MQTT

Presentado por:
Kevin Gómez Burgos – Código: 20192183784
Aura Cristina Lopez Carabali – Código: 20202192207

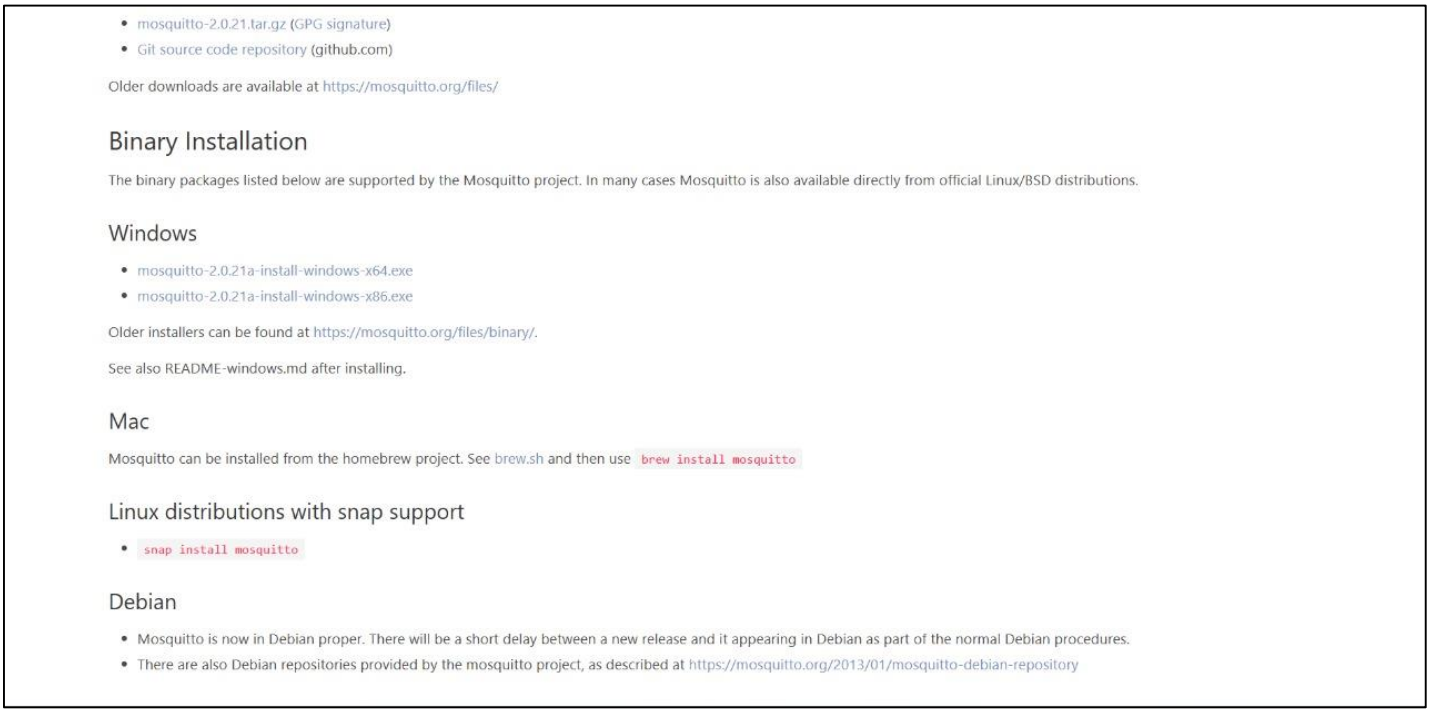
Docente:
Juan Castro

Universidad Surcolombiana
Facultad de ingeniería
Neiva Huila
2025

TABLA DE CONTENDIDO

Instalación

- Se ingresa a la página oficial https://mqtt.org/, descargamos el instalador de mosquitto https://bit.ly/joshua-electronics-mosquitto el cual es un .EXE.



- mosquitto-2.0.21.tar.gz (GPG signature)
- Git source code repository (github.com)

Older downloads are available at https://mosquitto.org/files/

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- mosquitto-2.0.21a-install-windows-x64.exe
- mosquitto-2.0.21a-install-windows-x86.exe

Older installers can be found at https://mosquitto.org/files/binary/.

See also README-windows.md after installing.

Mac

Mosquitto can be installed from the homebrew project. See brew.sh and then use `brew install mosquitto`

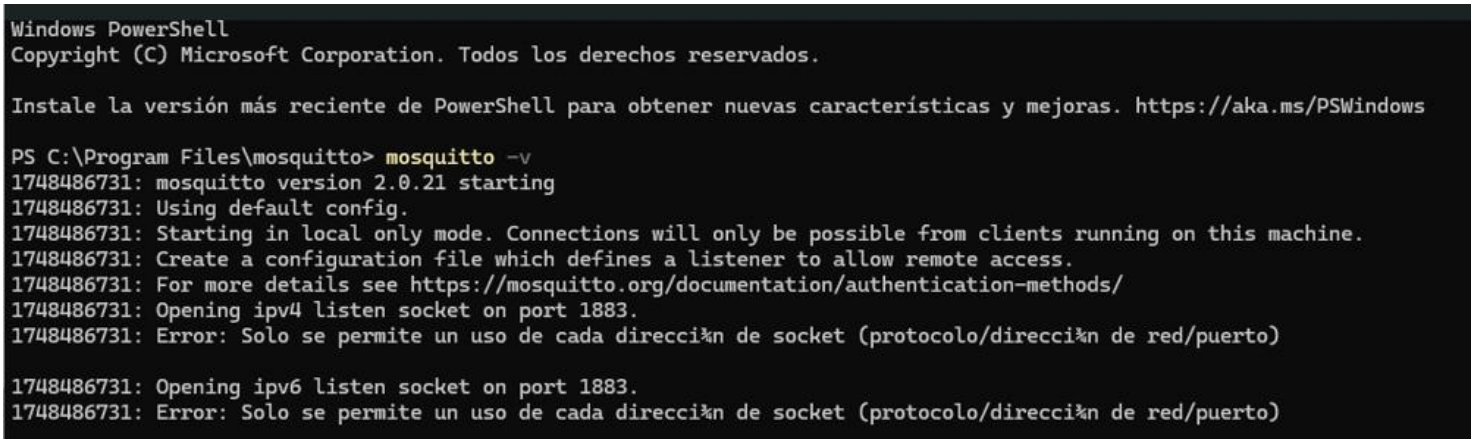
Linux distributions with snap support

- `snap install mosquitto`

Debian

- Mosquitto is now in Debian proper. There will be a short delay between a new release and it appearing in Debian as part of the normal Debian procedures.
- There are also Debian repositories provided by the mosquitto project, as described at https://mosquitto.org/2013/01/mosquitto-debian-repository

- Seguimos todos los pasos de instalación y se finaliza, una vez finalizado, abrimos nuestra terminal y revisamos la versión de nuestro mosquitto, con el comando: mosquitto -v



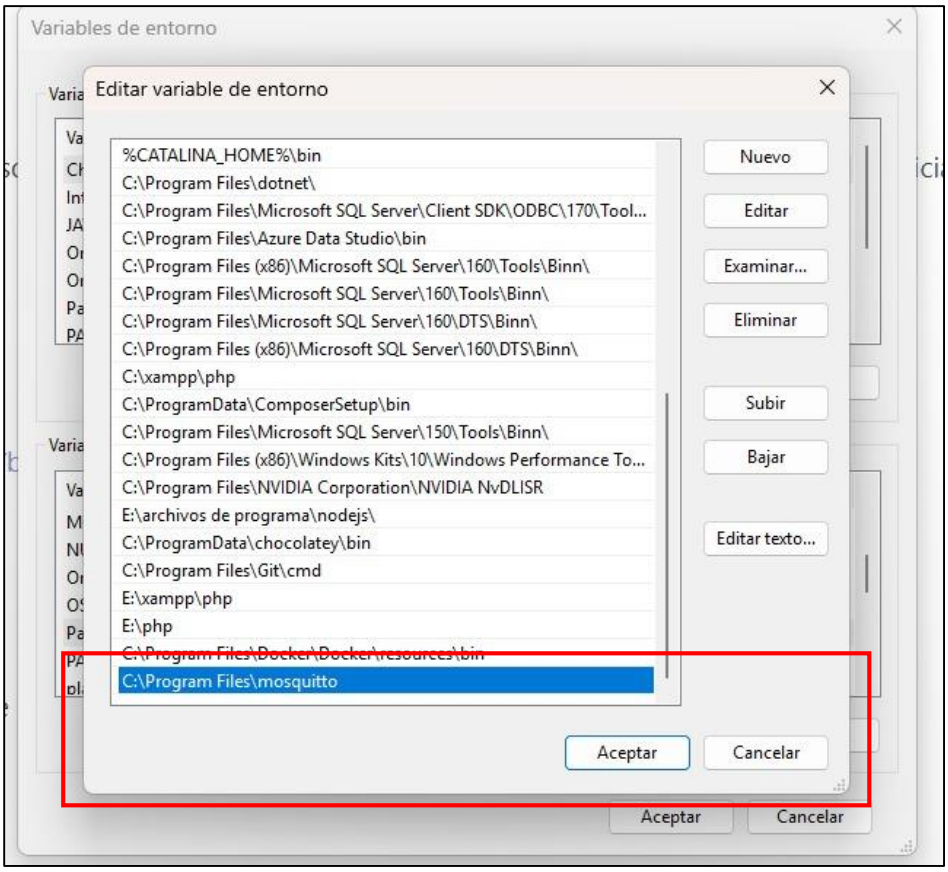
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Program Files\mosquitto> mosquitto -v
1748486731: mosquitto version 2.0.21 starting
1748486731: Using default config.
1748486731: Starting in local only mode. Connections will only be possible from clients running on this machine.
1748486731: Create a configuration file which defines a listener to allow remote access.
1748486731: For more details see https://mosquitto.org/documentation/authentication-methods/
1748486731: Opening ipv4 listen socket on port 1883.
1748486731: Error: Solo se permite un uso de cada direcci¾n de socket (protocolo/direcci¾n de red/puerto)

1748486731: Opening ipv6 listen socket on port 1883.
1748486731: Error: Solo se permite un uso de cada direcci¾n de socket (protocolo/direcci¾n de red/puerto)
```

- Una vez se haya hecho la instalación y verificado la versión, ajustamos las variables de entorno y agregamos mosquitto.



Script para enviar información al backend

- Script para simular el backend con flask, es decir para la información a la API.

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

app = Flask(__name__)

@app.route('/api/mensaje', methods=['POST'])
def recibir_mensaje():
    data = request.json
    print("📩 Mensaje recibido en el backend:", data)
    return jsonify({"status": "OK", "mensaje": data}), 200

if __name__ == '__main__':
    app.run(debug=True, port=5000)
```

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

BACKEND_URL = "http://localhost:5000/api/mensaje"  # Dirección del backend local

def on_connect(client, userdata, flags, rc):
    print("✅ Conectado a MQTT Broker")
    client.subscribe("sensor/temperatura")

def on_message(client, userdata, msg):
    payload = msg.payload.decode()
    print(f"📡 Mensaje MQTT: {msg.topic} -> {payload}")

    # Enviar al backend
    data = {
        "topic": msg.topic,
        "valor": payload
    }

    try:
        response = requests.post(BACKEND_URL, json=data)
        print("📤 Enviado al backend:", response.status_code)
    except Exception as e:
        print("❌ Error enviando al backend:", e)

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

client.connect("localhost", 1883, 60)
client.loop_forever()
```

Sub (subscribe)

Un **subscriber** (suscriptor) se registra para recibir automáticamente todos los mensajes publicados en ese tópico.

- Desde consola mediante sub, con el siguiente comando, enviamos información:

```
PS C:\Program Files\mosquitto> mosquitto_pub -h localhost -t sensor/temperatura -m "{'temperatura': 45.6}"
PS C:\Program Files\mosquitto>
```

pub (publish)

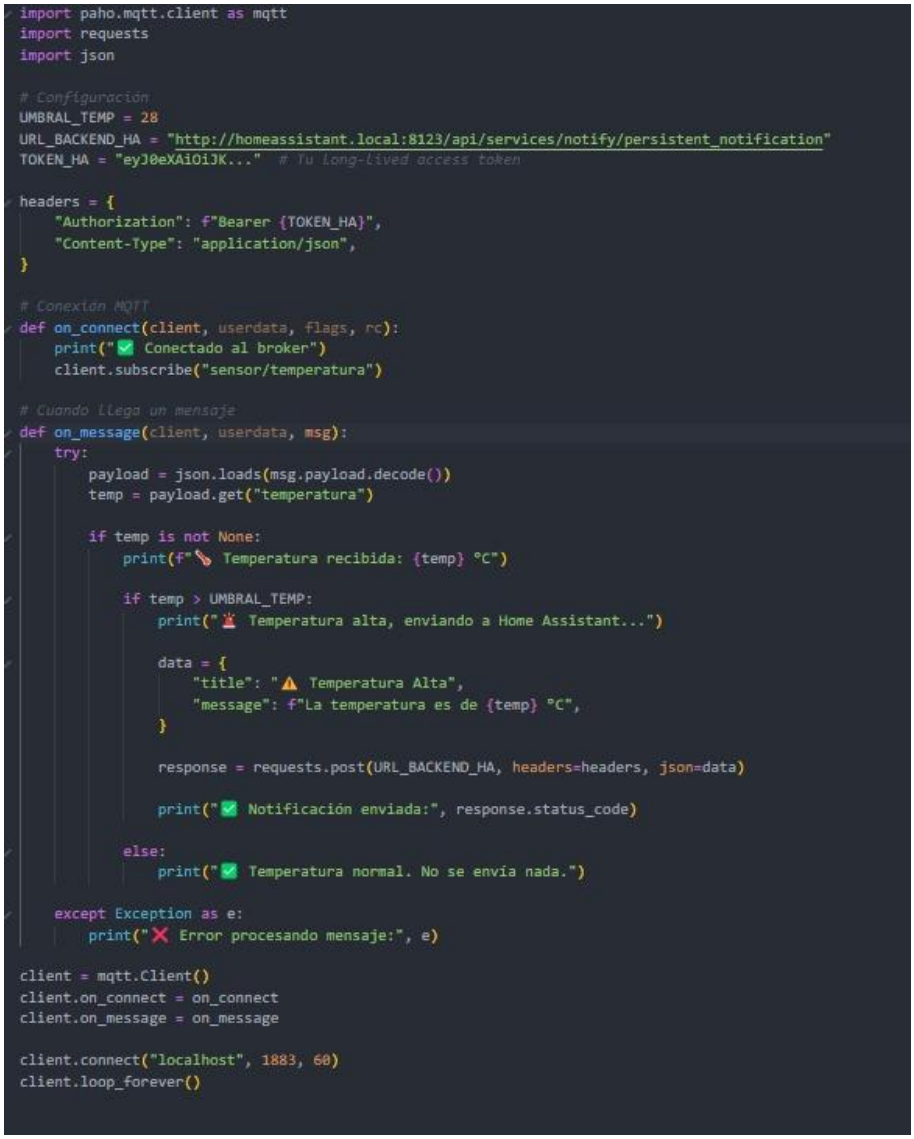
Un **publisher** (publicador) emite datos para que cualquier interesado pueda recibirlos.

- Así aparece en consola el mensaje recibido.

```
PS C:\Program Files\mosquitto> mosquitto_sub -h localhost -t sensor/temperatura
{'temperatura': 45.6}
```

Homeassistant

- Es la lógica creada para que cuando se reciba una información este logre encender una alerta o una acción. Solo envía el mensaje cuando sobrepasa un parámetro, si está por encima de cierta temperatura o por debajo, solo lo envía en estos casos, para no estar enviando datos de forma recurrente.



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

# Configuración
UMBRAL_TEMP = 28
URL_BACKEND_HA = "http://homeassistant.local:8123/api/services/notify/persistent_notification"
TOKEN_HA = "eyJ0eXAiOiJK..."  # Tu long-lived access token

headers = {
    "Authorization": f"Bearer {TOKEN_HA}",
    "Content-Type": "application/json",
}

# Conexión MQTT
def on_connect(client, userdata, flags, rc):
    print("✅ Conectado al broker")
    client.subscribe("sensor/temperatura")

# Cuando llega un mensaje
def on_message(client, userdata, msg):
    try:
        payload = json.loads(msg.payload.decode())
        temp = payload.get("temperatura")

        if temp is not None:
            print(f"🌡️ Temperatura recibida: {temp} °C")

            if temp > UMBRAL_TEMP:
                print("🚨 Temperatura alta, enviando a Home Assistant...")

                data = {
                    "title": "⚠️ Temperatura Alta",
                    "message": f"La temperatura es de {temp} °C",
                }

                response = requests.post(URL_BACKEND_HA, headers=headers, json=data)

                print("✅ Notificación enviada:", response.status_code)

            else:
                print("✅ Temperatura normal. No se envía nada.")

    except Exception as e:
        print("❌ Error procesando mensaje:", e)

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

client.connect("localhost", 1883, 60)
client.loop_forever()
```