

Implementación de sensores externos de la colmena inteligente

Para asegurar la salud de la colmena, hemos considerado monitorear constantemente la temperatura y la humedad externa y cercana a la colmena; así como, el incremento o decremento del peso de la colmena desde su colonización por el enjambre de las abejas.

Así que en este apartado sobre el desarrollo del proyecto Capstone, Habeetat Colmena saludable, se describe cómo se implementaron los siguientes puntos:

1. La adquisición de la temperatura, humedad y peso; circundantes a la colmena.
2. Activación de un actuador que funcione con 127Vca y hasta 1500W de potencia.
3. Implementación de una base de datos en MySQL para llevar un registro de las variables detectadas.
4. Interfaz gráfica y de control en NodeRed.
5. Graficado de variables registradas en Grafana.
6. Solicitud de estado de las condiciones de las variables en tiempo real por mensaje de un Bot en Telegram.

Material necesario

A continuación se muestra el material necesario para el monitoreo de la temperatura, humedad y peso de la colmena.

- 1 [ESP32-CAM, microcontrolador.](#)
- 1 [FTDI. Tarjeta controladora USB.](#)
- 1 [DHT11. Sensor de temperatura y humedad.](#)
- 1 [HX711. ADC-24 bits para escalas de peso.](#)
- 1 [Celda de carga, también conocida como Galga Extensiométrica.](#)
- 1 Resistor de 100ohms. (Café,Negro,Café,Dorado)
- 1 Resistor de 220ohms. (Rojo,Rojo,Café,Dorado)
- 1 Resistor de 330ohms. (Naranja,Naranja,Café,Dorado)
- 1 Resistor de 360ohms. (Naranja,Azul,Café,Dorado)
- 1 Resistor de 10Kohms. (Café,Negro,Naranja,Dorado)
- 1 BT137 (TRIAC)
- 1 [MOC3043, optoacoplador.](#)
- 1 Capacitor de 10nF (103)
- 1 cable USB a USB mini.
- Jumpers MM.

Software necesario

En la experimentación de esta práctica se debe contar con el siguiente software libre:

- Ubuntu 20.04.
- Arduino IDE.
- Mosquitto MQTT Broker, Listener en puerto 1883 para 0.0.0.0 y conexiones autenticadas activadas.
- NodeJS. NPM, NodeRed y Node Dashboard.
- MySQL.
- Grafana.

Material de referencia

Previamente a la realización de esta práctica, ha sido necesario el estudio de distintos temas, que se encuentran en la plataforma edu.codigoiot.com, en donde se explican conceptos y configuraciones necesarias, tales como:

- Instalación de virtual Box y Ubuntu 20.04.
- Configuración de Arduino IDE para ESP32CAM.
- Instalación de NodeRed.
- Introducción a NodeRed.
- Instalación de Mosquitto MQTT.

También se recomienda el estudio de las siguientes páginas, donde se explican algunas implementaciones de una báscula electrónica usando la tarjeta de desarrollo Arduino UNO.

- [Bogde](https://github.com/bogde/HX711) (<https://github.com/bogde/HX711>)
- [Balanza digital](https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-balanza-digital.html) (https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-balanza-digital.html)
- [Balanza Electrónica con HX711 y Arduino](https://controlautomaticoeducacion.com/arduino/balanza-electronica-hx711-arduino/) (<https://controlautomaticoeducacion.com/arduino/balanza-electronica-hx711-arduino/>)
- [ESP32 Troubleshooting Guide](https://randomnerdtutorials.com/esp32-troubleshooting-guide/) (<https://randomnerdtutorials.com/esp32-troubleshooting-guide/>)

Servicios

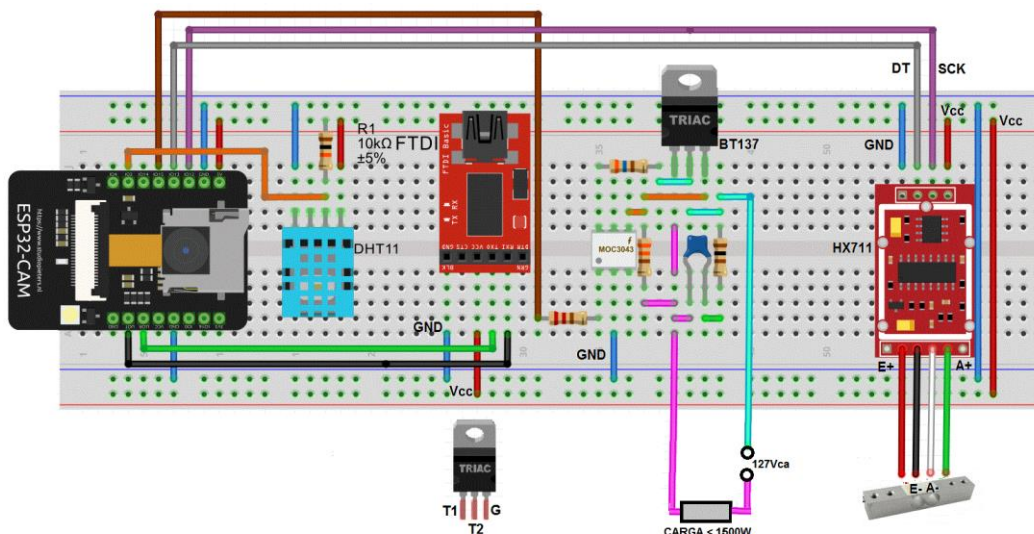
Adicional a lo ya indicado en líneas superiores, también es necesario contar con los siguientes servicios:

- [HiveMQ](#). Es un bróker público y que no demanda de contar con una cuenta.
- [Telegram Messenger](#). Servicio de mensajería.

La adquisición de la temperatura, humedad y peso

Nota: Se recomienda revisar la información previamente citada, antes de comenzar con el armado del circuito electrónico, para reducir la probabilidad de realizar malas conexiones entre los dispositivos, fallas en la polarización y en consecuencia el daño permanente de los dispositivos o daño parcial o total de su equipo de cómputo.

1. Se debe de armar el circuito electrónico mostrado en la figura teniendo cuidado de conectar a las terminales indicadas del ESP32-CAM, y cuidar la polaridad de los dispositivos.



Circuito Electrónico de Control y Adquisición de Peso, Temperatura y Humedad, y Control de Carga.

En las siguientes tablas, se puede observar la correspondencia de las terminales que se deben conectar entre el ESP32-CAM y los distintos sensores y el actuador.

Terminales de Conexión de ESP32-CAM a DHT11

ESP32-CAM DHT11

GPIO 2	DATA
5V	Vcc
GND	GND

Terminales de Conexión de ESP32-CAM a HX711

ESP32-CAM HX711

Vcc = 5V	Vcc
GND	GND
GPIO 12	SCK
GPIO 13	DT

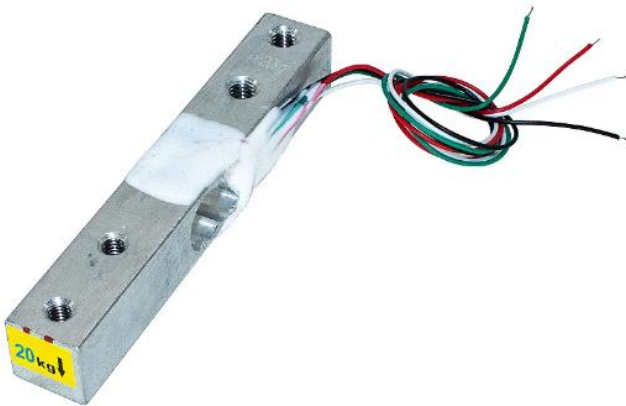
Terminales de Conexión de ESP32-CAM a MOC3043

ESP32-CAM MOC3043

GPIO 15 PIN 1

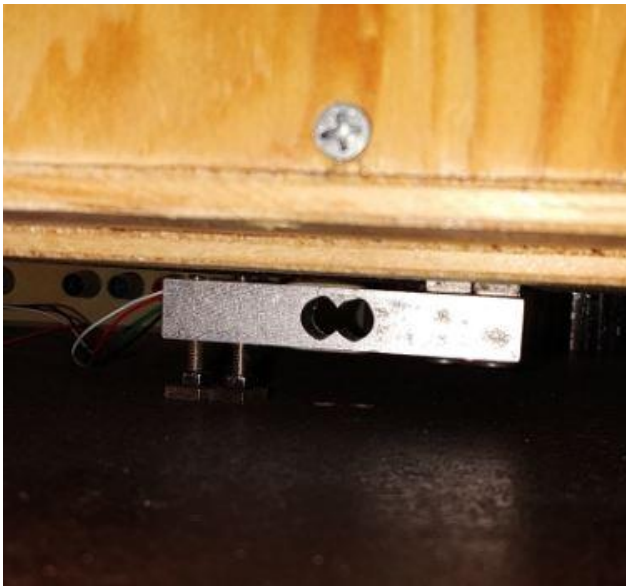
GND PIN 2

2. Se tiene que fijar la celda de carga de 20Kg., en una base sólida que evite el desequilibrio de la báscula, cuidando la posición de la celda de carga, la cual suele tener una flecha que debe señalar hacia la base inferior, para su correcta colocación, véase figura.



Forma Física de la Celda de Carga de 20kg., utilizada.

3. Fijar la base de la colmena en la parte superior de la celda, teniendo un registro previo del peso total de la colmena para poder destarar y ajustar el valor correspondiente al peso registrado. Ver figuras



Vista frontal del acoplado de la celda de carga entre la base y la colmena.



Vista diagonal del acoplado de la celda de carga entre la base y la colmena.

4. Cargar el [programa](https://github.com/OmarAbundis/Habeetat-Colmena-saludable/blob/main/Control_Sensores_Temp_Hum_Peso/Sensores_Temp_Humedad_Peso_MQTT/Sensores_Temp_Humedad_Peso_MQTT.ino) (https://github.com/OmarAbundis/Habeetat-Colmena-saludable/blob/main/Control_Sensores_Temp_Hum_Peso/Sensores_Temp_Humedad_Peso_MQTT/Sensores_Temp_Humedad_Peso_MQTT.ino) de control y adquisición de peso, temperatura y humedad.



```
1 *  
2 * Proyecto: HAbEeTat, Colmena saludable  
3 *  
4 * Programa de control con las siguientes características:  
5 *  
6 * -> Detecta el peso de la colmena, usando una galaga extensiométrica de 20Kg. y un acondicionador de señal HX711  
7 * -> Detecta la temperatura y la humedad de la colmena, usando el sensor DHT11  
8 * -> Las señales de los sensores ya procesadas por el ESP32CAM, son transformados a tipo JSON y comunicados a través del protocolo MQTT a un Flow que sirve de punto de monitoreo.  
9 *  
10 * Integrantes del equipo:  
11 *  
12 * Abundis Noyola Omar  
13 * García León Lizbeth  
14 * Soto Soto Raymundo  
15 *  
16 * Profesor e instructor del curso de Código IoT  
17 * Hugo Vargas  
18 *  
19 * Fecha: 21 de noviembre del 2022  
20 *  
21 *
```

Código de control cargado en el IDE de Arduino.

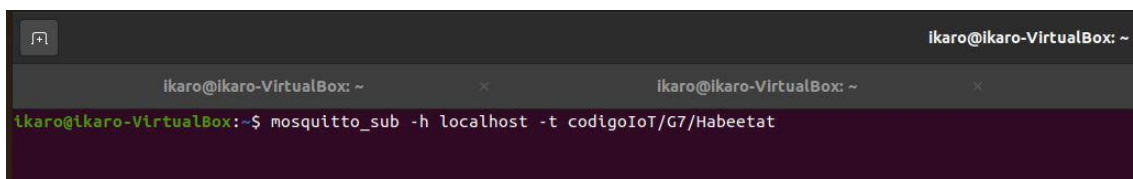
5. Hasta el momento, en esta parte del proyecto el circuito electrónico es capaz de detectar el peso de la colmena, la temperatura y humedad; al exterior y cercanas a la colmena. Para comprobarlo, se puede abrir el *Serial Monitor*, que es una herramienta integrada al IDE de Arduino, que nos permite observar los datos que mandemos a imprimir, en nuestro caso las variables de interés. La impresión de las variables están en formato JSON.



```
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.19}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.13}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.19}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.29}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":2079.49}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":1992.81}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.16}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada  
{\"id\":\"Habeetat_SensoresExt\",\"temp\":22.58,\"hum\":55.06,\"peso\":12.21}  
Llegó un mensaje en el tema: codigoloT/G7/Habeetat_infalse  
Mensaje concatenado en una sola variable: false  
Carga Desactivada
```

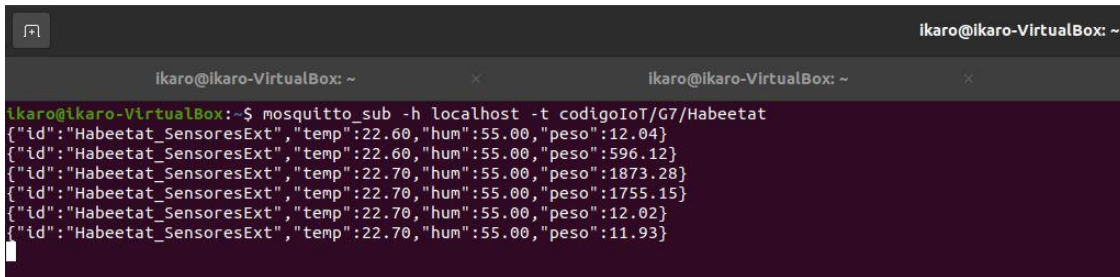
Observación de los datos impresos en el monitor serial del IDE de Arduino.

6. Ya comprobada la adquisición y debido procesamiento de las variables, podemos comprobar la comunicación al Bróker Mosquitto, realizando una subscripción al tópico creado, desconectando el cable USB a USB mini, de la computadora y observando los valores que se imprimen en la consola.



```
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox: ~$ mosquitto_sub -h localhost -t codigoloT/G7/Habeetat
```

Subscripción al tópico: codigoloT/G7/Habeetat. Desde la consola de Ubuntu 20.04.



```
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox:~$ mosquitto_sub -h localhost -t codigoIoT/G7/Habeetat  
{"id":"Habeetat_SensoresExt","temp":22.60,"hum":55.00,"peso":12.04}  
{"id":"Habeetat_SensoresExt","temp":22.60,"hum":55.00,"peso":596.12}  
{"id":"Habeetat_SensoresExt","temp":22.70,"hum":55.00,"peso":1873.28}  
{"id":"Habeetat_SensoresExt","temp":22.70,"hum":55.00,"peso":1755.15}  
{"id":"Habeetat_SensoresExt","temp":22.70,"hum":55.00,"peso":12.02}  
{"id":"Habeetat_SensoresExt","temp":22.70,"hum":55.00,"peso":11.93}
```

Observación de la impresión de los valores en la consola de Ubuntu 20.04.

Implementación de una base de datos en MySQL

Hasta este momento, se tiene resuelto la detección de las variables de interés, peso de la colmena, temperatura y humedad ambientales cercanas a la colmena y la activación de un actuador que se pueda conectar al suministro eléctrico (127 Vca a 60Hz, estándar en México), pero todavía le faltan más implementaciones para que sea considerado como parte del ecosistema del Internet de las cosas. Por ejemplo, ya desde hace mucho tiempo si estás suscrito a alguna plataforma de *streaming*, habrás notado que entre más haces uso de ella empieza después de un determinado tiempo, a sugerirte temas de tú interés, aunque tú no hayas empezado alguna búsqueda. ¿Cómo se realiza?

Estos sistemas de *streaming*, además de que tienen modelos de predicción o algoritmos de inteligencia artificial (que también se incluye en nuestro proyecto un ejemplo de implementación), para que funcionen necesitan de datos y entre más tiempo esté recopilando datos, se puede hacer una mejor predicción, en el caso de los *streaming*, de tus gustos.

En resumen, para implementar un modelo predictivo, entonces necesitamos recopilar datos, provenientes del peso, temperatura, humedad y CO2; de la colmena y almacenarlos para su posterior procesamiento con algún algoritmo de predicción.

A continuación, se muestra, cómo se puede implementar una rápida y sencilla base de datos utilizando **MySQL**, que es un software libre y que fácilmente se puede implementar en Ubuntu 20.04, nuestro sistema operativo en utilización.

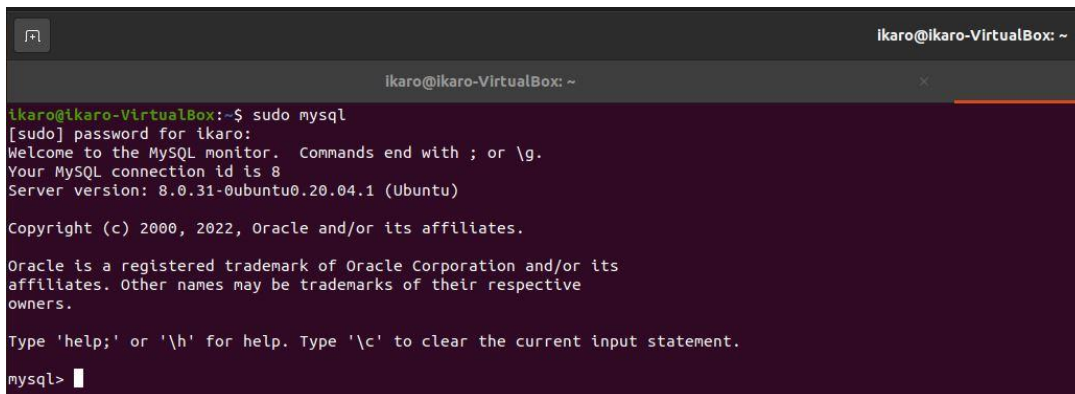
Nota: No olvide estudiar las referencias indicadas, ya que ahí se describe como instalar MySQL en Ubuntu 20.04.

Estos son los pasos por seguir:

1. Ya teniendo instalado MySQL, en la consola de Ubuntu 20.04, escribe el comando

```
sudo mysql
```

Te solicitará tu contraseña, la escribes sin preocuparte de que no se muestre, es por seguridad, pulsa ENTER e ingresarás a MySQL.



```
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox:~$ sudo mysql  
[sudo] password for ikaro:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 8  
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>
```

Imagen de la consola cuando ya accedió a MySQL.

2. Creas una base de datos

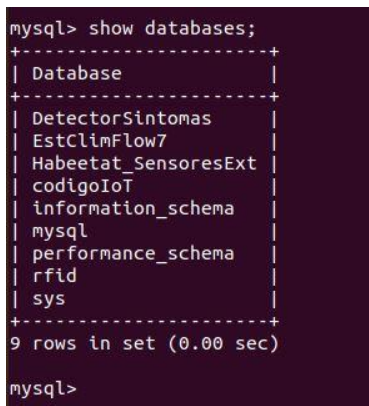
CREATED DATABASE “Nombre que quieres para tu base de datos”;

Ejemplo:

CREATED DATABASE Habeetat_SensoresExt

3. Corrobora su creación con el comando

SHOW databases;



```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| DetectorSintomas |  
| EstClimFlow7 |  
| Habeetat_SensoresExt |  
| codigoIoT |  
| information_schema |  
| mysql |  
| performance_schema |  
| rfid |  
| sys |  
+-----+  
9 rows in set (0.00 sec)  
  
mysql>
```

Nombres de las bases de datos existentes en MySQL.

4. Seleccionas tu base de datos creada con el comando

USE Habeetat_SensoresExt;


```
mysql> USE Habeetat_SensoresExt;
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> █
```

Indicación de que cambió la base de datos que se tenía por defecto.

5. Creas una tabla que contenga los campos deseados

```
create table Habeetat_SExt (id INT (6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, fecha
TIMESTAMP DEFAULT CURRENT_TIMESTAMP, Nombre CHAR (248) NOT NULL, Temperatura
FLOAT (4,2), Humedad INT (3), Peso INT (4));
```

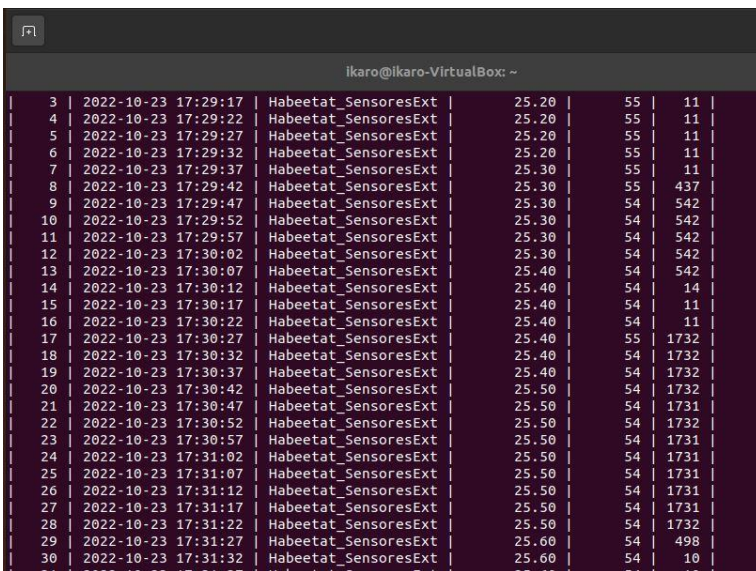
6. Para comprobar usas el comando

```
select * from Habeetat_SExt;
```

Y te mostrará tu tabla con los campos creados.

```
Database changed
mysql> select * from Habeetat_SExt; █
```

Escritura de comando en la consola de Ubuntu 20.04.



The screenshot shows a terminal window titled 'ikaro@ikaro-VirtualBox: ~'. It displays the output of a SQL query, which is a table with 7 columns: an index, a timestamp, the table name 'Habeetat_SensoresExt', and four numerical values. The data is as follows:

Index	Timestamp	Table Name	Temp	Humid	Peso	Id
3	2022-10-23 17:29:17	Habeetat_SensoresExt	25.20	55	11	11
4	2022-10-23 17:29:22	Habeetat_SensoresExt	25.20	55	11	11
5	2022-10-23 17:29:27	Habeetat_SensoresExt	25.20	55	11	11
6	2022-10-23 17:29:32	Habeetat_SensoresExt	25.20	55	11	11
7	2022-10-23 17:29:37	Habeetat_SensoresExt	25.30	55	11	11
8	2022-10-23 17:29:42	Habeetat_SensoresExt	25.30	55	437	437
9	2022-10-23 17:29:47	Habeetat_SensoresExt	25.30	54	542	542
10	2022-10-23 17:29:52	Habeetat_SensoresExt	25.30	54	542	542
11	2022-10-23 17:29:57	Habeetat_SensoresExt	25.30	54	542	542
12	2022-10-23 17:30:02	Habeetat_SensoresExt	25.30	54	542	542
13	2022-10-23 17:30:07	Habeetat_SensoresExt	25.40	54	542	542
14	2022-10-23 17:30:12	Habeetat_SensoresExt	25.40	54	14	14
15	2022-10-23 17:30:17	Habeetat_SensoresExt	25.40	54	11	11
16	2022-10-23 17:30:22	Habeetat_SensoresExt	25.40	54	11	11
17	2022-10-23 17:30:27	Habeetat_SensoresExt	25.40	55	1732	1732
18	2022-10-23 17:30:32	Habeetat_SensoresExt	25.40	54	1732	1732
19	2022-10-23 17:30:37	Habeetat_SensoresExt	25.40	54	1732	1732
20	2022-10-23 17:30:42	Habeetat_SensoresExt	25.50	54	1732	1732
21	2022-10-23 17:30:47	Habeetat_SensoresExt	25.50	54	1731	1731
22	2022-10-23 17:30:52	Habeetat_SensoresExt	25.50	54	1732	1732
23	2022-10-23 17:30:57	Habeetat_SensoresExt	25.50	54	1731	1731
24	2022-10-23 17:31:02	Habeetat_SensoresExt	25.50	54	1731	1731
25	2022-10-23 17:31:07	Habeetat_SensoresExt	25.50	54	1731	1731
26	2022-10-23 17:31:12	Habeetat_SensoresExt	25.50	54	1731	1731
27	2022-10-23 17:31:17	Habeetat_SensoresExt	25.50	54	1731	1731
28	2022-10-23 17:31:22	Habeetat_SensoresExt	25.50	54	1732	1732
29	2022-10-23 17:31:27	Habeetat_SensoresExt	25.60	54	498	498
30	2022-10-23 17:31:32	Habeetat_SensoresExt	25.60	54	10	10

Ejemplo de la tabla con datos ya almacenados.

7. Ahora a darle privilegios.

Si por ejemplo

Usuario es: **PanchoLopez**

Password es: **1234**

El comando quedaría como

```
CREATE USER 'PanchoLopez'@'localhost' IDENTIFIED BY '1234';
```

inmediatamente después,

```
GRANT ALL PRIVILEGES ON *.* TO 'PanchoLopez'@'localhost';
```

Nota: Guarda bien el usuario y la contraseña para su posterior uso.

¡Listo! Ya está la base de datos para el almacenamiento de los parámetros correspondientes a la variables temperatura, humedad y peso de la colmena.

Interfaz gráfica y de control en Node Red

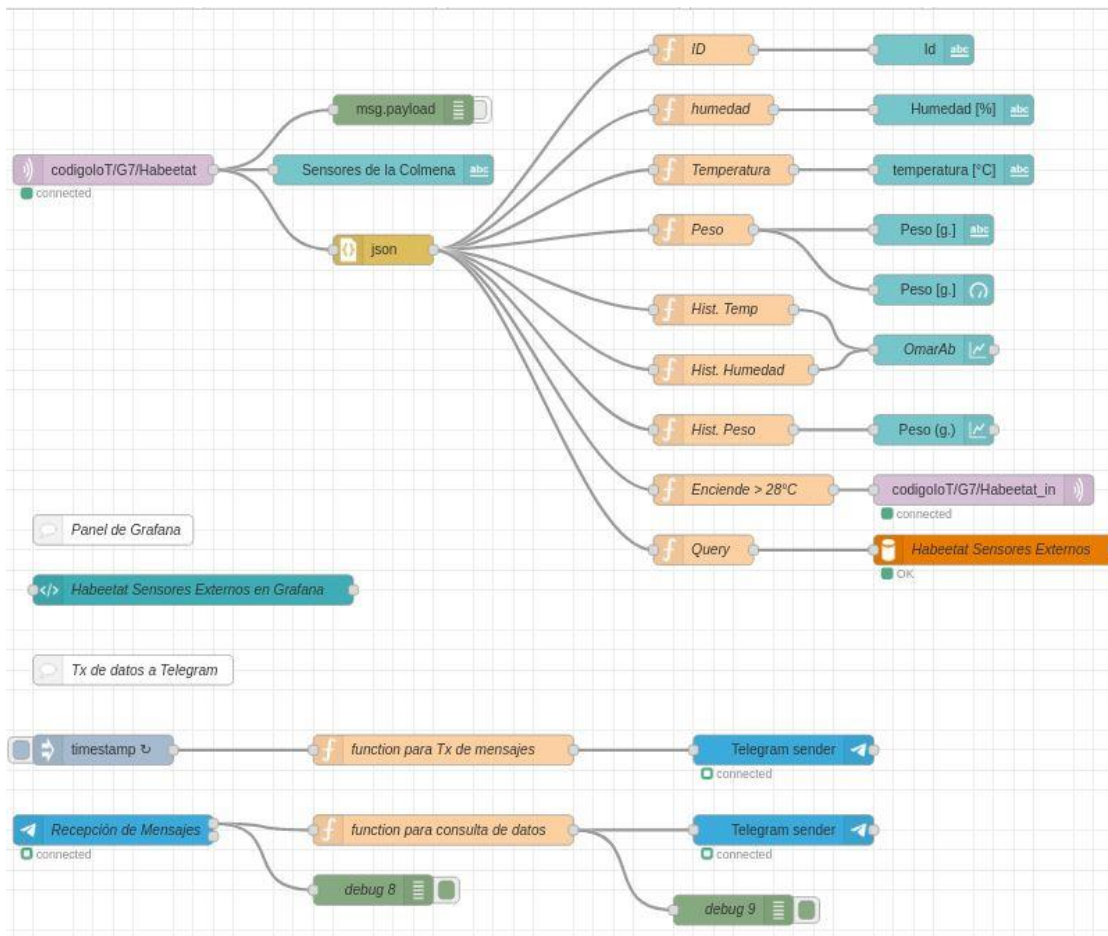
Hasta el momento, ya está implementado el circuito electrónico de control, ya se detectan las variables de interés temperatura, humedad y peso acumulado de la colmena, ya se cuenta con transmisión de los datos en formato JSON utilizando Bróker Mosquitto y se tiene implementada una base de datos en MySQL, para generar el almacenamiento del registro de los datos de la variable; pero aún le faltan implementaciones. por ejemplo, una interfaz gráfica en donde cualquier usuario pueda leer de una manera simple, precisa y exacta las variables de interés, le falta incluir algunas estructuras de control, para la activación de los actuadores y poder mandar alertas y mensajes sobre la salud de la colmena. Para estas implementaciones previamente señaladas, se va a utilizar Node-Red, que es una herramienta de desarrollo basada en flujo para programación visual, que permite conectar dispositivos de hardware, API y servicios en línea.

A continuación se indica cómo se realiza la implementación.

1. Ya teniendo instalado Node-Red, desde la consola de Ubuntu 20.04 se escribe el comando

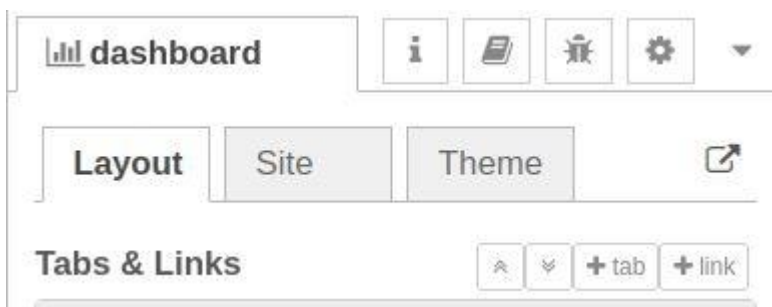
```
node-red
```

el cual pone en marcha la herramienta de desarrollo.



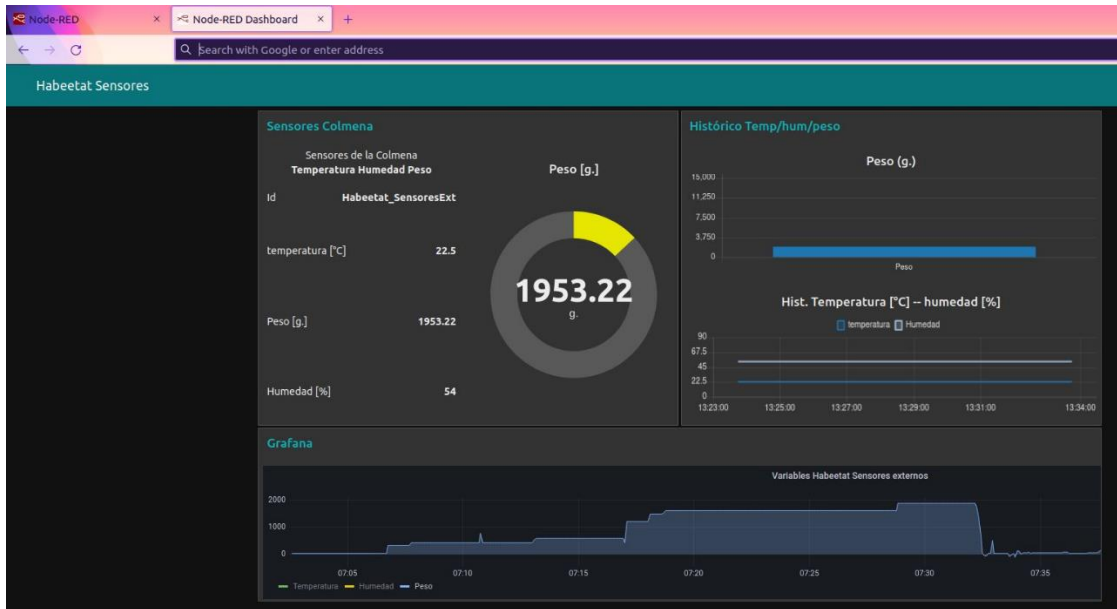
Flow de control de Habeetat de SensoresExt.

4. En la página de NodeRed se hace desplegado del *Dashboard*, dando clic en el cuadrito con flecha en diagonal apuntando hacia arriba, y se desplegará la interfaz gráfica.



Apertura del Dashboard.

5. Ya todo en operación el *Dashboard* muestra de arriba hacia abajo y de izquierda a derecha el grupo muestra el registro de medidas en tiempo real que provienen de los sensores ya procesadas y transmitidas por el microcontrolador ESP32-CAM vía Wi-Fi, haciendo uso del Bróker Mosquitto. En el segundo grupo se muestra el histórico de los valores detectados y en la parte inferior se muestra de forma embebida las gráficas realizadas en Grafana (a continuación se explicará en el siguiente tema), con los datos guardados en la base de datos **Habeetat_SensoresExt** de MySQL.



Ejemplo de vista del Dashboard.

Y, ¿cómo se hace interactuar Grafana con Node Red para visualizar los datos gráficos?

Graficado de variables registradas en Grafana

La intención de nuestro proyecto es que todos los datos recopilados y almacenados sean fácilmente interpretables y que mejor si se visualizan de manera gráfica. Para ello hemos utilizado Grafana, que es un software libre que permite la visualización y el formato de datos ya recopilados desde una base de datos, como MySQL.

****Nota:**** Se aconseja revisar el material de referencia indicado o se pueden consultar los siguientes vídeos realizados por el profesor del curso Hugo Vargas, en donde a detalle se explica cómo realizar la configuración, vinculado con una base de datos, el **Query** para el manejo de datos y el embebido de las gráficas en Node Red.

<https://youtu.be/4y-WazppL6U>

<https://youtu.be/r9HMQ4m5jYU>

<https://youtu.be/Xnm0O074qXE>

1. Activamos Grafana desde la consola de Ubuntu 20.04 con el siguiente comando

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

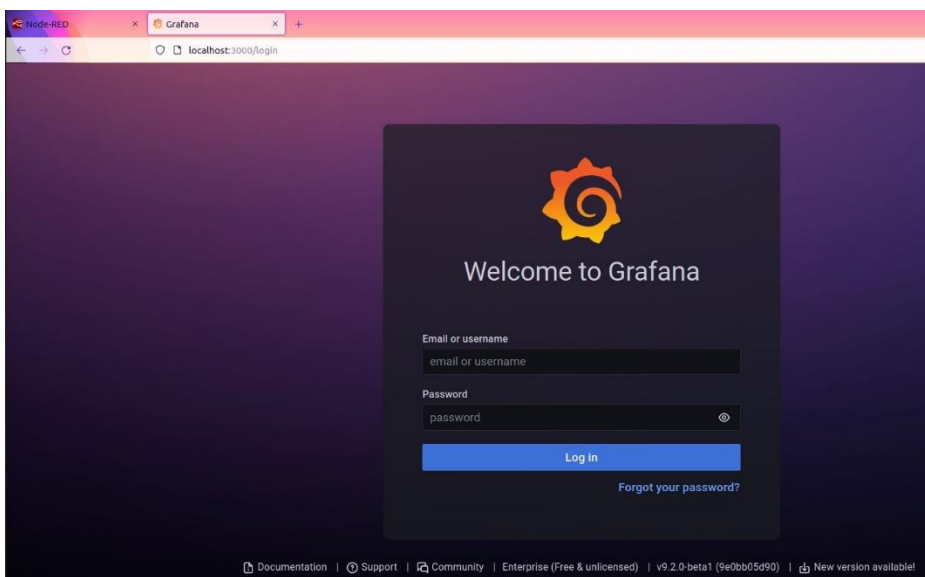
```
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox:~$ sudo /bin/systemctl start grafana-server  
[sudo] password for ikaro:  
ikaro@ikaro-VirtualBox:~$
```

Comando para iniciar Grafana desde la consola de Ubuntu 20.04.

2. Abrimos un explorador y escribimos:

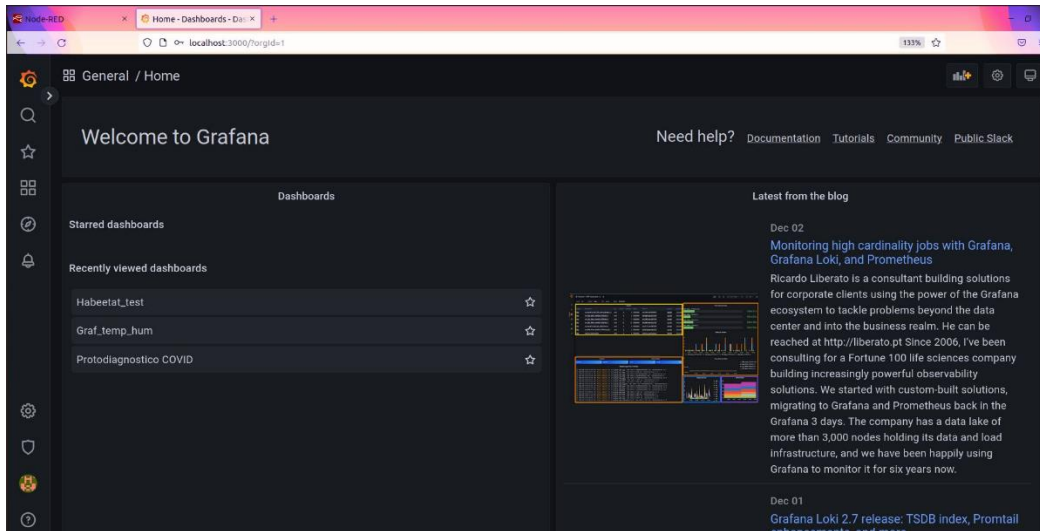
localhost:3000/

3. Registramos nuestros datos en donde se solicita.



Inicialización de Grafana.

4. Elegimos la base de datos a utilizar.



Elección de la base de datos.

5. Ya habiendo realizado toda la configuración, vinculación de la base de datos y elegido el tipo de gráfico que mejor se ajusta para el desplegado histórico de los datos recopilados, se procede a escribir los "query" correspondientes.

Query de temperatura

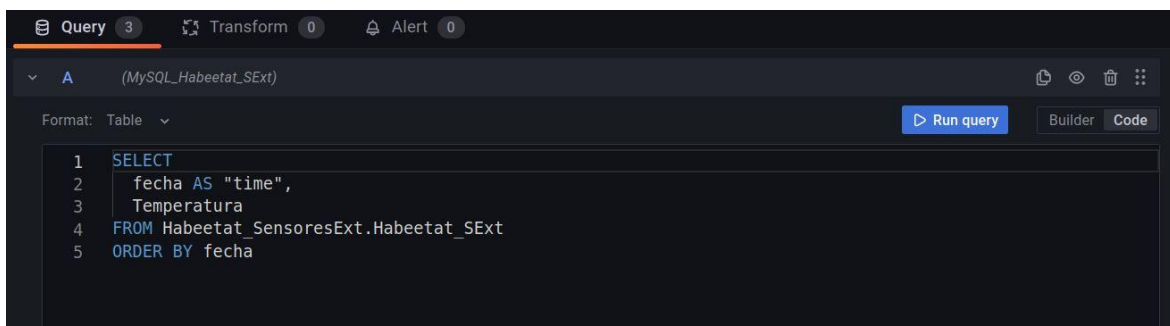
SELECT

fecha AS "time",

Temperatura

FROM Habeetat_SensoresExt.Habeetat_SExt

ORDER BY fecha



Configuración de Query de Temperatura.

Query de Humedad

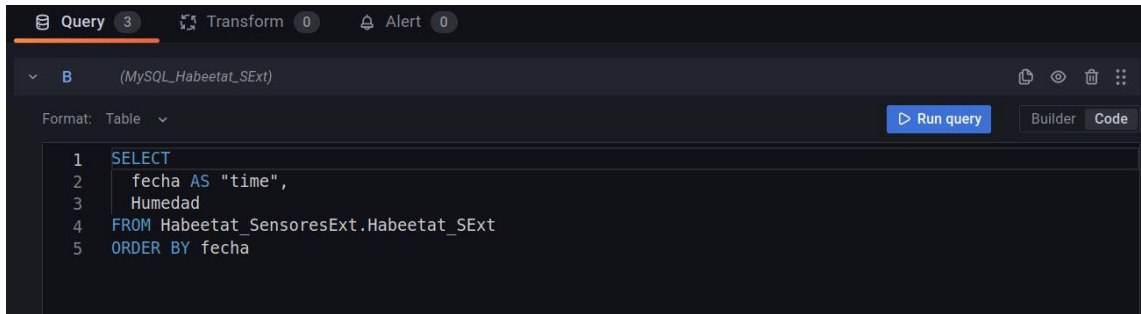
SELECT

fecha AS "time",

Humedad

FROM Habeetat_SensoresExt.Habeetat_SExt

ORDER BY fecha



Configuración de Query de Humedad.

Query de Peso

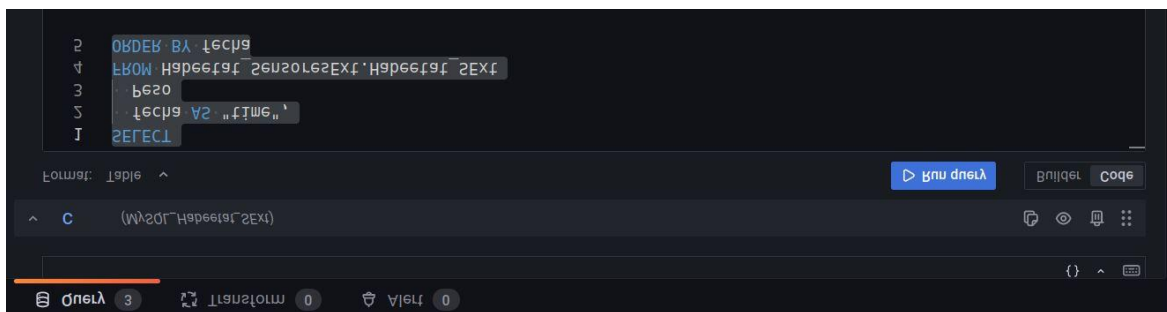
SELECT

fecha AS "time",

Peso

FROM Habeetat_SensoresExt.Habeetat_SExt

ORDER BY fecha



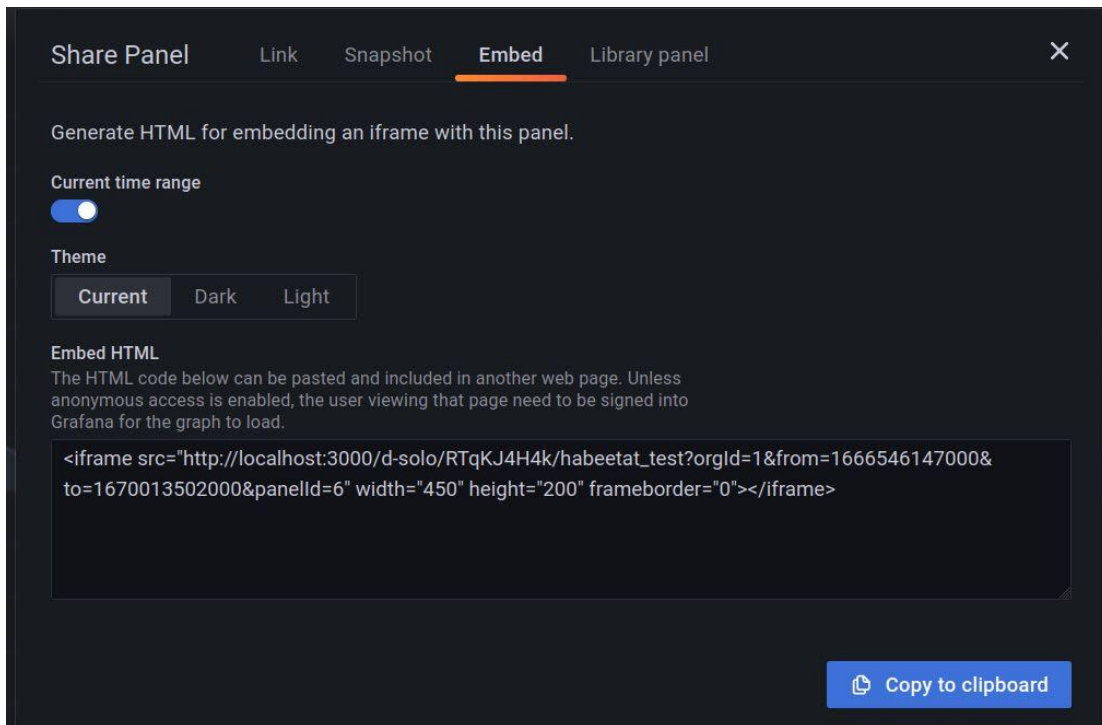
Configuración de Query de Peso.

6. Se aplican y se salvan todas las configuraciones hechas.



Gráficos correspondientes a los Query configurados.

- Se busca y se copia la **Embed HTML** para incrustarla en Node RED

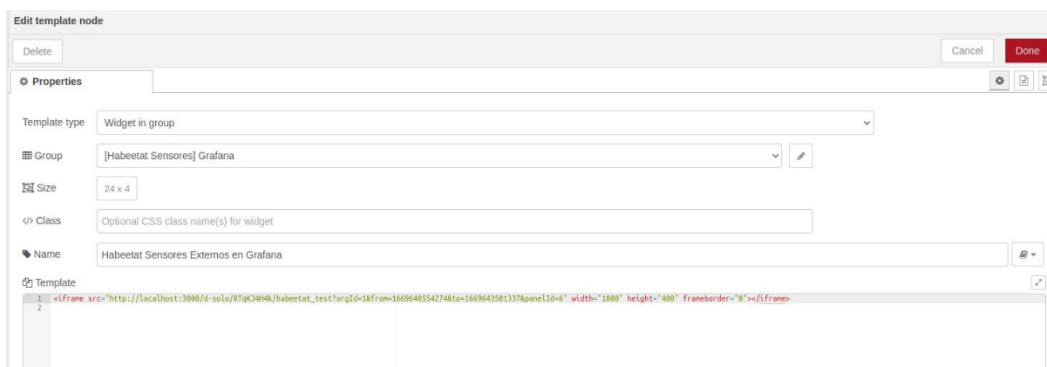


Panel.

8. Se configura el nodo HTML en Node Red

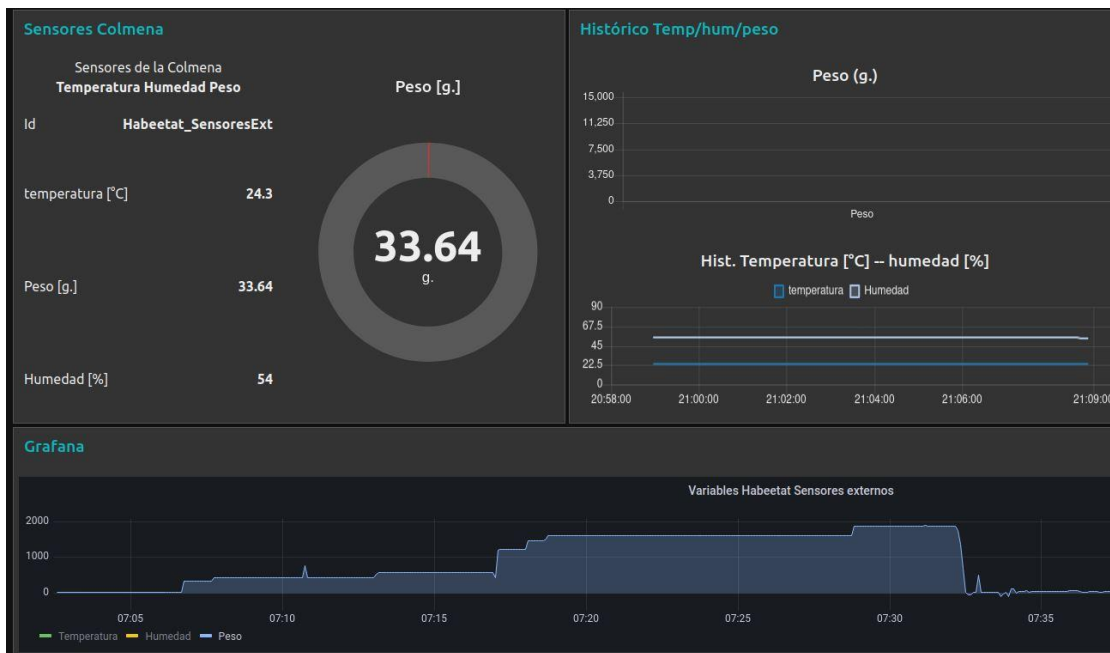


Nodo HTML.



Configuración hecha.

9. Se realiza **Deploy** al **Flow**.
10. Y queda lista la incrustación en el **Dashboard**.



Dashboard con gráficos incrustados.

Y se ha completado el **Dashboard** para visualizar los valores de las variables que ayudan a verificar la salud de la colmena.

Solicitud de estado de las condiciones de las variables en tiempo real por mensaje de un Bot en Telegram

¿Podríamos consultar los datos de la colmena haciendo una petición de un mensaje de texto por algún servicio de mensajería?

La respuesta es sí, se pueden solicitar los datos de las variables temperatura, humedad, peso y CO2; relacionadas con la colmena e incluso una imagen de su estado y en base a la información recopilada, una predicción esperada del peso de la colmena para los siguientes dos días.

¿Y cómo se hace?

Al estar utilizando Node Red, se le pueden agregar **Nodos** que te permiten hacer uso de **Telegram** para el envío y recepción de mensajes.

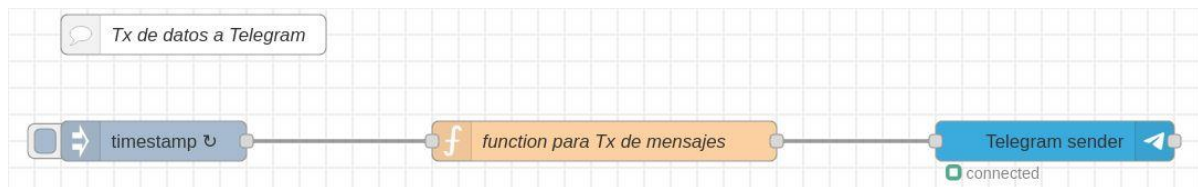
A continuación se agregan dos referencias. La primera es un tanto informal, pero te ayudarán a realizarlo de manera simple. La segunda te enlaza a las hojas de especificaciones de Telegram, para que hagas un estudio formal de lo que son los **Bots** y su utilidad.

<https://youtu.be/09XK3eeEHSU>

<https://core.telegram.org/bots>

Para mandar mensajes la configuración queda de la siguiente manera:

1. Interconectar los nodos, en Node Red, como se muestra en la figura.



Nodos para hacer uso de Telegram.

2. Se configura el *node function* con el siguiente código.

```
msg.payload = {};
```

```
msg.payload.chatId = -825211725//-775229224, -825211725;
```

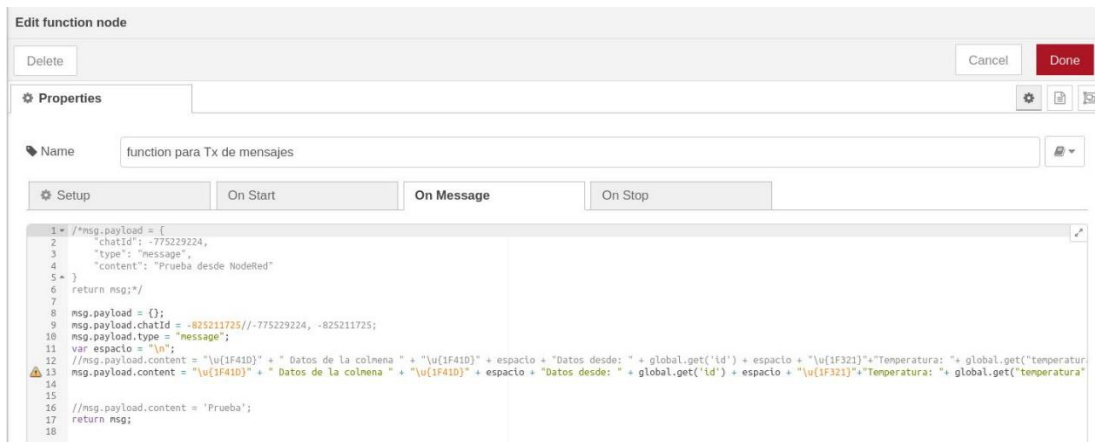
```
msg.payload.type = "message";
```

```
var espacio = "\n";
```

```
//msg.payload.content = "\u{1F41D}" + " Datos de la colmena " + "\u{1F41D}" + espacio + "Datos desde: " + global.get('id') + espacio + "\u{1F321}" + "Temperatura: " + global.get("temperatura")+ " °C"+espacio + "\u{2614}" + "Humedad: " + global.get("humedad") + " %"+ espacio + "\u{2696}" + "Peso de la Colmena: " + global.get("peso")+ " grms." + espacio + "Envia <consulta> para acceder a los datos más actuales de la colmena."
```

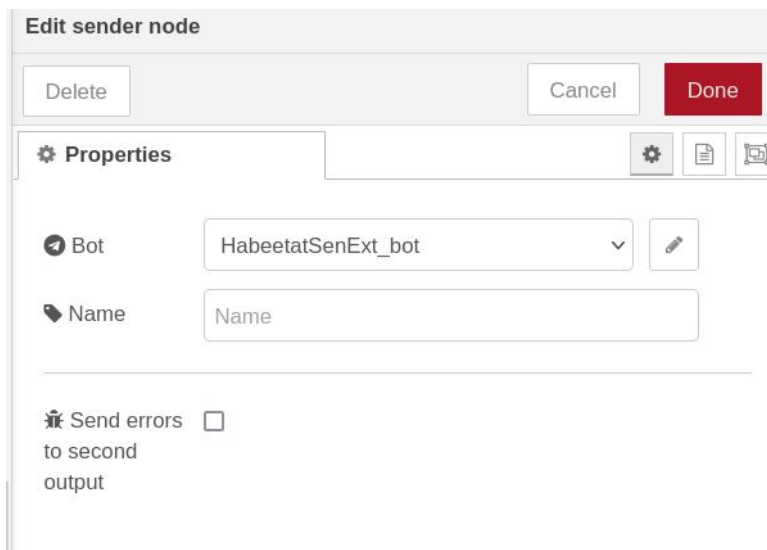
```
msg.payload.content = "\u{1F41D}" + " Datos de la colmena " + "\u{1F41D}" + espacio + "Datos desde: " + global.get('id') + espacio + "\u{1F321}" + "Temperatura: " + global.get("temperatura")+ " °C"+espacio + "\u{2614}" + "Humedad: " + global.get("humedad") + " %"+ espacio + "\u{2696}" + "Peso de la Colmena: " + global.get("peso")+ " grms." + espacio + ""
```

```
return msg;
```



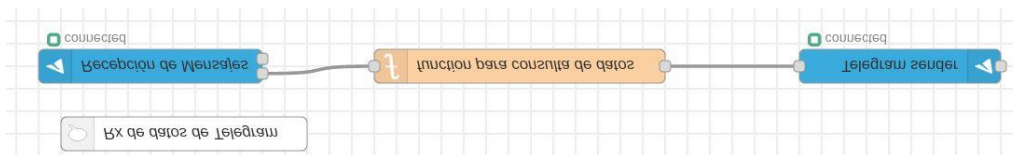
Configuración de Edit function node.

- Se configura el **sender node**



Configuración de Edit sender node.

- Se interconectan los nodos de recepción de datos de la siguiente manera



Nodos de recepción de mensaje.

- Se configura el **receiver node**

Edit receiver node

Delete Cancel Done

Properties

Bot: HabeetatSenExt_bot

Name: Recepción de Mensajes

Download Directory: Download directory

Filter: ☐ commands (from configured command nodes)

Configuración de Edit receiver node.

6. Se configura el *function node* para el receptor con el siguiente código.

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

    msg.payload.chatId = -825211725//-775229224, -825211725;

    msg.payload.type = 'message';

    var espacio = " \n";

    msg.payload.content = "\u{1F41D}" + " Datos de la colmena " + "\u{1F41D}" + espacio + "Datos
desde: " + global.get('id') + espacio + "\u{1F321}"+"Temperatura: " + global.get("temperatura")+ "
°C"+espacio + "\u{2614}"+"Humedad: " + global.get("humedad") + " %"+ espacio + "\u{2696}"+"Peso
de la Colmena: "+ global.get("peso")+ " g."+ "";

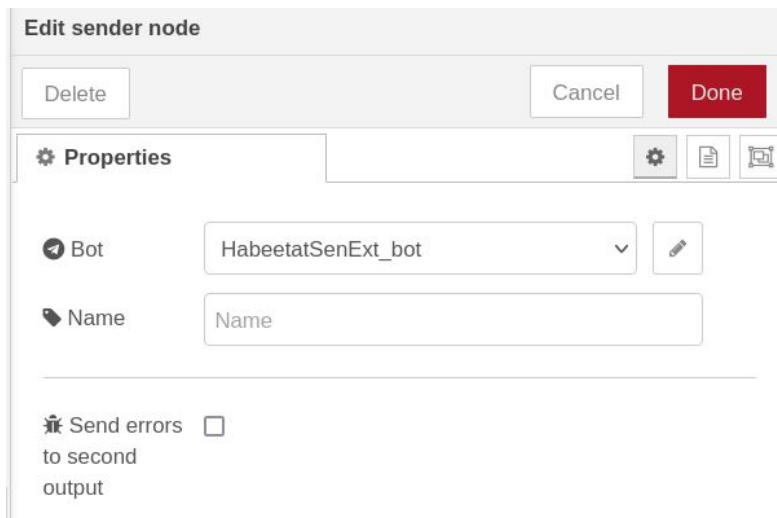
    //msg.payload.content = "datos"

    return msg;
}
```



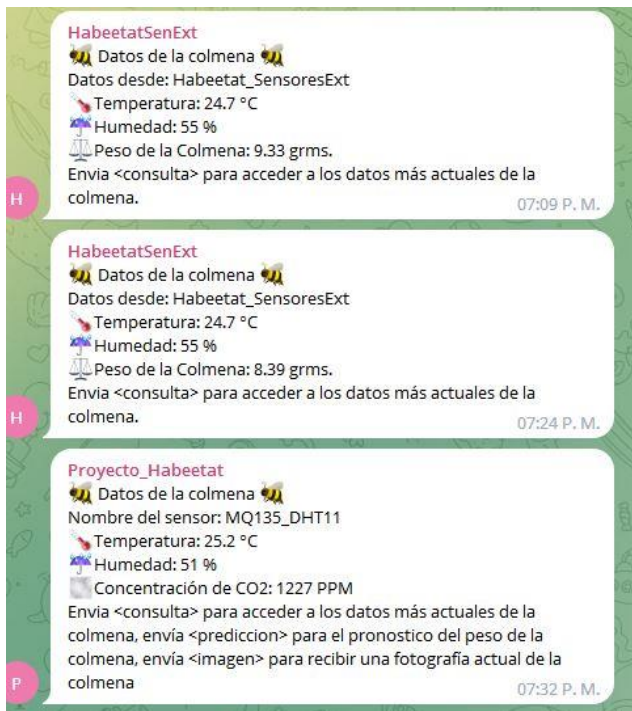

Configuración de Edit function node.

7. Se configura el **sender node**



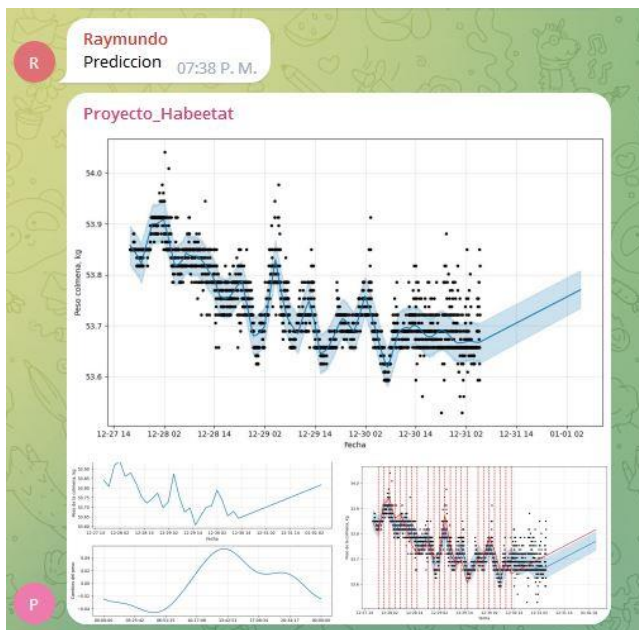
Configuración de Edit sender node.

8. Y ya funcionando se observa en Telegram de la siguiente manera, la consulta de valores.



Datos recibidos en Telegram.

9. Y la consulta de predicción de la siguiente manera.



Gráfica de predicción del peso de la colmena por Telegram.

Y ha quedado funcional la parte del proyecto encargada de la detección de la temperatura, humedad y peso de la colmena, control de carga conectada al suministro eléctrico, configuración en Node Red, el Dashboard para la visualización de datos, el respaldo de la información en una base de datos en

MySQL, graficado en Grafana, incrustado en el Dashboard de las gráficas obtenidas en Grafana y envío y recepción de mensajes por Telegram.

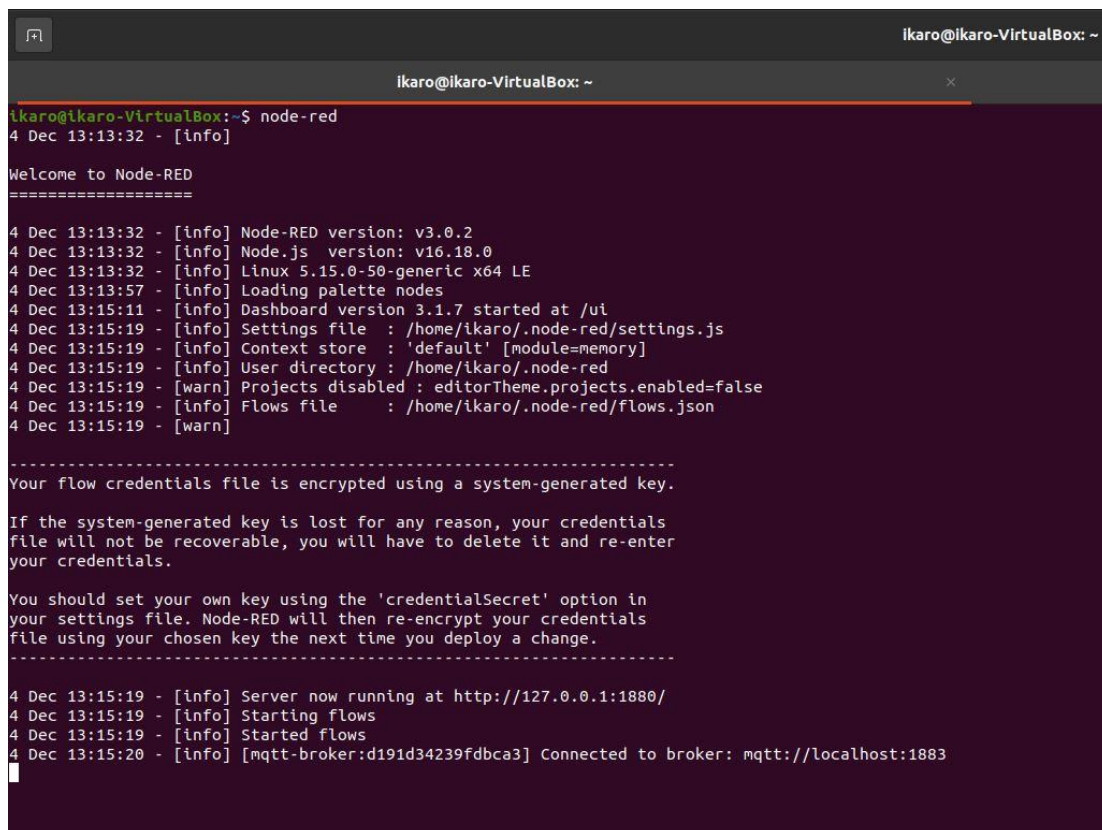
Instrucciones de operación

En lo que respecta a esta parte de la implementación del proyecto, no es muy complicada de utilizar, una vez que todas sus partes han sido creadas y programadas. Solo hay que seguir los siguientes pasos:

1. Inicialice su máquina virtual en donde tiene instalado Ubuntu 20.04.
2. Ya corriendo el sistema operativo Ubuntu 20.04, se abre una terminal y se escribe el comando

node-red

para inicializar esta herramienta de programación.



```
ikaro@ikaro-VirtualBox: ~  
ikaro@ikaro-VirtualBox:~$ node-red  
4 Dec 13:13:32 - [info]  
  
Welcome to Node-RED  
=====
```

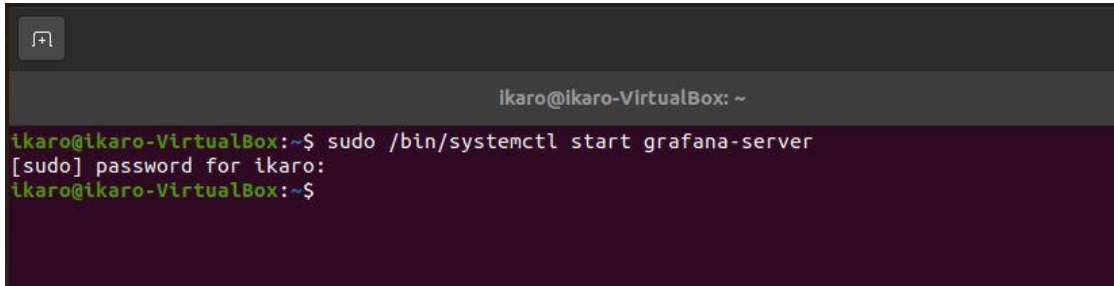
```
4 Dec 13:13:32 - [info] Node-RED version: v3.0.2  
4 Dec 13:13:32 - [info] Node.js version: v16.18.0  
4 Dec 13:13:32 - [info] Linux 5.15.0-50-generic x64 LE  
4 Dec 13:13:57 - [info] Loading palette nodes  
4 Dec 13:15:11 - [info] Dashboard version 3.1.7 started at /ui  
4 Dec 13:15:19 - [info] Settings file : /home/ikaro/.node-red/settings.js  
4 Dec 13:15:19 - [info] Context store : 'default' [module=memory]  
4 Dec 13:15:19 - [info] User directory : /home/ikaro/.node-red  
4 Dec 13:15:19 - [warn] Projects disabled : editorTheme.projects.enabled=false  
4 Dec 13:15:19 - [info] Flows file : /home/ikaro/.node-red/flows.json  
4 Dec 13:15:19 - [warn]  
  
-----  
Your flow credentials file is encrypted using a system-generated key.  
  
If the system-generated key is lost for any reason, your credentials  
file will not be recoverable, you will have to delete it and re-enter  
your credentials.  
  
You should set your own key using the 'credentialSecret' option in  
your settings file. Node-RED will then re-encrypt your credentials  
file using your chosen key the next time you deploy a change.  
-----  
  
4 Dec 13:15:19 - [info] Server now running at http://127.0.0.1:1880/  
4 Dec 13:15:19 - [info] Starting flows  
4 Dec 13:15:19 - [info] Started flows  
4 Dec 13:15:20 - [info] [mqtt-broker:d191d34239fdbca3] Connected to broker: mqtt://localhost:1883
```

Inicialización de Node Red.

3. En otra consola de Ubuntu 20.04, se escribe el comando

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

para activar Grafana. En este paso solicita la contraseña de administrador, después de escribirla, ya está activada la herramienta.

A terminal window with a dark background. The prompt is 'ikaro@ikaro-VirtualBox: ~'. The command 'sudo /bin/systemctl start grafana-server' is entered. The prompt changes to '[sudo] password for ikaro:', and then back to 'ikaro@ikaro-VirtualBox: ~\$' after the password is entered.

```
ikaro@ikaro-VirtualBox:~$ sudo /bin/systemctl start grafana-server
[sudo] password for ikaro:
ikaro@ikaro-VirtualBox:~$
```

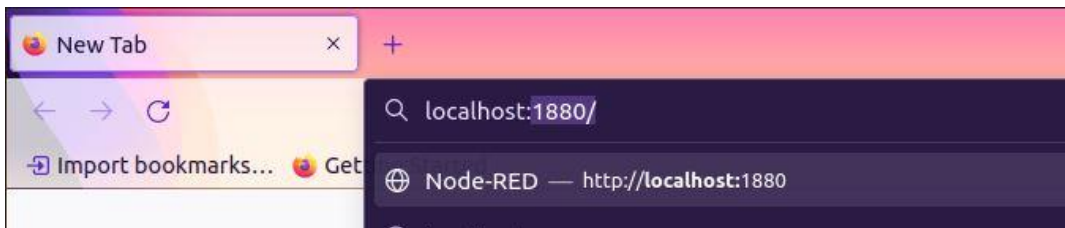
Inicialización de Grafana.

4. Conectar el FTDI mediante un cable USB a USB mini a una fuente de voltaje, por ejemplo, el que utiliza para suministrar carga a la batería de su smartphone.

5. Se abre una página de un explorador de internet y se escribe

localhost:1880/

para ingresar a Node Red.

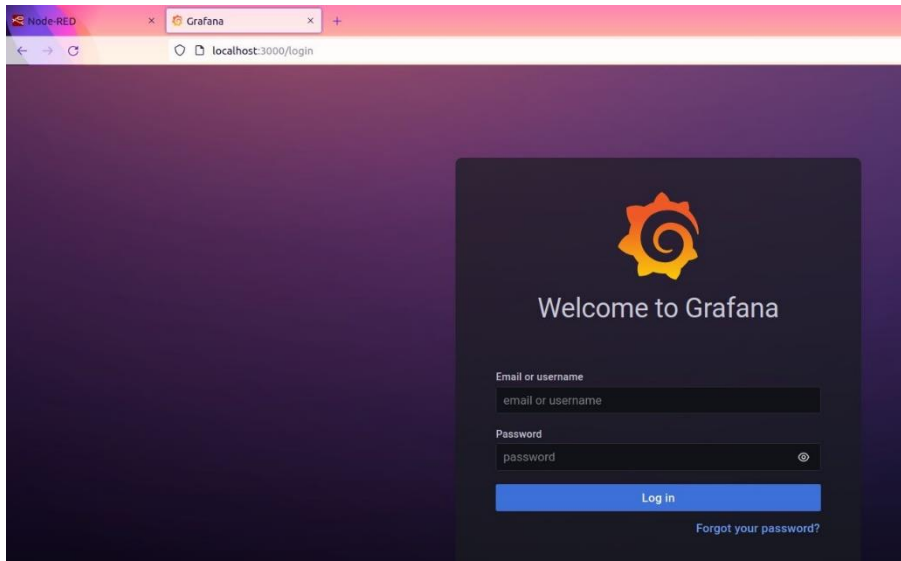


Ingreso a Node Red.

6. Se abre otra pestaña del navegador y se escribe

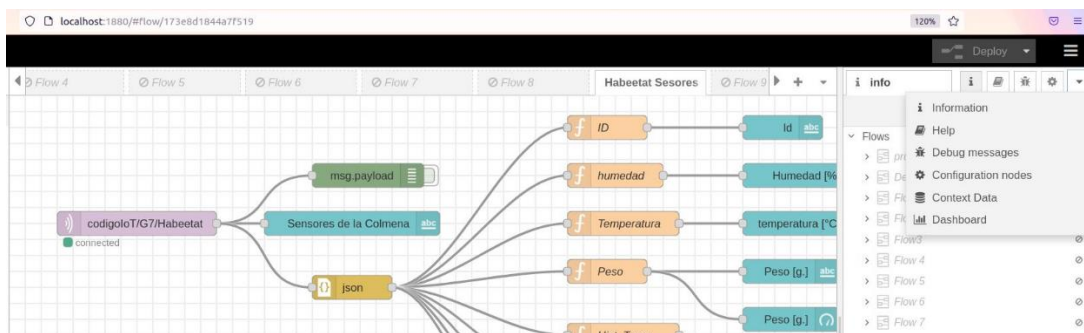
localhost:3000/

para activar Grafana, en donde hay que escribir el nombre del usuario y contraseña.

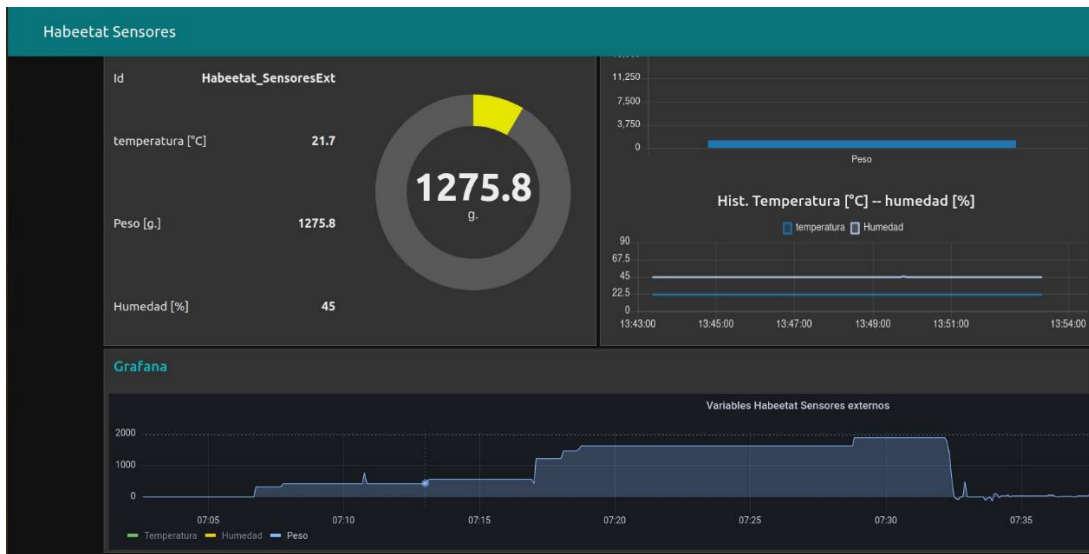


Ingreso a Grafana.

7. Desde Node Red se activa el Dashboard y ya se pueden visualizar las medidas de temperatura y humedad ambientales cercanas a la colmena así como su peso que va acumulando.



Inicialización del Dahnboard.



Visualización de datos y gráficas en el Dashboard.

Y ya también está listo para recibir el estado del sistema por mensaje de Telegram.

Evidencias

En los siguientes vídeos se describe lo realizado en la parte de programación y de diseño de circuito electrónico.

<https://youtu.be/f8K0dvlxA-s>

<https://youtu.be/PM99izaFDMA>