Solución IIoT con GNU Radio Parte 2

Introducción

Podemos ver esta práctica como un segundo sprint para la solución de IIoT con GNU Radio que comenzó en práctica anterior. Este es el <u>enlace de taller parte1</u>

El reto para el sprint

Se cuenta con una solución que registra el espectro de una señal en una Google Sheet. Queremos ver como una webapp logra visualizar el espectro a medida que se va registrando. Se cuenta con una webapp para la visualización del espectro, la cual está dada por la siguiente: documentación

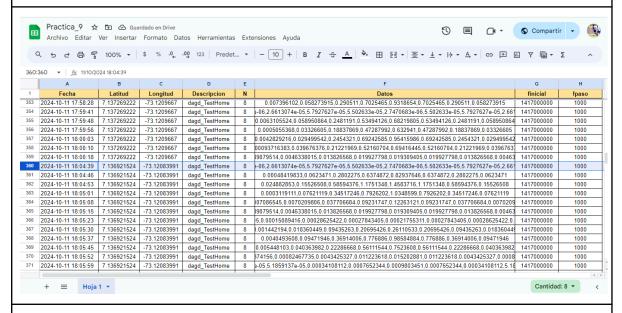
Tarea 1. Comprobar el funcionamiento de la Web App con tu Google Sheet

Paso 1. Reunir datos de entrada

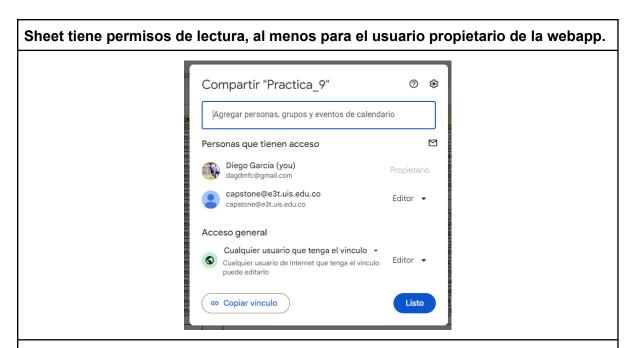
Cual es la URL de tu Google Sheet:

https://docs.google.com/spreadsheets/d/1ZI1pO5DJjVnh0ph0jH8sRjnLGqXHOz-jBXJ1ZF875JM/edit?usp=sharing

Encabezados y datos correctos en la google sheet. Muestra una captura de pantalla que demuestre que tu Google sheet tiene los encabezados como se explica en la documentación de la webapp, pero también que hayan datos registrados del espectro.



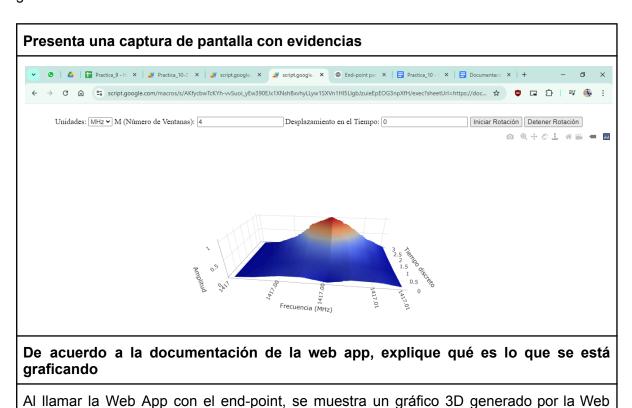
Permisos correctos. Muestra una captura de pantalla que demuestre que la Google



Cual es el end-point completo (URL más query string) a usar para la visualización, de acuerdo a la documentación de la web app:

https://script.google.com/macros/s/AKfycbwTcKYh-vvSuoi_yEw390EJx1X NshBxvhyLLyw1SXVn1Hl5LlgbJzuieEpE0G3npXfH/exec?sheetUrl=https://docs.google.com/spreadsheets/d/1ZI1p05DJjVnh0ph0jH8sRjnLGqXH0z-jBXJ1 ZF875JM/edit&unit=MHz&M=4

Paso 2. Llamar la webApp con el end-point y comprobar el funcionamiento sin correr gnuradio.



App que se está utilizando. Según la documentación del servicio, lo que se está graficando es una representación tridimensional de un espectro de señal, en donde:

- 1. **El eje X (Frecuencia en MHz)**: Representa la frecuencia en megahercios. En este caso, se observa un rango cercano a 1417 MHz.
- 2. El eje Y (Tiempo discreto o Ventanas espectrales): Este eje indica el número de ventanas espectrales o "tiempo discreto". El parámetro M, que se ha configurado a 4 en la URL del end-point, controla cuántas ventanas espectrales se visualizan. Cada ventana espectral representa un conjunto de datos correspondientes a un intervalo específico de tiempo o una segmentación temporal de la señal.
- 3. **El eje Z (Amplitud)**: Muestra la amplitud o magnitud de la señal en las diferentes frecuencias y ventanas temporales. En el gráfico, se puede observar cómo varía la amplitud de la señal conforme cambian las frecuencias y las ventanas espectrales.

Lo que se está viendo es cómo la **amplitud de la señal varía en función de la frecuencia** a través de varias ventanas temporales. La parte elevada del gráfico representa la máxima amplitud en una frecuencia específica, indicando una concentración o un pico en esa frecuencia en particular.

Este tipo de gráfico se utiliza para analizar cómo se distribuye la energía de la señal en distintas frecuencias y en diferentes intervalos de tiempo o ventanas temporales, lo cual es útil en estudios de espectros de señales.

Revise el código de la Web App, con el enlace que proporciona la documentación y selecciona falso verdadero en las siguientes preguntas:

El cálculo de los valores que aparecen en el eje de las frecuencias ocurre en el backend:

(Verdadero). Según lo que se encuentra en la documentación, los valores de *finicial* y *fpaso* se manejan desde la parte del **back-end**, ya que estos datos de frecuencia están relacionados a los datos que lee el servicio (Web App) en el Google Sheet, se procesan los datos antes de que se visualice el espectro en 3D.

La gráfica ocurre en el backend y se le transfiere al frontend:

(Falso). En la documentación no se menciona que la gráfica del espectro se genere en el back-end, lo que realmente sucede es que en el back-end se procesan los datos y en la parte del **front-end** es que realmente se visualiza la gráfica del espectro en 3D.

Realmente todo ocurre en el frontend:

(Falso). Es necesario el uso de la parte del **front-end** y **back-end**, ya que en el back-end se hace la parte de procesamiento de datos desde el Google Sheets, posteriormente se envían esos datos al front-end para poder visualizar la gráfica del espectro.

El mismo frontend se está encargando de la lectura de los datos de la Google Sheet:

(Falso). En la parte del back-end en donde realmente se hace lectura y procesamiento de los datos almacenados en el Google Sheets, para posteriormente enviar esa información a la parte del front-end y visualizar la gráfica del espectro.

Paso 3. Llamar la Web App con el end-point y comprobar el funcionamiento mientras en gnuradio corre un flujograma que calcula el espectro de una señal de audio

Captura de pantalla con evidencias Not titled yet y-Axis 0,0005 **ග** ^{1.50e+01} 1.00e+01 Frequency (kHz) 20.00 🖈 👸 🖙 한 💷 🚳 : Iniciar Rotación Detener Rota 🖈 🖀 🖙 🌣 :

Se agregan 3 imágenes diferentes donde se evidencia que se esta ejecutando el diagrama de flujo en GNU Radio y al mismo tiempo se observa la gráfica del espectro en 3D gracias a la Web App.

Tarea 2. Crear una solución que permita mostrar en un mapa de google maps, el lugar en que se hizo la última medición y los datos del espectro.

Para esta solución es importante que sepas que cuentas con una API Key de google maps que es: AlzaSyC4wgPaAvereqeB455sLVD1W3fd2CML_oo

Lo que se busca es algo así como esto:



Paso 1. Instrucciones para Chat GPT

Escriba abajo todo el intercambio en el Chat GPT para buscar la solución.

Inicialmente se le pidió que en base al código que se usó en la **Práctica 9 (Parte 1)** para el Python Block, me generará un código para implementar en Google Apps Script con el cual se observará una Web App con la última medición agregada en el Google Sheets por medio de la API de Google Maps, además de pedirle que ciertas especificaciones para la parte del HTML y tener una mejor visualización como resultado, también se pidió cambiar las coordenadas de latitud y longitud, el código generado por GPT es el siguiente:

```
function doGet() {
  const measurement = getLastMeasurement();
  if (!measurement) {
```

```
return HtmlService.createHtmlOutput("No hay datos disponibles.");
  }
  const lat = measurement.latitud;
  const lng = measurement.longitud;
  const description = measurement.descripcion;
  return HtmlService.createHtmlOutput(`
    <!DOCTYPE html>
    <html lang="es">
    <head>
      <meta charset="UTF-8">
                          name="viewport" content="width=device-width,
                   <meta
initial-scale=1.0">
      <title>Última Medición</title>
                                                                   <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC4wgPaAvereqeB455s
LVD1W3fd2CML_oo&callback=initMap" async defer></script>
      <style>
        /* Estilo general del cuerpo */
        body, html {
          margin: 0;
          padding: 0;
          height: 100%;
           font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
/* Tipografía profesional */
          display: flex;
          flex-direction: column;
          align-items: center;
          justify-content: center;
          background-color: #f0f0f0;
        /* Título principal */
        .main-title {
          font-size: 2em;
               color: #004080; /* Color azul oscuro para dar un toque
profesional */
          margin: 20px 0 5px 0;
          font-weight: bold;
        /* Subtitulos */
        .sub-title {
          font-size: 1.2em:
          color: #333;
          margin: 5px 0;
        /* Contenedor principal */
        .container {
          width: 90%;
```

```
max-width: 800px;
          background: #fff;
          padding: 20px;
          box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
          border-radius: 10px;
          text-align: center;
        /* Estilo del mapa */
        #map {
          height: 500px;
          width: 100%;
          margin-top: 20px;
          border-radius: 8px;
        /* Pie de página */
        .footer {
          font-size: 1em;
          color: #666;
          margin-top: 15px;
          font-style: italic;
      </style>
    </head>
    <body>
      <div class="container">
         <h1 class="main-title">Solución IIoT con GNUradio parte2 - Grupo
B</h1>
           <h2 class="sub-title">Realizado por: Diego García (2195533) y
David Florez (2174241)</h2>
        <h3 class="sub-title">IoT - 2024-2</h3>
        <h1>Última Medición</h1>
        <strong>Descripción:</strong> ${description}
        <div id="map"></div>
              <h3 class="footer">Universidad Industrial de Santander -
Ingeniería Electrónica</h3>
      </div>
      <script>
        function initMap() {
          const lat = ${lat};
          const lng = ${lng};
          const map = new google.maps.Map(document.getElementById('map'),
            center: {lat: lat, lng: lng},
            zoom: 15,
               mapTypeId: 'hybrid', // Cambia a 'hybrid' para una mejor
visualización
                     disableDefaultUI: true, // Oculta los controles
predeterminados
```

```
zoomControl: true // Mantén solo el control de zoom
          });
          new google.maps.Marker({
            position: {lat: lat, lng: lng},
           map: map,
            title: 'Última Medición',
              animation: google.maps.Animation.DROP, // Efecto de caída
para el marcador
           icon: {
                                                                     url:
'https://maps.google.com/mapfiles/ms/icons/red-dot.png'
          });
      </script>
    </body>
    </html>
  `);
function getLastMeasurement() {
  const sheetId = '1ZI1p05DJjVnh0ph0jH8sRjnLGqXH0z-jBXJ1ZF875JM';
    const sheet = SpreadsheetApp.openById(sheetId).getSheets()[0]; //
Cambiamos para usar la hoja 1 (índice 0)
  const data = sheet.getDataRange().getValues();
  Logger.log('Total de filas:', data.length); // Imprimir la cantidad de
filas
 if (data.length < 2) return null; // Asegúrate de que hay datos</pre>
 const lastRow = data[data.length - 1]; // Toma la última fila
  Logger.log('Última fila:', lastRow); // Imprime la última fila en el
registro
  const latitud = lastRow[1]; // Latitud en la columna 2
  const longitud = lastRow[2]; // Longitud en la columna 3
  Logger.log('Latitud (sin parsear):', latitud); // Imprime latitud sin
parsear
  Logger.log('Longitud (sin parsear):', longitud); // Imprime longitud
sin parsear
  // Intenta convertir a float y verifica si son válidas
  const parsedLat = parseFloat(latitud);
 const parsedLng = parseFloat(longitud);
    Logger.log('Latitud (parseada):', parsedLat); // Imprime latitud
parseada
   Logger.log('Longitud (parseada):', parsedLng); // Imprime longitud
parseada
```

```
// Asegúrate de que las coordenadas sean válidas
if (isNaN(parsedLat) || isNaN(parsedLng)) {
   return null;
}

return {
   latitud: parsedLat,
   longitud: parsedLng,
   descripcion: lastRow[3] // Tomar la descripción también
};
}
```

Luego de implementar ese código en Google Apps Script, se genera la Web App en la que se observará el mapa con la ubicación de la última medición realizada

Paso 2. Resultados



Se observa que la ubicación no es en la Universidad Industrial de Santander, se cambiaron las coordenadas que tiene la dirección *Carrera 26A #11-30*.

Paso 3. Documentación

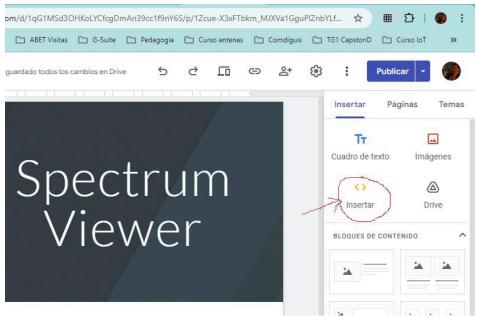
Cree un archivo en Google Docs con la documentación para el webApp. Usa para ello GPT, le puedes pasar la documentación de la WebApp de la tarea 1, para que GPT lo tome como ejemplo

Enlace al archivo de la documentación

Tarea 3. Crear un sitio web para mostrar la solución al público

Paso 1. Crear un Google Sites

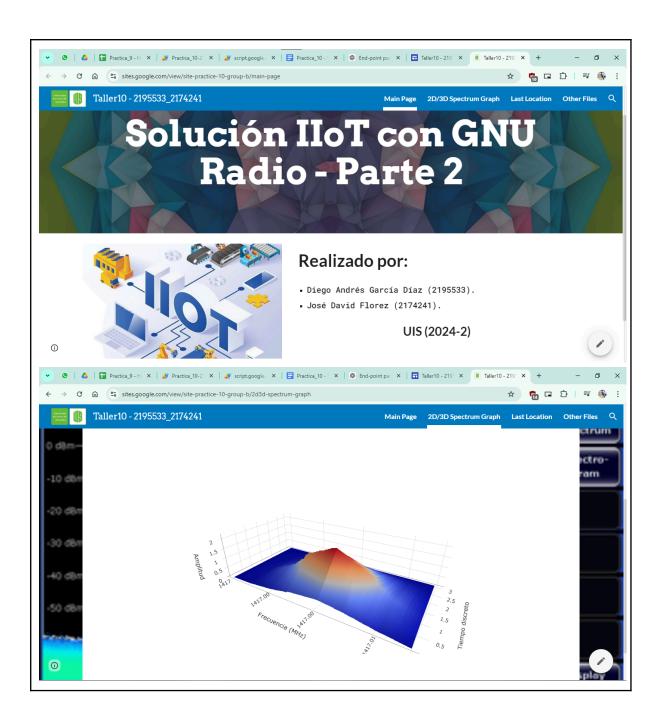
Paso 2. En el google sites ember la WebApp de visualización del espectro en 3D. Este es el proceso: abres el site creado en modo edición > Menú panel izquierdo

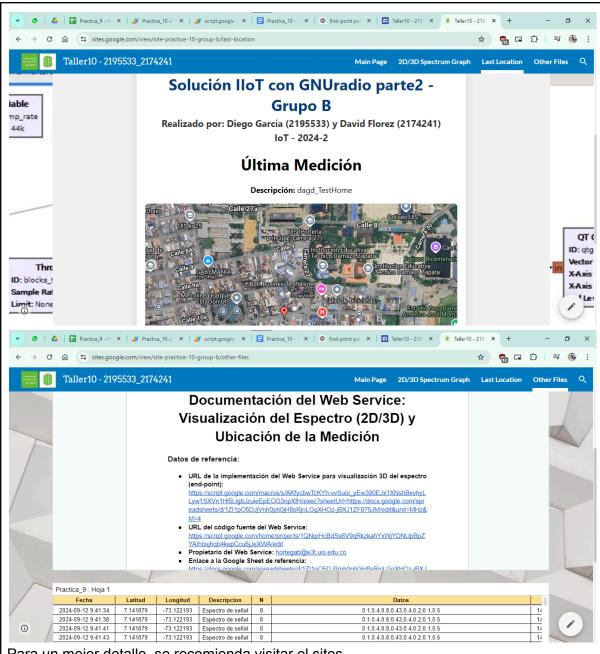


Por URL > escribe el endpoint es decir la URL con la query string

Paso 3. En el google sites ember la WebApp del espectro sobre Google maps. En el mismo site, crean una página adicional y hacen algo similar al paso 2, pero para la aplicación que muestra el registro del espectro en google maps.

Captura de pantalla con evidencias





Para un mejor detalle, se recomienda visitar el sites.

URL del Google Sites: https://sites.google.com/view/site-practice-10-group-b/main-page