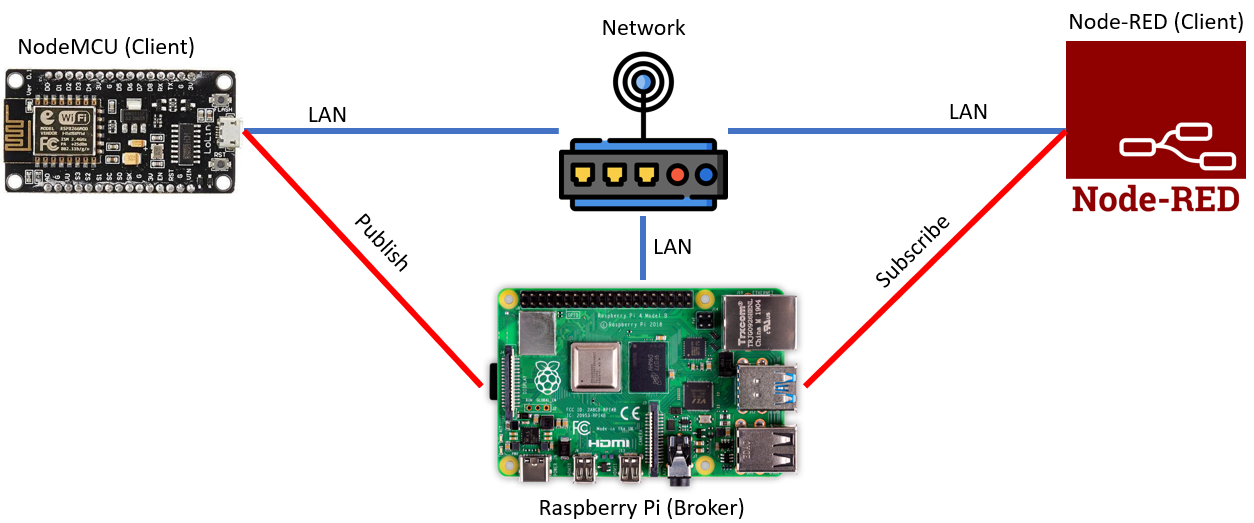
**MQTT**

**Introducción**

MQTT (Message Queue Server Telemetry Transport) es un protocolo de mensajería basado en publicación-suscripción. Funciona sobre el protocolo TCP / IP. Está diseñado para conexiones con ubicaciones remotas donde se requiere una "huella de código pequeño" o el ancho de banda de la red es limitado. El patrón de mensajería de publicación-suscripción requiere un intermediario de mensajes (Broker).



**Instalación y configuración**

Para la utilización del protocolo MQTT necesitaremos instalar un Broker en nuestro equipo Host que será el Raspberry Pi. Existen diferentes tipos de Broker con diferentes características en nuestro caso utilizaremos el Broker llamado mosquitto.

Antes de la instalación debemos asegurarnos de que nuestra Raspberry Pi este actualizada:

sudo apt update

Para instalar mosquitto y los servicios de cliente para realizar pruebas, ejecutaremos el siguiente comando:

**sudo apt install -y mosquitto mosquitto-clients**

**Una vez instalado los dos paquetes, utilizando un editor de texto, abriremos el archivo de configuración. En nuestro caso utilizaremos nano:**

sudo nano /etc/mosquitto/mosquitto.conf

Borraremos la siguiente línea que esta al final del archivo:

include\_dir /etc/mosquitto/conf.d

Luego agregaremos las siguientes líneas:

allow\_anonymous

falsepassword\_file /etc/mosquitto/pwfile

listener 1883

La primera línea permitirá que, para conectarse a nuestro Broker, se necesitara autenticación, de esta manera evitaremos que otras personas se conecten.

La segunda línea establece la ubicación del archivo que contendrá los nombres de usuarios y contraseñas los cuales están encriptados.

La tercera línea hará que nuestro Broker se comunique por el puerto 1883

Para agregar nuestro primer usuario ejecutaremos y nos pedirá ingresar una contraseña:

sudo mosquitto\_passwd -c /etc/mosquitto/pwfile username

Siendo username el nombre de usuario a utilizar.

Lo único que queda es reiniciar nuestro Raspberry Pi para que se apliquen los cambios.

sudo reboot

**Prueba de suscripción y publicación**

El protocolo MQTT funciona mediante un modelo de publicación y suscripción.

Para probar nuestro Broker abriremos dos terminales, en el primero nos subscribiremos a un topic que crearemos en un momento.

mosquitto\_sub -h 192.168.233.32 u- user -P pass -t “topic”

192.168.233.32 es el IP del Broker. Para saber el IP de nuestra Raspberry Pi ejecutaremos:

hostname -I

user es el nombre de usuario que creamos anteriormente

pass es la contraseña asociada al nombre de usuario anterior

topic es el tema o tópico al cual se suscribirá

En la segunda terminal publicaremos un mensaje:

mosquitto\_pub -h 192.168.233.32 -u user -P pass -t “topic” -m “mensaje”

Cambiamos sub a pub y agregamos el mensaje a publicar.

Al regresar a la primera terminal que esta subscrita nos aparecerá el mensaje que acabamos de enviar.

**SQLite**

**Instalación**

Para instalar SQLite en nuestro Raspberry Pi ejecutaremos el siguiente comando:

sudo apt-get install sqlite3

Con esto SQLite estará instalado y listo para usar

**Creación de base de datos**

Crearemos una base de datos con el siguiente comando:

sqlite name

Siendo name el nombre del archivo de base de datos

**Creación de tablas**

CREATE TABLE tablename (column datatype);

Siendo tablename el nombre de la tabla, column el nombre de la columna y datatype el tipo de dato de la columna.

**Insertar datos en una tabla**

Para insertar datos se utilizan comandos SQL en formas de query como, por ejemplo:

INSERT INTO tablename (column) VALUES (data);

Siendo tablename el nombre de la tabla, column el nombre de la columna y data el dato o valor a ingresar en tal columna

**Mostrar datos de una tabla**

SELECT \* FROM tablename;

**Borrar tabla**

DROP TABLE IF EXISTS tablename;

**Node-RED**

**Introducción**

Node-RED es una herramienta multiplataforma open source de desarrollo basada en flujo para la programación visual desarrollada originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte de Internet of Things. Escrito en JavaScript, utiliza la plataforma Node.js y se encuentra bajo la licencia Apache 2.0.

**Instalación y configuración**

Node-RED viene preinstalado en dispositivos Raspberry Pi con sistema operativo Raspbian, pero no posee npm (node packet manager) que sirve para agregar extensiones que necesitaremos.

Abriremos una terminal y ejecutaremos:

update-nodejs-and-nodered

Este comando actualizara nuestra versión de Node.js, Node-RED y instalara npm.

Node-RED puede ser accedido por cualquiera que este conectado a nuestra red de área local por lo es recomendable agregar autenticación.

Cambiamos al directorio de Node-RED:

cd ~/.node-red

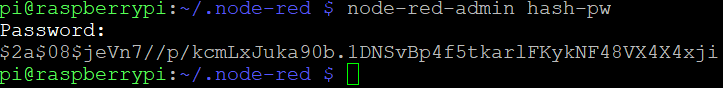
Instalamos los paquetes de administrador de Node-RED:

sudo npm install -g node-red-admin

Ahora podremos crear un hash (contraseña cifrada) para que la contraseña no sea leíble en el archivo de configuración.

node-red-admin hash-pw

Al ingresar el comando anterior nos pedirá ingresar una contraseña y después nos generará un hash de esta contraseña.

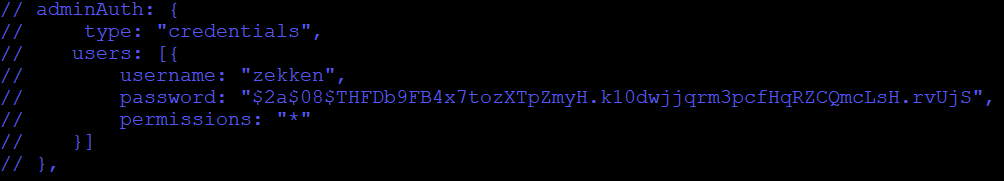


Seleccionamos el hash y lo copiaremos.

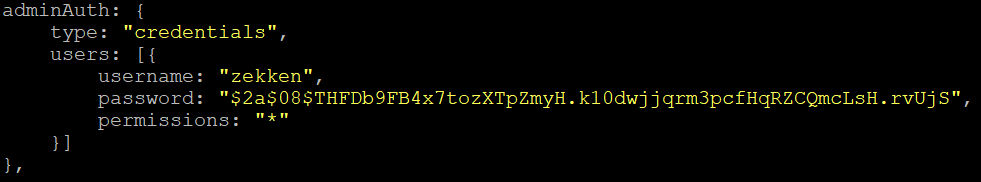
Abrimos otra terminal y editaremos el archivo de configuración de Node-RED con un editor de texto como nano:

nano ~/.node-red/settings.js

Buscamos la configuración adminAuth y lo quitaremos los signos de comentario (//)



Editamos el nombre de usuario y en la contraseña quitaremos el hash viejo y agregaremos el hash que generamos anteriormente.



Para proteger la interfaz de usuario también agregaremos autenticación. Quitaremos el signo de comentario en la configuración de httpNodeAuth.



Agregamos el mismo nombre de usuario y el hash.



Guardamos y salimos del archivo.

Ahora Node-RED tendrá autenticación para poder ser accedido.

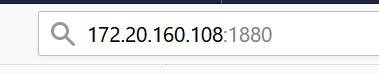
Para que Node-RED se inicie automáticamente cuando encendamos nuestro Raspberry Pi ejecutaremos:

sudo systemctl enable nodered.service

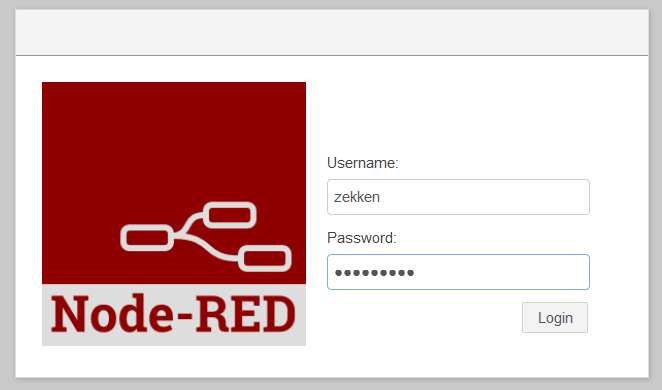
Con esto el servicio de Node-RED está completamente configurado.

**Para acceder a Node-Red**

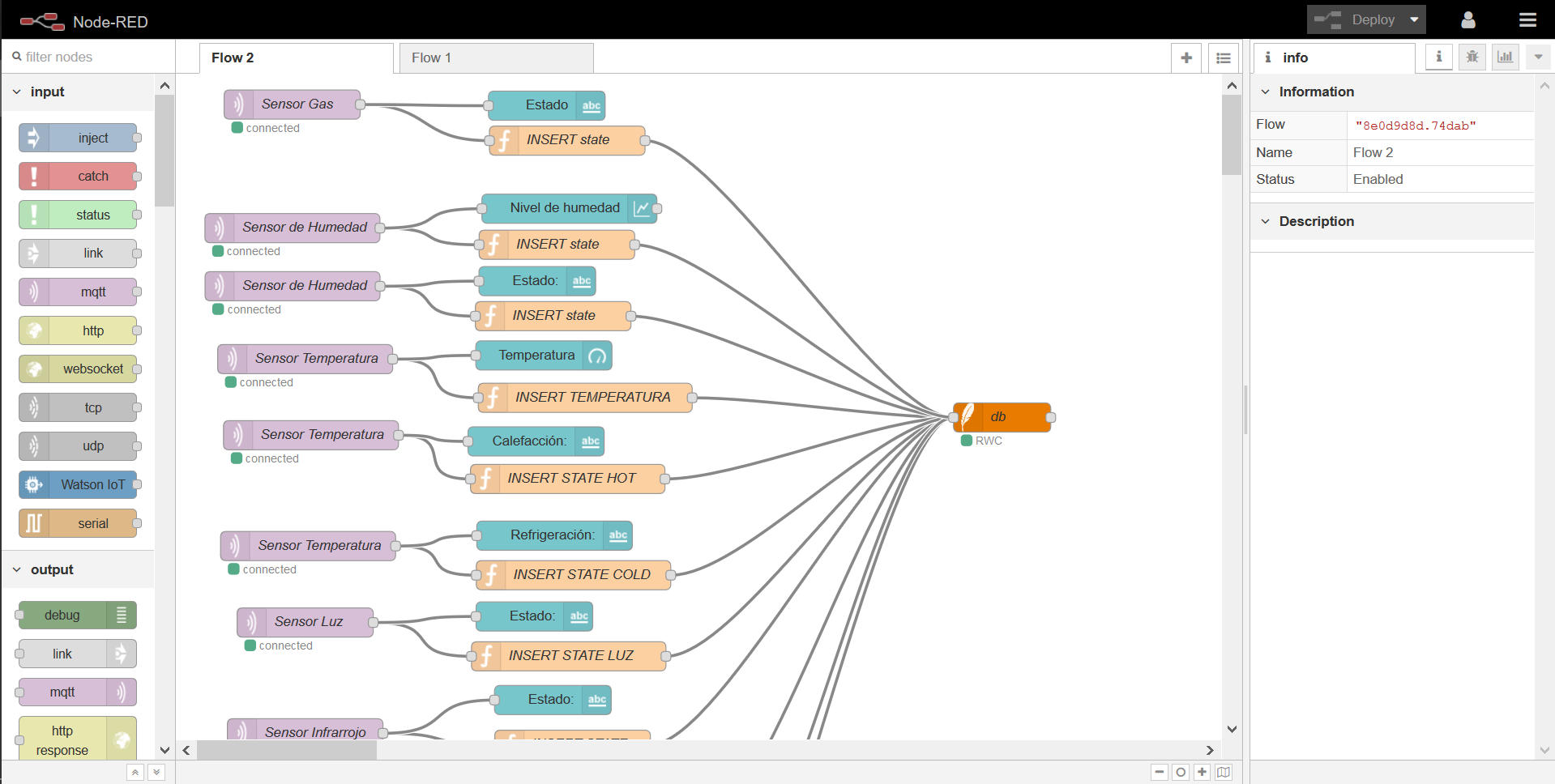
Escribimos en el navegador de nuestra preferencia la dirección IP de nuestro servidor, en nuestro caso, la Raspberry Pi. Donde 172.20.160.108 es la dirección IP del servidor y utilizamos el puerto 1880 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión.



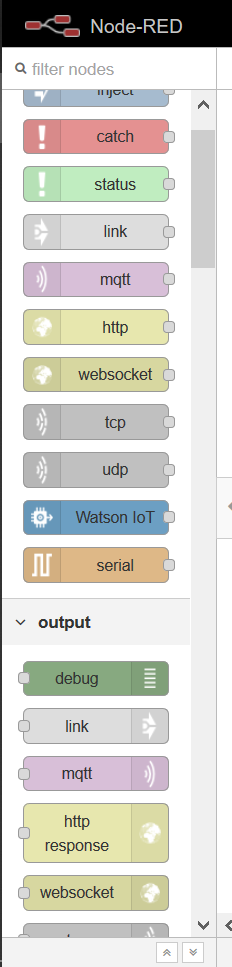
Node-Red pedirá autenticación para poder acceder, en este caso el usuario es **zekken**, password es **lilzekken.**



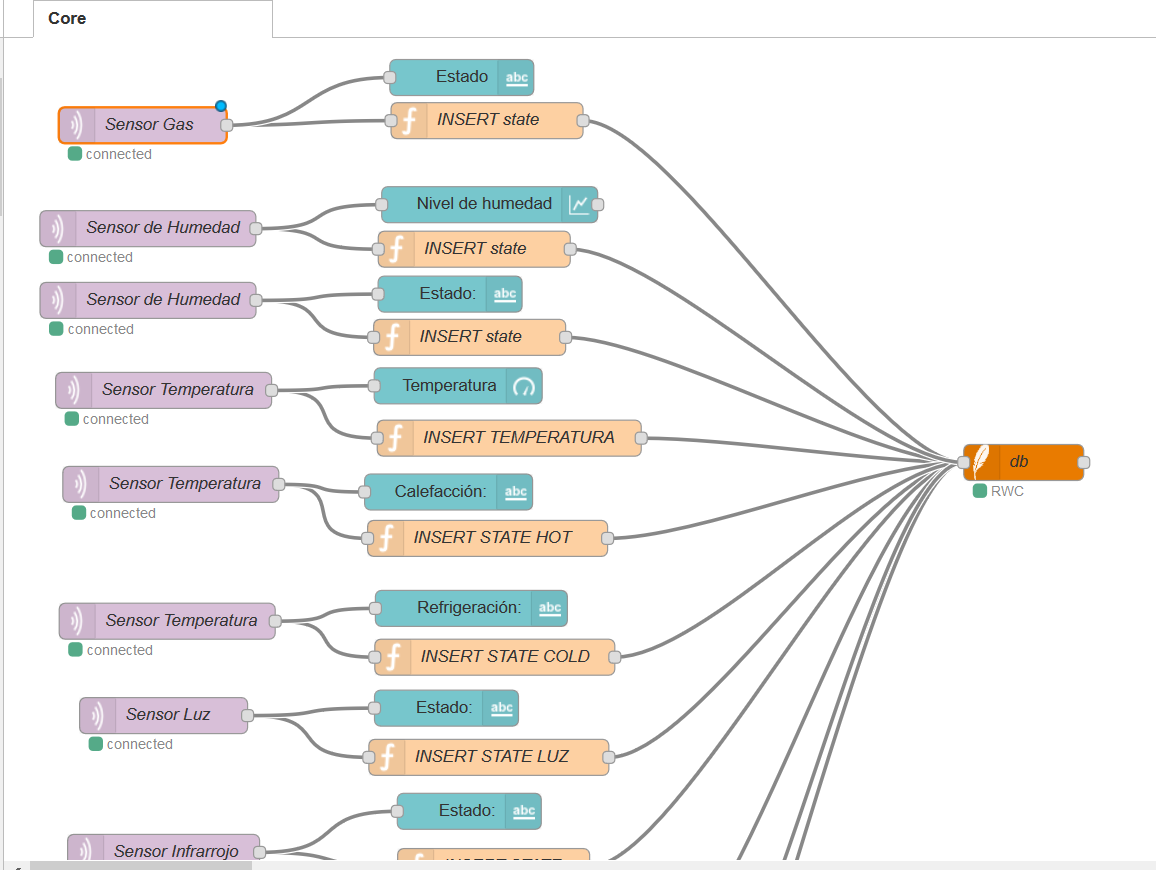
Esta es la pantalla principal de Node-Red



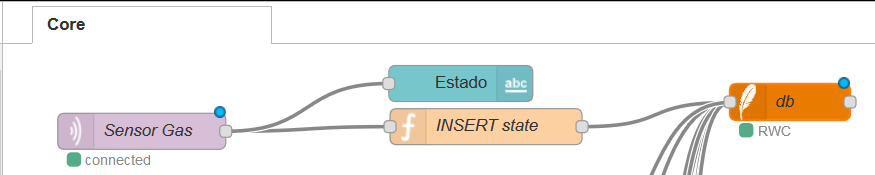
En la parte izquierda se puede encontrar la paleta de nodos, contiene nodos preinstalados, pero también se pueden añadir nodos, tanto creados por ti mismo o por la comunidad.



Aquí se encuentra el flujo, es donde añadiremos los nodos para su configuración y conexión entre ellos.

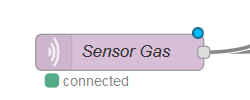


**Sensor de gas**



Los nodos utilizados son:

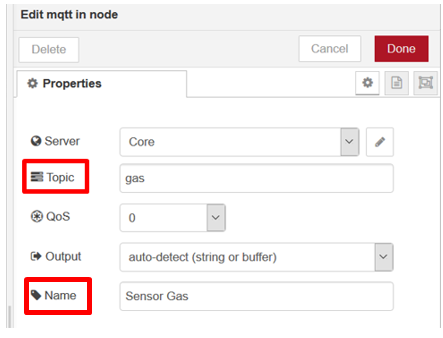
* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



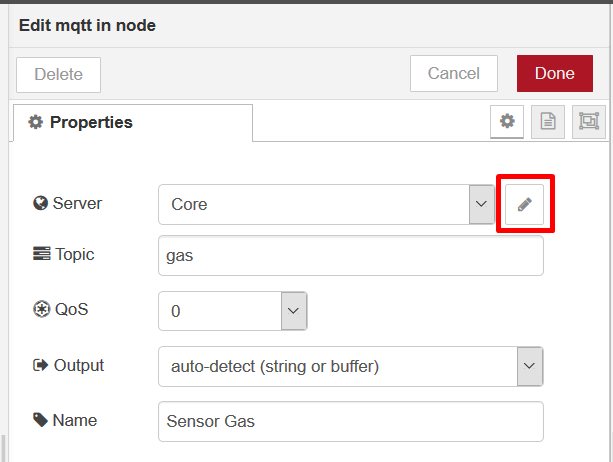
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

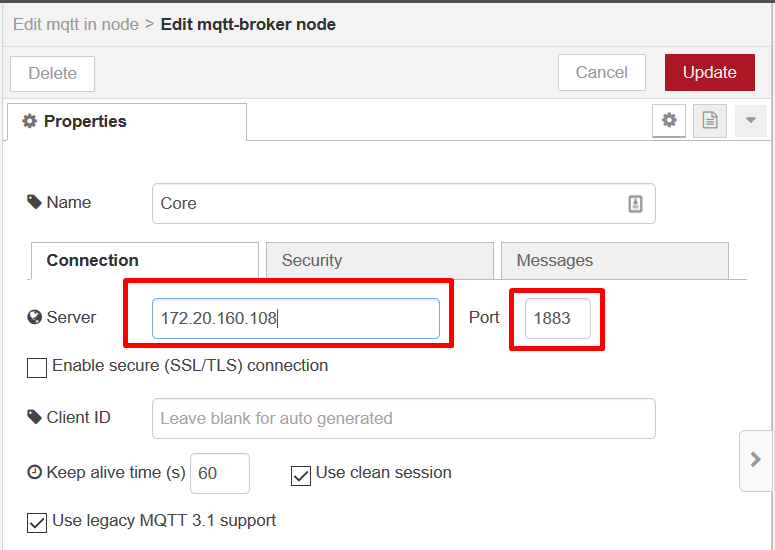
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.



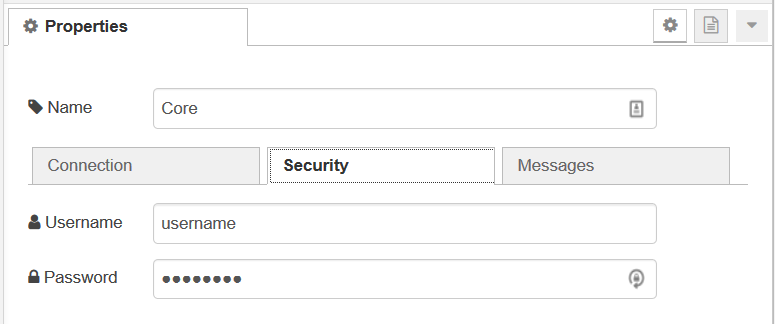
Para configurar el servidor hacemos clic en la imagen del lápiz.



Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

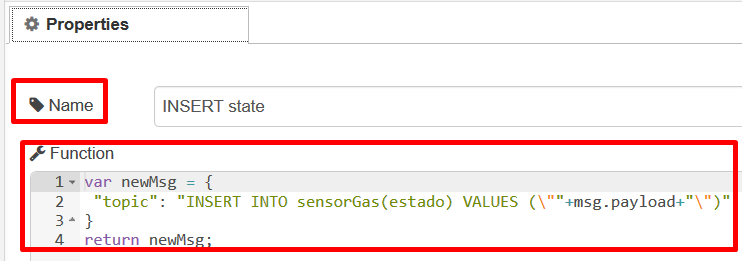


**Configuración del nodo**

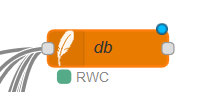
En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



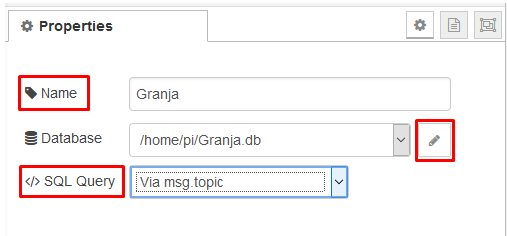
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

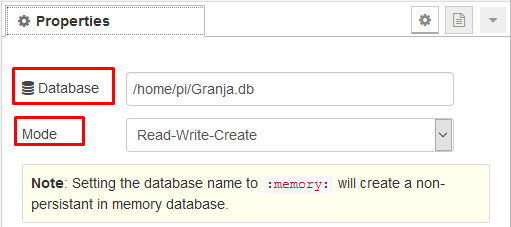
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

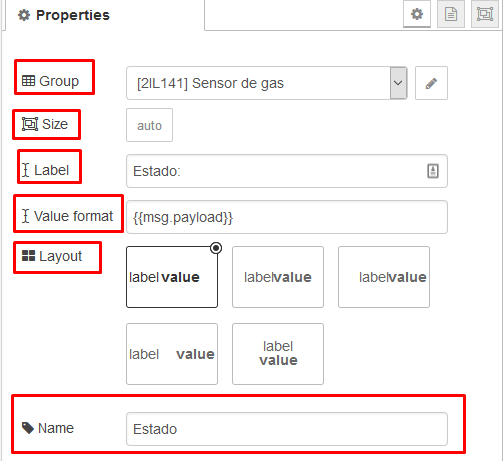
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



* Dashboard node: Mostrará un campo de texto no editable en la interfaz de usuario.



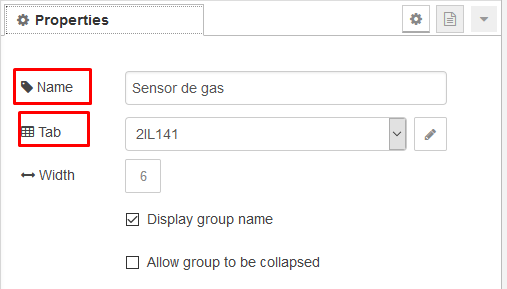
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

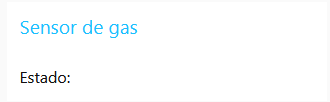
Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

Es la configuración en la que queremos que se muestre el **label** y el **value**

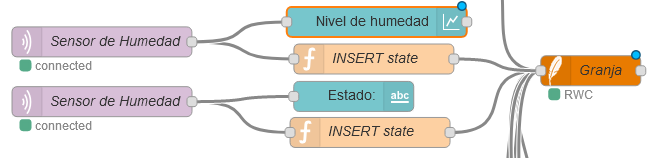


1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él

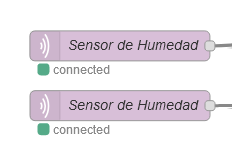


**Sensor de humedad**



Nodos utilizados:

* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



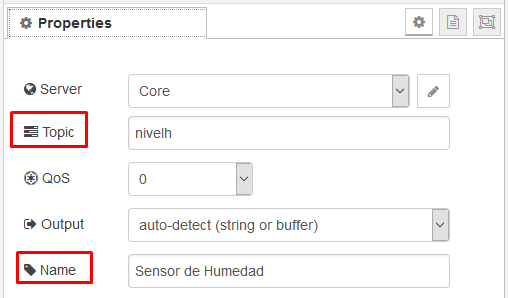
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

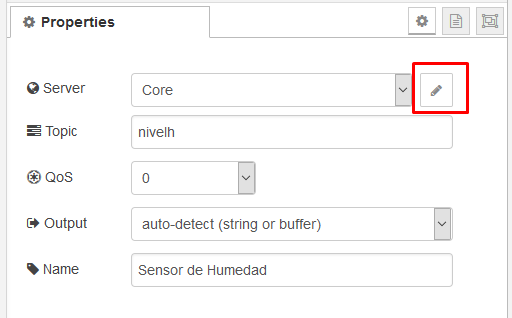
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.

En este caso particular se crean 2, cada uno trabajará bajo un topic diferente.

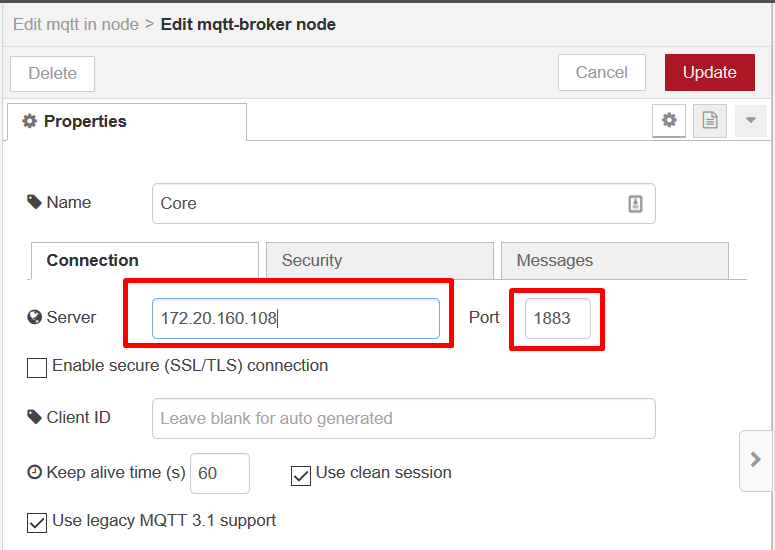
Uno llamado **nivelh** y otro llamado **humedad**.



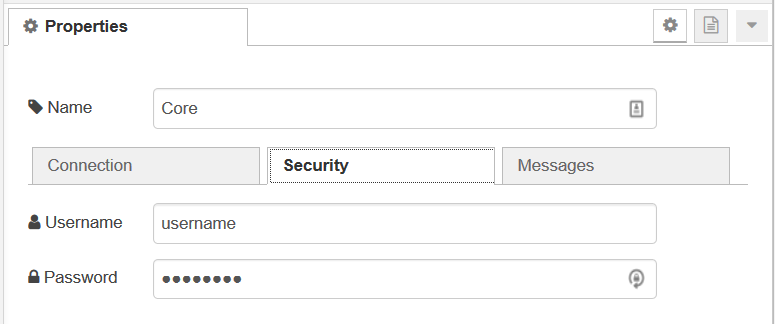
Para configurar el servidor hacemos clic en la imagen del lápiz.



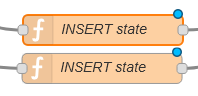
Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

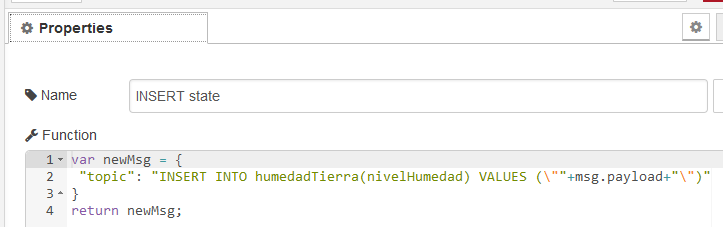


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



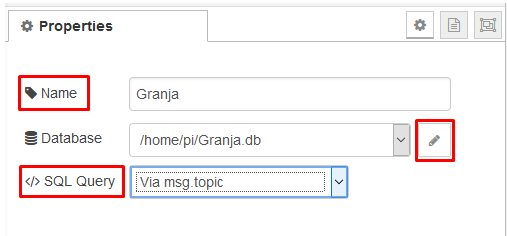
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

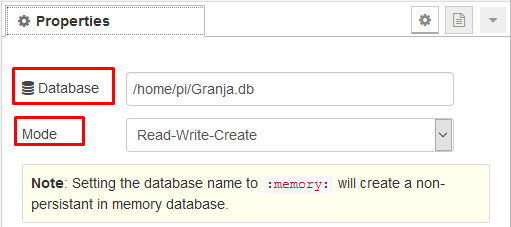
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



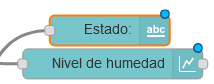
Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

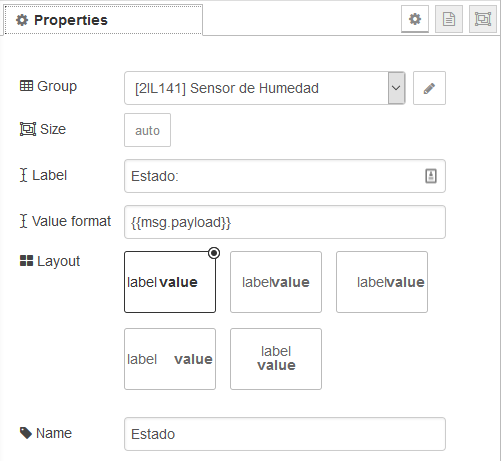
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



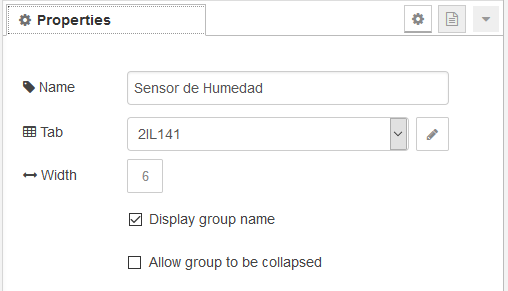
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

Es la etiqueta que tendrá en el dashboard

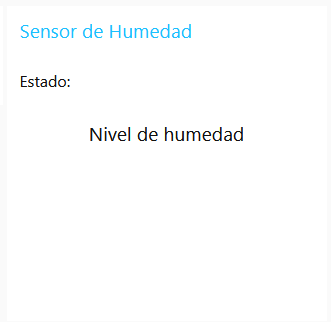
1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

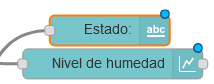
Es la configuración en la que queremos que se muestre el **label** y el **value**

**NOTA:** Todos los nodos de dashboard bajo el mismo nombre se agruparán de la siguiente manera.

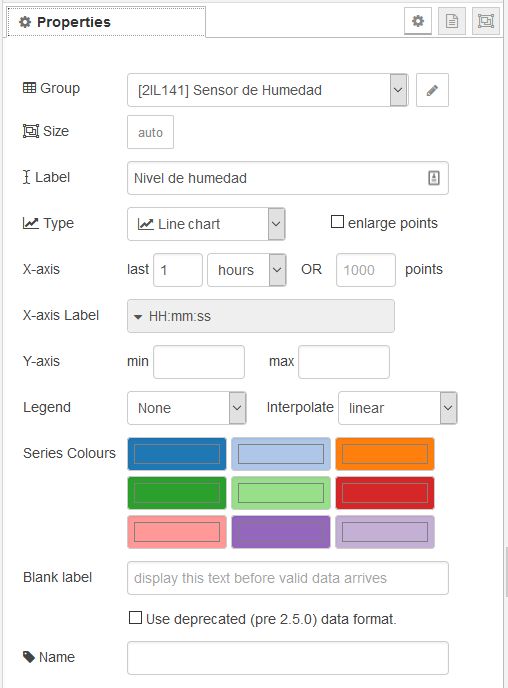


1. **Name**

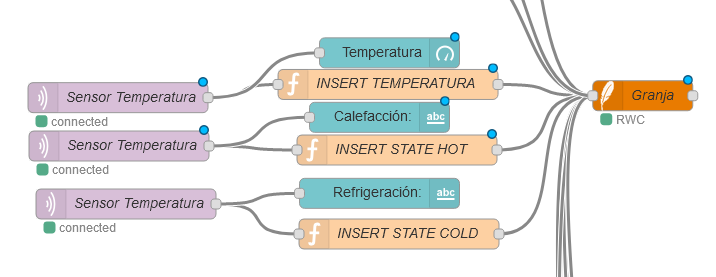
Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él



Para la gráfica son prácticamente los mismos pasos con respecto al tab y grupo, el resto son configuraciones sobre como quieres que se vea la tabla, en nuestro caso, dejamos todos los valores por defecto.

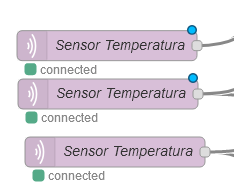


**Sensor de temperatura**



Nodos utilizados

* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



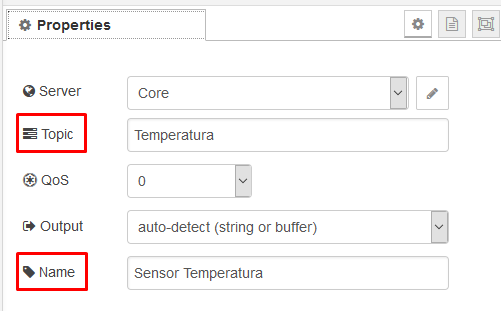
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

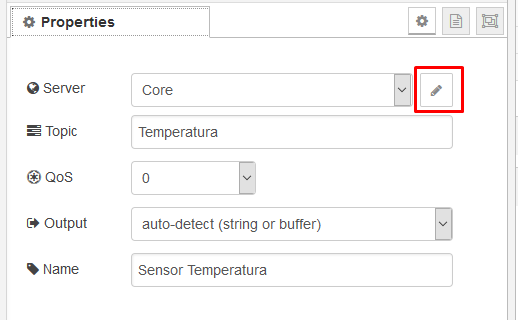
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.

En este caso particular se crean 3, cada uno trabajará bajo un topic diferente.

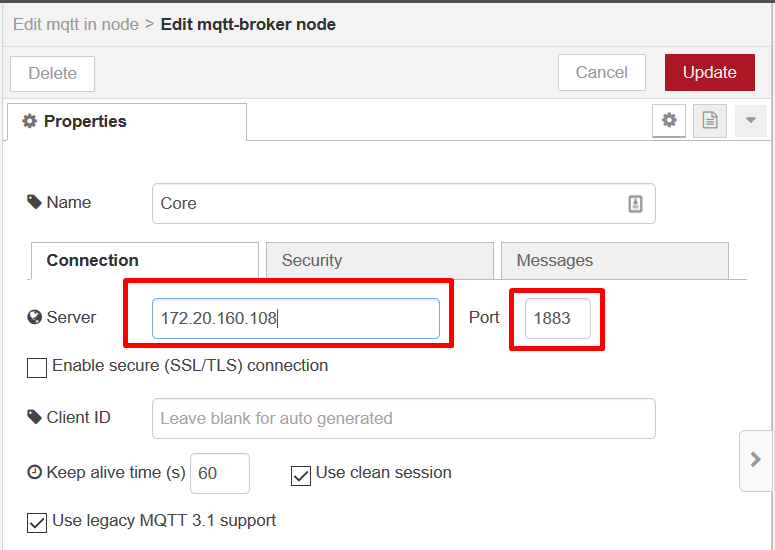
Uno llamado **Temperatura**, uno llamado **Hot** y otro llamado **Cold**.



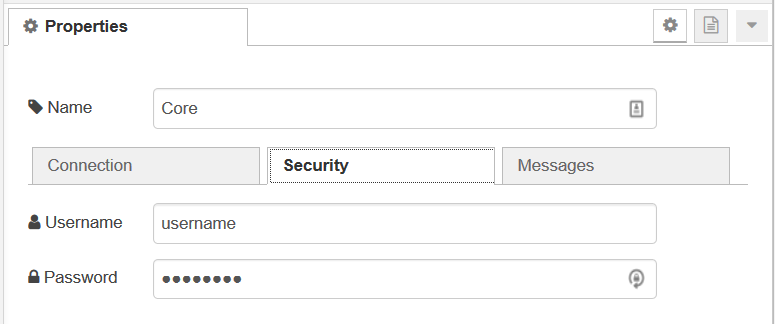
Para configurar el servidor hacemos clic en la imagen del lápiz.



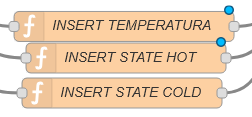
Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

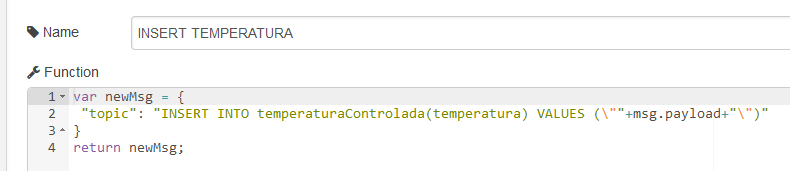


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



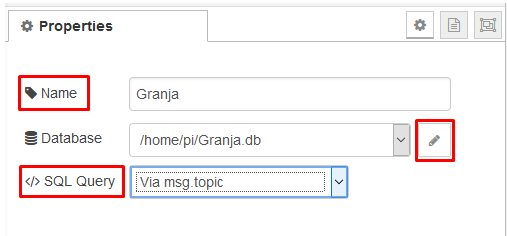
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

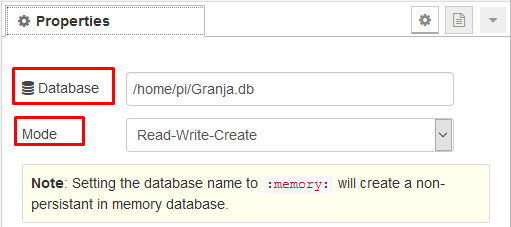
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

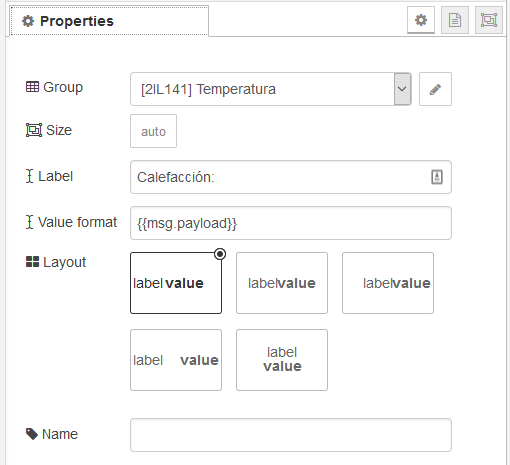
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



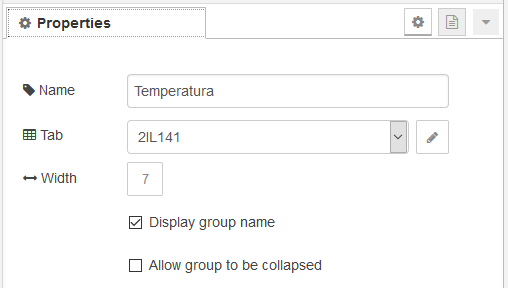
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

Es la etiqueta que tendrá en el dashboard

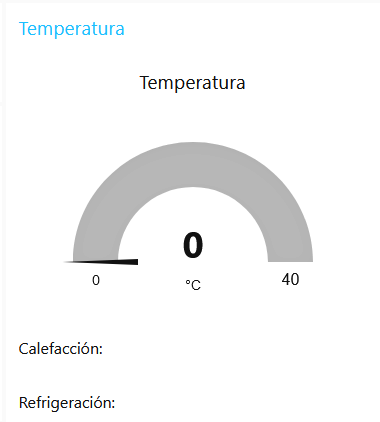
1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

Es la configuración en la que queremos que se muestre el **label** y el **value**

**NOTA:** Todos los nodos de dashboard bajo el mismo nombre se agruparán de la siguiente manera.

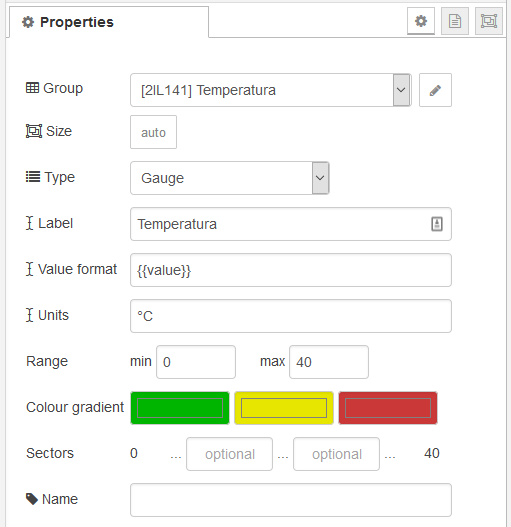


1. **Name**

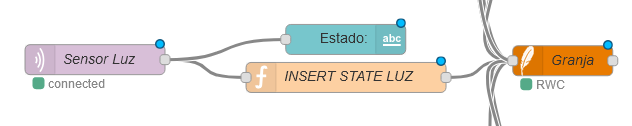
Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él



Para la gráfica Gauge son prácticamente los mismos pasos con respecto al tab y grupo, el resto son configuraciones sobre cómo quieres que se vea la tabla, en nuestro caso, solo ajustamos el rango de 0º Celsius a 40º Celsius y la unidad de medida a ºC.

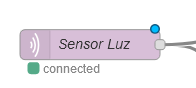


**Sensor de luz**



Nodos utilizados

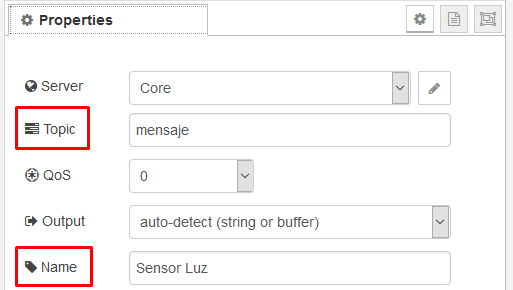
* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



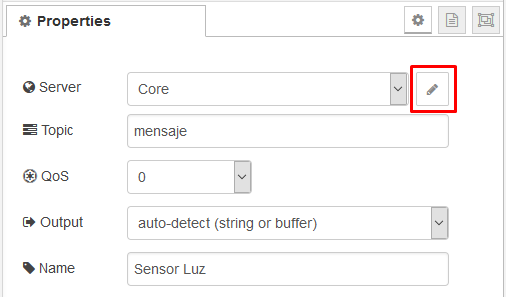
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

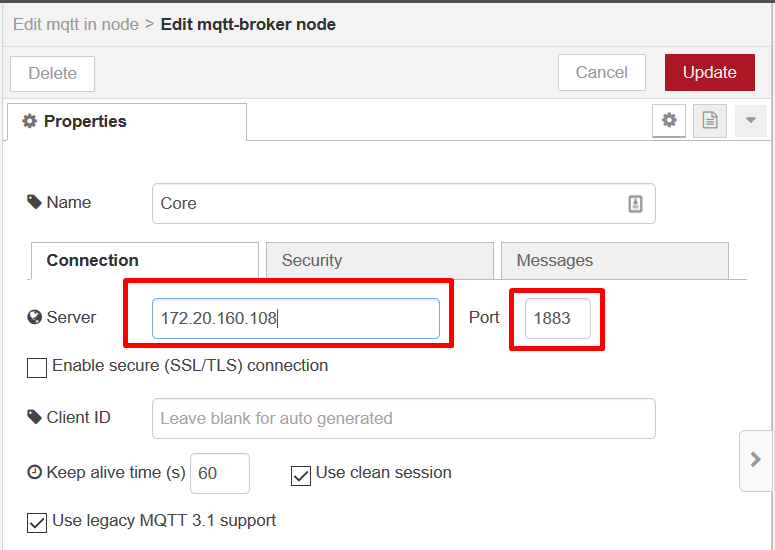
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.



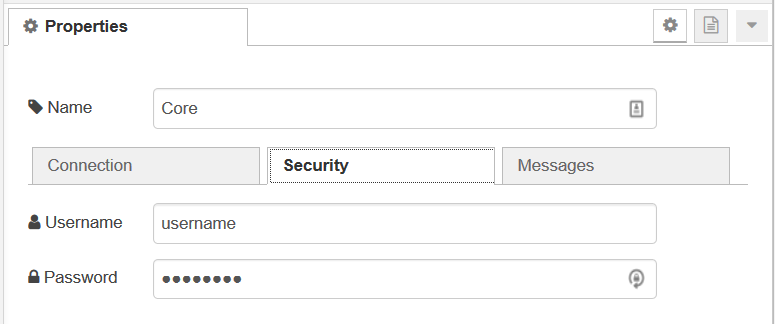
Para configurar el servidor hacemos clic en la imagen del lápiz.



Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

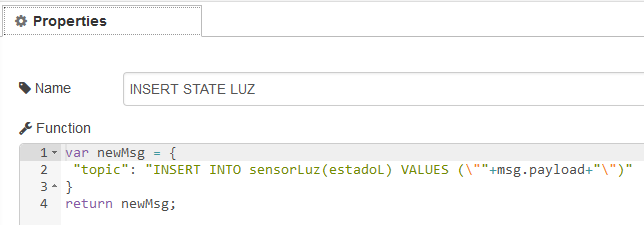


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



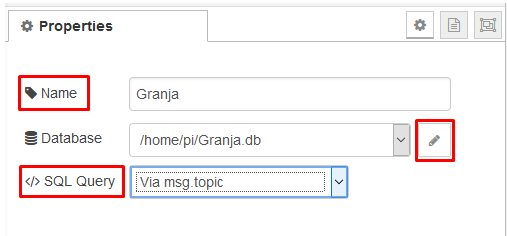
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

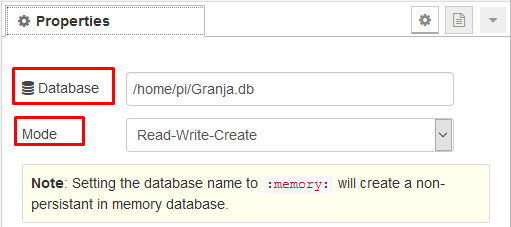
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



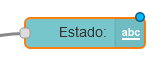
Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

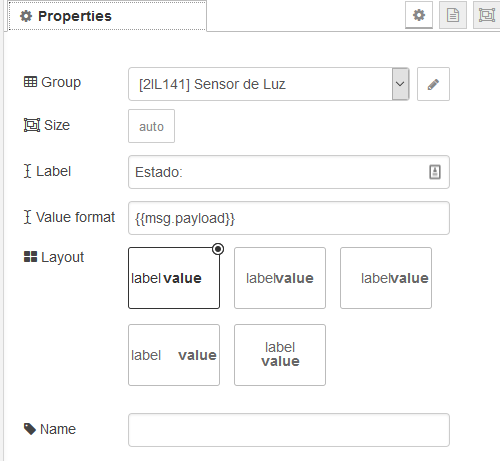
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

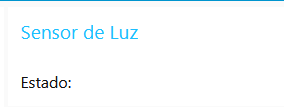
Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

Es la configuración en la que queremos que se muestre el **label** y el **value**

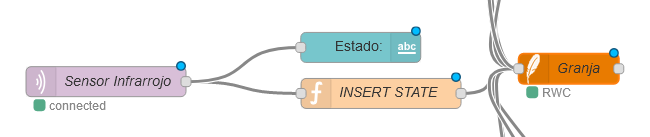


1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él



**Sensor infrarrojo**



Los nodos utilizados son:

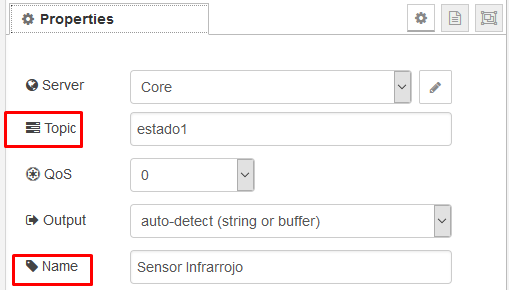
* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



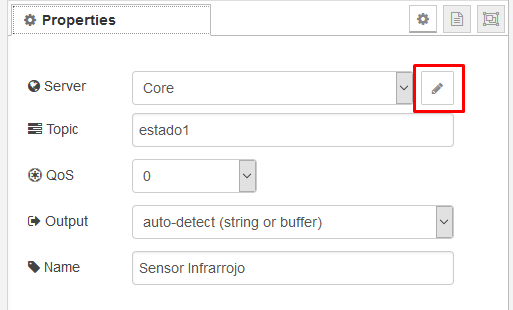
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

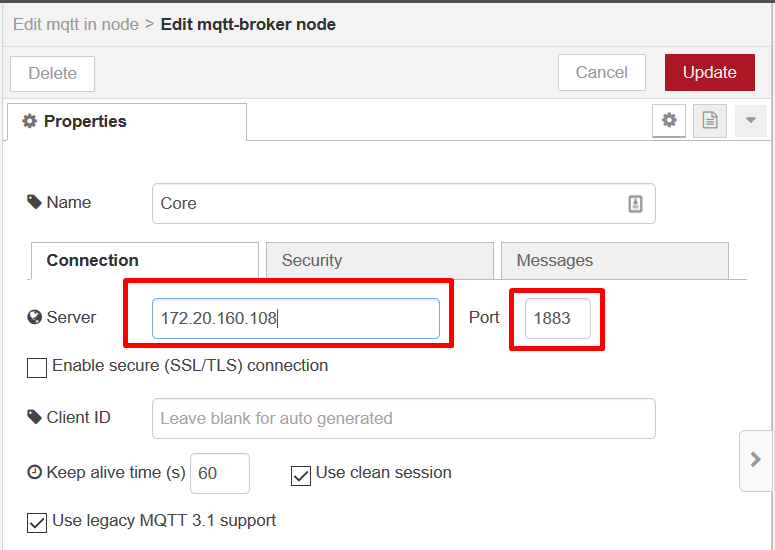
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.



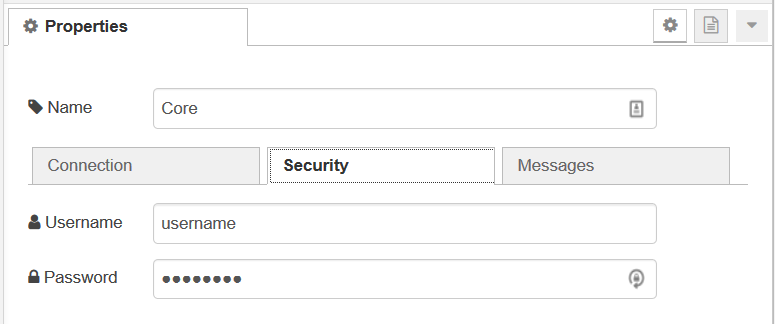
Para configurar el servidor hacemos clic en la imagen del lápiz.



Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

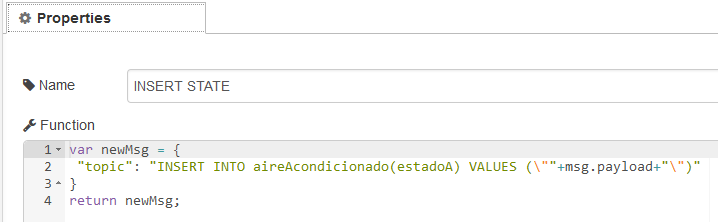


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



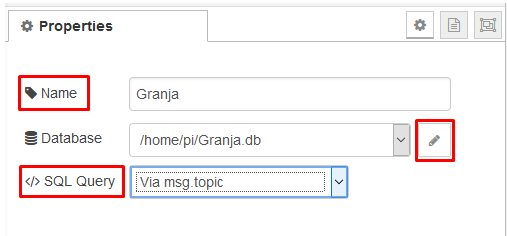
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

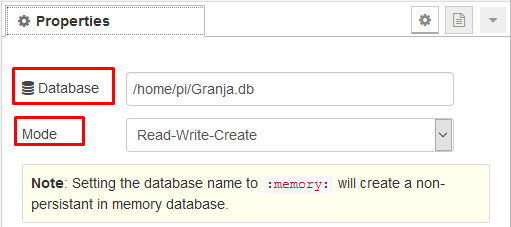
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



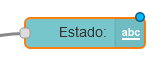
Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

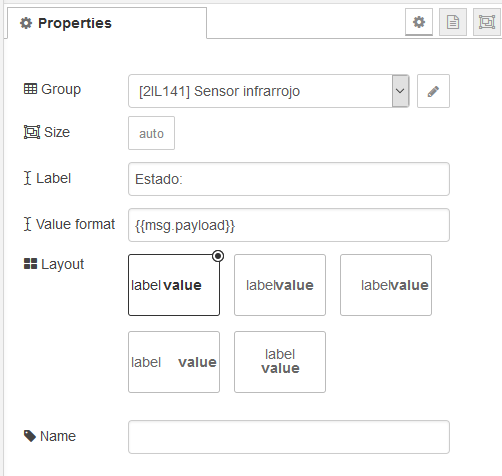
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



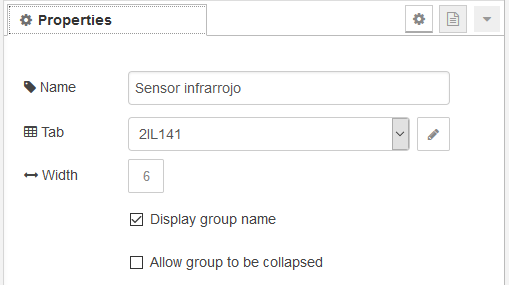
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

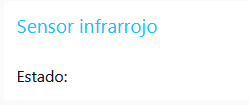
Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

Es la configuración en la que queremos que se muestre el **label** y el **value**

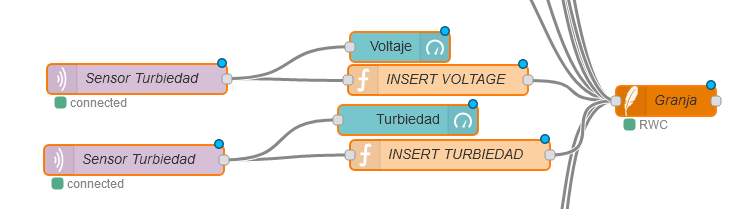


1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él

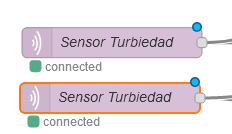


**Sensor de turbiedad**



Los nodos utilizados son:

* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



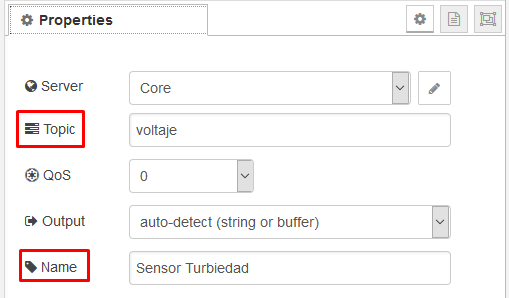
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

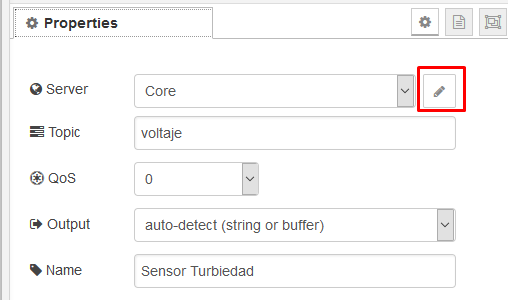
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.

En este caso particular se crean 2, cada uno trabajará bajo un topic diferente.

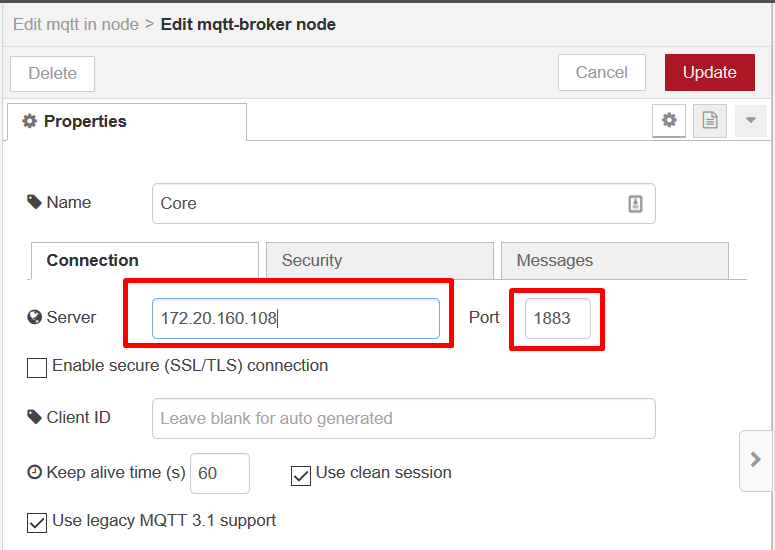
Uno llamado **voltaje**, otro llamado **turbiedad**.



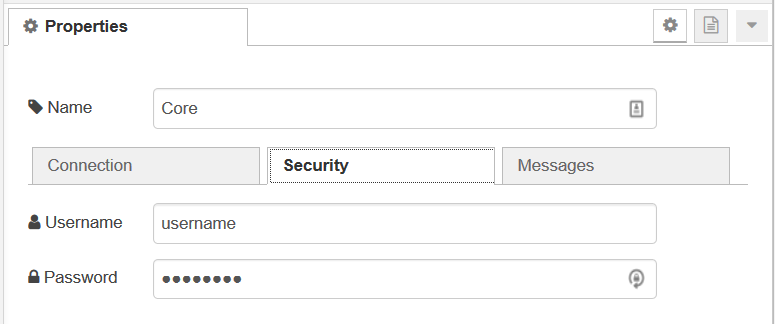
Para configurar el servidor hacemos clic en la imagen del lápiz.



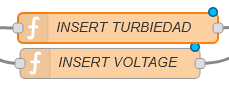
Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

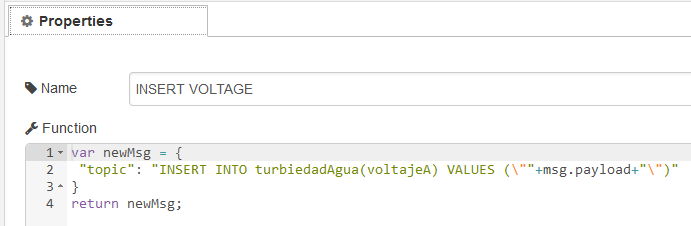


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



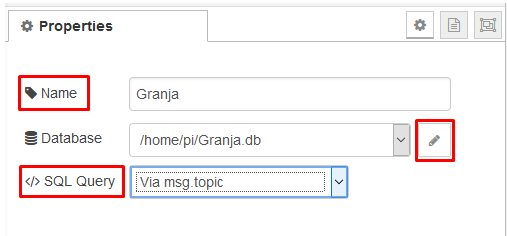
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

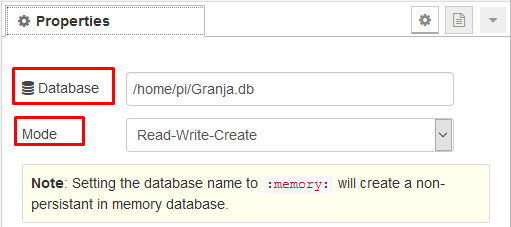
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



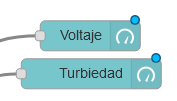
Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

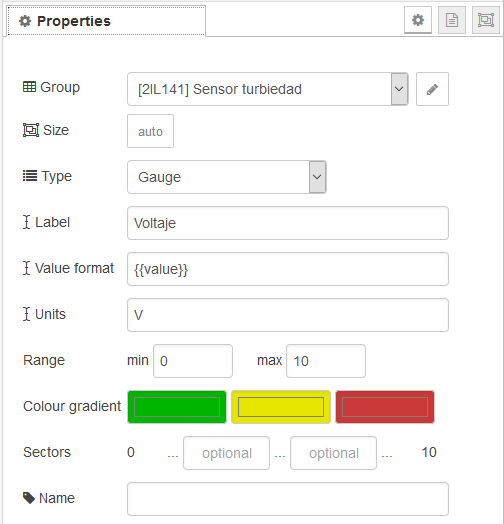
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



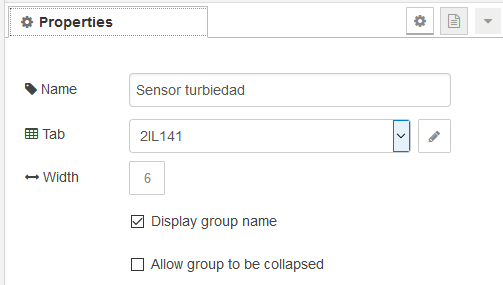
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

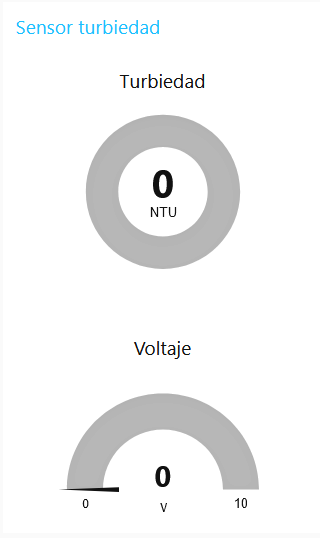
Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Units**

Es la unidad en la que se encuentra el valor.

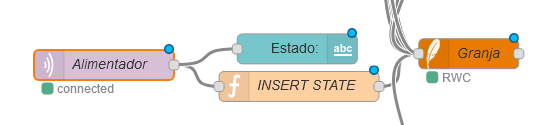


1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él



**Alimentador**



Los nodos utilizados son:

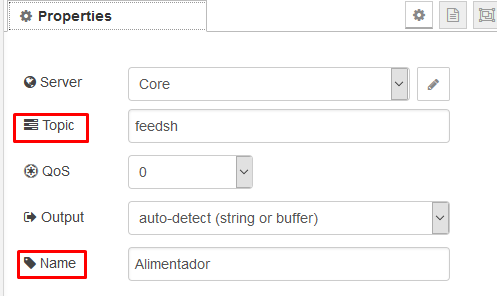
* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



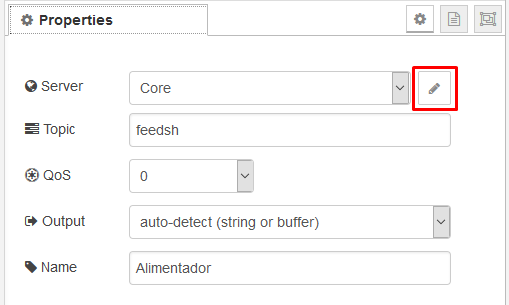
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

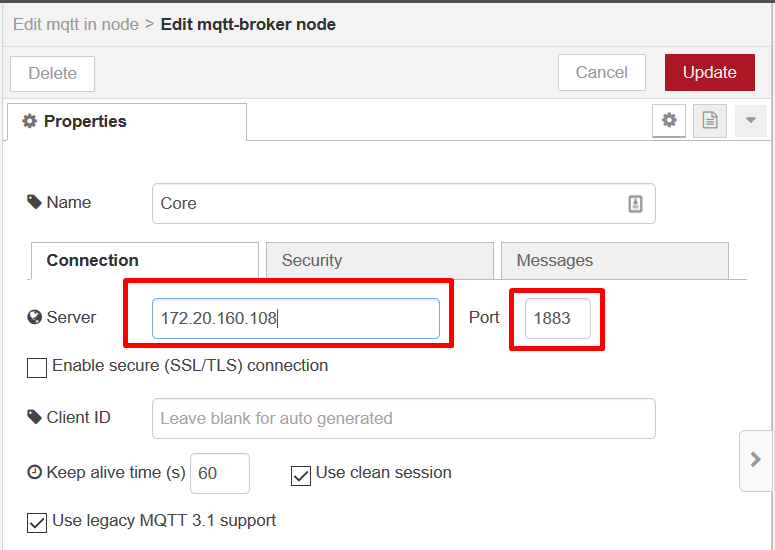
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.



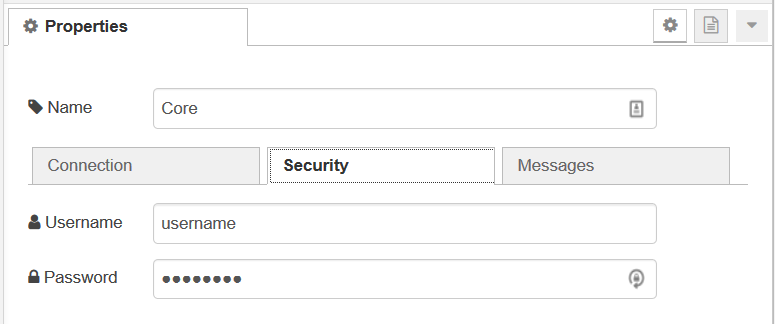
Para configurar el servidor hacemos clic en la imagen del lápiz.



Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

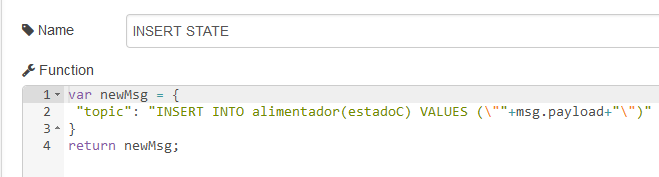


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



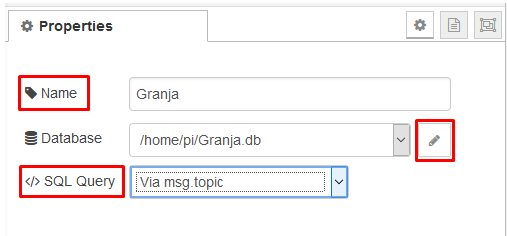
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

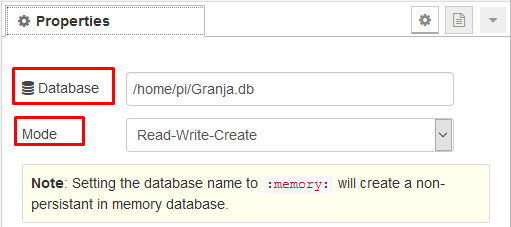
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

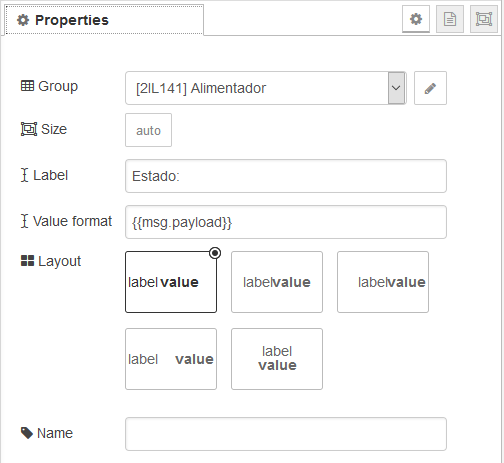
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



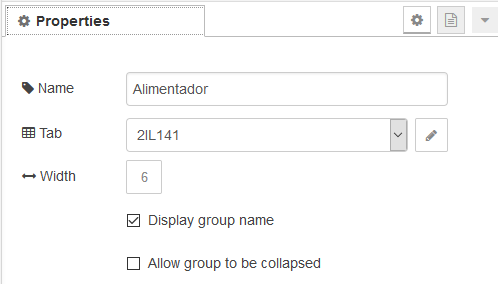
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

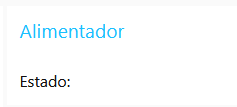
Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Layout**

Es la configuración en la que queremos que se muestre el **label** y el **value**

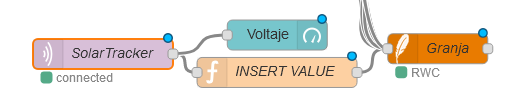


1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él



**Solar Tracker**



Los nodos utilizados son:

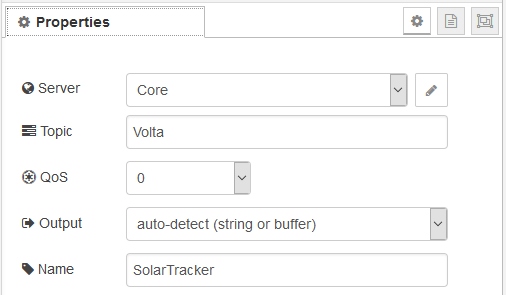
* MQTT-input: Se conecta a un agente MQTT y se suscribe a los mensajes del tema especificado.



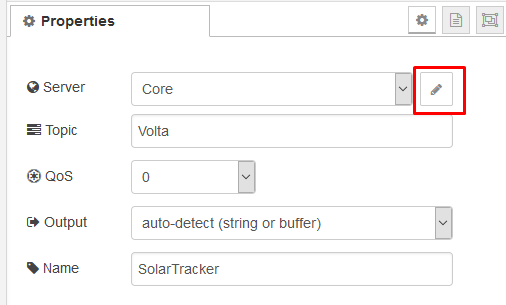
**Configuración del nodo**

En **topic** se coloca el tema bajo el cual el módulo publica los mensajes, debe ser el mismo topic definido en la programación de Arduino.

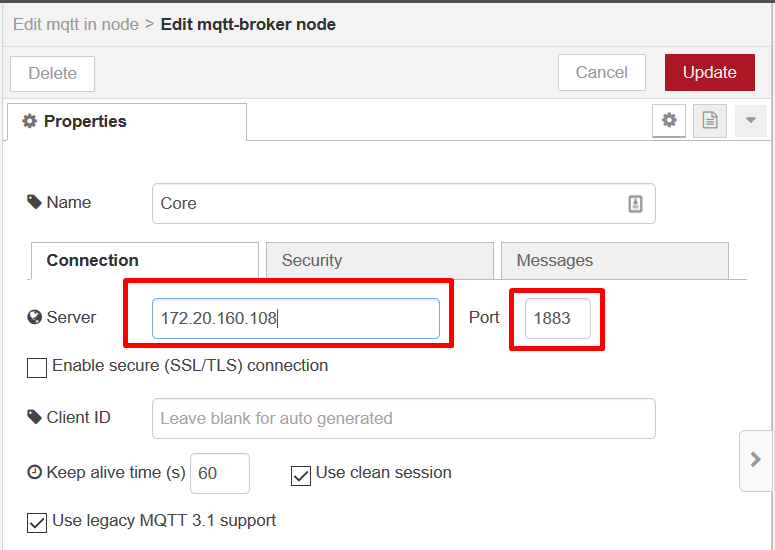
En **name** se coloca el nombre del módulo, este servirá como etiqueta para reconocer a qué modulo pertenece ese nodo.



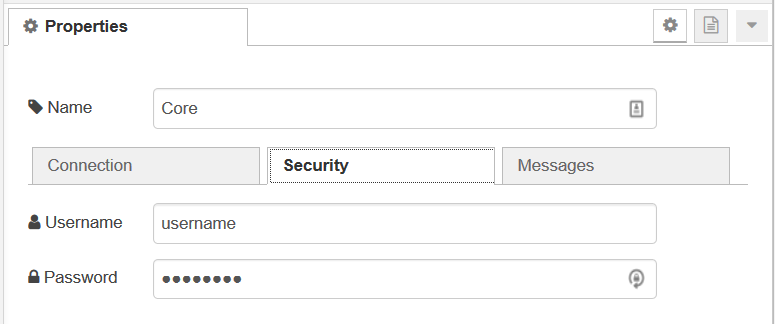
Para configurar el servidor hacemos clic en la imagen del lápiz.



Aquí colocamos la dirección IP del servidor 172.20.160.108 y el puerto 1883 ya que usa el Protocolo de Control de Transmisión (TCP). Es uno de los protocolos principales en redes TCP/IP. TCP es un protocolo orientado en la conexión



En la pestaña de seguridad se coloca el usuario y contraseña del MQTT Broker. En este caso en particular el usuario es **username** y contraseña **12345678**



* Function node: Un nodo de funciones de JavaScript para ejecutarse ante los mensajes que recibe el nodo. Se espera que la función devuelva un objeto de mensaje.

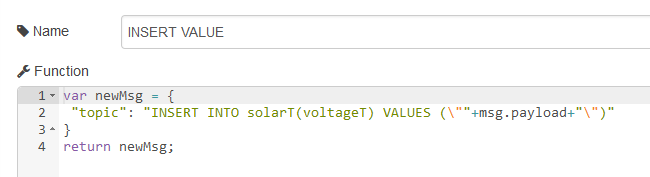


**Configuración del nodo**

En **name** se coloca el nombre de la función, servirá como una etiqueta para reconocer qué hace la función.

En **function** se coloca el SQL query que se enviará más adelante al nodo de base de datos.

Utilizando el formato CREATE TABLE nombredelatabla(nombredelacolumna) VALUES (msg.payload) donde msg.payload es el mensaje que recibe del MQTT Broker



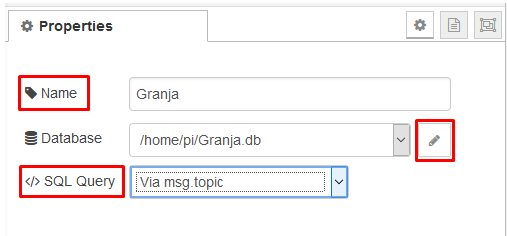
* SQLite node: Permite el acceso a una base de datos de SQLite.



**Configuración del nodo**

En **name** coloca un nombre que servirá como una etiqueta para saber con cual base de datos se está trabajando, en caso de tener más de una base de datos.

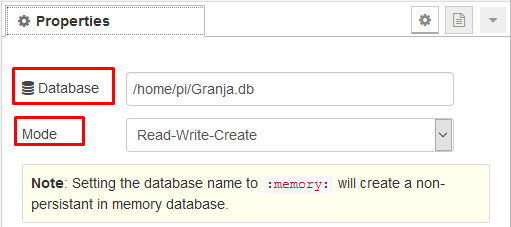
En **SQL query** se coloca la vía por la que el nodo de base de datos recibe el query, en este caso lo recibe vía msg.topic dado por el nodo de función explicado anteriormente.



Dando clic en el lápiz podemos entrar a los ajustes de la base de datos.

En **Database** se coloca la ruta donde se encuentra la base de datos dentro de nuestra Raspberry Pi, en este caso es el directorio pi dentro del directorio home.

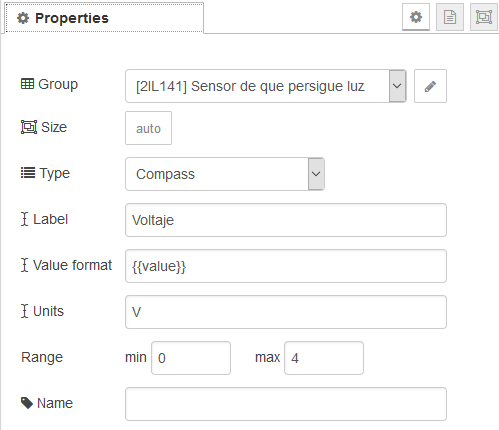
El **Mode** nos permite especificar los permisos que tiene node red sobre esta base de datos, puede ser Read-Write-Create, Read-Write o Read only.



Dashboard node:



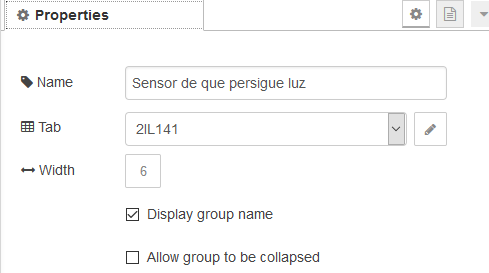
**Configuración del nodo**



1. **Group**

Se deberá crear un **Tab** y nombrarlo en este caso lo nombraremos 2IL141. Se puede tener más de un dashboard con diferentes nombres, en nuestro caso, todos irán bajo el tab 2IL141.

El **nombre** **del grupo** será el nombre del sensor al que pertenece.



1. **Size**

Permite ajustar el tamaño que ocupará en el dashboard

1. **Label**

Es la etiqueta que tendrá en el dashboard

1. **Value**

Es el mensaje que mostrará en el dashboard, en este caso es el mensaje que recibe el MQTT Broker, es decir, msg.payload.

1. **Units**

Es la unidad en la que se encuentra el valor.



1. **Name**

Servirá como una etiqueta para saber qué es el nodo sin tener que entrar en él

