

SISTEMAS EMBEBIDOS

GRUPO - 1

Proyecto

Enviar datos de un Raspberry pi a la plataforma thingsboard

Nombre Profesor:

MIGUEL ANGEL QUIROZ MARTINEZ

Nombre de Integrantes Grupo:

- I. Mario Antonio Arguello Calle
- II. Jaime Andres Barriga Torres
- III.** Elias Alberto Reyes Conforme

ThingsBoard

Es una plataforma de IoT de código abierto para la gestión de dispositivos, la recopilación, el procesamiento y la visualización de datos para sus proyectos de IoT.

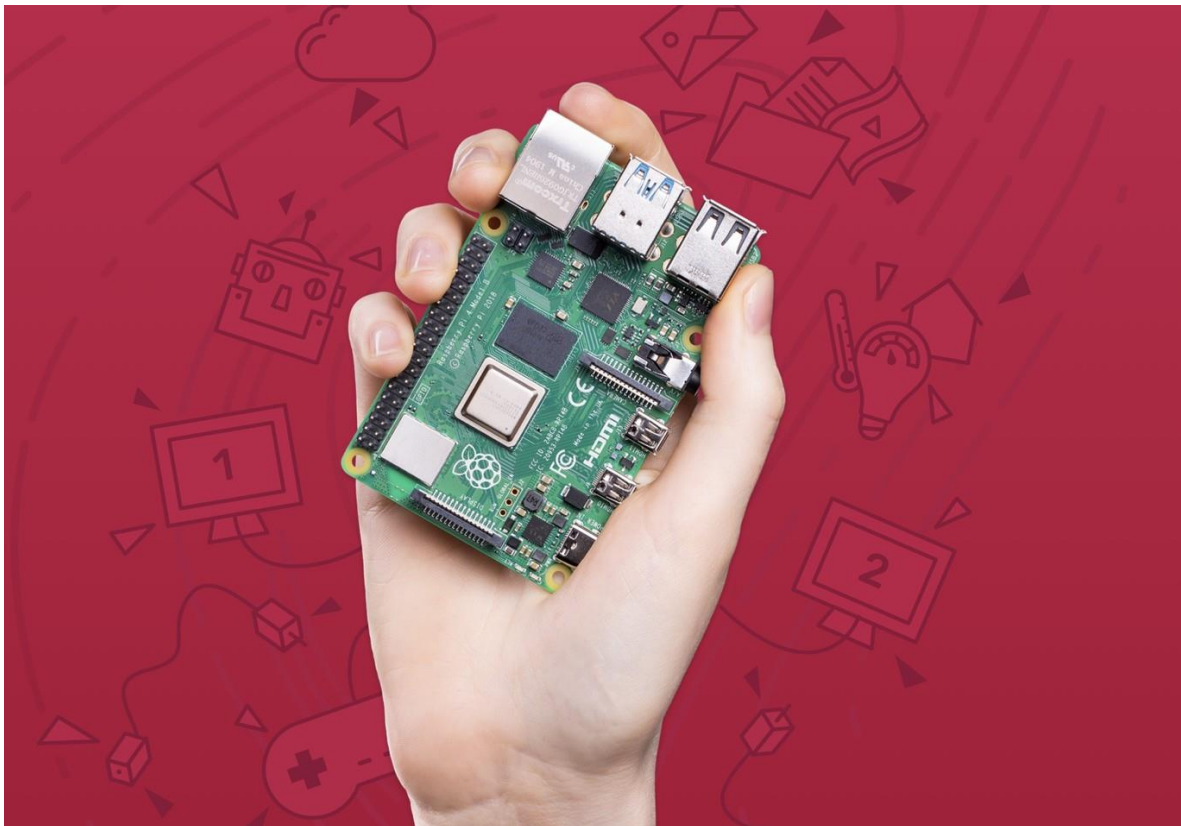


Características

- Es multi tenant, lo que permite servir a distintos clientes con configuraciones diferentes. A su vez, cada cliente puede tener múltiples usuarios y cada usuario gestionar distintos dispositivos.
- La plataforma puede utilizar CoAP, HTTP o MQTT para comunicarse con los dispositivos IoT.
- Dispone de más de 30 plugins para visualizar la información recolectada, como gráficas en función del tiempo, indicadores gráficos, mapas, etc.
- El manejo de eventos se realiza de manera gráfica, conectando bloques de código para enviar datos a otras aplicaciones, enviar notificaciones o disparar alarmas.

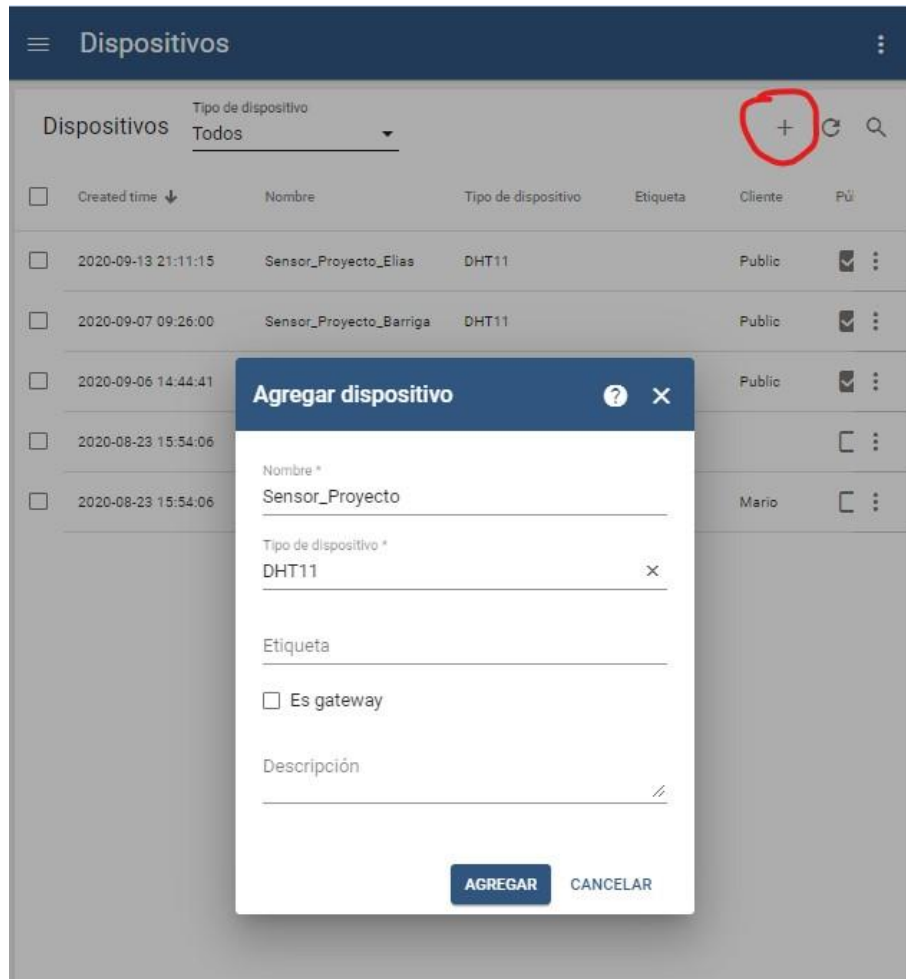
¿Qué es Raspberry Pi?

La Raspberry Pi es un ordenador de bajo coste y tamaño reducido, tanto es así que cabe en la palma de la mano, pero puedes conectarle un televisor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora.



Configuración de ThingsBoard

Para configurar el ThingsBoard tenemos que irnos a la opción Dispositivos y crear uno nuevo.



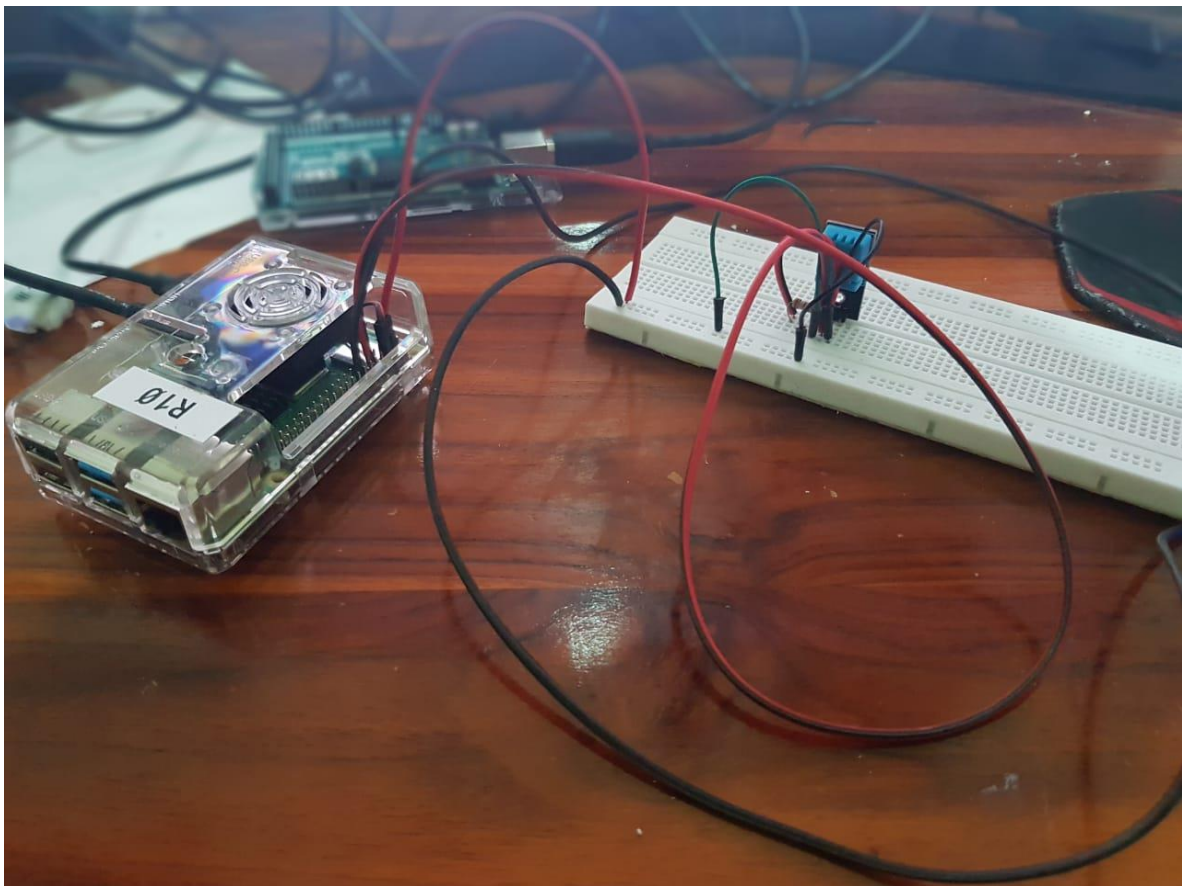
Una vez le damos agregar nos creará un nuevo dispositivo la cual tendrá nuestro **ACCESS TOKEN** que nos ayudará a poder conectarnos con la plataforma y el dispositivo.

Ese Access token lo pegamos en el código del raspberry ejemplo:

```
ACCESS_TOKEN = 'zQBQs1HCUVPcR6UQKBYW'
```

Configuración del Raspberry e instalacion del sensor DHT11

Conectamos los cables del sensor en este caso estamos usando el sensor DHT11 que nos permitirá medir la temperatura y humedad que tengamos en el lugar que estemos, tenemos que conectar con 3.3v, el out en el gpio4 y el gnd va con el negativo del sensor.



Una vez tengamos bien conectado el sensor encendemos nuestro raspberry pi e comenzamos a descargar el adafruit con el paho-mqtt.

Código adafruit:

```
git clone https://github.com/adafruit/Adafruit_python_DHT.git
```

una vez instalado el adafruit entramos a la carpeta Adafruit_python_DHT con el siguiente comando:

```
cd Adafruit_python_DHT
```

he instalamos en archivo necesario en Python:

```
sudo python3 setup.py install
```

como estamos usando Python3 tenemos que modificar el adafruit nos dirigimos a la carpeta Adafruit_python_DHT/examples y abrimos el archivo llamado "**AdafruitDHT.py**"

Una vez abierto modificamos la primera línea de comando

```
#!/USR/BIN/PYTHON
```

A

```
#!/USR/BIN/PYTHON3
```

Abrimos el terminal e instalamos el paho-mqtt con el siguiente comando:

```
pip3 install paho-mqtt python-etcd
```

Abrimos el Thonny y escribimos el código:

```

import os
import time
import sys
import Adafruit_DHT as dht
import paho.mqtt.client as mqtt
import json

THINGSBOARD_HOST = 'demo.thingsboard.io'
ACCESS_TOKEN = 'DHT22_DEMO_TOKEN'

INTERVAL=2

sensor_data = {'temperature': 0, 'humidity': 0}

next_reading = time.time()

client = mqtt.Client()

# Set access token
client.username_pw_set(ACCESS_TOKEN)

# Connect to ThingsBoard using default MQTT port and 60 seconds keepalive interval
client.connect(THINGSBOARD_HOST, 1883, 60)

client.loop_start()

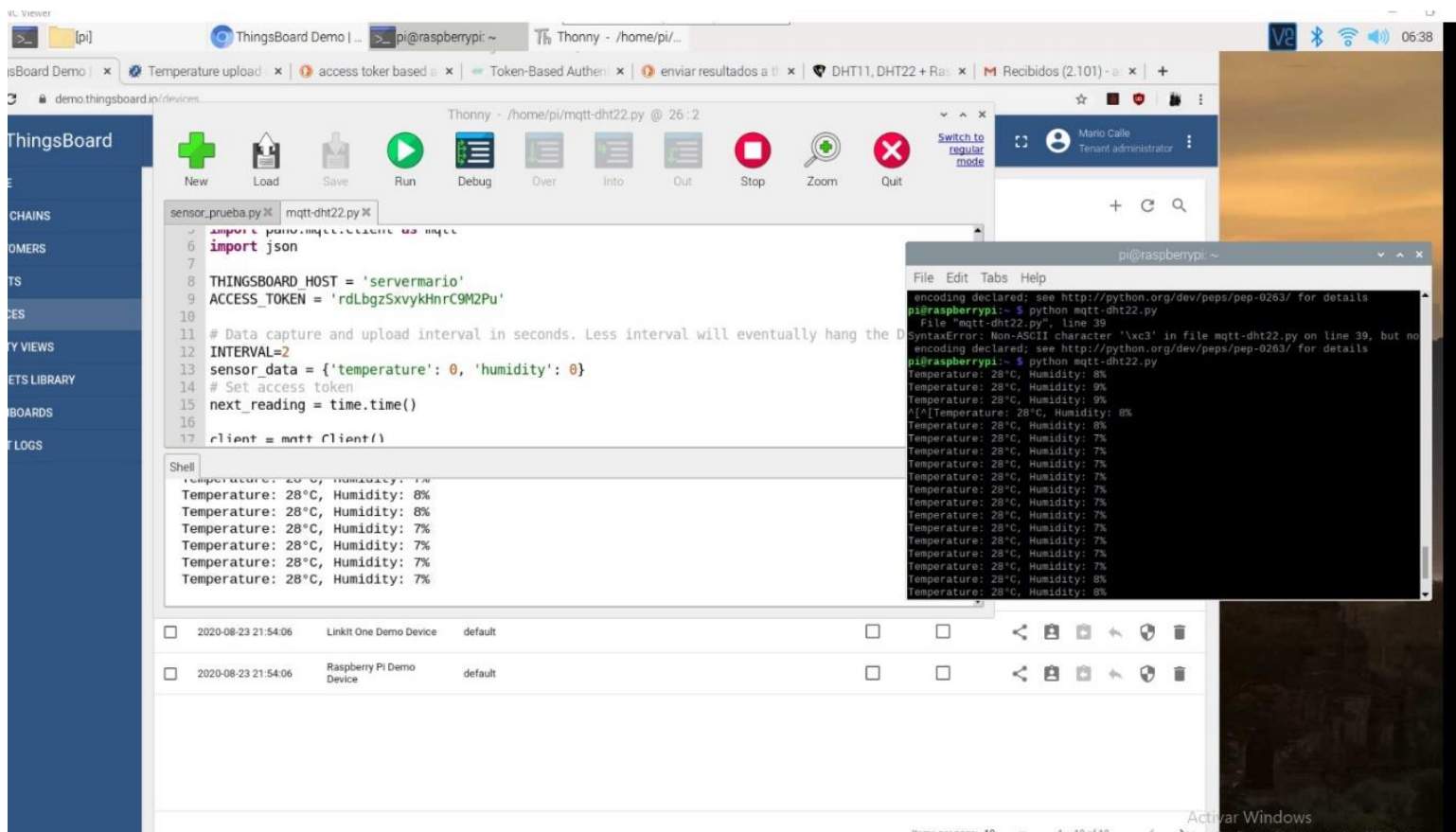
try:
    while True:
        humidity, temperature = dht.read_retry(dht.DHT22, 4)
        humidity = round(humidity, 2)
        temperature = round(temperature, 2)
        print(u"Temperature: {:g}\u00b0C, Humidity: {:g}%".format(temperature, humidity))
        sensor_data['temperature'] = temperature
        sensor_data['humidity'] = humidity

        # Sending humidity and temperature data to ThingsBoard
        client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)

        next_reading += INTERVAL
        sleep_time = next_reading-time.time()
        if sleep_time > 0:
            time.sleep(sleep_time)
except KeyboardInterrupt:
    pass

client.loop_stop()
client.disconnect()

```

Para confirmar que envía datos a la plataforma tenemos que irnos a Ultima telemetria que nos permite visualizar los datos recibidos y también tenemos la opción de mostrar esos datos en WIDGET.

