

Implementación de sensores internos de la colmena

Resumen

En esta sección del proyecto Habeetat se construye el módulo encargado de medir la temperatura, humedad y concentración de CO2 dentro de la colmena. Se usa un microcontrolador ESP32-WROOM, un sensor de temperatura-Humedad DHT11 y un sensor de gases MQ135. El microcontrolador se programa para que mida las 3 variables y publique mediante el protocolo MQTT en un broker local, al mismo tiempo, se crea una base de datos con MySQL donde se guardan las lecturas de los sensores usando un flow creado en NodeRed; este flow nos ayuda a guardar las lecturas de los sensores en la base de datos y enviar notificaciones a un chat de Telegram usando un bot, el bot se programa para que reciba instrucciones y regrese consultas de la bases de datos, además de enviar notificaciones del estado de la colmena (alertas y datos) cada cierto periodo de tiempo. Los datos históricos y actuales recopilados son mostrados en gráficos en un dashboard de Grafana.

Software y bibliotecas a utilizar

- IDE de Arduino 1.8.20 <https://www.arduino.cc/en/software>
- NodeRed v3.0.2 <https://nodered.org/>
- Mosquitto 2.0.15 <https://mosquitto.org/>
- MySQL 8.0.30 - 0ubuntu0.22.04.1 (Ubuntu) <https://ubuntu.com/server/docs/databases-mysql>
- Grafana v9.1.7 <https://grafana.com/>
- Python 3.8 <https://www.python.org/downloads/release/python-380/>

Bibliotecas de Arduino para programar el microcontrolador y los sensores

- Wifi <https://github.com/arduino-libraries/WiFi>
- PubSubclient <https://github.com/knolleary/pubsubclient>
- DHT11 <https://github.com/adafruit/DHT-sensor-library>

Consideraciones para la realización del circuito, base de datos, flow de NodeRed, uso de la cámara, predicción y panel de Grafana.

Esta parte del proyecto hace uso de los siguientes archivos para su buen funcionamiento:

- [Programa del ESP32 WROOM](#)
- [Flow de node red para recibir, guardar y enviar datos, así como imágenes](#)
- [Programa para el ESP32 CAM para el envío de imágenes y vídeo al flow de Nodered](#)
- [Programa en Python para la predicción del peso de la colmena](#)

En las lecciones posteriores se explica con más detalle el funcionamiento ya que cada una de las partes interactúa entre sí para enviar y recibir la información.

Material necesario

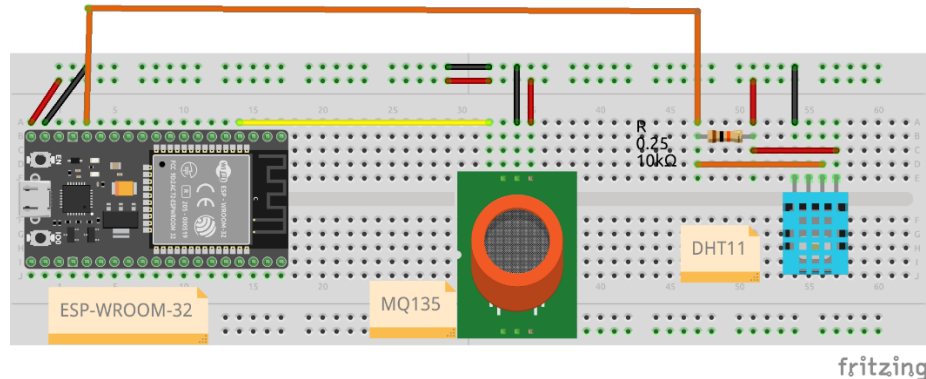
- Computadora
- 1 microcontrolador ESP32-WROOM
- 1 sensor DHT11
- 1 sensor MQ135
- 1 Protoboard
- Cables jumper MM
- 1 resistencia de 10 kilohms
- 1 cable microusb a USB

Material de referencia y servicios a usar

- Guía de instalación de Nodos de Telegram para Nodered (<https://flows.nodered.org/node/node-red-contrib-telegrambot>)
- Telegram bot API <https://core.telegram.org/bots/api>

Instrucciones para construir el circuito de medición de temperatura, humedad y concentración de CO2

El circuito a crear es el siguiente

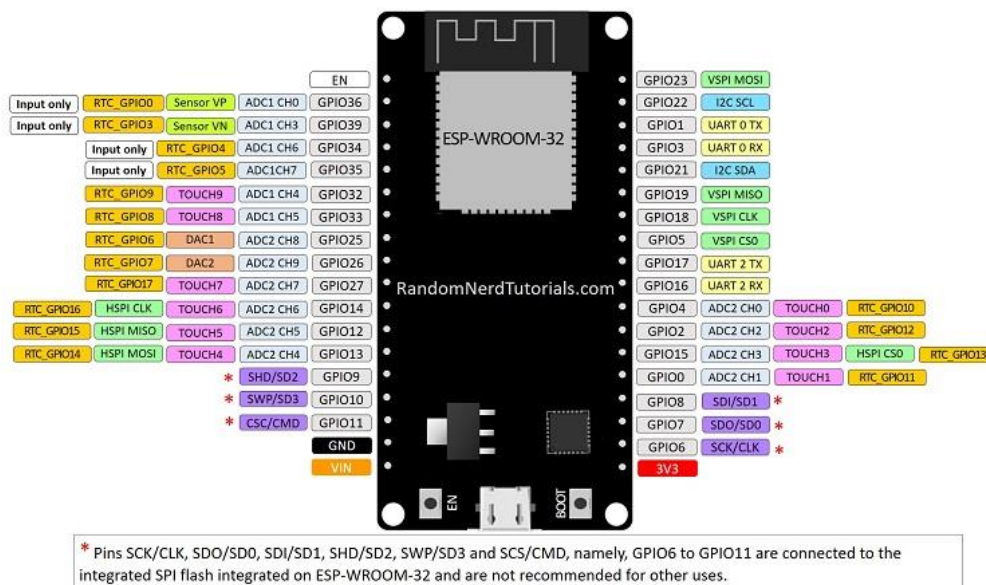


ATENCIÓN: Deben tomarse en cuenta las siguientes consideraciones para el armado del circuito:

- No energizar el circuito hasta que todo esté debidamente conectado.

Los pines del ESP32 se consideran así de acuerdo a la siguiente nomenclatura

ESP32 DEVKIT V1 – DOIT

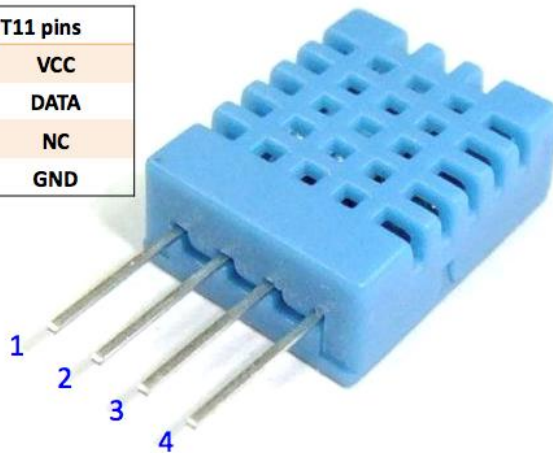


Los pines del sensor MQ135 son los siguientes:



Los pines del sensor DHT11 son los siguientes

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



- El sensor MQ135 debe conectarse al microcontrolador a través de los siguientes pines

ESP32 -- MQ135

Vcc (5V) -- Vcc

GND -- GND

GPIO 34 -- A0

- El sensor DHT11 debe conectarse al microcontrolador través de los siguientes pines

ESP32 -- DHT11

VCC (5V) -- VCC

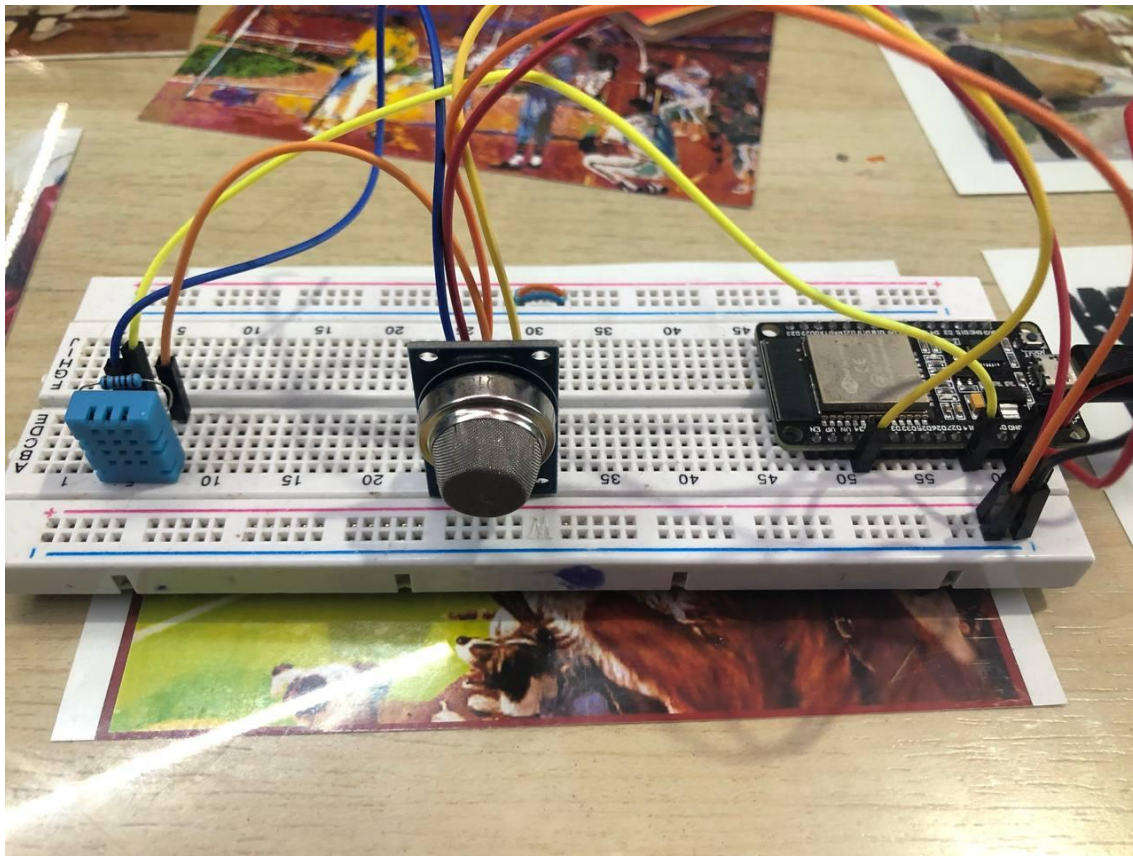
GND -- GND

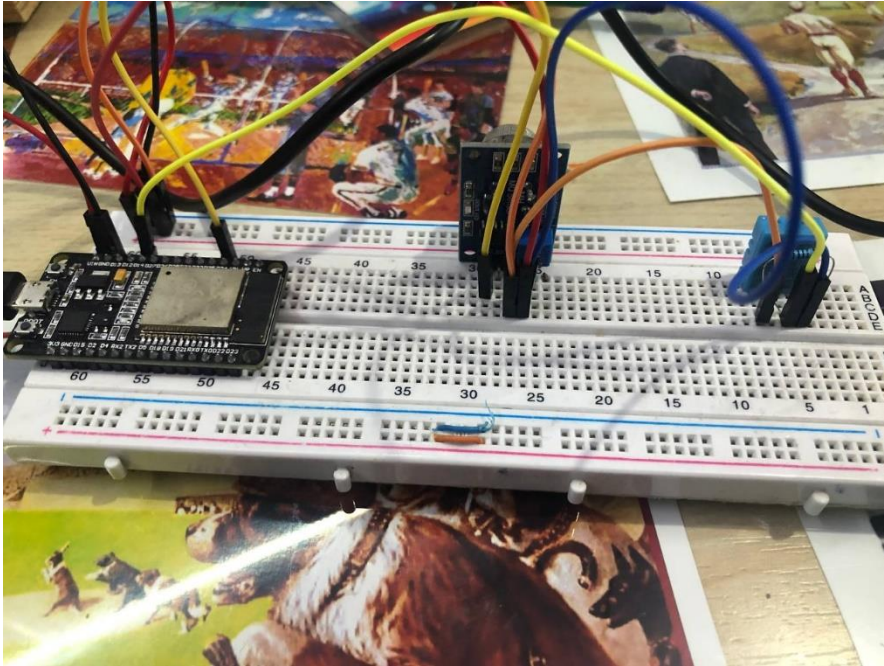
GPIO 25 (14) -- DATA (2)

- La tierra y la corriente se pueden conectar en paralelo para ser comunes.

La siguiente imagen muestra el circuito del MQ135 y el DHT11 conectados al ESP32 WROOM.

Vista frontal





Instrucciones para la programación del ESP32 WROOM

El programa para leer los sensores se encuentra en la siguiente carpeta del [repositorio de Github](#) . Este programa hace uso de las bibliotecas Wifi, PubSubClient y DHT11 por lo que es necesario instalarlas en el IDE de Arduino antes de subir el programa al ESP32, así como de preparar todo el entorno en sistema operativo (Ubuntu) para que se conecte el ESP32 WROOM a la computadora ([IDE de Arduino y ESP32](#)). Una vez que se realizan las lecturas del sensor éstas son publicadas en un broker local mediante el protocolo MQTT usando [Mosquitto](#) por lo que hay que configurarlo también en la computadora.

Una vez construido el circuito anterior se carga el programa usando el IDE de arduino,

El programa del ESP32 WROOM permite las siguientes acciones:

- Conexión del microcontrolador a Wifi
- Lectura del Voltaje analógico del sensor MQ-135 (concentración de CO₂)
- Lectura de Temperatura y Humedad con el sensor DHT11
- Publicación y suscripción a tópicos en el broker local usando MQTT
- Envío de los datos en formato JSON al broker local

Una vez conectado el ESP32 adquiriendo los datos de temperatura, humedad y CO₂ creamos un flow en NodeRed ([Link del flow de NodeRed](#)) que realiza las siguientes acciones:

- Lectura y adquisición del archivo JSON con los datos desde el broker local hecho en MQTT para:

1. Guardar los datos en una base de datos local hecha con MySQL ([Link detalles de la creación de base de datos](#)).

2. Generar reportes con el bot de telegram desde nodered ([Link de los detalles para creación del bot de telegram con node red](#)).

- El bot envía reportes programados o permite realizar consultas de los datos más recientes.

3. Envío de imágenes y vídeo con la esp32CAM y el bot de telegram desde nodered ([link del programa del esp32CAM](#)).

4. El flow de node-red también tiene un panel donde podemos visualizar vídeo en tiempo real desde la colmena, esta cámara se sitúa en el exterior de la colmena para vigilancia.

5. Envío de la predicción del peso de la colmena en los siguientes días ([Link del código de predicción](#)).

6. Cuando las condiciones de Temperatura, humedad o concentración de CO2 no son las favorables para la colmena también se envía un mensaje de alerta.

Creación de la base de datos MySQL para los datos del interior de la colmena

La base de datos se creó en una máquina virtual de Ubuntu 20.04 usando MySQL. Los datos obtenidos de los sensores MQ135 y DHT11 son enviados por MQTT en formato json al broker local donde son leídos por el flow de NodeRed y guardados en nuestra base de datos local usando el nodo MySQL. Posteriormente esta base de datos es usada como fuente de datos para un panel de Grafana donde visualizamos el comportamiento local.

Software a utilizar

Máquina virtual con Ubuntu 20.04

[My Sql para Linux](#)

Instrucciones para creación de base de datos

1. Abrir una terminal e instalar mysql server con los comandos

- `sudo apt update`
- `sudo apt install mysql-server`

2. Ejecutar mysql

- `sudo mysql`

3. Crear una nueva base de datos para guardar los datos de los sensores

- `CREATE DATABASE sensor_MQ_DHT;`

4. Seleccionar base de datos

- `use sensor_MQ_DHT;`

5. Crear una tabla llamada Datos_Sensores que contenga todos los campos necesarios

- `CREATE TABLE Datos_Sensores (n_registro INT(7) UNSIGNED AUTO_INCREMENT PRIMARY KEY, fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, id_sensor CHAR(248) NOT NULL, ValorAnalogico FLOAT(7,3) NOT NULL, VoltajeAnalogico FLOAT(7,3) NOT NULL, Temperatura FLOAT(7,3) NOT NULL, Humedad FLOAT(7,3) NOT NULL);`

Esto creará la base de datos que guardará nuestros datos que son

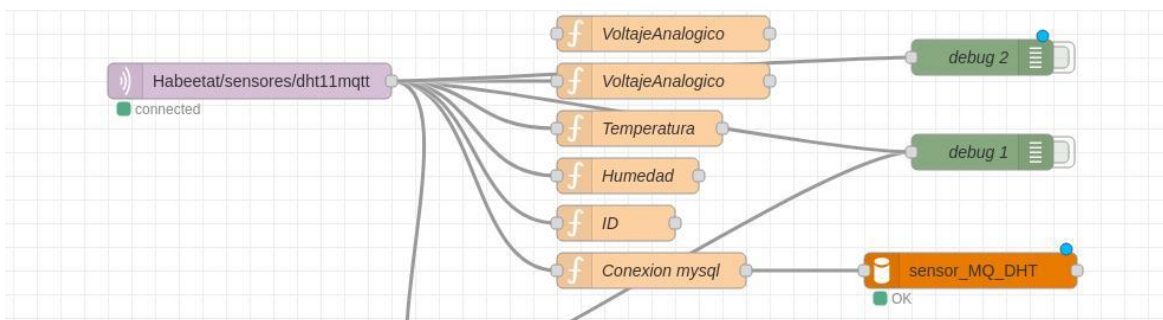
- id_sensor
- Timestamp
- ValorAnalogico
- VoltajeAnalogico
- Temperatura
- Humedad

6. Se requiere crear un usuario y contraseña para poder ingresar posteriormente desde Grafana a nuestra base de datos, lo creamos usando las instrucciones

- `CREATE USER 'raymundo_soto'@'localhost' IDENTIFIED BY 'contrasena';`
- `GRANT ALL PRIVILEGES ON *.* TO 'raymundo_soto'@'localhost';`

Esto nos permitirá el acceso a nuestra base de datos local.

7. Para guardar desde Node se usan los siguientes Nodos mqtt, función y MySQL



- El nodo mqtt se conecta al tópico `Habeetat/sensores/dht11mqtt` del broker local donde publica el ESP32
- El nodo función `Conexion mysql` se programa para que publique los datos en la base de datos creada anteriormente, el código del nodo es:

```
msg.topic = "INSERT INTO Datos_Sensores (id_sensor, ValorAnalogico, VoltajeAnalogico, Temperatura, Humedad) VALUES (" + msg.payload.id + ",
```

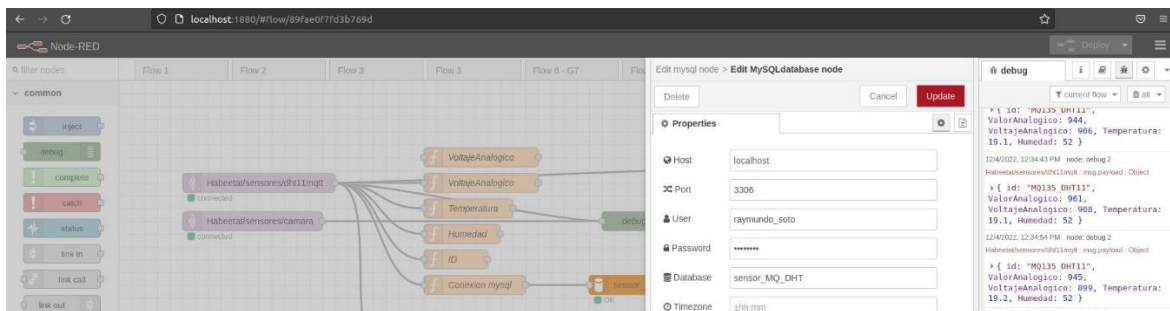


```
" + msg.payload.ValorAnalogico + "," + msg.payload.VoltajeAnalogico + "," + msg.payload.Temperatura + "," + msg.payload.Humedad + "),"
```

```
return msg;
```

Este nodo estará enviando los datos a la tabla Datos_Sensores para guardarlos en la base de datos.

* El nodo mysql *sensor_MQ_DHT* se configura para conectarse la base creada en el punto 3; la database debe ser *sensor_MQ_DHT*, el Host localhost, en el puerto 3306 y el usuario y contraseña del punto 6.



Una vez creada la base de datos podemos usarla para visualizar datos en Grafana o tomarlos para hacer predicciones de comportamiento.

Resultados

La base de datos con la tabla creada se muestra a continuación

```
raymundo@raymundo-VirtualBox: ~  
(base) raymundo@raymundo-VirtualBox:~$ sudo mysql;  
[sudo] password for raymundo:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11  
Server version: 8.0.31-0ubuntu0.20.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| RFID |  
| codigoIoT |  
| datosclima |  
| detectorsintomas |  
| information_schema |  
| mysql |  
| performance_schema |  
| sensor_MQ_DHT |  
| sensormq135 |  
| sys |  
| tiktok |  
+-----+  
11 rows in set (0.02 sec)  
  
mysql> use sensor_MQ_DHT  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_sensor_MQ_DHT |  
+-----+  
| Datos_Sensores |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

Los datos guardados desde el flow de nodered se muestra en esta imagen

```
16402 | 2022-12-04 12:32:08 | MQ135_DHT11 | 976.000 | 893.000 | 19.000 | 53.000 |  
16403 | 2022-12-04 12:32:19 | MQ135_DHT11 | 971.000 | 940.000 | 19.000 | 52.000 |  
16404 | 2022-12-04 12:32:30 | MQ135_DHT11 | 983.000 | 951.000 | 19.100 | 52.000 |  
16405 | 2022-12-04 12:32:41 | MQ135_DHT11 | 962.000 | 910.000 | 19.100 | 52.000 |  
16406 | 2022-12-04 12:32:52 | MQ135_DHT11 | 959.000 | 899.000 | 19.100 | 51.000 |  
16407 | 2022-12-04 12:33:03 | MQ135_DHT11 | 965.000 | 908.000 | 19.100 | 51.000 |  
16408 | 2022-12-04 12:33:14 | MQ135_DHT11 | 962.000 | 906.000 | 19.100 | 51.000 |  
16409 | 2022-12-04 12:33:25 | MQ135_DHT11 | 931.000 | 910.000 | 19.100 | 52.000 |  
16410 | 2022-12-04 12:33:36 | MQ135_DHT11 | 947.000 | 909.000 | 19.100 | 51.000 |  
16411 | 2022-12-04 12:33:47 | MQ135_DHT11 | 959.000 | 910.000 | 19.100 | 51.000 |  
16412 | 2022-12-04 12:33:58 | MQ135_DHT11 | 956.000 | 901.000 | 19.100 | 52.000 |  
16413 | 2022-12-04 12:34:10 | MQ135_DHT11 | 947.000 | 911.000 | 19.100 | 52.000 |  
16414 | 2022-12-04 12:34:21 | MQ135_DHT11 | 957.000 | 893.000 | 19.100 | 52.000 |  
16415 | 2022-12-04 12:34:32 | MQ135_DHT11 | 944.000 | 906.000 | 19.100 | 52.000 |  
16416 | 2022-12-04 12:34:43 | MQ135_DHT11 | 961.000 | 908.000 | 19.100 | 52.000 |  
16417 | 2022-12-04 12:34:54 | MQ135_DHT11 | 945.000 | 899.000 | 19.200 | 52.000 |  
16418 | 2022-12-04 12:35:05 | MQ135_DHT11 | 938.000 | 899.000 | 19.200 | 52.000 |  
+-----+  
16418 rows in set (0.17 sec)  
  
mysql> SELECT * FROM Datos_Sensores;
```

Ya tenemos lista la base de datos.

Notas del uso de los programa y del flow de NodeRed

A continuación se muestra un esquema general del uso de los programas para monitorear las variables del interior de la colmena, en las siguientes lecciones se profundiza en su explicación y desarrollo

- Colocar el SSID y el password de la red Wifi a la que se conectará el ESP32 WROOM al igual que la ESP32CAM

```
const char* ssid = "NOMBRE DE LA RED";
```

```
const char* password = "Contraseña";
```

- - Definir los datos de la ip del broker local, deben tener el siguiente formato

```
const char* mqtt_server = "192.168.xxx.xx";
```

```
IPAddress server(192,168,xxx,xx);
```

- El json que envía los datos con los sensores al broker mqtt local tiene el siguiente formato

```
String json =  
"{\"id\":\"MQ135_DHT11\", \"ValorAnalogico\": \""+String(analogValue)+\", \"VoltajeAnalogico\": \""+String(  
analogVolts)+\", \"Temperatura\": \""+String(t)+\", \"Humedad\": \""+String(h)+\"}\"";
```

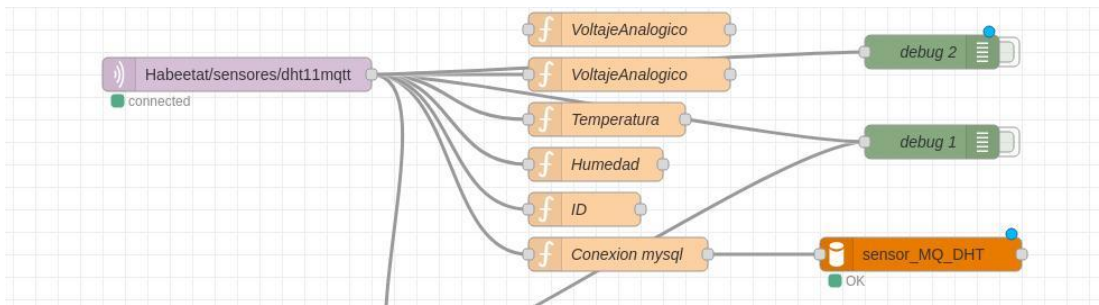
Es importante que sea en este formato porque posteriormente en el flow de NodeRed se leerá el json

El tópico de MQTT donde se publican los datos es "Habeetat/sensores/dht11mqtt", en el programa de arduino corresponde a la instrucción

```
client.publish("Habeetat/sensores/dht11mqtt", char_array);
```

Para leer los datos desde el broker en Nodered y guardarlos en la base de datos local se configura un nodo MQTT que se suscriba al tópico `Habeetat/sensores/dht11mqtt` y se programa la función que nos permita guardar en la base de datos local (el nombre de la base es *sensor_MQ_DHT*, la tabla donde se insertan los datos es *Datos_Sensores*), el código para la función *conexión mysql* es

```
msg.topic = "INSERT INTO Datos_Sensores (id_sensor, ValorAnalogico, VoltajeAnalogico,  
Temperatura, Humedad) VALUES ('" + msg.payload.id + "', " + msg.payload.VaporAnalogico + ", " +  
msg.payload.VoltajeAnalogico + ", " + msg.payload.Temperatura + ", " + msg.payload.Humedad  
+");return msg;`
```



Los datos son guardados en la base de datos que posteriormente será la fuente de datos para el panel de grafana.

El paquete de Nodered que nos permite enviar y recibir con un bot de telegram es [node-red-contrib-telegrambot](#), para poder ser usado debe crearse un bot desde la aplicación de Telegram, usar la API key, conseguir el ID del chat donde se publicara, configurar los nodos *receiver* con ayuda de un nodo *__funcion__* para que acepten una palabra clave y después el nodo *sender* envíe la información solicitada.

En este proyecto se puede hacer consulta usando las siguientes opciones:

Opciones de envío chat de Telegram del proyecto Habeetat

Palabra a través del chat	Acción del bot
consulta	Envía los datos más recientes de los sensores
imagen	Envía una imagen reciente de la cámara en la colmena
predicción	Envía 3 imágenes con resultados de la predicción del peso de la colmena en los siguientes días

A continuación se muestra un ejemplo de la función para enviar datos de los sensores a través del *sender*

```
if (msg.payload.content === "consulta" || msg.payload.content === "Consulta")
{
    msg.payload = {};

    msg.payload.chatId = -XXXXXXXXXX; // Aquí se coloca el ID del chat a usar para enviar mensajes

    msg.payload.type = 'message';

    var espacio = " \n";

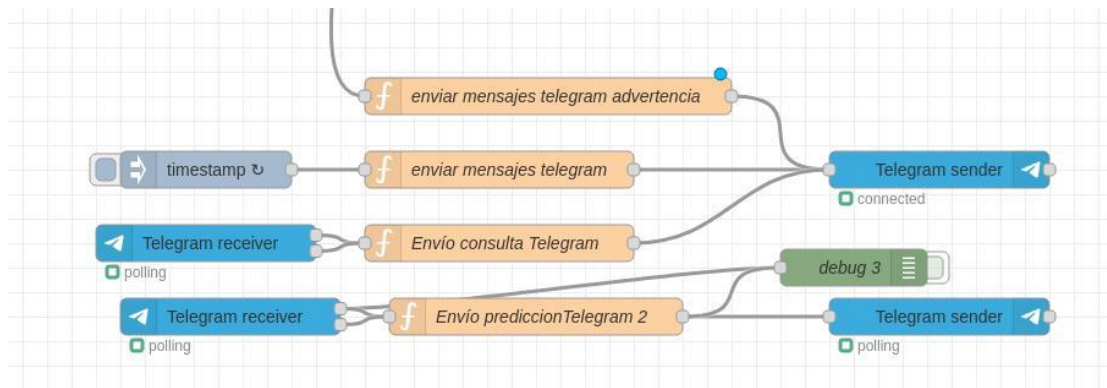
    msg.payload.content = "\u{1F41D}" + " Datos de la colmena " + "\u{1F41D}" + espacio + "Nombre del sensor: " + global.get('id') + espacio + "\u{1F321}" + "Temperatura: " + global.get("Temperatura") + " °C" + espacio + "\u{2614}" + "Humedad: " + global.get("Humedad") + " %" + espacio + "\u{1F32B}" + "Concentración de CO2: " + espacio + global.get("VoltajeAnalogico") + " PPM"+
```

espacio + "Envía <consulta> para acceder a los datos más actuales de la colmena, envía <prediccion> para el pronostico del peso de la colmena, envía <imagen> para recibir una fotografía actual de la colmena";

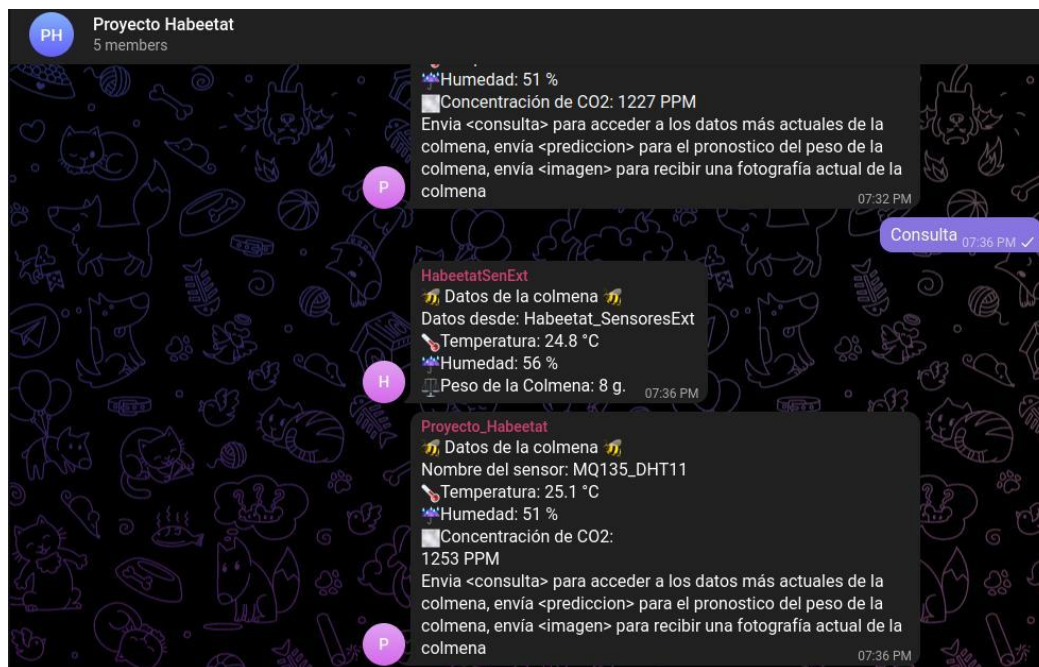
```
return msg;
```

```
}
```

Los nodos configurados para el envío se muestra en la siguiente imagen.



Ejemplo de la consulta de datos

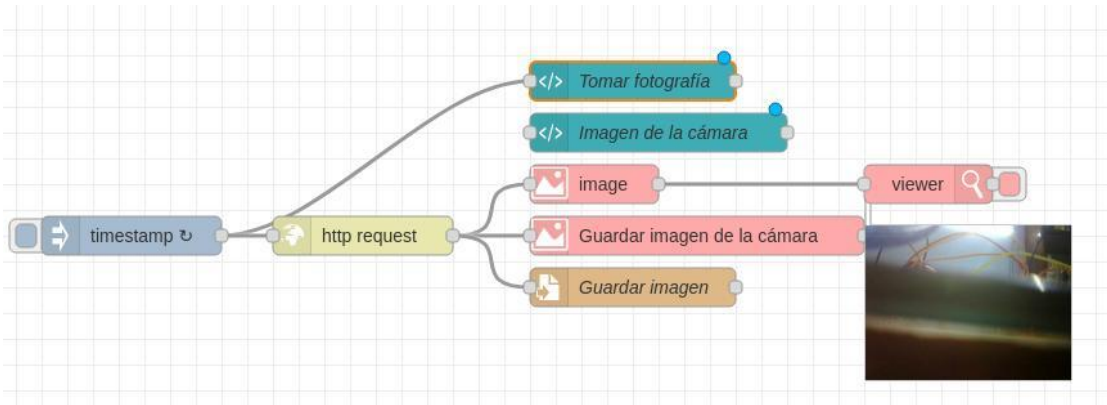


El flow de Nodered se diseña para adquirir vídeo en tiempo real y al mismo tiempo tomar una imagen que se almacena de forma local y que posteriormente se puede enviar a través del chat de telegram haciendo la petición enviando la palabra *imagen*. El flow contiene un dashboard donde se ve el vídeo en tiempo real desde el ESP32CAM y la imagen capturada.

Nodos que envía la imagen con el bot de telegram



Se guarda la imagen cada minuto para su envío cuando se consulta por telegram, los nodos encargados de la captura, guardado de la imagen y publicación del vídeo en el dashboard de node red son los siguientes:



Se configuran dos nodos *template* de nodered para poder ver el vídeo y la fotografía en el dashboard, el código de los nodos es el siguiente:

Nodo _tomar fotografía

```
<div style="margin-bottom: 10px;">
```

```
  
```

```
</div>
```

La línea `http://192.168.100.xxx:81/capture` debe incluir la ip completa que nos envía el monitor serial de arduino cuando programamos el ESP32 CAM.

Nodo _imagen de la cámara

```
<div style="margin-bottom: 10px;">
```

```
  
```

```
</div>
```

La línea `http://192.168.100.xxx` debe incluir la ip completa que nos envía el monitor serial de arduino cuando programamos el ESP32 CAM.

El dashboard con el vídeo y la fotografía se muestran a continuación



Cuando se envía la palabra *imagen* al chat de telegram La imagen se recibe en el chat de elegido de la siguiente forma:

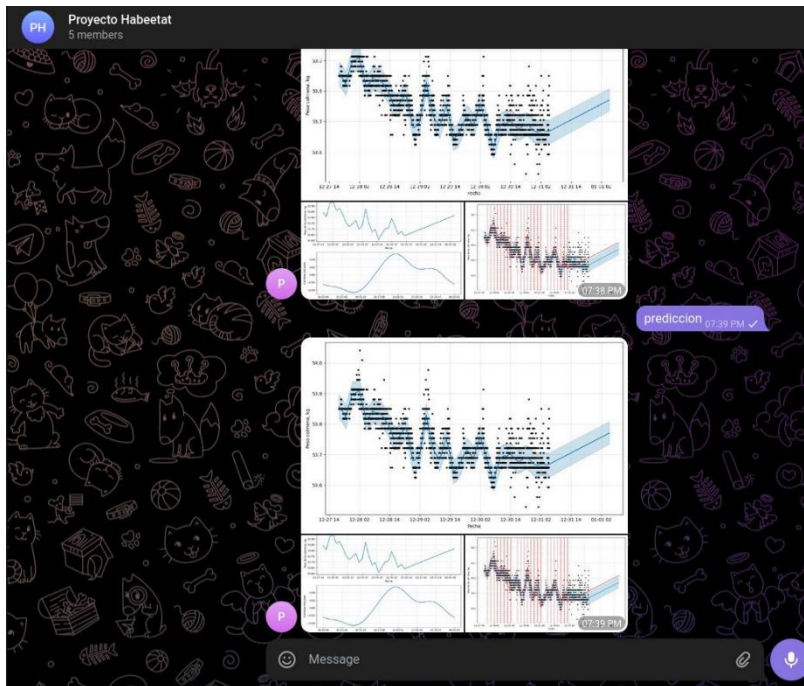


La predicción del comportamiento del peso de la colmena se realiza usando una biblioteca llamada *Prophet* desarrollada por el equipo de Facebook y de acuerdo a la (página oficial de [Prophet](#)). *Prophet* es una biblioteca desarrollada para pronosticar datos de series temporales basado en un modelo aditivo en el que las tendencias no lineales se ajustan a la estacionalidad anual, semanal y diaria, además de los efectos de los datos faltantes. Funciona mejor con series temporales que tienen fuertes efectos estacionales y varias temporadas de datos históricos. Prophet es resistente a los datos faltantes y los cambios en la tendencia, y por lo general maneja bien los valores atípicos. Este tipo de comportamientos son comunes en sistemas donde se miden variables climáticas, además es relativamente fácil de implementar en Python. Dado que no se cuenta todavía con datos reales para hacer la implementación de la predicción, se toma un conjunto de datos reales que han sido medidos *_in situ_* en colmenas ([dataset usado para la predicción](#)), de tal forma que los datos esperados en nuestras mediciones tengan características similares a este data set.

Para hacer la petición de la predicción de los datos de peso de la colmena se envía la palabra *prediccion* en el chat de telegram, esto retornará 3 gráficas que mostrarán los datos históricos del peso de la colmena (se define en el programa el tiempo de predicción), junto con la línea de tendencia y la predicción del peso en los siguientes días. De forma adicional hay 2 gráficas que nos muestran las variaciones estadísticas por hora del peso a lo largo día y los puntos (días) donde existe una variación significativa del peso. Estas gráficas juntos con los datos recopilados

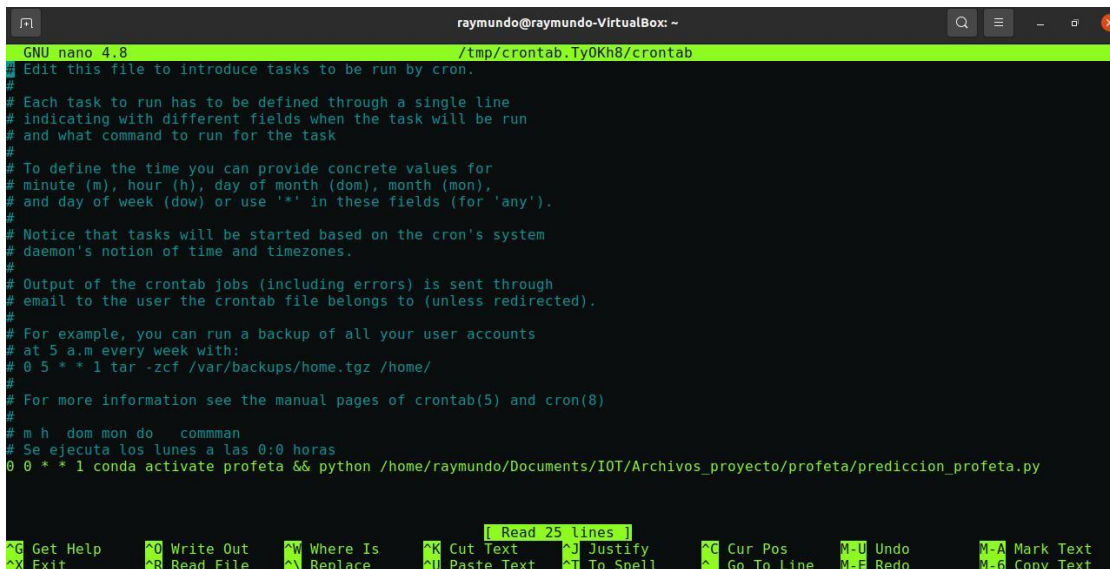
tienen el objetivo de ser el soporte de la toma de decisiones con el fin de mantener la salud de la colmena.

Se muestra a continuación un ejemplo de la petición



El programa de la predicción se realizó en Python y se ejecuta automáticamente usando un administrador de procesos en segundo plano para Linux llamado **Cron** que ejecuta procesos, programas o scripts a intervalos regulares de tiempo, es decir, con crontab podemos elegir en qué momento ejecutar algo. Las tareas de Cron se registran y administran en un archivo especial llamado *crontab*. Cada perfil de usuario del sistema Linux puede tener su propio *crontab* para programar tareas, que se almacenan en `_/var/spool/cron/crontabs`. En este proyecto se programaron las tareas para que se ejecute la predicción de forma automática cada 8 días los lunes a las 00:00 horas y que los datos estén disponibles para toda la semana en el momento que el usuario invoque la predicción. Una vez editado el archivo *crontab* se guarda, se cierra y estará trabajando en segundo plano hasta que se desactive.

El archivo *crontab* editado para este proyecto se ve así



```
GNU nano 4.8 /tmp/crontab.Ty0Kh8/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
# Se ejecuta los lunes a las 0:0 horas
0 0 * * * 1 conda activate profeta && python /home/raymundo/Documents/IOT/Archivos_proyecto/profeta/prediccion_profeta.py
```

La última línea es la que especifica la hora y día en que se ejecutará el comando que invoca el script de python para la predicción. En nuestro caso, la última línea indica que se active un ambiente virtual llamado *profeta* y ejecute el archivo *.py* con el programa para la predicción.

Con lo anterior ya es posible obtener datos del interior de la colmena, imágenes y vídeo en tiempo real y además se realiza la analítica de los datos mediante un modelo matemático.

Creación del dashboard en grafana para mostrar los datos del interior de la colmena

En este apartado se muestra el procedimiento para la creación del dashboard de Grafana en los que se muestran los datos recopilados con los sensores MQ135 y DHT11 para monitorear la temperatura, humedad y concentración de CO₂ al interior de la colmena. El dashboard de grafana usa como fuente de datos la base de datos creada anteriormente.

El panel final consta de 3 gráficas de series de tiempo mostrando la temperatura, humedad y concentración de CO₂ históricos y 3 indicadores de aguja que muestran los datos actuales de la variables medidas. Estas gráficas proporcionan un resumen gráfico muy vistoso de lo que ocurre al interior de la colmena.

Software necesario

- Máquina virtual con Ubuntu 22.04
- Grafana

Instrucciones de instalación

1. Abrir una terminal de Ubuntu y teclear las siguientes instrucciones

- `sudo apt-get install -y apt-transport-https`
- `sudo apt-get install -y software-properties-common wget`
- `sudo wget -q -O /usr/share/keyrings/grafana.key https://packages.grafana.com/gpg.key``
- `echo "deb [signed-by=/usr/share/keyrings/grafana.key] https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list`
- `sudo apt-get update`
- `sudo apt-get install grafana-enterprise`

2. Una vez terminada la instalación hay que arrancar Grafana y poder abrirlo desde el navegador, este comando se ejecuta cada vez que queramos abrir Grafana

- `sudo /bin/systemctl enable grafana-server`

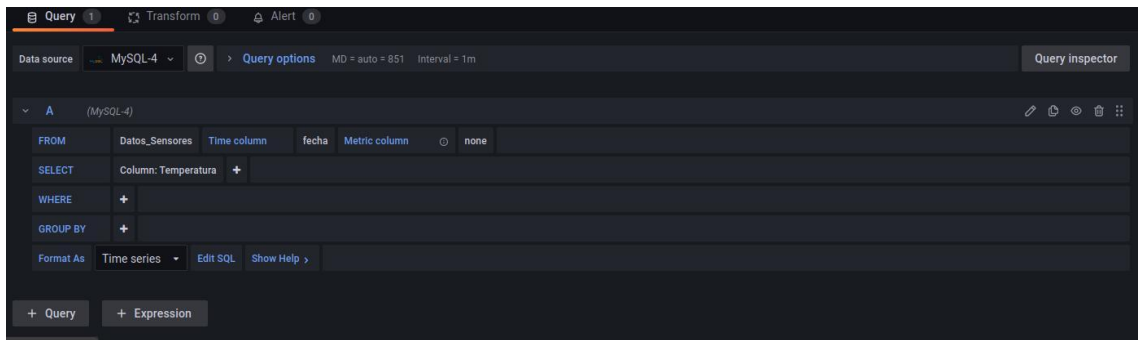
3. Para abrir Grafana abrimos un navegador y nos vamos a la dirección `http://localhost:3000/` donde nos logeamos con el usuario y contraseña por default, se recomienda actualizar esos datos.

- User: admin
- Password: admin

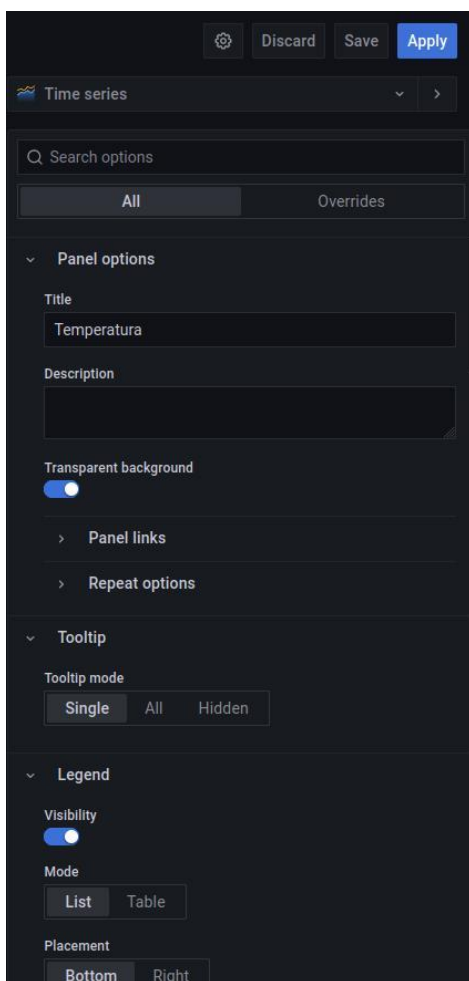
Con lo anterior podemos crear ya un panel de Grafana

Instrucciones de uso

1. Antes de comenzar a crear un panel de Grafana debemos añadir una fuente de datos (Add your first data source o add data source), para este panel elegimos MySQL en la lista de opciones de la que disponemos.
2. Posteriormente le colocamos un nombre a nuestra fuente.
3. El host es ``localhost:3306``
4. El nombre de la base de datos elegida debe ser el nombre con el que creamos la base: en este caso `sensor_MQ_DHT`
5. El usuario y contraseña es el que creamos en nuestra base de datos MySQL
6. Todas las demás opciones las dejamos por default
7. Oprimimos `Save & test`
8. Si todo está bien aparecerá un mensaje indicando que se agregó la base de datos
9. Agregamos un nuevo panel oprimiendo el botón superior derecho que contiene un símbolo de + y una gráfica de barras (``Add panel``)
10. Configuramos el panel
 - Eligiendo un tipo de gráfica (Time series, Bar chart, stat, gauge, table, heatmap, etc)
 - Elegimos la fuente de datos (data source)
 - Seleccionamos la columnas de nuestra base de datos a graficar
 - Seleccionar tabla
 - Seleccionar columna de tiempo
 - `SELECT column value` (datos de temperatura, VoltajeAnalógico, Humedad)
 - En estas opciones se seleccionan todos los datos a mostrar en nuestra gráfica



- Posteriormente se le da formato al gráfico para seleccionar colores, tipo de líneas, etiquetas a mostrar, etc
- Oprimimos Apply



- La gráfica queda guardada y lista para ser mostrada

11. Se repiten los pasos anteriores para agregar nuevas gráficas a nuestro dashboard
 12. Se guardan los cambios del panel

13. Si se desea se puede exportar el panel en formato HTML para embeberlo en páginas web

El panel final de Grafana para los sensores internos tiene el siguiente aspecto



Preguntas frecuentes y compatibilidad

Preguntas frecuentes

P: ¿Qué hacer si al conectar el esp32 cam o el esp32 WROOM no existe comunicación serial?

- Revisar que los cables para conexión USB sirven para transmitir datos y no son sólo de carga
- Checar si la computadora tiene instalados los drivers para establecer comunicación entre el IDE de arduino y el microcontrolador
- Revisar haber elegido el puerto adecuado en el IDE de arduino

P: ¿Por qué veo símbolos el monitor serial de arduino que no corresponden a los datos de los sensores?

- Revisa que la velocidad de los baudios esté en el valor correcto, cambiarlos de mayor a menor hasta observar el valor adecuado

P: ¿Cómo se calibra el sensor MQ135?

- La calibración del sensor deben de realizarse en condiciones controladas y muy específicas, una guía para una buena calibración está en este [link](#)

Compatibilidad

En este proyecto se usaron los microcontroladores ESP32-CAM y ESP32 WROOM, se puede usar Arduino Uno haciendo algunos cambios en las conexiones

Evidencias de funcionamiento

Videos de YouTube

- Funcionamiento de los sensores con MQTT, Nodered y la base de datos MySQL

<https://youtu.be/seBQrWVgVqA>

<https://youtu.be/s4fPHFBnQRI>

<https://youtu.be/pgujhDJBkKk>