

5. Recibe los datos del ESP en NodeRED con MQTT

Hasta ahora todo bastante facil, ¿verdad? Vamos a complicarlo un poco más!

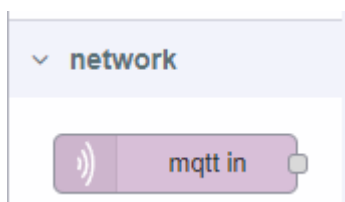
Ahora vamos a graficar mediante los nodos de dashboard los datos de los sensores que recibimos mediante MQTT. Para ello deberás usar los nodos mqtt-in y mqtt-out para subscribirte a los topic de MQTT correspondientes.

Vamos a paso a paso para que no te pierdas:

1. Conexión al broker MQTT

En primer lugar nos vamos a conectar al broker servidor MQTT de donde vamos a enviar/recibir datos, al igual que lo hemos hecho con el ESP8266 en Arduino. Recuerda que tanto el dispositivo Arduino como el NodeRED deben estar conectados al mismo servidor MQTT, si no no podrán intercambiar información.

Para ello empieza introduciendo en el workspace un nodo **mqtt in** y presiona doble click para entrar en la configuración:



Te saldrá totalmente vacío y listo para añadir un nuevo mqtt-broker. Haz click en el botón de **editar**:

Edit mqtt in node

Delete
Cancel
Done

⚙️ Properties

⚙️
📄
🔍

Server
Add new mqtt-broker...

Action
Subscribe to single topic

Topic
Topic

QoS
2

Output
auto-detect (parsed JSON object, string or buf

Name
Name

E introduce los datos de conexión para el broker MQTT de MakersUPV:

Datos conexión MQTT

Server MQTT	mqtt.makersupv.com
Port	1883
Client ID	(blank)
Username	makersupv
Password	makersupv

Asegurate de poner bien el usuario y contraseña, si no no te funcionará!

Te debería quedar de la siguiente manera:

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel Update

Properties

Name Broker MakersUPV

Connection Security Messages

Server mqtt.makersupv.com Port 1883

☒ Connect automatically
☐ Use TLS

Protocol MQTT V3.1.1

Client ID Leave blank for auto generated

Keep Alive 60

Session ☒ Use clean session

Edit mqtt in node > **Add new mqtt-broker config node**

Cancel Add

Properties

Name Name

Connection Security Messages

Username makersupv

Password

Haz click en el botón de Añadir para guardar el broker MQTT. Tan solo tendrás que realizar este procedimiento la primera vez (con el primer nodo que añadas)

2. Subscripción y publicación al tópic MQTT

Una vez guardado el servidor MQTT, posteriormente rellena el topic de subscripción para el sensor de temperatura:

Edit mqtt in node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

🌐 Server

Broker MakersUPV

✎

Action

Subscribe to single topic

▼

📄 Topic

taller/t00/sens/DHT11/temp

⚙ QoS

2

▼

➡ Output

auto-detect (parsed JSON object, string or buf

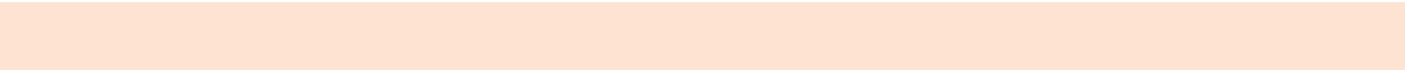
▼

🏷 Name

DHT11 Temperatura

Estos son los temas (topics) que debes usar:

Tipo	Descripción	Unidades	Topic MQTT
Sensor	Lectura sensor temperatura DHT11	°C	taller/ tXX /sens/DHT11/temp
Sensor	Lectura sensor humedad relativa DHT11	%rh	taller/ tXX /sens/DHT11/hum
Sensor	Lectura sensor luminosidad LDR	Volts	taller/ tXX /sens/LDR
Actuador LED	Indicador LED Verde		taller/ tXX /leds/verde
Actuador LED	Indicador LED Rojo		taller/ tXX /leds/rojo



Recuerda cambiar **t00** por el número de tu equipo

Haz click en **Done**. Se recomienda que en este punto añadas un nodo de debug y hagas un deploy para comprobar que se están recibiendo los datos del ESP8266 correctamente antes de pasar a los demás.

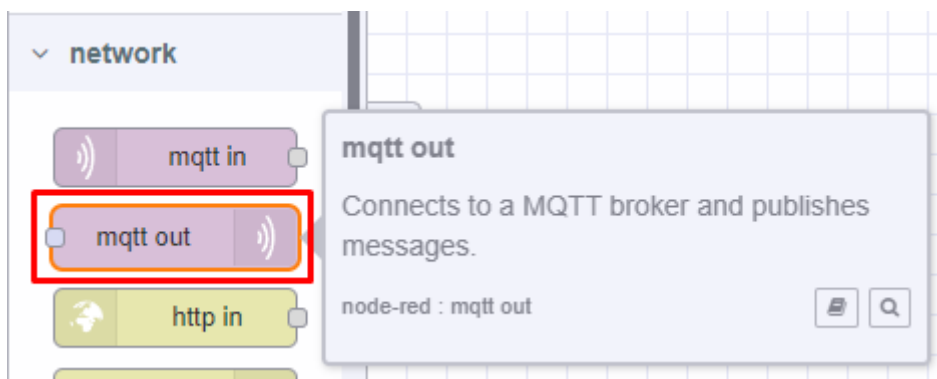
Una vez te hayas asegurado de que se reciben correctamente los datos, repite el proceso para los demás sensores (humedad y LDR),

Recuerda que puedes usar Copiar y Pegar (Ctrl+C, Ctrl+V) para los nodos

Ah! y para los LEDs... Deberás usar ¿que nodo?

Solución: ¿Que nodo uso para los LEDs?

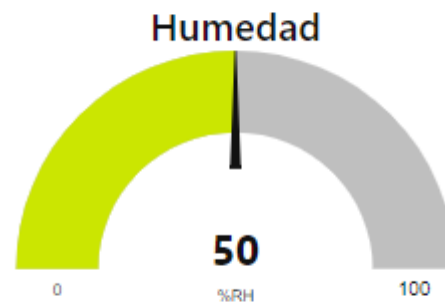
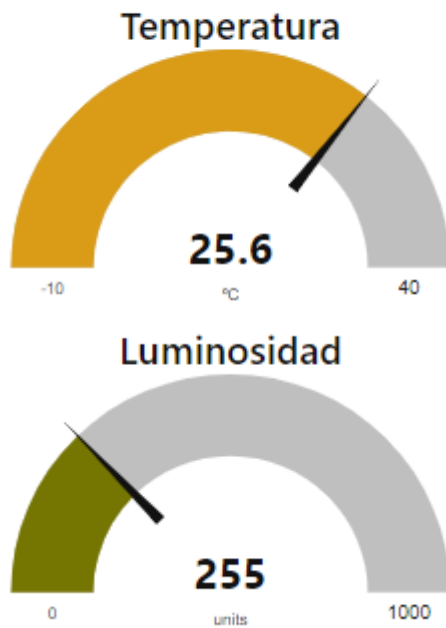
Para los LEDs deberás usar el nodo **mqtt-out**, ya que lo que queremos es que el mensaje se mande dirección salida hacia el Arduino. Es decir, queremos usarlo para publicar un mensaje al topic correspondiente.



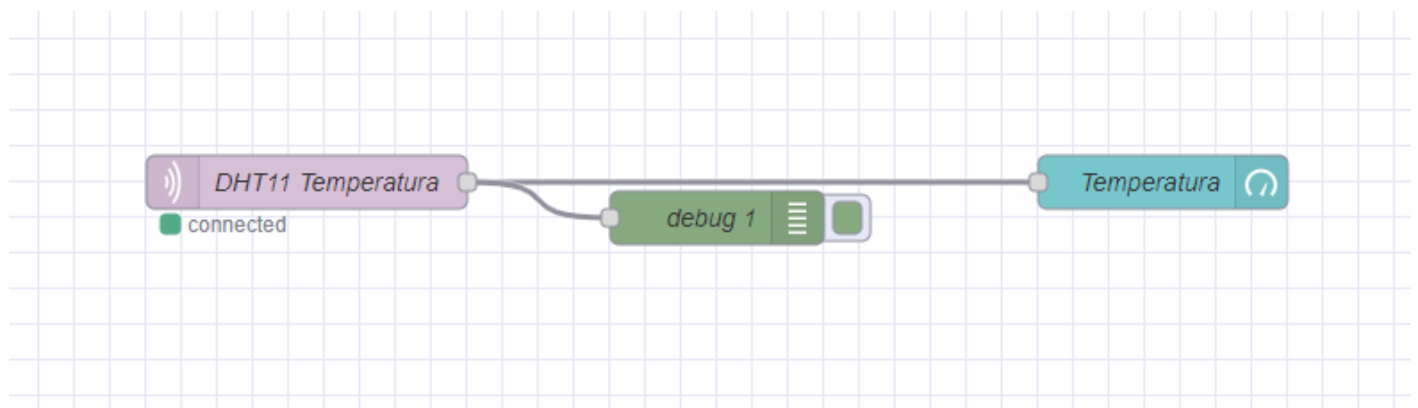
3. Gauges

Las *gauges*, o indicadores en español, nos permiten mostrar un dato de manera visual mediante un indicador tipo aguja, similar a por ejemplo el velocímetro de un coche o un manómetro de los analógicos.

Sensores



Ya los hemos usado anteriormente en el ejemplo con el Timestamp, pero no los hemos configurado. Comenzamos por añadir uno y conectarlo al nodo mqtt-in de la temperatura:



Ahora vamos a configurarlo según los requisitos específicos de nuestra aplicación. Para ello haz click en el nodo correspondiente y cambia los valores de la etiqueta, unidades, rango mínimo y máximo, colores del gradiente, etc. También podrás cambiar entre los distintos tipos de indicador (tipo aguja, donut, compás...)

Te recomiendo usar los nodos de **Debug** para cerciorarte que los datos llegan y en el formato que esperas

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

🗃 Group

[ESP8266] Sensores

✎

📏 Size

6 x 3

☰ Type

Gauge

▼

🏷 Label

Temperatura

🏷 Value format

{{value}}

🏷 Units

°C

Range

min

-10

max

40

Colour gradient

Sectors

-10

...

optional

...

optional

...

40

Fill gauge from centre.

☐

📄 Class

Optional CSS class name(s) for widget

🏷 Name

Temperatura

Experimenta las opciones que te da el nodo de Gauge hasta que encuentres la opción que más te guste para representar la temperatura! Recuerda que deberás de pinchar en el botón de **Deploy** para desplegar los cambios cada vez que edites algo.

Comprueba que los datos se grafican como toca y que el gráfico se va actualizando en tiempo real según van variando las condiciones climáticas

Utiliza tu mano para calentar el sensor de temperatura DHT11 y tirale vaho con la boca para ver como aumenta la humedad relativa

Una vez tengas el de temperatura funcionando deberás hacer lo mismo para el sensor de humedad y para el de luminosidad.

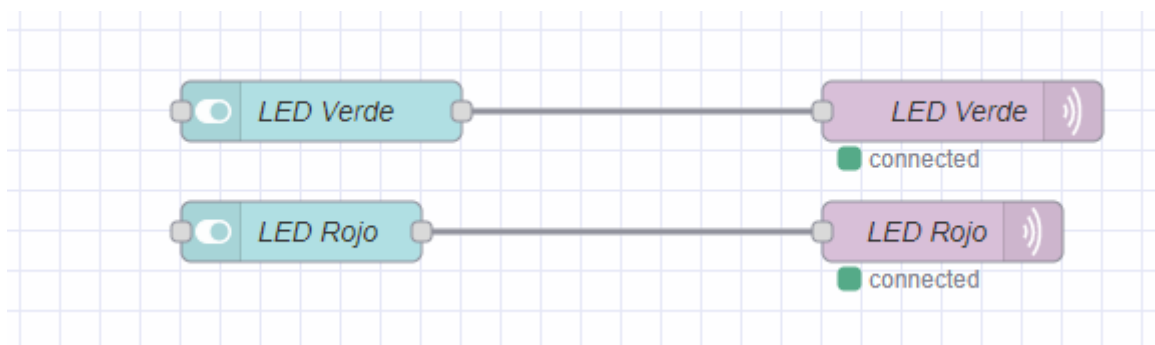
Cuando hayas acabado, el resultado final debería ser algo parecido a la imagen de arriba, con un Gauge mostrando cada uno de los valores de los sensores.

4. Interruptores/botones

Ahora nos faltaría introducir una manera de encender y apagar los LEDs o actuadores de nuestro circuito. En este ejemplo sencillo que estamos haciendo para el taller son simples LEDs, pero podrían ser perfectamente bombillas, una lámpara, un enchufe inteligente, un relé conectado al motor de la depuradora de casa, la calefacción o cualquier otra cosa, ya pillas la idea...

Para encender y apagar los LED vamos a utilizar el nodo Switch, el cual no es más que un simple interruptor de dos posiciones: ON y OFF.

Será tan facil como añadir un interruptor para cada uno de los LEDs y conectarlo al nodo MQTT de salida correspondiente:



A continuación deberás configurar los nodos interruptor para que manden un 1 cuando el estado sea ON y un 0 cuando sea OFF. De la siguiente manera:

Edit switch node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖼️

📁 Group

[ESP8266] Actuadores

✎

📏 Size

4 x 2

🏷️ Label

LED Verde

📖 Tooltip

optional tooltip

🖼️ Icon

Default

▼

➡️ Pass through msg if payload matches valid state:

☐

🔘 Indicator

Switch icon shows state of the output

▼

☑️ When clicked, send:

On Payload

▼ 0₉ 1

Off Payload

▼ 0₉ 0

Topic

▼ msg. topic

</> Class

Optional CSS class name(s) for widget

🏷️ Name

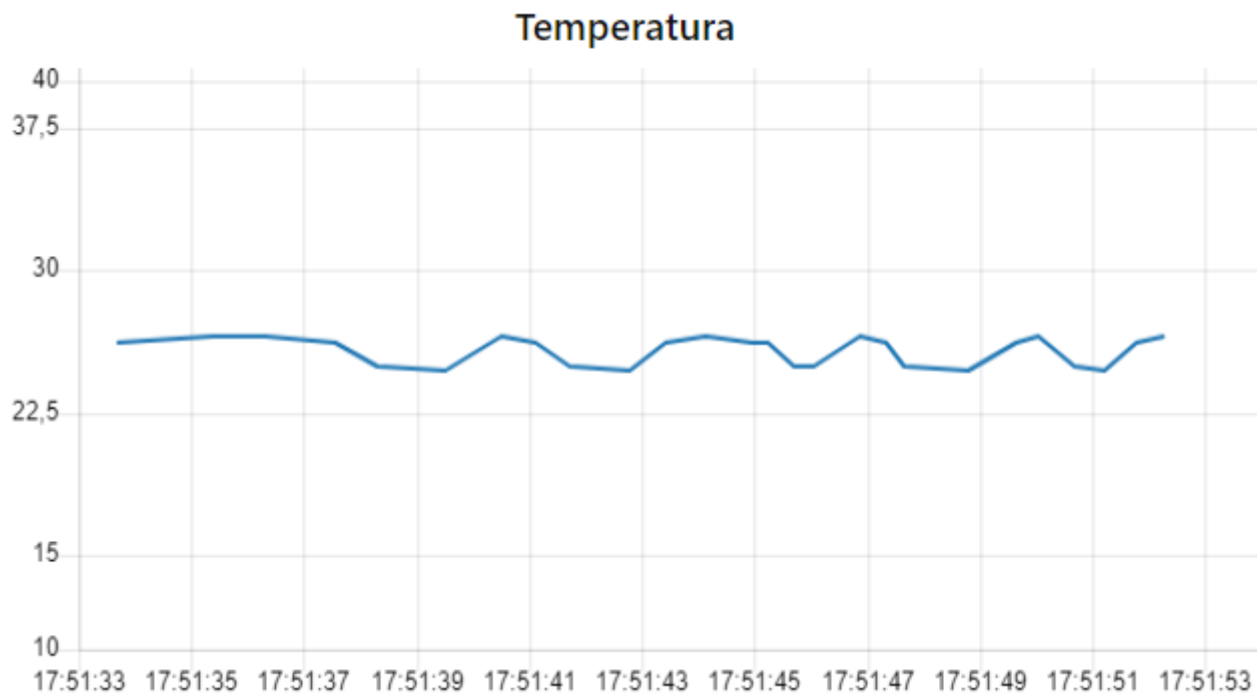
LED Verde

Ejecutalo y prueba el resultado.

5. Gráfico temporal

Las gauges están muy bien para ver los datos en tiempo real, pero quizá te interese ver la evolución de la temperatura a lo largo de unas horas, ¿no? (o de unos minutos, para que nos dé tiempo a verlo en la clase)

Vamos a usar los gráficos (graph) para poder representar datos en un eje temporal:



Prueba a añadir un nodo Chart

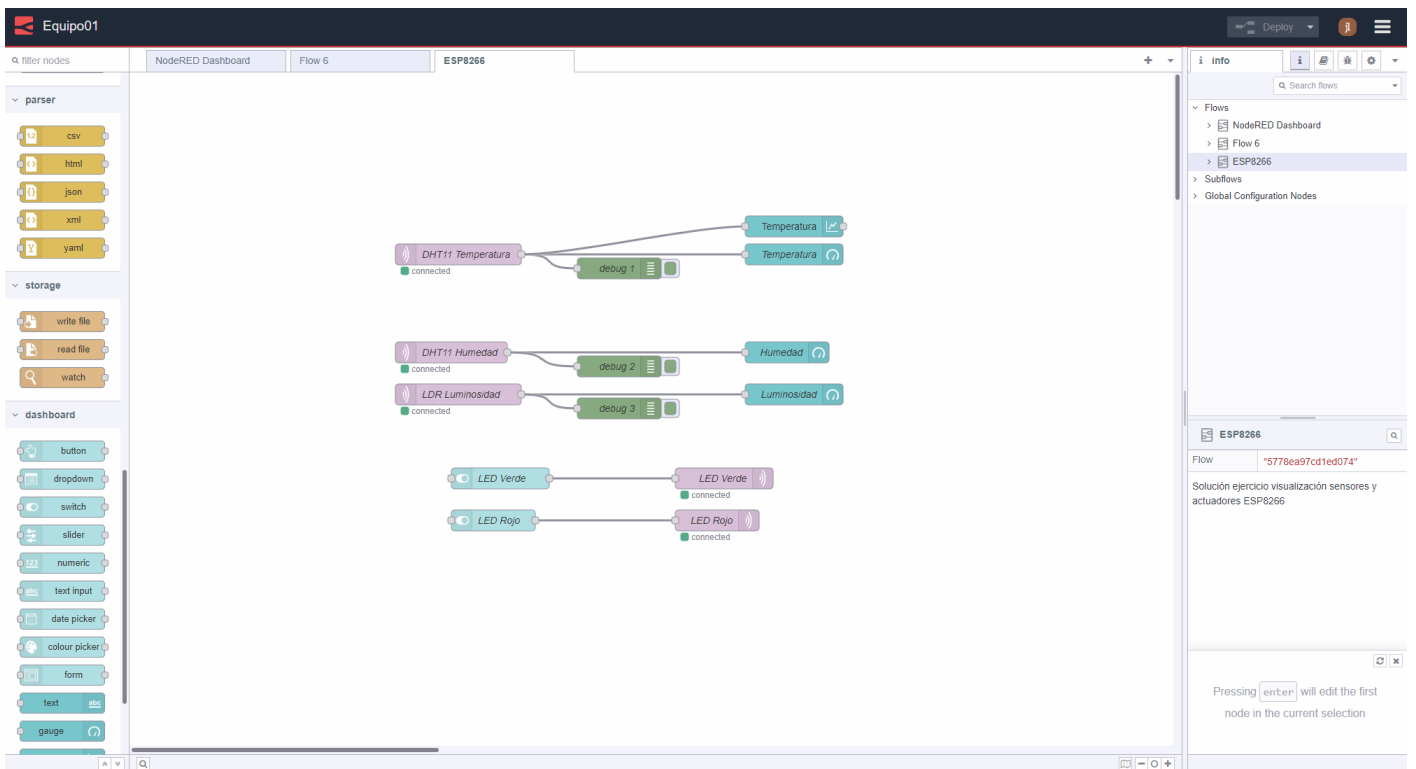


Cambia las opciones para que aparezca donde te interese y despliega los cambios. Esperate a que lleguen datos de los sensores y verás como se va rellenando el gráfico en tiempo real.

Cabe destacar que estos gráficos son muy sencillos ya que no guardan los datos en ninguna base de datos y por tanto están bastante limitados. Pero para cosas sencillas nos valen

5. Resultado final

Una vez tengas todo, el resultado final debería ser algo así:



Te animo a que experimentes... Seguro que no te cuesta nada conseguir un resultado MUUUUUUCHO mejor que el mio!

¿Y si quisiéramos guardar las métricas en una base de datos?

Pues eso daría para otra sesión de un par de horitas seguramente, pero NodeRED dispone de plugins creados para almacenar datos en una base de datos temporal tipo InfluxDB por ejemplo, o incluso también permite hacer enlaces con bases de datos SQL y NoSQL como MongoDB, etc.

Otra manera de guardar los datos podría ser en un archivo CSV separados por coma mediante el nodo de guardar en fichero, de esa manera los podríamos tener guardados en un archivo en el disco, aunque lo suyo para cosas serias es usar una base de datos como dios manda

Solución [No abrir, no seas tramposo]

```
[
  {
    "id": "5778ea97cd1ed074",
    "type": "tab",
    "label": "ESP8266",
    "disabled": false,
    "info": "Solución ejercicio visualización sensores y actuadores ESP8266",
    "env": []
  },
  {
    "id": "02973140566ef55c",
    "type": "mqtt in",
    "z": "5778ea97cd1ed074",
    "name": "DHT11 Temperatura",
    "topic": "taller/t00/sens/DHT11/temp",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "b5bdeb4155381a5d",
    "nl": false,
    "rap": true,
    "rh": 0,
```

```
    "inputs": 0,
    "x": 450,
    "y": 160,
    "wires": [
      [
        "7aa944fea3c07822"
      ]
    ]
  },
  {
    "id": "609bc96d69db82e8",
    "type": "mqtt in",
    "z": "5778ea97cd1ed074",
    "name": "DHT11 Humedad",
    "topic": "taller/t00/sens/DHT11/hum",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "b5bdeb4155381a5d",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 440,
    "y": 220,
    "wires": [
      [
        "9e44099297dcc3a5"
      ]
    ]
  },
  {
    "id": "e0f0357e049277d2",
    "type": "mqtt in",
    "z": "5778ea97cd1ed074",
    "name": "LDR Luminosidad",
    "topic": "taller/t00/sens/LDR",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "b5bdeb4155381a5d",
```

```
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 450,
    "y": 280,
    "wires": [
      [
        "280c63846cb13519"
      ]
    ]
  },
  {
    "id": "d34f1d5a1ba21375",
    "type": "mqtt out",
    "z": "5778ea97cd1ed074",
    "name": "LED Verde",
    "topic": "taller/t00/leds/verde",
    "qos": "",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "b5bdeb4155381a5d",
    "x": 830,
    "y": 400,
    "wires": []
  },
  {
    "id": "971b18f52f42468f",
    "type": "mqtt out",
    "z": "5778ea97cd1ed074",
    "name": "LED Rojo",
    "topic": "taller/t00/leds/rojo",
    "qos": "",
    "retain": "",
    "respTopic": "",
```

```
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "b5bdeb4155381a5d",
    "x": 820,
    "y": 460,
    "wires": []
  },
  {
    "id": "7aa944fea3c07822",
    "type": "ui_gauge",
    "z": "5778ea97cd1ed074",
    "name": "Temperatura",
    "group": "3de4d1022cadb73f",
    "order": 0,
    "width": "6",
    "height": "3",
    "gtype": "gage",
    "title": "Temperatura",
    "label": "°C",
    "format": "{{value}}",
    "min": "-10",
    "max": "40",
    "colors": [
      "#00b3b0",
      "#e6e600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "diff": false,
    "className": "",
    "x": 930,
    "y": 160,
    "wires": []
  },
  {
    "id": "9e44099297dcc3a5",
```

```
    "type": "ui_gauge",
    "z": "5778ea97cd1ed074",
    "name": "Humedad",
    "group": "3de4d1022cadb73f",
    "order": 1,
    "width": "6",
    "height": "3",
    "gtype": "gage",
    "title": "Humedad",
    "label": "%RH",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
      "#1500b3",
      "#cbe600",
      "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "diff": false,
    "className": "",
    "x": 920,
    "y": 220,
    "wires": []
  },
  {
    "id": "280c63846cb13519",
    "type": "ui_gauge",
    "z": "5778ea97cd1ed074",
    "name": "Luminosidad",
    "group": "3de4d1022cadb73f",
    "order": 2,
    "width": "6",
    "height": "3",
    "gtype": "gage",
    "title": "Luminosidad",
    "label": "units",
    "format": "{{value}}",
```



```
"min": 0,
"max": "1000",
"colors": [
  "#000000",
  "#e6e600",
  "#ffffff"
],
"seg1": "",
"seg2": "",
"diff": false,
"className": "",
"x": 930,
"y": 280,
"wires": []
},
{
  "id": "34e6bf482f2f4e81",
  "type": "ui_switch",
  "z": "5778ea97cd1ed074",
  "name": "LED Verde",
  "label": "LED Verde",
  "tooltip": "",
  "group": "71d9266564131fe1",
  "order": 0,
  "width": "4",
  "height": "2",
  "passthru": false,
  "decouple": "false",
  "topic": "topic",
  "topicType": "msg",
  "style": "",
  "onvalue": "1",
  "onvalueType": "num",
  "onicon": "",
  "oncolor": "",
  "offvalue": "0",
  "offvalueType": "num",
  "officon": "",
  "offcolor": "",
```

```
    "animate": true,
    "className": "",
    "x": 510,
    "y": 400,
    "wires": [
      [
        "d34f1d5a1ba21375"
      ]
    ]
  },
  {
    "id": "6a9f570b1bcbf20c",
    "type": "ui_switch",
    "z": "5778ea97cd1ed074",
    "name": "LED Rojo",
    "label": "LED Rojo",
    "tooltip": "",
    "group": "71d9266564131fe1",
    "order": 1,
    "width": "4",
    "height": "2",
    "passthru": true,
    "decouple": "false",
    "topic": "topic",
    "topicType": "msg",
    "style": "",
    "onvalue": "1",
    "onvalueType": "num",
    "onicon": "",
    "oncolor": "",
    "offvalue": "0",
    "offvalueType": "num",
    "officon": "",
    "offcolor": "",
    "animate": false,
    "className": "",
    "x": 500,
    "y": 460,
    "wires": [
```

```
        [
            "971b18f52f42468f"
        ]
    ],
    {
        "id": "b5bdeb4155381a5d",
        "type": "mqtt-broker",
        "name": "Broker MakersUPV",
        "broker": "mqtt.makersupv.com",
        "port": "1883",
        "clientid": "",
        "autoConnect": true,
        "usetls": false,
        "protocolVersion": "4",
        "keepalive": "60",
        "cleansession": true,
        "birthTopic": "",
        "birthQos": "0",
        "birthPayload": "",
        "birthMsg": {},
        "closeTopic": "",
        "closeQos": "0",
        "closePayload": "",
        "closeMsg": {},
        "willTopic": "",
        "willQos": "0",
        "willPayload": "",
        "willMsg": {},
        "userProps": "",
        "sessionExpiry": ""
    },
    {
        "id": "3de4d1022cadb73f",
        "type": "ui_group",
        "name": "Sensores",
        "tab": "be3445ce783f56f1",
        "order": 1,
        "disp": true,
```

```
    "width": "12",
    "collapse": false,
    "className": ""
  },
  {
    "id": "71d9266564131fe1",
    "type": "ui_group",
    "name": "Actuadores",
    "tab": "be3445ce783f56f1",
    "order": 2,
    "disp": true,
    "width": "12",
    "collapse": false,
    "className": ""
  },
  {
    "id": "be3445ce783f56f1",
    "type": "ui_tab",
    "name": "ESP8266",
    "icon": "fa-thermometer-half",
    "disabled": false,
    "hidden": false
  }
]
```

Revision #16

Created 9 April 2023 19:13:52 by Jaime Laborda

Updated 11 May 2023 14:34:37 by Jaime Laborda