

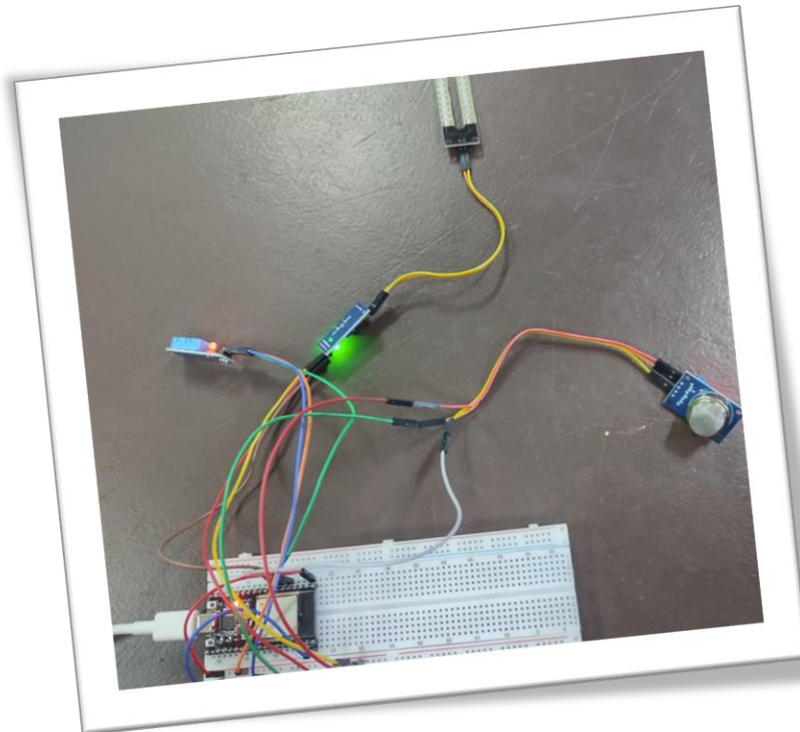


Facultad: FACULTAD DE INFORMATICA Y ELECTRONICA

Carrera: INGENIERÍA DE SISTEMAS INFORMATICOS

Materia: INGLÉS TÉCNICO 2

Docente: CESAR SUAREZ



Infraestructura Completa de iot

Integrantes:

Jhael Elisa Guevara Silva

Julia Rafaela Herrera Heiden

Grupo:

"B"

Fecha: 01/04/2025

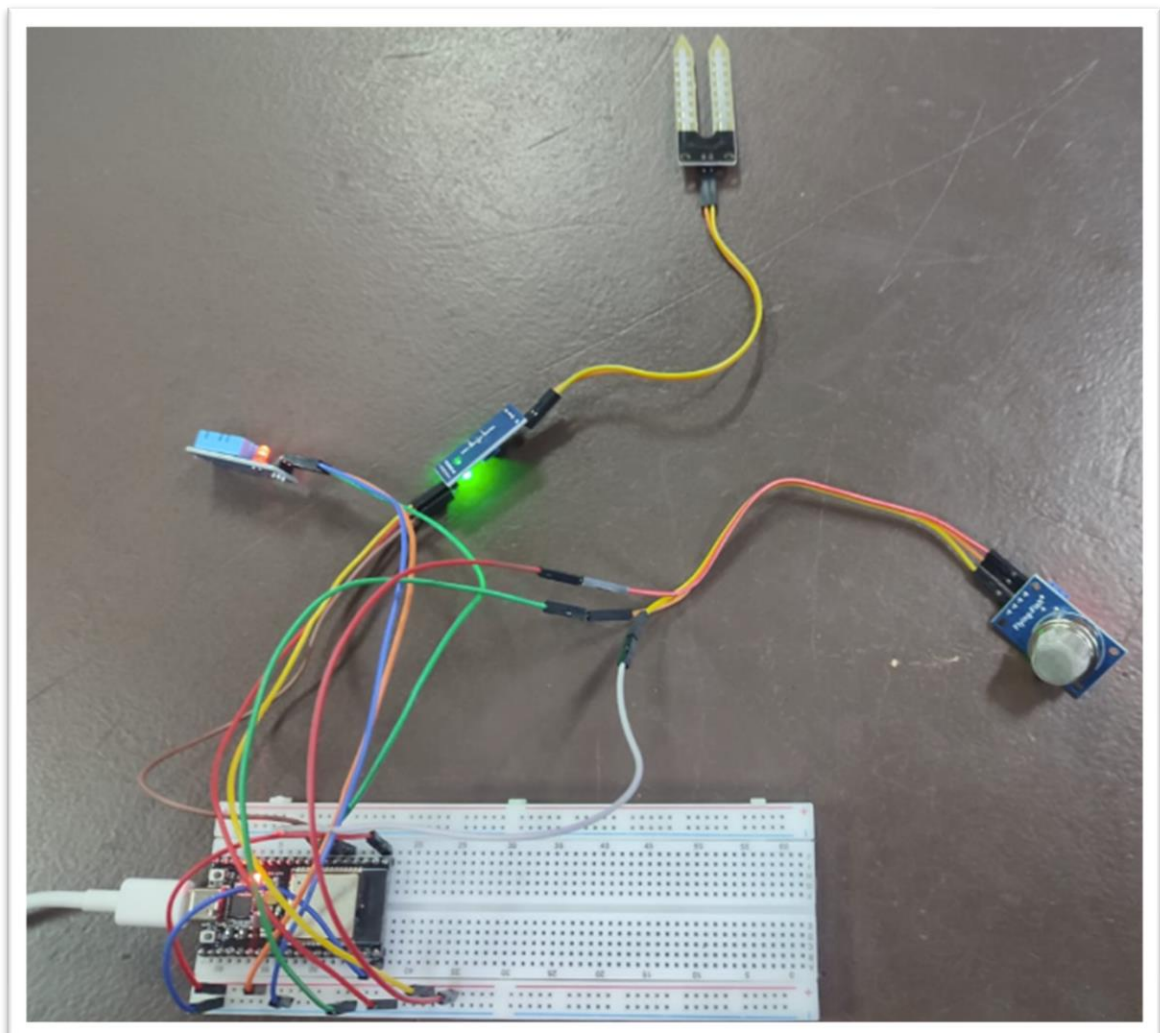
INTRODUCCIÓN

En este informe se detalla los pasos y las herramientas utilizadas al momento de la implementación de la infraestructura completa (IOT) acerca de sensores de temperatura-Humedad, Humedad del Suelo y Gas. Empezando por el ensamblado/armado del circuito.

ARMADO

- Protoboard
- ESP32 WROOM 38 pines
- DHT11 Temperatura y humedad
- MQ2 Gas
- FC-28 Humedad del Suelo
- Cables macho - hembra, macho - macho, hembra - hembra

Usamos puertos analógicos para la lectura de los datos, pines como: **32** DHTPIN (temperatura) , **33** GAS_SENSOR_PIN ,**34** HUMEDAD_SUELO_PIN, **DHTTYPE** DHT11(Humedad).



ARDUINO

Para el código de ARDUINO, usamos los puertos ya mencionados anteriormente, donde también realizamos la conexión al servidor MQTT. La conexión del ESP32 se realiza mediante WIFI:

// 🚀 Configuración WiFi

```
const char* ssid = "Estudiantes"; // Cambia esto por tu red WiFi
```

```
const char* password = "EstWifi.0224";
```

Para la conexión con MQTT usamos el servidor MQTTX :

[Easy-to-Use Online MQTT Client | Try Now](#)

Que está configurado con tópicos: sensor/datos

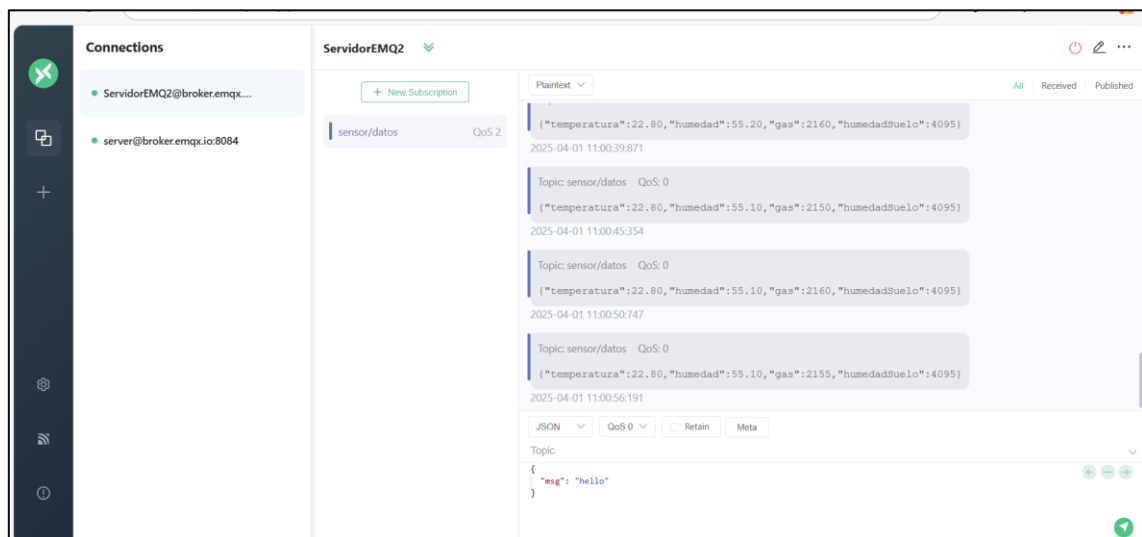
// 🚀 Configuración del broker EMQX

```
const char* mqtt_server = "broker.emqx.io"; // Servidor MQTT EMQX
```

```
const int mqtt_port = 1883; // Puerto MQTT sin SSL
```

```
const char* mqtt_topic = "sensor/datos"; // Tópico donde se enviarán los datos
```

A través de esta conexión se realiza el envío de datos mediante el ESP32:



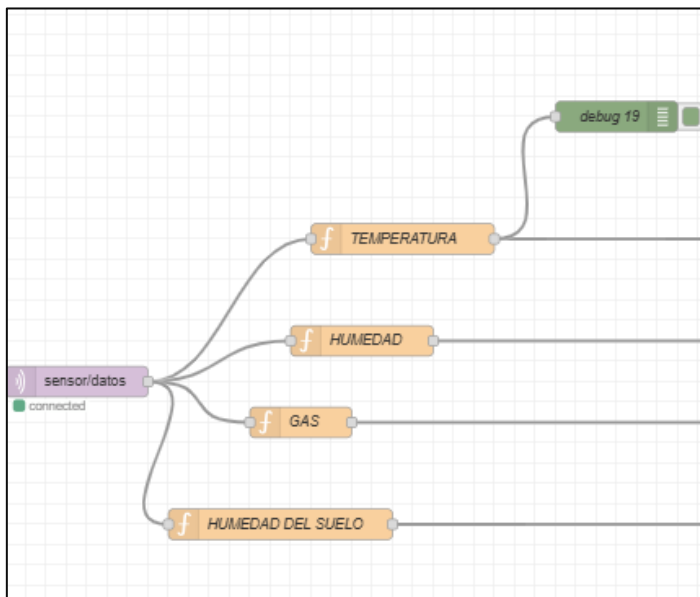
NODE_RED

Posteriormente en node-red usamos el nodo MQTT IN donde recuperamos los datos de los sensores:



Donde se configura el tópic en el que se están almacenando los datos.

Estos datos se reciben mediante funciones, donde se realiza una validación para ver si el valor del dato recibido tiene un cambio de al menos un dígito, esto para no tener datos de manera redundante:



Usamos un nodo **debug** para comprobar los datos de llegada.

La manera en la que controlamos el cambio de un dígito es:

```
if (temperaturaAnterior === undefined || Math.abs(temperaturaActual -  
temperaturaAnterior) >= 1) {  
  
    // Guardar la nueva temperatura como la actual  
  
    global.set("temperatura", temperaturaActual);  
}
```

A través de Math.Abs Que nos permite verificar esa diferencia, ya sea que aumente o disminuya.

Ejemplo del nodo function Temperatura:

```
// Obtener la temperatura anterior guardada en una variable global
var temperaturaAnterior = global.get("temperatura");

// Convertir el JSON entrante
try {
    var datos = JSON.parse(msg.payload);

    var temperaturaActual = datos.temperatura; // Extraer la nueva temperatura

    // Verificar si hay un cambio de al menos 1 unidad
    if (temperaturaAnterior === undefined || Math.abs(temperaturaActual -
temperaturaAnterior) >= 1) {

        // Guardar la nueva temperatura como la actual
        global.set("temperatura", temperaturaActual);

        msg.payload = temperaturaActual;

        msg.topic = "temperatura";

        return msg; // Retornar el mensaje con la nueva temperatura
    } else {

        // No se retorna ningún mensaje si no hay cambio significativo

        return null;

    }
} catch (error) {

    // Manejar posibles errores en el JSON

    node.error("Error al procesar el JSON: " + error.message);

    return null;

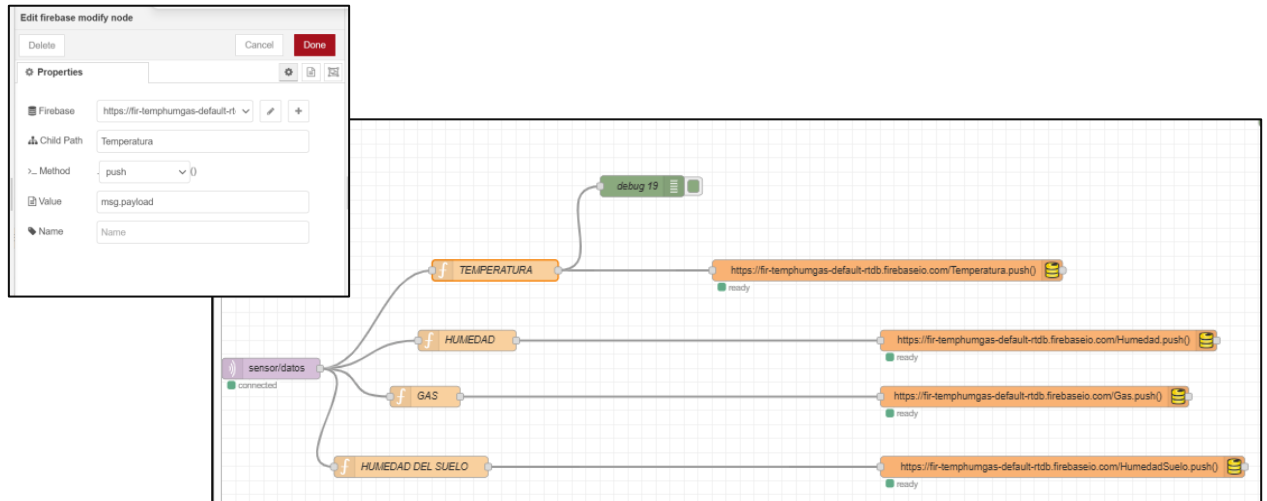
}
```

NOTA: Se usan variables globales para almacenar los datos, y que cualquier nodo pueda acceder a esa información sin necesidad de enlaces entre ellos.

IMPLEMENTACIÓN CON FIREBASE

Usamos los nodos **push** (Para mandar los datos recuperados del servidor MQTT a través de node-red).

Configuración del nodo:



A través del push se reciben los datos en el servidor de FIREBASE:



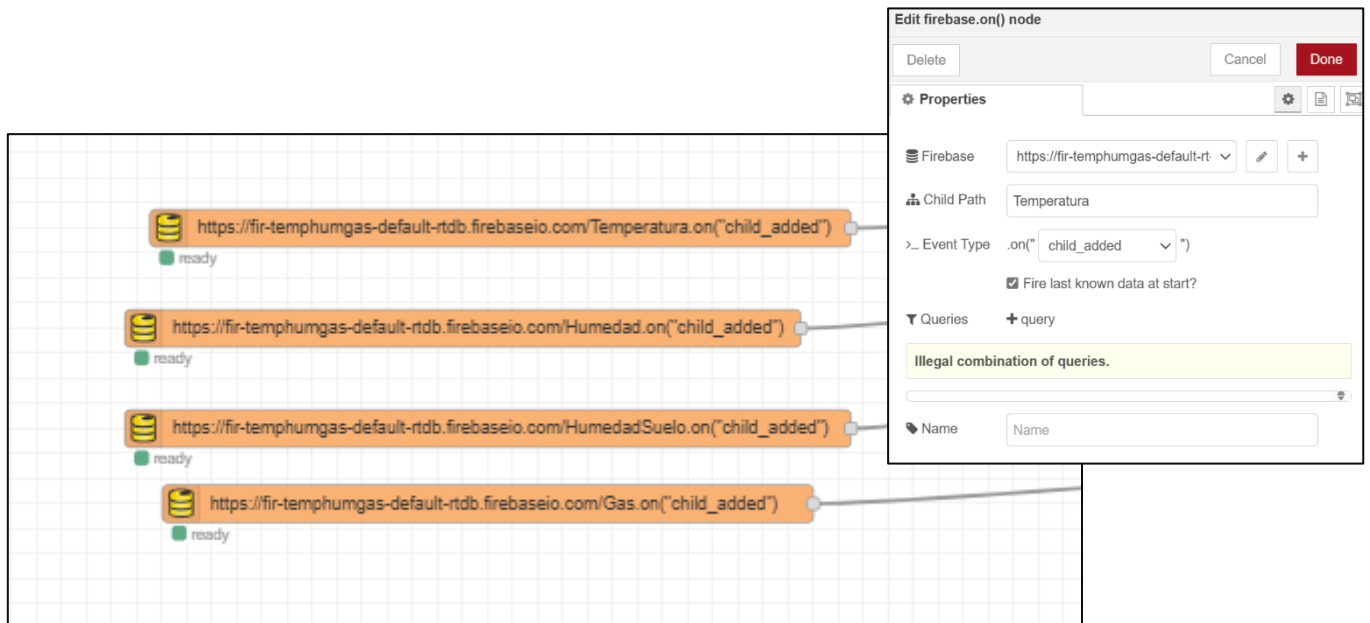
Estos son los nombres de los hijos, que es por donde se reciben los datos en el servidor.

Ejemplo de los datos que recibe:



Y el nodo `.on` Para mandar los datos recibidos:

Configuración del nodo:



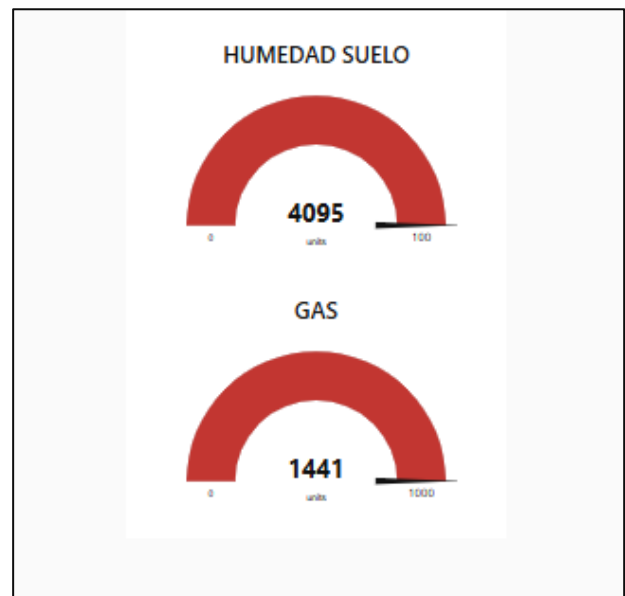
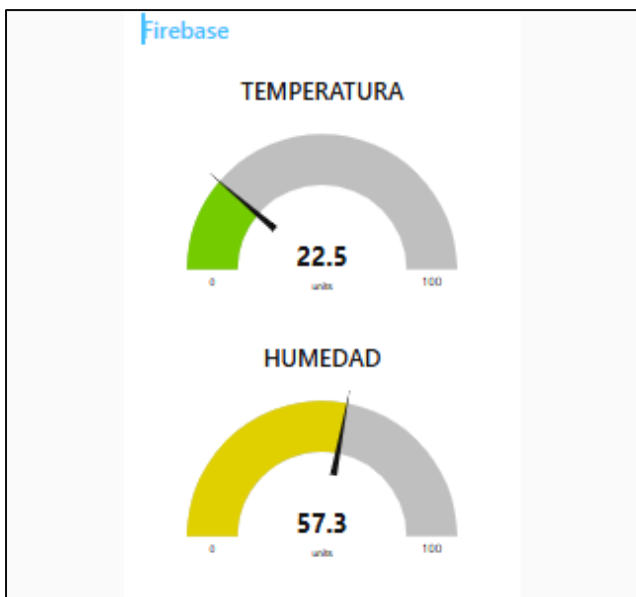
The screenshot shows a Node-RED workspace with four Firebase nodes, each with a 'ready' status. The nodes are:

- `https://fir-temphumgas-default-rtdb.firebaseio.com/Temperatura.on("child_added")`
- `https://fir-temphumgas-default-rtdb.firebaseio.com/Humedad.on("child_added")`
- `https://fir-temphumgas-default-rtdb.firebaseio.com/HumedadSuelo.on("child_added")`
- `https://fir-temphumgas-default-rtdb.firebaseio.com/Gas.on("child_added")`

The configuration panel for the selected node is shown on the right:

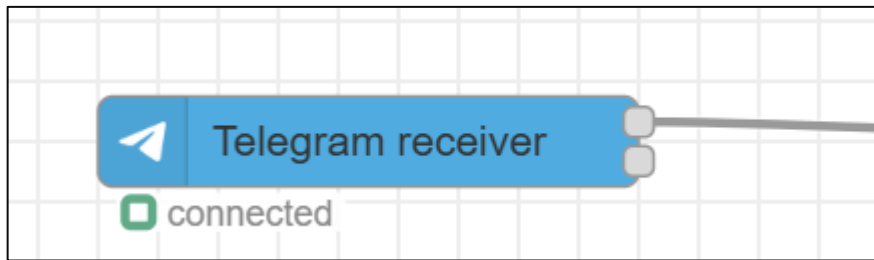
- Edit firebase.on() node**
- Properties**
- Firebase:** `https://fir-temphumgas-default-rt`
- Child Path:** `Temperatura`
- Event Type:** `.on("child_added")`
- ☒ **Fire last known data at start?**
- Queries:** `+ query`
- Illegal combination of queries.**
- Name:** `Name`

Al obtener los datos, estos son graficados en un dashboard:

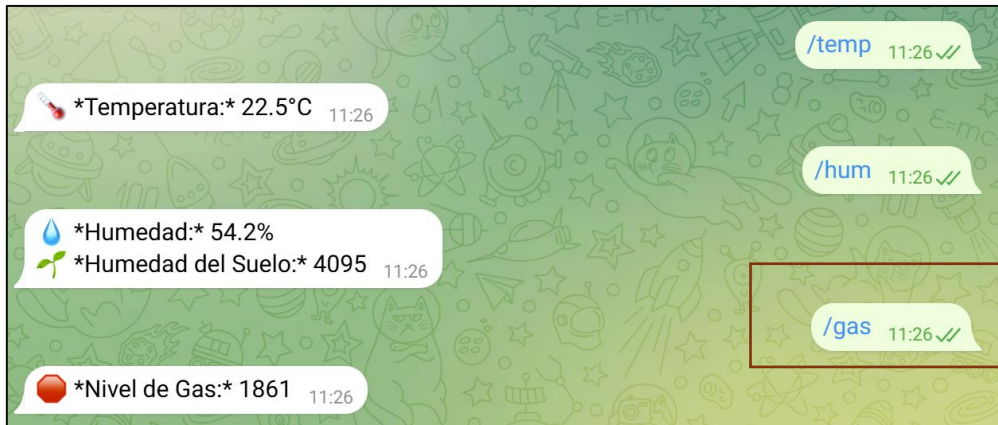


TELEGRAM

Implementamos nodos como : telegram receiver:



Recibe el mensaje de
nuestro bot :
@agucate_bot



Ejemplo desde la terminal de node:red (formato JSON)

```
1/4/2025, 12:01:05 node: debug 20
msg.payload : Object
  ► { chatId: 6250890163, messageId: 343,
    type: "message", content: "/temp", date:
    1743523265 }
```

Estos datos se reciben desde un nodo function, donde se configuran datos como:

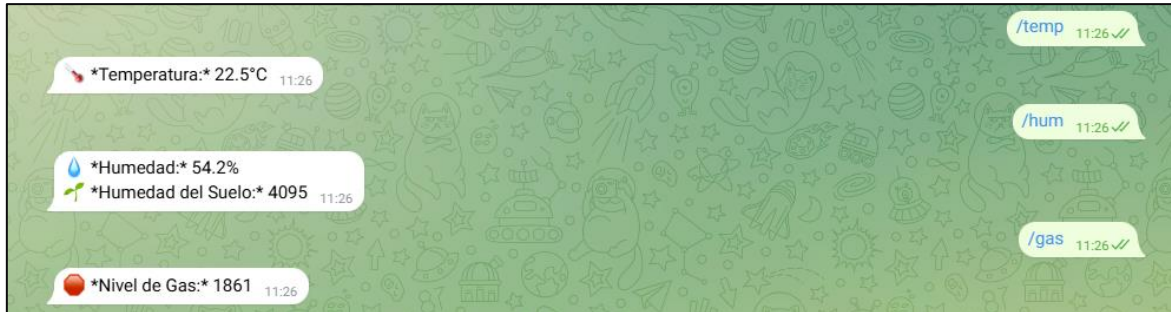
- Chat id
- Type (salida)
- Variables globales
- Verifica el tipo de mensaje, para devolver un valor diferente

Ejemplo:

/temp Temperatura

/hum Humedad DHT11 – Humedad del Suelo FC28

/Gas Gas



Y en caso de no ingresarse alguno de ellos:



Código del nodo function:

```
var chatIdPermitido = 6250890163; // Chat ID autorizado
var chatId = msg.payload.chatId; // Chat ID del mensaje recibido
var comando = msg.payload.content; // Extraer el comando
var respuesta = "";

// Validar que el mensaje venga del chat correcto
if (chatId !== chatIdPermitido) {
    msg.payload = { chatId: chatId, type: "message", content: "❌ No tienes permisos para usar este bot." };
    return msg;
}

// Procesar comandos
if (comando === "/temp") {
    var temperatura = global.get("temperatura") || "No disponible";
    respuesta = "🌡️ *Temperatura:* " + temperatura + "°C";
}

else if (comando === "/hum") {
    var humedad = global.get("humedad") || "No disponible";
    var humedadSuelo = global.get("humedadSuelo") || "No disponible";
    respuesta = "💧 *Humedad:* " + humedad + "%\n🌱 *Humedad del Suelo:* " + humedadSuelo;
}






else if (comando === "/gas") {
    var gas = global.get("gas") || "No disponible";
    respuesta = "💣 *Nivel de Gas:* " + gas;
}

else {
    respuesta = "❌ *Comando no reconocido.*\n\nUsa:\n/temp → 🌡️ Temperatura\n/hum → 💧 Humedad y Suelo\n/gas → 💣 Nivel de Gas";
}

// Enviar mensaje solo al chat autorizado
msg.payload = { chatId: chatIdPermitido, type: "message", content: respuesta };
return msg;
```

HTML

Para el código html se usa los datos de la api generada al añadir una web.

Apps web	Sobrenombre de la app
 Gráficos Estadísticos App web	Gráficos Estadísticos 
Apps pendientes de eliminación	ID de la app 
1 app pendiente de eliminación	1:743969157528:web:d53ff41a6c18aad7d57bed
	Sitio de Firebase Hosting vinculado
	 fir-temphumgas-cba44 

Ejemplo:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCufJCzZsN4kpfnGo6gg2Qy2ymgLUC1h7I",
  authDomain: "fir-temphumgas.firebaseio.com",
  databaseURL: "https://fir-temphumgas-default-rtdb.firebaseio.com",
  projectId: "fir-temphumgas",
  storageBucket: "fir-temphumgas.firebaseio.com",
  messagingSenderId: "743969157528",
  appId: "1:743969157528:web:d53ff41a6c18aad7d57bed"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Esto debe de configurarse dentro del código de HTML:

```
const firebaseConfig = {
  apiKey: "AIzaSyCufJ...",
  authDomain: "fir-temphumgas.firebaseio.com",
  databaseURL: "https://fir-temphumgas-default-rtdb.firebaseio.com",
  projectId: "fir-temphumgas",
  storageBucket: "fir-temphumgas.firebaseio.com",
  messagingSenderId: "743969157528",
  appId: "1:743969157528:web:d53ff41a6c18aad7d57bed"
};
```

Esto para poder recibir los datos y así ser graficados.

El enlace nos sirve para tener un acceso a los datos.

PAGINA WEB

Usamos Firebase Hosting para subir la página:

Donde creaos una carpeta que contiene el mismo nombre del proyecto:

 **FirestoreTempHumGas** 01/04/2025 11... Carpeta de ...

Y dentro de la carpeta en cmd, accediento al directorio en el que se encuentra:

```
C:\Users\jhael>cd ..
```

```
C:\Users>cd ..
```

```
C:\>cd fir-temphumgas-cba44
```

Luego se configura firebase dentro del documento para poder subir el código html:

```
C:\FirestoreTempHumGas>firebase init
```

```
#####  #####  #####  #####  #####  #####  #####  #####  
##      ##      ##      ##      ##      ##      ##      ##  
#####  ##  #####  #####  #####  #####  #####  #####  
##      ##      ##      ##      ##      ##      ##      ##  
##      #####  ##      ##  #####  #####  ##      ##  #####
```

You're about to initialize a Firebase project in this directory:

```
C:\FirestoreTempHumGas
```

Before we get started, keep in mind:

- * You are currently outside your home directory



Se seleccionan las características a instalar:

```
? Are you ready to proceed? Yes  
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to  
confirm your choices. Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys,  
Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance  
=== Project Setup  
  
First, let's associate this project directory with a Firebase project.  
You can create multiple project aliases by running firebase use --add,  
but for now we'll just set up a default project.  
  
? Please select an option: Use an existing project  
? Select a default Firebase project for this directory: fir-temphumgas (FirestoreTempHumGas)  
i Using project fir-temphumgas (FirestoreTempHumGas)  
  
=== Hosting Setup  
  
Your public directory is the folder (relative to your project directory) that  
will contain Hosting assets to be uploaded with firebase deploy. If you  
have a build process for your assets, use your build's output directory.  
  
? What do you want to use as your public directory? public  
? Configure as a single-page app (rewrite all urls to /index.html)? No  
? Set up automatic builds and deploys with GitHub? No  
+ Wrote public /404.html  
+ Wrote public /index.html
```

Para al final el documento creado se vea de esta manera:

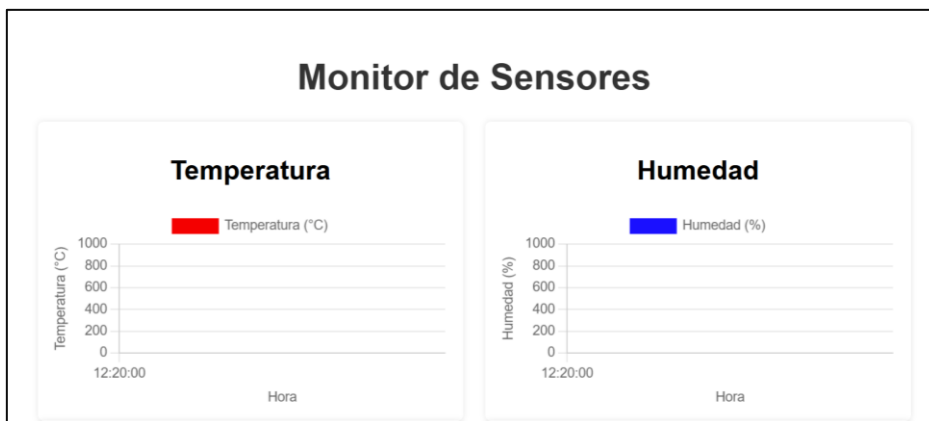
Nombre	Fecha de modi...	Tipo	Tamaño
 .firebase	01/04/2025 11...	Carpeta de ...	
 public	01/04/2025 11...	Carpeta de ...	
 .firebaserc	01/04/2025 11...	Archivo FIR...	1 KB
 .gitignore	01/04/2025 11...	Archivo de ...	2 KB
 1.html	30/03/2025 18...	Microsoft E...	6 KB
 database.rules.json	01/04/2025 11...	Archivo JSO...	1 KB
 firebase.json	01/04/2025 11...	Archivo JSO...	1 KB

Se accede a la carpeta public

 404.html	01/04/2025 11...	Microsoft E...	2 KB
 index.html	01/04/2025 11...	Microsoft E...	6 KB

Y se hace click en: index-html:

Esta es la página configurada donde se alojará la web.



Para subir el código se modifica el código a través de un txt y se lo guarda:

```
node-rex arduino. inde *
Archivo  Editor  Ver
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=
1.0">
  <title>Monitor de Sensores</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
    }
    .container {
      width: 80%;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
  </style>
</head>
<body>
  <div class="container">
    <div>
      <h2>Monitor de Sensores</h2>
      <div>
        <div>
          <h3>Temperatura</h3>
          <img alt="Line chart for Temperature (°C) with a red line." data-bbox="175 585 425 665"/>
        </div>
        <div>
          <h3>Humedad</h3>
          <img alt="Line chart for Humidity (%) with a blue line." data-bbox="455 585 705 665"/>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Se debe de ejecutar el comando firebase deploy para que se actualicen los cambios, y así poder acceder a la web actualizada:

```
C:\FirebaseTempHumGas>firebase deploy

=== Deploying to 'fir-temphumgas'...

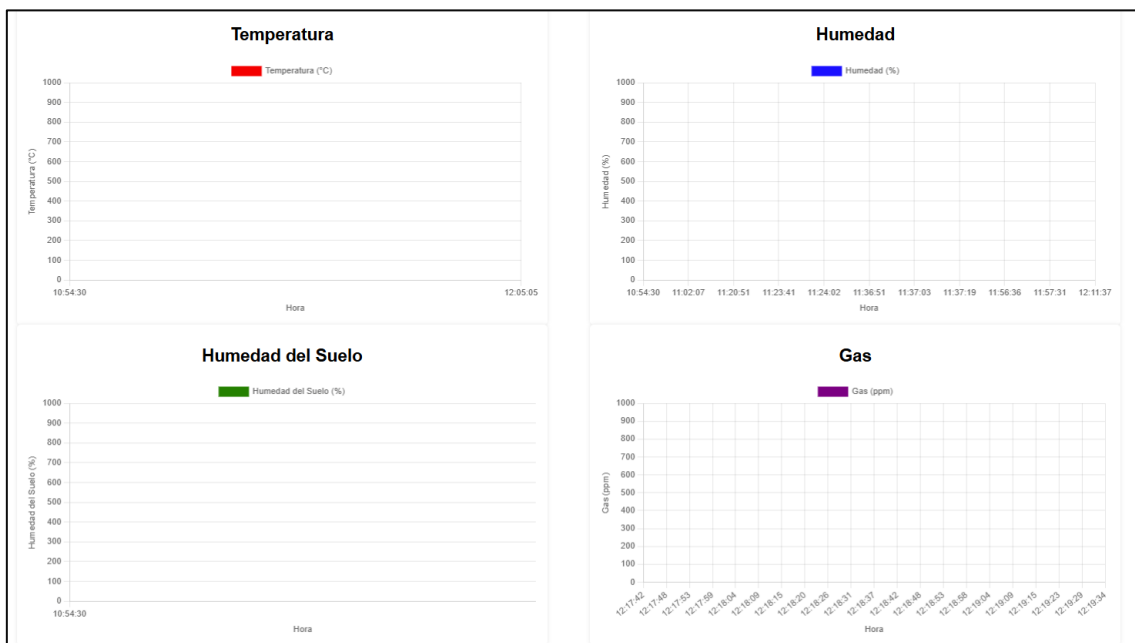
i deploying database, hosting
i database: checking rules syntax...
+ database: rules syntax for database fir-temphumgas-default-rtdb is valid
i hosting[fir-temphumgas]: beginning deploy...
i hosting[fir-temphumgas]: found 2 files in public
+ hosting[fir-temphumgas]: file upload complete
i database: releasing rules...
+ database: rules for database fir-temphumgas-default-rtdb released successfully
i hosting[fir-temphumgas]: finalizing version...
+ hosting[fir-temphumgas]: version finalized
i hosting[fir-temphumgas]: releasing new version...
+ hosting[fir-temphumgas]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/fir-temphumgas/overview
Hosting URL: https://fir-temphumgas.web.app
```

Posteriormente se ingresa al link que se encuentra en la parte inferior, pero también se puede acceder desde el index.html de la carpeta public.

Y el resultado de la web, se vería de esta manera:



Donde cada tabla recibe datos de manera independiente, y son registrados en el gráfico según la hora en la que se produjo el cambio.

RECURSOS

Monitor de Sensores (link de la app)

Easy-to-Use Online MQTT Client | Try Now (Servidor Mqttx usado)

FirebaseTempHumGas - Realtime Database - Datos - Firebase console
(Servidor Firebase)