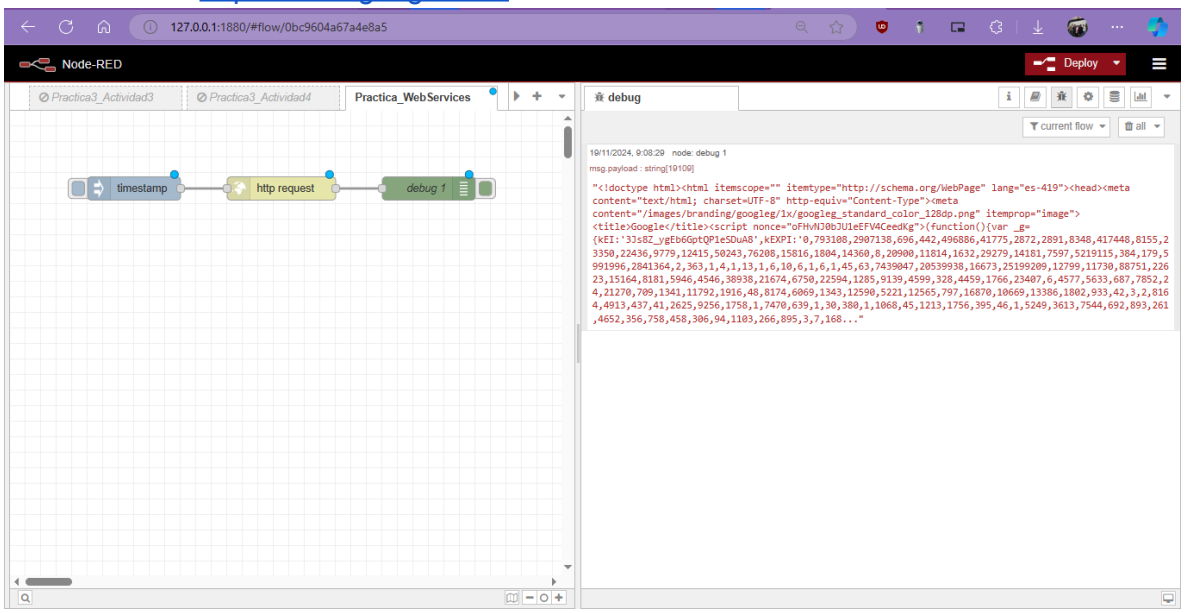
EVIDENCIAS PRACTICA WEBSERVICE

Integrantes: JOSE DAVID FLOREZ RAMOS - 2174241
DIEGO ANDRES GARCIA DIAZ - 2195533
Fecha: 19/11/2024 y 26/11/2024

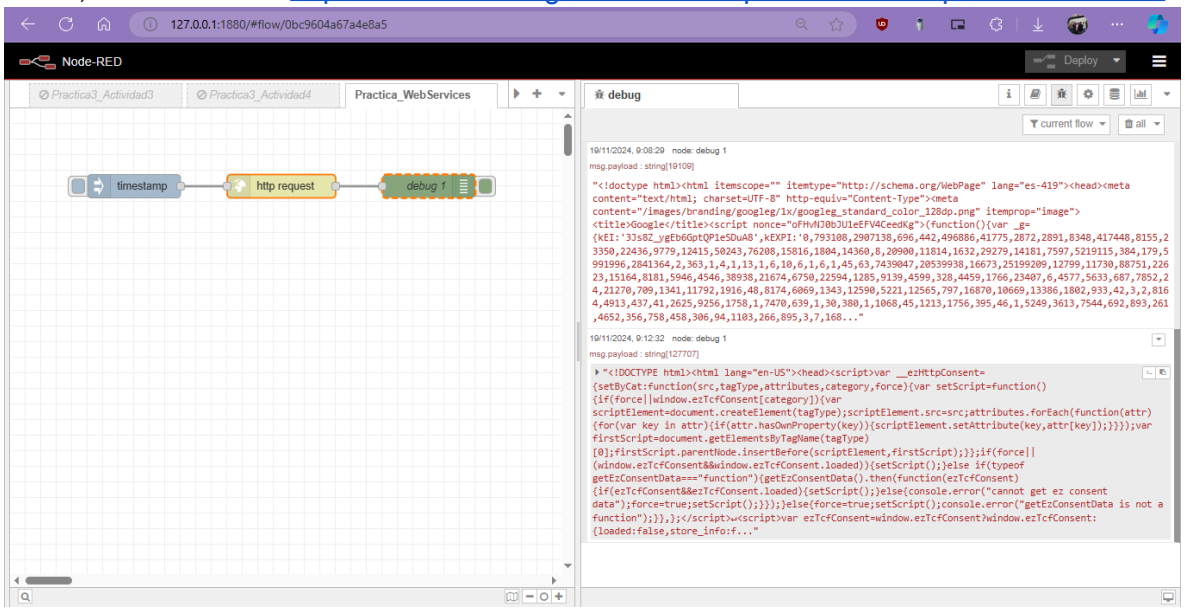
Actividad 1:

Pegar aquí los pantallazos de los resultados que evidencien la recepción de mensajes en el nodo **debug** en node red hacerlo con una página diferente a la de la guia:

Usando el link: https://www.google.com



Ahora, usando el link: https://lastminuteengineers.com/esp32-wroom-32-pinout-reference/

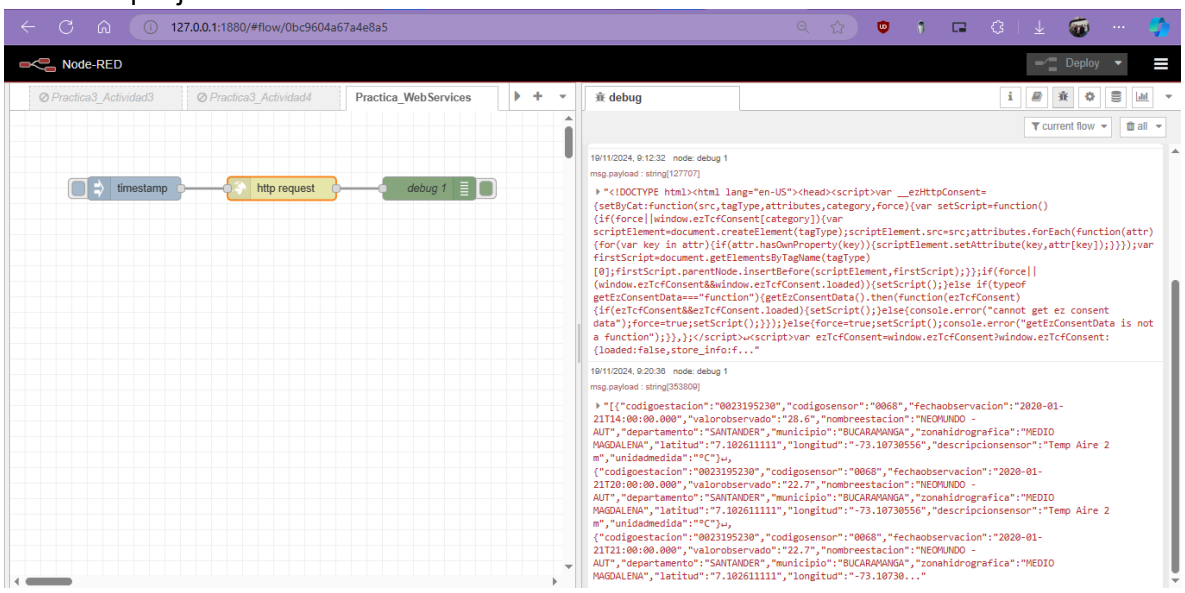


Actividad 2:

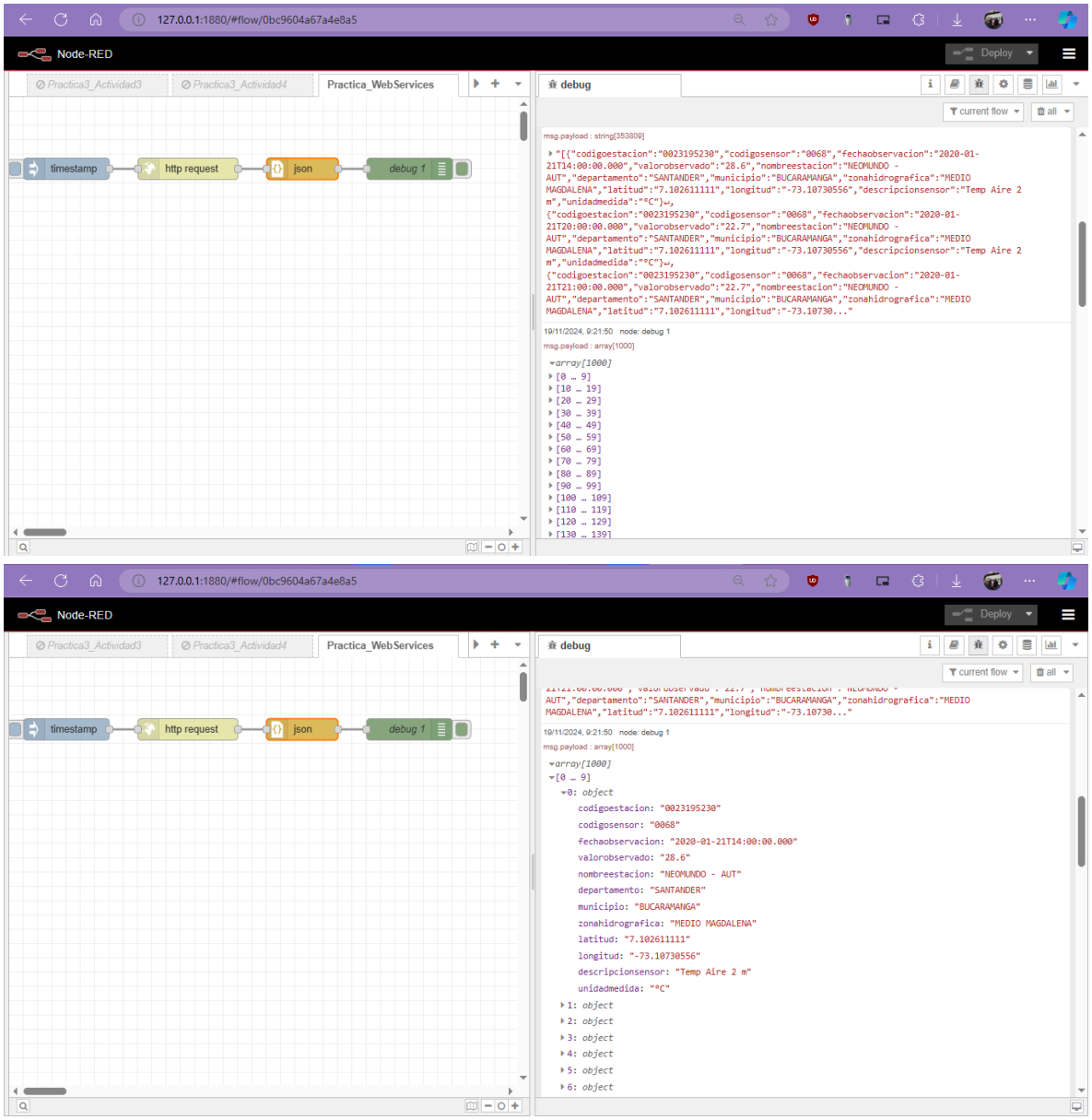
Pegar aquí el pantallazo del proceso para obtener datos meteorológicos de una estación diferente:

Estación seleccionada:
https://www.datos.gov.co/resource/sbwg-7ju4.json?codigoestacion=0023195230

Sin el bloque json:



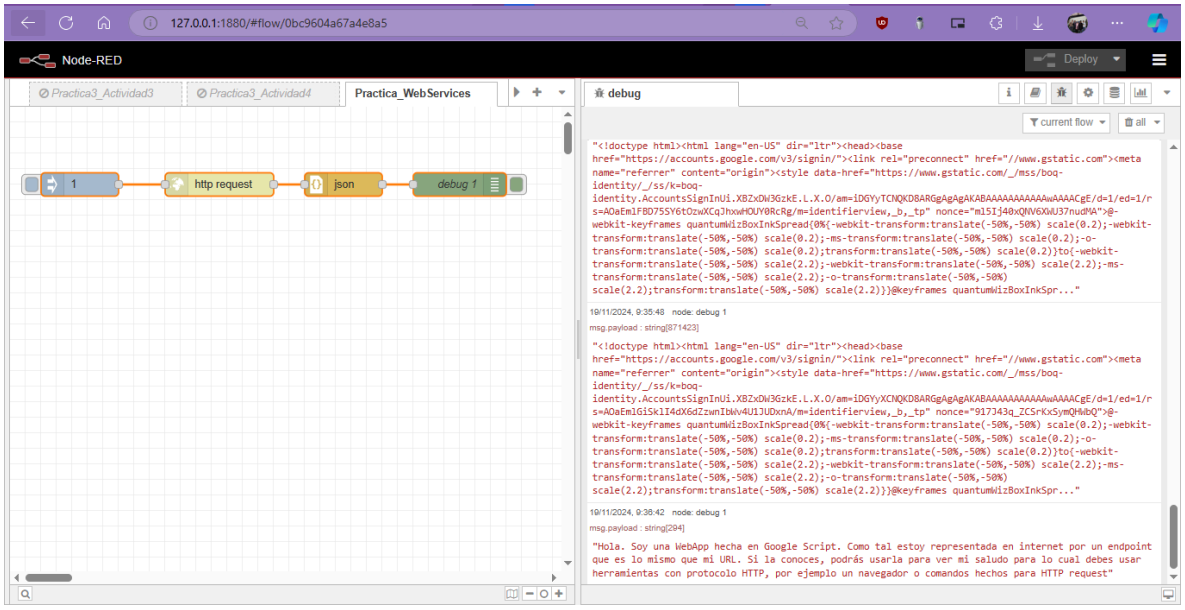
Con el bloque json:



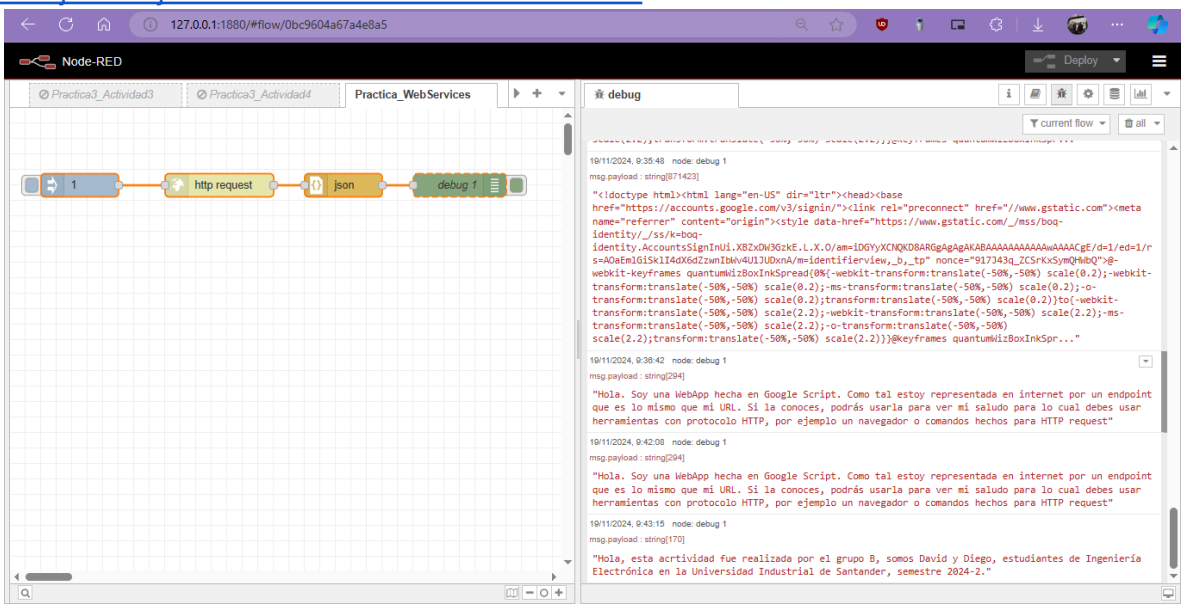
Actividad 3:

Enviar un mensaje diferente al del WebService de la guía:

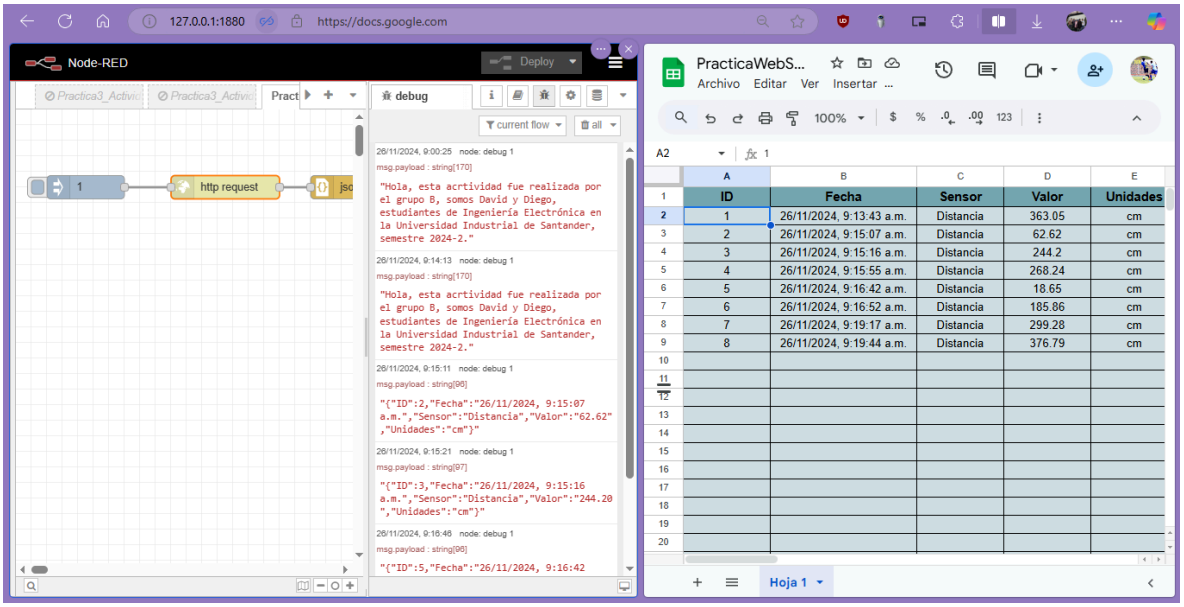
Primero se evidencia el mensaje propuesto desde Node-RED, gracias a la siguiente implementación hecha en Google Apps Script: https://script.google.com/macros/s/AKfycbwi33wotzxMNq-jZ9c-XCI7u0aTKM0JyiPVh_p0aOluXQg7U6C7pi7dehOeTTGQPY_6pg/exec



Con un mensaje diferente queda de la siguiente forma: https://script.google.com/macros/s/AKfycbxF0547bVSVQPTR5S18MFZORo4fTQtzLBT6yB47fjCnWFtjrh2xWbWGESEXW80J3FKOIA/exec



Actividad 26/11/2024:
Inicialmente se modificó el código en GAS para que agregara datos a un Google Sheets desde la ejecución del código en GAS y también desde el bloque **inject** en Node-RED:



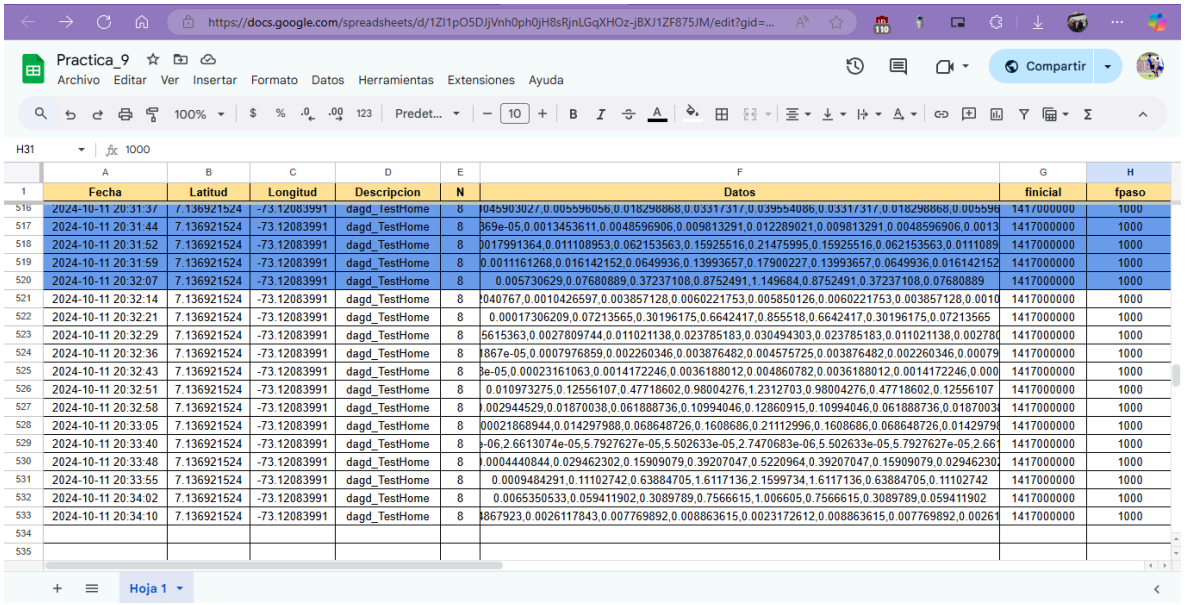
El código que se uso en GAS fue el siguiente:

```
//// CÓDIGO PARA CARGAR DATOS AL GOOGLE SHEETS Y VISUALIZAR EN NODE-RED
// URL del Google Sheets
const SHEET_ID = "1hJeggCC7pwvjH1Z5ZW4-vlTiQLOoQ5lerBBLEbd7WhE";
const SHEET_NAME = "Hoja 1";

// Función para manejar las solicitudes GET
function doGet() {
  // Generar un ID único
  const id = generateId();

  // Obtener la fecha y hora actual
  const fecha = new Date().toLocaleString();

  // Simular un valor aleatorio para el sensor de distancia
  const valorSensor = Math.random() * (400 - 10) + 10; // Rango: 10 a 400
cm

  // Unidades del sensor
  const unidades = "cm";

  // Construir el objeto de datos
  const data = {
    ID: id,
    Fecha: fecha,
    Sensor: "Distancia",
    Valor: valorSensor.toFixed(2),
    Unidades: unidades
  };

  // Guardar los datos en Google Sheets
  guardarEnSheets(data);
```

```
  // Enviar los datos en formato JSON
  return ContentService.createTextOutput(JSON.stringify(data))
                       .setMimeType(ContentService.MimeType.JSON);
}

// Función para guardar datos en Google Sheets
function guardarEnSheets(data) {
  const ss = SpreadsheetApp.openById(SHEET_ID);
  const sheet = ss.getSheetByName(SHEET_NAME);

  if (sheet) {
        sheet.appendRow([data.ID,  data.Fecha,  data.Sensor,  data.Valor,
data.Unidades]);
  }
}

// Generar un ID único incremental
function generateId() {
  const ss = SpreadsheetApp.openById(SHEET_ID);
  const sheet = ss.getSheetByName(SHEET_NAME);
  const lastRow = sheet.getLastRow();

  if (lastRow === 1) {
    // Si solo están los encabezados
    return 1;
  } else {
    // Incrementar el último ID
    const lastId = sheet.getRange(lastRow, 1).getValue();
    return lastId + 1;
  }
}
```

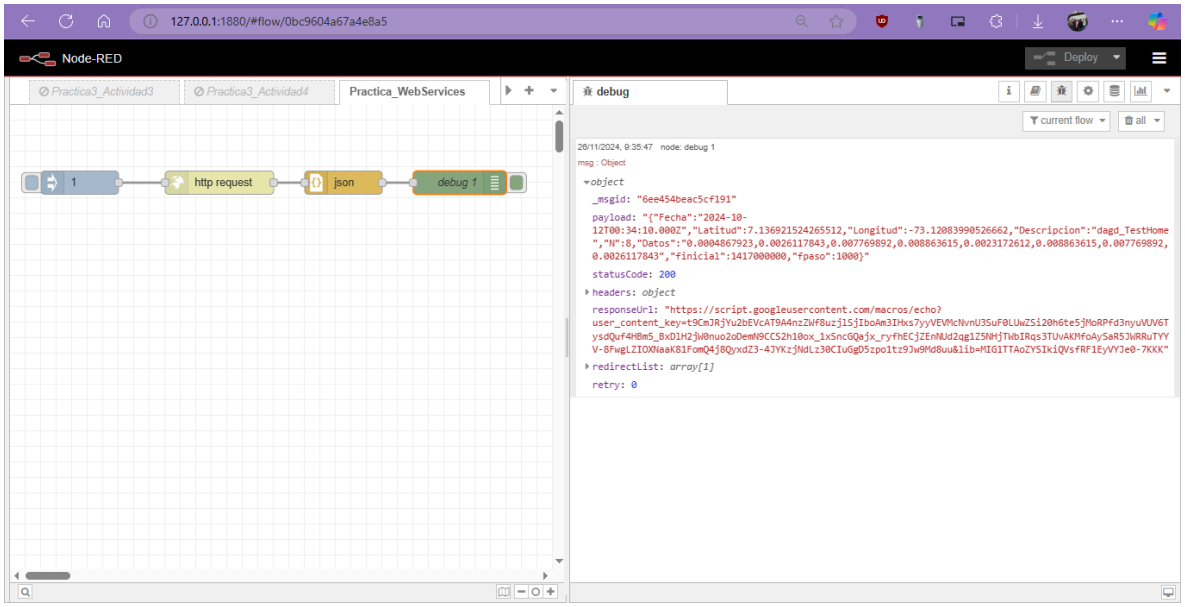
Ahora, la idea es que de una tabla con datos en google sheets, se tome una fila de esos datos y se observen dichos datos desde Node-RED y en formato JSON:

Esta es la tabla de la que se toman los datos:



La url de Google Sheets: https://docs.google.com/spreadsheets/d/1ZI1pO5DjVnh0ph0jH8sRjnLGqXHOz-jBXJ1ZF875JM/edit?usp=sharing

Lo que se visualiza en Node-RED:



Finalmente la url de la implementación en Google Apps Script: https://script.google.com/macros/s/AKfycby85SpUCOY6ZpOKbIGLdX_t3XSHr3neLfzu6_OXmXFiLNbZFHlc82gSPMjk9gQymSrGmw/exec

Código usado en Google Apps Script:

```
//// CÓDIGO PARA LEER DATOS DEL GOOGLE SHEETS Y VISUALIZAR EN NODE-RED
// URL del Google Sheets
const SHEET_ID = "1ZI1pO5DjVnh0ph0jH8sRjnLGqXHOz-jBXJ1ZF875JM";
const SHEET_NAME = "Hoja 1";

// Función para manejar las solicitudes GET
```

```
function doGet() {
  // Obtener el último registro de la hoja de cálculo
  const ultimoRegistro = obtenerUltimoRegistro();

  // Enviar los datos en formato JSON
  return ContentService.createTextOutput(JSON.stringify(ultimoRegistro))
                       .setMimeType(ContentService.MimeType.JSON);
}

// Función para obtener el último registro de la hoja de cálculo
function obtenerUltimoRegistro() {
  const ss = SpreadsheetApp.openById(SHEET_ID);
  const sheet = ss.getSheetByName(SHEET_NAME);

  if (!sheet) {
    throw new Error("No se encontró la hoja especificada.");
  }

  const lastRow = sheet.getLastRow();

  if (lastRow <= 1) {
    // Si no hay datos (aparte de los encabezados)
    throw new Error("No hay datos en la hoja.");
  }

  // Leer los encabezados y el último registro
            const      headers      =      sheet.getRange(1,      1,      1,
sheet.getLastColumn()).getValues()[0];
          const     lastData     =     sheet.getRange(lastRow,     1,     1,
sheet.getLastColumn()).getValues()[0];

  // Crear un objeto JSON basado en los encabezados y el último registro
  const data = {};
  headers.forEach((header, index) => {
    data[header] = lastData[index];
  });

  return data;
}
```