Raspberry Pi

Samuel Alexander Perdomo Fajardo (20222208795)

Kevin Enrique Gómez Burgos (20192183784)

Neiva

31 Mayo de 20232

Universidad Colombiana

**Introducción**

Este documento describe la configuración e implementación de una base de datos MongoDB en un dispositivo Raspberry Pi, destinada a recopilar y gestionar datos ambientales provenientes de un sistema de monitoreo para un invernadero. La base de datos fue configurada para permitir conexiones desde cualquier dispositivo en la misma red local.

**Tabla de Contenido**

**1.**objetivo 1 proyecto (setup environment Raspberry Pi y Mosquitto)

**2.**Objetivo 2 proyecto (instalación base de datos para el backend)

## 1. Objetivo 1 proyecto (setup environment Raspberry Pi y Mosquitto)

Proceso de instalación del sistema operativo Raspberry Pi OS en una Raspberry Pi, la instalación del broker Mosquitto MQTT a partir del repositorio oficial y la validación de su funcionamiento mediante una prueba de publicación y suscripción (pub/sub). Esta implementación permite establecer una comunicación basada en el protocolo MQTT, útil en sistemas de IoT y automatización.

## 2. Requisitos

### Hardware

* Raspberry Pi (modelo 3 o superior)
* Tarjeta microSD (mínimo 16 GB, clase 10)
* Fuente de alimentación adecuada
* Monitor, teclado y mouse (si no se configura acceso remoto)
* Conexión a Internet

### Software

* Raspberry Pi Imager (descargable desde raspberrypi.com)
* Sistema operativo: Raspberry Pi OS (32-bit)
* Terminal (acceso por consola o SSH)

## 3. Instalación de Raspberry Pi OS

### 3.1 Descarga de Raspberry Pi Imager

1. Visitar:<https://www.raspberrypi.com/software>
2. Descargar el instalador según el sistema operativo del computador personal (Windows, macOS, Linux).
3. Instalar y ejecutar el programa.

### 3.2 Preparación de la tarjeta SD

1. Insertar la tarjeta microSD al PC.
2. En Raspberry Pi Imager:  
   * Seleccionar Raspberry Pi OS (32-bit)
   * Seleccionar el dispositivo (tarjeta SD)
   * (Opcional) En el ícono de configuración:  
     + Establecer nombre de host
     + Configurar usuario y contraseña
     + Activar SSH
     + Conectar Wi-Fi automáticamente
3. Click en Write y esperar a que finalice la escritura.

### 3.3 Primer arranque

1. Insertar la tarjeta en Raspberry Pi.
2. Conectar el monitor, teclado, mouse y la fuente de energía.
3. Completar la configuración inicial del sistema (zona horaria, usuario, etc.)

## 4. Actualización del Sistema

Es fundamental mantener el sistema actualizado antes de instalar cualquier software adicional:

| sudo apt update && sudo apt upgrade -y |
| --- |

## 5. Instalación del Broker Mosquitto MQTT

### 5.1 Agregar repositorio oficial

1. **Descargar y agregar la llave del repositorio:**

| sudo apt install wget -y wget https://mosquitto.org/files/repos/mosquitto-repo.gpg.key sudo apt-key add mosquitto-repo.gpg.key |
| --- |

1. **Añadir el repositorio (para Raspberry Pi OS Bullseye):**

| cd /etc/apt/sources.list.d/ sudo wget https://mosquitto.org/files/repos/mosquitto-bullseye.list |
| --- |

*Nota:* Cambiar bullseye por buster o bookworm si corresponde a la versión usada.

1. **Actualizar lista de paquetes:**

| sudo apt update |
| --- |

### 5.2 Instalar Mosquitto y clientes

| sudo apt install mosquitto mosquitto-clients -y |
| --- |

### 5.3 Habilitar y arrancar el servicio

| **sudo systemctl enable mosquitto sudo systemctl start mosquitto sudo systemctl status mosquitto** |
| --- |

## 6. Prueba de Funcionamiento (Pub/Sub)

Se realizará una prueba básica de suscripción y publicación local usando la terminal.

### 6.1 Suscriptor (terminal 1)

| mosquitto\_sub -h localhost -t test/topic |
| --- |

### 6.2 Publicador (terminal 2)

| mosquitto\_pub -h localhost -t test/topic -m "Hola MQTT desde Raspberry Pi" |
| --- |

### Resultado esperado:

El mensaje "Hola MQTT desde Raspberry Pi" debe aparecer en la primera terminal (suscriptor), validando el correcto funcionamiento del broker.

## 7. Integración con Home Assistant (WebSocket)

Para permitir que Home Assistant se conecte al broker Mosquitto mediante WebSocket, es necesario realizar algunos ajustes en el archivo de configuración de Mosquitto.

### 10.1 Habilitar soporte para WebSockets

1. Abrir (o crear si no existe) el archivo principal de configuración de Mosquit**to:**

**bash**

| **sudo nano /etc/mosquitto/mosquitto.conf** |
| --- |

1. **Agregar las siguientes líneas al final del archivo:**

**yaml**

| **# Default listener (puerto MQTT estándar) listener 1883 protocol mqtt  # Listener para WebSocket (para Home Assistant o aplicaciones web) listener 9001 protocol websockets** |
| --- |

**Nota: El puerto 9001 es común para WebSocket, pero puedes cambiarlo si ya está en uso.**

### 7.3 Reiniciar el servicio

| **sudo systemctl restart mosquitto** |
| --- |

## 8. Conclusiones

Se ha instalado exitosamente el sistema operativo Raspberry Pi OS y el broker MQTT Mosquitto desde su repositorio oficial. Se validó su correcto funcionamiento mediante una prueba básica de publicación y suscripción. Este entorno está listo para aplicaciones IoT o sistemas distribuidos que utilicen el protocolo MQTT.

**1. objetivo 2 proyecto (instalación base de datos para el backend)**

* Implementar una base de datos MongoDB en una Raspberry Pi.
* Crear una base de datos y colección para almacenar información relevante sobre el entorno de una planta.
* Configurar el servidor para permitir conexiones externas desde la red local

1. **Entorno De Desarrollo**

Base de datos

* Dispositivo: Raspberry Pi
* Sistema Operativo: Raspberry Pi OS
* Motor de Base de Datos: MongoDB
* Puerto habilitado: 1878
* Acceso remoto: Permitido para todas las IPs de la red (0.0.0.0)

1. **Base De Datos**

***3.1 Instalación de MongoDB***

Se instaló MongoDB directamente desde la terminal de la Raspberry Pi usando el gestor de paquetes correspondiente:

| *sudo apt update sudo apt install -y mongodb* |
| --- |

Una vez instalado, se habilitó el servicio:

| sudo systemctl enable mongodb sudo systemctl start mongodb |
| --- |

***3.2 Configuración de Red de MongoDB***

Para permitir conexiones desde cualquier dirección IP de la red local, se modificó el archivo de configuración de MongoDB ubicado en:

| */etc/mongodb.conf* |
| --- |

Se cambió la dirección de enlace (bindIp) a:

| *bindIp: 0.0.0.0 port: 1878* |
| --- |

Luego se reinició el servicio para aplicar los cambios:

| *sudo systemctl restart mongodb* |
| --- |

***3.3 Creación de la Base de Datos y Colección***

Una vez iniciado el servicio de MongoDB, se accedió al shell de MongoDB desde la terminal:

| *mongo --port 1878* |
| --- |

Dentro del shell, se creó la base de datos denominada invernadero y una colección llamada data. Esta colección está diseñada para almacenar parámetros relacionados con el entorno de las plantas. Los datos registrados incluyen:

* id: Identificador único de la planta o sensor.
* temperatura: Medición de la temperatura ambiente.
* humedad: Nivel de humedad relativo.
* tamaño: Tamaño o altura de la planta.
* Estado: SI la polenta está saludable o no

Así está construida la tabla



| db.createCollection("data", {  validator: {  $jsonSchema: {  bsonType: "object",  required: ["id", "temperatura", "humedad", "tamaño"],  properties: {  id: {  bsonType: "string",  description: "Debe ser una cadena y es obligatorio"  },  temperatura: {  bsonType: "double",  description: "Debe ser un número decimal (double) y es obligatorio"  },  humedad: {  bsonType: "double",  description: "Debe ser un número decimal (double) y es obligatorio"  },  tamaño: {  bsonType: "double",  description: "Debe ser un número decimal (double) y es obligatorio"  },  luz: {  bsonType: "string",  description: "Debe ser una cadena (opcional)"  },  estado: {  bsonType: "string",  description: "Estado de la planta (opcional)"  },  timestamp: {  bsonType: "date",  description: "Fecha de registro del dato (opcional)"  }  }  }  } }) |
| --- |

1. **Conclusión**

Con esta configuración, el sistema está listo para recibir datos desde sensores o aplicaciones externas a través de la red. MongoDB en Raspberry Pi ofrece una solución ligera y eficaz para el almacenamiento de información crítica del entorno de cultivo, siendo accesible para cualquier dispositivo conectado en la red local

1. **Recomendaciones Futuras**

* Implementar autenticación y encriptación para mejorar la seguridad de la base de datos