

## **Apéndice I. Configuración para Generar Gráficas.**

Este anexo describe cómo configurar el sistema para mostrar gráficas interactivas en tiempo real utilizando Node-RED, basadas en los datos recibidos a través de los protocolos UDP y TCP. Las gráficas representan el espectro de frecuencia (frecuencia frente a amplitud) de los datos recibidos y se actualizan en tiempo real para ofrecer una visualización continua del comportamiento de las señales.

### **1. Visualización Gráfica Interactiva**

El sistema utiliza Plotly JS, una herramienta de visualización interactiva, para mostrar las gráficas en un panel del dashboard de Node-RED. A continuación, se explica cómo se crean y se configuran estas gráficas.

Procesamiento de los Datos, los nodos de Node-RED reciben datos en tiempo real a través de los protocolos UDP o TCP. Estos datos se procesan y se convierten en dos listas: una para las frecuencias (eje X) y otra para las amplitudes (eje Y). Estas listas son luego enviadas a los gráficos correspondientes.

Gráfico UDP, el gráfico UDP muestra los datos del protocolo UDP en un gráfico donde el eje horizontal representa la frecuencia (en GHz) y el eje vertical muestra la amplitud (en dB).

Cada vez que se recibe un nuevo paquete de datos UDP, se procesa la información y se actualiza el gráfico.

## 1.2 Código relacionado con la gráfica UDP.

El gráfico de UDP se genera utilizando el siguiente fragmento de código:

### Figura 1

*Gráfica UDP.*

```
function draw(d){  
  if(!d||!d.x||!d.y) return;  
  last=d;  
  Plotly.newPlot('chart_udp',[{x:d.x,y:d.y,type:'scatter',mode:'lines',line:{color:'blue'}}],  
    [{autosize:true,margin:{t:40,r:40,b:40,l:40},xaxis:{title:'Frecuencia (GHz)'},  
    yAxis:{title:'dB'}}],{responsive:true});  
}
```

*Nota.* Línea de código donde se ajusta la gráfica UDP en la interfaz de Node-RED, solamente afecta la parte gráfica de este protocolo.

En este código:

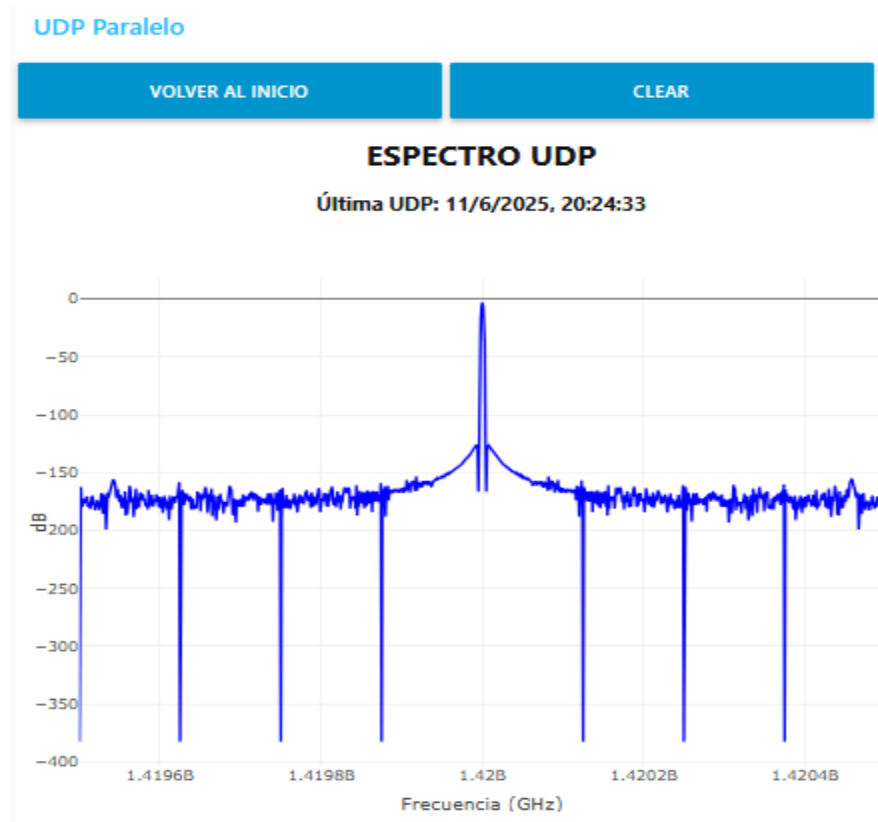
msg.payload.x y msg.payload.y contienen los valores de frecuencia y amplitud.

type: 'scatter' y mode: 'lines' indican que la gráfica es una línea continua.

line: {color: 'blue'} establece el color de la línea de la gráfica UDP como azul.

### Figura 2

*Gráfica UDP vista en la interfaz.*



*Nota.* La imagen corresponde al resultado de la gráfica por protocolo UDP vista en la interfaz, como se mencionó anteriormente en el código da el nombre de los ejes según se eligió (en este caso eje X Frecuencia (GHz) y eje Y dB) y el color que se eligió para el protocolo.

### 1.3 Gráfico TCP

Similar al gráfico de UDP, el gráfico TCP muestra los datos del protocolo TCP con el eje horizontal representando la frecuencia y el eje vertical representando la amplitud.

### Figura 3

*Gráfica TCP.*

```

15 function draw(d){
16     if(!d||!d.x||!d.y) return;
17     last=d;
18     Plotly.newPlot('chart',[{x:d.x,y:d.y,type:'scatter',mode:'lines',line:{color:'red'}}],
19     {autosize:true,margin:{t:40,r:40,b:40,l:40},xaxis:{title:'Frecuencia (GHz)'},
20     ,yaxis:{title:'dB'}},{responsive:true});
21 }

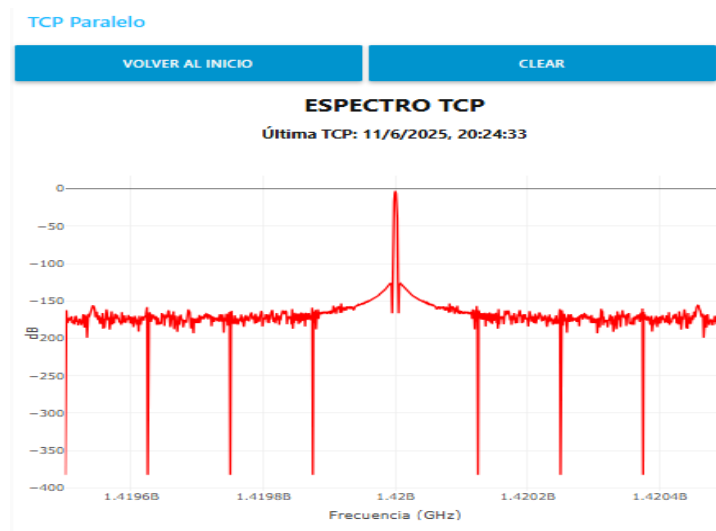
```

*Nota.* El código para crear el gráfico TCP es casi idéntico al de UDP, aquí se le da otro color a la línea de la gráfica de TCP para que cada protocolo tenga su color único.

line: {color: 'red'} cambia el color de la línea del gráfico TCP a rojo.

#### Figura 4

*Gráfica UDP vista en la interfaz.*



*Nota.* La imagen corresponde al resultado de la gráfica por protocolo TCP vista en la interfaz, como se mencionó anteriormente es idéntica a la gráfica del protocolo UDP con la característica del color rojo.

### 1.4 Configuración para Visualización Dual.

El sistema permite ver ambos gráficos al mismo tiempo mediante un botón denominado Dual. Al presionar este botón, se activan ambos gráficos, que se muestran uno al lado del otro para facilitar la comparación.

Gráfico UDP (azul) y Gráfico TCP (rojo), el botón Dual solo organiza las gráficas visualmente, pero cada gráfica sigue siendo independiente y se actualiza con sus propios datos.

Código de la interfaz para el botón "Dual":

La configuración en Node-RED tiene un nodo llamado Navegador, que es el responsable de todo lo que tiene que ver con la interfaz (colores, nombres), en esta configuración le damos el color que va a representar el botón Dual, en este caso el color cian.

#### Figura 5

*Configuración botón Dual.*

```

163 <!-- Botones exteriores -->
164 <button class="boton-animado azul" data-tab="Espectro UDP">UDP</button>
165 <button class="boton-animado cian" data-tab="Espectros Paralelos">Dual</button>
166 <button class="boton-animado rojo" data-tab="Espectro TCP">TCP</button>
167 <button class="boton-animado morado" data-tab="Appscript ">Appscript</button>
168 <button class="boton-animado naranja" data-tab="TABLA TCP">Tabla TCP</button>
169 <button class="boton-animado turquesa" data-tab="Tabla Appscript">Tabla Appscript</button>
170 <button class="boton-animado verde" data-tab="TABLA UDP">Tabla UDP</button>
171 </div>

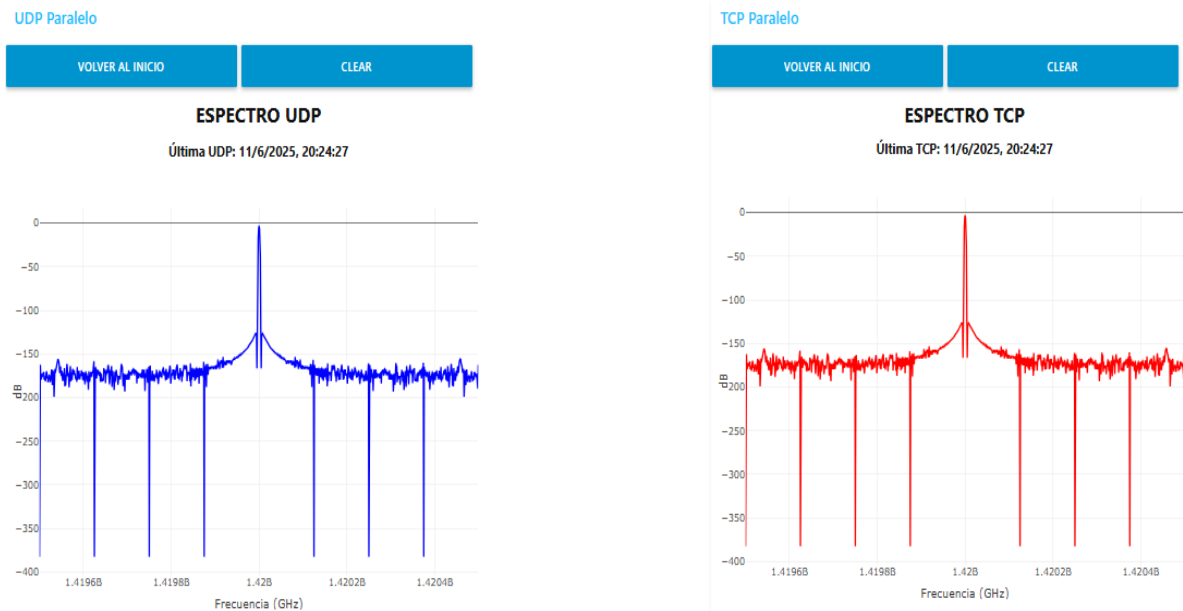
```

Nota. La imagen corresponde a las líneas código que se les proporciona a cada botón, en este caso nos interesa el color del botón Dual.

Al presionar el botón, se activan ambos gráficos.

#### Figura 6

*Visualización de gráficas después de apretar el botón Dual.*



*Nota.* Al apretar el botón Dual vemos ambas gráficas recibiendo información al tiempo, este botón Dual existe para comparación de una misma información por protocolo.

Actualización en Tiempo Real: Ambos gráficos se actualizan en tiempo real con los datos correspondientes, pero no están sincronizados en cuanto a la frecuencia o el tiempo. Las gráficas permanecen estáticas cuando no reciben nuevos datos.

### 1.5 Gestión de la Hora del Último Paquete Recibido.

Cada gráfica muestra la hora del último paquete recibido, lo que permite al usuario saber exactamente cuándo se recibió el último dato.

### 1.6 Código para la hora del último paquete

Su función es la de registrar el último dato ingresado y que en la interfaz pueda ser visto por cualquier persona, esto es muy útil ya que con esto podemos de manera visible ver que el sistema está funcionando bien.

## Figura 7

*Actualización del último registro del paquete recibido.*

```

17 | const lbl=document.getElementById('hora-udp-paralelo');
18 | function setTime(t){ if(lbl) lbl.textContent='Última UDP: '+t; }
19 | const saved=localStorage.getItem('hora-udp-paralelo'); if(saved) setTime(saved);

```

*Nota.* Este código toma la hora del último paquete recibido y la guarda en el localStorage del navegador. Esto asegura que la hora se mantenga visible incluso si el usuario recarga la página o cambia de sección en la interfaz.

### 1.7 Redimensionado de las Gráficas

Las gráficas se adaptan automáticamente al tamaño de la ventana del navegador. Si el usuario cambia el tamaño de la ventana, las gráficas se redimensionan de forma adecuada para seguir siendo legibles y claras.

#### Figura 8

*Código redimensionado.*

```

:
27 | if(msg.topic==='ui_control'){setTimeout(()=>
28 | [Plotly.Plots.resize(document.getElementById('chart_udp'))];],300);return;}
--

```

*Nota.* Esta línea de código asegura que las gráficas se ajusten automáticamente al tamaño de la ventana del navegador sin perder claridad.

### 1.8 Estilos de Plotly

Las gráficas generadas utilizan los estilos predeterminados de Plotly JS, que proporcionan una presentación clara y profesional.

Estilos de las gráficas:

Título: El gráfico de UDP tiene el título "ESPECTRO UDP" en color azul, y el gráfico de TCP tiene el título "ESPECTRO TCP" en color rojo.

Ejes: Los ejes de cada gráfico están etiquetados con "Frecuencia (GHz)" en el eje horizontal y "Amplitud (dB)" en el eje vertical.

Colores: Las líneas en el gráfico de UDP son de color azul y en el gráfico de TCP son de color rojo, lo que facilita la distinción visual entre ambos protocolos.

Este sistema permite una visualización clara y continua del espectro de frecuencia de los datos UDP y TCP en tiempo real, con la posibilidad de visualizar ambos gráficos simultáneamente para facilitar su comparación. Las gráficas son interactivas, redimensionables y cuentan con un mecanismo para mostrar la hora del último paquete recibido, asegurando que el usuario siempre esté al tanto de la información más reciente sin perder visibilidad de los datos históricos.

El uso de Plotly JS para la creación de gráficas y la configuración detallada de la interfaz permite que los usuarios interactúen de manera eficiente y fácil con los datos, sin necesidad de conocimientos técnicos avanzados.