



Inteligencia Artificial

Documentación de Instalación y Configuración en la Raspberry PI 3 Model B

Realizado Por:

Duban Cardozo - 20222209187

Nicolas Calderon - 20202192323

UNIVERSIDAD SURCOLOMBIANA

NEIVA – HUILA

2025

Vigilada Mineducación



Proyecto: Entorno de Desarrollo IoT + Backend

Este documento proporciona una guía paso a paso para la instalación y configuración de un entorno de desarrollo completo para proyectos de Internet de las Cosas (IoT) con integración de servicios backend. La configuración está orientada a sistemas basados en Linux (como Raspberry Pi) y combina diversas tecnologías clave:

- Mosquitto como broker MQTT para la comunicación entre dispositivos.
- MongoDB como base de datos NoSQL, ideal para datos no estructurados o en tiempo real.
- Flask como framework web ligero para la creación de APIs REST.
- Python para pruebas rápidas de scripts y automatizaciones.

Esta documentación está diseñada para facilitar el despliegue inicial del sistema, desde la instalación de servicios hasta comandos prácticos para verificar su funcionamiento. Es ideal para desarrolladores, estudiantes o técnicos que busquen implementar soluciones conectadas y modulares.



1. Mosquitto (Broker MQTT)

Instalación del broker

```
1 - wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
2 - sudo apt-key add mosquitto-repo.gpg.key
3 - sudo apt-get install mosquitto
4
```

Habilitar e iniciar el servicio

```
1 sudo systemctl enable mosquitto
2 sudo systemctl start mosquitto
```

Instalación del cliente

```
1 sudo apt install -y mosquitto-clients
```

Comandos útiles (Ejemplos)

```
1 • Escuchar (subscribirse a un tópico):
2 mosquitto_sub -h localhost -t test/topic
3
4 • Publicar mensajes:
5 mosquitto_pub -h 192.168.221.183 -t usco/topic -m "Hola desde Raspberry Pi"
6 mosquitto_pub -h 192.168.221.183 -t usco/topic -m '{"bombilla": "ON"}'
```



2. MongoDB (versión ARM para Raspberry Pi)

Instalación manual (v5.0.6 ARM64) – Repositorio No oficial



```
1 - wget -q -O MongoARM64.zip https://github.com/Inqnuam/  
2 MongoDB-ARMv8/releases/download/v5.0.6/MongoDBv5.0.6_ARM64.zip  
3 - unzip MongoARM64.zip  
4 - rm MongoARM64.zip
```

Crear archivo de configuración



```
1 //Crear archivo de configuración  
2 en la siguiente ruta  
3 /etc/mosquitto/mongod.conf  
4  
5 //Crear archivo con nano  
6 sudo nano /etc/mosquitto/mongod.conf  
7  
8 //Contenido del archivo mongod.conf  
9 storage:  
10   dbPath: /var/lib/mongodb  
11   systemLog:  
12     destination: file  
13     path: /var/log/mongodb/mongod.log  
14     logAppend: true  
15   net:  
16     bindIp: 127.0.0.1  
17     port: 27017  
18   processManagement:  
19     fork: false  
20
```



Crear servicio systemd

```
1 //MongoDB archivo de servicio con nano
2 sudo nano /etc/systemd/system/mongod.service
3
4 //Contenido:
5 [Unit]
6 Description=MongoDB Database Server
7 After=network.target
8
9 [Service]
10 User=mongodb
11 Group=mongodb
12 ExecStart=/usr/bin/mongod --config /etc/mongod.conf
13 PIDFile=/var/run/mongodb/mongod.pid
14 Restart=on-failure
15
16 [Install]
17 WantedBy=multi-user.target
```

Crear carpetas y asignar permisos

```
1 sudo mkdir -p /var/log/mongodb /var/lib/mongodb
2 sudo chown -R root:root /var/log/mongodb /var/lib/mongodb
3 // Activar MongoDB
4 sudo systemctl daemon-reload
5 sudo systemctl enable mongod
6 sudo systemctl start mongod
```



3. MongoDB Shell (mongosh)

Descargar desde:

<https://www.mongodb.com/try/download/shell>

Pasos para instalar

```
1
2 # Descomprimir archivo descargado
3 tar -xvzf nombre_archivo.tgz
4 cd nombre_carpeta
5
6 # Mover el ejecutable
7 sudo mv bin/mongosh /usr/local/bin/
8
9 # Ejecutar mongosh
10 mongosh
11
```

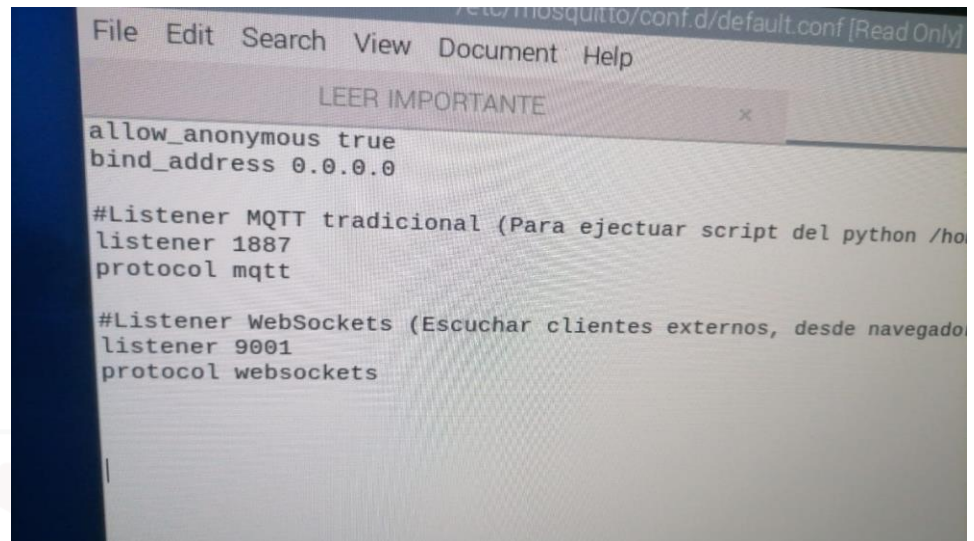
4. Configuración de Mosquitto (mqtt y websocket)

Crear un archivo de configuración de Mosquitto en la carpeta raíz de instalación, llamado mosquitto.conf e incluye lo siguiente

```
1 include_dir /etc/mosquitto/conf.d
```



Crear una carpeta llamada conf.d y dentro de ella crear un archivo default.conf con el siguiente contenido

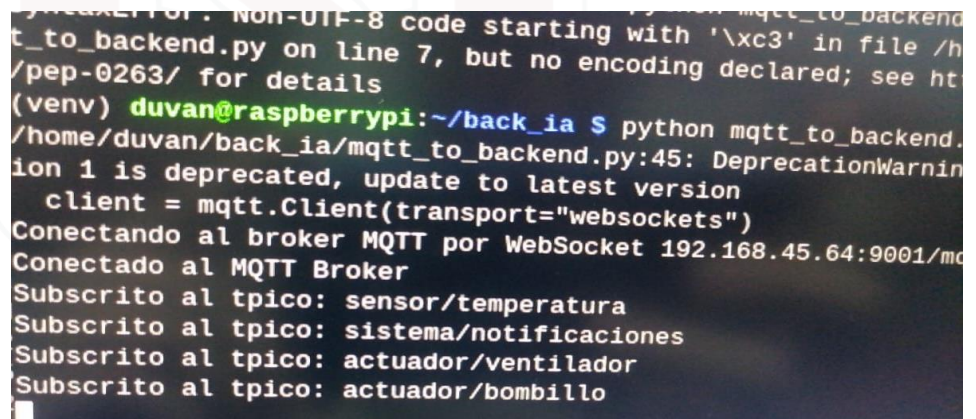


```
File Edit Search View Document Help
LEER IMPORTANTE x
allow_anonymous true
bind_address 0.0.0.0

#Listener MQTT tradicional (Para ejecutar script del python /ho
listener 1887
protocol mqtt

#Listener WebSockets (Escuchar clientes externos, desde navegado
listener 9001
protocol websockets
```

5. Ejecución de Mosquitto desde un servicio del backend (repositorio proporcionado por otro compañero)



```
FileNotFoundError: Non-UTF-8 code starting with '\xc3' in file /home/duvan/back_ia/mqtt_to_backend.py on line 7, but no encoding declared; see https://pep-0263/ for details
(venv) duvan@raspberrypi:~/back_ia $ python mqtt_to_backend.py
/home/duvan/back_ia/mqtt_to_backend.py:45: DeprecationWarning: io.BytesIO() is deprecated, update to latest version
  client = mqtt.Client(transport="websockets")
Conectando al broker MQTT por WebSocket 192.168.45.64:9001/mosquitto
Conectado al MQTT Broker
Subscrito al tpico: sensor/temperatura
Subscrito al tpico: sistema/notificaciones
Subscrito al tpico: actuador/ventilador
Subscrito al tpico: actuador/bombillo
```



6. Servidor

- Descargar el repositorio: https://github.com/f3lipe22/back_ia
- Ubicarse en la raíz del proyecto
- Crear un entorno virtual

```
1 python -m venv venv
```

- Activar el entorno Virtual

```
1 source venv/bin/activate
```

- Instalar librerías necesarias

```
1 pip install -r requirements.txt
```

- Ejecutar servidor backend

```
1 python app.py
```




7. Servidor Corriendo (La explicación detallada del funcionamiento del backend la suministra los compañeros que fueron asignados)

```
van@raspberrypi ~$ duvan@raspberrypi ~$ duvan@raspberrypi ~$
duvan@raspberrypi ~/back.js
File Edit Tabs Help
{"temperature":25.50,"humidity":50.80}
2025-05-28 21:20:06,534 - mqtt_controller - INFO - Datos de temperatura no procesados: 25
2025-05-28 21:20:06,538 - werkzeug - INFO - 127.0.0.1 - - [28/May/2025 21:20:06] "POST /api/mensaje HTTP/1.1" 201 -
2025-05-28 21:20:11,511 - mqtt_controller - WARNING - No se pudo convertir el valor a float: {"temperatu
":25.50,"humidity":51.00}
2025-05-28 21:20:11,517 - mqtt_controller - INFO - Mensaje MQTT recibido y guardado: sensor/temperatura
{"temperature":25.50,"humidity":51.00}
2025-05-28 21:20:11,528 - mqtt_controller - INFO - Datos de temperatura no procesados: 25
2025-05-28 21:20:11,532 - werkzeug - INFO - 127.0.0.1 - - [28/May/2025 21:20:11] "POST /api/mensaje HTTP/1.1" 201 -
2025-05-28 21:20:21,514 - mqtt_controller - WARNING - No se pudo convertir el valor a float: {"temperatu
":25.60,"humidity":79.20}
2025-05-28 21:20:21,520 - mqtt_controller - INFO - Mensaje MQTT recibido y guardado: sensor/temperatura
{"temperature":25.60,"humidity":79.20}
2025-05-28 21:20:21,531 - mqtt_controller - INFO - Datos de temperatura no procesados: 26
2025-05-28 21:20:21,535 - werkzeug - INFO - 127.0.0.1 - - [28/May/2025 21:20:21] "POST /api/mensaje HTTP/1.1" 201 -
2025-05-28 21:20:21,562 - mqtt_controller - WARNING - No se pudo convertir el valor a float: {'ventilador
': 'ON'}
2025-05-28 21:20:21,566 - mqtt_controller - INFO - Mensaje MQTT recibido y guardado: sensor/temperatura -
mo {'ventilador': 'ON'}
2025-05-28 21:20:21,573 - mqtt_controller - INFO - Datos de temperatura no procesados: 27
2025-05-28 21:20:21,575 - werkzeug - INFO - 127.0.0.1 - - [28/May/2025 21:20:21] "POST /api/mensaje HTTP/1.1" 201 -
```