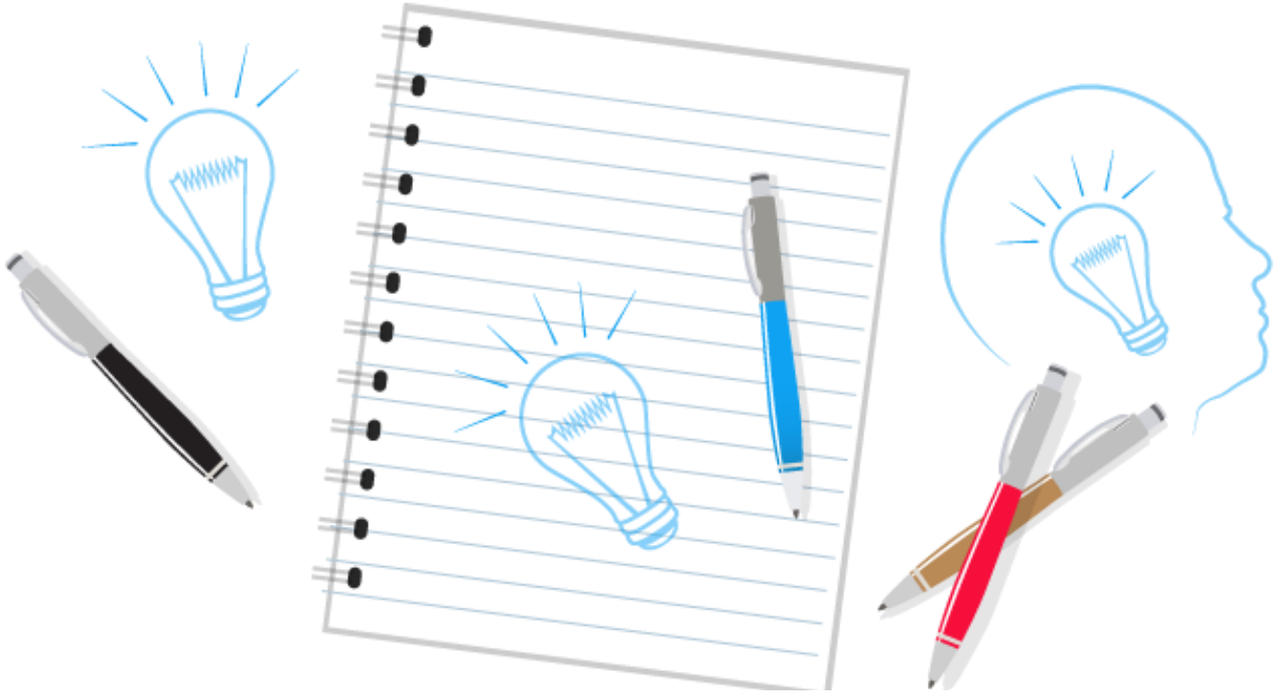
CAPÍTULO 8. Home Automation IoT

v.1.4 MARZO 2024

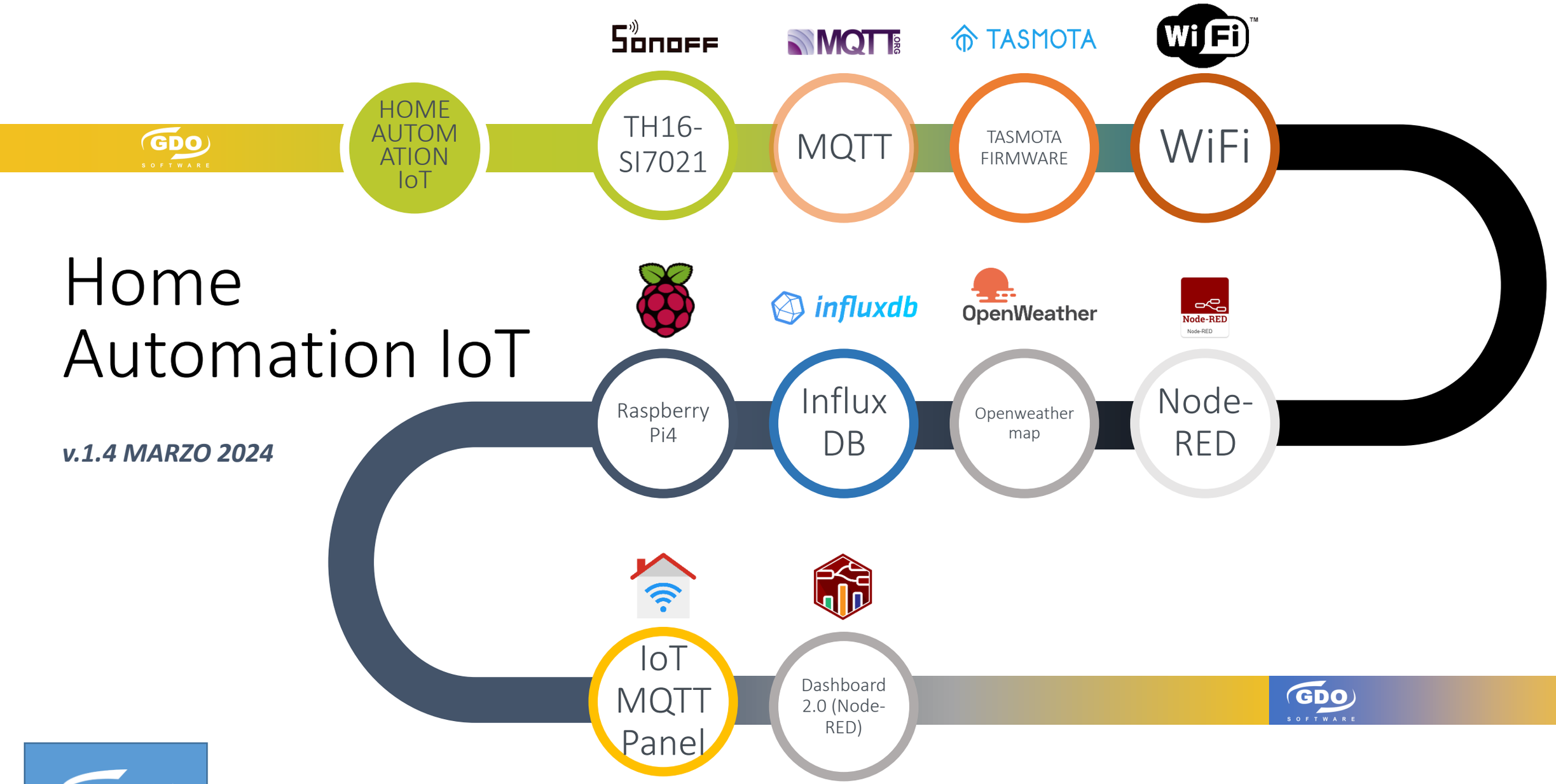
Ricardo Moraleda Gareta

[Director departamento de software de GDO Software]



Home Automation IoT

v.1.4 MARZO 2024



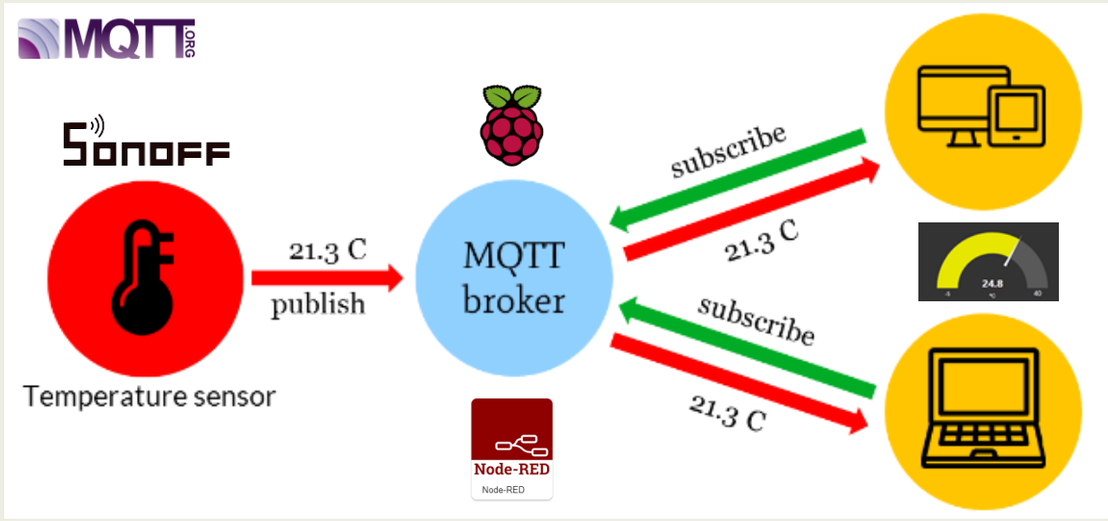
Home Automation IoT

Objetivo

El objetivo es mostrar un dashboard con información meteorológica (20 datos) del exterior y del interior de casa de manera completamente inalámbrica en una red local.

Los 5 conceptos básicos en MQTT:

- Publisher (sensor Tª y Hª de Sonoff en este caso)
- Subscriber (PC, Tablet, Smartphone, RPI4)
- Messages (información intercambiada)
- Topics (etiquetas de las variables a leer/escribir)
- Broker (Es quien gestiona los mensajes recibidos y los envía a los suscriptores, RPI4)



Sensor IoT

Sensor TEMP/HR

La fuente de datos, en este caso, es un sensor de temperatura y humedad modelo SI7021 de Sonoff para la placa TH16.

Con esta combinación podemos disponer de un sensor WiFi que envía datos (publica) a un broker por el protocolo MQTT (Message Queue Telemetry Transport).

https://sonoff.tech/product/wifi-diy-smart-switches/th10-th16

El sensor dispone de conector minijack de audio para conectarse con TH16.

- Humedad de 0 a 100%
- Temperatura de -40 °C a +85 °C
- Punto de rocío (°C)

La placa TH16 venía de serie con el firmware TASMOTA v8.1.0 Doris (by Theo Arends). Se puede actualizar a la v13.4.0 (la más reciente)

https://ota.tasmota.com/tasmota/release/

El conjunto puede salir por unos 35 €.

Sensor IoT

Configurar TH16

Lo primero, al alimentar el dispositivo, crea una propia red WiFi internamente. En mi caso "tasmota-3583".

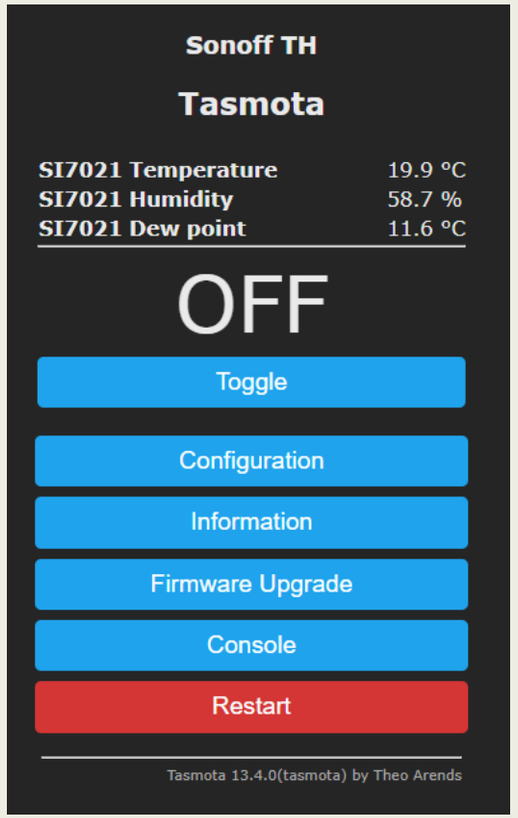
Con el PC conéctate a esa red y abre un navegador web a la IP 192.168.4.1 y se puede configurar la WiFi deseada (SSID y password).

Yo la he cambiado a la red WiFi de mi casa y ahora automáticamente se conecta a la IP 192.168.1.165.

Para ver la IP exactamente, con el software para PC Advanced Port Scanner v.2.5.3581 detecto la IP y puerto (HTTP) del menú web de configuración.

Puesto que no dispongo del último firmware, selecciono la opción "Firmware Upgrade" para pasar a la 13.4.0. Esta opción la he realizado vía OTA "https://ota.tasmota.com/tasmota/release/tasmota.bin"

Firmware TASMOTA



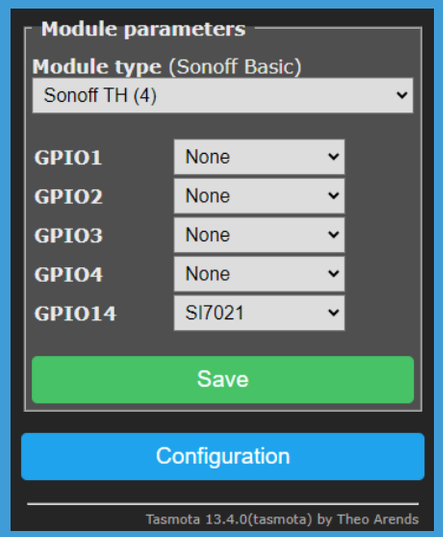
Sensor IoT

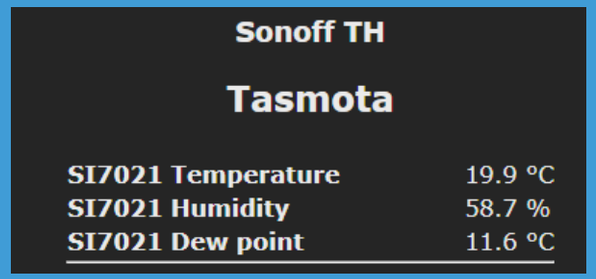
Configurar TH16

Entrando en el menú Configuration aparecen otras opciones a configurar.

En mi caso voy a configurar:

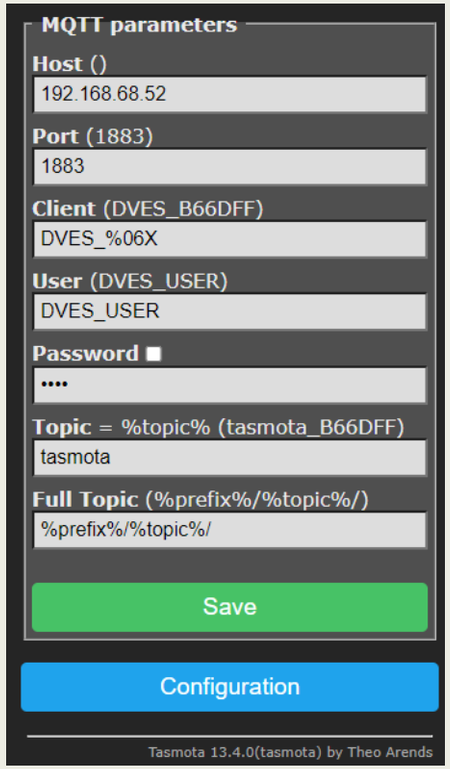
- **Module**. Seleccionando TH y en el GPIO14 el sensor a conectar. En este caso SI7021.





Sensor y MQTT

- **MQTT**. Broker y datos de conexión y topics.



@IP del broker MQTT (Raspberry Pi4)

Puerto

Nombre del cliente

Usuario

Password

Topic. En esta caso tele/<topic>/…

Full topic

Los topics serán:

- **tele/tasmota/LWT**
- **tele/tasmota/INFO1**
- **tele/tasmota/INFO2**
- **tele/tasmota/STATE**
- **tele/tasmota/SENSOR**

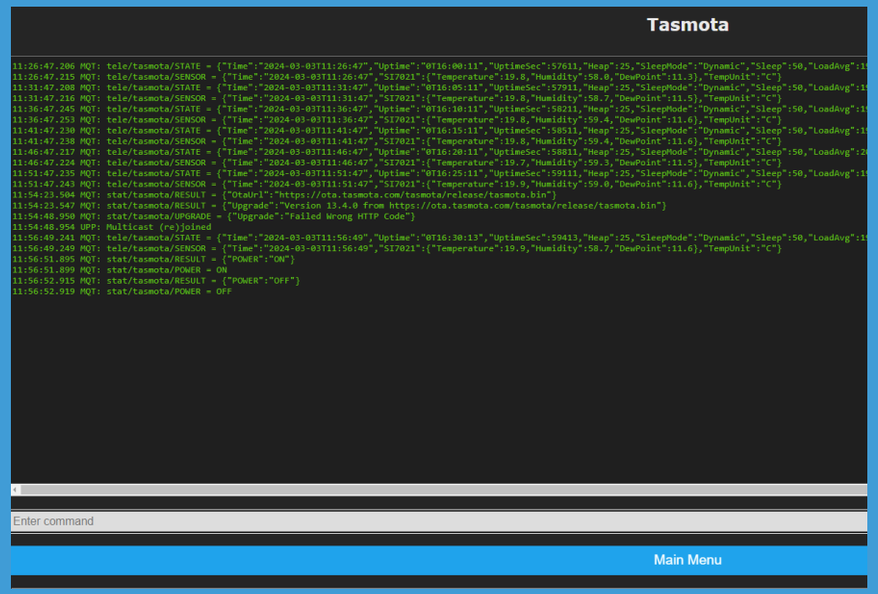
Sensor IoT

TASMOTA

Configurar TH16

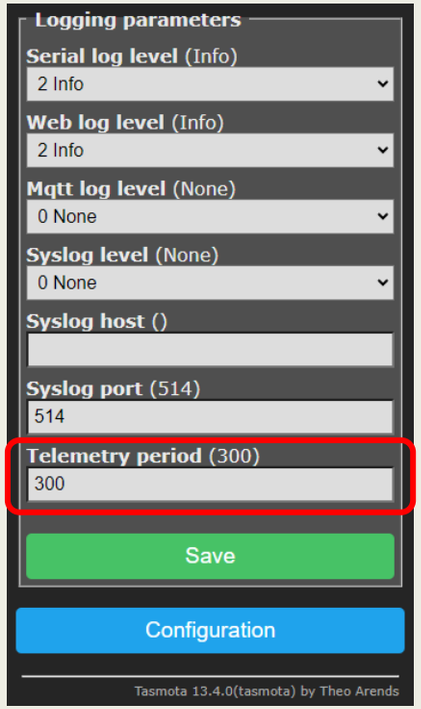
- La zona horaria. Entrar en “Console” desde el menú principal. Al escribir el comando “timezone” te muestra cual es. En mi caso +1. Si tu país tiene variabilidad de hora en verano e invierno escribir el comando “timezone 99” para que el NTP lo entienda variable.

  https://tasmota.github.io/docs/Commands/ (lista de comandos)



Más configuraciones

- Frecuencia de envío de datos. Entrar en Configuration>Configure Logging. Modificar el parámetro “Telemetry period” (segundos). El mínimo es 10 segundos.



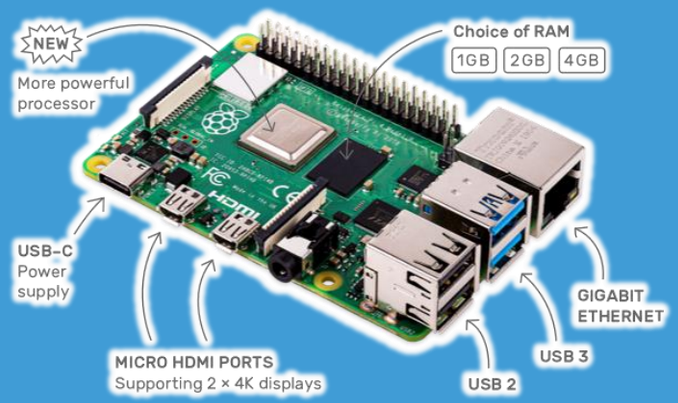
MQTT Broker Server

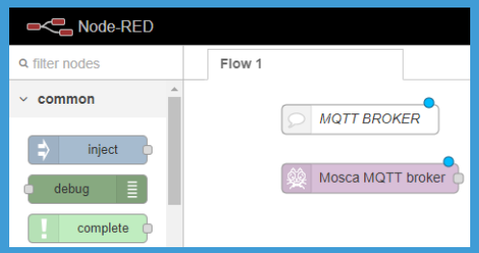
Raspberry Pi 4

El broker MQTT que utilizaremos estará en una Raspberry Pi 4 conectado en la misma Wi-Fi que el sensor (TH16-Si7021).

Se utilizará Node-Red V.1.0.6 para ello y el nodo "node-red-contrib-aedes" v0.6.0, llamado **Aedes.**

```
pi@raspberrypi:~ $ sudo apt install mosquitto
```





IP: 192.168.1.163

Puerto MQTT: 1883

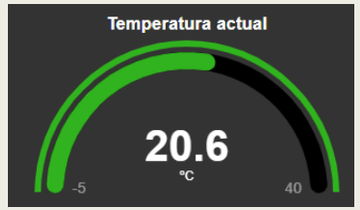
Comprobar que se tiene ese puerto escuchando con el siguiente comando:

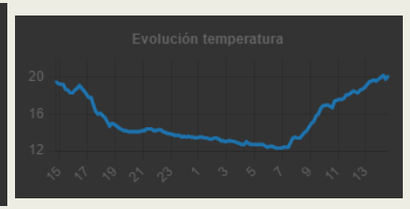
```
pi@raspberrypi:~ $ sudo netstat -lptu
tcp        0      0 0.0.0.0:1883            0.0.0.0:*               LISTEN      4526/mosquitto
```

Server

La Raspberry Pi4 no sólo hace de broker si no que además tiene las siguientes funciones (así programadas):

1. Hace de MQTT client para recoger los datos del sensor TH16
2. Comprueba la conexión con el sensor mediante PING
3. Recibe datos del tiempo meteorológico externo a través de un servicio llamado "**openweathermap**"
4. Historiza los datos de temperatura y humedad, tanto interior como exterior.
5. Presenta un Dashboard con los datos actuales, información del tiempo y gráficas.





MQTT Clients

Diferentes MQTT clients

Los diferentes clientes podrán publicar y suscribirse a topics para escribir y leer información.

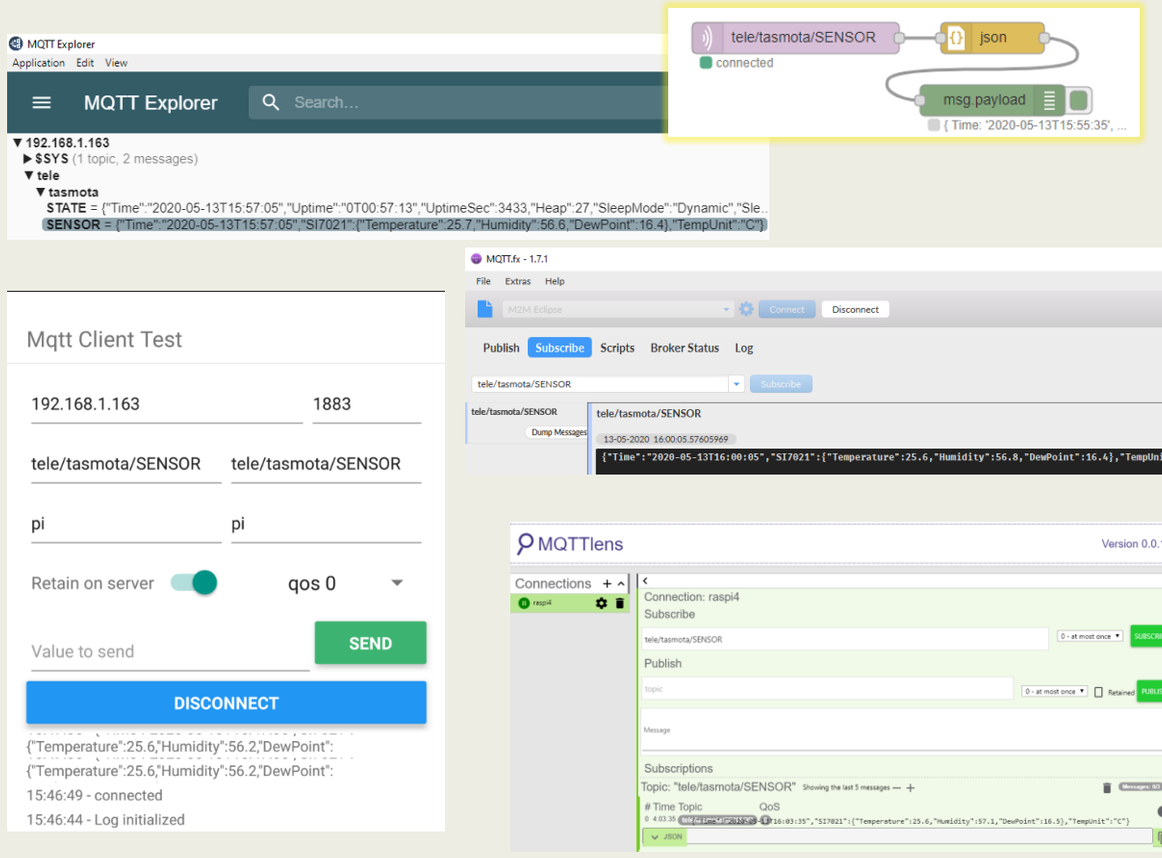
Como clientes MQTT tendremos:

- Publicador: Sensor de temperatura y humedad TH16-Si7021.
- Suscriptor: PC, Smartphone, Tablet

Para simular he realizado pruebas de suscripción al mismo topic: "**tele/tasmota/SENSOR**" con diferentes clientes:

- Mqtt Client Test 0.4.1 para Android
- MQTT Explorer 0.3.5 para PC
- MQTT.fx 1.7.1 para PC
- MQTT Lens 0.0.14 extensión para Google Chrome
- **Nodo mqtt in de Node-Red 1.0.6. Es el que usaremos como ej.**

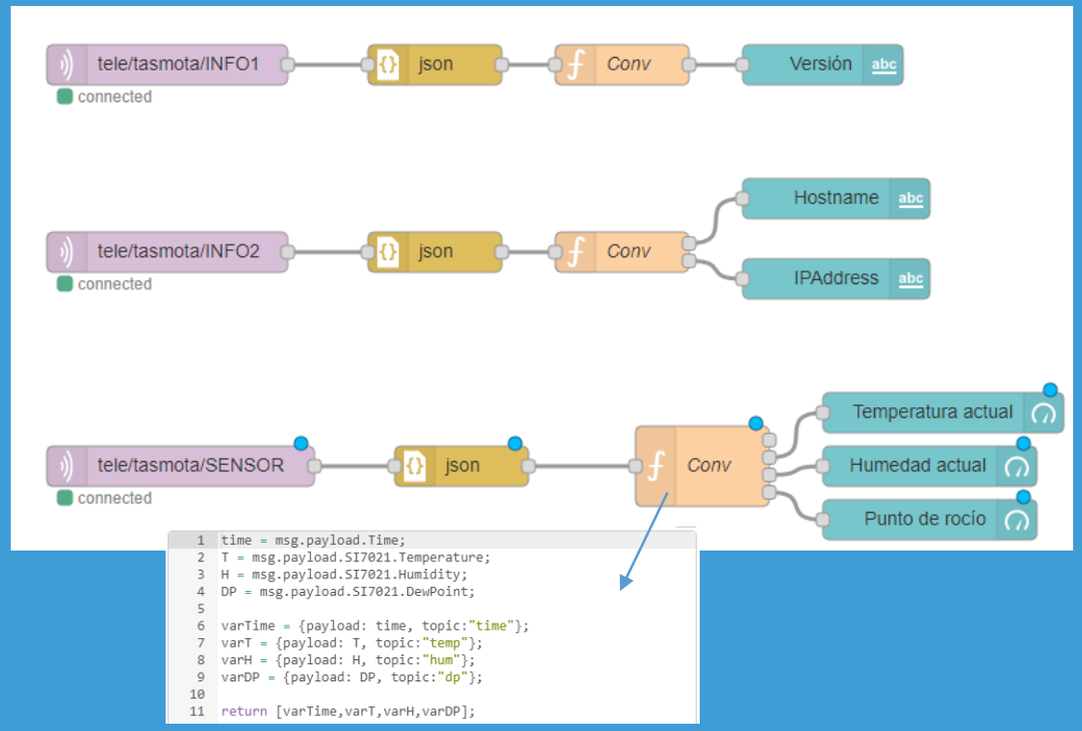
TEST MQTT CLIENTS



Obtención de datos

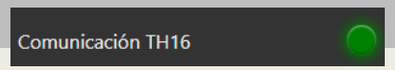
Node-Red - MQTT

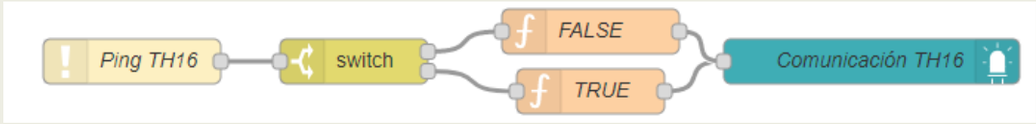
Un ejemplo de suscripción a los siguientes topics y visualización en dashboard. Datos del sensor interior.



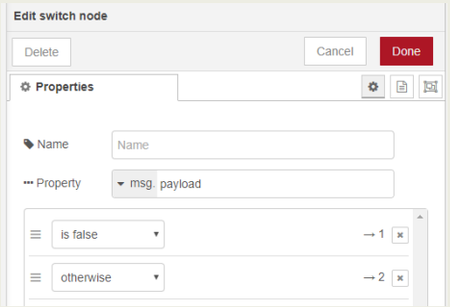

```
time = msg.payload.Time;
T = msg.payload.SI7021.Temperature;
H = msg.payload.SI7021.Humidity;
DP = msg.payload.SI7021.DewPoint;

varTime = {payload: time, topic:"time"};
varT = {payload: T, topic:"temp"};
varH = {payload: H, topic:"hum"};
varDP = {payload: DP, topic:"dp"};

return [varTime,varT,varH,varDP];
```

Estado conexión sensor





La función ping devuelve un entero con los ms del tiempo de ping y false si no hay conexión. Si no funcionase, revisad la indicación de abajo.



2 On some versions on Raspbian (Raspberry Pi) `ping` seems to be a root only command. The fix is to allow it as follows

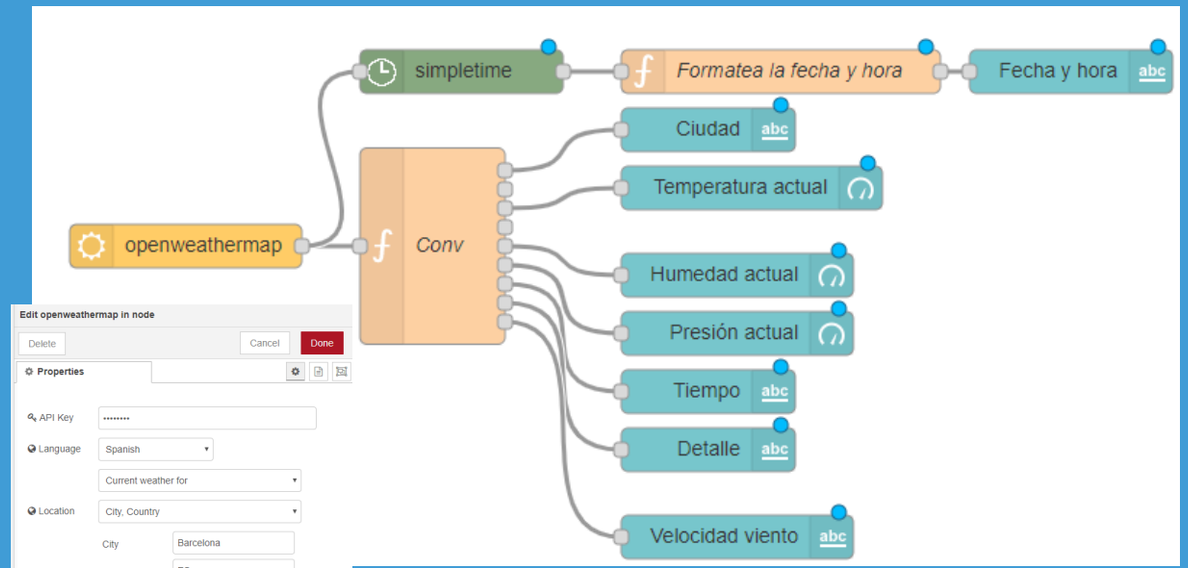

```
sudo setcap cap_net_raw=ep /bin/ping
sudo setcap cap_net_raw=ep /bin/ping6
```

Obtención de datos

Node-Red

Para obtener el tiempo exterior utilizo el nodo "openweathermap".

Para su utilización es necesario generar una API key en su web. En mi caso, la FREE. https://openweathermap.org/price

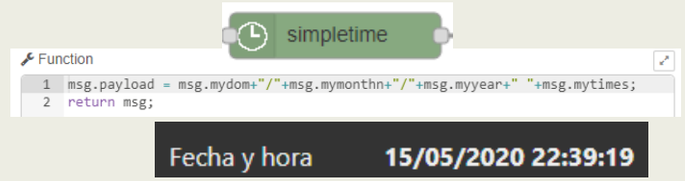


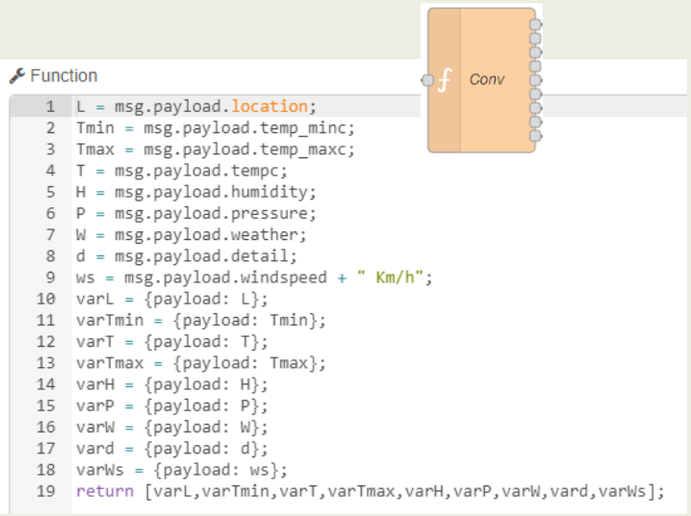
Datos recibidos

El resultado en debug de la salida del nodo "openweathermap" es la siguiente:

msg.payload : Object

```
object
  id: 801
  weather: "Clouds"
  detail: "algo de nubes"
  icon: "02n"
  tempk: 287.85
  tempc: 14.7
  temp_maxc: 15.5
  temp_minc: 13.8
  humidity: 87
  pressure: 1007
  maxtemp: 288.71
  mintemp: 287.04
  windspeed: 3.1
  winddirection: 330
  location: "Barcelona"
  sunrise: 1589430797
  sunset: 1589482937
  clouds: 20
  description: "The weather in Barcelona at coordinates: 41.39, 2.16 is Clouds (algo de nubes)."
```





InfluxDB

Influx DB (database)

Para historizar los datos partimos de tener instalado el software de base de datos InfluxDB en la Raspberry Pi 4. Me he descargado la versión 1.8.0

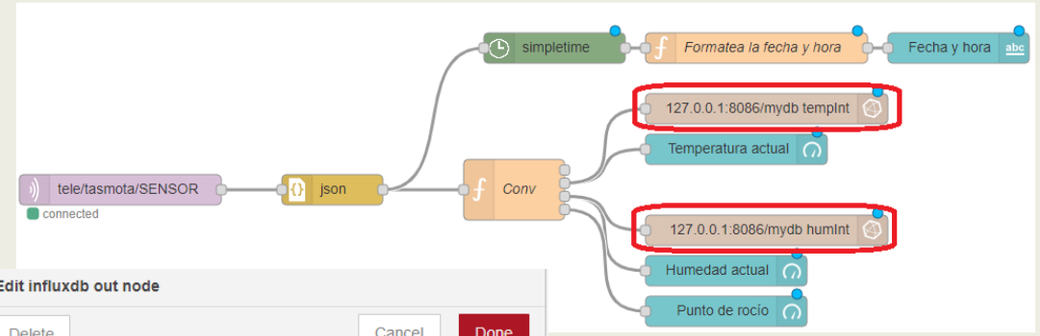
La tengo configurada como servicio. Para abrir la consola (Shell) de influxdb poner "**influx**".

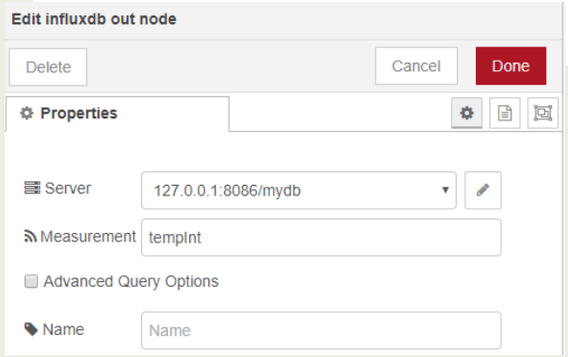
```
pi@raspi4: ~
Archivo  Editar  Pestañas  Ayuda
pi@raspi4:~ $ influx
Connected to http://localhost:8086 version 1.8.0
InfluxDB shell version: 1.8.0
> show databases
name: databases
name
----
_internal
mydb
> use mydb
Using database mydb
> select * from tempINt
> select * from tempINt
> select * from tempInt
name: tempInt
time                value
----                -----
1589395205907643001 25
1589395235810560166 25
1589395265899879372 25
1589395295821714452 25
1589395325849323832 25
1589395355851486965 25
1589395385829523581 25
> select * from tempExt
name: tempExt
time                value
----                -----
1589395184529831841 19.8
> select * from humExt
name: humExt
time                value
----                -----
1589395184530002468 77
```

Historización de datos

Para crear la base de datos de ejemplo poner "**create database mydb**". mydb es el nombre de la base de datos.

Luego, en Node-Red iré poniendo los 4 valores a persistir en los nodos influxdb out en la opción Measurement.





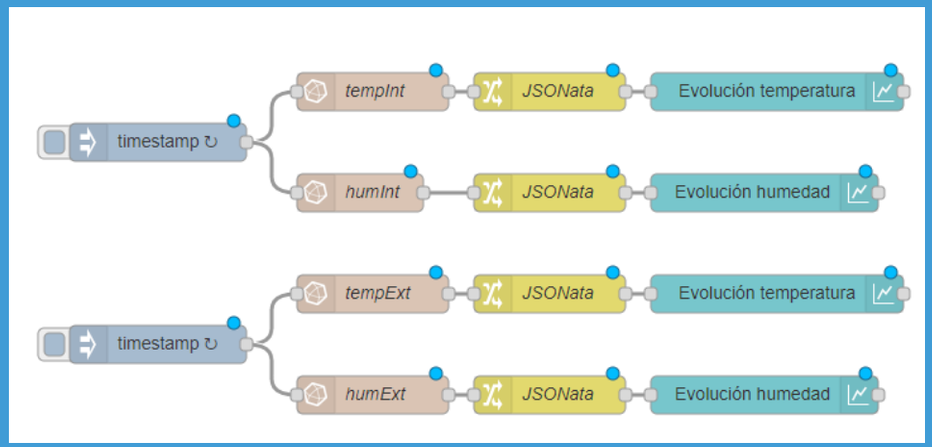
Measurements:

- tempInt
- humInt
- tempExt
- humExt

InfluxDB

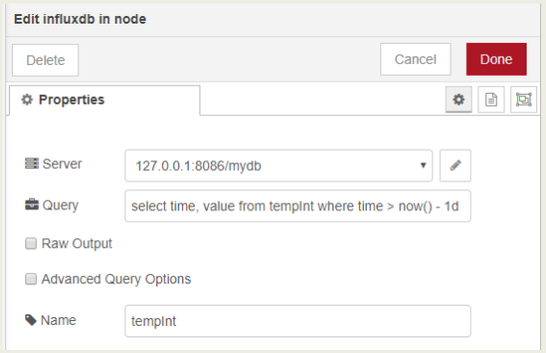
Influx DB (database)

Para leer los datos historizados y mostrarlos en gráficas de línea hago lo siguiente. Se temporizan las queries a 1 minuto.

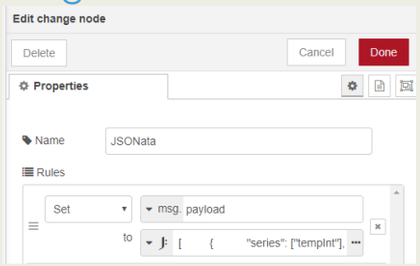


Consulta de datos

Un nodo de influxdb in con una query (select) de los datos del instante de ejecución de la query, 1 día atrás.



Importante el change node (llamado en mi caso JSONata) para que los datos se formateen adecuadamente como entrada de la gráfica.




```
[
    {
        "series": ["tempInt"],
        "data": [[msg.payload.{"x": time, "y": $."value"}
    }
]
```

Home Automation IoT

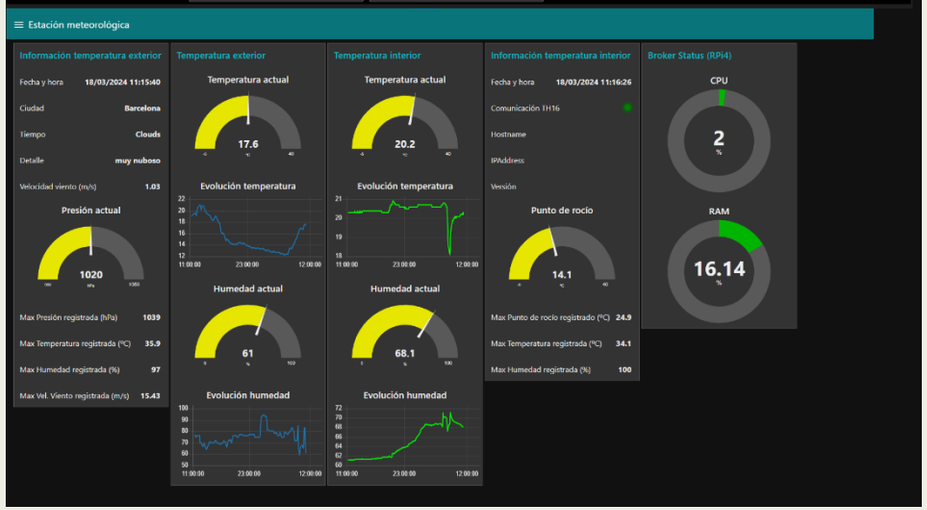
Resumen

- Detallado dashboard (2.0) que puedes ver en cualquier dispositivo local (Smartphone, PC, Tablet) al ser un entorno web accesible mediante la siguiente URL http://192.168.68.52:1880/dashboard/em
- Utilización de un sensor de Temperatura y Humedad ya preprogramado y sólo configurarlo gracias al firmware tasmota (plug, set up and play).
- Complemento con otros datos del tiempo meteorológico externo de tu ciudad mediante una API (servicio web).
- Historización de los datos para poder graficar y poder recuperar las gráficas en caso de caída del servidor (Raspberry Pi4).
- Comprobación de la comunicación entre el broker MQTT y el sensor (publicador).

Resultado gráfico

Dashboard 2.0

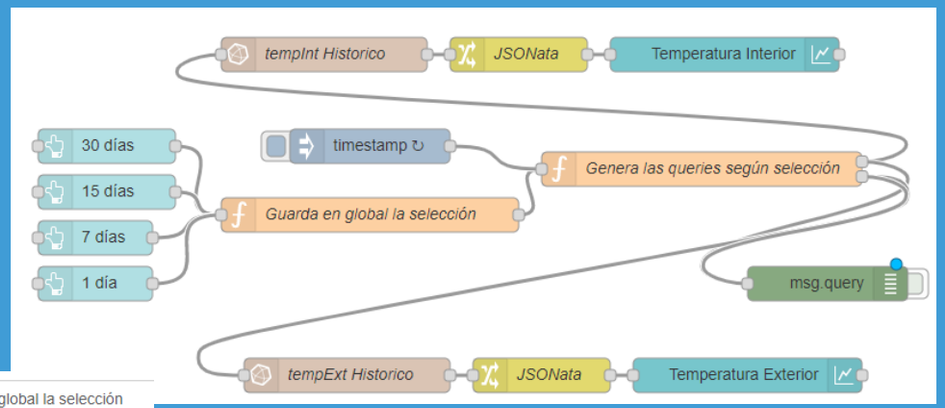
Dashboard 1.0



Home Automation IoT

Históricos

- Puesto que vamos guardando todos los registro en la BBDD InfluxDB podemos mostrar las tendencias históricas de 1-7-15-30 días atrás.



Name: Guarda en global la selección

Function

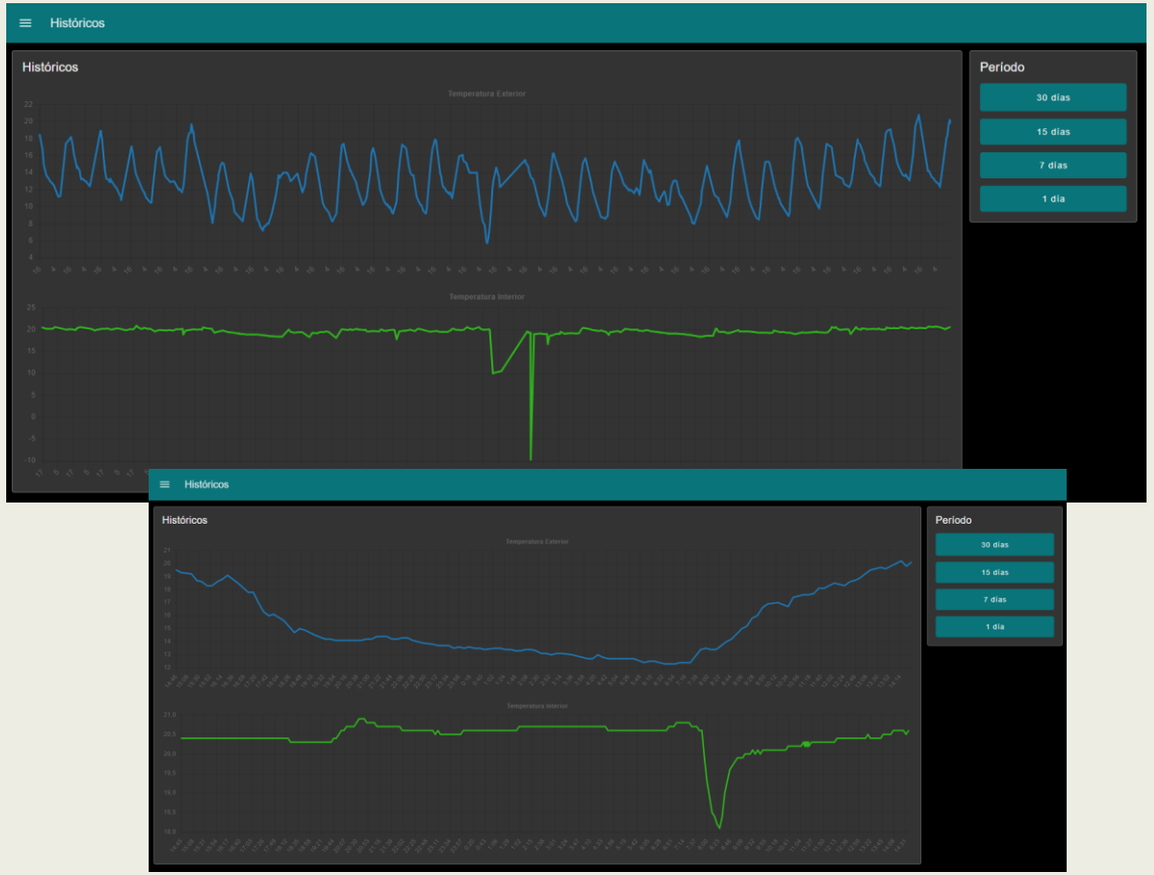
```
global.set('periodo', msg.payload);
return msg;
```

Name: Genera las queries según selección

Function

```
var selectQueryInt = "select time, value from tempInt where time > now() - "+global.get('periodo')+"d tz('Europe/Madrid')";
var selectQueryExt = "select time, value from tempExt where time > now() - "+global.get('periodo')+"d tz('Europe/Madrid')";

varInt = {query: selectQueryInt, topic:"int"};
varExt = {query: selectQueryExt, topic:"ext"};

return [varInt,varExt];
```

Trend



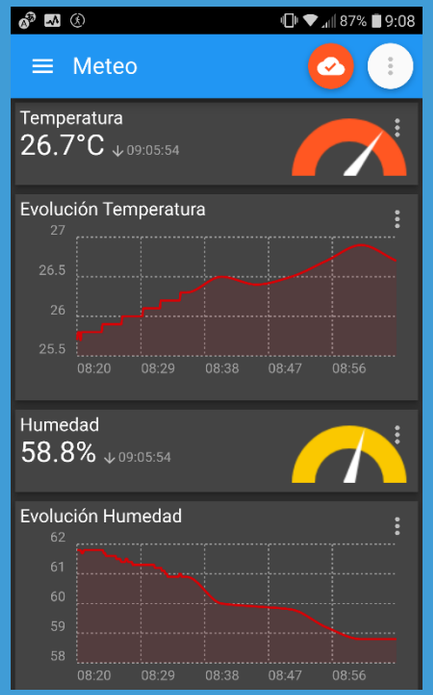
Home Automation IoT

IoT MQTT Panel

GET IT ON Google Play

Rahul Kundu

En Play Store existe una app para Smartphone/Tablet bastante configurable para hacerte un dashboard a medida con los datos del sensor (SI7021 con TH16).



1º - Configurar broker:

- @IP del broker MQTT (Raspberry Pi 4)
- Puerto 1883 TCP
- Usuario y password

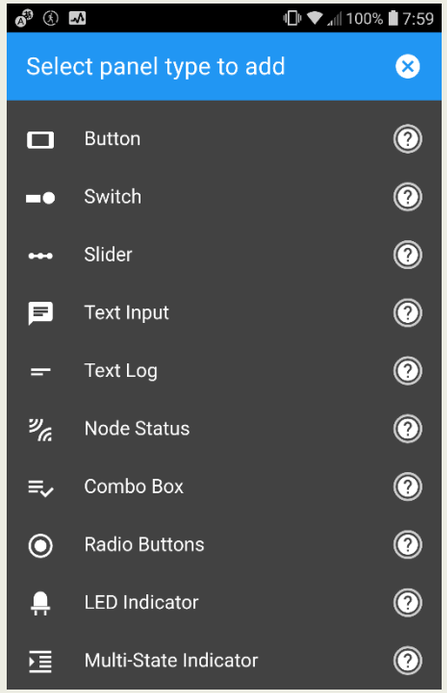
2º - Configurar el topic: tele/tasmota/SENSOR

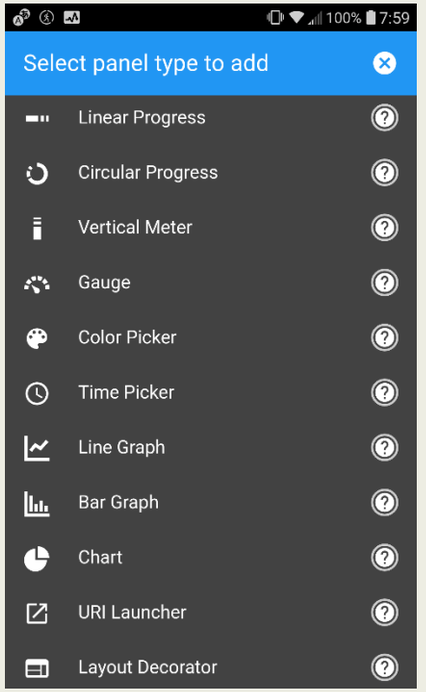
3º - Configurar el path JSON

- **Temperatura**. El payload como es JSON, el JsonPath es $.SI7021.Temperature
- **Humedad**. El payload como es JSON, el JsonPath es $.SI7021.Humidity

IoT MQTT Panel v.0.39.42

- **La app dispone de multitud de componentes para añadir y configurar tu propio dashboard.**





Home Automation IoT
v.1.4 MARZO 2024

https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099

https://www.linkedin.com/company/gdo-electric1996/

RICARDO MORALEDA GARETA

GDO SOFTWARE