



## **Tutorial de la Recepción y Reproducción del Barrido de la señal GPS con GNU Radio**

**Jhon Alexander García Sierra**  
*Ingeniería de Sistemas y Telecomunicaciones*  
*Universidad Sergio Arboleda - Bogotá, Colombia*

**Juan Felipe Osorio Tellez**  
*Ingeniería de Sistemas y Telecomunicaciones*  
*Universidad Sergio Arboleda - Bogotá, Colombia*

**Directores:**  
**Ing. Juan Manuel Aranda Lopez King, Ph.D.(C)**  
**Ing. Marco Tulio Teran de la Hoz, Ph.D.(C)**

### **Resumen**

Para este proyecto la versión con la que se está trabajando GNU Radio es la 3.8.1.0. Para entender el tutorial se necesita tener conocimientos en Python, puesto que se verán temas de tipos de datos, ciclos y condicionales, la señal GPS que se obtenga del GNSS-SDR se guardará en un archivo .dat y se reproducirá en el software de GNU Radio para posteriormente utilizar la Adquisición con Matlab.

### **Palabras clave:**

**Tutorial, GNU Radio, GNSS-SDR, MATLAB, Posicionamiento GPS**

# Índice

<b>1.</b>	<b>Introducción</b>	<b>4</b>
<b>2.</b>	<b>Marco teórico</b>	<b>4</b>
2.1.	Software Utilizado . . . . .	4
2.1.1.	GNU Radio . . . . .	4
2.1.2.	GNSS-SDR . . . . .	4
2.1.3.	MATLAB . . . . .	4
2.2.	Hardware Utilizado . . . . .	4
2.2.1.	Antena GPS . . . . .	4
2.2.2.	HackRF One . . . . .	4
2.2.3.	Nooelec Tiny TCX0 . . . . .	4
<b>3.</b>	<b>Procedimiento:</b>	<b>4</b>
3.1.	Instalación del Software: . . . . .	4
3.1.1.	Instalación del GNSS-SDR y GNU Radio . . . . .	4
3.1.2.	Instalación de MATLAB . . . . .	4
3.2.	Introducción a GNU Radio: . . . . .	5
3.2.1.	Partes de una interfaz de GNU Radio: . . . . .	5
3.2.2.	Forma de buscar un bloque en GNU: . . . . .	5
3.2.3.	Tipos de datos . . . . .	6
3.2.4.	Modificar los parámetros del bloque Options: . . . . .	6
3.2.5.	Primer diagrama de Flujo . . . . .	7
3.2.6.	Barrido FM en GNU Radio . . . . .	8
3.3.	Obtención del Posicionamiento . . . . .	16
3.3.1.	Esquemático del GNSS-SDR . . . . .	16
3.3.2.	Código De Recepción: . . . . .	20
3.3.3.	Código de Reproducción: . . . . .	23
3.4.	Introducción a MATLAB . . . . .	25
3.4.1.	Partes del Entorno en MATLAB . . . . .	25
	<b>Referencias</b>	<b>31</b>

## 1. Introducción

## 2. Marco teórico

### 2.1. Software Utilizado

**2.1.1. GNU Radio.** Software de radio libre y de código abierto basado en Python en cuanto a su aplicación mientras que para su procesamiento se utiliza el lenguaje C++, este software es utilizado para unir el hardware con un entorno programable, suplantando algunas de las características físicas necesarias para la recepción y transmisión de señales análogas y digitales gracias a su amplia colección de bloques de procesamiento. GNU Radio es un conjunto de herramientas de software gratuito utilizado por aficionados, profesionales y estudiantes para la creación de radios y manipulación de la señal. [1], [2]

**2.1.2. GNSS-SDR.** Proyecto de código abierto que implementa un receptor definido por software de sistema global de navegación por satélite en C++, este software proporciona una interfaz para diferentes interfaces de Radio Frecuencia (FM) e implementa la solución resultante de la navegación. Su diseño le permite al usuario una edición completa de las fuentes de la señal, algoritmos de procesamiento, interoperatividad con otros sistemas, formatos de salida y ofrece interfaces para todas las señales intermedias, parámetros y variables. [3]

**2.1.3. MATLAB.** Plataforma de programación y de cálculo numérico con la capacidad de analizar datos, desarrollar algoritmos y crear modelos. Millones de ingenieros y científicos emplean MATLAB para manejar sistemas de control, Deep Learning, procesamiento de imágenes y visión artificial, Machine Learning, Mantenimiento predictivo, robótica, procesamiento de señales, prueba y medición, comunicaciones inalámbricas, entre otras. [7]

### 2.2. Hardware Utilizado

**2.2.1. Antena GPS.** Las antenas GPS permiten la aplicación de sistemas de navegación por satélite (GNSS) permitiendo un rastreo y localización de la ubicación de cualquier objeto que se encuentre en el rango de los satélites. Estas antenas captan las señales de banda L transmitidas del espacio y las transfieren a una unidad de procesamiento para determinar la ubicación de los respectivos receptores. [8]

**2.2.2. HackRF One.** Es un dispositivo auxiliar e independiente utilizado por SDR para transmitir o recibir señales de radio desde 1 MHz hasta 6 GHz. Diseñado para facilitar el desarrollo de las tecnologías de la comunicación tanto actuales como en el desarrollo de las nuevas generaciones de tecnologías de radio junto con sus correspondientes protocolos de comunicación [9].

**2.2.3. Nooelec Tiny TCX0.** Complemento de 10 MHz para HackRF, este mide 0,58" x 0,4". Tiene la capacidad establecer un ruido de fase ultra bajo y una estabilidad de frecuencia en casi cualquier condición. Este es un excelente complemento para realizar experimentaciones con alta precisión con el HackRF en cuanto a proyectos relaciones con el GPS. [16]

## 3. Procedimiento:

### 3.1. Instalación del Software:

**3.1.1. Instalación del GNSS-SDR y GNU Radio.** Tanto el software del GNSS-SDR como el de GNU Radio se pueden instalar con el siguiente comando a partir del Sistema Operativo (SO) de Debian 9 o Ubuntu 16.04 Linux:

```
$ sudo apt-get install gnss-sdr
```

EL bloque Osmocom y bloque RTL-SDR no están instalados por defecto en el GNU Radio, el proceso para instalarlos es el siguiente en la terminal de Linux: [17]:

```
$ sudo apt-get install libportaudio2 → $ sudo apt-get install git-core → $ sudo apt-get install gnuradio gr-somosdr
```

**3.1.2. Instalación de MATLAB.** : Este software es compatible tanto para el sistema operativo de Linux y Windows, este se puede descargar mediante el siguiente enlace: [Link](#)

## 3.2. Introducción a GNU Radio:

### 3.2.1. Partes de una interfaz de GNU Radio:

- **Título** : Denominación del proyecto actual.
- **Menú del proyecto** : Muestra las acciones que se pueden, editar o modificar del proyecto en general.
- **Barra de herramientas** : Contiene las acciones que se pueden realizar con el GRC, como lo sería el guardado, la eliminación de las conexiones, la ejecución del espacio del trabajo, generar el .py, entre otras.
- **Espacio de trabajo** : Lugar donde se ubican todos los bloques y las conexiones que estos pueden realizar.
- **Terminal** : Muestra la ejecución del .grc, la existencia de algún error, lugar donde se puede ver la impresión de la función print de un bloque.
- **Biblioteca** : Contiene todos los bloques que se encuentren instalados actualmente en el GNU Radio.



Figura 1. Entorno de GNU Radio

**3.2.2. Forma de buscar un bloque en GNU:** En caso de necesitar un bloque y no saber la carpeta en donde este se encuentra ubicado se debe escribir el nombre del bloque en el icono de la lupa en la barra de herramientas, otra forma en la que el usuario pueda buscar el bloque es abrir categoría por categoría como se puede ver en la imagen 40

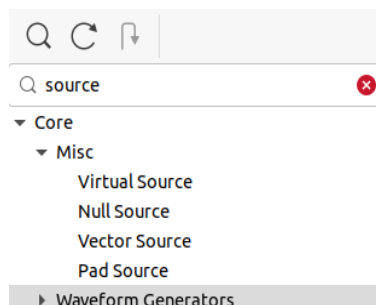


Figura 2. Lista de Bloques de GNU Radio

**3.2.3. Tipos de datos.** Los bloques de GNU cuentan 5 tipos de variables, entre ellas están **complex**, **float**, **int**, **short** y **byte**. No obstante la versión del GNU Radio puede variar el procesamiento de los bloques en cuanto a la tasa de información que pueda manejar.

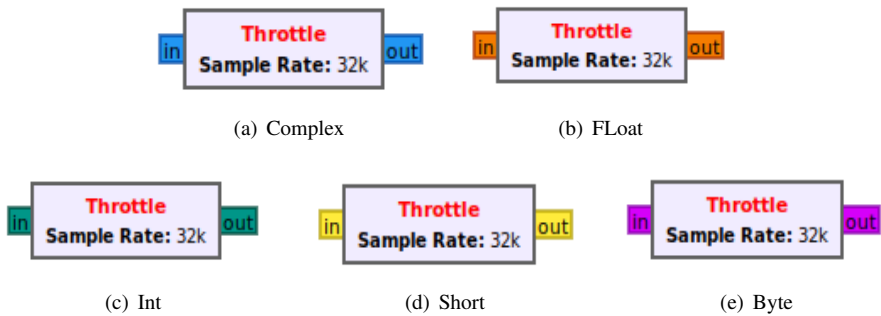


Figura 3. Tipos de datos con el bloque Throttles

**3.2.4. Modificar los parámetros del bloque Options:.** El bloque de Options define el comportamiento del archivo de GNU Radio (.grc) y la forma en la que se ejecuta.

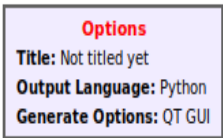


Figura 4. Bloque Options

Al ingresar en en el bloque de la figura 4, aparecerá la siguiente ventana emergente:

Figura 5. Configuración del Bloque Options

- **ID:** Cada bloque por lo general tiende a tener un ID diferente, el cual reconocerá el archivo .py que generará el diagrama de bloques.
- **Title, author y description:** Parámetros que identifica al bloque en el .grc.
- **Canvas size:** Modifica las dimensiones del editor de diagrama de flujo, estos parámetros puede variar entre (300, 300) para width y (4096, 4096) para height.
- **Generate Options:** Tipo de la interfaz gráfica que tendrá el diagrama, estos son: [6]
  - **QT GUI:** Utilización de elementos gráficos de la biblioteca en la categoría QT.
  - **NO GUI:** No genera ninguna interfaz gráfica en el momento de la ejecución con lo cual no se podrán utilizar elementos de la categoría GUI Widgets.
  - **Hier Blocks:** Bloques usado como contenedores para que la simplificación de varios bloques de GNU se puedan crear como un unico bloque. [5]

Antes de realizar algún cambio en los parámetros se debe revisar la documentación del bloque con el fin de tener los resultados deseados, se puede dar el caso que los bloques no contengan la información necesaria para su uso con lo cual se le recomienda al usuario ingresar al link que aparecerá adjunto en Documentation:

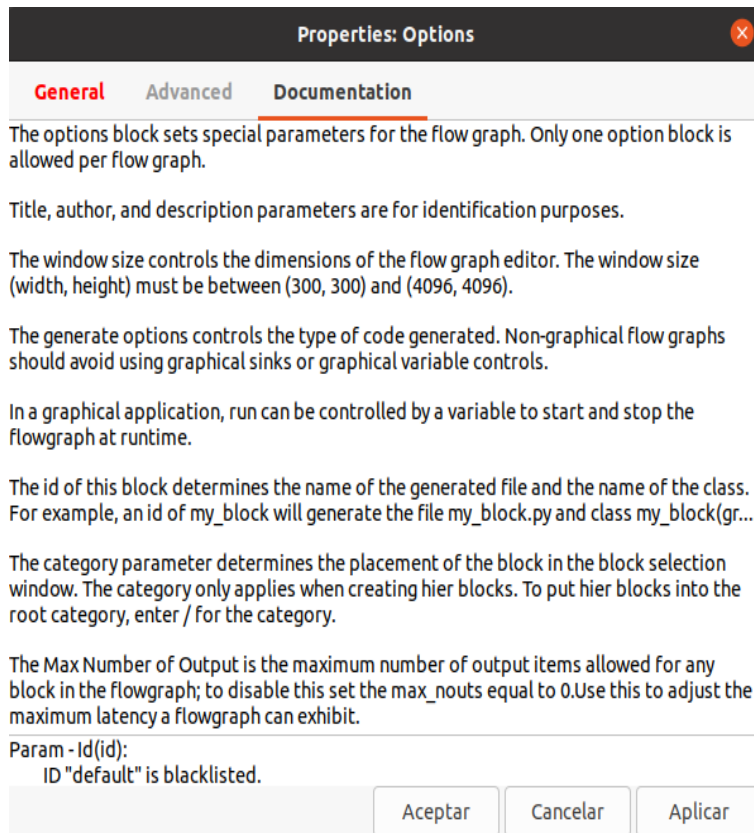


Figura 6. Documentación del BLOque Options

**3.2.5. Primer diagrama de Flujo.** El ID y el Title del archivo .grc **Options** puede ser modificado a gusto del usuario, mientras que el resto de las opciones se pueden dejar por defecto. Los bloques **Signal Source**, **Throttle** y **QT GUI Time Sink** y se arrastran al espacio de trabajo, para que estos se puedan unir se hace clic en el puerto de colores del tipo de dato y se vuelve hacer clic al bloque que se desea conectar, se debería tener algo parecido al Figura 7. [6]

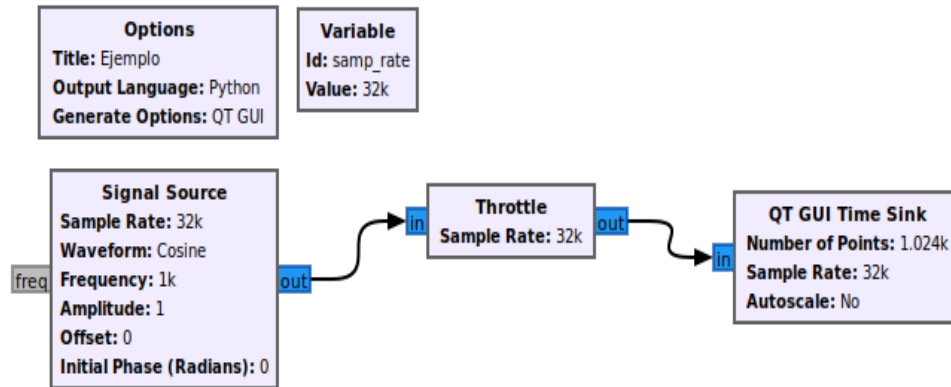


Figura 7. Primer Diagrama de Flujo

Para ejecutar el GNU Radio es necesario guardar previamente el archivo .grc.



Figura 8. Opciones de Ejecución GRC

Al ejecutar el archivo .grc genera un código fuente de .py el cual se guardara en la misma carpeta en donde se encuentra el archivo .grc, se visualizará la siguiente Figura 9

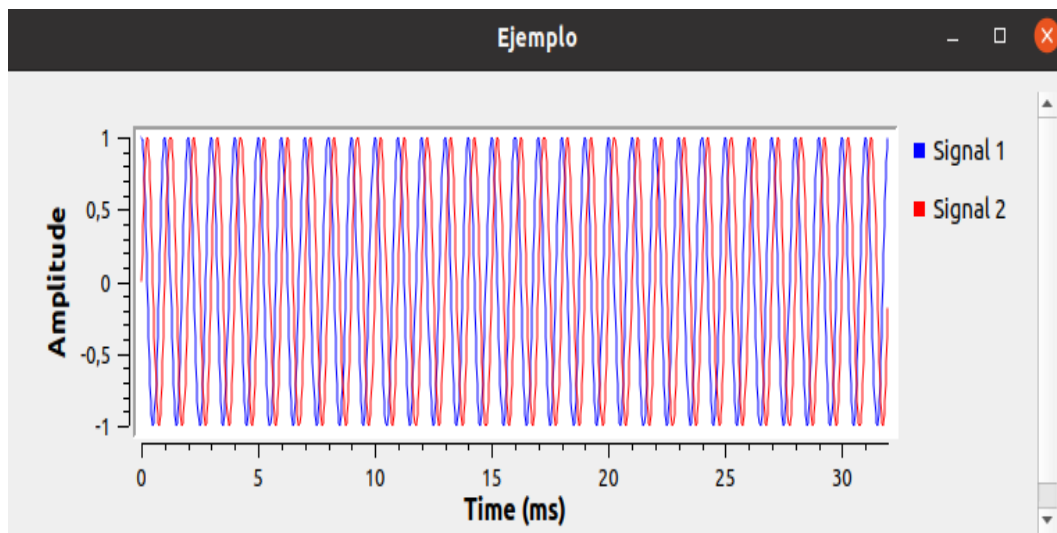


Figura 9. Ejecución del primer diagrama de Flujo

**3.2.6. Barrido FM en GNU Radio.** Para obtener el diagrama de bloques de la Figura 32, se deben realizar los siguientes pasos (para tener una mejor visualización del espectro de la señal captada por la antena se realizarán los pasos 1 y 2, sin embargo estos pasos pueden ser omitidos por el usuario si así lo desea a excepción del bloque **Osmocom Source**): [12]

1. Ingresar en el espacio de trabajo el bloque **Osmocom Source**, **QT GUI Waterfall Sink**, **QT GUI Frequency Sink**. Los dos bloques QT deberán estar conectados con el bloque **Osmocom Source**.



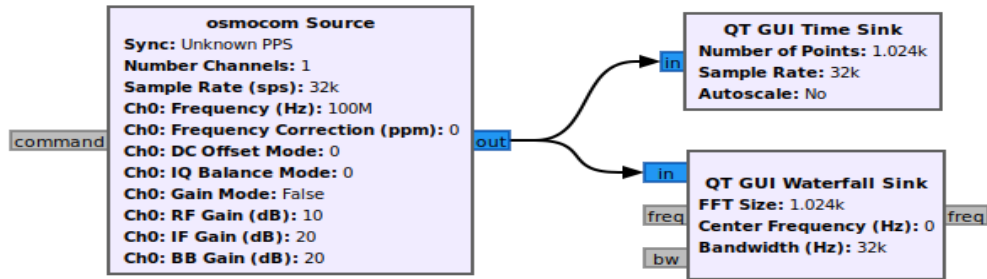


Figura 10. Diagrama de Bloques para la visualización del Espectro

2. Agregar el bloque **Band Pass Filter**, **QT Gui Time Sink**, **QT GUI Frequency Sink**, estos bloques QT estarán conectados al **Band Pass Filter** y a la vez este se conectará al **Osmocom Source**

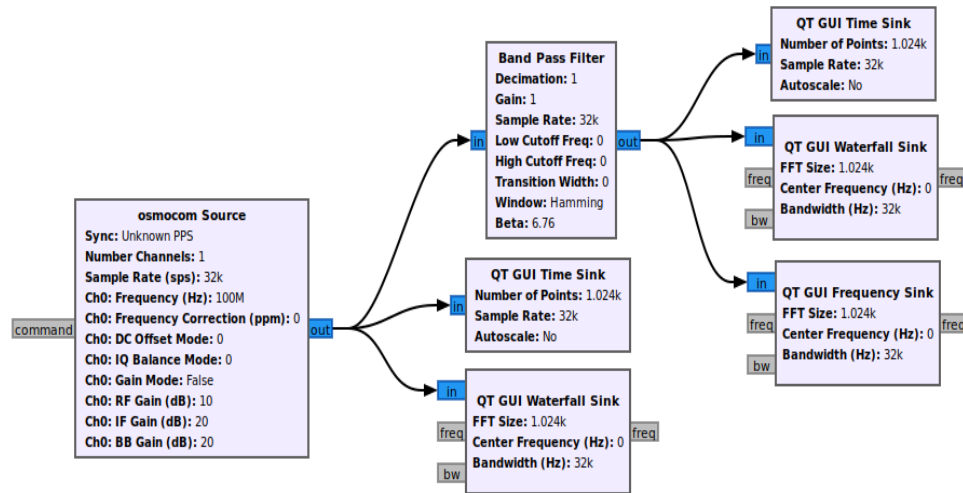


Figura 11. Diagrama de Bloques para la visualización del Espectro junto con el Filtro pasa Banda

3. Para escuchar la señal FM se captada por el HackRF One se agregarán los bloques **Rational Resampler**, **Simple Squelch**, **WBFM Receive**, **Rational Resampler**, **Multiply Const**, **Audio Sink**, para mayor facilidad visual del lector los bloques del paso 1 y 2 no se tendrán en cuenta a excepción del bloque **Osmocom Source**

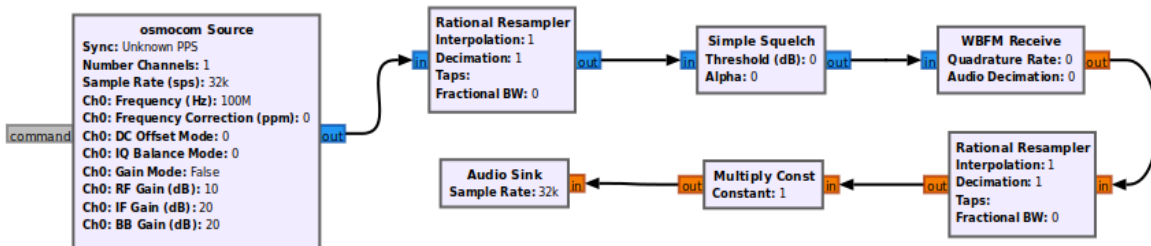
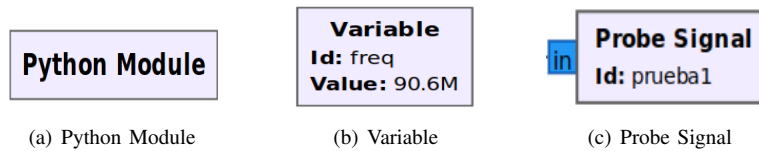


Figura 12. Creación de los bloques de Sweeper Paso 3

4. Agregar el bloque **Python Module**, **Variable**, **Probe Signal**. El bloque **Python Module** le permite al usuario incrustar un módulo de Python en el entorno de trabajo del archivo .grc, no obstante en esta versión carece del ID, la solución de este problema se verá en el siguiente paso [13], el bloque **Variable**, este bloque asigna un valor

único el cual puede ser usado como parámetro para otros bloques [14], el bloque **Probe Signal** tiene la función de tomar muestras de flujo y escribirlas en una variable, este bloque trabaja conjuntamente con el bloque **Function Probe** [15]:



El bloque **Probe Signal** se conectará con el bloque **Osmocom Source**, mientras que otros bloques pueden estar en cualquier parte del entorno de trabajo.

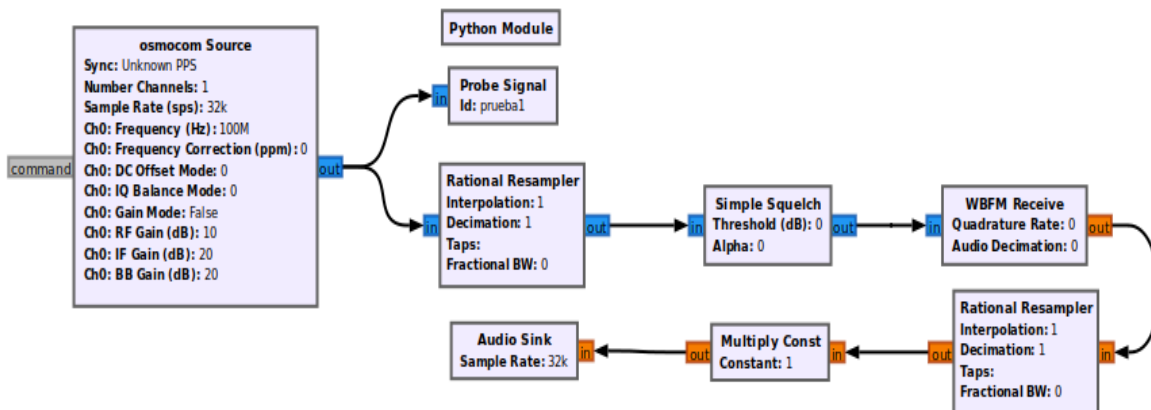


Figura 13. Bloque Python Module Paso 4

Al presionar doble click sobre el bloque Python Module se abrirá la ventana de la Figura 13:

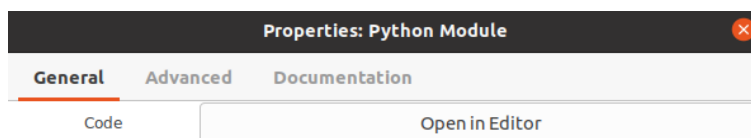


Figura 14. Bloque Python Module Paso 4

Al seleccionar el 'Open Editor', aparecerán las siguientes opciones: Choose Editor, Use Default y Cancelar. Si el usuario un editor de código con cual este familiarizado o desee probar deberá elegir la opción Choose Editor, en caso contrario, se deberá presionar Use Default, esta opción abrirá el editor de texto predeterminado por el Sistema Operativo del Equipo.

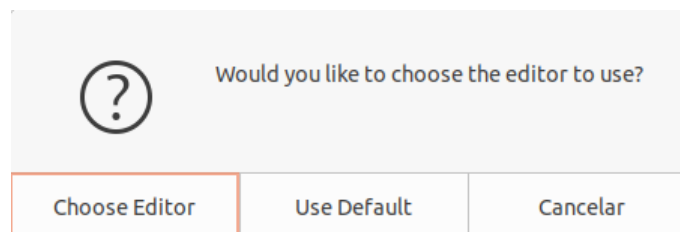
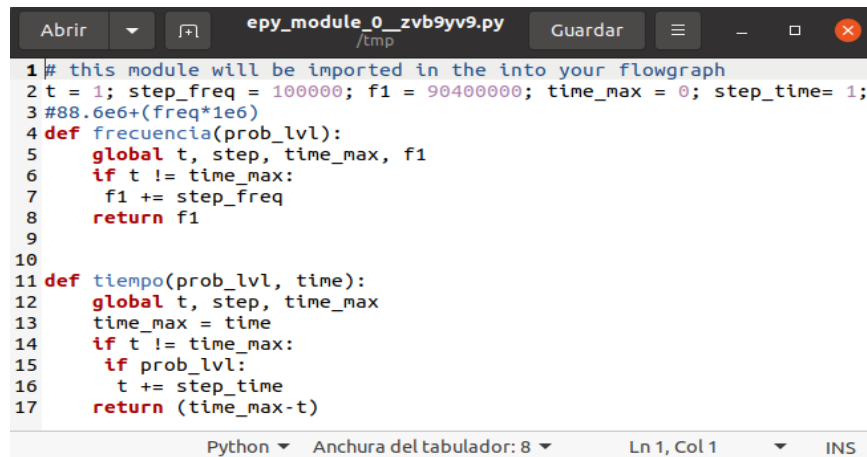


Figura 15. Bloque Python Module Paso 4

Se ingresará el siguiente código Python en el editor de texto, se debe guardar el archivo de texto en la misma carpeta en donde se encuentra el archivo .grc:



```

1 # this module will be imported in the into your flowgraph
2 t = 1; step_freq = 100000; f1 = 90400000; time_max = 0; step_time= 1;
3 #88.6e6+(freq*1e6)
4 def frecuencia(prob_lvl):
5     global t, step, time_max, f1
6     if t != time_max:
7         f1 += step_freq
8     return f1
9
10
11 def tiempo(prob_lvl, time):
12     global t, step, time_max
13     time_max = time
14     if t != time_max:
15         if prob_lvl:
16             t += step_time
17     return (time_max-t)

```

Figura 16. Bloque Python Module Paso 4

Se obtienen los siguientes archivos:

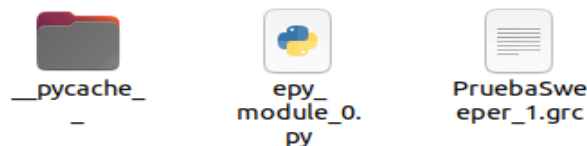


Figura 17. Opciones del Bloque Function Probe

5. Algunas versiones del GNU Radio cuentan con algunos inconvenientes como lo es la obtención del identificador en algunos bloques como lo es en el caso de **Probe Signal**:

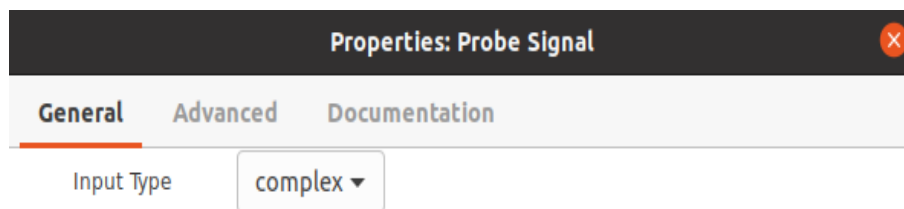


Figura 18. Bloque Python Module Paso 4

Para habilitar el identificador (Id) de un bloque se deben seguir los siguientes pasos, es importante tener esto en cuenta puesto que el archivo no es editable.

- a) Ingresar en la carpeta de bloques '*/usr/local/share/gnuradio/grc/blocks*', la dirección de la carpeta se encuentra en la segunda dirección de la terminal del GNU Radio:

```

<<< Welcome to GNU Radio Companion 3.8.1.0 >>>

Block paths:
/usr/share/gnuradio/grc/blocks
/usr/local/share/gnuradio/grc/blocks

```

Figura 19. Inicio de la Terminal de GNU Radio

- b) Abrir una terminal e ingresar como súper usuario el siguiente código:

```
$ nano blocks_probe_signal_x.block.yml
```

Al ingresar el anterior comando se puede visualizar el siguiente script:

```
id: blocks_probe_signal_x
label: Probe Signal
flags: [ python, cpp ]

parameters:
- id: type
  label: Input Type
  dtype: enum
  options: [complex, float, int, short, byte]
  option_attributes:
    fcn: [c, f, i, s, b]
  hide: part

inputs:
- domain: stream
  dtype: ${ type }

templates:
imports: from gnuradio import blocks
make: blocks.probe_signal_${type.fcn}()

cpp_templates:
includes: ['#include <gnuradio/blocks/probe_signal.h>']
declarations: 'blocks::probe_signal_${type.fcn}::sptr ${id};'
make: 'this->${id} = blocks::probe_signal_${type.fcn}::make();'

documentation: |-
  Available functions to probe: level()

  Use with the function probe block.

file_format: 1
```

Figura 20. Inicio de la Terminal de GNU Radio

- c) Escribir 'show\_id' dentro de la lista de flags como se ve en la figura 21.

```
GNU nano 4.8
id: blocks_probe_signal_x
label: Probe Signal
flags: [ show_id, python, cpp ]
```

Figura 21. Inicio de la Terminal de GNU Radio

Al terminar de realizar los anteriores pasos se puede apreciar el Id se encuentra habilitado en el bloque **Probe Signal**

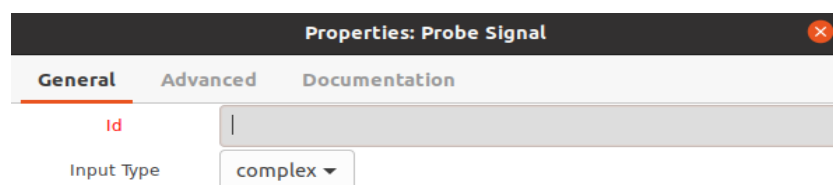


Figura 22. Inicio de la Terminal de GNU Radio

6. Una vez que se habilito el ID del bloque **Probe Signal** en el paso anterior, se procede a escribirle el siguiente nombre como identificador:

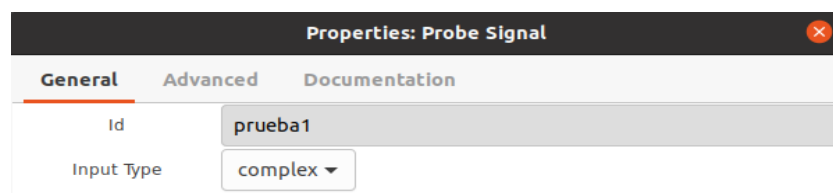


Figura 23. Ini

El bloque **Function Probe** se agrega al entorno de trabajo, este bloque ejecuta una función y establece el identificador de retorno, este bloque se utiliza en conjunto con el bloque **Probe Signal** [18]:

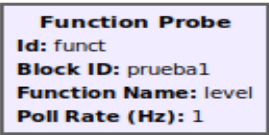


Figura 24. BLoque del Function Probe

Se configura con los siguientes parametros:

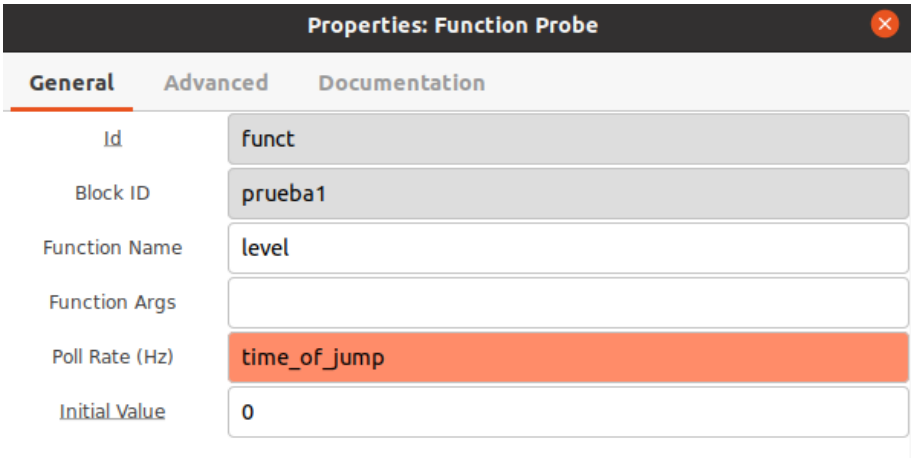


Figura 25. Opciones del Bloque Function Probe

- 7. Para utilizar las funciones del bloque se ingresa en la variable el nombre del archivo y el id del bloque **Function Probe** como el parámetro de la función:

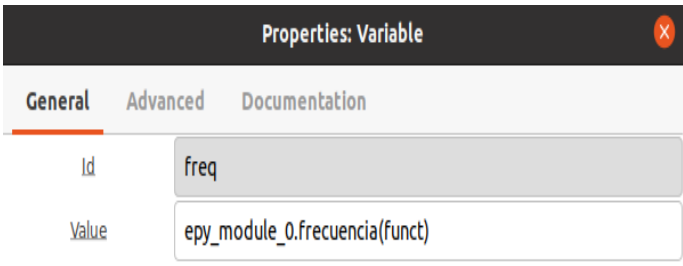


Figura 26. Opciones del Bloque Function Probe

- 8. Como ultimo paso se ejecutará el archivo .grc, al momento de la ejecución se puede apreciar que se creá un nuevo archivo de python con el nombre exacto de la función que se había creado anteriormente

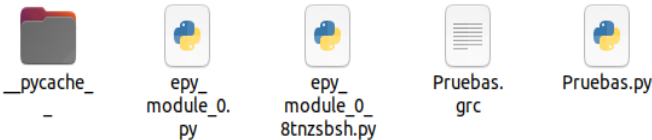


Figura 27. Opciones del Bloque Function Probe

Posteriormente a la creación del archivo de la figura 31, se puede apreciar las siguiente figuras:

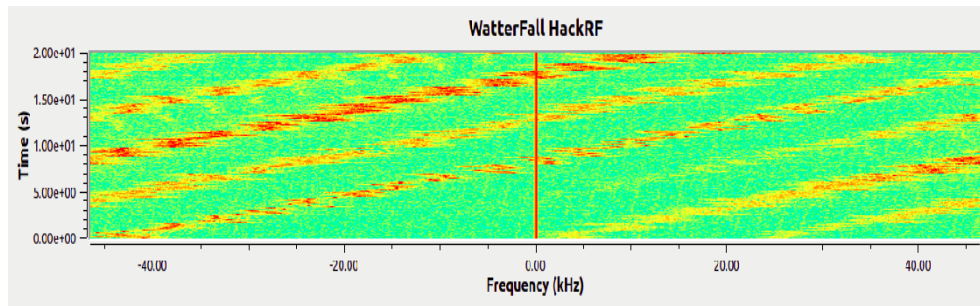


Figura 28. Epspectro visto con el WatterFall

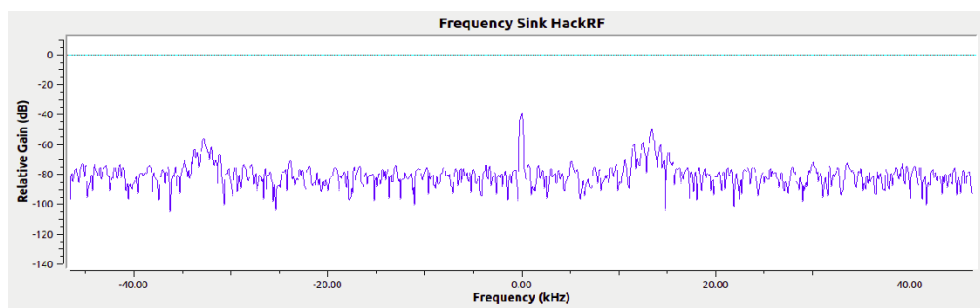


Figura 29. Opciones del Bloque Function Probe

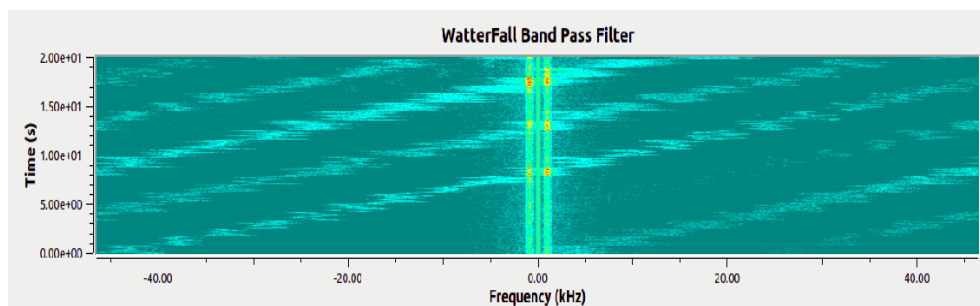


Figura 30. Frecuencia del Espectro

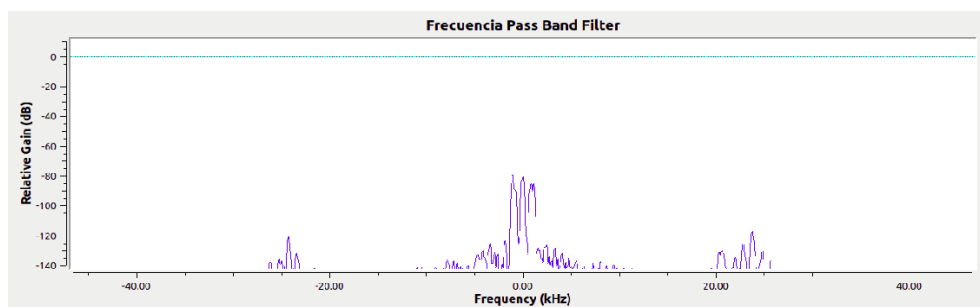


Figura 31. Espectro de la Frecuencia con Filtro Pasa Banda



### 3.3. Obtención del Posicionamiento

Mediante el software GNSS-SDR se obtiene la posición de la ubicación de donde se encuentra el usuario, para ello la antena y el equipo Hackrf One deben estar conectados a un computador con un sistema operativo de Linux, esto es requerido para obtener la señal GPS, más sin embargo esto no es necesario para llevar a cabo la reproducción de la grabación.



Figura 33. GNSS SDR [19]

**3.3.1. Esquemático del GNSS-SDR.** Los bloques de posicionamiento de la señal de la figura 34 muestra el proceso que ejecuta el software GNSS-SDR para obtener la posición. El proceso inicia con un dispositivo SDR (**RF Front-End**) o con un archivo de la señal previamente grabada en un formato aceptable como lo es el .dat (**Raw Signal File**), la señal grabada junto con algunos parámetros que son determinados por el usuario (la frecuencia intermedia, la tasa de muestreo y la cantidad de bits por muestra) son enviados al bloque **Signal Source**, este bloque esta encargado de la implementación del controlador del hardware o de la lectura de muestras de un archivo que fue anteriormente almacenado sin previo procesamiento [20] para ser enviado al bloque **Signal Conditioner**, un bloque encargado de la adaptación de bits de la muestra obtenida a un tipo de dato que el computador trabaje entregando en un formato unificado el flujo de datos de muestra a los canales de procesamiento [21].

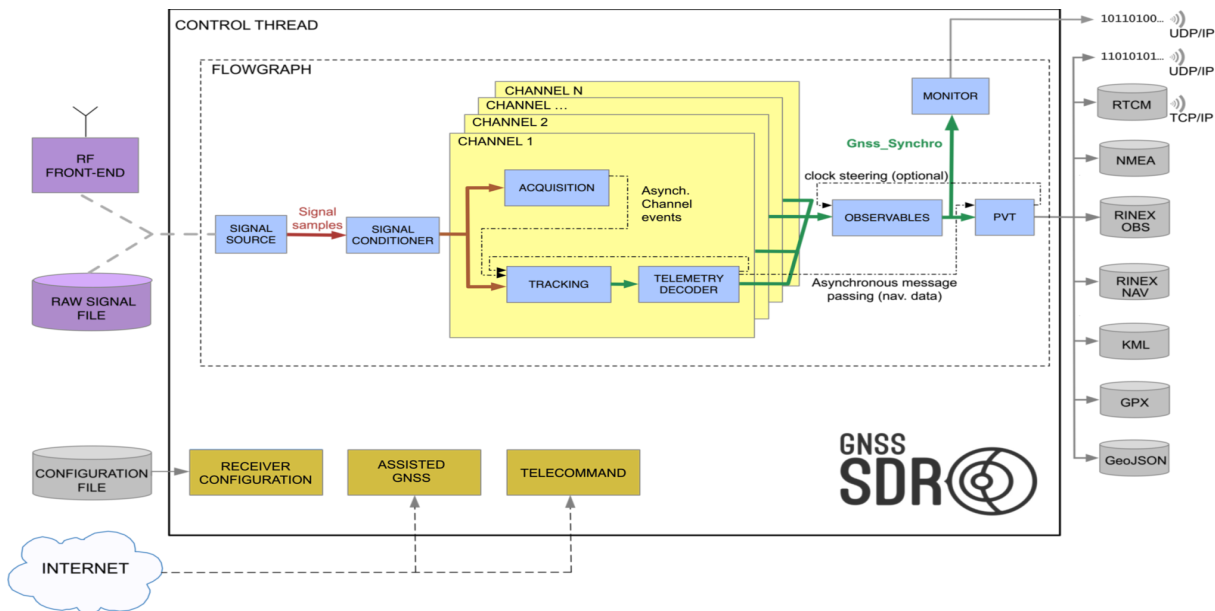


Figura 34. Bloques de procesamiento de la Señal [19]

Los canales encapsulan los bloques **Adquisition**, **Tracking** y **Demodulation of the navigation message** de un solo satélite. El usuario puede definir los canales que el software instanciará, esto se puede hacer en el archivo de configuración `Channels_XX.count`, donde X es uno de los identificadores de la tabla del cuadro 2. Mediante el parametro `Channels.in acquisition` cuyo valor será el número total de canales. Se debe tener en cuenta que se pueden obtener mejores resultados al establecer este parámetro en 1 para aligerar la carga computacional. [22]



Identificador	Señal	Frecuencia de la Señal
1G	Glonass L1 C/A	1602.00 MHz
1C	GPS L1 C/A	1575.42 MHz
1B	Galileo E1 B/C	1575.42 MHz
B1	Beidou B1I	1561.098 MHz
E6	Galileo E6B	1278.75 MHz
B3	Beidou B3I	1268.520 MHz
2G	Glonass L2 C/A	1246.00 MHz
2S	GPS L2 L2CM	1227.60 MHz
7X	Galileo E5b	1207.140 MHz
5X	Galileo E5a	1176.450 MHz
L5	GPS L5C	1176.45 MHz

Cuadro 1. TABLA DE IDENTIFICADORES DE FRECUENCIA [22]

EL bloque **Adquisition** esta encargado de la detección de la presencia y ausencia de la señal obtenida del boque **Signal Conditional**. En el caso de que se detecte la señal se estimaran aproximación de la fase del código y el desplazamiento Doppler. Este bloque se puede editar en el documento de configuración mediante los siguientes parámetros: [23]

Parámetro	Descripción	Requerido
GNSS-SDR.internal_fs_sps	Tasa de muestreo de entrada a los canales de procesamiento por segundo	Obligatorio
GNSS-SDR.use_acquisition_resampler	True: El bloque de adquisición utiliza la frecuencia de muestreo mínima posible durante la adquisición configurando un remuestrador en su entrada, permitiendo una reducción en el tamaño de la Transformada Rápida de Fourier (FFT). False: Configuración por defecto.	Opcional
implementation	GPS_1_CA_PCPS_Acquisition	Obligatorio
item_type	[ gr_complex, cshort, cbyte]: establezca el tipo de datos de muestra esperado en la entrada del bloque. Por defecto es gr_complex.	Opcional
doppler_max	valor Doppler máximo en la cuadrícula de búsqueda, en Hz. El valor predeterminado es 5000 Hz.	Opcional
doppler_step	Paso de frecuencia en la grilla de búsqueda, en Hz. El valor predeterminado es 500 Hz.	Opcional
threshold	Umbral de decisión a partir de la cual se considerará presente una señal. Por defecto es 0.0 (todas las señales se declaran presentes).	Opcional
coherent_integration_time_ms	Establecer el tiempo de integración, en ms. El valor predeterminado es 1 ms.	Opcional
bit_transition_flag	[ true, false]: Si se establece en true, tiene en cuenta la posible presencia de una transición de bit, por lo que se duplica el tiempo efectivo de integración. Cuando se establece, invalida el valor de max_dwells. Por defecto es false.	Opcional
max_dwells	[ true, false]: Si se establece en true, el bloque volverá a buscar el mismo satélite una vez descartada su presencia. Útil para la prueba. Por defecto es false.	Opcional
repeat_satellite	Establezca el número máximo de permanencias no coherentes para declarar una señal presente. El valor predeterminado es 1.	Opcional
blocking	[ true, false]: si se establece en false, la carga de trabajo de adquisición se ejecuta en un subproceso separado, fuera del programador de GNU Radio que administra el gráfico de flujo, y el bloque omite las muestras que llegan mientras el subproceso de procesamiento está ocupado. Esto es especialmente útil en la operación en tiempo real utilizando interfaces de radiofrecuencia, superando el cuello de botella de procesamiento para frecuencias de muestreo medias y altas. Sin embargo, esto rompe el determinismo proporcionado por el programador de GNU Radio y se pueden obtener diferentes resultados de procesamiento en diferentes máquinas. No utilice esta opción para el procesamiento de archivos. Por defecto es true.	Opcional
make_two_steps	[ true, false]: si se establece en true, se realiza una etapa de refinamiento de adquisición después de que se declara presente una señal. Esto permite proporcionar una estimación Doppler actualizada y refinada al bloque de seguimiento. Por defecto es false.	Opcional
second_nbins	Si make_two_steps se establece en true, este parámetro establece el número de intervalos realizados en la etapa de refinamiento de adquisición. Por defecto es 4.	Opcional
second_doppler_step	Si make_two_steps se establece en true, este parámetro establece el paso Doppler aplicado en la etapa de refinamiento de la adquisición, en Hz. El valor predeterminado es 125 Hz.	Opcional
dump	[true, false]: Si se establece en true, habilita el registro del archivo de datos binarios internos de Adquisición. Por defecto es false.	Opcional
dump_filename	Si dump se establece en true, nombre del archivo en el que se almacenarán los datos internos. Este parámetro acepta una ruta relativa o absoluta; si hay carpetas especificadas que no existen, se crearán. El valor predeterminado es , por lo que se generarán ./acquisitionarchivos con nombre ./acquisition_G_1C_ch_N_K_sat_P.mat(donde N es el número de canal definido por dump_channel, K es el número de volcado y es el número PRN del satélite de destino).P	Opcional
dump_channel	Si dump se establece en true, número de canal desde el que se almacenarán los datos internos. Por defecto es 0.	Opcional

Cuadro 2. PARÁMETROS DE ACQUISITION [23]

El bloque **Tracking** es seguir la evolución de los parámetros de sincronización de la señal como lo es la fase de código, desplazamiento Doppler y fase de la portadora. Este bloque se puede configurar con los siguientes parámetros en el archivo de configuración: [24]

Parámetro	Descripción	Requerido
implementation	GPS_L1_CA_DLL_PLL_Tracking	Obligatorio
item_type	[ gr_complex]: establezca el tipo de datos de muestra esperado en la entrada del bloque. Por defecto es gr_complex.	Opcional
extend_correlation_symbols	Establece el número de símbolos de correlación que se extenderán después de lograr la sincronización de bits. Cada símbolo es de 1 ms, por lo que establecer este parámetro en 20 significa un tiempo de integración coherente de 20 ms. Cada bit es de 20 ms, por lo que el valor de este parámetro debe ser un divisor del mismo ( por ejemplo , 2, 4, 5, 10, 20). Cuanto mayor sea este parámetro, se requerirá una mejor estabilidad del reloj local. El valor predeterminado es 1.	Opcional
pll_bw_hz	Ancho de banda del filtro de paso bajo PLL, en Hz. El valor predeterminado es 50 Hz.	Opcional
pll_bw_narrow_hz	Ancho de banda del filtro de paso bajo PLL después de la sincronización de bits, en Hz. El valor predeterminado es 20 Hz.	Opcional
pll_filter_order	[ 2, 3]. Establece el orden del filtro de paso bajo PLL. Por defecto es 3.	Opcional
enable_fll_pull_in	[ true, false]. Si se establece en true, habilita el FLL durante el tiempo de extracción. Por defecto es false	Opcional
enable_fll_steady_state	[ true, false]. Si se establece en true, el FLL se habilita más allá de la etapa pull-in. Por defecto es false.	Opcional
fll_bw_hz	Ancho de banda del filtro de paso bajo FLL, en Hz. El valor predeterminado es 35 Hz.	Opcional
dll_bw_hz	Ancho de banda del filtro de paso bajo DLL, en Hz. El valor predeterminado es 2 Hz.	Opcional
dll_bw_narrow_hz	Ancho de banda del filtro de paso bajo DLL después de la sincronización de bits, en Hz. El valor predeterminado es 2 Hz.	Opcional
dll_filter_order	[ 1, 2, 3]. Establece el orden del filtro de paso bajo DLL. Por defecto es 2.	Opcional
arly_late_space_chips	Espaciamiento entre correladores temprano y rápido y entre rápido y tardío, normalizado por el período del chip $T_c$ . Por defecto es 0.5	Opcional
early_late_space_narrow_chips	Espaciamiento entre correladores temprano y rápido y entre rápido y tardío, normalizado por el período del chip $T_c$ , después de la sincronización de bits. Por defecto es 0.5	Opcional
carrier_aiding	[ true, false]. Si se establece en true, el bucle de código es asistido por el bucle de la portadora. Por defecto es true.	Opcional
cn0_samples	Número de P salidas del correlador utilizadas para la estimación de CN0. El valor predeterminado es 20	Opcional
cn0_min	CN0 mínimo válido (en dB-Hz). El valor predeterminado es 25 dB-Hz.	Opcional
max_lock_fail	Número máximo de fallas de bloqueo antes de soltar un satélite. El valor predeterminado es 50.	Opcional
carrier_lock_th	Umbral de bloqueo de portadora (en rad). El valor predeterminado es 0,85 rad.	Opcional
cn0_smoother_samples	Número de muestras utilizadas para suavizar el valor de la estimación $C/N_0$ . El valor predeterminado es 200 muestras..	Opcional
cn0_smoother_alpha	factor de olvido de la $C/N_0$ más suave, como en $y_k = \alpha x_k + (1 - \alpha)y_{k-1}$ . El valor predeterminado es 0,002.	Opcional
carrier_lock_test_smoother_samples	Número de muestras utilizadas para suavizar el valor de la prueba de bloqueo del portador. El valor predeterminado es 25 muestras.	Opcional
carrier_lock_test_smoother_alpha	El factor de olvido del detector de bloqueo del portador es más suave, como en $y_k = \alpha x_k + (1 - \alpha)y_{k-1}$ . El valor predeterminado es 0,002.	Opcional
dump	[ true, false]: si se establece en true, habilita el registro de archivos de datos binarios internos de seguimiento, en forma de archivos ".dat". Este formato se puede recuperar y trazar en Matlab/Octave, consulte los scripts en gnss-sdr/src/utlis/matlab/. Por defecto es false.	Opcional
dump_filename	Si dumpse establece en true, nombre del archivo en el que se almacenarán los datos internos. Este parámetro acepta una ruta relativa o absoluta; si hay carpetas especificadas que no existen, se crearán. El valor predeterminado es ./track_ch, por lo que se generarán archivos con el formato ".track_chX.dat", donde X está el número de canal.	Opcional
dump_mat	[true, false]. Si dump=true, cuando el receptor sale, puede convertir los archivos ".dat" almacenados por este bloque en archivos ".mat" directamente legibles desde Matlab y Octave. Si el receptor ha procesado más de unos pocos minutos de señal, esta conversión puede llevar mucho tiempo. En sistemas con recursos limitados, puede desactivar esta conversión configurando este parámetro en false. El valor predeterminado es true, por lo que los archivos ".mat" se generan de forma predeterminada si dump=true	Opcional
remove_dat	[true, false]: si dump=true y dump_matno está establecido, o está establecido en true, este parámetro controla si el .datarchivo binario interno se elimina después de la conversión a .mat, dejando una salida más limpia si el usuario no está interesado en el .datarchivo. De forma predeterminada, este parámetro se establece en false.	Opcional
dump_crc_stats	[true, false]: si se establece en true, la tasa de éxito de la verificación de CRC al decodificar los mensajes de navegación se informa en un archivo generado al final del procesamiento (o al salir con q+ [Enter]). De forma predeterminada, este parámetro se establece en false.	Opcional
dump_crc_stats_filename	Si dump_crc_stats=true, este parámetro establece el nombre base de los archivos en los que se informa la tasa de éxito de CRC. El valor predeterminado es , por lo que se crearán telemetry_crc_statslos archivos con el nombre del número de canal.telemetry_crc_statschN.txtNchN	Opcional

Cuadro 3. PARÁMETROS DE TRACKING [24]

El bloque **Telemetry Decoder** tiene la tarea de obtener los bits de datos del mensaje de navegación del bloque **Tracking**. Este bloque tiene los siguientes parámetros de configuración. [25]

Parámetro	Descripción	Requerido
implementation	GPS_L1_CA_Telemetry_Decoder	Obligatorio
dump	[true, false]: si se establece en true, habilita el registro de archivos de datos binarios internos del decodificador de telemetría (consulte la sección Salida binaria a continuación para obtener más detalles). Por defecto es false.	Opcional
dump_filename	Si dump se establece en true, nombre base de los archivos en los que se almacenarán los datos internos. El valor predeterminado es ./telemetry, por lo que los archivos se nombrarán ./telemetryN, donde N es el número de canal (agregado automáticamente).	Opcional
dump_filename	[Si dump se establece en true, nombre base de los archivos en los que se almacenarán los datos internos. El valor predeterminado es ./telemetry, por lo que los archivos se nombrarán ./telemetryN, donde N es el número de canal (agregado automáticamente).	Opcional

Cuadro 4. PARÁMETROS DE TELEMETRY DECODER [25]

La función del bloque **Observable** es de la recopilación de los datos de sincronización de todos los canales y de cualquier valor de las medidas básicas del GNSS (Pseudorange, fase de la portadora y desplazamiento Doppler). Se puede configurar de la siguiente manera: [26]

Parámetro	Descripción	Requerido
implementation	Hybrid_Observables	Obligatorio
enable_carrier_smoothing	[true, false]: si se establece en true, habilita el suavizado de portadores de pseudorange de código. Por defecto es false.	Opcional
smoothing_factor	Si enable_carrier_smoothing se establece en true, este parámetro establece el factor de suavizado. Por defecto es 200.	Opcional
dump	[ true, false]: si se establece en true, habilita el registro de archivos de datos binarios internos de Observables. El almacenamiento en archivos .mat legibles desde Matlab, Octave y Python está disponible a partir de GNSS-SDR v0.0.10, consulte a continuación. Por defecto es false.	Opcional
dump_filename	Si dump se establece en true, nombre del archivo en el que se almacenarán los datos internos. Este parámetro acepta una ruta relativa o absoluta; si hay carpetas especificadas que no existen, se crearán. Por defecto es ./observables.dat	Opcional
dump_mat	[ true, false]. Si dump=true, cuando el receptor sale, puede convertir los archivos ".dat" almacenados por este bloque en archivos ".mat" directamente legibles desde Matlab y Octave. Si el receptor ha procesado más de unos pocos minutos de señal, esta conversión puede llevar mucho tiempo. En sistemas con recursos limitados, puede desactivar esta conversión configurando este parámetro en false. El valor predeterminado es true, por lo que los archivos ".mat" se generan de forma predeterminada si dump=true.	Opcional

Cuadro 5. PARÁMETROS DE TELEMETRY DECODER [26]

El bloque TVP tiene la función de calcular la solución de navegación y entregar la información en un formato determinado para su posterior procesamiento o representación de datos. Esta configurando mediante los siguientes parámetros: [27]

Parámetro	Descripción	Requerido
implementation	RTKLIB_PVT	Obligatorio
output_rate_ms	Velocidad a la que se calcularán las soluciones PVT, en ms. El mínimo es de 20 ms, y el valor debe ser un múltiplo de él. El valor predeterminado es 500 ms.	Opcional
display_rate_ms	Velocidad a la que se mostrarán las soluciones PVT en el terminal, en ms. Debe ser múltiplo de output_rate_ms. El valor predeterminado es 500 ms.	Opcional
positioning_mode	[Single, PPP_Static, PPP_Kinematic] Establecer el modo de posicionamiento. Single: Posicionamiento de punto único. PPP_Static: Posicionamiento de punto preciso con modo estático. PPP_Kinematic: Posicionamiento de punto preciso para un receptor en movimiento. Por defecto es Single.	Opcional
num_bands	[1: L1 Frecuencia única, 2: L1 y L2 Frecuencia dual, 3: L1, L2 y L5 Frecuencia triple] Esta opción se configura automáticamente según la configuración de Canales. Esta opción puede ser útil para forzar alguna configuración (por ejemplo , solución de banda única en un receptor de doble frecuencia).	Opcional
elevation_mask	[0: Desactivado, 1: Activado] Configure el modelo dinámico del receptor. Si se establece en 1 y , la posición del receptor se predice con la velocidad y la aceleración estimadas. Por defecto es 0 (sin modelo dinámico). y PVT.positioning_mode=PPP_Kinematic(sin modelo dinámico).	Opcional

iono_model	[ OFF, Broadcast, Iono-Free-LC]. Establecer opciones de corrección ionosférica. OFF: No aplicar la corrección ionosférica. Broadcast: Aplicar el modelo ionosférico de difusión. Iono-Free-LC: La combinación lineal libre de ionosfera con mediciones de doble frecuencia (L1-L2 para GPS o L1-L5 para Galileo) se utiliza para la corrección ionosférica. Por defecto es OFF(sin corrección ionosférica)	Opcional
trop_model	[ OFF, Saastamoinen, Estimate_ZTD, Estimate_ZTD_Grad]. Establezca si los parámetros troposféricos (retraso total del cenit en las posiciones del móvil y de la estación base) se estiman o no. OFF: No aplicar la corrección de la troposfera. Saastamoinen: Aplicar el modelo de Saastamoinen. Estimate_ZTD: Estime los parámetros ZTD (retraso total cenital) como indica EKF. Estimate_ZTD_Grad: Estime los parámetros ZTD y de gradiente horizontal como indica EKF. Si el valor predeterminado es OFF(sin corrección de la troposfera).	Opcional
enable_rx_clock_correction	[ true, false]: si se establece en true, el receptor utiliza la solución PVT para corregir la temporización en los observables y, por lo tanto, proporciona mediciones continuas en largos períodos de observación. Si se establece en false, la solución de tiempo solo se usa en el cálculo de observables cuando la estimación del desplazamiento del reloj supera el valor de max_clock_offset_ms. Este parámetro por defecto es false.	Opcional
enable_rx_clock_correction	[ true, false]: si se establece en true, el receptor utiliza la solución PVT para corregir la temporización en los observables y, por lo tanto, proporciona mediciones continuas en largos períodos de observación. Si se establece en false, la solución de tiempo solo se usa en el cálculo de observables cuando la estimación del desplazamiento del reloj supera el valor de max_clock_offset_ms. Este parámetro por defecto es false.	Opcional
max_clock_offset_ms	Si enable_rx_clock_correction se establece en false, este parámetro establece el desplazamiento de reloj local máximo permitido con respecto a la solución de tiempo. Si el desplazamiento estimado excede este parámetro, se aplica una corrección de reloj al cálculo de Observables. El valor predeterminado es 40 ms.	Opcional
code_phase_error_ratio_l1	Relación código/error de fase $R_r$ para la banda L1. Por defecto es 100 para la banda L1. Por defecto es.	Opcional
carrier_phase_error_factor_a	Factor de error de fase de la portadora $a_{\sigma}^2$ . Por defecto es 0,003 m. Por defecto es 0.003 metro.	Opcional
carrier_phase_error_factor_b	Factor de error de fase de la portadora $b_{\sigma}^2$ . Por defecto es 0,003 m. Por defecto es metro.	Opcional
slip_threshold	Establezca el umbral de deslizamiento de ciclo (m) de la diferencia de fase de portadora LC sin geometría entre épocas. Por defecto es 0,05.	Opcional
threshold_reject_GDOP	Establezca el umbral de rechazo de GDOP. Si el GDOP está por encima del valor, el observable se excluye del proceso de estimación como un valor atípico. Por defecto es 30,0.	Opcional
threshold_reject_innovation	Establezca el umbral de rechazo de la innovación (preajuste residual) (m). Si la innovación está por encima del valor, el observable se excluye del proceso de estimación como un valor atípico. Por defecto es 30,0 m.metro.	Opcional
number_filter_iter	Establezca el número de iteraciones en la actualización de medidas del filtro de estimación. Si la longitud de la línea de base es muy corta, como 1 m, la iteración puede ser efectiva para manejar la no linealidad de la ecuación de medición. El valor predeterminado es 1.	Opcional
sigma_bias	Establezca la desviación estándar del ruido del proceso de polarización de fase portadora $\sigma_{bias}$ , en ciclos/ $\sqrt{s}$ . Por defecto es 0,0001 ciclos/ $\sqrt{s}$ , en ciclos/. Por defecto es 0.0001	Opcional
sigma_trop	Establezca la desviación estándar del ruido del proceso del retraso troposférico cenital $\sigma_Z$ , en m/ $\sqrt{s}$ . Por defecto es 0,0001 m/ $\sqrt{s}$ , En m/. Por defecto es 0.0001 metro $\sqrt{s}$	Opcional
raim_fde	[0, 1]: establezca si la función RAIM (monitoreo autónomo de la integridad del receptor) FDE (detección y exclusión de fallas) está habilitada o no. Por defecto es 0 (RAIM no habilitado)(RAIM no habilitado)	Opcional
reject_GPS_IIA	[0, 1]: establece si se excluyen o no los satélites GPS Block IIA. Esos satélites a menudo degradan las soluciones PPP debido al comportamiento imprevisto de la actitud de guiñada. Por defecto es 0 (sin rechazo).(sin rechazo).	Opcional
phwindup	[ 0, 1]: Establezca si se aplica o no la corrección de liquidación de fase $\phi_{pw}$ para los modos PPP. El valor predeterminado es 0 (sin corrección de fase).para los modos PPP se aplica o no. Por defecto es(sin corrección de liquidación de fase).	Opcional
output_enabled	[ true, false]: si se establece en false, los archivos de datos de salida no se almacenan. Por defecto es true.	Opcional
rtcm_output_file_enabled	[true, false]: si se establece en false, los archivos binarios RTCM no se almacenan. Por defecto es false.	Opcional
gpx_output_enabled	[true, false]: si se establece en false, los archivos GPX no se almacenan. Por defecto es output_enabled.	Opcional
geojson_output_enabled	[true, false]: si se establece en false, los archivos GeoJSON no se almacenan. Por defecto es output_enabled	Opcional
xml_output_enabled	[true, false]: si se establece en false, los archivos XML no se almacenan. Por defecto es output_enabled	Opcional

Cuadro 6. PARÁMETROS DE TELEMETRY DECODER [27]

**3.3.2. Código De Recepción:.** Se debe tener el siguiente código en un archivo de texto con el nombre que el usuario desee finalizado en ".conf", preferiblemente ubicada en una carpeta apartada para la ejecución del proyecto: [10]

[GNSS-SDR]

```
##### GLOBAL OPTIONS #####
GNSS-SDR.internal_fs_sps=2000000

##### SIGNAL_SOURCE CONFIG #####
SignalSource.implementation=Osmosdr_Signal_Source
SignalSource.item_type=gr_complex
SignalSource.sampling_frequency=2000000
SignalSource.freq=1575420000
SignalSource.gain=40
SignalSource.rf_gain=40
SignalSource.if_gain=30
SignalSource.AGC_enabled=false
SignalSource.samples=0
SignalSource.repeat=false
;# Next line enables the internal HackRF One bias (3.3 VDC)
SignalSource.osmosdr_args=hackrf,bias=1
SignalSource.enable_throttle_control=false
SignalSource.dump=true
SignalSource.dump_filename="Nombre_Archivo".dat

##### SIGNAL_CONDITIONER CONFIG #####
SignalConditioner.implementation=Signal_Conditioner

##### DATA_TYPE_ADAPTER CONFIG #####
DataAdapter.implementation=Pass_Through

##### INPUT_FILTER CONFIG #####
InputFilter.implementation=Freq_Xlating_Fir_Filter
InputFilter.decimation_factor=1
InputFilter.input_item_type=gr_complex
InputFilter.output_item_type=gr_complex
InputFilter.taps_item_type=float
InputFilter.number_of_taps=5
InputFilter.number_of_bands=2
InputFilter.band1_begin=0.0
InputFilter.band1_end=0.85
InputFilter.band2_begin=0.9
InputFilter.band2_end=1.0
InputFilter.ampl1_begin=1.0
InputFilter.ampl1_end=1.0
InputFilter.ampl2_begin=0.0
InputFilter.ampl2_end=0.0
InputFilter.band1_error=1.0
InputFilter.band2_error=1.0
InputFilter.filter_type=bandpass
InputFilter.grid_density=16
InputFilter.dump=false
InputFilter.dump_filename=Lectura/Grabacion_#27_04_2022INPUTFILTER.dat

##### RESAMPLER CONFIG #####
Resampler.implementation=Pass_Through

##### CHANNELS GLOBAL CONFIG #####
Channels_1C.count=8
Channels.in_acquisition=1

##### ACQUISITION GLOBAL CONFIG #####
Acquisition_1C.implementation=GPS_L1_CA_PCPS_Acquisition
Acquisition_1C.item_type=gr_complex
Acquisition_1C.coherent_integration_time_ms=1
Acquisition_1C.pfa=0.01
```

```

Acquisition_1C.doppler_max=5000
Acquisition_1C.doppler_step=250
Acquisition_1C.max_dwells=1
Acquisition_1C.dump=false
Acquisition_1C.dump_filename=Lectura/Grabacion_#07_04_2022ACQUISITION.dat

##### TRACKING GLOBAL CONFIG #####
Tracking_1C.implementation=GPS_L1_CA_DLL_PLL_Tracking
Tracking_1C.item_type=gr_complex
Tracking_1C.extend_correlation_symbols=10
Tracking_1C.early_late_space_chips=0.5
Tracking_1C.early_late_space_narrow_chips=0.15
Tracking_1C.pll_bw_hz=40
Tracking_1C.dll_bw_hz=2.0
Tracking_1C.pll_bw_narrow_hz=5.0
Tracking_1C.dll_bw_narrow_hz=1.50
Tracking_1C.fll_bw_hz=10
Tracking_1C.enable_fll_pull_in=true
Tracking_1C.enable_fll_steady_state=false
Tracking_1C.dump=false
Tracking_1C.dump_filename=tracking_ch_

##### TELEMETRY DECODER GPS CONFIG #####
TelemetryDecoder_1C.implementation=GPS_L1_CA_Telemetry_Decoder
TelemetryDecoder_1C.dump=false

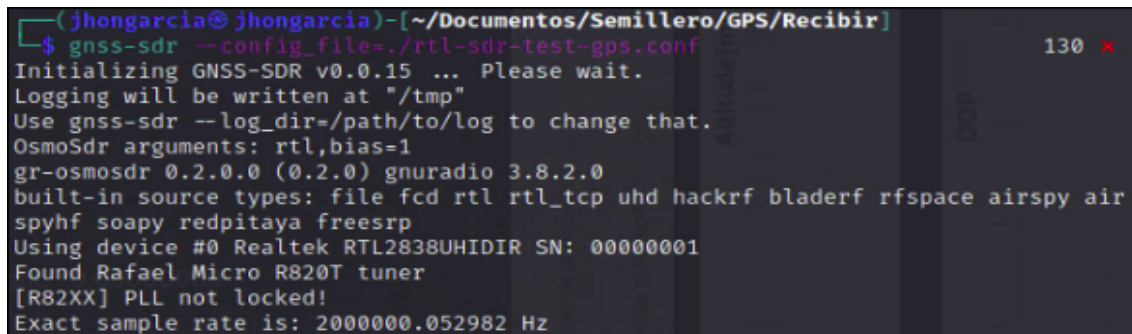
##### OBSERVABLES CONFIG #####
Observables.implementation=Hybrid_Observables
Observables.dump=false
Observables.dump_filename=./observables.dat

##### PVT CONFIG #####
PVT.implementation=RTKLIB_PVT
PVT.positioning_mode=Single
PVT.output_rate_ms=100
PVT.display_rate_ms=500
PVT.iono_model=Broadcast
PVT.trop_model=Saastamoinen
PVT.flag_rtcn_server=true
PVT.flag_rtcn_tty_port=false
PVT.rtcn_dump_devname=/dev/pts/1
PVT.rtcn_tcp_port=2101
PVT.rtcn_MT1019_rate_ms=5000
PVT.rtcn_MT1077_rate_ms=1000
PVT.rinex_version=2

```

Para ejecutar el código anterior se deberá abrir la terminal y estar en el directorio de trabajo del archivo guardado.

*\$ gnss-sdr -config\_file=./"Nombre del archivo".conf*



```

(jhongarcia@jhongarcia)-[~/Documentos/Semillero/GPS/Recibir]
$ gnss-sdr -config_file=./rtl-sdr-test-gps.conf
Initializing GNSS-SDR v0.0.15 ... Please wait.
Logging will be written at "/tmp"
Use gnss-sdr --log_dir=/path/to/log to change that.
OsmoSdr arguments: rtl,bias=1
gr-osmosdr 0.2.0.0 (0.2.0) gnuradio 3.8.2.0
built-in source types: file fcd rtl rtl_tcp uhd hackrf bladerf rfspc airspy air
spyhf soapy redpitaya freesrp
Using device #0 Realtek RTL2838UHIR SN: 00000001
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Exact sample rate is: 2000000.052982 Hz

```

Figura 35. Ejecución del código de grabación GNSS-SDR

En el transcurso de la ejecución se crearan archivos .dat, .gpx, .nmea, etc. Más sin embargo el archivo que se deberá tener en cuenta para futuras reproducciones es el archivo .dat

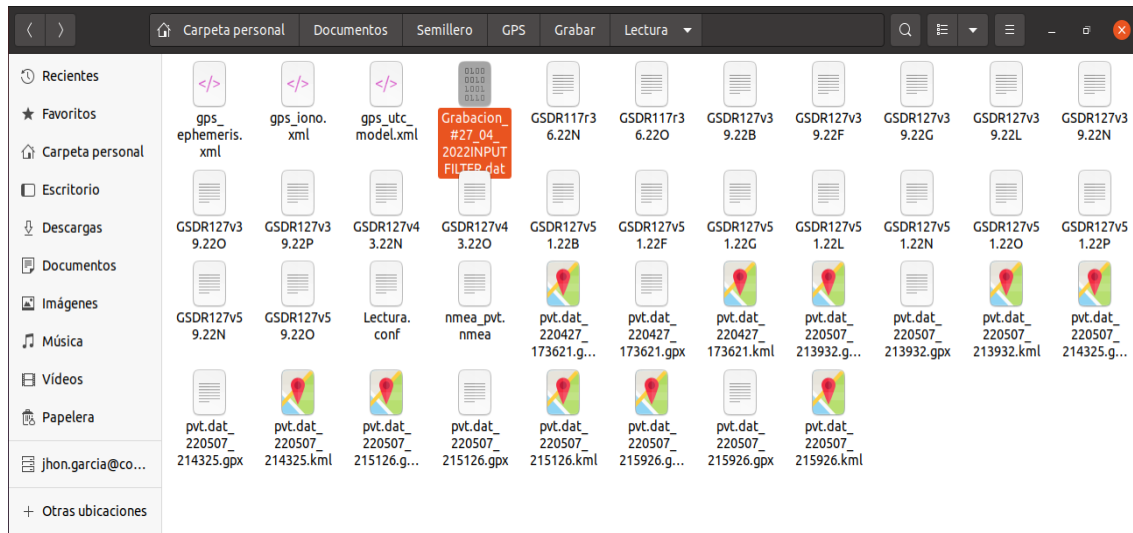


Figura 36. Archivos creados por el GNSS-SDR

El tiempo para obtener la posición con el GNSS-SDR puede variar por el entorno en el que se encuentre el usuario.

```
First position fix at 2021-Nov-04 21:31:18.140000 UTC is Lat = 4.6609 [deg], Long
= -74.0599 [deg], Height= 2575.97 [m]
Position at 2021-Nov-04 21:31:18.500000 UTC using 4 observations is Lat = 4.66092
1501 [deg], Long = -74.059964782 [deg], Height = 2579.336 [m]
Velocity: East: -1.453 [m/s], North: 0.798 [m/s], Up = 2.867 [m/s]
Current receiver time: 1 min 6 s
Position at 2021-Nov-04 21:31:19.000000 UTC using 4 observations is Lat = 4.66090
6671 [deg], Long = -74.059951383 [deg], Height = 2578.855 [m]
Velocity: East: 0.523 [m/s], North: -0.073 [m/s], Up = -0.010 [m/s]
Position at 2021-Nov-04 21:31:19.500000 UTC using 4 observations is Lat = 4.66091
4356 [deg], Long = -74.059934608 [deg], Height = 2577.014 [m]
Velocity: East: -1.263 [m/s], North: 0.275 [m/s], Up = 0.539 [m/s]
Current receiver time: 1 min 7 s
Position at 2021-Nov-04 21:31:20.000000 UTC using 4 observations is Lat = 4.66091
8616 [deg], Long = -74.059920211 [deg], Height = 2576.617 [m]
Velocity: East: 0.845 [m/s], North: -2.400 [m/s], Up = -4.034 [m/s]
Position at 2021-Nov-04 21:31:20.500000 UTC using 4 observations is Lat = 4.66092
3676 [deg], Long = -74.059925244 [deg], Height = 2577.886 [m]
Velocity: East: -0.721 [m/s], North: -0.407 [m/s], Up = 1.407 [m/s]
Current receiver time: 1 min 8 s
```

Figura 37. Monitor GNSS-SDR con la ubicación del usuario

**3.3.3. Código de Reproducción:** Se debe tener el siguiente código en un archivo de texto, preferiblemente en una carpeta apartada para la ejecución del proyecto:

```
[GNSS-SDR]

##### GLOBAL OPTIONS #####
GNSS-SDR.internal_fs_sps=2000000

##### SIGNAL_SOURCE CONFIG #####
SignalSource.implementation=File_Signal_Source
SignalSource.filename="NombreArchivo".dat
SignalSource.item_type=gr_complex
SignalSource.sampling_frequency=4000000
SignalSource.samples=0

##### SIGNAL_CONDITIONER CONFIG #####
SignalConditioner.implementation=Pass_Through
```

```

##### CHANNELS GLOBAL CONFIG #####
Channels_1C.count=8
Channels.in_acquisition=1
Channel.signal=1C

##### ACQUISITION GLOBAL CONFIG #####
Acquisition_1C.implementation=GPS_L1_CA_PCPS_Acquisition
Acquisition_1C.item_type=gr_complex
Acquisition_1C.pfa=0.01
Acquisition_1C.doppler_max=10000
Acquisition_1C.doppler_step=250

##### TRACKING GLOBAL CONFIG #####
Tracking_1C.implementation=GPS_L1_CA_DLL_PLL_Tracking
Tracking_1C.item_type=gr_complex
Tracking_1C.pll_bw_hz=40.0;
Tracking_1C.dll_bw_hz=4.0;

##### TELEMETRY DECODER GPS CONFIG #####
TelemetryDecoder_1C.implementation=GPS_L1_CA_Telemetry_Decoder

##### OBSERVABLES CONFIG #####
Observables.implementation=Hybrid_Observables

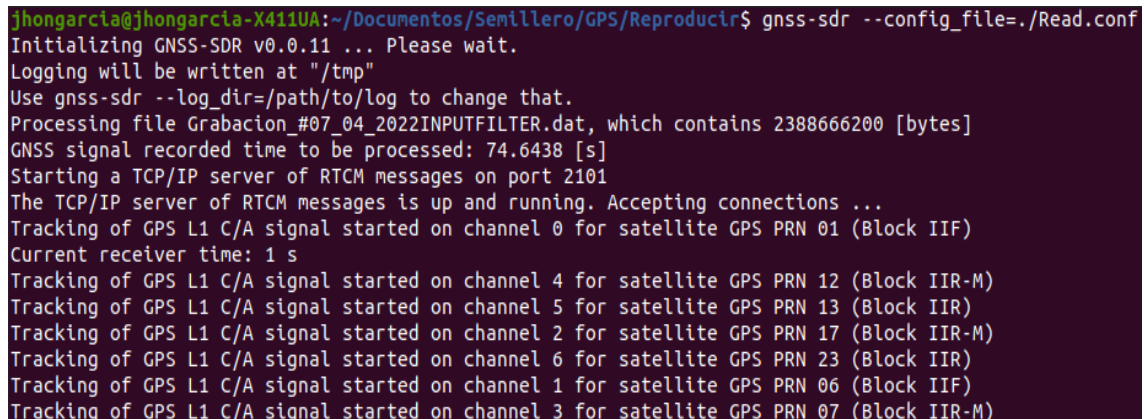
##### PVT CONFIG #####
PVT.implementation=RTKLIB_PVT
PVT.positioning_mode=Single
PVT.output_rate_ms=100
PVT.display_rate_ms=500
PVT.iono_model=Broadcast
PVT.trop_model=Saastamoinen
PVT.flag_rtcn_server=true
PVT.flag_rtcn_tty_port=false
PVT.rtcn_dump_devname=/dev/pts/1
PVT.rtcn_tcp_port=2101
PVT.rtcn_MT1019_rate_ms=5000
PVT.rtcn_MT1077_rate_ms=1000
PVT.rinex_version=2

```

Para ejecutar el código anterior se deberá abrir la terminal y estar en el directorio de trabajo del archivo guardado, se deberá tener en cuenta que la ubicación del archivo guardado .dat tendrá que estar configurado en el archivo de grabación.

*\$ gnss-sdr --config\_file=./"Nombre del archivo".conf*

Al ingresar el código en la terminal del Linux se obtiene:



```

jhongarcia@jhongarcia-X411UA:~/Documentos/Semillero/GPS/Reproducir$ gnss-sdr --config_file=./Read.conf
Initializing GNSS-SDR v0.0.11 ... Please wait.
Logging will be written at "/tmp"
Use gnss-sdr --log_dir=/path/to/log to change that.
Processing file Grabacion_#07_04_2022INPUTFILTER.dat, which contains 2388666200 [bytes]
GNSS signal recorded time to be processed: 74.6438 [s]
Starting a TCP/IP server of RTCM messages on port 2101
The TCP/IP server of RTCM messages is up and running. Accepting connections ...
Tracking of GPS L1 C/A signal started on channel 0 for satellite GPS PRN 01 (Block IIF)
Current receiver time: 1 s
Tracking of GPS L1 C/A signal started on channel 4 for satellite GPS PRN 12 (Block IIR-M)
Tracking of GPS L1 C/A signal started on channel 5 for satellite GPS PRN 13 (Block IIR)
Tracking of GPS L1 C/A signal started on channel 2 for satellite GPS PRN 17 (Block IIR-M)
Tracking of GPS L1 C/A signal started on channel 6 for satellite GPS PRN 23 (Block IIR)
Tracking of GPS L1 C/A signal started on channel 1 for satellite GPS PRN 06 (Block IIF)
Tracking of GPS L1 C/A signal started on channel 3 for satellite GPS PRN 07 (Block IIR-M)

```

Figura 38. Monitor GNSS-SDR con la ubicación del usuario



El tiempo que puede demorar en obtener la posición dependerá de la grabación que se realizó y de la configuración del archivo.

```
Current receiver time: 1 min 26 s
First position fix at 2022-Apr-07 23:20:38.500000 UTC is Lat = 4.66124 [deg], Long = -74.0603 [deg], Height= 2440.64 [m]
Position at 2022-Apr-07 23:20:38.500000 UTC using 4 observations is Lat = 4.661238098 [deg], Long = -74.060277574 [deg], Height = 2440.639 [m]
Position at 2022-Apr-07 23:20:39.000000 UTC using 4 observations is Lat = 4.661150254 [deg], Long = -74.060235409 [deg], Height = 2480.781 [m]
Position at 2022-Apr-07 23:20:39.500000 UTC using 4 observations is Lat = 4.661055366 [deg], Long = -74.060121963 [deg], Height = 2560.053 [m]
Position at 2022-Apr-07 23:20:40.000000 UTC using 4 observations is Lat = 4.661126308 [deg], Long = -74.060185752 [deg], Height = 2524.839 [m]
Current receiver time: 1 min 27 s
Position at 2022-Apr-07 23:20:40.500000 UTC using 4 observations is Lat = 4.661086927 [deg], Long = -74.060123610 [deg], Height = 2571.358 [m]
Position at 2022-Apr-07 23:20:41.000000 UTC using 4 observations is Lat = 4.660985436 [deg], Long = -74.060032856 [deg], Height = 2628.593 [m]
Current receiver time: 1 min 28 s
Position at 2022-Apr-07 23:20:41.500000 UTC using 4 observations is Lat = 4.661013094 [deg], Long = -74.060098579 [deg], Height = 2597.429 [m]
New GPS NAV message received in channel 1: subframe 5 from satellite GPS PRN 20 (Block IIR)
New GPS NAV message received in channel 0: subframe 5 from satellite GPS PRN 05 (Block IIR-M)
New GPS NAV message received in channel 7: subframe 5 from satellite GPS PRN 28 (Block IIR)
New GPS NAV message received in channel 2: subframe 5 from satellite GPS PRN 17 (Block IIR-M)
New GPS NAV message received in channel 3: subframe 5 from satellite GPS PRN 11 (Block IIR)
New GPS NAV message received in channel 4: subframe 5 from satellite GPS PRN 12 (Block IIR-M)
Position at 2022-Apr-07 23:20:42.000000 UTC using 4 observations is Lat = 4.661063493 [deg], Long = -74.060112822 [deg], Height = 2572.123 [m]
Current receiver time: 1 min 29 s
Position at 2022-Apr-07 23:20:42.500000 UTC using 4 observations is Lat = 4.660978569 [deg], Long = -74.060010800 [deg], Height = 2628.314 [m]
Position at 2022-Apr-07 23:20:43.000000 UTC using 4 observations is Lat = 4.661083112 [deg], Long = -74.060094173 [deg], Height = 2571.268 [m]
```

Figura 39. Entorno de GNU Radio

### 3.4. Introducción a MATLAB

3.4.1. Partes del Entorno en MATLAB. Este software esta dividido en 6 partes:

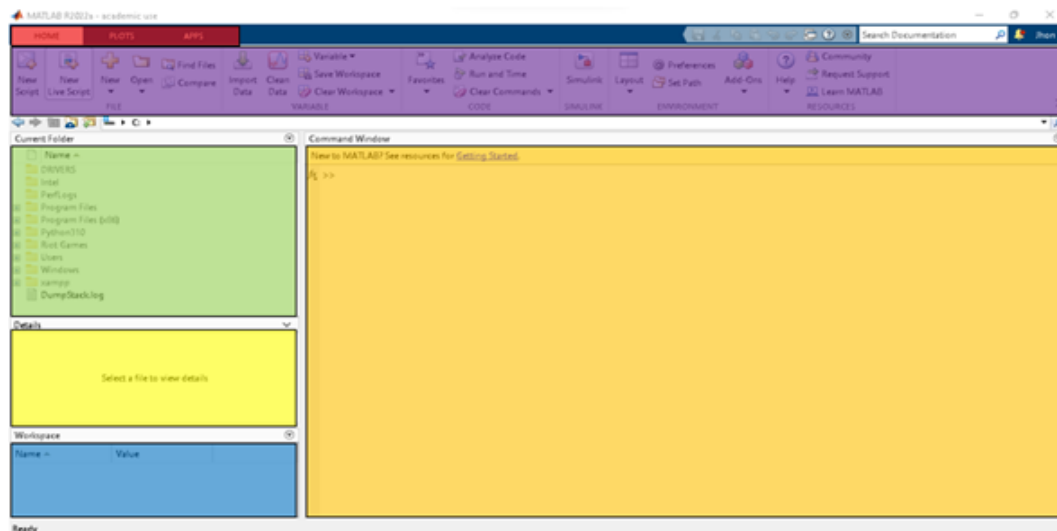


Figura 40. Entorno de MATLAB

- **Barra de Menús:** Se pueden encontrar tres opciones orientados a distintas actividades que se pueden realizar en el programa, estos son **HOM**, **PLOTS**, **APPS**. **HOME** Esta es la pestaña principal en donde se pueden gestionar scripts (archivos de código matlab). **PLOTS** Esta es la pestaña de gráficos con la cual se puede tener acceso al menú de gráficos para la representación de variables. **APPS** Esta es la pestaña de aplicaciones, con ella se puede tener acceso al menú de gestion de aplicaciones que pueden correr en MATLAB. [11]
- **Barra de Herramientas:** Permiten acceder de una manera fácil y rápida los menús más utilizados. [11]
- **Directorio de Trabajo:** Directorio en el que se pueden encontrar las funciones del trabajo que estemos realizando, en esta sección se pueden gestionar los archivos que hay en el directorio actual. [11]

- **Detalles** : Al momento de hacer click sobre algun archivo, nos muestra el contenido del archivo. [11]

Al descargar el archivo .zip del siguiente repositorio y abrirlo en MATLAB se puede visualizar la siguiente imagen

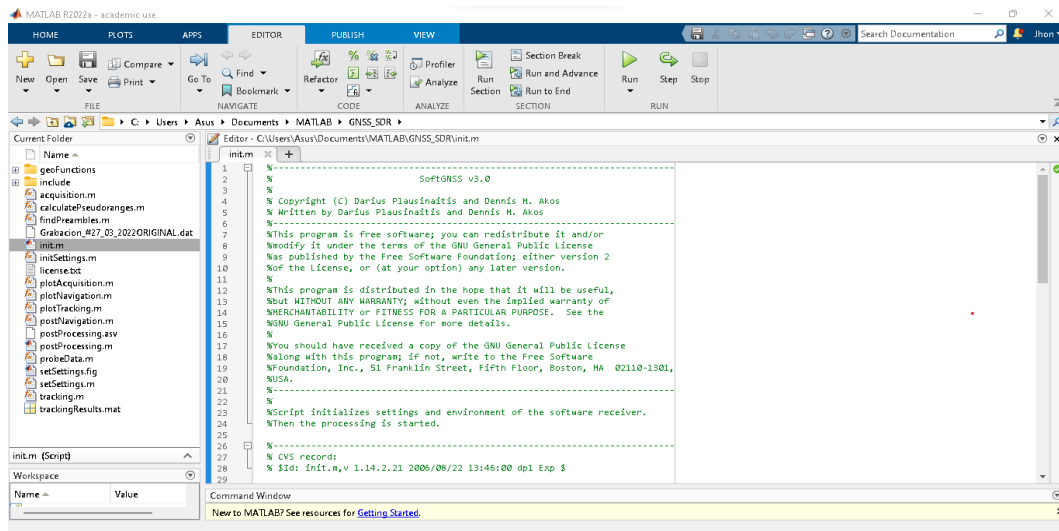


Figura 41. Entorno de MATLAB

Los archivos que se presentan son los siguientes:

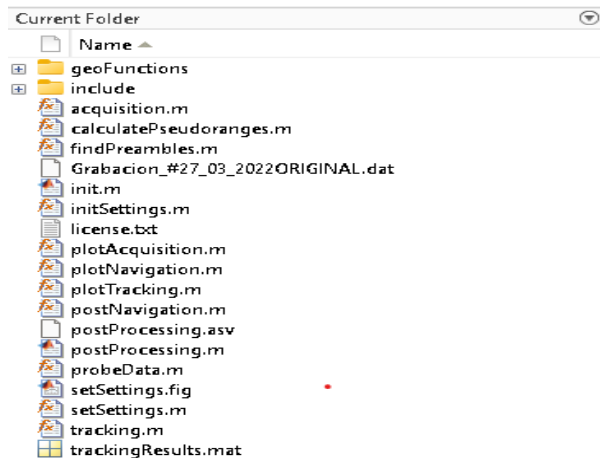


Figura 42. Entorno de MATLAB

Al momento de ejecutar el programa SoftGNSS V3.0 se deben ingresar los siguientes parámetros:

```
Welcome to:  softGNSS

An open source GNSS SDR software project initiated by:

    Danish GPS Center/Aalborg University

The code was improved by GNSS Laboratory/University of Colorado.

The software receiver softGNSS comes with ABSOLUTELY NO WARRANTY;
for details please read license details in the file license.txt. This
is free software, and you are welcome to redistribute it under
the terms described in the license.

-----

Probing data (Grabacion_#27_03_2022ORIGINAL.dat)...
Raw IF data plotted
(run setSettings or change settings in "initSettings.m" to reconfigure)

Enter "1" to initiate GNSS processing or "0" to exit : 1
```

Figura 43. Entorno de MATLAB

Al momento de ingresar los parámetros anteriores se muestra la siguiente ventana con algunos parámetros iniciales:

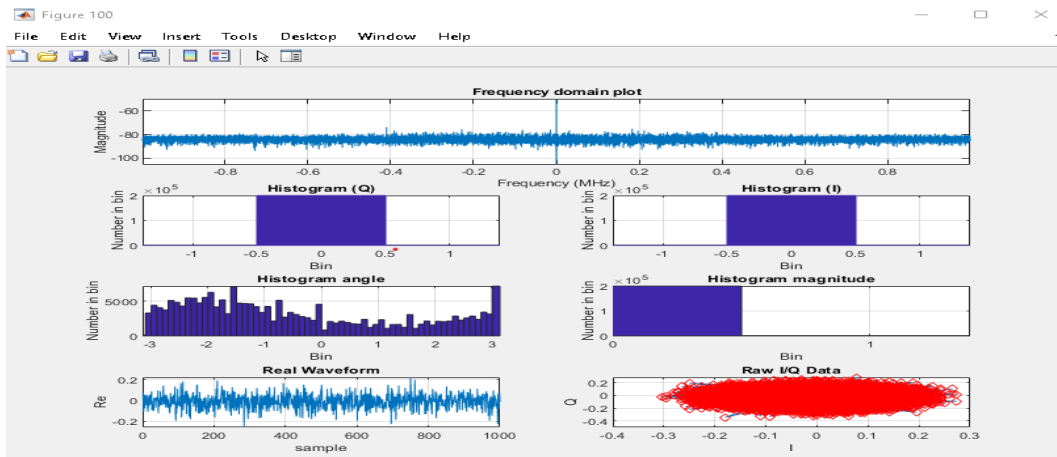


Figura 44. Entorno de MATLAB

En la terminal se muestran los satélites encontrados:

```

=====
| Channel | PRN | Frequency | Doppler | Code Offset | Status |
=====
| 1 | 14 | 2.67792e+03 | 2678 | 1792 | T |
| 2 | 20 | -1.13678e+03 | -1137 | 191 | T |
| 3 | 19 | -1.24359e+03 | -1244 | 1780 | T |
| 4 | 12 | -3.40271e+03 | -3403 | 1834 | T |
| 5 | 5 | -2.99835e+03 | -2998 | 389 | T |
| 6 | 28 | 8.16345e+02 | 816 | 1558 | T |
| 7 | 17 | -9.68933e+02 | -969 | 1934 | T |
=====

```

Figura 45. Entorno de MATLAB

Al graficar la energía de los satélites se obtiene la siguiente figura:

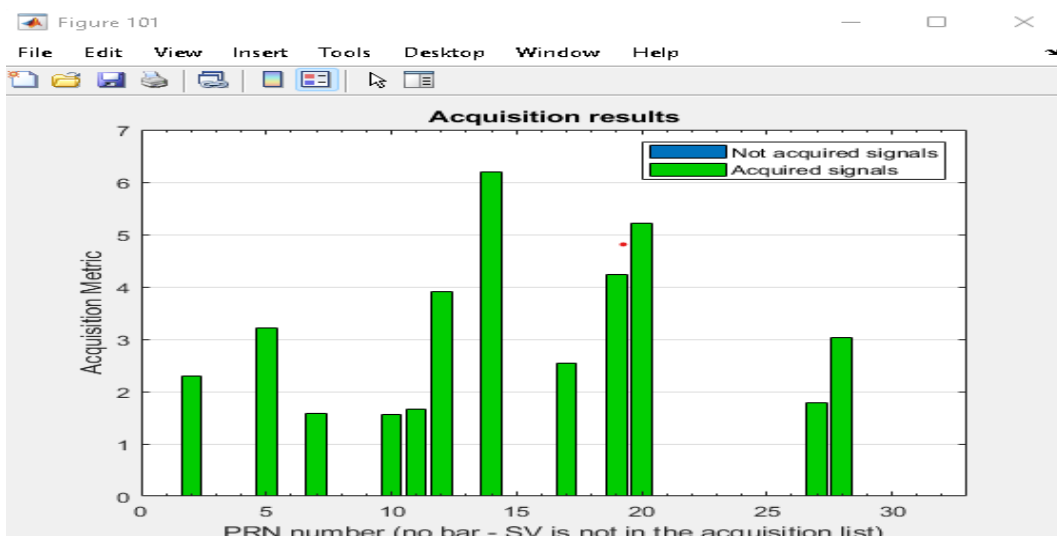


Figura 46. Entorno de MATLAB

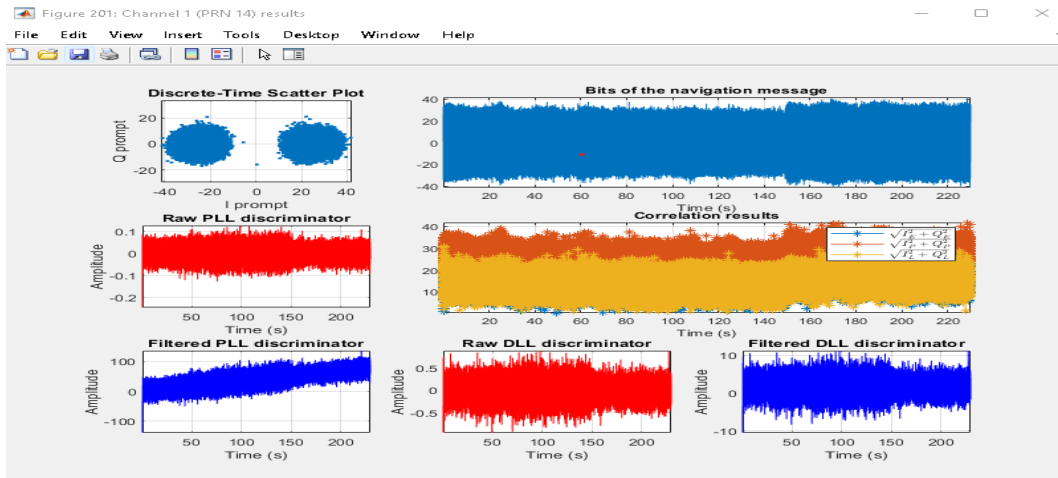


Figura 47. Entorno de MATLAB

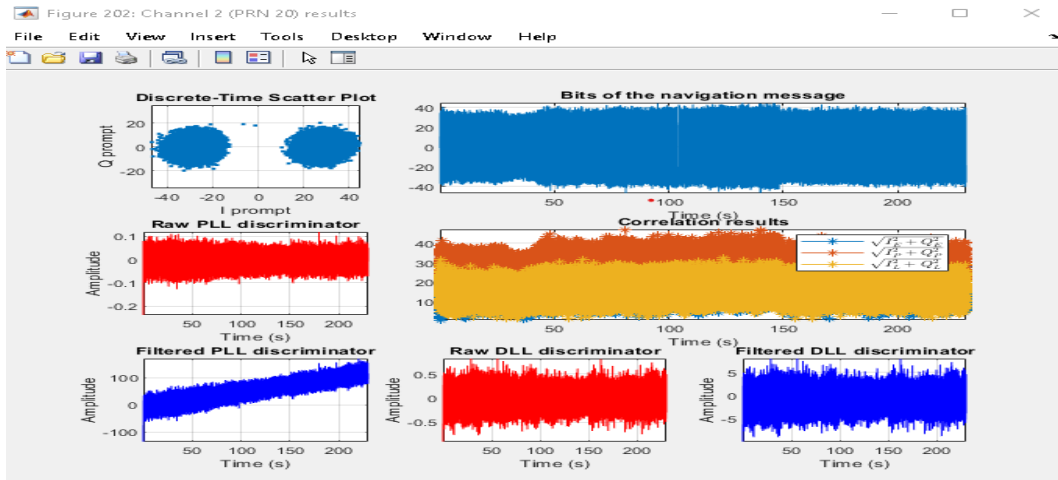


Figura 48. Entorno de MATLAB

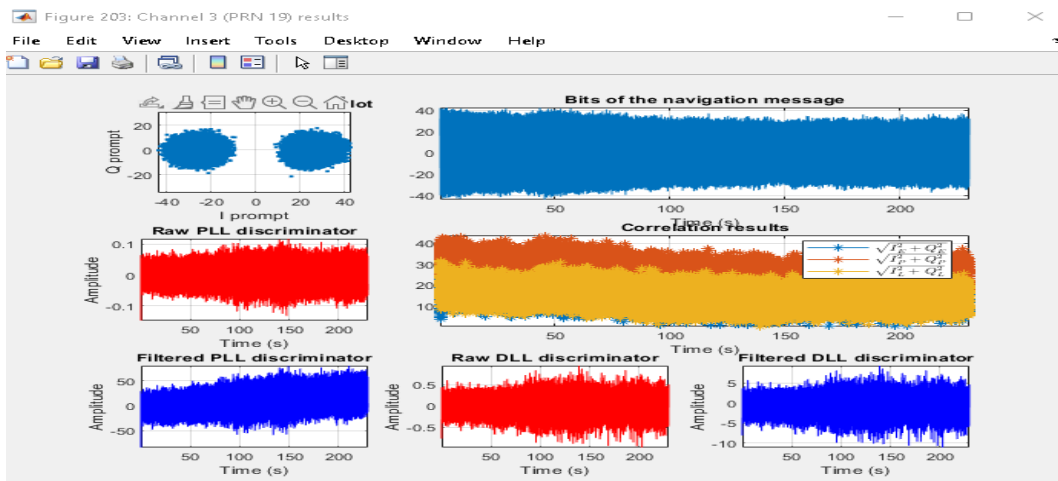


Figura 49. Entorno de MATLAB

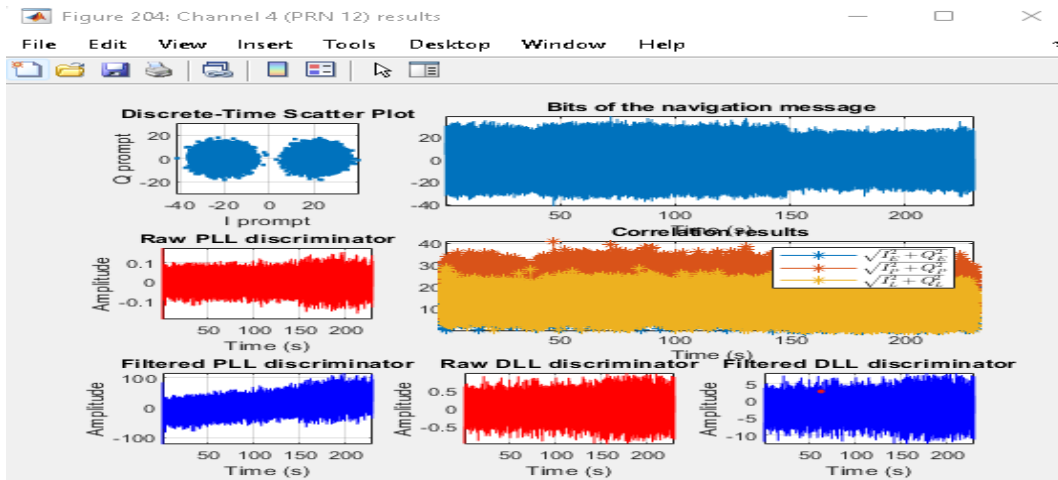


Figura 50. Entorno de MATLAB

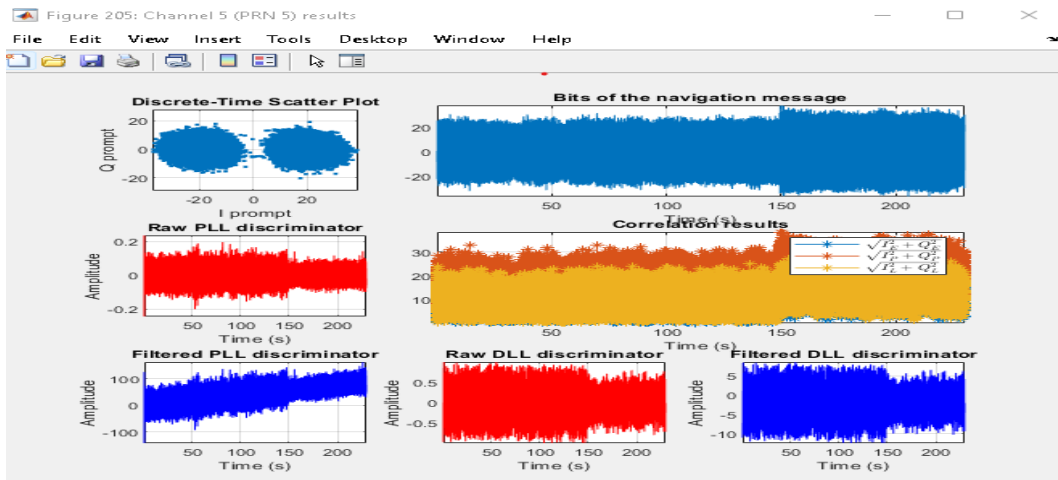


Figura 51. Entorno de MATLAB

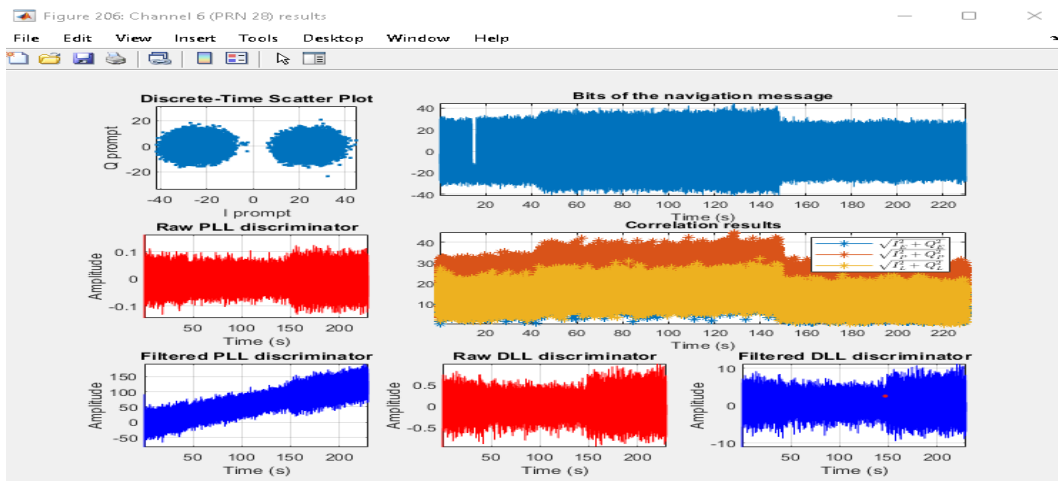


Figura 52. Entorno de MATLAB

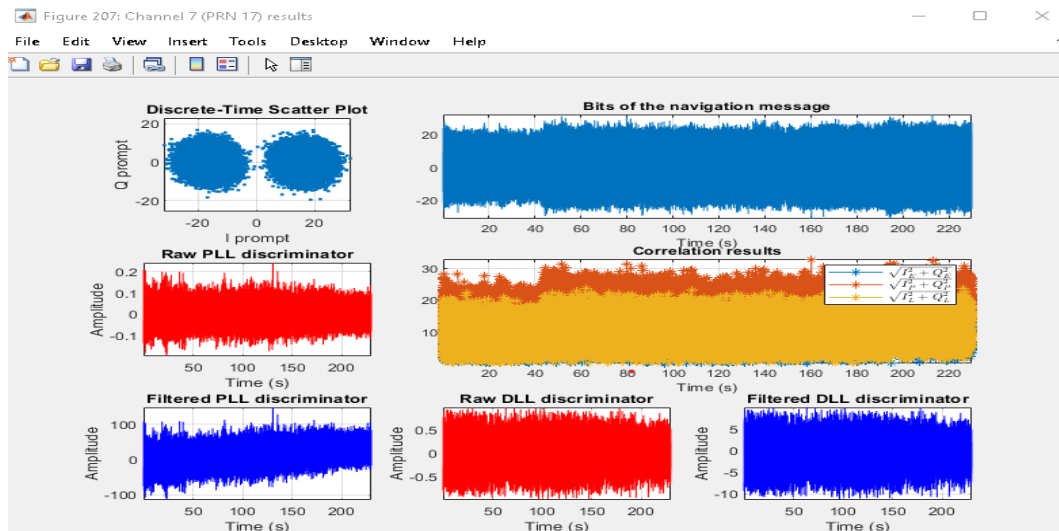


Figura 53. Entorno de MATLAB

El resultado final del programa SoftGNSS V3.0 se obtiene la siguiente ventana:

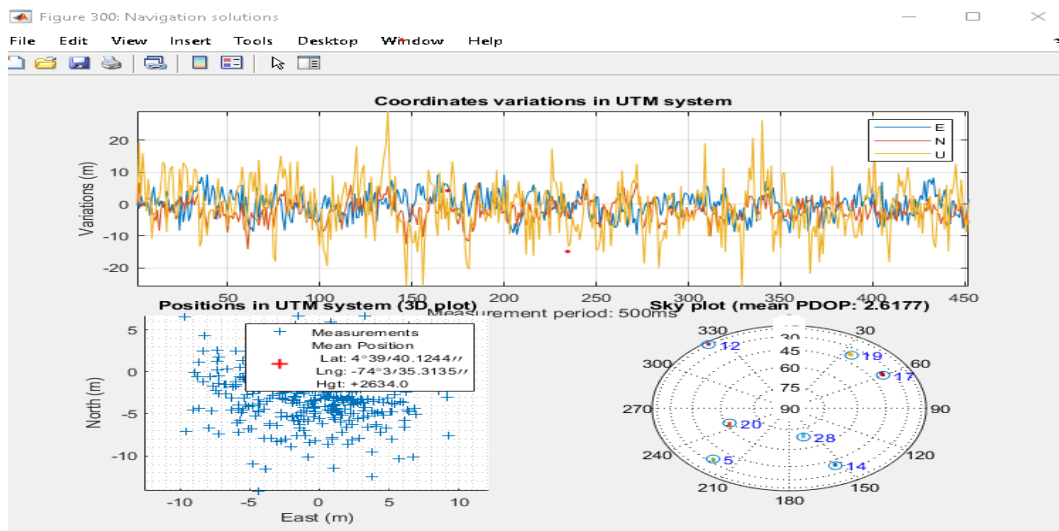


Figura 54. Entorno de MATLAB

## Referencias

- [1] Blossom, E. (2004). GNU radio: tools for exploring the radio frequency spectrum. *Linux journal*, 2004(122), 4.
- [2] Como crear un bloque en Gnu Radio.docx. (s/f). Google Docs. Recuperado el 17 de julio de 2022, de <https://docs.google.com/document/d/14Sa9hfC3vMA1BOWZr8pqLB1qh0AeWV3C/edit>
- [3] Fernández-Prades, C. (2022, julio 16). Overview. GNSS-SDR. <https://gnss-sdr.org/docs/overview/>
- [4] ¿Qué es GNU Radio? (s/f). Security Hack Labs. Recuperado el 17 de julio de 2022, de <https://securityhacklabs.net/articulo/que-es-gnu-radio>
- [5] Hier blocks and parameters. (s/f). Gnuradio.org. Recuperado el 17 de julio de 2022, de [https://wiki.gnuradio.org/index.php?title=Hier\\_Blocks\\_and\\_Parameters](https://wiki.gnuradio.org/index.php?title=Hier_Blocks_and_Parameters)
- [6] Como crear un bloque en Gnu Radio.docx. (s/f). Google Docs. Recuperado el 17 de julio de 2022, de <https://docs.google.com/document/d/14Sa9hfC3vMA1BOWZr8pqLB1qh0AeWV3C/edit>
- [7] MATLAB. (s/f). Mathworks.com. Recuperado el 17 de julio de 2022, de <https://la.mathworks.com/products/matlab.html>
- [8] Las últimas antenas GPS/GNSS hacen que el posicionamiento en centímetro sea una realidad. 2020. Retrieve October 25, 2021, from <https://www.digikey.com/es/blog/latest-gps-gnss-antennas-making-centimeter-positioning-a-reality>
- [9] Rodriguez, J. (2017). An. Análisis software y hardware del SDR HackRF One. Recuperado el 2022, de [https://digibug.ugr.es/bitstream/handle/10481/48019/RodriguezHaro\\_PFC\\_SDR\\_HackRF.pdf;jsessionid=144FFD6511FB03A9C52D53BA509D33F3?sequence=1](https://digibug.ugr.es/bitstream/handle/10481/48019/RodriguezHaro_PFC_SDR_HackRF.pdf;jsessionid=144FFD6511FB03A9C52D53BA509D33F3?sequence=1)
- [10] Fernández-Prades, C. (2022, julio 17). My first position fix. GNSS-SDR. <https://gnss-sdr.org/my-first-fix/>
- [11] Bibliografía El entorno de trabajo matlab. (s/f). Ulpge.es. Recuperado el 19 de julio de 2022, de [https://estadistica-dma.ulpge.es/FCC/matlab-0-Entorno\\_de\\_trabajo.html](https://estadistica-dma.ulpge.es/FCC/matlab-0-Entorno_de_trabajo.html)
- [12] RS AI94. (2020). gnuradio function probe part2 — frequency sweep [Video]. Recuperado de <https://www.youtube.com/watch?v=9bhmF7WRvMQ>
- [13] Python module. (s/f). Gnuradio.org. Recuperado el 22 de julio de 2022, de [https://wiki.gnuradio.org/index.php/Python\\_Module](https://wiki.gnuradio.org/index.php/Python_Module)
- [14] Variable. (s/f). Gnuradio.org. Recuperado el 22 de julio de 2022, de <https://wiki.gnuradio.org/index.php?title=Variable>
- [15] Probe signal. (s/f). Gnuradio.org. Recuperado el 22 de julio de 2022, de [https://wiki.gnuradio.org/index.php?title=Probe\\_Signal](https://wiki.gnuradio.org/index.php?title=Probe_Signal)
- [16] HackRF one - “tiny TCXO” 10MHz 0.5PPM TCXO module. (s/f). Nooelec.com. Recuperado el 23 de julio de 2022, de <https://www.noelec.com/store/tiny-tcxo.html>
- [17] Arágones, N. (2019). Instalación de bloque RTL-SDR y Osmocom en GNURadio. [Video]. Recuperado de <https://www.youtube.com/watch?v=JXaVj1MjDqw>
- [18] Function probe. (s/f). Gnuradio.org. Recuperado el 24 de julio de 2022, de [https://wiki.gnuradio.org/index.php/Function\\_Probe](https://wiki.gnuradio.org/index.php/Function_Probe)
- [19] Fernández-Prades, C. (2022, julio 31). Signal Processing Blocks. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/>
- [20] Fernández-Prades, C. (2022b, julio 31). Signal Source. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/signal-source/>
- [21] Fernández-Prades, C. (2022, julio 31). Signal Conditioner. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/signal-conditioner/>
- [22] Fernández-Prades, C. (2022, julio 31). Channels. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/channels/>
- [23] Fernández-Prades, C. (2022, julio 31). Acquisition. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/acquisition/>
- [24] Fernández-Prades, C. (2022, julio 31). Tracking. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/tracking/>
- [25] Fernández-Prades, C. (2022, julio 31). Telemetry Decoder. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/telemetry-decoder/>
- [26] Fernández-Prades, C. (2022, julio 31). Observables. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/observables/>
- [27] Fernández-Prades, C. (2022, julio 31). PVT. GNSS-SDR. <https://gnss-sdr.org/docs/sp-blocks/pvt/>