Medición del Campo Electromagnético Con SDR (CoNAE)

Proyecto Final Comunicaciones Digitales

La Banda Ancha de Amigos

FCEFyN
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas, Físicas y Naturales

11 de mayo de 2023

Contenidos

Introducción

Estas filminas corresponden a la primera presentación sobre el proyecto de medición de campo electromagnético en el campo de CoNAE. Este proyecto está siendo desarrollado el grupo llamado "La banda ancha de amigos", cuyos integrantes son:

- Cabrera Augusto Gabriel
- Mansilla Josías Leonel
- Moroz Esteban Mauricio
- Villar Federico Ignacio

Idea del proyecto

La idea principal de este proyecto es armar un mapa de calor que muestre la potencia del campo electromagnético de una cierta frecuencia. Esta frecuencia es la de trabajo de los radares de CoNAE. Se pretende obtener algo similar a lo que se muestra en la imagen a continuación.

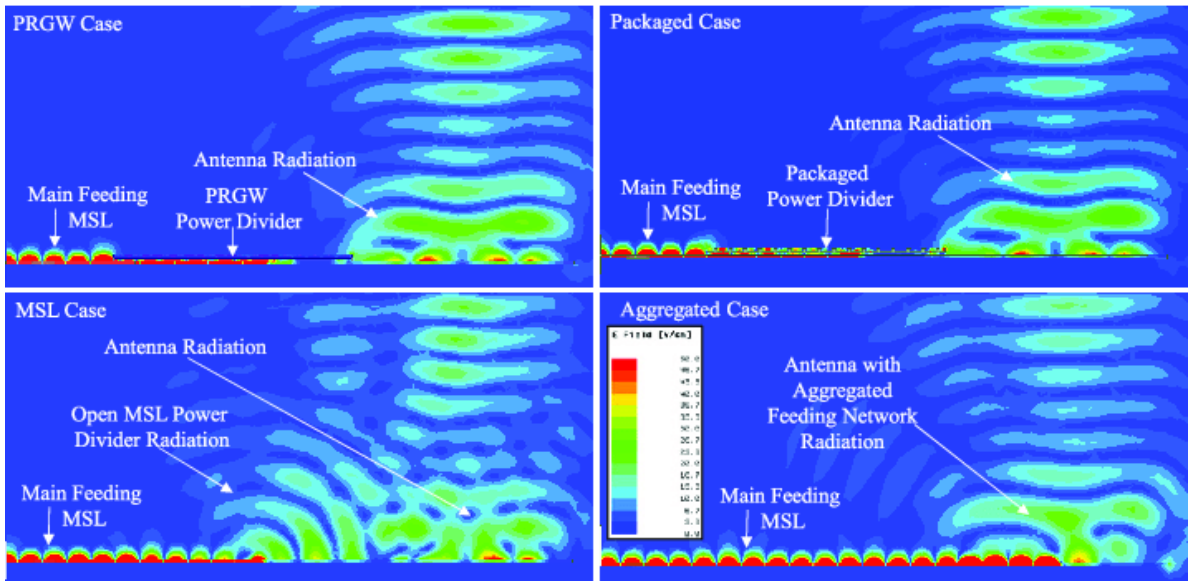


Figura 1: Ejemplo de mapa de calor.

Herramientas de hardware

Las herramientas de hardware que se utilizarán son:

- ADALM-PLUTO SDR
- Drone DJI Matrice 300 RTK

Figura 2: SDR a utilizar.

Herramientas de software

Para el software del proyecto a implementar, se trabaja con:

- Visual Studio Code: para programar en Python y poder realizar las pruebas del SDR.
- DJI Pilot: para obtener los datos del vuelo del drone.
- GitHub: para trabajar en la programación de forma colaborativa.

Idea a implementar

Se puede desarrollar la idea mediante una serie de pasos:

1. Se coloca el SDR en el drone y se sobrevuela la zona a analizar.
2. El SDR toma muestras de la señal centrada en la frecuencia dictaminada por CoNAE y las almacena.
3. Una vez finalizado el vuelo, se procede a obtener el registro de vuelo del drone.
4. Con el registro de vuelo, se obtiene una base de datos en donde se registran las coordenadas del drone en determinado momento y las muestras que el SDR tomó en ese instante.
5. Se procesa la información de las muestras, obteniendo diferentes parámetros de interes, como: PSD, potencia de la señal, espectrogramas, etc.
6. Con la información procesada, se crea un mapa interactivo que posee todos los datos obtenidos.

Programación

La programación del SDR se realizó enteramente en Python. Para ello se utilizaron las librerías:

- Numpy
- Matplotlib
- Scipy
- Geopandas
- Pandas
- pyrtlsdr

Configuración del SDR en Python

El dispositivo RTL SDR debe configurarse para cada programa. Por ello, al inicio de cada uno se coloca lo siguiente.

Código 1: Configuración del RTL-SDR.

```
1  sdr = RtlSdr()
2  sdr.sample_rate = 2.4e6
3  sdr.center_freq = 100e6
4  sdr.gain = 'auto'
5  samples = sdr.read_samples(256*1024)
```

Luego, a partir de la lista que posee las muestras (**samples**), se realizan diferentes análisis, que serán mostrados posteriormente en la sección de los primeros resultados.
Es importante que luego de realizar las mediciones y el procesado de la señal, se cierre el objeto SDR, por lo que se escribe:

Código 2: Cierre del objeto RTL-SDR.

```
1  sdr.close()
```

Primeras pruebas

Se realizaron pruebas con un RTLSDR de hobbie, con ese dispositivo, se obtuvieron distintas gráficas de utilidad, que podrían usarse en el proyecto final, siempre adecuandose a los requerimientos impuestos por CoNAE. Los códigos se ejecutaron dentro de Visual Studio Code haciendo uso de Code Runner. En las filminas siguientes se muestran algunos ejemplos de las gráficas obtenidas. Cabe destacar que en la mayoría de las ocasiones se trabajó con una frecuencia central de $100\ MHz$.

Transformada rápida de Fourier FFT

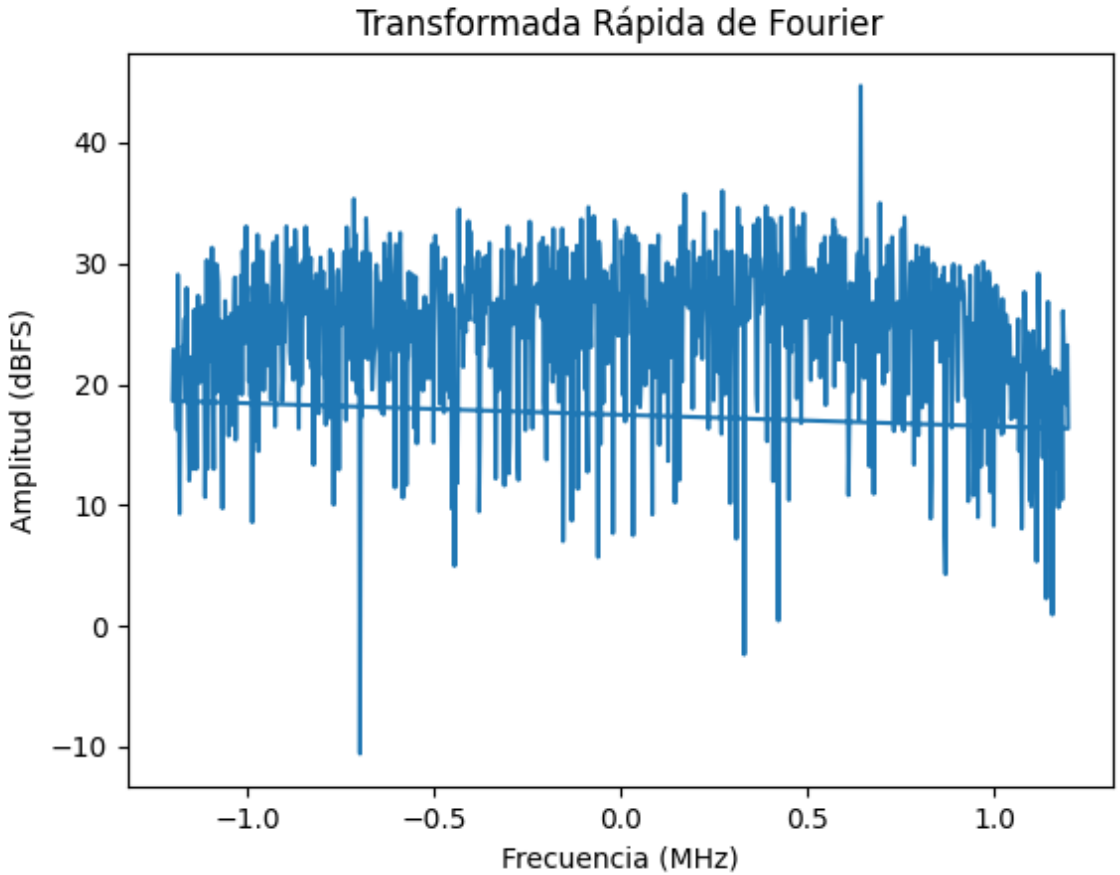


Figura 3: FFT a partir de muestras del SDR.

Transformada rápida de Fourier FFT (continuación)

El código a continuación es el que se utilizó para obtener el gráfico de la filmina anterior. Se hace uso de Numpy y Matplotlib.

Código 3: Gráfico de la FFT.

```
fft = np.fft.fft(samples)
freq = np.fft.fftfreq(len(samples), 1/sdr.sample_rate)
plt.plot(freq/1e6, 20*np.log10(np.abs(fft)))
plt.xlabel('Frecuencia (MHz)')
plt.ylabel('Amplitud (dBFS)')
plt.show()
```

Diagrama de Constelaciones para BPSK

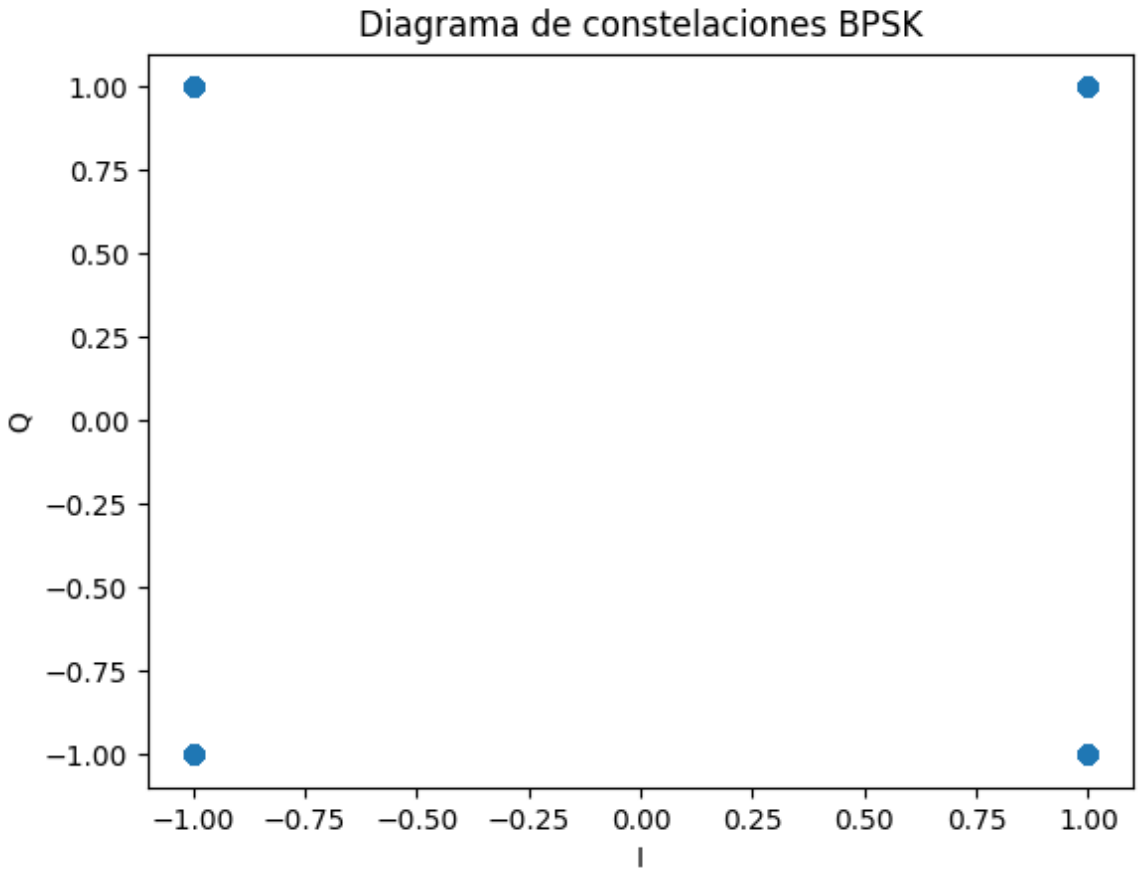


Figura 4: Diagrama de constelaciones en BPSK.

Diagrama de Constelaciones para BPSK (continuación)

Código 4: Obtención del diagrama de constelaciones para BPSK.

```
bits = np.real(samples) > 0
bits = bits.astype(int)
bits = 2*bits - 1
symb = np.zeros(len(bits)//2, dtype=complex)
symb.real = bits[::2]
symb.imag = bits[1::2]
plt.scatter(symb.real, symb.imag)
plt.xlabel('I')
plt.ylabel('Q')
plt.title('Diagrama de constelaciones BPSK')
plt.show()
```

Espectrograma

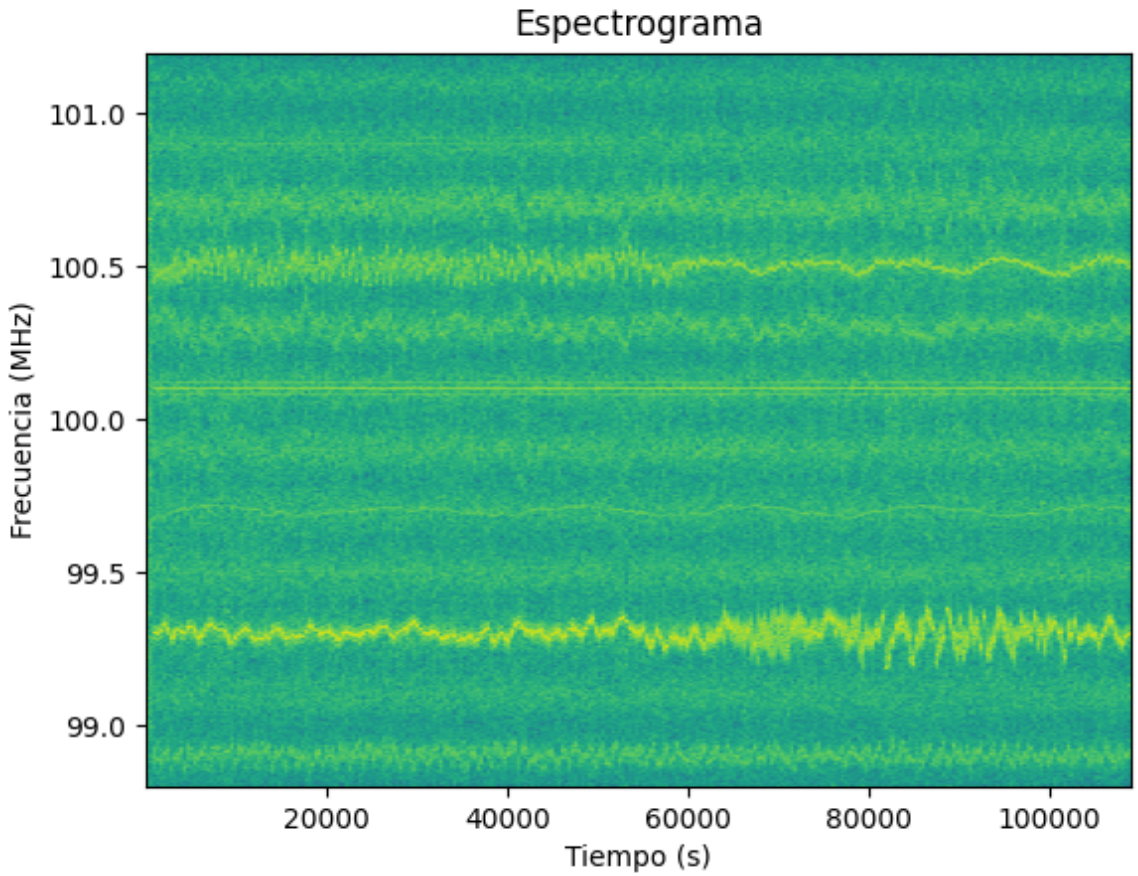


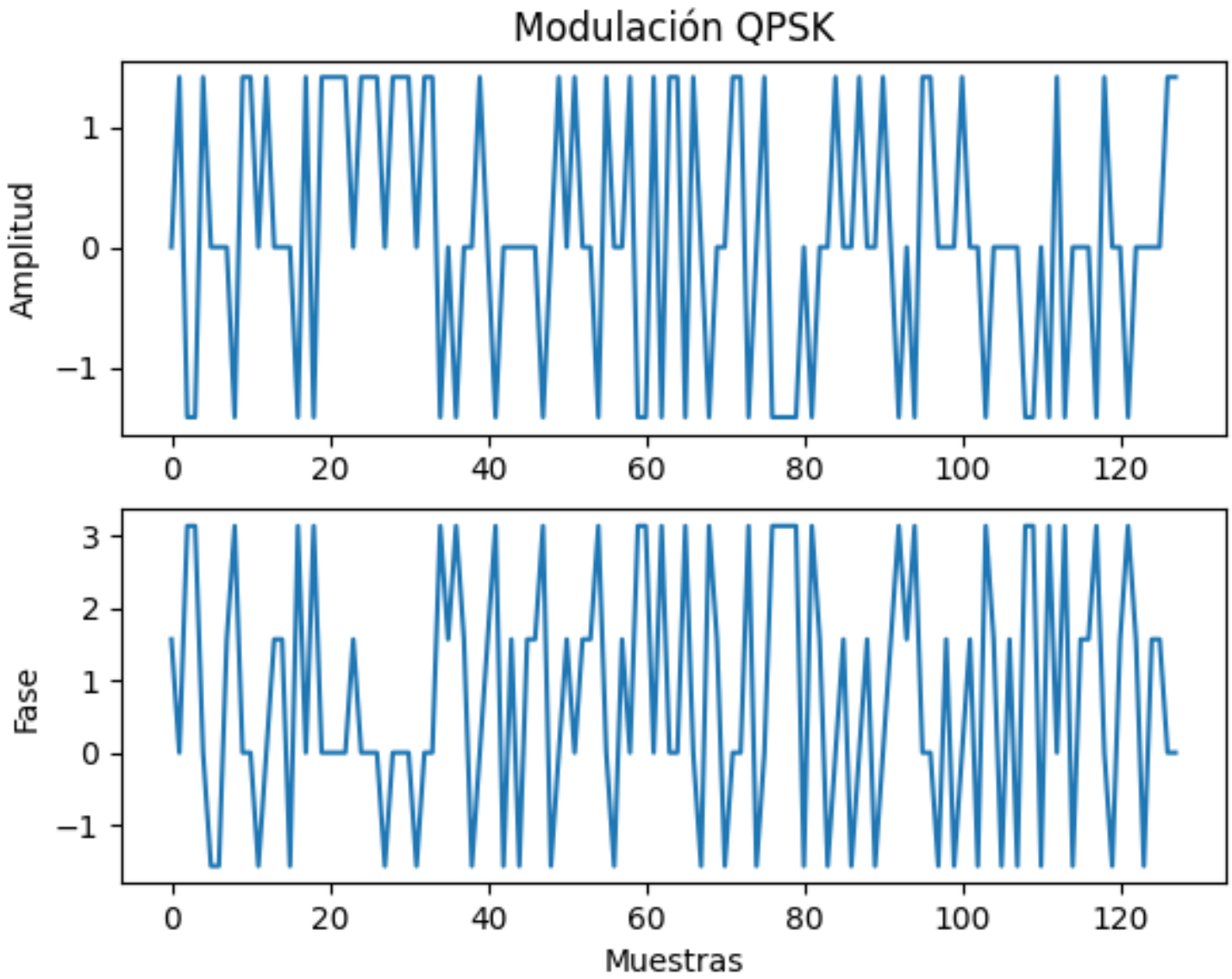
Figura 5: Espectrograma.

Espectrograma (continuación)

Código 5: Obtención del espectrograma.

```
plt.specgram(samples, NFFT=1024, Fs=sdr.sample_rate/1e6, Fc=sdr.center_freq/1e6
    ↪ )
plt.xlabel('Tiempo (s)')
plt.ylabel('Frecuencia (MHz)')
plt.show()
```

Modulación QPSK



Modulación QPSK (continuación)

Código 6: Obtención del gráfico de modulación QPSK.

```
bits = np.real(samples) > 0
bits = bits.astype(int)
bits = 2*bits - 1
symb = np.zeros(len(bits)//2, dtype=complex)
symb.real = (bits[::2] + bits[1::2])/np.sqrt(2)
symb.imag = (bits[::2] - bits[1::2])/np.sqrt(2)
fig, axs = plt.subplots(2, 1)
axs[0].plot(symb.real)
axs[0].set_ylabel('Amplitud')
axs[0].set_title('Modulación QPSK')
axs[1].plot(np.angle(symb))
axs[1].set_ylabel('Fase')
axs[1].set_xlabel('Muestras')
```

Ejemplo de mapa de calor

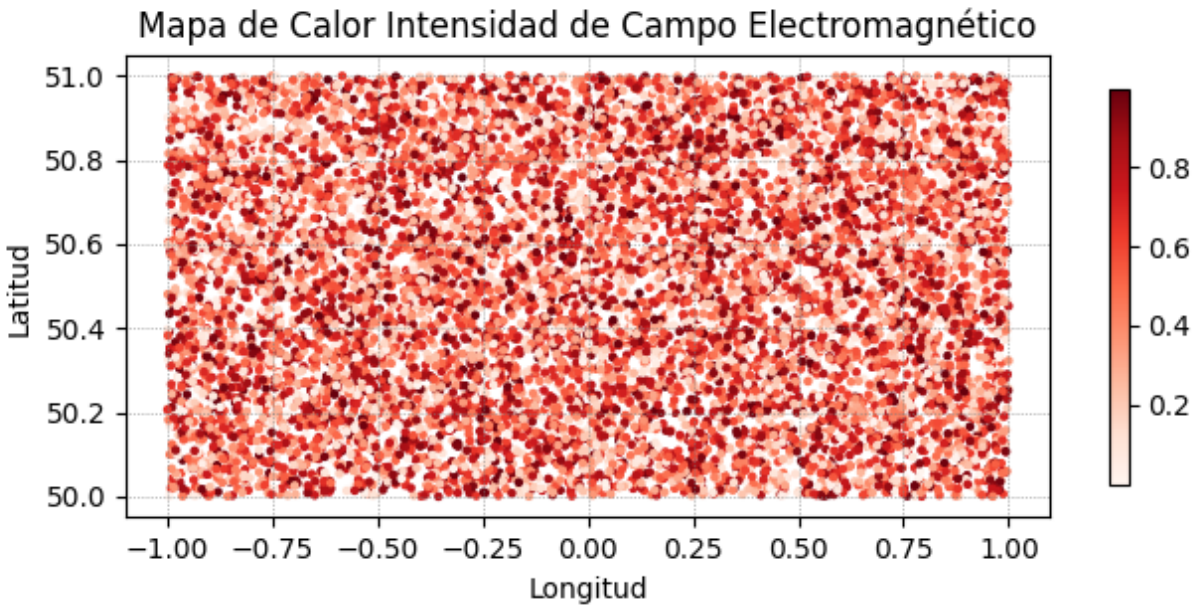


Figura 7: HeatMap con Geopandas.

Ejemplo mapa de calor (continuación)

En el código a continuación se obtienen de forma aleatoria valores de latitud y longitud. Y, a cada par de estos se les asigna un valor aleatorio de campo electromagnético asociado, de forma de generar un mapa de calor a modo ilustrativo con el uso del paquete Geopandas.

Código 7: Ejemplo de mapa de calor.

```
n_points = 10000
lat = np.random.uniform(50.0, 51.0, n_points)
lon = np.random.uniform(-1.0, 1.0, n_points)
geometry = [Point(xy) for xy in zip(lon, lat)]
df = gpd.GeoDataFrame({'intensidad': np.random.rand(n_points)}, geometry=
    ↪ geometry)
fig, ax = plt.subplots(figsize=(10, 10))
df.plot(column='intensidad', cmap='Reds', markersize=5, alpha=1, legend=True,
      ax=ax, legend_kwds={'shrink': 0.5})
ax.grid(True, linestyle=':', linewidth=0.5, color='gray')
ax.set_xlabel('Longitud')
ax.set_ylabel('Latitud')
ax.set_title('Mapa de Calor Intensidad de Campo Electromagnético')
plt.show()
```

Medición de la potencia del campo electromagnético

Para medir la intensidad del campo electromagnético a una determinada frecuencia, se plantea la medición de la potencia de la señal. Partiendo de la definición de la misma, se obtiene una potencia en dBFS (decibelios a escala completa) haciendo uso de Numpy.

Código 8: Medición de la potencia.

```
power = np.abs(samples)**2
power_db = 10*np.log10(np.mean(power))
print('Intensidad de campo electromagnético: %0.2f dBFS' % power_db)
```

El código anterior mostrará así entonces una estimación de la potencia de la señal. Este valor puede almacenarse en un dataFrame para luego ser mostrado en el mapa de calor.

Propuestas para mejoras

Se tienen diferentes propuestas para mejorar el proyecto:

- Debugger en tiempo real para medir la radiación de campo electromagnético a distancia durante el vuelo del drone.
- Mapa interactivo con Javascript.
- Obtención de los datos de vuelo con DJI Assitant.
- Trazado de la ruta de vuelo sobre un mapa con imágenes satelitales.