Práctica 2 GSM y MQTT – Diplomatura IoT (Internet of Things)

Bienvenidos a la segunda práctica de la diplomatura IoT. En este documento se van a describir los elementos a utilizar, y la manera en que se va a proceder para llevar a cabo la práctica que veremos en el video demostrativo.

Tabla de contenido

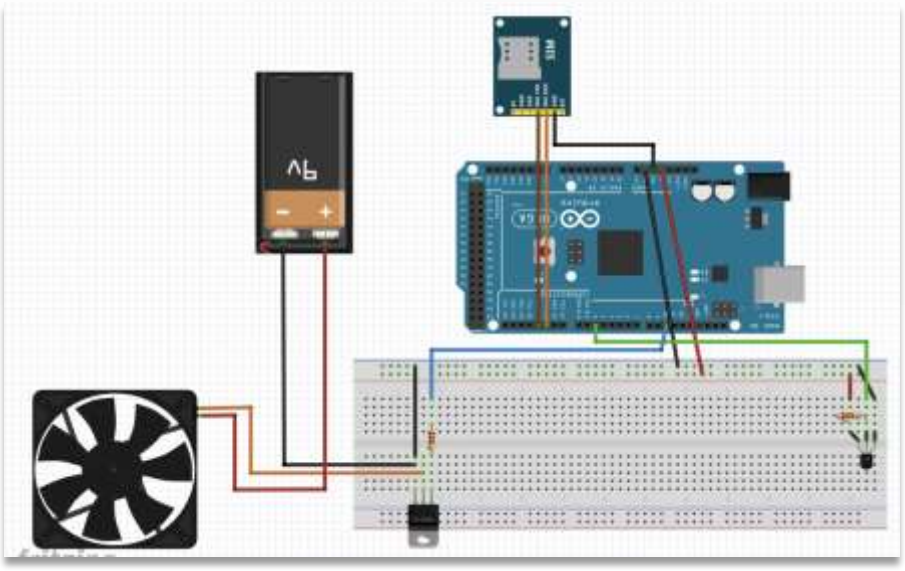
Hardware a utilizar

Para llevar a cabo la práctica, vamos a necesitar los siguientes elementos:

- Placa Arduino Mega 2560
- Módulo SIM808
- Sensor sumergible de temperatura y humedad DS18B20
- Resistencias PullUp/PullDown de 4.7k
- Cables macho a macho y macho a hembra para interconectar las placas
- Protoboard
- Transistor 31c
- Ventilador de CPU (de 5 o 12v)
- Fuente de 12v para alimentar el ventilador

Esquema de conexión

El esquema que se muestra a continuación podrán encontrarlo en los archivos adjuntos correspondientes a ésta práctica para poder analizar las conexiones en detalle. En este caso, estamos utilizando una fuente (batería) de 9v para representar la conexión al fan cooler, pero mencionamos que en la práctica trabajaremos con una fuente de 12v



Software a utilizar

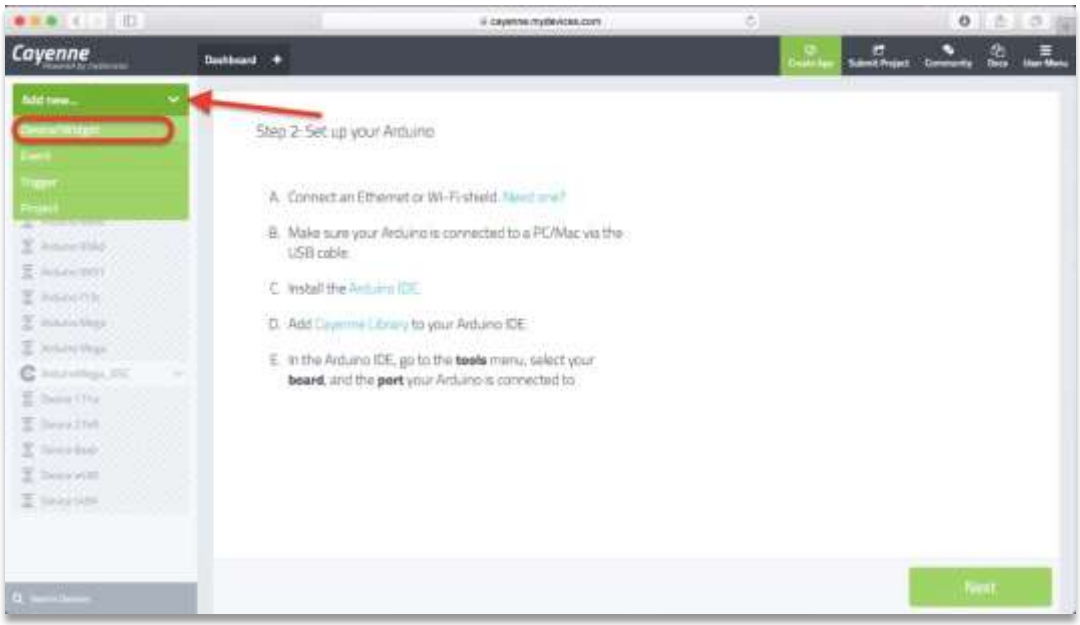
Arduino IDE: https://www.arduino.cc/en/main/software

Broker MQTT y Dashboard de Cayenne: https://cayenne.mydevices.com/cayenne/login

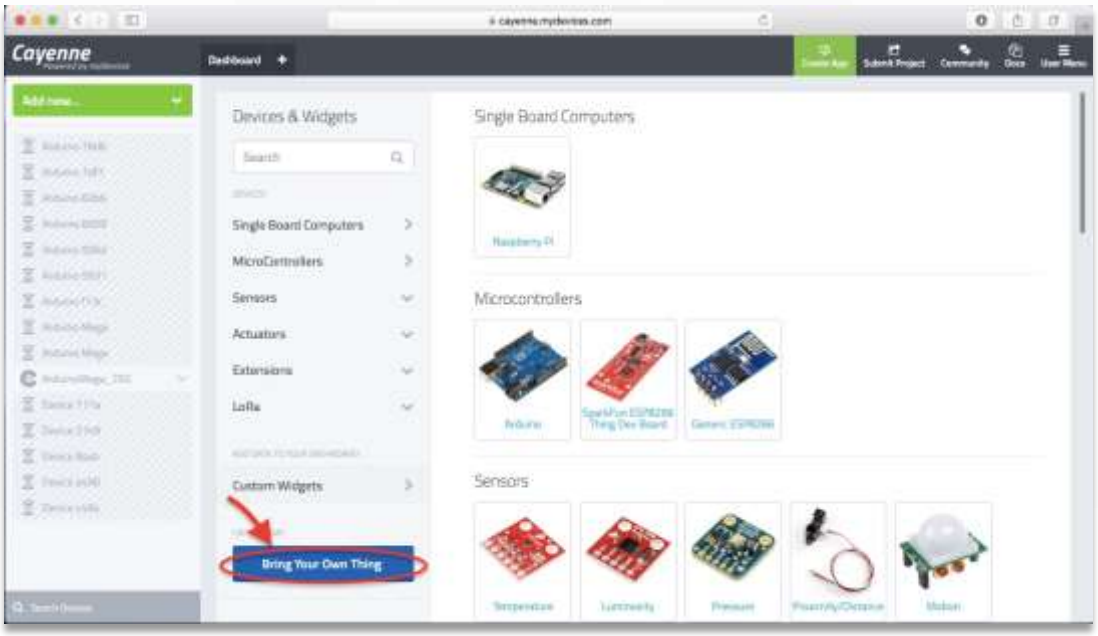
Deberemos crear una nueva cuenta en la opción "SingUp" que figura en la parte inferior

Configuración del dashboard y código

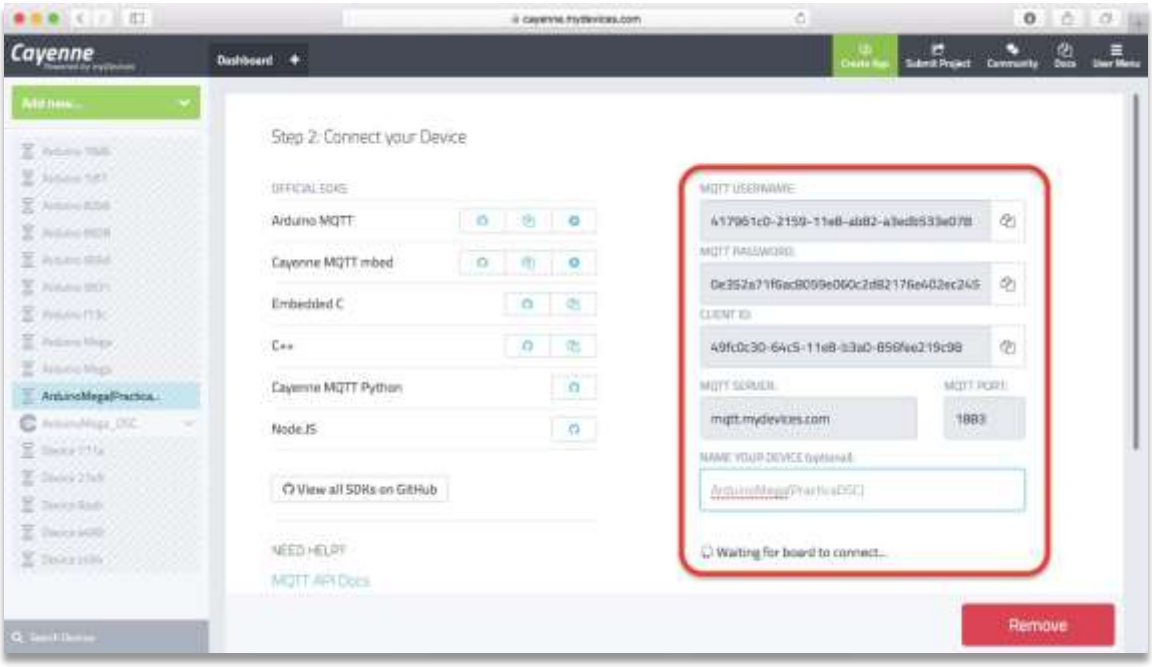
Una vez creada la cuenta, ingresamos y procedemos a agregar nuestro dispositivo de la siguiente manera:



En la siguiente pantalla vamos a seleccionar el botón "Bring Your Own Thing"



A continuación, podremos visualizar los datos que vamos a necesitar para conectar nuestra placa al servidor o Broker MQTT



La información a tener en cuenta para configurar en nuestro código es la que figura en el recuadro rojo

MQTT USERNAME, MQTT PASSWORD, CLIENT ID, MQTTSERVER, y MQTT PORT.

En el campo "NAME YOUR DEVICE" podemos escribir un nombre que nos sirva para luego poder identificar nuestra placa en la lista de equipos ubicada en el panel izquierdo del sitio.

Los datos que hemos obtenido en el sitio deberemos agregarlos en nuestro código como se muestra a continuación:

```
const char* broker = "mqtt.mydevices.com";
```

y en la función **setServer** debería quedar

```
mqtt.setServer(broker, 1883);
```

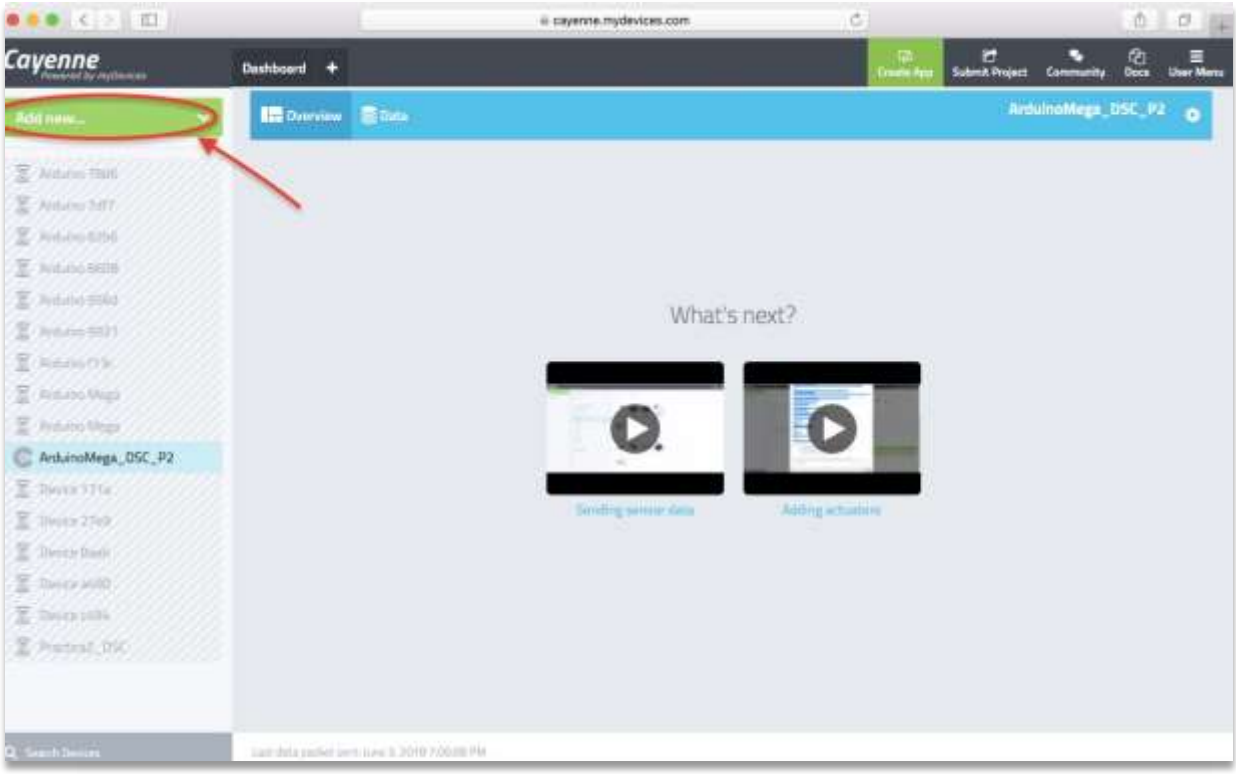
Pero con eso no es suficiente, ya que deberemos facilitar los datos adquiridos a la función **mqttConnect** de la siguiente manera:

```
mqtt.connect recibe (const char ClientId, const char UserID, const char Passowrd)
```

Es decir, dentro del paréntesis de la función vamos a completar con los datos ya mencionados.

Verificamos que dentro de nuestro método loop() esté comentada la línea que ejecuta la función `//mqtt.publish(topicTemp, mensaje);` ya que todavía no vamos hacer ningún publish al momento, solo establecer conexión con el servidor.
Con estas funciones completas con sus datos, compilamos el programa y lo subimos a nuestro Arduino.

Si pudimos establecer conexión correctamente con Cayenne, pasaremos a ver la siguiente página:

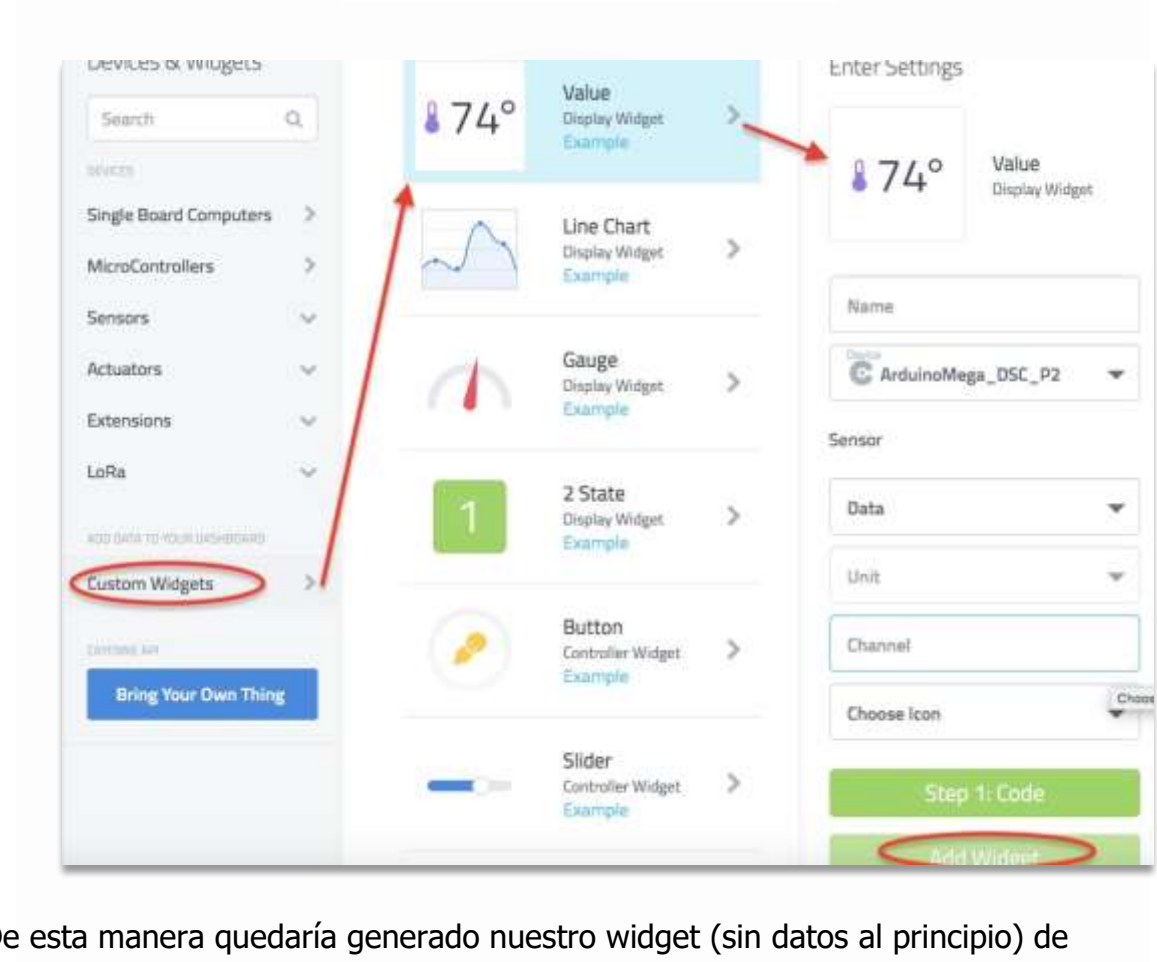


Es momento de agregar un widget a nuestro dashboard, para eso nos dirigimos nuevamente al botón Add new, y luego click en la opción Device/Widget.
Procedemos de la siguiente manera:

1. Seleccionar el botón "Custom Widgets"
2. Seleccionar widget del tipo "Value"

3. Configurar los datos del widget escribiendo el nombre a mostrar como título (puede ser "temperatura"), elegir la placa desde dónde reporta, en tipo de sensor debe quedar seleccionado "Temperatura", el tipo de unidad es "Celsius", en el canal escribiremos "1" (sin las comillas), y el ícono es a elección.
4. Una vez que completamos la configuración del widget, hacemos clic en "Add widget"

   Los pasos anteriores se describen en la imagen a continuación.



De esta manera quedaría generado nuestro widget (sin datos al principio) de temperatura en el dashboard. Por lo tanto, volvemos al código a la línea donde habíamos comentado la función del publish de temperatura y la decomentamos, quedando de la siguiente manera: `mqtt.publish(topicTemp, mensaje);`

A su vez, debemos completar el topic correspondiente al widget de temperatura. La

variable que almacena esos datos será "topicTemp" y está definida al inicio del código, y su estructura según nos indica Cayenne debe respetar el siguiente formato:

v1/username/things/clientID/data/channel

Los valores a reemplazar son los mismos con los que contamos, el username, clientID, y channel (configurado en nuestro widget), entonces la variable donde vamos a guardar este topic debe verse como se muestra a continuación:

```
//Publish sobre el widget de temperatura
const char * topicTemp = "v1/417961c0-2159-11e8-ab82-a3edb533e078/things/3cfad960-6774-11e8-a25e-d5e347246797/data/1";
```

Realizada estas modificaciones, compilamos el código, y si éste no muestra errores, lo subimos a nuestro Arduino.

Si todo ha iniciado correctamente, según podremos ver en la consola de arduino IDE, vamos a notar en el dashboard de Cayenne que el widget de temperatura ha comenzado a recibir datos, y este los irá actualizando cada 10 segundos, correspondiente al tiempo que nosotros definimos en el delay del loop()

Hemos logrado agregar, configurar y hacer publish sobre nuestro primer widget; ahora lo que queremos hacer es controlar nuestro hardware (fan cooler) desde el dashboard, y esto lo haremos mediante la función **mqttCallback** definida en nuestro código.

Volveremos al principio de nuestro código, y encontraremos los siguientes topics:

```
//Subscripcion al Widget de control del ventilador(actuator)
const char* topicFan = "v1/417961c0-2159-11e8-ab82-a3edb533e078/things/3cfad960-6774-11e8-a25e-d5e347246797/cmd/2";
//Publish del resultado del comando enviado por el control del led (actuator)
const char* topicEstadoFan = "v1/417961c0-2159-11e8-ab82-a3edb533e078/things/3cfad960-6774-11e8-a25e-d5e347246797/data/2";
//Publish del response al actuator sobre el sistema, si resulto OK o en Error
const char* response = "v1/417961c0-2159-11e8-ab82-a3edb533e078/things/3cfad960-6774-11e8-a25e-d5e347246797/response";
```

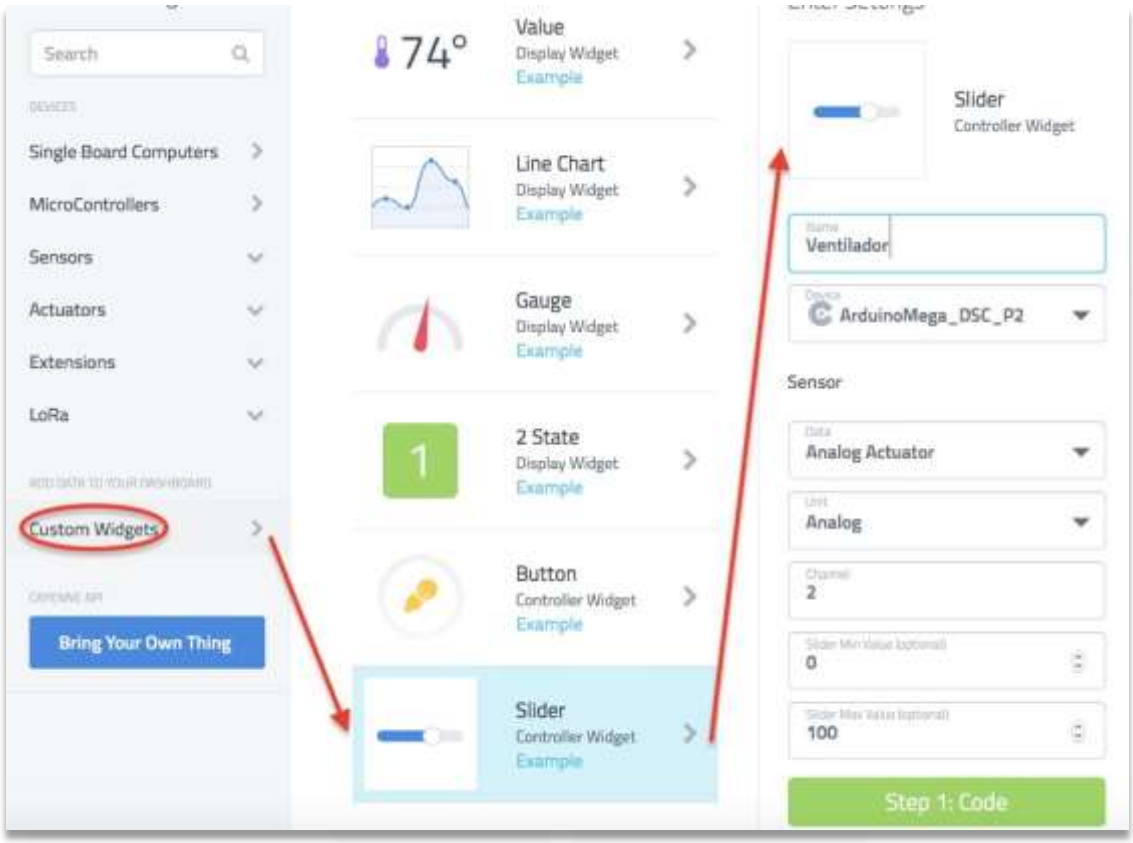
El formato para subscribirse a un widget del tipo actuator será el siguiente:
v1/username/things/clientID/cmd/channel

Éste funciona de la siguiente manera, nuestro dispositivo se subscribe al broker MQTT, desde Cayenne modificamos nuestro valor en el widget y éste hace un publish del valor hacia nuestro dispositivo para luego aplicarlo al ventilador (en este caso). Nuestro dispositivo debe responder a través de un publish el valor que tiene actualmente nuestro ventilador, y otro publish para confirmar que hemos hecho el cambio.
El publish del valor aplicado es similar al que hemos visto antes, pero el del response es distinto y más básico, ya que no requiere un canal para transmitirlo. Su nomenclatura es la siguiente:

v1/username/things/clientID/response

Procedemos a completar los datos en los topics que corresponden, compilamos el código y lo subimos a nuestro Arduino.

Es momento de agregar un nuevo widget desde el cuál controlaremos nuestro ventilador. Para ello seguimos los pasos de la imagen adjunta a continuación:



Para la configuración del widget del tipo Slider, en la parte de Sensor, seleccionaremos el tipo de dato como "Analog Actuator", en unidad elegimos "Analog", el canal será el "2", y los valores del slider de mínimo a máximo irán de 0 a 100. Al finalizar hacemos clic en "Add widget" y ya tendremos nuestro widget Slider en el dashboard.

A partir de ahora, el valor que dejemos seleccionado en nuestro widget Slider será el valor que enviaremos a nuestro Arduino, para que éste lo envíe como pulso a nuestro ventilador y así incrementar o disminuir su velocidad.