

**NODO EDGE**



**Universidad  
del Cauca®**

**VALERIA PINO RIVERA  
JUAN DIEGO ROSERO BENAVIDES  
JUAN CAMILO LOPEZ MORILLO**

**FULVIO YESID VIVAS CANTERO  
DOCENTE**

**UNIVERSIDAD DEL CAUCA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
SISTEMAS EMBEBIDOS Y TIEMPO REAL  
POPAYÁN  
SEPTIEMBRE DEL 2024**

## TABLA DE CONTENIDO

Diagrama en bloques del nodo Edge.....	3
Diseño base de datos. ....	3
Escenario de pruebas.....	4
Modelo de comunicación .....	7
Circuito físico final .....	8
Problemas encontrados y sus soluciones .....	9
Evidencia fotográfica de los datos en dashboard.....	11
Configuración Interfaz de Grafana .....	15

## Diagrama en bloques del nodo Edge.

Este sistema está diseñado para recopilar y gestionar información sobre condiciones ambientales (temperatura y humedad) y el estado de un dispositivo LED RGB, facilitando la interacción entre los nodos mediante la red MESH y la sincronización con una base de datos en la nube.

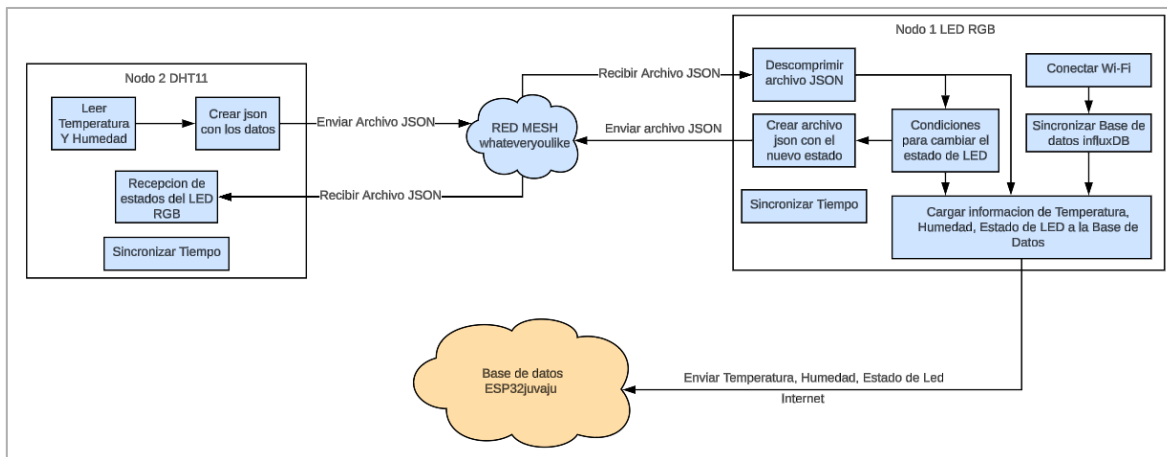


Diagrama 1. Diagrama del funcionamiento

## Diseño base de datos.

El diseño de la base de datos consistió en seguir los pasos sugeridos por el docente, desde classroom, y a su vez muy bien explicados en el siguiente enlace <https://randomnerdtutorials.com/esp32-influxdb/>.

Para la creación de esta base de datos se utilizó INFLUXDB, plataforma que permite recibir y almacenar los datos transmitidos desde el ESP32. En esta ocasión:

```
#define INFLUXDB_URL "https://us-east-1-1.aws.cloud2.influxdata.com"
#define INFLUXDB_TOKEN
"lFwWiXtp445KCHUrv4nofpaSE3dnGpTKxtfWJPn803tNLcGcADrgw0NVPXiwiKtFcIn5Ec9P7V3NE58s9g
BOnw=="
#define INFLUXDB_ORG "16446c2508ada63c"
#define INFLUXDB_BUCKET "ESPfinal"
```

Como datos importantes tenemos, el URL HTTP, el token y el nombre del bucket en el que se estará recopilando la información.

Posterior a esto se enciende la base de datos (solo con abrir la página es suficiente) y se espera a que el ESP32 cargue la información a la misma, para que se generen las gráficas.

El propósito de la base de datos es almacenar los datos extraídos a partir del monitoreo de temperatura, humedad y estado del Led, proporcionados por el nodo 2 de la red Mesh. Los datos relevantes recopilados de esta base de datos son: Temperatura, Humedad, Estado de Led y tiempo. La relación que podemos observar en este caso es con respecto a los valores de la temperatura y estado de led.

## Escenario de pruebas.

- **DHT11.** Se realiza una prueba de funcionamiento del DHT11, consiste en ejecutar un código que solo involucra el ESP32 y el sensor en cuestión, se cuenta con la entrada de datos proporcionados y como salidas se tiene la temperatura en grados Celsius [°C], y el porcentaje de humedad en el ambiente [%]. EN la figura 1 se presenta la evidencia de la prueba.

Datasheet: <https://components101.com/sensors/dht11-temperature-sensor>

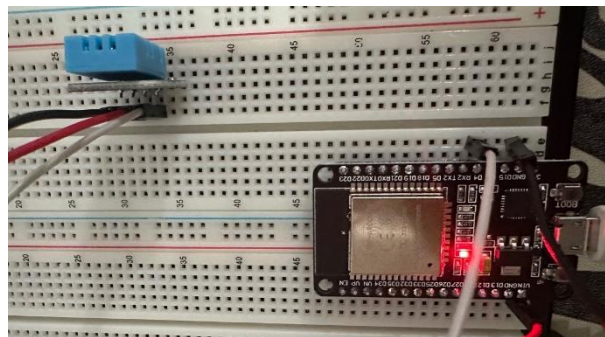


Figura 1. Funcionamiento DHT11

- **Verificación LED RGB.** Prueba que se realiza con el fin de asegurar el correcto funcionamiento del led RGB, además se comprobó la continuidad de corriente con cada uno de los cátodos del led (Figura 2).

Datasheet: <https://www.electronicoscaldas.com/datasheet/LED5D-RGB-CA.pdf>

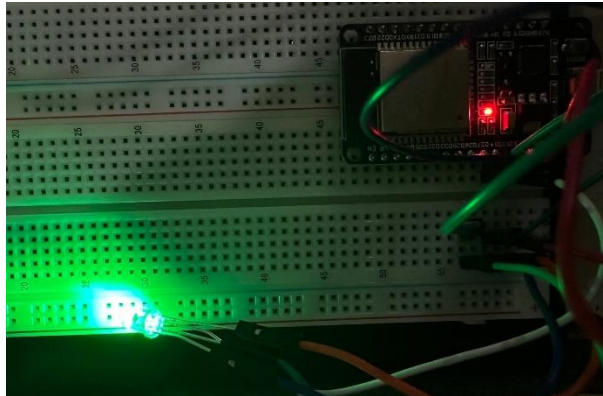


Figura 2. Funcionamiento LED RGB

- **Conexión.** Como continuidad de las pruebas individuales, se prueba que las conexiones e interacción entre ambos sistemas funcionen correctamente, aquí es donde se inician las pruebas de intercambio de información entre ambos ESP's (Figura 3).

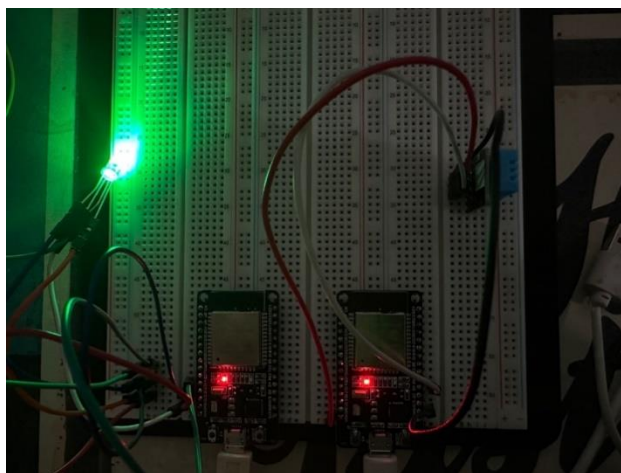


Figura 3. Interacción entre elementos

- **Tiempos NTP**, prueba para obtener el sincronismo horario mediante una función que brinda Arduino, en la figura 4 la demostración a la salida de ambos ESP's.

```
23:14:05.809 -> Enviando estado del LED RGB al Nodo 1:
23:14:05.809 -> {"node":2,"ledRed":0,"ledGreen":255,"ledBlue":0}
23:14:07.874 -> Hora ajustada 932850201. Offset = -7018
23:14:07.908 -> Hora ajustada 932897268. Offset = 7838
23:14:08.090 -> Hora ajustada 933107573. Offset = -5964
23:14:08.305 -> Hora ajustada 933325331. Offset = 3016
23:14:10.920 -> Mensaje recibido desde 3380935133: {"node":2,"temp":29.299999237060547,"hum":33}
23:14:10.920 -> Nodo: 2
```

Figura 4. Interacción entre elementos

- **Cuando varíe la temperatura, cambia el color del led.** Prueba de funcionamiento total de sistema, en el que dependiendo de las entradas de la temperatura registradas desde el nodo 1, se cambia el estado de iluminación (color) de led, y se envía el estado del led de vuelta del nodo 2 al nodo 1.

Se realiza la prueba de la conexión y envió de información entre ambos ESP's, esto se hace siguiendo las instrucciones del docente, creando un código que registre la temperatura y en base a estos datos enviados de un ESP a otro se cambie el color del LED bajo las siguientes condiciones:

Rojo	( $t > 30$ )
Verde	( $20 < t < 30$ )
Azul	( $t < 20$ )

Las pruebas se evidencian en las figuras 5 y 6.

```
23:09:05.767 -> Temperatura: 30.20 °C
23:09:05.767 -> Humedad: 33.00 %
23:09:05.767 -> Enviando estado del LED RGB al Nodo 1:
23:09:05.767 -> {"node":2,"ledRed":255,"ledGreen":0,"ledBlue":0}
```

Figura 5. Respuesta en el primera ESP32

```
23:09:10.885 -> Temperatura: 29.90 °C
23:09:10.885 -> Humedad: 34.00 %
23:09:10.885 -> Enviando estado del LED RGB al Nodo 1:
23:09:10.885 -> {"node":2,"ledRed":0,"ledGreen":255,"ledBlue":0}
```

Figura 6. Respuesta en el segundo ESP32

- **Visualización en base de datos.**

Prueba para conectar el ESP32 a la nube, donde se observan sus datos registrados por el ESP32 enviados a la base de datos de InfluxDB.

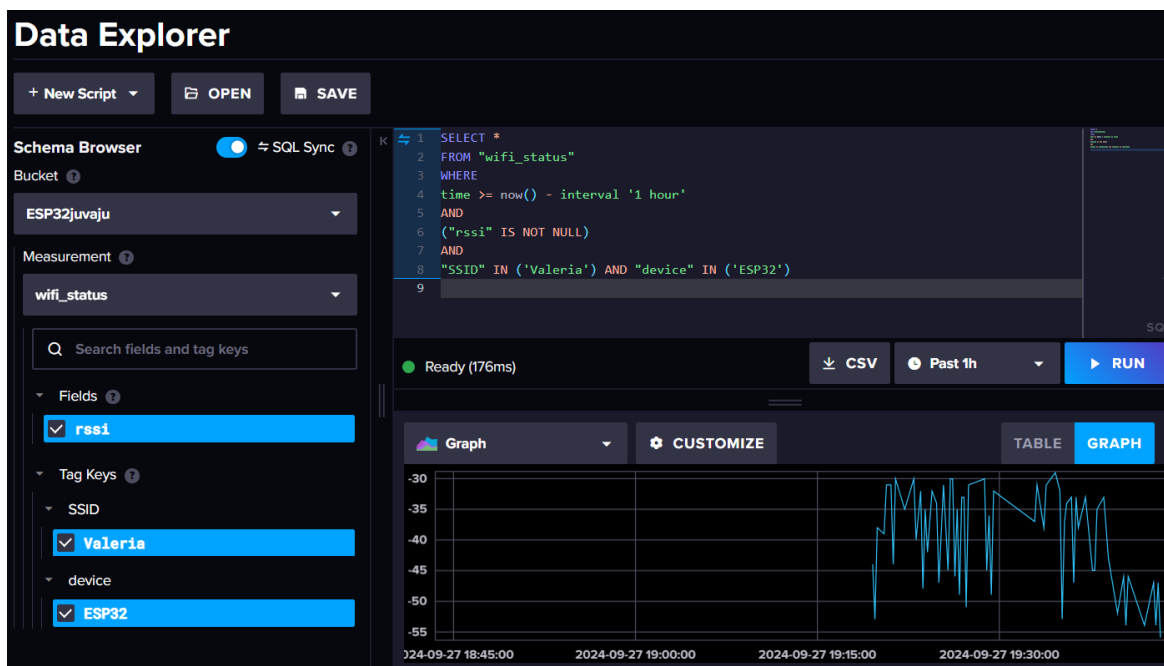


Figura 7. Data Explorer

## Modelo de comunicación

En el diagrama 2 se muestra la comunicación entre dos nodos en una red IoT. El Nodo 2 (DHT11) se encarga de leer datos de temperatura y humedad, que son enviados a través de Red MESH al Nodo 1 (LED RGB). Este nodo procesa los datos y actualiza el estado de un LED RGB, además de establecer una conexión Wi-Fi para sincronizarse con una Base de Datos llamada ESP32final (InfluxDB), donde se almacenan los datos recopilados. El flujo de información se mantiene bidireccional entre los nodos, mientras el Nodo 1 envía datos a la base de datos a través de la conexión Wi-Fi.

El modelo de comunicación utilizado fue Exclusive Pair Model, ya que describe una comunicación directa y bidireccional entre dos nodos.

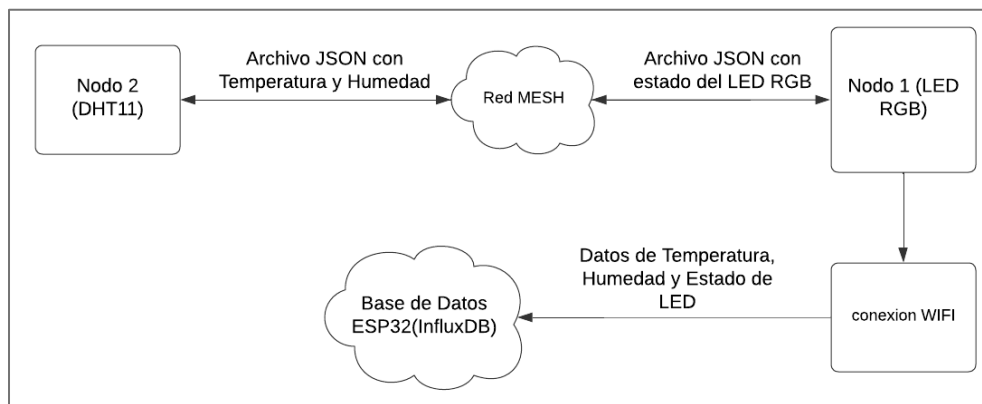


Diagrama 2. Diagrama de la comunicación

### Circuito físico final

En esta sección podemos observar cómo se ve el montaje físico de nuestro circuito, el cual este compuesto por los ESP32, el DHT11 y el led RGB.

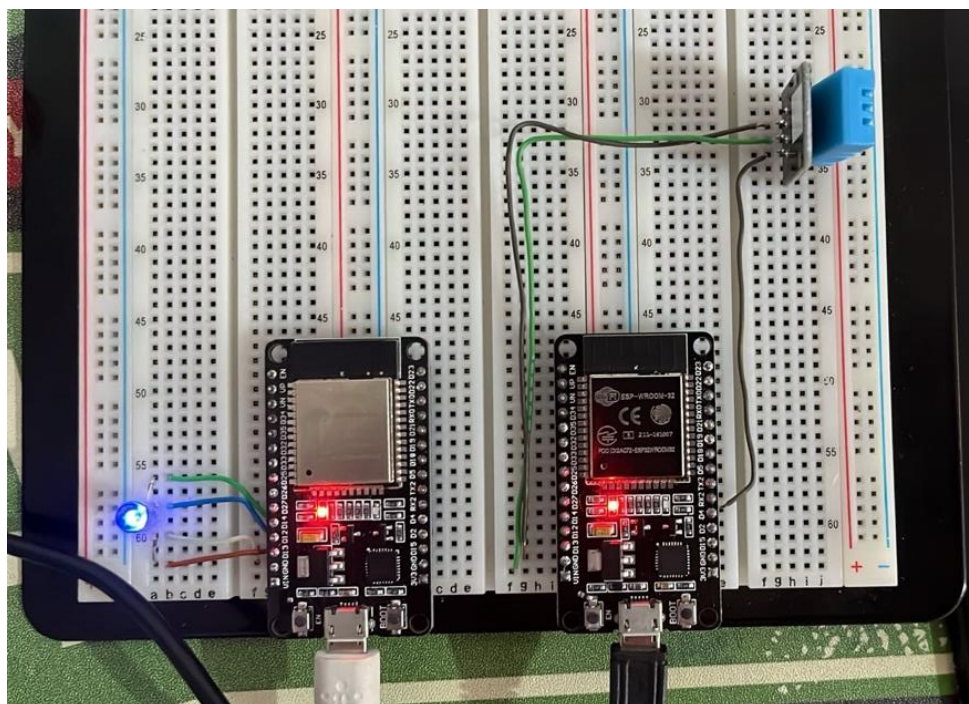


Figura 8. Circuito Físico



## Problemas encontrados y sus soluciones

Los problemas encontrados para el desarrollo de este proyecto están localizados, en su gran mayoría, en la conexión y compatibilidad de lo físico y desarrollado anteriormente con la implementación de la base de datos. En esta sección se mencionan los problemas más importantes y las soluciones que implementamos.

- **Cantidad de recursos en InfluxDB**

Como programa para la base de datos se utilizó InfluxDB, el cual tiene recursos limitados en cuanto a almacenamiento, en demasiadas ocasiones nos encontramos con que teníamos que rehacer la DB, ya sea por compatibilidad o funcionamiento. El problema consiste en que en InfluxDB no se pueden crear Buckets infinitos.

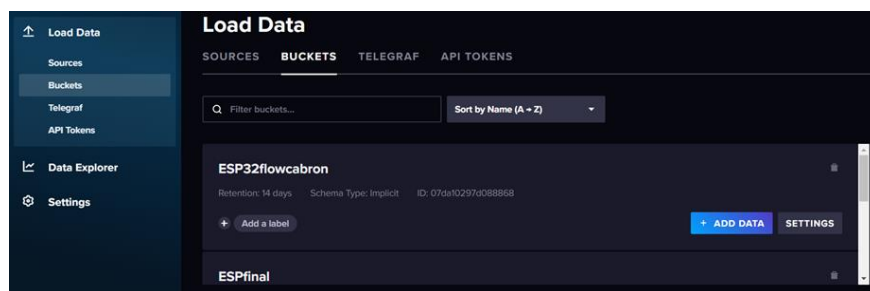


Figura 9. Panel Buckets en InfluxDB

Para solucionar este problema se llegó a la conclusión de que por cuenta no pueden existir más de 2 Buckets activos y por lo tanto si se desea continuar y crear uno nuevo se debe borrar los anteriores, o al menos asegurarse que siempre se debe tener un espacio disponible para un Bucket nuevo.

- **SQL Sync**

Al momento de graficar dentro de InfluxDB se puede generar un error el cual impide observar los datos respecto al tiempo, explícitamente sale un cuadro o panel de color rojo que impide continuar a observar las gráficas, esto significa que la manera en cómo se planteó el código para graficar es errado y no se pueden obtener los datos de las tablas de muestreo.

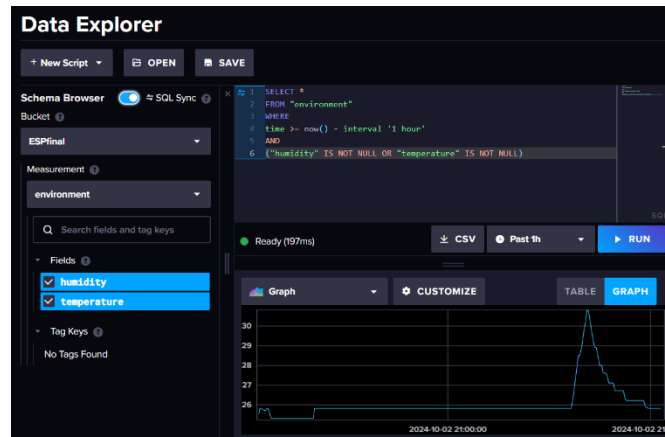


Figura 10. Problema del Data Explorer

Como solución a la problemática lo que se debe realizar es desactivar por momento la sincronización con la base de datos DBSync y asegurarse que la manera y forma como se hace la consulta de datos es la adecuada.

- **Query Language**

Como plataforma para la visualización de los datos se escogió a Grafana, aplicación que permite tener una basta cantidad de paneles que puede mostrar la información de distintas maneras, ahora el problema se encuentra al momento de la creación del dashboard y la manera en cómo se conecta con InfluxDB, el problema consiste en que dependiendo de los ajustes respecto al Query se podrá garantizar la conexión, compatibilidad y mantenimiento de la base de datos.

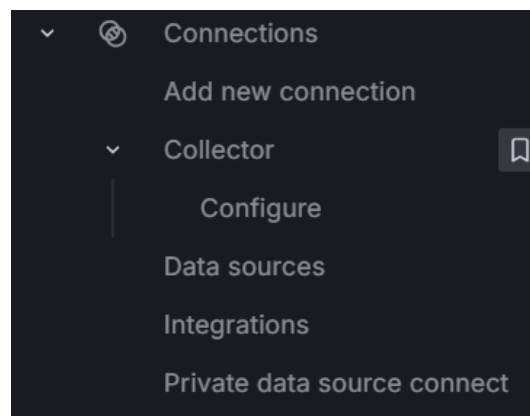


Figura 11. Ubicación Data Sources

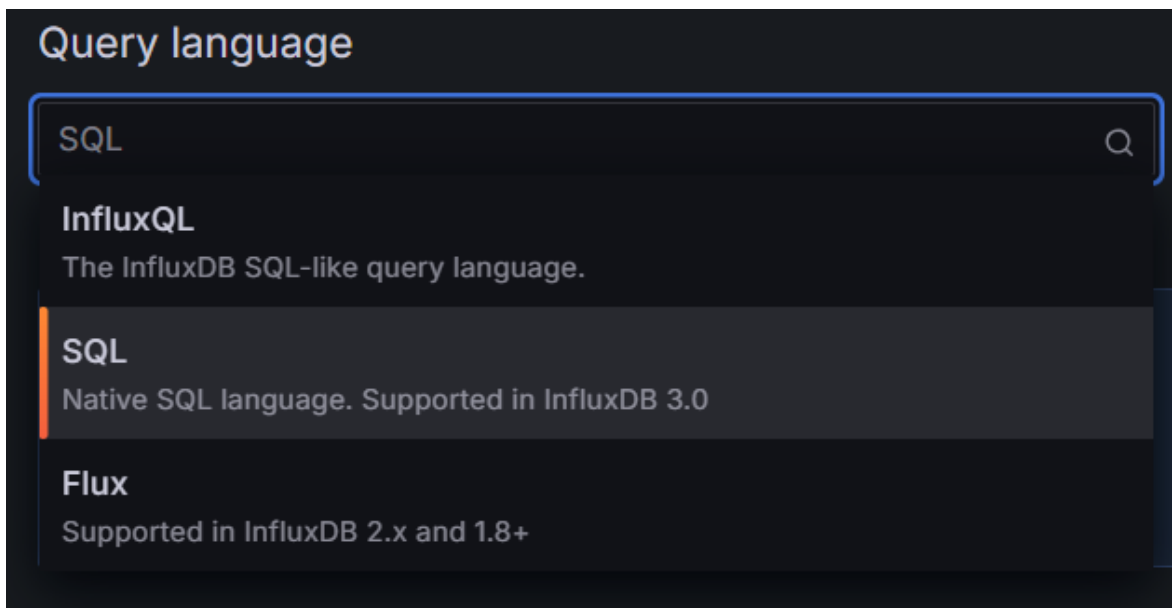


Figura 12. Selección correcta Query Language

La solución es asegurarse que el Query Language dentro de las configuraciones de Grafana se encuentre ubicado en SQL para esta ocasión, ya que esta es la que soporta la información subida desde InfluxDB 3.0 (Cloud).

### Evidencia fotográfica de los datos en dashboard

Se evidencian muestras del funcionamiento, observando cómo los datos de temperatura y humedad, enviados desde el Nodo 2, son visualizados y gestionados en tiempo real en el dashboard de InfluxDB. Se visualizan los cambios en los estados del LED RGB controlado por el Nodo 1. Además, se mostrará cómo el dashboard permite realizar un seguimiento detallado de las variaciones de temperatura y humedad a lo largo del tiempo, validando la funcionalidad del sistema IoT implementado.

```
Humidity: 57.00% Temperature: 25.80°C
{"node":2,"temp":25.799999237060547,"hum":58}
Humidity: 57.00% Temperature: 25.80°C
{"node":2,"temp":25.799999237060547,"hum":57}
Humidity: 57.00% Temperature: 25.80°C
{"node":2,"temp":25.799999237060547,"hum":57}
Humidity: 57.00% Temperature: 25.80°C
{"node":2,"temp":25.799999237060547,"hum":57}
Humidity: 57.00% Temperature: 26.20°C
{"node":2,"temp":26.200000762939453,"hum":57}
Humidity: 58.00% Temperature: 26.20°C
{"node":2,"temp":26.200000762939453,"hum":58}
```

Figura 13. Información de temperatura y humedad

```

1 SELECT *
2 FROM "environment"
3 WHERE
4   time >= now() - interval '5 minutes'
5 AND
6   ("humidity" IS NOT NULL OR "temperature" IS NOT NULL)
7

```

Figura 14. Código para visualizar graficas

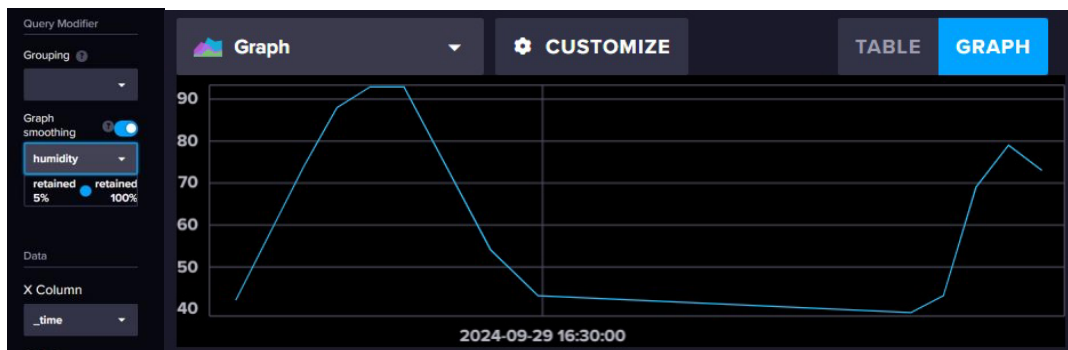


Figura 15. Gráfico humedad

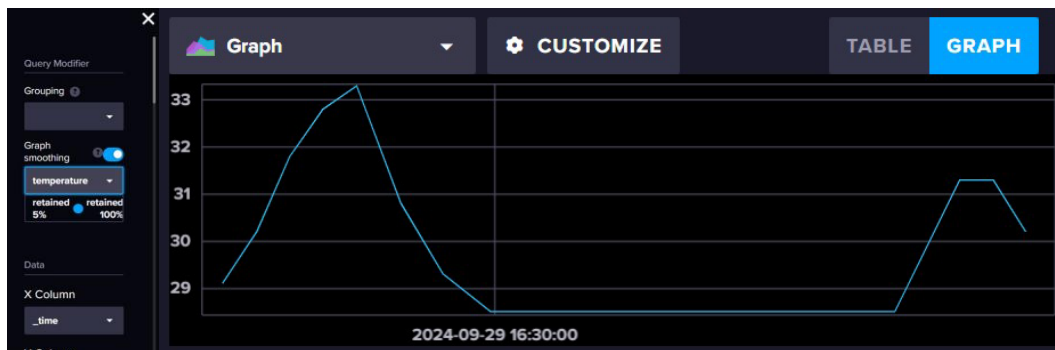


Figura 16. Gráfico temperatura

Grafana es una herramienta de visualización de datos que permite crear dashboard interactivos para monitorear en tiempo real los datos del proyecto. En este caso, se conecta a InfluxDB para extraer y mostrar las mediciones de temperatura, humedad y el estado del LED RGB, capturados por los nodos ESP32 de la red MESH. Grafana permite visualizar estas métricas de manera clara mediante gráficos, facilitando el análisis del sistema IoT implementado y permitiendo identificar patrones o anomalías en los datos de forma intuitiva.

A continuación, se muestra el proceso de conexión de Grafana con la base de datos de InfluxDB:

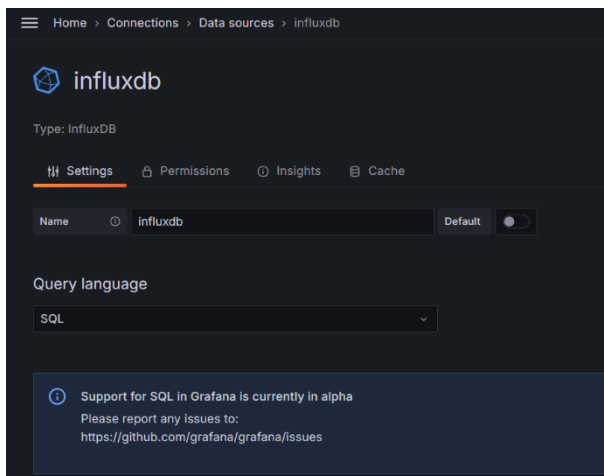


Figura 17. Configuración Grafana 1

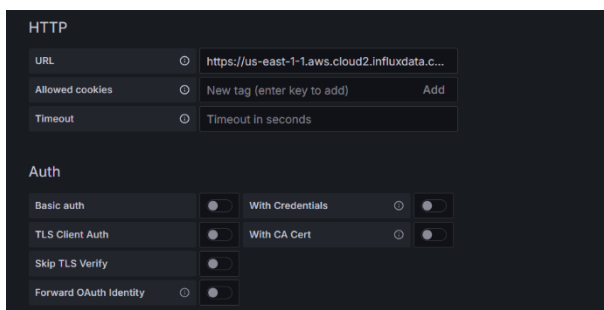


Figura 18. Configuración Grafana 2

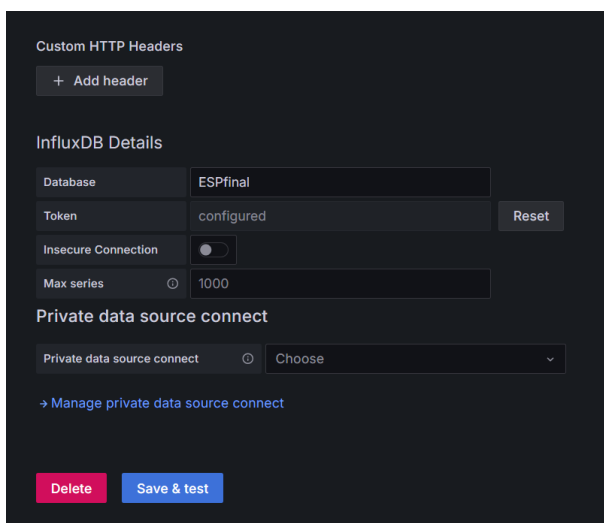


Figura 19. Configuración Grafana 3

El proceso de conexión consiste en asignar los valores del bucket de InfluxDB ya antes mencionados, se sincroniza con la base de datos, con el token, el HTTP y dependiendo de la manera en cómo se haya conectado, la información personal de la cuenta de usuario.

La interfaz final de Grafana queda de la siguiente manera, en donde podemos observar de distintas formas la información capturada por los sensores.

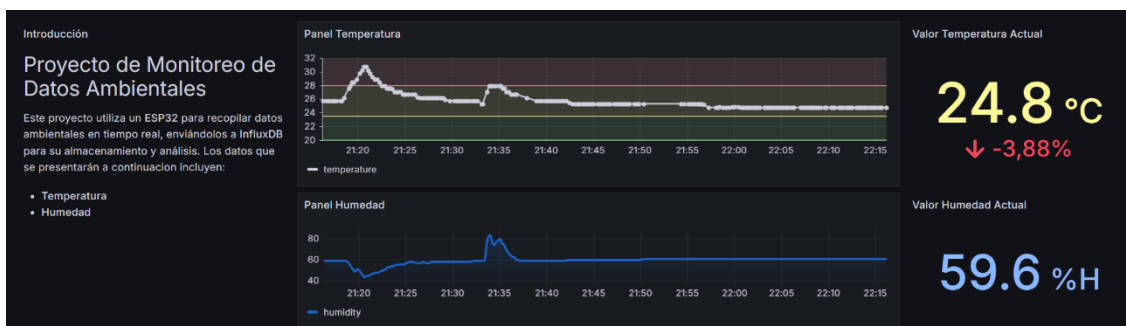


Figura 20. Interfaz final Grafana 1

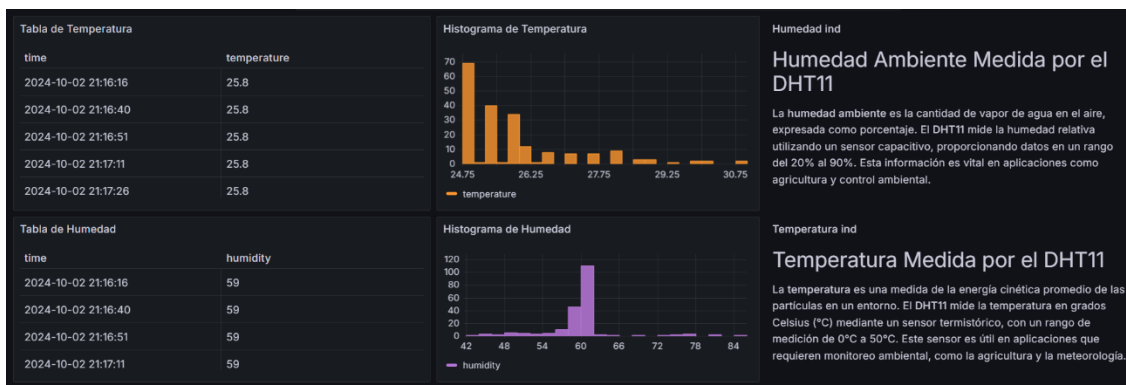


Figura 21. Interfaz final Grafana 2

## Configuración Interfaz de Grafana

La Interfaz en Grafana consiste en distintos módulos que fueron configurados específicamente bajo el contexto del proyecto, a continuación, se explica de manera general como se configuro cada panel de visualización:

- **Panel Temperatura**

Este panel consiste en graficar la temperatura obtenida respecto a cada instante de tiempo, como datos utilizados se encuentran el tiempo y la temperatura, adicionalmente se configuró los umbrales (colores verde, amarillo y rojo) con el fin de dar una guía grafica de cómo se debería comportar la temperatura. También se configuro el rango del eje Y desde (20° a 32°).

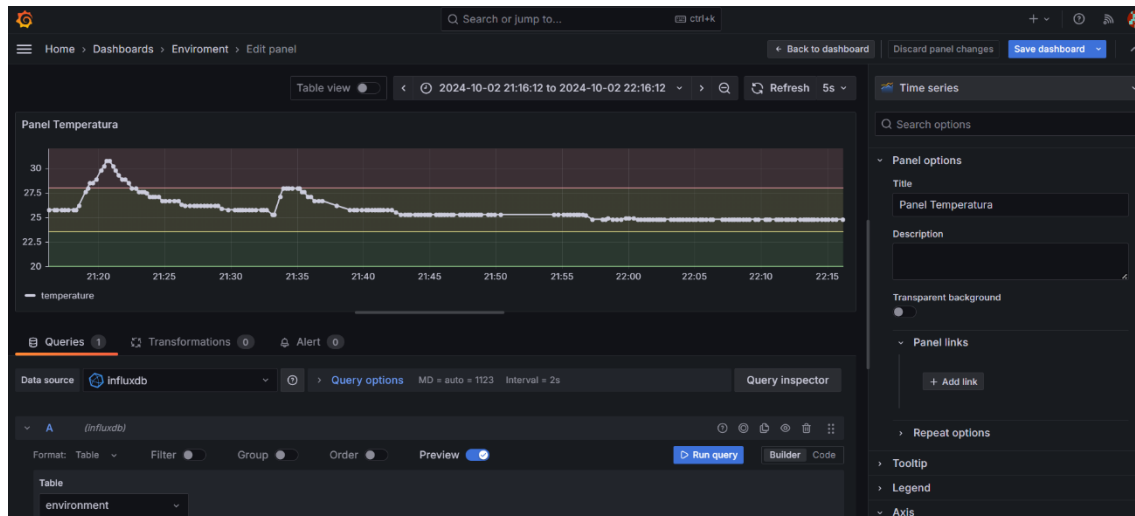


Figura 22. Configuración panel temperatura

- **Panel Humedad**

En este panel se observa la humedad ambiente medida desde el DHT11 respecto al tiempo, como configuraciones especiales es importante mencionar el relleno bajo la curva, y la configuración del rango del eje Y respecto a cuanto ha variado esa temperatura en el rango de tiempo medido.

Es importante resaltar que Grafana para este proyecto está configurada de tal manera que se actualicen los datos y cambios de visualización de paneles cada 10 segundos de manera automática.

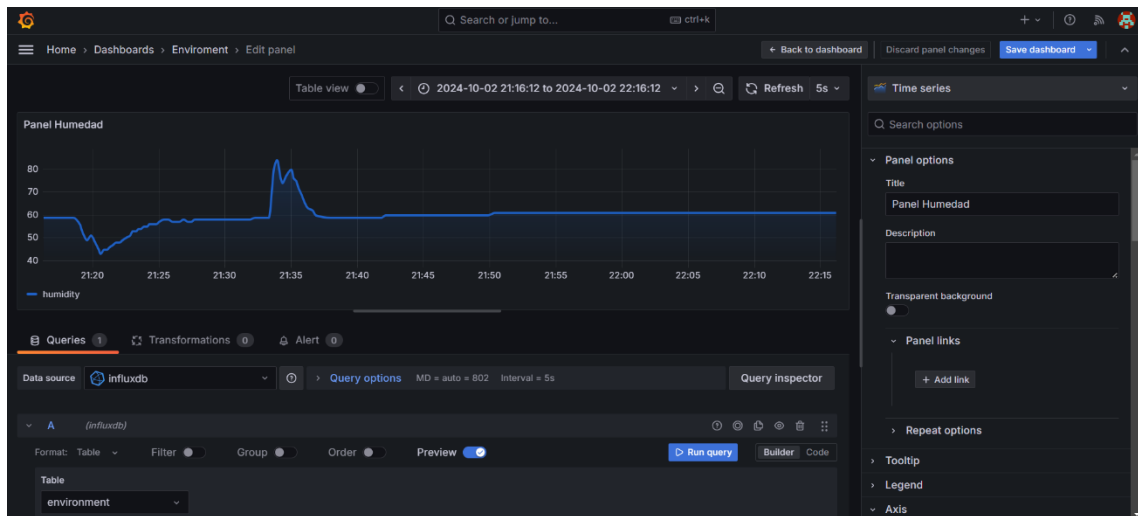


Figura 23. Configuración panel humedad

- Panel Valor Temperatura Actual**

En este panel se puede observar la temperatura muestreada (solamente la actual), además se observa la tasa de cambio respecto al valor anterior, entre las configuraciones especiales se encuentran, el color de los números, que depende de los umbrales del panel temperatura.

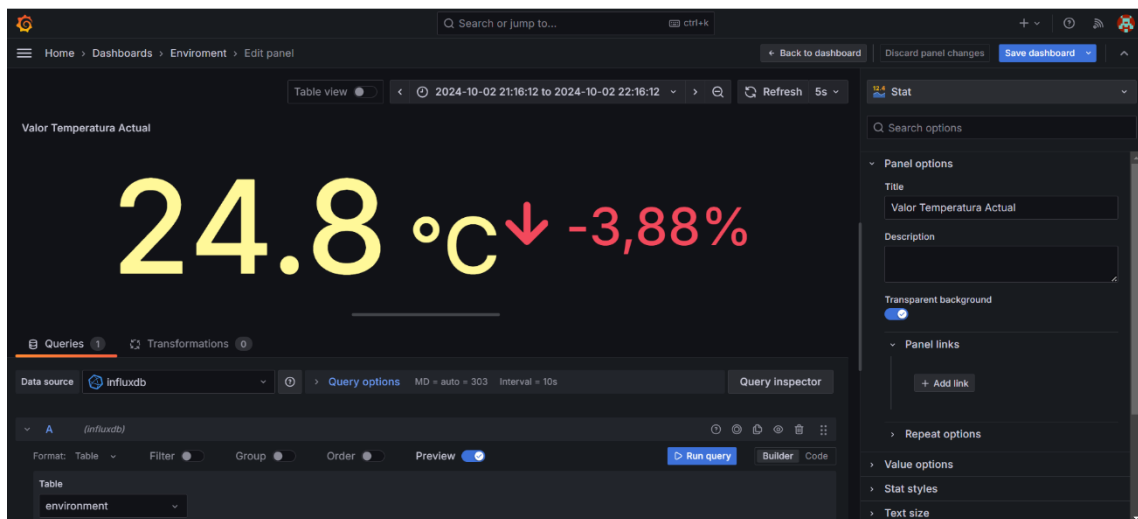


Figura 24. Configuración panel valor temperatura actual



- **Panel Valor Humedad Ambiente Actual**

En este panel, como en el anterior, se puede observar cual es el valor de la humedad actual, configuración a resaltar se encuentran las unidades en las que se escriben los datos, en este caso porcentaje de humedad en el ambiente (%H).

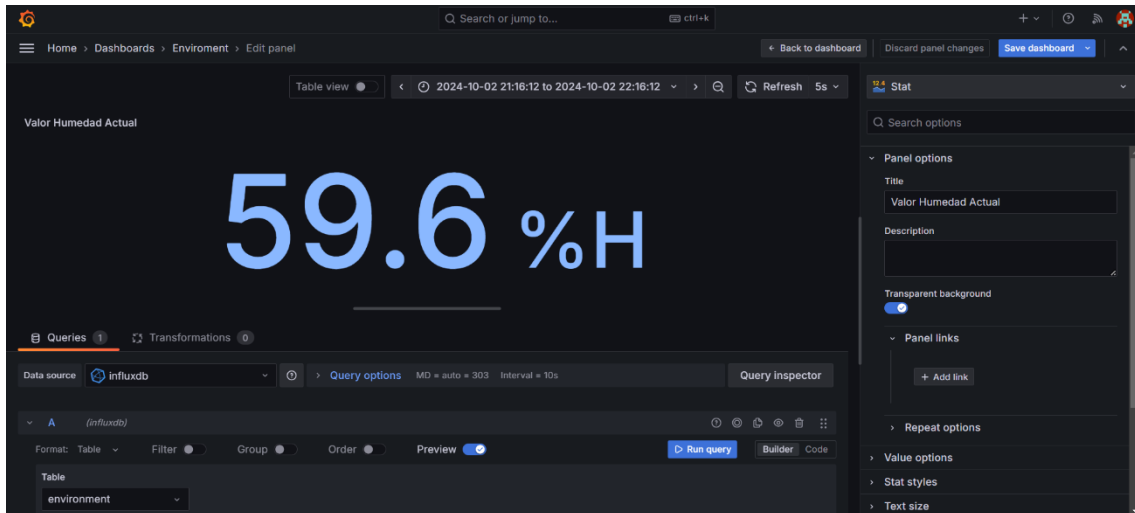


Figura 25. Configuración panel valor humedad actual

- **Tablas de Temperatura y Humedad**

En estas tablas se encuentran los datos en bruto en cuanto a la información muestreada se refiere, de las configuraciones especiales está la alineación hacia la izquierda, el cambio de formato de la hora, y el ajuste de los datos.

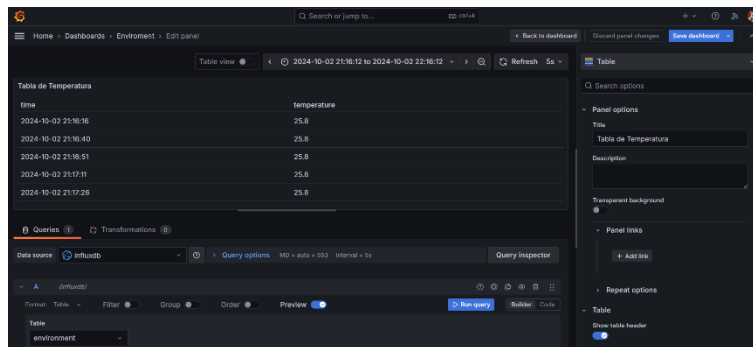


Figura 26. Configuración de tabla de temperatura

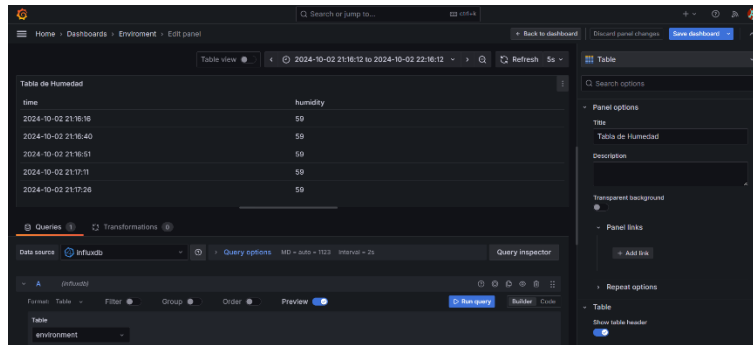


Figura 27. Configuración de tabla humedad

- **Histogramas de Temperatura y Humedad**

En los histogramas se observa la distribución que toman los datos muestreados tanto de temperatura como de humedad, estos datos presentados son discretos, de estos datos con el tiempo podríamos estimar y predecir valores futuros respecto de los antiguos, algo parecido al Machine Learning y el procesamiento de datos.

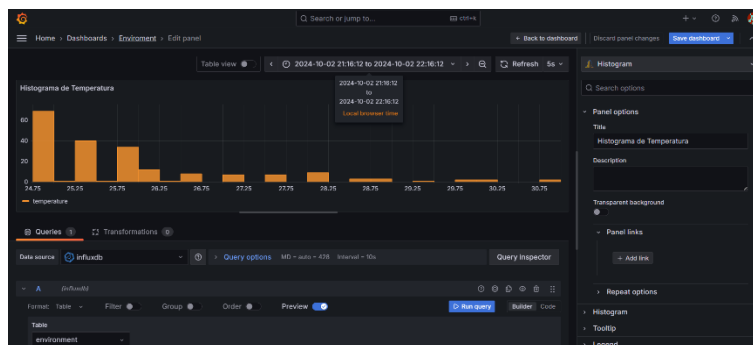


Figura 28. Configuración de histograma temperatura

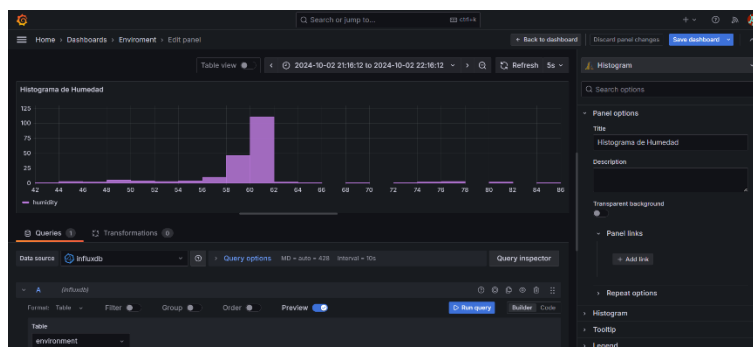


Figura 29. Configuración de histograma humedad

- **Módulos de visualización de tiempo**

Los módulos consisten en describir las demás visualizaciones y darle estética al dashboard, estos módulos funcionan en Markdown Lenguaje, entonces como configuración especial se “programa” el texto que contiene.

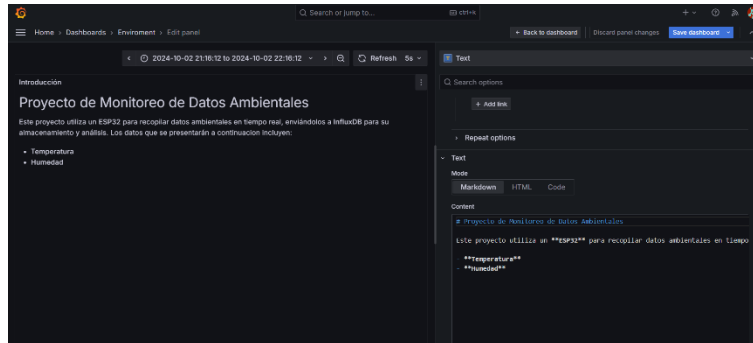


Figura 30. Configuración módulo de texto introducción

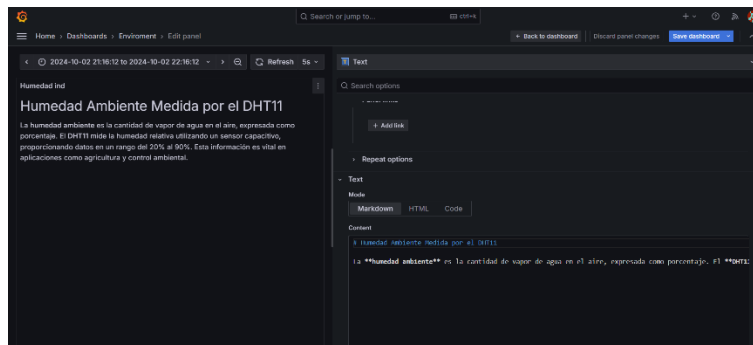


Figura 31. Configuración de texto Humedad individual

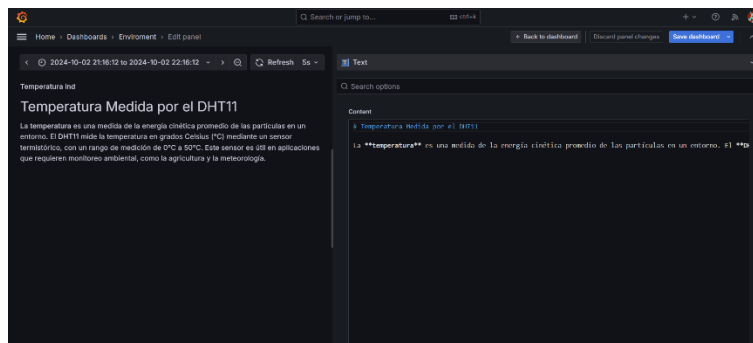


Figura 32. Configuración de texto Temperatura individual