

Proyecto Curso Raspberry IoT

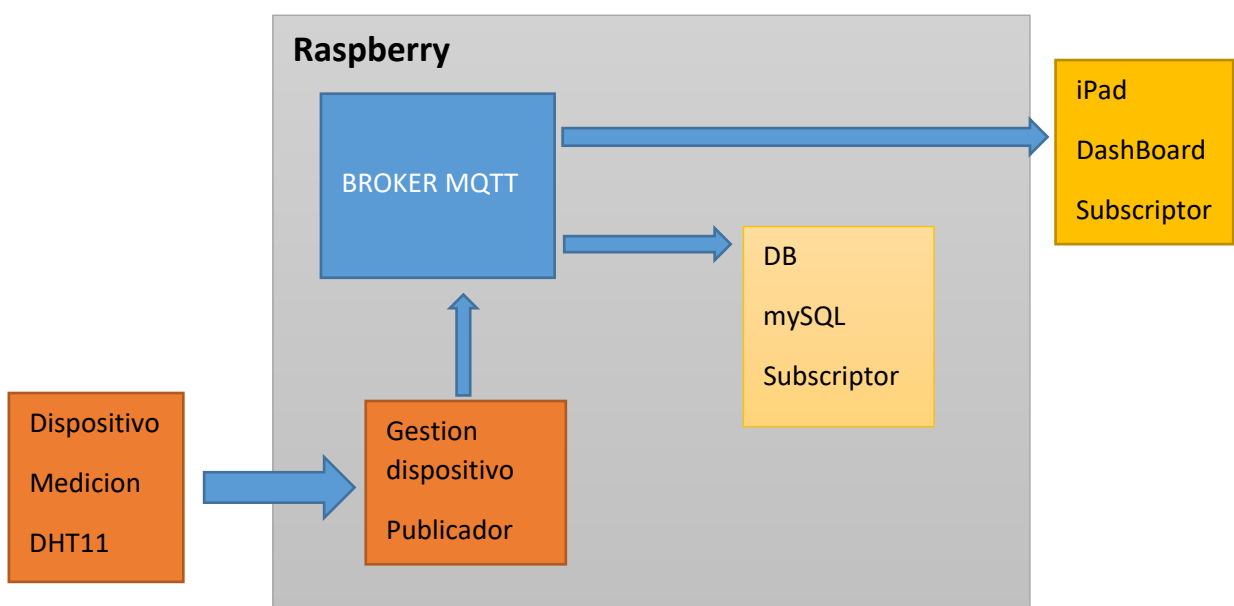
1. Objetivos

- Crear un Broker con protocolo MQTT, para la gestion de informacion de dispositivos de control interior para viviendas.
- Trasladar la informacion recogida a una base de datos, para su posterior análisis y gestion.

2. Requerimientos

- Unidad raspberry pi 3.
- Base de datos instalable en raspberry. (Mariadb)
- Dispositivo detector de temperatura y humedad. (DHT11)
- Visualizacion de la informacion en un dispositivo mòbil, iPad o Dashboard.

3. Diagrama de bloques de funcionamiento.



Programa control DHT11

Import time

Import RPi.GPIO as GPIO

Import Adafruit_DHT

Sensor = Adafruit_DHT.DHT11

Pin = 16

While 1:

Humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

If humidity is NONE and temperature is NONE:

Print(" ha fallado la lectura")

else:

print("temperatura: ", temperature) #enviar dato al broker

print("humedad: ",humidity) #enviar dato al broker

para saber la hora

import la libreria time

y utilizar la funcion de python time.strftime("%H:%M:%S")

instruccion para arrancar mosquito desde la consola abierta escribir mosquitto

Requisitos para la programacion en python

- Instalar paquete de MQTT. pip3 install paho-mqtt
- Instalar paquete de mariadb. Pip3 install mariadb
-

PROGRAMA EN PYTHON

```
IMPORT MARIADB
IMPORT SYS
IMPORT TIME
IMPORT DATETIME
IMPORT SSL
IMPORT PAHO.MQTT.CLIENT AS MQTT
IMPORT PAHO.MQTT.PUBLISH AS PUBLISH

# INSTRUCCION PARA ACCEDER A LA BADE DE DATOS MARIADB
# MYSQL -U DATOS_IOTUSER -P PASSWORD C1966

# INSTRUCCION PARA ARRANCAR MOSQUITO DESDE LA CONSOLA ABIERTA ESCRIBIR
MOSQUITTO

AHORA = DATETIME.DATETIME.NOW()

MQTT_SERVER = "LOCALHOST"
MQTT_PATH = "CASA/SALA1/+"
TOPICO_TEMP = "CASA/SALA1/TEMPERATURA"
TOPICO_HUME = "CASA/SALA1/HUMEDAD"

TEMPERATURA = 0
HUMEDAD = 0
FECHA = ""

#FUNCION PARA ESCRIBIR DATOS EN LOS CAMPOS DE LA BASE DE DATOS
SELECCIONADA
DEF ESCRIBIR_VALORES(TOPICO, VALOR, FECHA):
    TRY:
        CONN = MARIADB.CONNECT(
            USER="DATOS_IOTUSER",
            PASSWORD="C1966",
            HOST="LOCALHOST",
            PORT=3386,
            DATABASE="DATOS_IOT")
    EXCEPT MARIADB.ERROR AS E:
        PRINT(F"ERROR CONECTANDO CON LA PLATAFORMA: {E}")
        SYS.EXIT(1)

CUR = CONN.CURSOR()

#LEE DATOS DE BASE DE DATOS MARIADB
```

```
#CUR.EXECUTE("SELECT FIRST_NAME, LAST_NAME FROM DATOS_IOT  
FIRST_NAME=?, (SOME_NAME,))
```

```
#INSERTA VALORES EN MARIADB  
CUR.EXECUTE("INSERT INTO DATOS_IOT.DATOS (TOPICO,VALOR, FECHA) VALUES  
(?, ?, ?)", (TOPICO, VALOR, FECHA))
```

```
CONN.COMMIT()  
#PRINT(F"LAST INSERTED ID: {CUR.LASTROWID}")
```

```
CONN.CLOSE()  
# DEVOLUCION DE LA LLAMADA CUANDO EL CLIENTE RECIBE RESPUESTA  
DEF ON_CONNECT(CLIENT, USERDATA, FLAGS, RC):  
    PRINT("CONNECTED WITH RESULT CODE "+STR(RC))
```

```
# RESPUESTA CUANDO EL MENSAJE PUBLICADO ES RECIBIDO DESDE EL SERVIDOR  
MQTT
```

```
DEF ON_MESSAGE(CLIENT, USERDATA, MSG):  
    PRINT(MSG.TOPIC+" "+STR(MSG.PAYLOAD))  
    PRINT(STR(MSG.PAYLOAD))
```

```
IF ((MSG.TOPIC) == TOPICO_TEMP):  
    TEMPERATURA = FLOAT(MSG.PAYLOAD)  
    TOPICO = (MSG.TOPIC)  
    AHORA = DATETIME.DATETIME.NOW()  
    PRINT("TEMPERATURA: ",TEMPERATURA)  
    PRINT(AHORA)  
    ESCRIBIR_VALORES(TOPICO, TEMPERATURA, AHORA)
```

```
IF ((MSG.TOPIC) == TOPICO_HUME):  
    HUMEDAD = FLOAT(MSG.PAYLOAD)  
    TOPICO = (MSG.TOPIC)  
    AHORA = DATETIME.DATETIME.NOW()  
    PRINT("HUMEDAD: ",HUMEDAD)  
    PRINT(AHORA)  
    ESCRIBIR_VALORES(TOPICO, HUMEDAD, AHORA)
```

```
DEF MAIN():
```

```
CLIENT = MQTT.CLIENT()  
CLIENT.ON_CONNECT = ON_CONNECT  
CLIENT.ON_MESSAGE = ON_MESSAGE  
CLIENT.CONNECT(HOST='LOCALHOST', PORT=1883)  
CLIENT.SUBSCRIBE(MQTT_PATH)  
CLIENT.LOOP_FOREVER()
```

```
MAIN()
```


CONFIGURAR SERVIDOR MQTT MOSQUITTO EN RASPBERRY PI

INSTALACION DEL SERVIDOR MQTT MOSQUITTO

Al igual que en otras ocasiones utilizaremos el sistema operativo Raspbian.

1. Antes de comenzar a instalar **mosquitto** en nuestra Raspberry Pi, primero debemos hacer esto ejecutando los siguientes dos comandos:

```
sudo apt update
```

```
/* esta upgrade instruccion actualiza todos los paquetes y no es prudente
```

```
sudo apt upgrade
```

2. En la consola que hemos abierto ponemos el comando.

```
sudo apt-get install mosquitto mosquitto-clients
```

Tras un tiempo de descarga de internet queda el Mosquitto instalado.

Ahora lo convertimos en un Servicio de arranque con el comando.

```
Sudo systemctl enable mosquitto.service
```

```
carles66@raspberrypi:/etc/mosquitto $ ls
ca_certificates  certs  conf.d  mosquitto.conf
carles66@raspberrypi:/etc/mosquitto $ sudo nano mosquitto.conf
carles66@raspberrypi:/etc/mosquitto $ mosquitto
1667554435: mosquitto version 2.0.11 starting
1667554435: Using default config.
1667554435: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1667554435: Create a configuration file which defines a listener to allow remote
access.
1667554435: For more details see https://mosquitto.org/documentation/authenticat
ion-methods/
1667554435: Opening ipv4 listen socket on port 1883.
1667554435: Opening ipv6 listen socket on port 1883.
1667554435: mosquitto version 2.0.11 running
```

Podemos probar el funcionamiento del sistema abriendo dos ventanas y usando los comandos `mosquitto_sub` y `mosquitto_pub`, que permiten suscribirse a un tópico o publicar un tópico en el server MQTT. Abre dos ventanas de consola y escribe en cada una de ellas:

Para publicar un tópico escribe:

```
mosquitto_pub -h localhost -t "casa/sala1/temperatura" -m "20"
```

```
carles66@raspberrypi:~ $ sudo mosquitto_pub -h localhost -t "casa/sala1/temperatura" -m "54"
carles66@raspberrypi:~ $ █
```

En donde `-h` indica el hostname; sería **localhost** en este caso. La opción `-t` indica el tópico (intentamos enviar la temperatura de la sala dentro de mi casa) y finalmente la opción `-m` indica el mensaje (indico que estamos a 20 grados celsius)

y en la otra

Para suscribir un tópico escribe:

```
mosquitto_sub -h localhost -t "casa/sala1/temperatura"
```

```
carles66@raspberrypi:~ $ mosquitto_sub -h localhost -t "casa/sala1/temperatura"
54
█
```

De modo que, cuando se publique en el tópico temperatura de la sala1 se me mostrará el mensaje (presiona CTRL + C para detenerlo)

Verás que en la consola en la que suscribiste el tópico casa/sala1/temperatura aparece el valor que publicaste en la otra consola, la de publicación, el valor de la temperatura.

CREANDO EL FICHERO MOSQUITTO.CONF

Todo el comportamiento del Broker Mosquitto, se controla desde un fichero unico llamado mosquitto.conf, que debe estar situado en el mismo directorio de instalacion de Mosquitto, deberia estar en:

/etc/mosquitto que es donde el broker lo busca añ arrancar.

```
carles66@raspberrypi:/etc/mosquitto $ ls
ca_certificates  certs  conf.d  mosquitto.conf
carles66@raspberrypi:/etc/mosquitto $
```

Borra todo el fichero de configuracion y escribe las siguientes lineas:

```
listener 1883
allow_anonymous false

log_dest file /etc/mosquitto/mosquitto.log
password_file /etc/mosquitto/passwd
```


CONFIGURAR MYSQL(MARIADB) EN RASPBERRY PI

COMO CONFIGURAR MYSQL EN LA RASPBERRY PI

Al igual que en otras ocasiones utilizaremos el sistema operativo Raspbian.

1. Antes de comenzar a instalar MySQL en nuestra Raspberry Pi, primero debemos actualizarla. Podemos hacer esto ejecutando los siguientes dos comandos:

```
sudo apt update
```

```
sudo apt upgrade
```

2. El siguiente paso es instalar el software del servidor MySQL en tu Raspberry Pi.

```
sudo apt install mariadb-server
```

3. Con el software del servidor MySQL instalado en la Raspberry Pi, ahora necesitaremos asegurarlo estableciendo una contraseña para el usuario " root ".

De forma predeterminada, MySQL se instala sin contraseña configurada, lo que significa que puede acceder al servidor MySQL sin ninguna autenticación.

Por lo que primero estableceremos la contraseña:

```
sudo mysql_secure_installation
```

Simplemente sigue las instrucciones para establecer una contraseña para el usuario root y asegurar su instalación de MySQL.

Para una instalación más segura, debe responder " Y " a todas las preguntas cuando se le pida que responda " Y " o " N ". Estas indicaciones eliminarán las funciones que permiten a alguien obtener acceso al servidor más fácilmente.

Asegúrate de escribir la contraseña que estableciste durante este proceso, ya que tendremos que usarla para acceder al servidor MySQL y crear bases de datos y usuarios..

4. Ahora, si quieres acceder al servidor MySQL en la Raspberry Pi y comenzar a realizar cambios en tus bases de datos, puedes ingresar el siguiente comando.

```
sudo mysql -u root -p
```

5. Te pedirá que ingreses la contraseña del paso anterior para el usuario root de MySQL.

6. Ahora puedes introducir comandos MYSQL para crear, alterar y eliminar bases de datos. A través de esta interfaz, también puedes crear o eliminar usuarios y asignarles los derechos para administrar cualquier base de datos.

7. Hay dos formas diferentes de salir de la línea de comandos MYSQL, la primera es escribir " exit " en la interfaz MySQL.

La otra manera es presionar **Ctrl + D** .

8. En este punto, ahora habrás configurado con éxito MySQL en tu Raspberry Pi. Nuestras próximas secciones se utilizarán para hacer un mejor uso de esta base de datos.

CREAR UNA BASE DE DATOS MYSQL Y UN USUARIO

1. Antes de proceder a crear un usuario y una base de datos MySQL en nuestra Raspberry Pi, primero debemos volver a iniciar sesión en la herramienta de línea de comandos MySQL.

Ejecuta el siguiente comando para iniciar sesión en la línea de comandos de MySQL. Te pedirá que introduzcas la contraseña para la cuenta " **root** " que configuraste anteriormente.

```
sudo mysql -u root -p
```

2. Comencemos creando una base de datos MySQL usando el siguiente comando. Este comando es súper simple y es solo " **CREATE DATABASE** " seguido del nombre que desees dar a la base de datos.

En nuestro ejemplo, llamaremos a esta base de datos " **datos IoT** ".

```
CREATE DATABASE datos_IoT;
```

3. A continuación, crearemos un usuario MySQL que asignaremos a nuestra nueva base de datos. Podemos crear este usuario ejecutando el siguiente comando.

Para este ejemplo, llamaremos al usuario " **datos IoTuser** " y le daremos la contraseña " **c1966** ". Al crear el tuyo, asegúrate de reemplazar ambos.

```
CREATE USER 'datos_IoTuser'@'localhost' IDENTIFIED BY 'c1966';
```

4. Con el usuario creado, ahora podemos seguir adelante y otorgar todos los privilegios al usuario para que pueda interactuar con la base de datos.

Este comando otorgará todos los permisos a nuestro " **usuario** " para todas las tablas dentro de nuestra base de datos " **datos IoT** ".

```
GRANT ALL PRIVILEGES ON datos_IoT.* TO 'datos_IoTuser'@'localhost';
```

5. Lo último que debemos hacer para que se finalice nuestra base de datos MySQL y el usuario, es eliminar la tabla de privilegios. Sin vaciar la tabla de privilegios, el nuevo usuario no podrá acceder a la base de datos.

Podemos hacer esto ejecutando el siguiente comando.

```
FLUSH PRIVILEGES;
```

CREAR UNA TABLA DENTRO DE LA BASE DE DATOS.

1. Entramos en la base de datos, poniéndola en uso-

Podemos hacer esto ejecutando el siguiente comando.

```
USE datos_IoT;
```

2. Creamos la tabla identificando los campos y el tipo de datos de cada campo

Podemos hacer esto ejecutando el siguiente comando.

```
CREATE TABLE datos(  
    Id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    Topico VARCHAR(50),  
    valor DECIMAL(4,2),  
    Fecha DATETIME  
);
```

```
MariaDB [datos_IoT]> describe datos;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type                | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id    | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |  
| topico | varchar(50)         | YES  |     | NULL    |                |  
| valor | decimal(4,2)        | YES  |     | NULL    |                |  
| fecha | datetime            | YES  |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.005 sec)
```

3. Para visualizar que hemos creado la tabla o que tablas hay en la base de datos.

Podemos hacer esto ejecutando el siguiente comando.

```
SHOW TABLES;
```

4. Para visualizar la estructura de la tabla que hemos creado u otra tabla existente.

Podemos hacer esto ejecutando el siguiente comando.

```
DESCRIBE datos_IoT;
```

6. Para visualizar los datos introducidos o contenidos en la tabla.

Podemos hacer esto ejecutando el siguiente comando.

```
SELECT id, temperatura, humedad, fecha FROM datos_IoT;
```

```

MariaDB [datos_IoT]> SELECT id, topico, valor, fecha FROM datos;
+----+-----+-----+-----+
| id | topico                | valor | fecha                |
+----+-----+-----+-----+
|  1 | casa/sala1/humedad    | 54.44 | 2022-11-03 11:48:10 |
|  2 | casa/sala1/temperatura | 54.22 | 2022-11-03 11:48:52 |
|  3 | casa/sala1/temperatura |  4.20 | 2022-11-03 11:48:52 |
|  4 | casa/sala1/temperatura | 54.00 | 2022-11-03 11:48:52 |
|  5 | casa/sala1/humedad    | 66.11 | 2022-11-03 11:48:52 |
|  6 | casa/sala1/humedad    | 66.11 | 2022-11-03 11:48:52 |
|  7 | casa/sala1/humedad    | 66.11 | 2022-11-03 12:00:36 |
|  8 | casa/sala1/temperatura | 54.00 | 2022-11-03 12:00:50 |
|  9 | casa/sala1/humedad    | 54.44 | 2022-11-03 12:43:35 |
| 10 | casa/sala1/humedad    | 66.11 | 2022-11-03 13:38:29 |
| 11 | casa/sala1/humedad    | 54.44 | 2022-11-03 14:08:40 |
+----+-----+-----+-----+

```

/* Password mariadb rtyhjk66nm

/* instruccion para acceder a la base de datos mariadb

/* mysql -u datos_IoTuser -p password c1966