py para publicar los datos que se encuentran en la base de datos utilizada en el TP anterior. Y realizamos algunas modificaciones al código de mqtt_sub_r1.py para que, además de recibir y mostrar por consola los datos recibidos al suscribirse al tópico "sensores/datos", estos se guarden en otra base de datos que llamamos "datos_sensores_sub".

```
mqtt_sub_r1.py X
mqtt_sub_r1.py > on_message
1  import logging
2  import paho.mqtt.client as mqtt
3  import json
4  import sqlite3
5
6  # Configuración del logging para depuración
7  logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(levelname)s: %(message)s')
8
9  MQTT_BROKER = "localhost"
10 MQTT_PORT = 1883
11 MQTT_TOPIC = "sensores/datos"
12
13 def on_connect(client, userdata, flags, rc):
14     if rc == 0:
15         logging.info("Conectado al broker MQTT")
16         client.subscribe(MQTT_TOPIC)
17     else:
18         logging.error(f"Conexión fallida con código de resultado: {rc}")
19
20 def on_message(client, userdata, msg):
21     logging.debug(f"Mensaje recibido en tópico {msg.topic}: {msg.payload.decode()}")
22     data = json.loads(msg.payload.decode())
23     insertar_registro(data)
```

Imagen 4: Código mqtt_sub_r1.py (parte A)



```
25 def create_table():
26     conn = sqlite3.connect('datos_sensores_sub.db')
27     cursor = conn.cursor()
28     cursor.execute('''
29         CREATE TABLE IF NOT EXISTS lectura_sensores (
30             id INTEGER PRIMARY KEY,
31             co2 REAL,
32             temp REAL,
33             hum REAL,
34             fecha TEXT,
35             lugar TEXT,
36             altura REAL,
37             presion REAL,
38             presion_nm REAL,
39             temp_ext REAL
40         )''')
41     conn.commit()
42     conn.close()
43
44 def insertar_registro(data):
45     co2 = data.get('co2')
46     temp = data.get('temp')
47     hum = data.get('hum')
48     fecha = data.get('fecha')
49     lugar = data.get('lugar')
50     altura = data.get('altura')
51     presion = data.get('presion')
52     presion_nm = data.get('presion_nm')
53     temp_ext = data.get('temp_ext')
54
55     conn = sqlite3.connect('datos_sensores_sub.db')
56     cursor = conn.cursor()
57     cursor.execute('''
58         INSERT INTO lectura_sensores (co2, temp, hum, fecha, lugar, altura, presion, presion_nm, temp_ext)
59         VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
60     ''', (co2, temp, hum, fecha, lugar, altura, presion, presion_nm, temp_ext))
61     conn.commit()
62     conn.close()
```

Imagen 5: Código mqtt_sub_r1.py (parte B)

```
64 create_table()
65 client = mqtt.Client()
66 client.on_connect = on_connect
67 client.on_message = on_message
68
69 try:
70     client.connect(MQTT_BROKER, MQTT_PORT, 60)
71     logging.info("Conectando al broker MQTT")
72     client.loop_forever()
73 except Exception as e:
74     logging.error(f"Error al conectar al broker MQTT: {e}")
75
```

Imagen 6: Código mqtt_sub_r1.py (parte C)



DB Browser for SQLite - C:\Users\lamel\CloudDrive\UCA\Tercer año\Protocolos de Internet\TP4\datos_sensores_sub.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragma Execute SQL

Table: lectura_sensores Filter in any column

	id	co2	temp	hum	fecha	lugar	altura	presion	presion_nm	temp_ext
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	310.09	17.01	50.54	14-Jun-2024 (23:40:27.255712)	caballito	40.0	1013.0	1013.0	15.23
2	2	530.24	17.44	73.53	14-Jun-2024 (23:40:32.267658)	caballito	40.0	1013.0	1013.0	15.23
3	3	692.99	25.13	58.0	14-Jun-2024 (23:40:37.277885)	caballito	40.0	1013.0	1013.0	15.23
4	4	678.62	16.53	65.29	14-Jun-2024 (23:40:42.285201)	caballito	40.0	1013.0	1013.0	15.23
5	5	100.0	18.0	9.0	2024-06-21T05:43	Recoleta	10.0	1011.0	1011.0	2.0
6	6	10.0	1.0	90.0	2024-06-15T01:43	Almagro	10.0	2.0	30.0	4.0
7	7	310.09	17.01	50.54	14-Jun-2024 (23:40:27.255712)	caballito	40.0	1013.0	1013.0	15.23
8	8	530.24	17.44	73.53	14-Jun-2024 (23:40:32.267658)	caballito	40.0	1013.0	1013.0	15.23

Imagen 7: Base de datos creada (datos_sensores_sub.db)