

# Laboratorios SDR con GNU Radio

## Creación Colectiva

IPA2020

Riaño Walter, Barragan Wilmar, Montana Camilo, Giraldo Martin, Alfonso Andres, Hellcenk Murillo, Cifuentes Jorge, Lizcano Gonzalo, Marín Laura, Torres Omar, Guayara Camila, Firigua Lorena

IPA 2018

Amaya Lina, Ávila Daniel, Beltrán Camilo, Bonilla William, Casallas Santiago, Cruz Johan, Cubillos Neil, Flórez Jenny, Gómez Daniel, Hernández Cristian, López Lizeth, López Catalina, Malaver Cristian, Otálora Brayan, Puentes Carlos, Trujillo Santiago, Vargas Carlos.

IIPA 2017

Achury Brian, Caicedo William, Contreras Dimitri, Cruz Ángela, Flórez Jhonattan, Flórez Santiago, Gaitán Jorge, García Steven, Hernández Eduard, Leal Sebastián, Ledesma Yerson, Loaiza Andrés, Muñoz Melissa, Navarro Cristian, Ricardo Karen, Román María, Roncancio Jonhattan, Salamanca Camila, Sanabria Luisa, Sánchez David, Silva Daniel, Tobón Brayan, Vargas Alejandro, Vásquez Camilo, Vera Ángel, Zamora José.  
IPA 2017

Baquero Miguel, Buitrago Miguel, Campos Álvaro, Contreras Michael, Espinosa Kevin, Mahecha Jonathan, Moreno Santiago, Muñoz Alejandra, Rodríguez Anderson, Santamaría Cristian.

## Tutor

Rodríguez Mújica Leonardo

Facultad de Ingeniería  
Universidad de Cundinamarca

# Contenidos generales

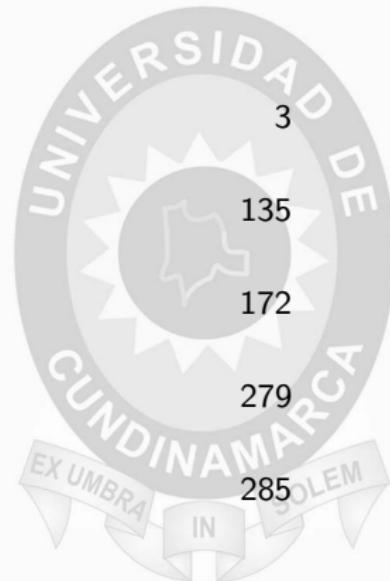
LABORATORIOS CON SOFTWARE

CONFIGURACIÓN E INSTALACIÓN DE HARDWARE

LABORATORIOS CON SOFTWARE Y HARDWARE

BIBLIOGRAFÍA

SOLUCIONES



# LABORATORIOS CON SOFTWARE

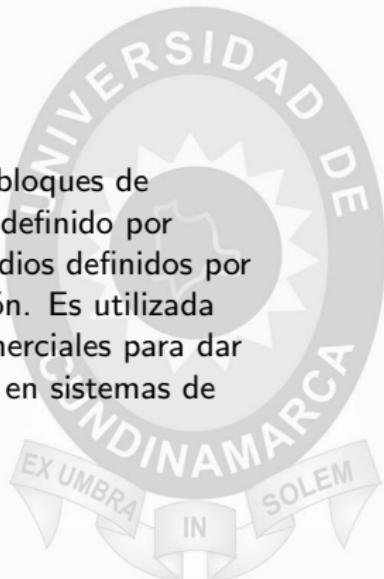
# Parte I - Tabla de contenidos

Introducción a GNU Radio	5
Lab1: Primeros pasos	17
Lab2: Osciloscopio y FFT	53
Lab3: Audio	79
Lab4: Modulación ASK en GRC	102
Lab5: Modulación BPSK en GRC	118
Lab6: Interfaces gráficas de WX GUI	129

# Introducción a GNU Radio

# ¿Qué es GNU Radio?

Es una herramienta de desarrollo libre y abierta que provee bloques de procesamiento de señal para implementar sistemas de radio definido por software. Puede utilizarse con hardware de RF para crear radios definidos por software o sin hardware para crear un ambiente de simulación. Es utilizada extensivamente en ambientes académicos, aficionados y comerciales para dar soporte a la investigación en comunicaciones inalámbricas y en sistemas de radio en el mundo.



# Aplicaciones

## RASTREO SATELITAL



## REDES DIGITALES

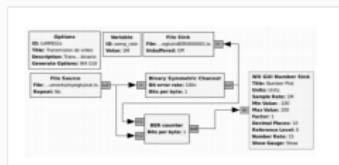


## SISTEMAS DE RADAR

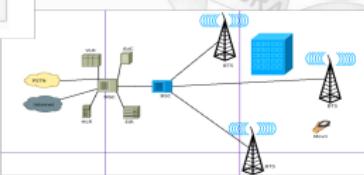


## AUDIO PROCESAMIENTO

## comunicaciones móviles



## REDES GSM



# Instalación de GNU Radio en Linux

Para instalar GNU Radio se deben seguir los siguientes pasos:

1. Ingresar a la ventana de órdenes (o terminal) del sistema de su equipo.
2. Estando conectado a internet, escriba dentro del terminal:

```
sudo apt-get install gnuradio
```

3. Si su dispositivo tiene contraseña, debe ingresarla, al ser solicitada y oprimir .
4. Luego se deben aceptar los términos de la instalación oprimiendo la letra **s** seguido de .
5. Una forma de verificar la correcta instalación es volviendo a ingresar la orden indicada en el punto 2, y si aparece un mensaje anunciando que GNU Radio ya está en su versión más reciente, su instalación fue correcta.

# Paquetes

Con el objetivo de clonar el repositorio y obtener los ejemplos de GNU Radio en nuestro ordenador se deben instalar los siguientes paquetes:

- build-essential

Build essential es un paquete que contiene herramientas necesarias para la creación, compilación e instalación de programas.

- cmake

Es un sistema de construcción de código abierto multiplataforma. Se trata de un conjunto de herramientas diseñadas para construir, testear y empaquetar software. Se utiliza para controlar el proceso de compilación de software utilizando una plataforma sencilla y unos archivos de configuración independientes del compilador.

# Paquetes

- git

Este paquete contiene un sistema de control de versiones distribuidas de código abierto desarrollado originalmente por Linux Torvalds para apoyar el desarrollo del kernel de Linux.

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se puedan recuperar versiones específicas más adelante.

- libboost-all-dev

Boost es un conjunto de bibliotecas para el lenguaje de programación C++ que suministra un apoyo para tareas y estructuras como álgebra lineal, generación de números pseudoaleatorios, procesamiento de imágenes, expresiones regulares y pruebas unitarias. En el momento contiene 162 bibliotecas individuales.

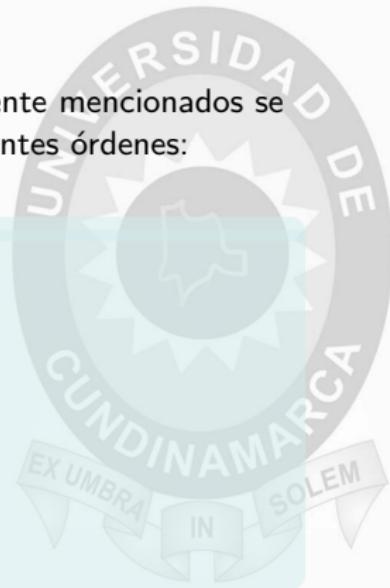
# Paquetes

- `libcppunit-dev`
  - Biblioteca de pruebas unitarias para C++.
  - Una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo, en diseño estructurado o en diseño funcional, una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correcta y eficientemente por separado.
- `doxygen` Es una herramienta para generar documentación a partir de código fuente. Es un sistema de documentación para C++, C, Java, Python. Es necesario solo si se desea generar referencias a documentación externa de la que no tiene las fuentes.

# Instalación de paquetes

1. La instalación de cada uno de los paquetes anteriormente mencionados se realiza colocando en la ventana de terminal, las siguientes órdenes:

```
sudo apt-get install build-essential  
sudo apt-get install cmake  
sudo apt-get install git  
sudo apt-get install libboost-all-dev  
sudo apt-get install libcppunit-dev  
sudo apt-get install doxygen
```



## Clonar repositorio

El código fuente de los ejemplos está almacenado en github, por lo tanto para clonar el repositorio se debe realizar lo siguiente:

- Abrir la ventana de órdenes o terminal.
- Despues se debe ingresar la siguiente orden para clonar el directorio git:

```
git clone https://github.com/gnuradio/gr-tutorial
```

- Una vez clonado el directorio, gr-tutorial, en el PC empleado se deben ver exactamente los mismos archivos y carpetas que los del repositorio github.

## Instalación de módulos

- Luego de haber clonado el repositorio, debemos buscar la carpeta “**gr-tutorial**” e ingresar a ella desde el terminal, para ello se digitán las siguientes órdenes:

```
ls
```

```
cd gr-tutorial
```

Es importante mencionar que al escribir la primera orden se podrán observar la diferentes carpetas que se encuentran en el dispositivo, por lo tanto “**gr-tutorial**” debe aparecer entre las opciones para poder cambiar de directorio.

# Instalación de módulos

- Estando dentro de la carpeta, desde la terminal, se deben escribir las siguientes órdenes, con la finalidad de instalar las soluciones o módulos:

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make -j8
```

```
sudo make install
```

```
sudo ldconfig
```



# Resultado

En el directorio examples encontrará los tutoriales:

```
tutorial1  
tutorial2  
...  
tutorial7
```

Puede explorar los ejemplos de los desarrolladores de GNU Radio



# Lab1

## Primeros pasos

# Primeros pasos

The screenshot shows the GNU Radio Companion (GRC) application window. On the left, there's a toolbar with various icons. The main area contains a flow graph with several blocks connected by arrows. A 'Signal Source' block is connected to a 'Throttle' block, which is then connected to a 'WX GUI Scope Sink' block. To the left of the graph, there are three variable blocks: 'Variable ID: samp\_rate Value: 32k', 'Variable ID: my\_var Value: 11', and 'WX GUI Slider ID: freq Label: frequency Default Value: 1k Minimum: 0 Maximum: 16k Converter: Float'. A 'List of blocks' panel on the right lists categories such as [Audio], [Boolean Operators], etc. At the bottom, a terminal window shows the command line and file paths.

**Arrastra los bloques de la lista hacia el lienzo**

**Lista de bloques: presione CTRL + F para buscar por nombre.**

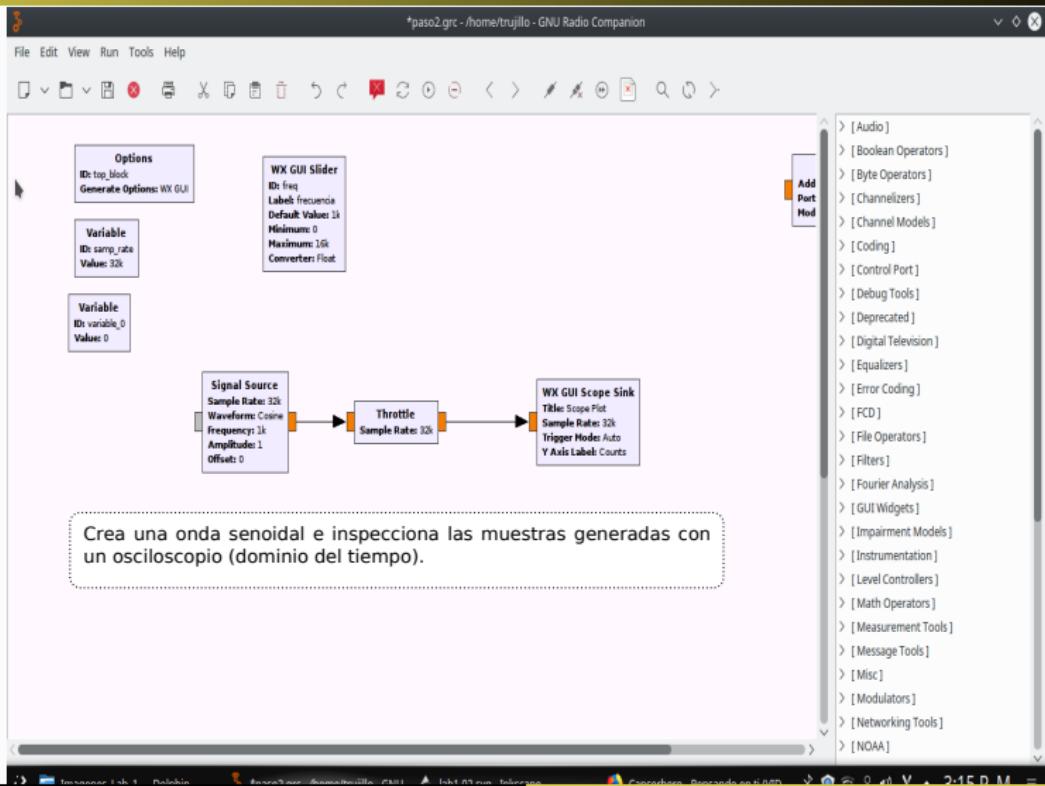
**Lienzo (área de construcción del diagrama de flujo)**

**Conecte los puertos haciendo clic en el puerto elegido de un bloque y, a continuación, haga clic en el puerto del otro bloque. Puede eliminar las conexiones haciendo clic en la línea de la conexión y presionando la tecla Suprimir.**

**Ventana de registros: terminal del programa**

```
/usr/share/gnuradio/grc(blocks
/home/kamila/.grc_gnuradio
Loading: "/home/kamila/lab1.grc"
>>> Done
Showing: "/home/kamila/lab1.orc"
```

# Primeros pasos



# Primeros pasos

The screenshot shows the GNU Radio Companion interface with a flow graph. A 'Options' block is selected, and its properties are displayed in the 'Properties: Options' dialog. The 'General' tab is active, showing fields like ID (top\_block), Title, Author, Description, Canvas Size, Generate Options (set to 'WX GUI'), Run (set to 'Autostart'), Max Number of Output (set to 0), and Realtime Scheduling (set to Off). A tooltip '(Doble clic)' points to the 'Generate Options' dropdown. A note at the bottom states: 'El bloque "opciones" se utiliza para establecer parámetros globales.' To the right is a sidebar with categories like Audio, Boolean Operators, Byte Operators, etc.

\*paso2.grc - /home/trujillo - GNU Radio Companion

File Edit View Run Tools Help

Properties: Options

General Advanced Documentation

ID: top\_block

Title

Author

Description

Canvas Size

Generate Options: WX GUI

Run: Autostart

Max Number of Output: 0

Realtime Scheduling: Off

(Doble clic)

Variable: variable\_0 Value: 0

Signal Source Sample Rate: 32k Waveform: Cosine Frequency: 1k Amplitude: 1 Offset: 0

El bloque "opciones" se utiliza para establecer parámetros globales.

Add Part Mod

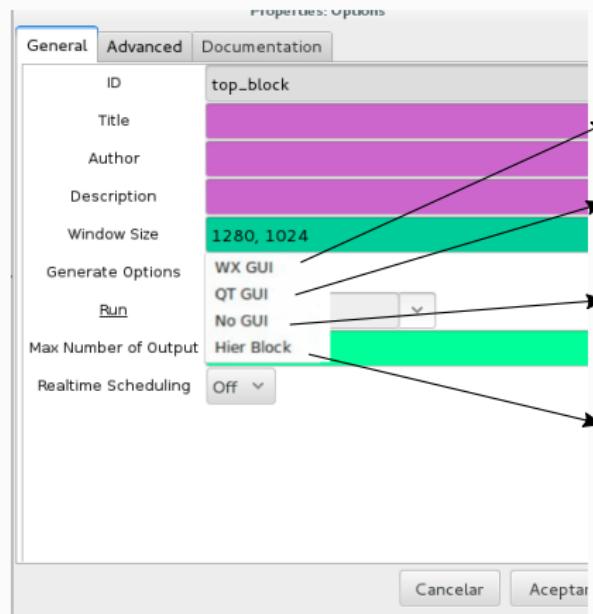
Audio Boolean Operators Byte Operators Channelizers Channel Models Coding Control Port Debug Tools Deprecated Digital Television Equalizers Error Coding FCD File Operators Filters Fourier Analysis GUI Widgets Impairment Models Instrumentation Level Controllers Math Operators Measurement Tools Message Tools Misc Modulators Networking Tools NOAA

# Primeros pasos

Properties: Options

General	Advanced	Documentation
ID	top_block	Nombre del archivo generado en Python
Title		Título de la ventana GUI principal o nombre del bloque jerárquico
Author		
Description		
Window Size	1280, 1024	Tamaño del lienzo de GRC
Generate Options	No GUI	Tipo de código a generar (ver siguiente)
Run Options	Prompt for Exit	Cómo iniciar y detener el diagrama de flujo.
Max Number of Output	0	Avanzado: limita el número de muestras de salida de iteración de la función de trabajo de cada bloque.
Realtime Scheduling	Off	Si el código se ejecuta como "root" (por ejemplo con "sudo") solicita al núcleo del SO que priorice este proceso.

# Primeros pasos



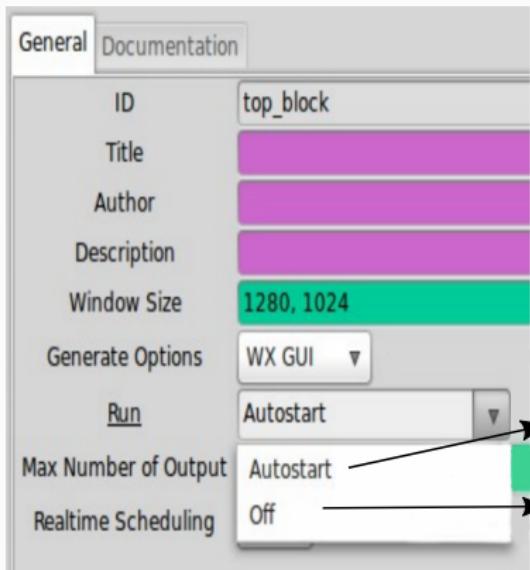
Aplicación de la GUI que utiliza el kit de herramientas WX (utiliza bloques GUI WX).

Aplicación GUI que utiliza el kit de herramientas QT (utiliza bloques GUI Qt).

Aplicación de la línea de órdenes sin GUI (basada en texto, ejecutada en una consola).

Cree un bloque jerárquico que aparezca en la lista de bloques (un componente reutilizable, no una aplicación: use los bloques Pad Source/Sink para exponer puertos y los bloques paramétricos para exponer variables de configuración).

## Primeros pasos



# Primeros pasos

General Documentation

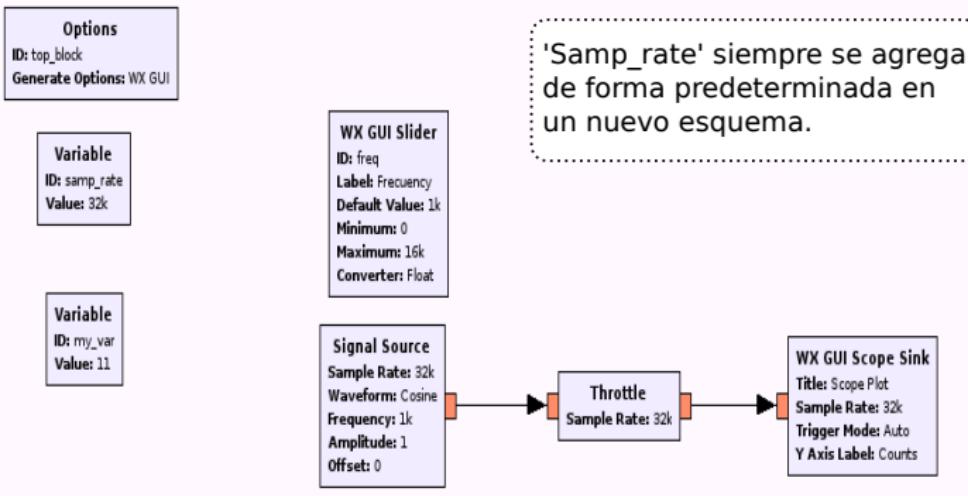
ID	top_block
Title	
Author	
Description	
Window Size	1280, 1024
Generate Options	No GUI ▾
Run Options	Run to Completion
Max Number of Output	Prompt for Exit
Realtime Scheduling	Off ▾

Saldrá automáticamente cuando termine

Presionando ENTER saldrá



# Primeros pasos



'Samp\_rate' siempre se agrega de forma predeterminada en un nuevo esquema.

Variable: un bloque que contiene una expresión de Python arbitraria.

# Primeros pasos

The image shows a screenshot of the GNU Radio Block Diagram Editor. On the left, a flowgraph is displayed with several blocks: an 'Options' block, a 'Variable' block (ID: samp\_rate, Value: 32k), another 'Variable' block (ID: my\_var, Value: 11), a 'WX GUI Slider' block, and a 'Signal Source' block. A callout bubble points from the 'my\_var' block to a tooltip containing the text '(Doble clic)'. An arrow points from the 'Value' field of the 'samp\_rate' variable in the configuration dialog to the same tooltip.

**General Advanced Documentation**

**ID:** samp\_rate  
**Value:** 32000

**ID:** (Python) nombre de la variable.  
**Valor:** Expresión de Python arbitraria  
Ejemplo:  
3200: (el valor predeterminado) un entero  
32e6: 32000.0 (número de punto flotante).  
int(32e3): 32000 (conversión a int de un número a punto flotante)

Cancelar Aceptar

# Primeros pasos

**Options**  
 ID: top\_block  
 Generate Options: WX GUI

**Variable**  
 ID: samp\_rate  
 Value: 32k

**Variable**  
 ID: my\_var  
 Value: 11

(Doble clic)

**WX GUI Slider**  
 ID: freq  
 Label: Frecuency  
 Default Value: 1k  
 Minimum: 0  
 Maximum: 16k  
 Converter: Float

**Signal Source**  
 Sample Rate: 32k  
 Waveform: Cosine  
 Frequency: 1k  
 Amplitude: 1  
 Offset: 0

'my\_var' es sólo para mostrar aquí (en realidad no hace nada útil en este diagrama)

Otro ejemplo de una simple expresión de Python arbitraria.  
 Coloque el cursor sobre cualquier recuadro con una expresión matemática y verá la descripción emergente con el resultado (aquí  $5 + 6 = 11$ ).  
**Nota:** las expresiones arbitrarias sólo pueden escribirse en campos que tengan un fondo blanco (campos 'abiertos').

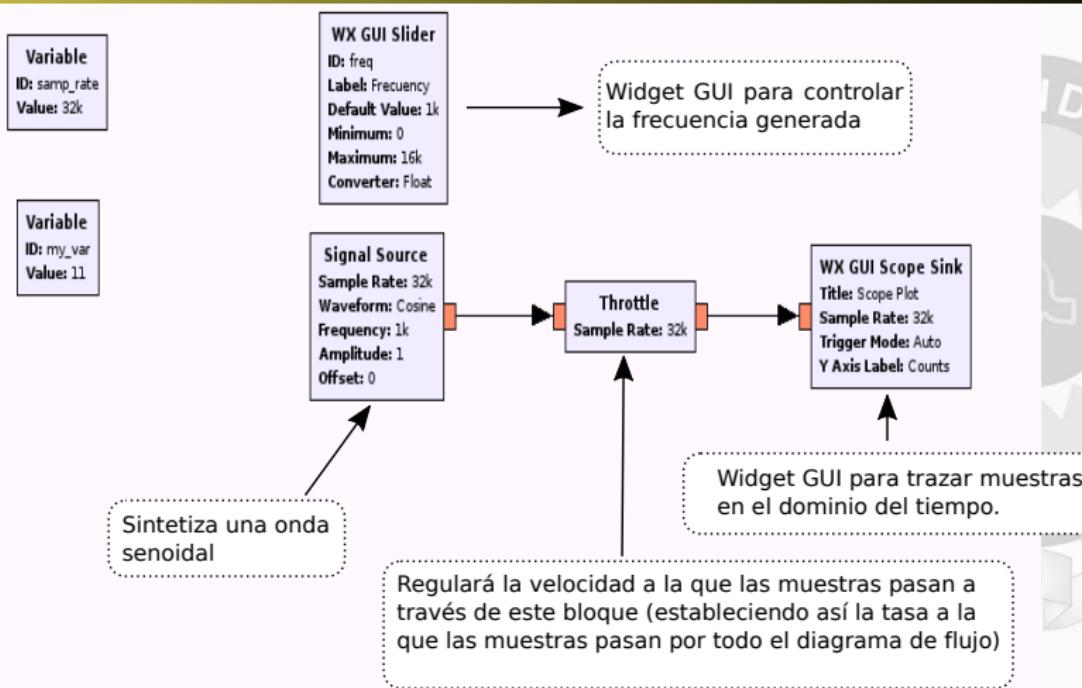
Properties: Variable

General Advanced Documentation

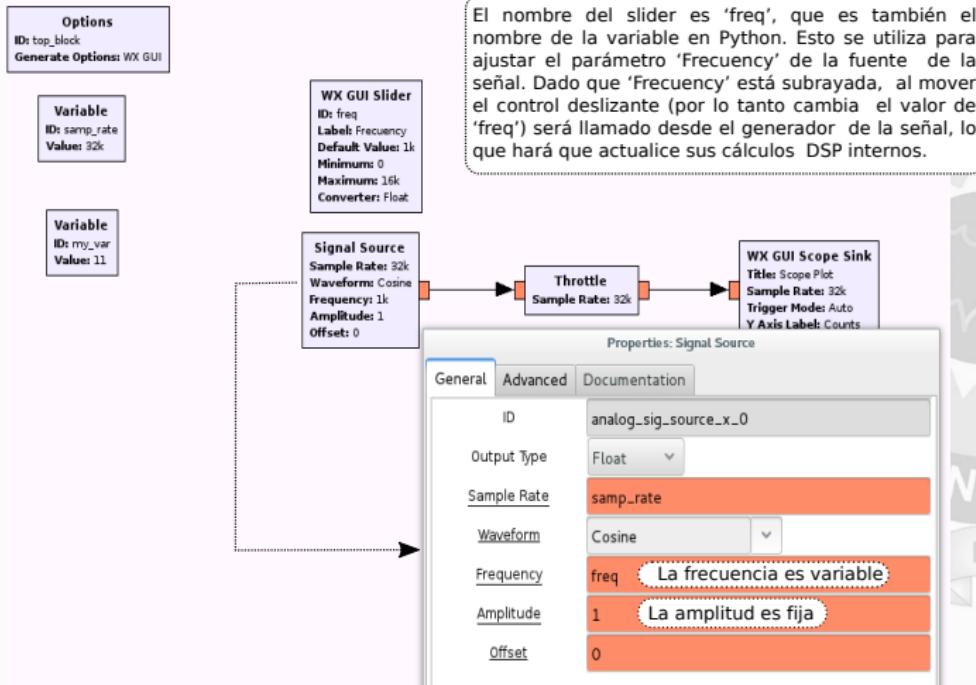
ID: my\_var  
 Value: 5+6

Cancelar Aceptar

# Primeros pasos



# Primeros pasos



# Primeros pasos

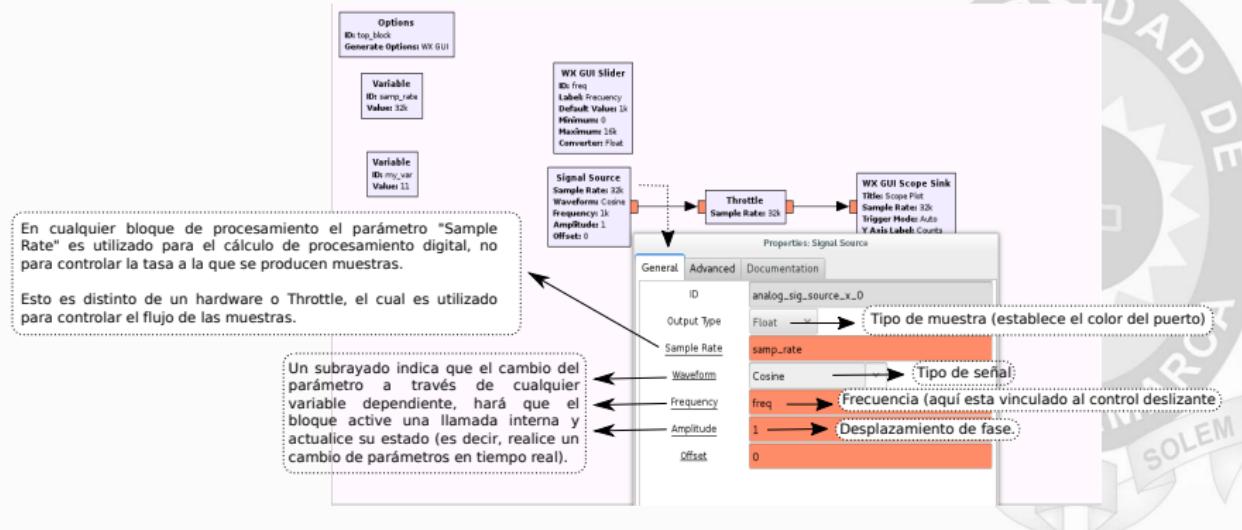
The screenshot shows a software interface for configuring a GNU Radio block diagram. On the left, there are several configuration panels:

- Options**:  
IDs: top\_block  
Generate Options: WX GUI
- Variable**:  
ID: samp\_rate  
Value: 32k
- Variable**:  
ID: my\_var  
Value: 11
- Signal Source**:  
Sample Rate: 32k  
Waveform: Cosine  
Frequency: 1k  
Amplitude: 1  
Offset: 0
- WX GUI Slider**:  
ID: freq  
Label: Frequency  
Default Value: 1k  
Minimum: 0  
Maximum: 16e3  
Converter: Freq

A dashed arrow points from the "WX GUI Slider" panel to a "Properties: WX GUI Slider" dialog box on the right. This dialog has tabs for General, Advanced, and Documentation. The General tab is selected and contains the following fields:

Parameter	Value	Description
ID	freq	
Label	Frequency	Etiqueta junto al widget en la GUI
Default Value	1e3	1000.0 en notación científica
Minimum	0	Frecuencia mínima del slider
Maximum	16e3	Frecuencia máxima del slider
Num Steps	1000	El parámetro 'freq' debería ser un número de punto flotante o un número entero.
Style	Horizontal	
Converter	Float	
Grid Position		
Notebook		

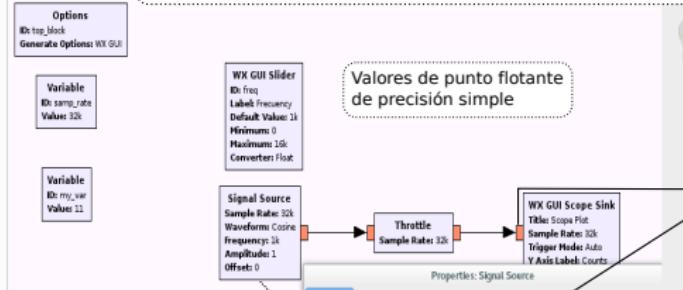
# Primeros pasos



# Primeros pasos

El color de un puerto indica el tipo de muestras que fluyen a través de él. Los colores también se aplican a los parámetro de los bloques.

Types

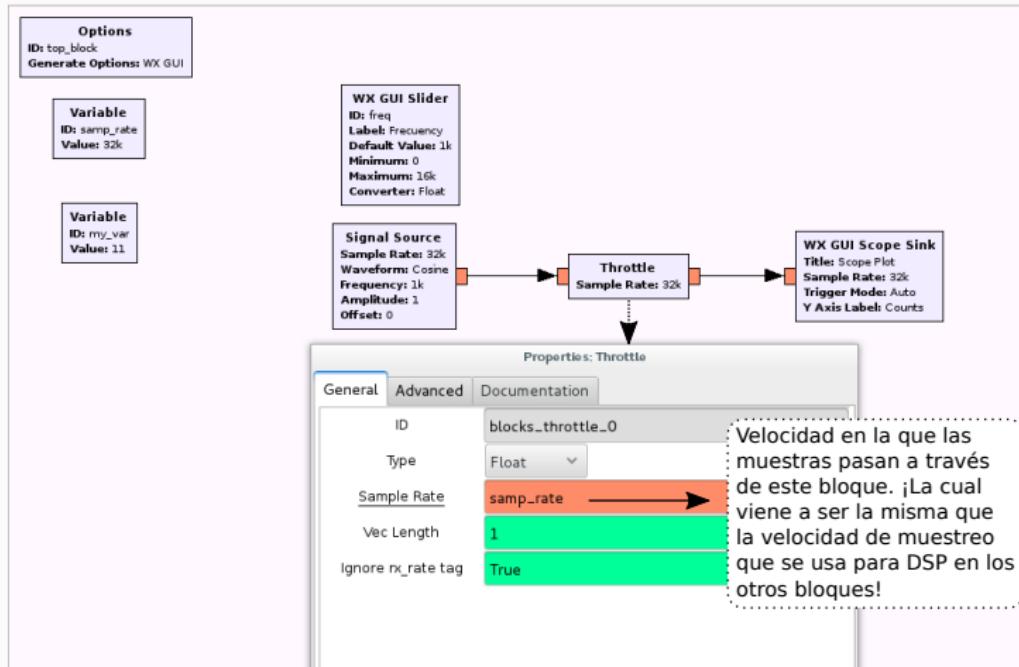


Color Mapping

Complex Float 64
Complex Float 32
Complex Integer 64
Complex Integer 32
Complex Integer 16
Complex Integer 8
Float 64
Float 32
Integer 64
Integer 32
Integer 16
Integer 8
Message Queue
Async Message
Bus Connection
Wildcard

Consejo:  
 Después de hacer clic en el bloque, presione las teclas de flecha arriba / abajo para cambiar el tipo de muestras.  
 (En las opciones de bloque se puede configurar manualmente, en el primer parámetro)

# Primeros pasos



# Primeros pasos

**Properties: WX GUI Scope Sink**

	General	Advanced	Documentation
ID	wxgui_scopesink2_0	! Esto es solamente para generar los pasos correctos en el eje X!	
Type	Float	'0' hará que el gráfico se auto-escala a la señal entrante. Al ingresar cualquier otro valor, se mostrará en una escala/ desplazamiento fijo en esa dimensión.	
Title	Scope Plot	<b>WX GUI Scope Sink</b> Titles: Scope Plot Sample Rate: 32k Trigger Mode: Auto Y Axis Label: Counts	
Sample Rate	samp_rate		
V Scale	0		
V Offset	0		
T Scale	0		
AC Couple	Off		
XY Mode	Off	Traza varias señales (¡pueden no estar sincronizadas cuando se dibujan!)	
Num Inputs	1		
Window Size			
Grid Position		Esto será explicado más adelante, pero por ahora revise la pestaña documentación donde se discute	
Notebook			
Trigger Mode	Auto	(*) Trazar dos flujos de tipo flotante en sincronización cambiando el tipo de ámbito a Complejo y utilizar el bloque Flotante en complejo antes.	
Y Axis Label	Counts		

# Primeros pasos

The screenshot shows the GNU Radio Companion (GRC) application window titled "Lab\_1.grc - /home/ba". The menu bar includes File, Edit, View, Build, Help, and a toolbar with various icons. A circular watermark for "UNIVERSIDAD DE CUNDINAMARCA" is visible on the right.

The flow graph consists of the following components:

- WX GUI Slider**:  
ID: freq  
Label: Frequency  
Default Value: 1k  
Minimum: 0  
Maximum: 16k  
Converter: Float
- Signal Source**:  
Sample Rate: 32k  
Waveform: Cosine  
Frequency: 1k  
Amplitude: 1  
Offset: 0
- Throttle**:  
Sample Rate: 32k

The "Signal Source" block has an "out" port connected to the "in" port of the "Throttle" block. The "Throttle" block has an "out" port.

On the left, there are three variable definitions:

- Variable**: ID: samp\_rate, Value: 32k
- Variable**: ID: my\_var, Value: 11
- Options**: ID: top\_block, Generate Options: WX GUI

Two annotations with arrows point to the top right of the GRC window:

- Genera el código Python (F5)
- Genera y ejecuta el código Python (F6)

# Primeros pasos

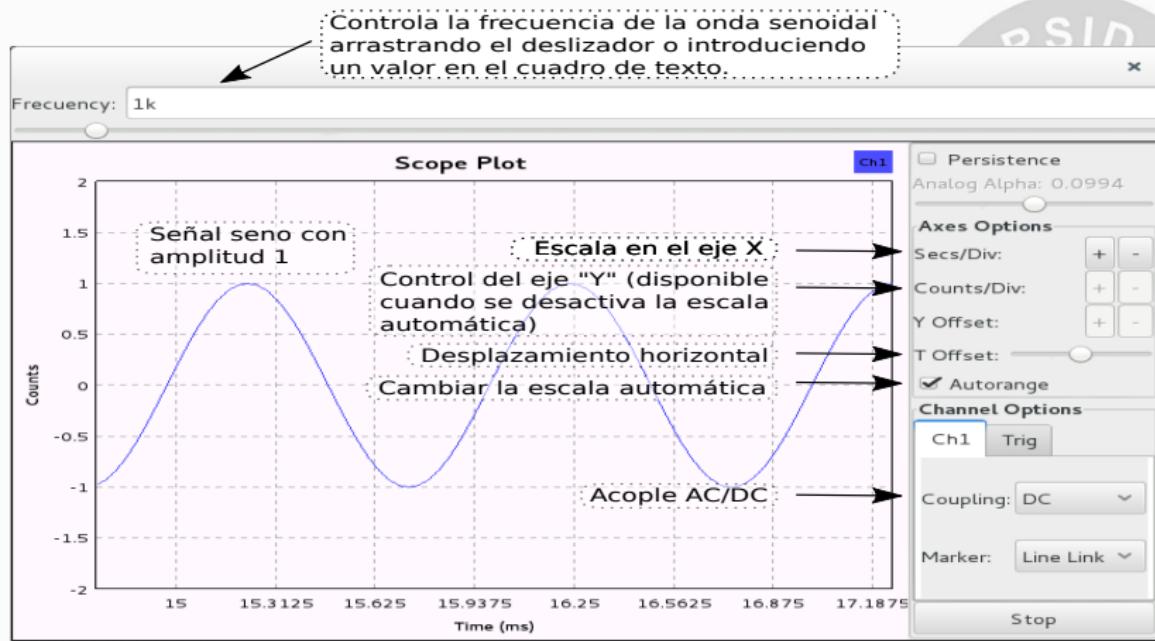
## Programa en Python generado por GRC

```

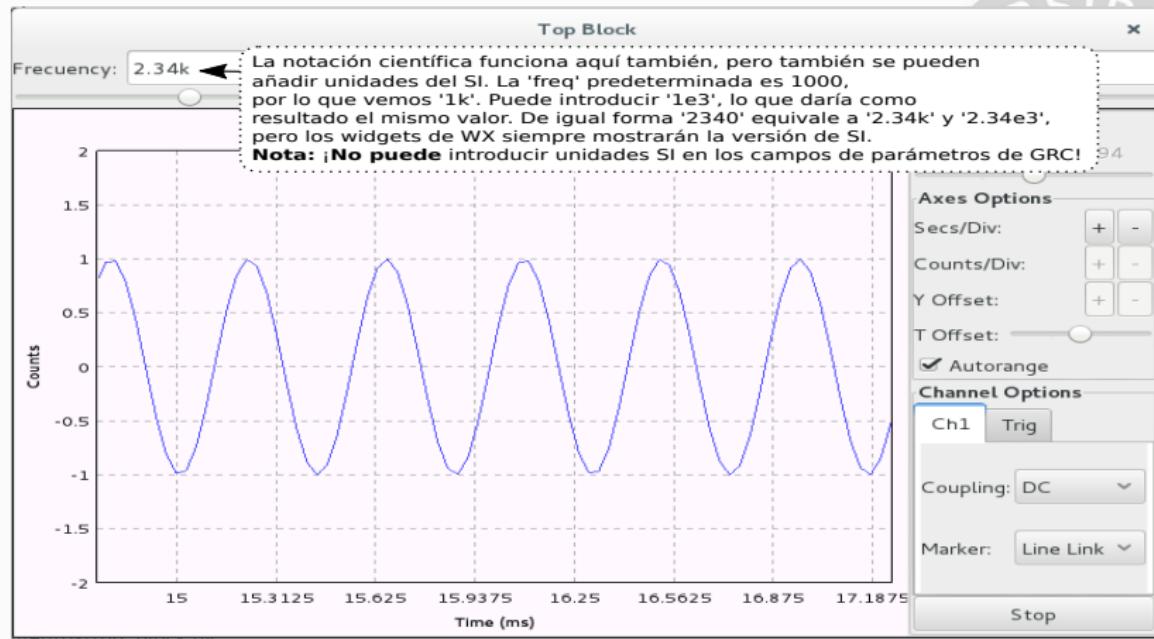
1 #!/usr/bin/env python
2 #####
3 # Gnuradio Python Flow Graph
4 # Title: Tcp Server
5 # Generated: Sun Mar 19 13:47:53 2017
6 #####
7
8 from gnuradio import eng_notation
9 from gnuradio import gr
10 from gnuradio import wxgui
11 from gnuradio.eng_option import eng_option
12 from gnuradio.filter import firdes
13 from gnuradio.wxgui import scopesink2
14 from grc.gruado import blks2 as grc.blks2
15 from grc.gruado import wxgui as grc.wxgui
16 from optparse import OptionParser
17 import wx
18
19 class tcp_server(grc.wxgui.top.block.gui):
20
21     def __init__(self):
22         grc.wxgui.top.block.gui.__init__(self, title="Tcp Server")
23         _icon_path = "/usr/share/icons/hicolor/32x32/apps/gnuradio-grc.png"
24         self.SetIcon(wx.Icon(_icon_path, wx.BITMAP_TYPE_ANY))
25
26     #####
27     # Variables
28     #####
29     self.samp_rate = samp_rate = 32000
30
31     #####
32     # Blocks
33     #####
34     self.wxgui_scopesink2_0 = scopesink2.scope_sink(f(
35         self.GetWin()),
36         title="Scope Plot",
37         sample_rate=samp_rate,
38         v_scale=0,
39         v_offset=0,
40         t_scale=0,
41         ac_couple=False,
42
43         xy_mode=False,
44         num_inputs=1,
45         trig_mode=wxgui.TRIG_MODE.AUTO,
46         y_axis_label="Counts",
47     )
48     self.Add(self.wxgui_scopesink2_0.win)
49     self.blks2.tcp.source_0 = grc.blks2.tcp.source(
50         itemsize=grc.sizeof_float * 1,
51         addr="0.0.0.0",
52         port=12345,
53         server=True,
54     )
55
56 #####
57 # Connections
58 #####
59 self.connect((self.blks2.tcp.source_0, 0), (self.
60         wxgui_scopesink2_0, 0))
61
62 def get_samp_rate(self):
63     return self.samp_rate
64
65 def set_samp_rate(self, samp_rate):
66     self.samp_rate = samp_rate
67     self.wxgui_scopesink2_0.set.sample_rate(self.samp_rate)
68
69 if __name__ == '__main__':
70     import ctypes
71     import sys
72     if sys.platform.startswith('linux'):
73         try:
74             x11 = ctypes.cdll.LoadLibrary('libX11.so')
75             x11.XInitThreads()
76         except:
77             print "Warning: failed to XInitThreads()"
78
79 parser = OptionParser(option_class=eng_option, usage="%prog %options")
80 (options, args) = parser.parse_args()
81 tb = tcp_server()
82 tb.Start(True)
83 tb.Wait()

```

## Primeros pasos

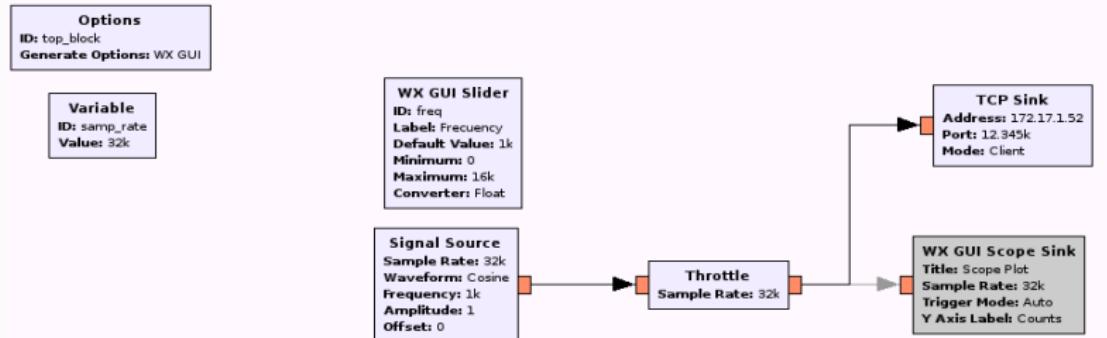


# Primeros pasos

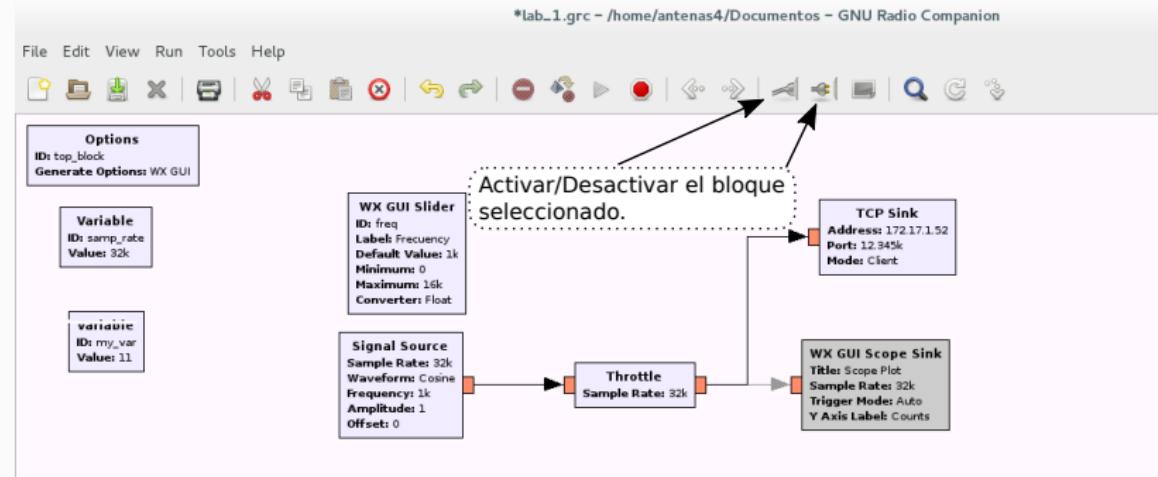


# Primeros pasos

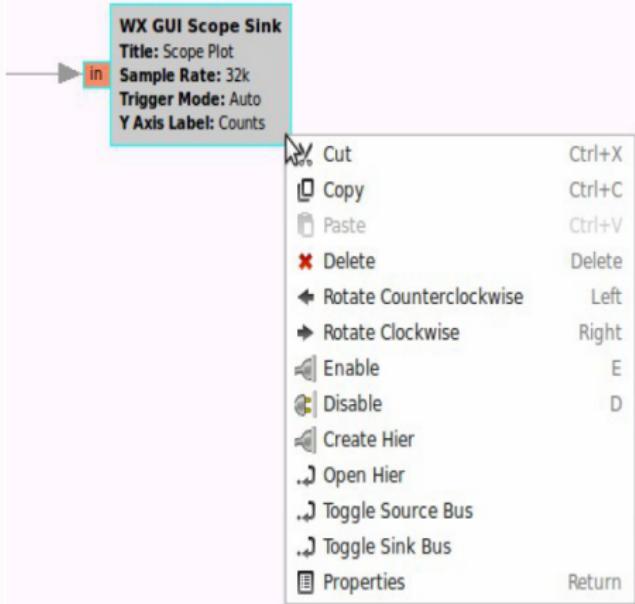
Crear una onda senoidal y transmitir las muestras generadas a través de una conexión TCP.



# Primeros pasos

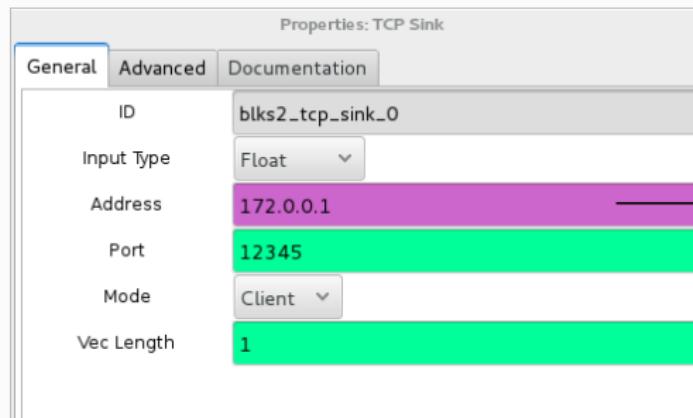


## Primeros pasos



**Consejo:**  
Tome nota de los atajos de teclado en el menú contextual del bloque.

# Primeros pasos



TCP Sink  
Address: 172.17.152  
Port: 12.345k  
Mode: Client

Se debe introducir la dirección Ip del servidor.

WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 32k  
Trigger Mode: Auto  
Y Axis Label: Counts

## Consejo:

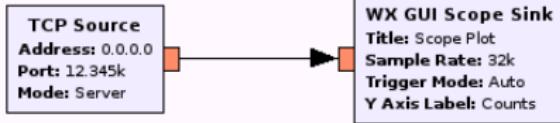
El diagrama de flujo no arrancará a menos que se establezca una conexión TCP. Si falla la conexión TCP, se produce una excepción de Python y el programa no arranca.

\* La implementación de cliente / servidor TCP actual no funciona en Windows.

# Primeros pasos

**Options**  
**ID:** tcp\_server  
**Generate Options:** WX GUI

**Variable**  
**ID:** samp\_rate  
**Value:** 32k



Recibe muestras de una conexión TCP entrante y las dibuja en un osciloscopio.

\* Se debe crear un nuevo archivo

# Primeros pasos

The screenshot shows the Gqrx software interface. A TCP Source block is connected to a WX GUI Scope Sink block. The TCP Source block has its Address set to 0.0.0.0 and Port set to 12345. The WX GUI Scope Sink block has its Sample Rate set to 32k. A callout box points to the Address field of the TCP Source block with the text: "No es necesario conocer la dirección IP del Cliente 0.0.0.0 lo hará escuchar todas las direcciones." A separate callout box contains the text: "Consejo: El diagrama de flujo no se iniciará hasta que se acepte una conexión TCP. En este caso, la GUI no aparecerá hasta que el cliente se haya conectado."

**Properties: TCP Source**

General Advanced Documentation

ID: blks2\_tcp\_source\_0

Output Type: Float

Address: 0.0.0.0

Port: 12345

Mode: Server

Length: 1

**TCP Source**  
Address: 0.0.0.0  
Port: 12345k  
Mode: Server

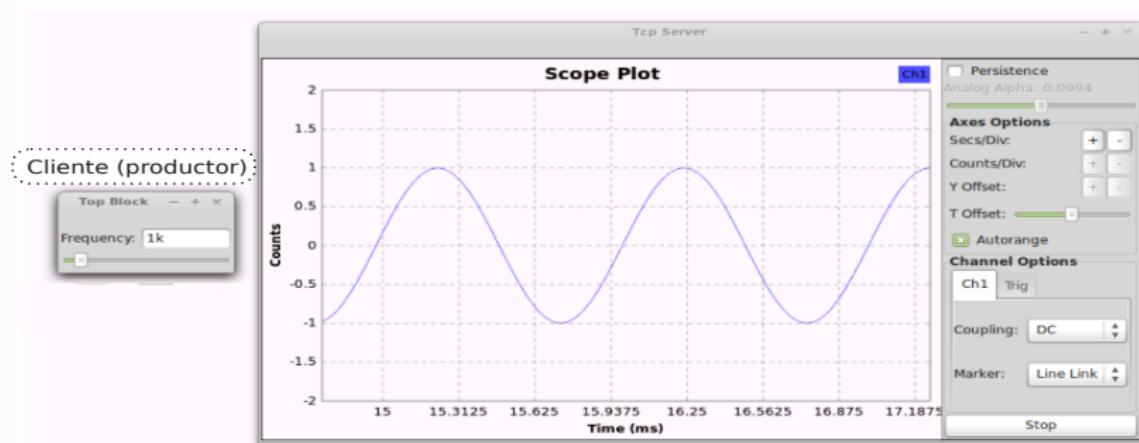
**WX GUI Scope Sink**  
Title: Scope Plot  
Sample Rate: 32k  
Trigger Mode: Auto  
Y Axis Label: Counts

**Options**  
ID: tcp\_server  
Generate Options: WX GUI

**Variable**  
ID: samp\_rate  
Value: 32k

**Consejo:**  
El diagrama de flujo no se iniciará hasta que se acepte una conexión TCP.  
En este caso, la GUI no aparecerá hasta que el cliente se haya conectado.

# Primeros pasos



Cliente (productor)

Servidor (consumidor)

Puede ejecutar cada aplicación por separado en dos máquinas conectadas en red. Simplemente cambie la dirección IP de destino del cliente a la máquina en la que se está ejecutando el servidor. Recuerde ejecutar primero el archivo del servidor y luego el del cliente.

## *Actividades*

# Transmisión de señales por multiplexación

En esta actividad usted debe transmitir 3 señales periódicas por medio del TCP (Protocolo de Control de Transmisión) desde el cliente, mediante el proceso de multiplexación de señales, al servidor, que se encargará de demultiplexar la señal recibida y mostrar las 3 que fueron transmitidas en el Scope Sink.

Es importante saber que:

1. La multiplexación es el proceso mediante el cual diferentes mensajes de información (Señales) se combinan en una única señal con el fin de trasmitirla.
2. La demultiplexación es el proceso mediante el cual la señal multiplexada recibida se divide en cada una de las señales que la generaron.

## Pistas para la actividad

Las pistas son:

1. Modo cliente: Bloques y conexiones de los primeros pasos. (WX GUI Slider, Signal Source, TCP Sink, Scope Sink, Throttle, Stream Mux, Variable)
2. Modo servidor: Bloques y conexiones de los primeros pasos. (TCP Source, Scope Sink, Stream to Streams)
3. Multiplexación: Bloque Stream Mux. En el parámetro Lengths se debe agregar el número de ítems de cada señal, en forma de lista. Para la actividad las señales cuentan con un solo ítem. Como ejemplo de lo anterior tenemos que:
  - 2 señales = 1,1
  - 3 señales = 1,1,1
  - 4 señales = 1,1,1,1
4. Demultiplexación: Bloque Stream to Streams.
5. El tipo de dato en todos los bloques debe ser el mismo. (float)
6. Habilitar las entradas o salidas suficientes para los bloques.

# Modificación de las variables de una señal periódica

Aprender a variar los componentes básicos de una señal periódica (amplitud, frecuencia, fase y nivel DC) en GNU Radio.

Es importante saber que:

1. La forma más simple de representar una señal periódica es una senoidal como se presenta matemáticamente a continuación

$$x(t) = A \sin(2\pi f t + \phi) + K$$

En donde:

2. Amplitud ( $A$ ): Es el valor máximo que toma la señal, es decir, la distancia entre el punto máximo de la señal y cero, este punto puede ser tanto positivo como negativo.
3. Frecuencia ( $f$ ): Es el número de ciclos que realiza la señal por unidad de tiempo, esta medida está dada en Hertz (Hz), que equivale a un ciclo por segundo, lo que significa que 80 Hz son 80 ciclos por cada segundo que transcurre.

# Modificación de las variables de una señal periódica

1. La fase ( $\phi$ ): Indica la magnitud de una de variación cíclica, siendo la fracción del período que transcurre desde el instante tomado al estado hasta la referencia, es decir el desplazamiento que tiene una señal en grados con respecto a su referencia.
2. Nivel DC ( $K$ ): Es el valor medio de la señal, lo que quiere decir que es un voltaje en DC que se le suma a la señal AC, para obtener un desplazamiento en la amplitud de la señal, puede ser tanto positivo como negativo.

# Actividad

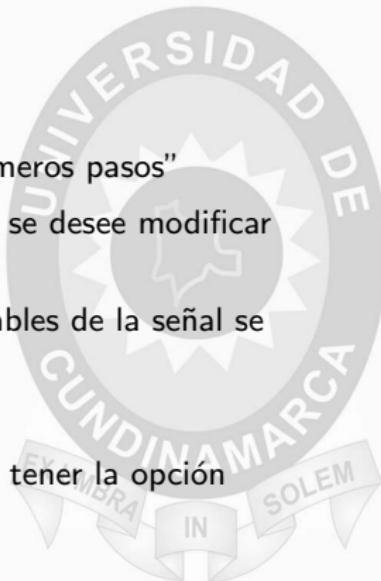
1. Realizar un programa que permita variar los componentes básicos de una señal periódica (amplitud, frecuencia, fase y nivel DC) en GNU Radio.



## Pistas para la actividad

Las pistas son:

1. Repasar los bloques utilizados en la primera guía "Primeros pasos"
2. Debe existir un WX GUI Slider para cada variable que se desee modificar en el osciloscopio.
3. Para una mejor apreciación de los cambios en las variables de la señal se sugiere añadir otra señal que sirva como referencia
4. Coherencia entre los tipos de dato entre bloques
5. Para variar el nivel DC de la señal, el osciloscopio debe tener la opción "Coupling" en DC

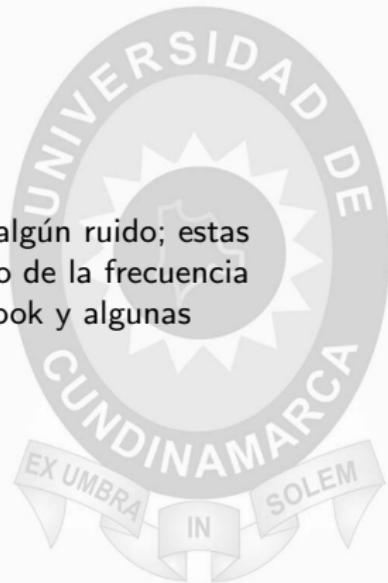


# Lab2

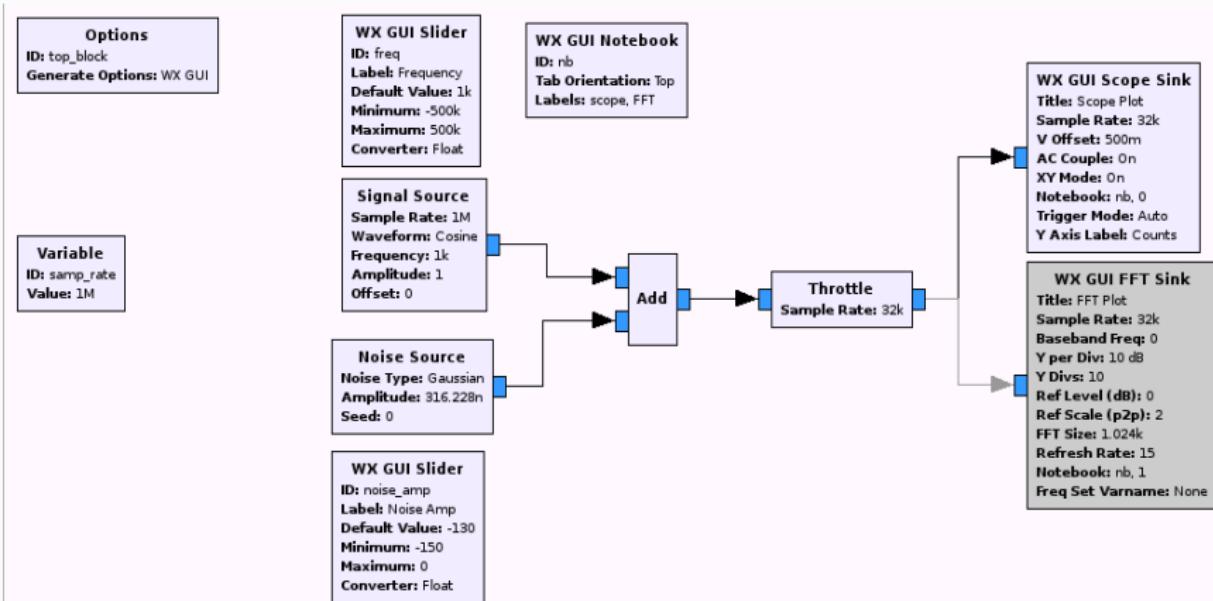
## Osciloscopio y FFT

# Osciloscopio y FFT

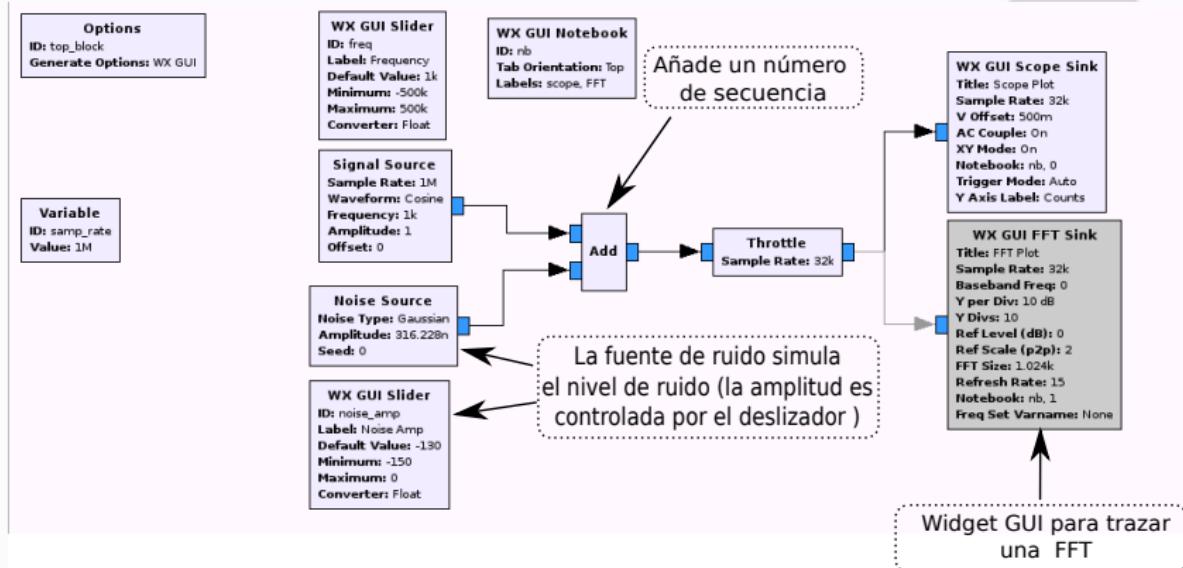
En este laboratorio se genera una onda senoidal y a su vez algún ruido; estas dos señales se suman y se observa el resultado en el dominio de la frecuencia (FFT) y del tiempo (Scope). Se aprende a utilizar el notebook y algunas herramientas básicas del GRC.



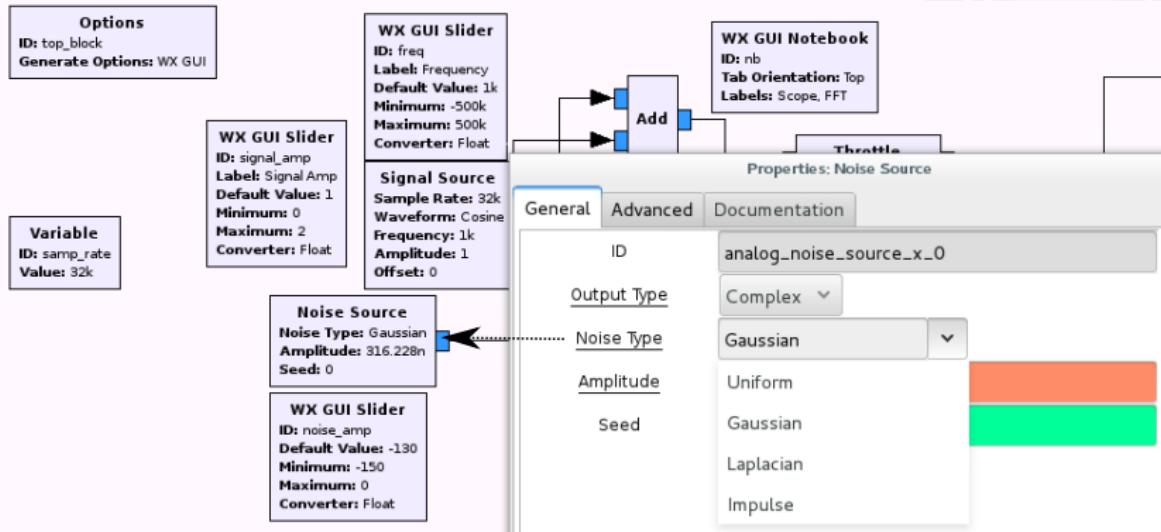
# Osciloscopio y FFT



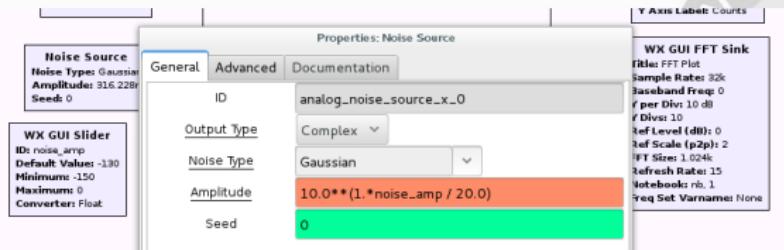
# Osciloscopio y FFT



# Osciloscopio y FFT

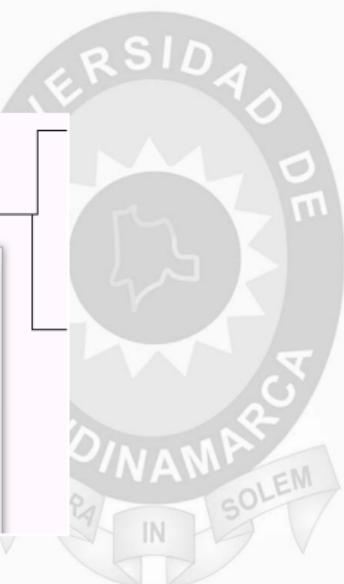
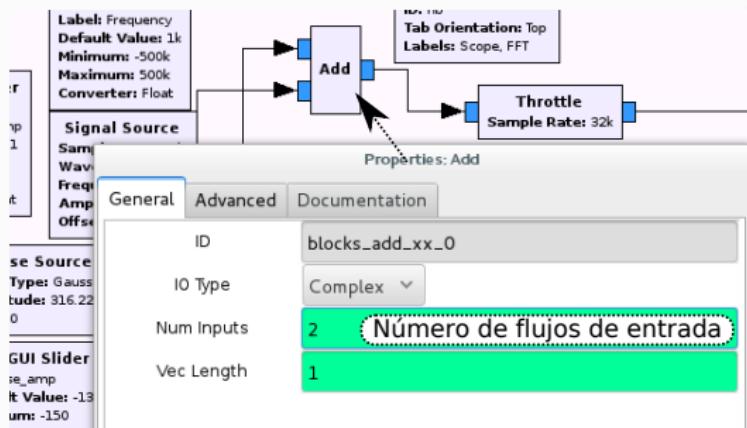


# Osciloscopio y FFT

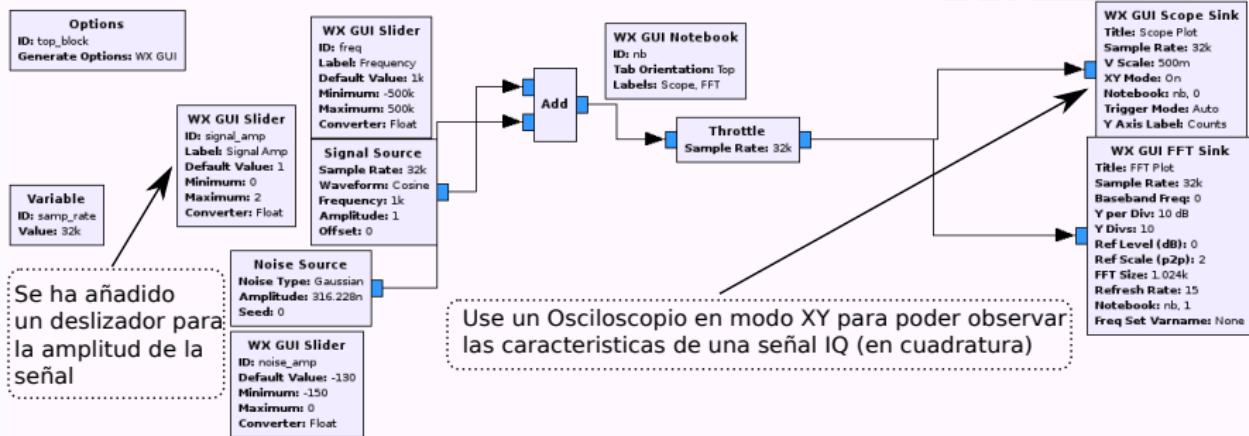


'Noise\_amp' es el valor del deslizador, que (aquí) interpretamos en dB, en oposición a un valor de amplitud de muestra lineal (por ejemplo '1,0'). Por lo tanto, necesitamos convertir el valor en dB a un valor real de amplitud lineal ('voltios') para su uso en el bloque (es decir, invertir la función 'log10'). Los puntos decimales se añaden para obligar a Python a calcular con valores de punto flotante (de lo contrario redondea y produce números enteros).

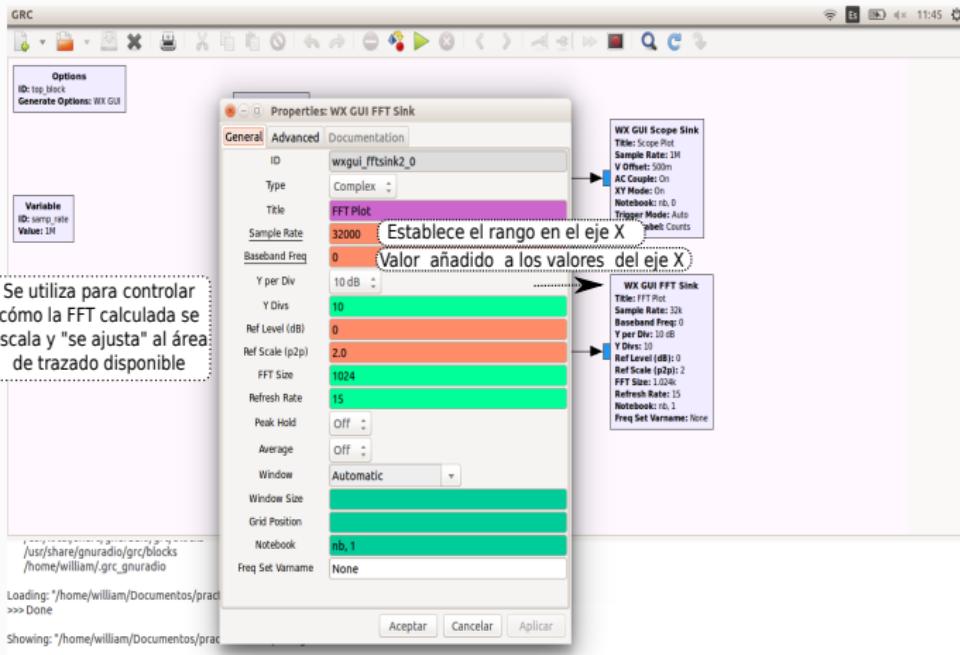
# Osciloscopio y FFT



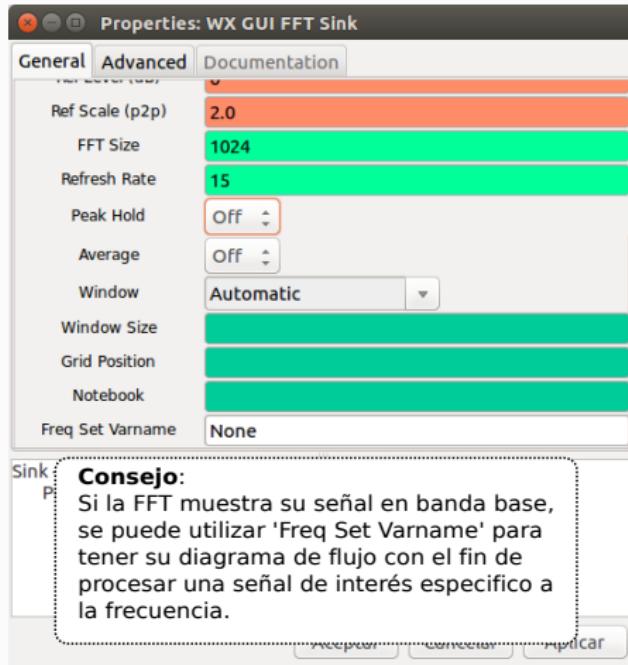
# Osciloscopio y FFT



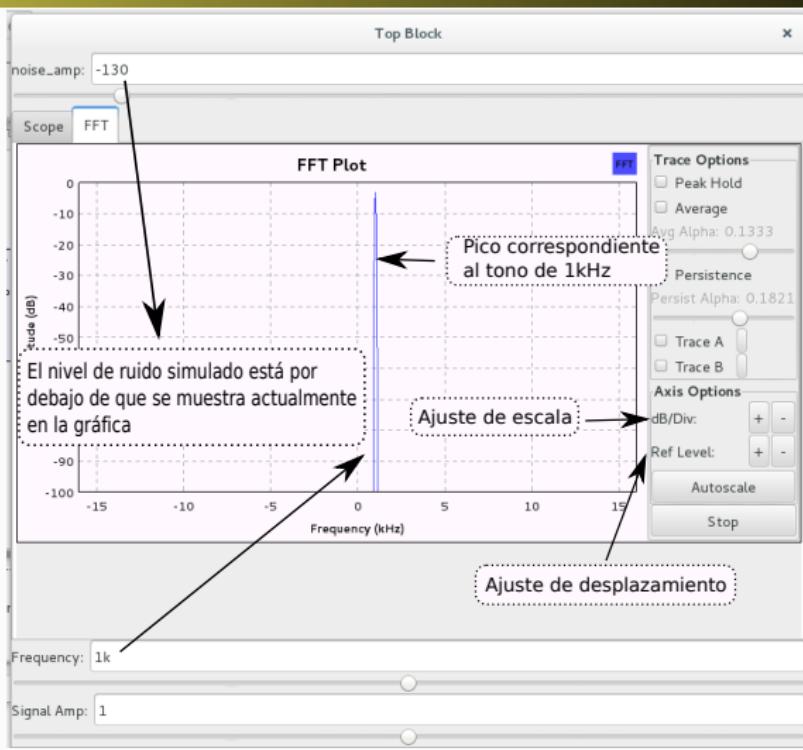
# Osciloscopio y FFT



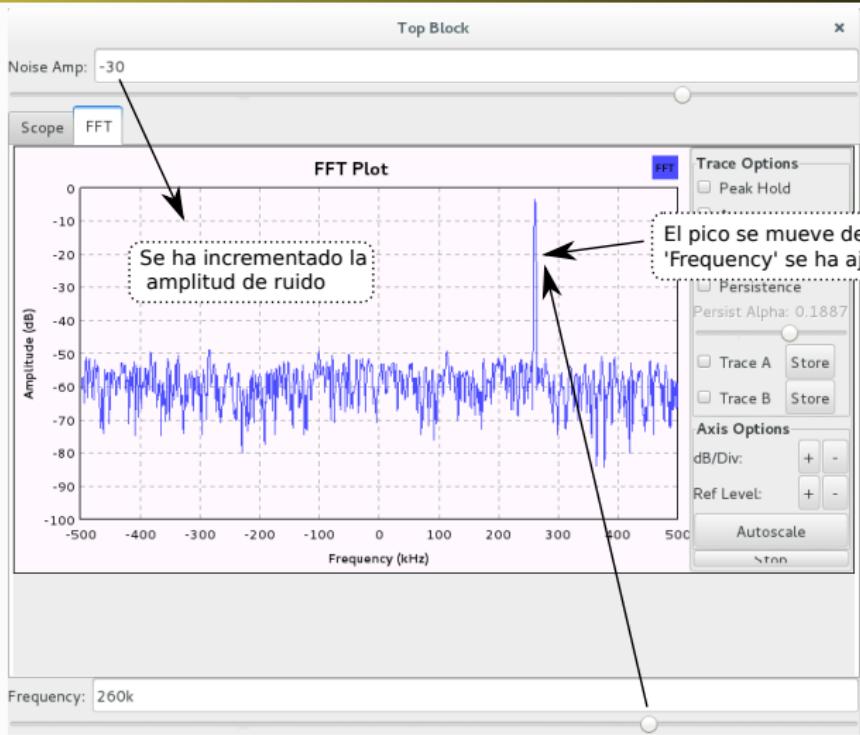
# Osciloscopio y FFT



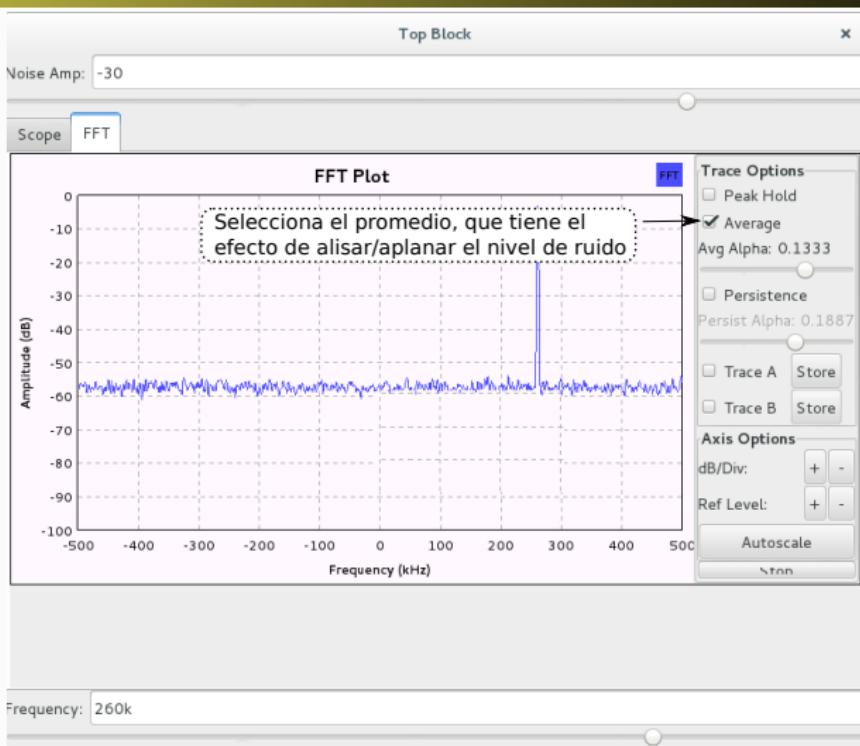
# Osciloscopio y FFT



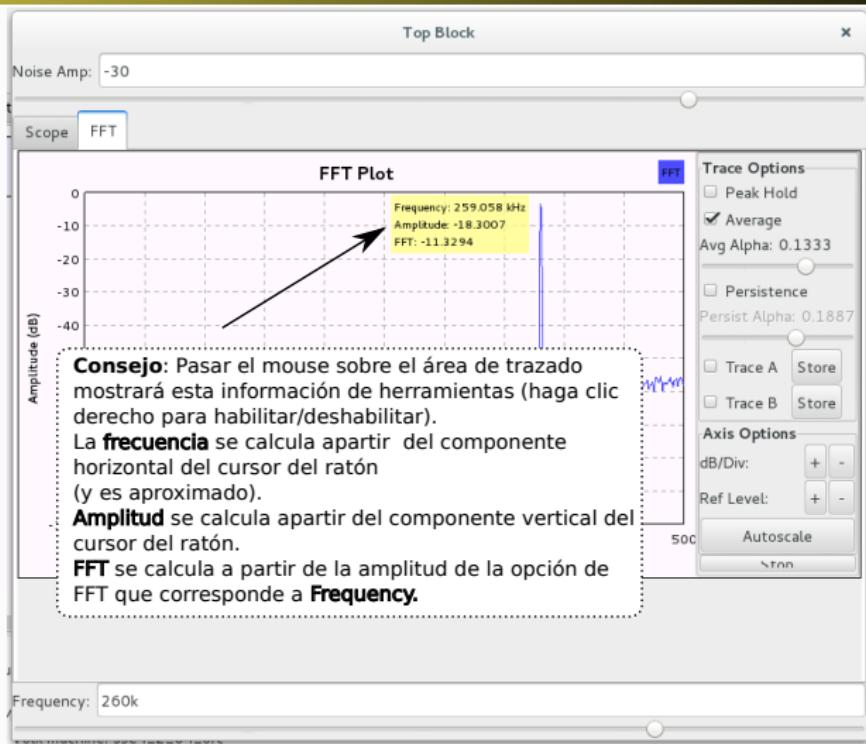
# Osciloscopio y FFT



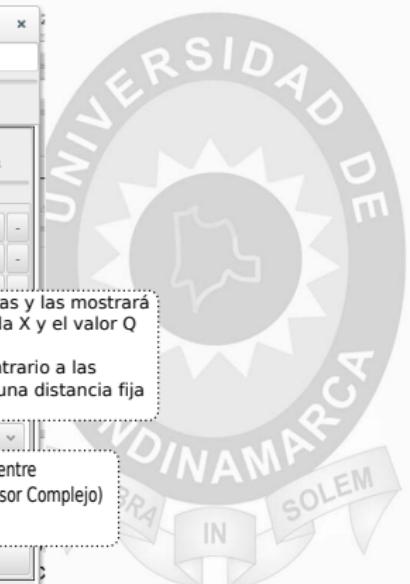
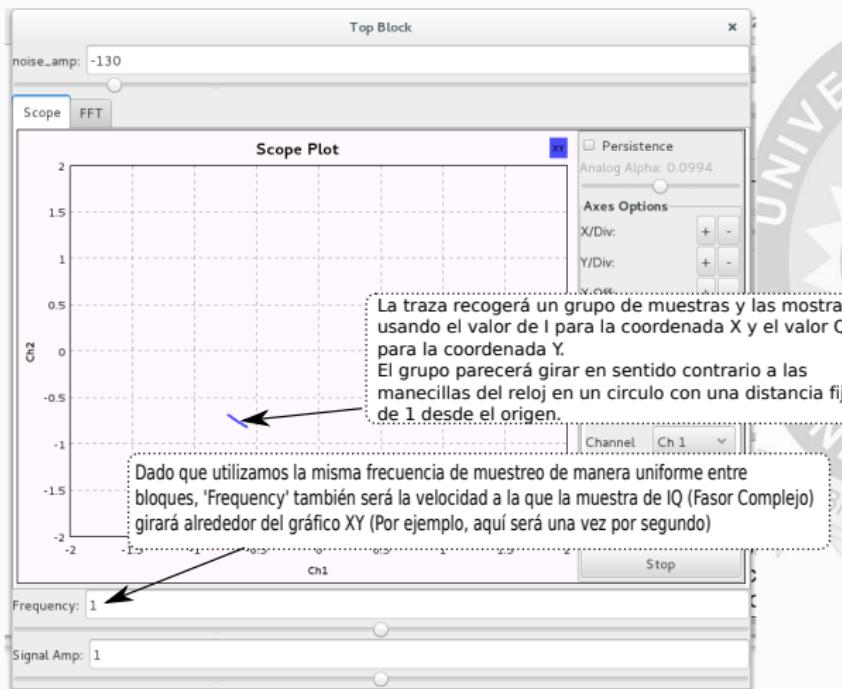
# Osciloscopio y FFT



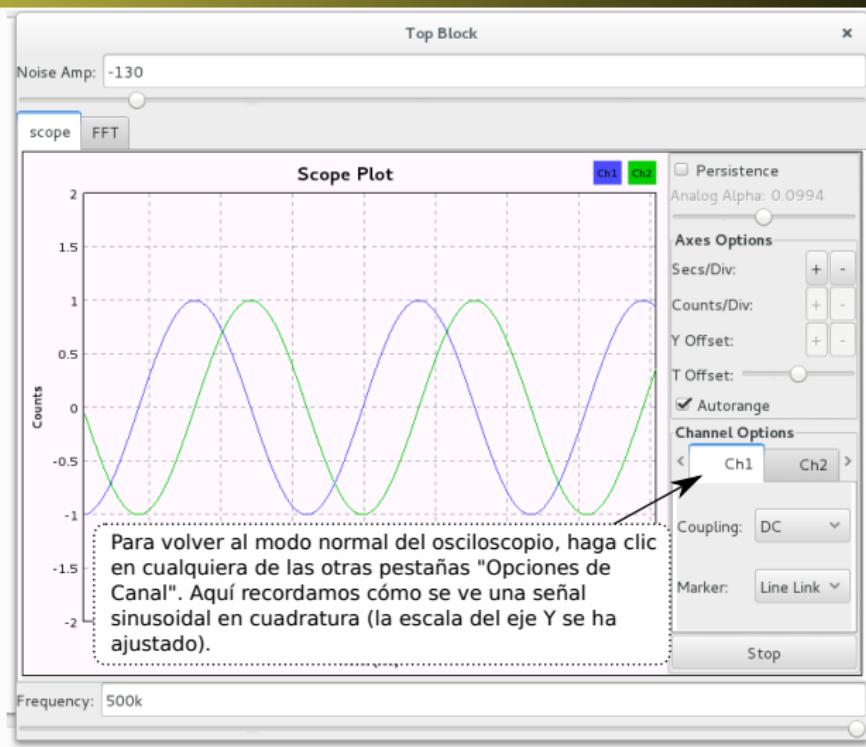
# Osciloscopio y FFT



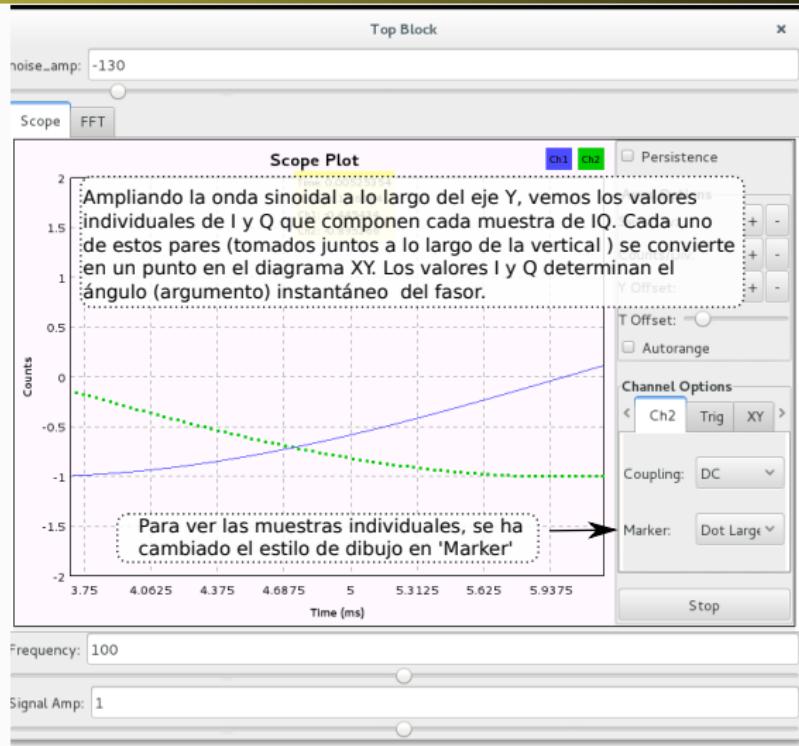
# Osciloscopio y FFT



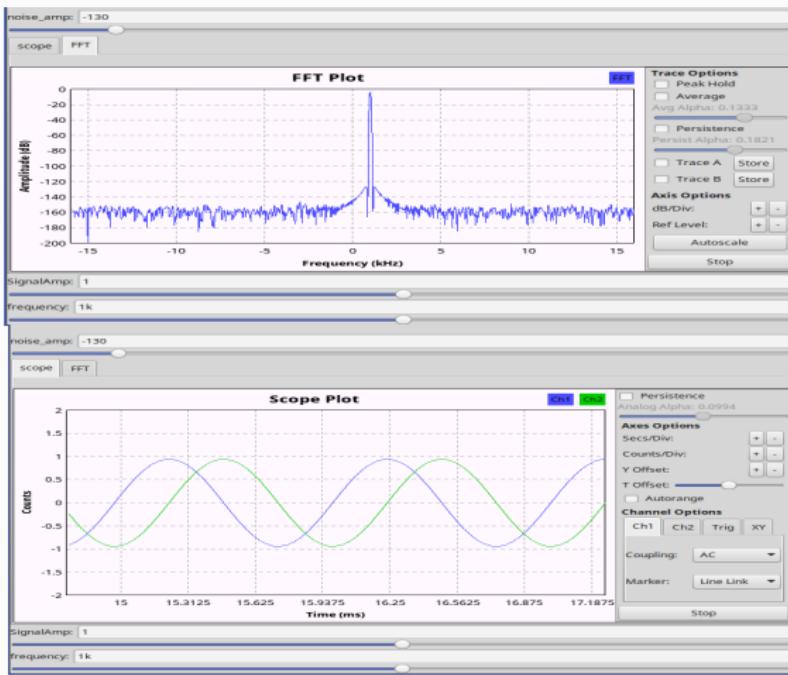
# Osciloscopio y FFT



# Osciloscopio y FFT

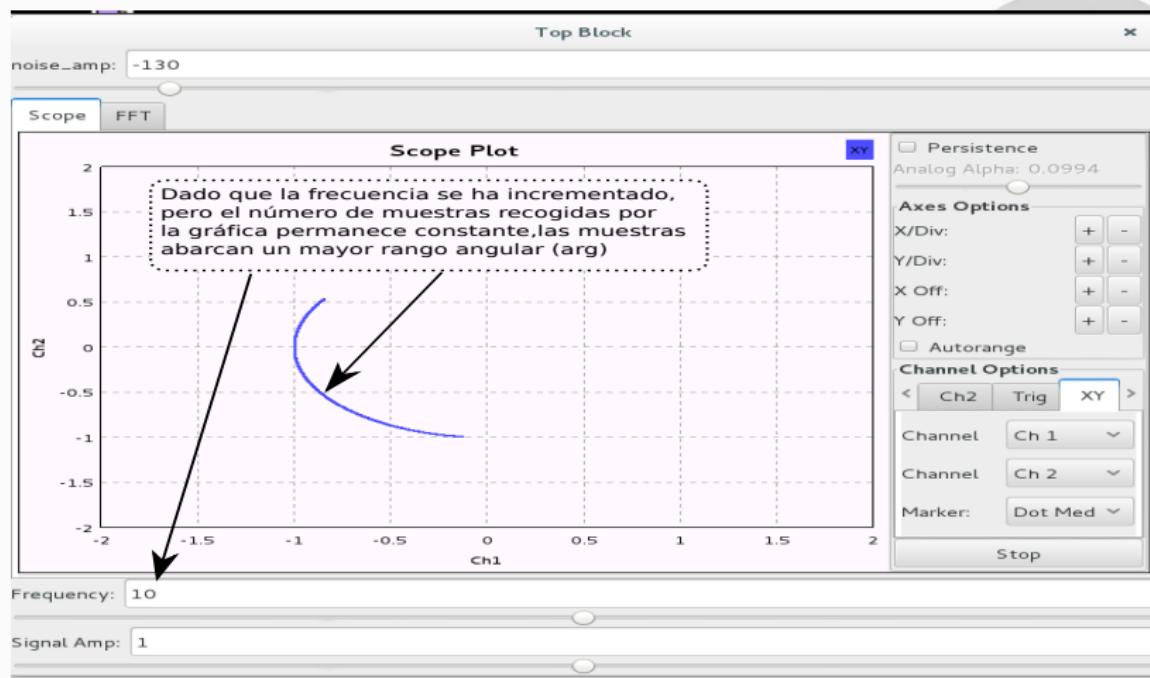


# Osciloscopio y FFT

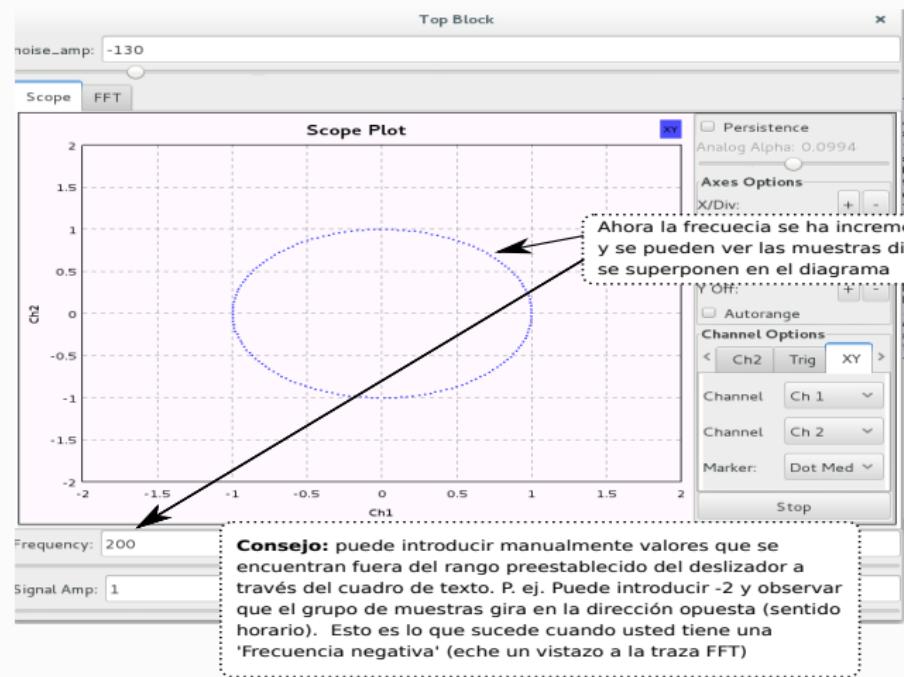


Se puede observar que tanto la gráfica de FFT, como la de Scope tienen los mismos parámetros (ruido, frecuencia, amplitud), sin embargo el ruido es más perceptible solo en los términos de los dBs, para que se pueda observar el ruido en el dominio del tiempo, éste debe ser comparable con la amplitud de la señal.

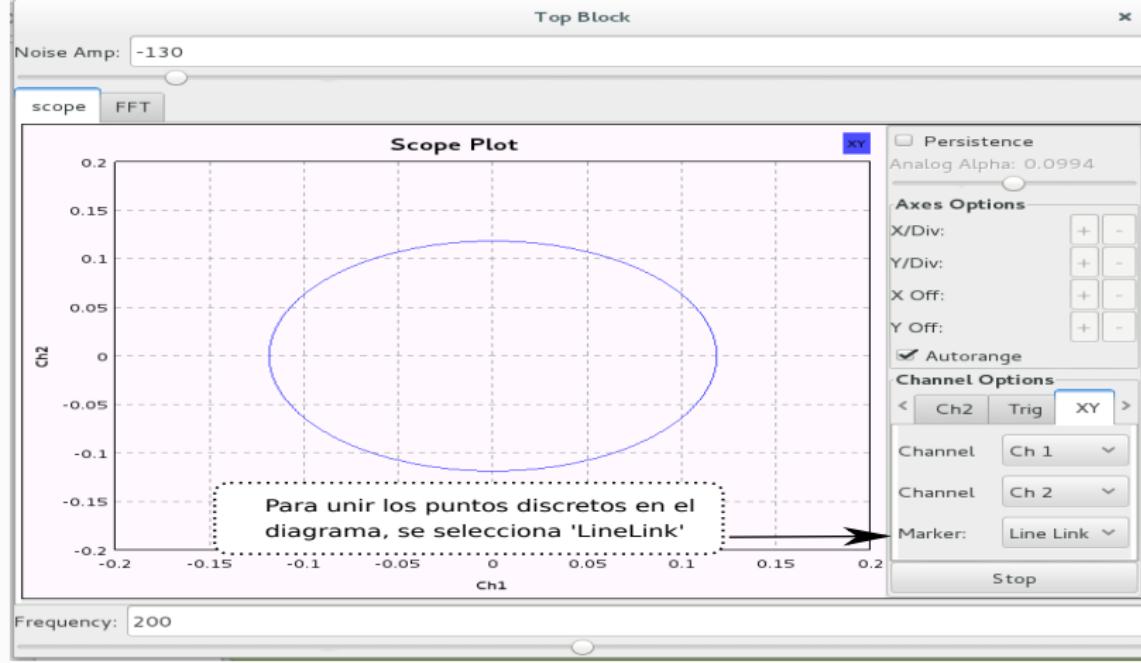
# Osciloscopio y FFT



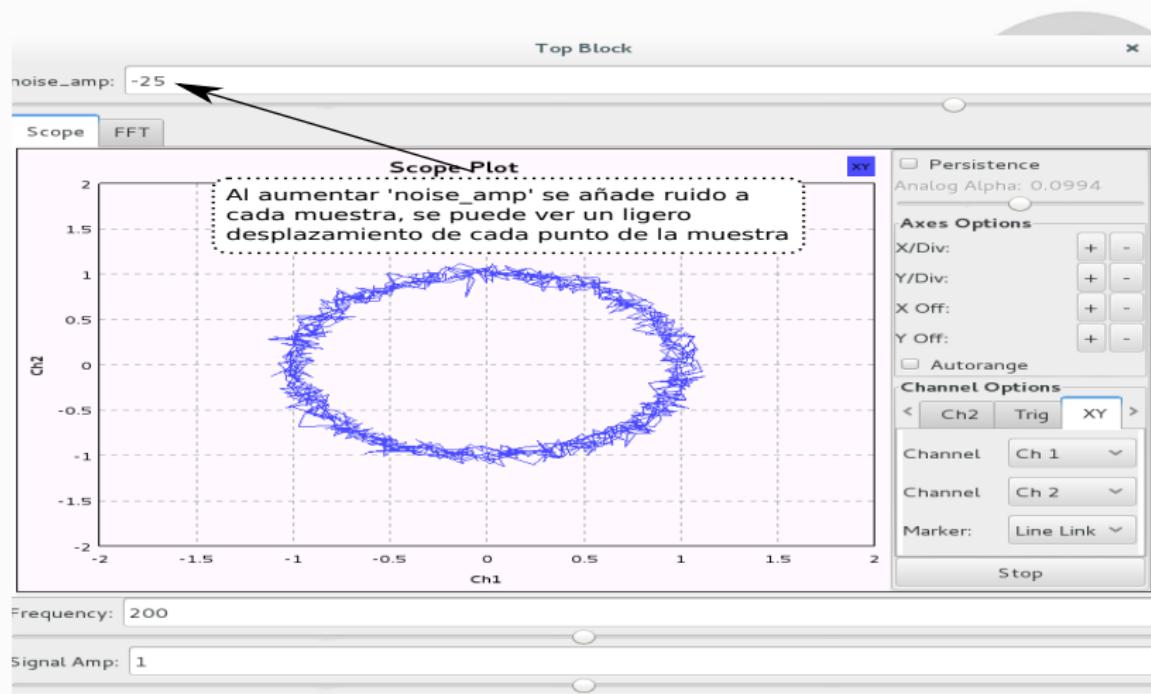
# Osciloscopio y FFT



# Osciloscopio y FFT



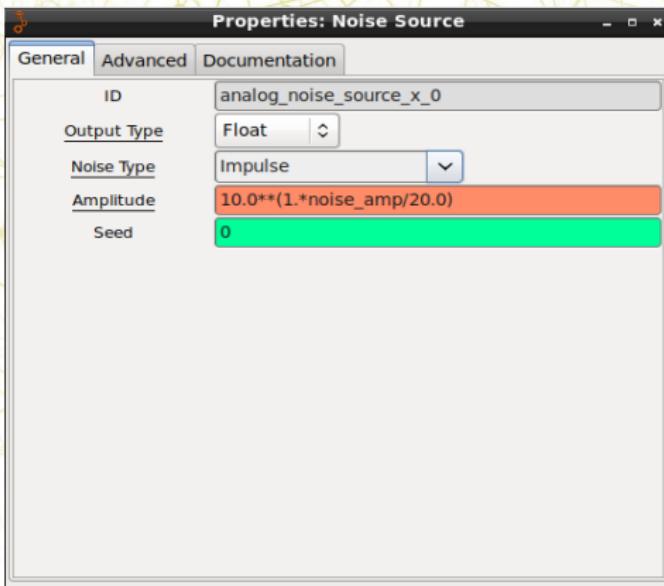
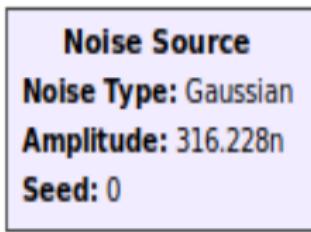
# Osciloscopio y FFT



## *Actividades*

## Actividad 1 lab 2

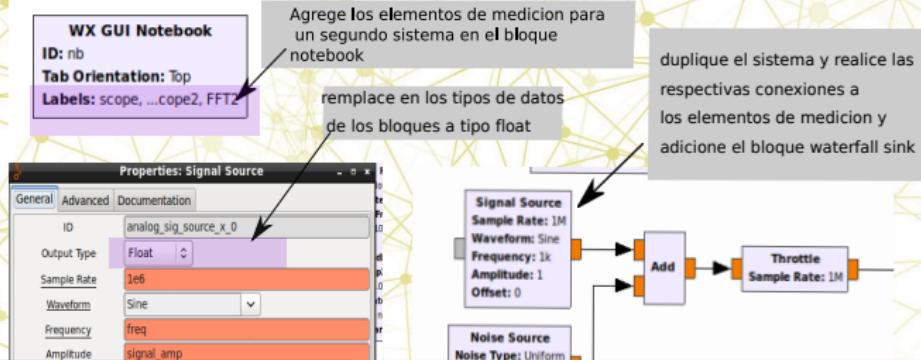
Mencione los diferentes tipos de ruido del bloque (noise source), sus características, describa la función de densidad de probabilidad para cada uno de estos y por que se puede ocasionar



## Actividad 2 lab 2

### Objetivos:

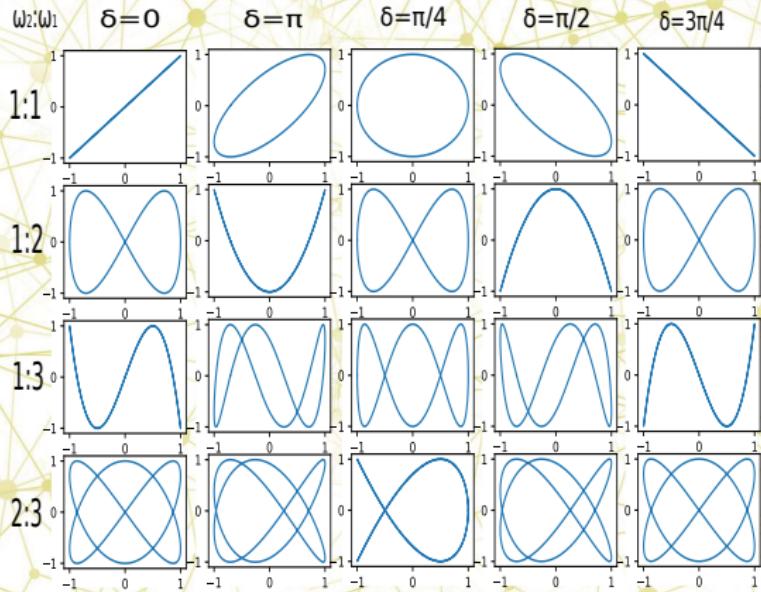
- Dar a conocer a lector la distribución de los diferentes tipos de ruido y sus propiedades.
- Relacionar las funciones de densidad con cada uno de los diferentes ruidos que aparecen en el modulo noise source.
- Mediante una breve configuración al lab 2 compare y evidencie que lo anterior se cumpla.



## Actividad 3 lab 2

Objetivo:

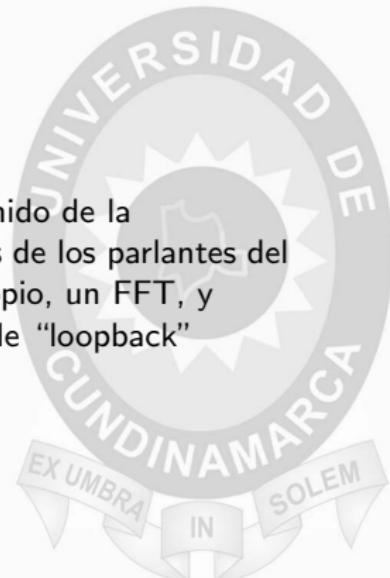
-Variar la amplitud y fase de la señal para generar las curvas de Lissajous.



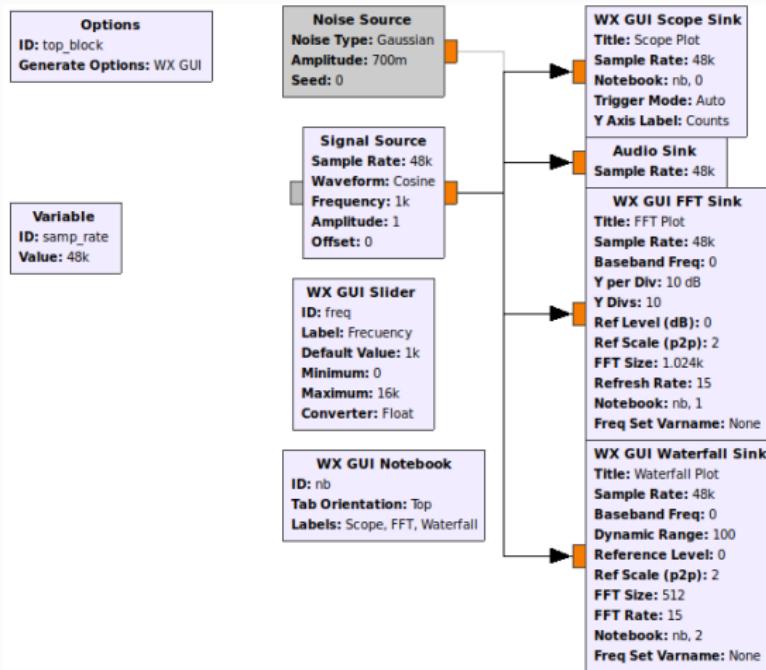
# Lab3 Audio

# Audio

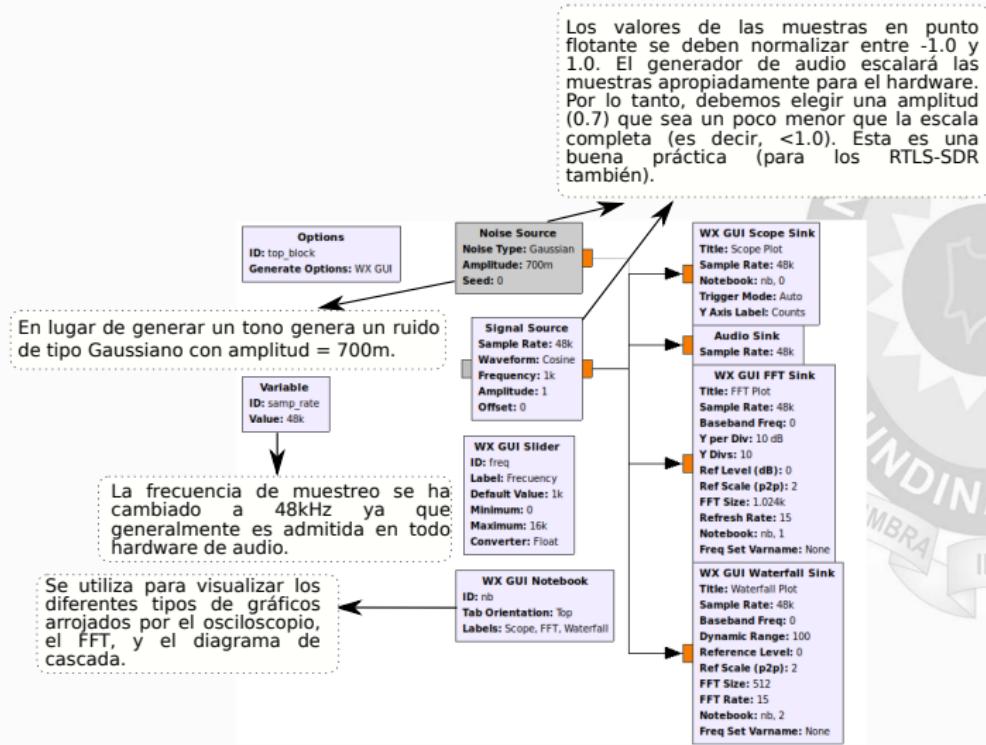
En esta práctica se generará un tono desde la tarjeta de sonido de la computadora, originado desde el software y emitido a través de los parlantes del computador, dicha señal será visualizada desde un osciloscopio, un FFT, y diagrama de cascada (espectrograma), realizando pruebas de “loopback” usando el micrófono de la computadora.



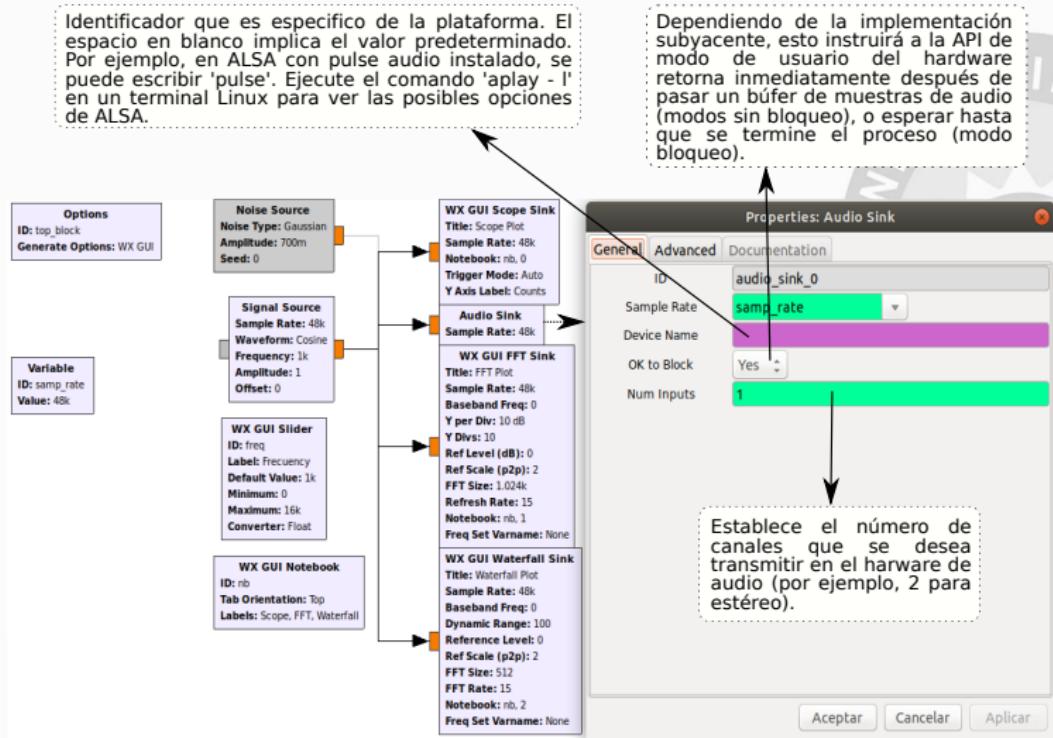
# Diagrama: “emisión de audio desde la computadora”



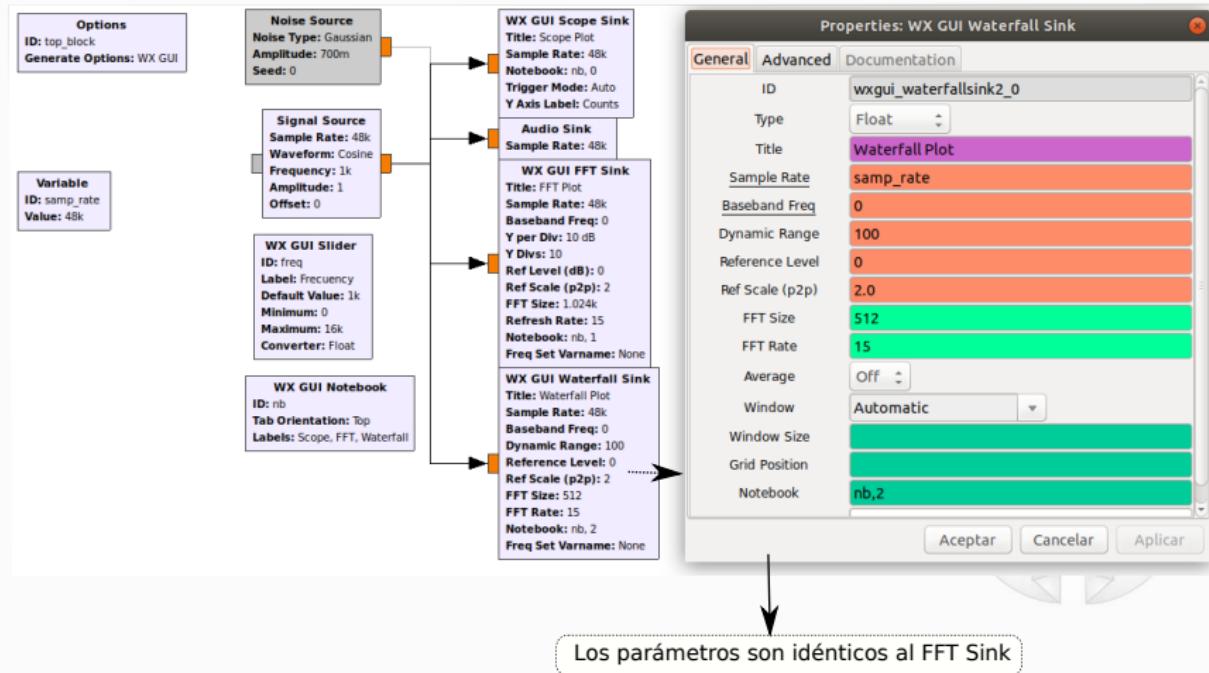
# Diagrama: “emisión de audio desde la computadora”



# Diagrama: “emisión de audio desde la computadora”



# Diagrama: “emisión de audio desde la computadora”



# Audio

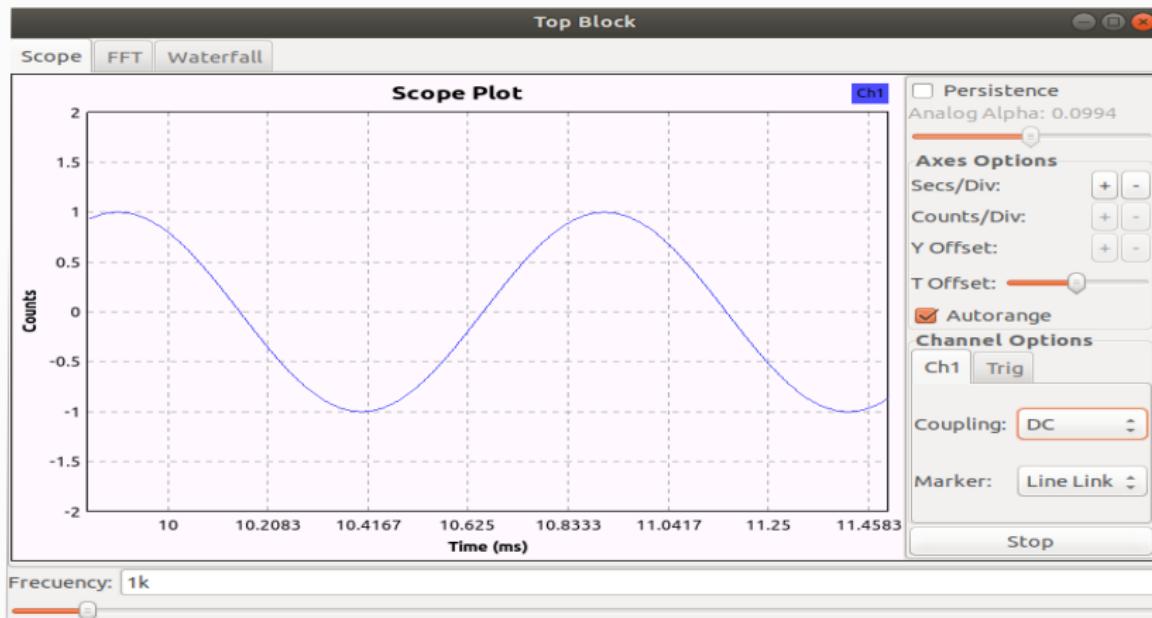
Es importante saber que:

- Cuando Audio Sink es el único dispositivo hardware en el diagrama de bloques capaz de generar audio, el modo de bloqueo ('OK to Block') aplicará un regulador a la producción de muestras del sample\_rate, para que opere eficazmente al reproducir el sonido[1].
- Esto puede ser problemático si la fuente del diagrama de flujo es, por ejemplo, un RTL-SDR. La fuente es también un hardware que tiene su propio reloj interno y será regulado a la tasa de producción de las muestras, mientras que el Audio Sink regula el uso con su propio reloj no sincronizado. Esto se llama el problema de "dos relojes".

# Audio

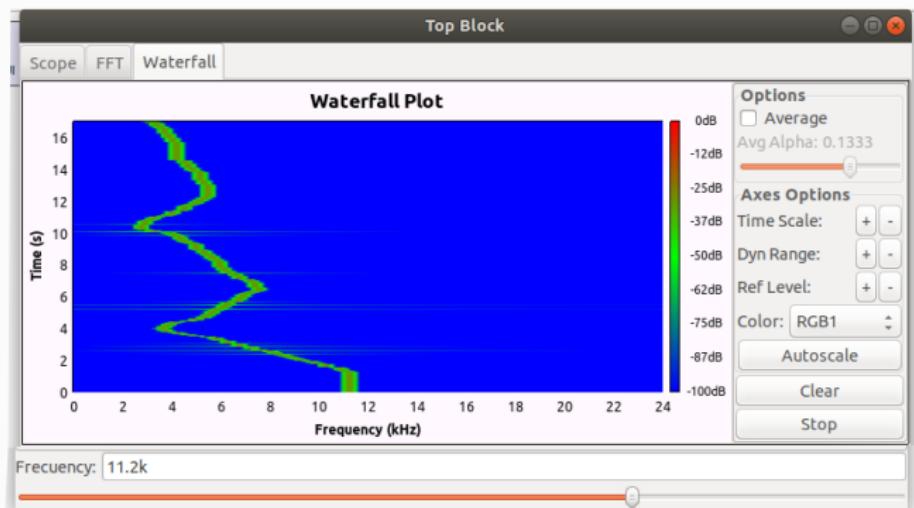
- Para solucionar este problema de dos relojes, se coloca un regulador de audio en modo sin bloqueo (no dar click 'Botón de Bloqueo') de tal forma que nunca interrumpa el diagrama de bloque (es decir, no aplicar el regulador controlado). Esto usará muestras de forma normal, pero si hay un exceso (por ejemplo, el RTL-SDR está produciendo muestras un poco más rápido de lo que el Audio Sink puede usar), se perderán las muestras (podría causar fallas de audio).
- Esto no soluciona el caso en el que las muestras se producen más lentamente que la tasa de uso del Audio Sink (esto producirá una ejecución lenta: el audio sonará agitado y se imprimirá 'aU' en la ventana de registro).

# Audio



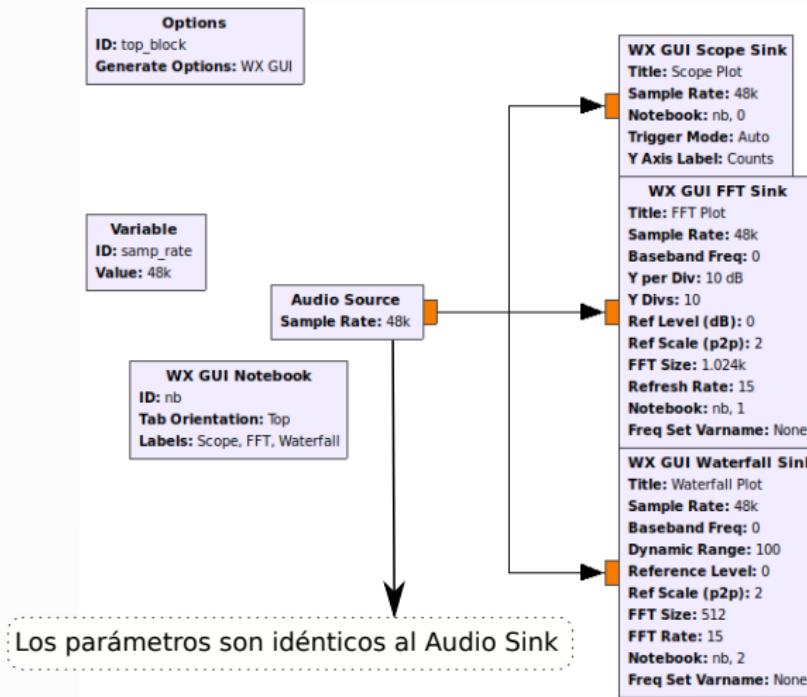
La misma onda seno de las prácticas anteriores, pero ahora se puede escuchar emitida por los parlantes del computador.

# Audio

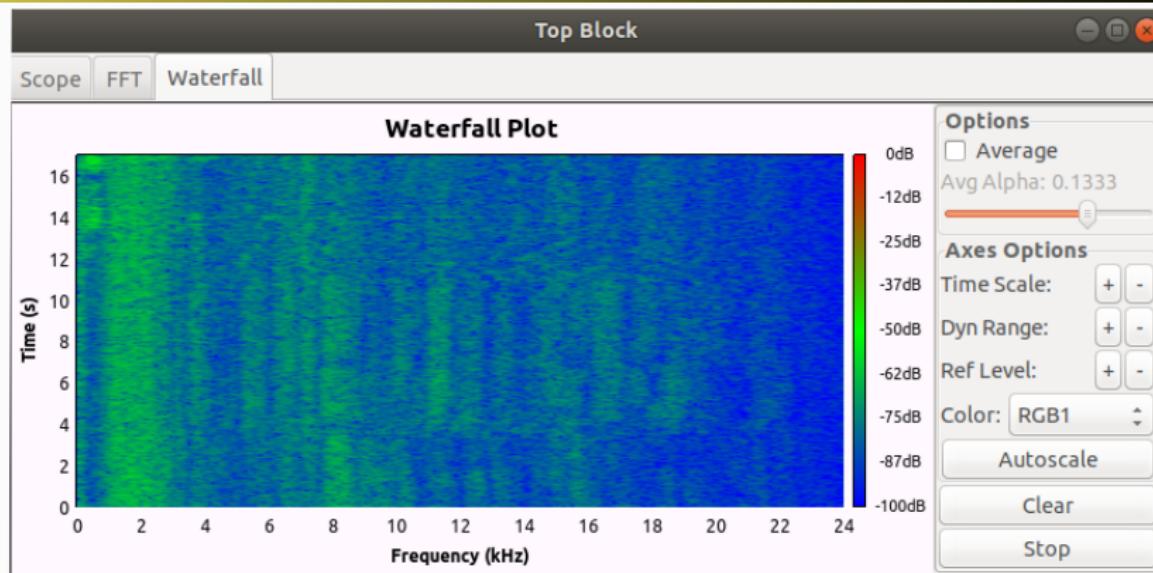


Visualiza el FFT que se desplaza en el tiempo mediante el diagrama de cascada (espectrograma) de la señal emitida. Se añade un bloque de prueba por medio de un generador de señales y un variador deslizante con lo cual se escucha el tono variado en el Audio Sink y poder ver la variación de la frecuencia en el diagrama de cascada.

## Diagrama: recepción de audio

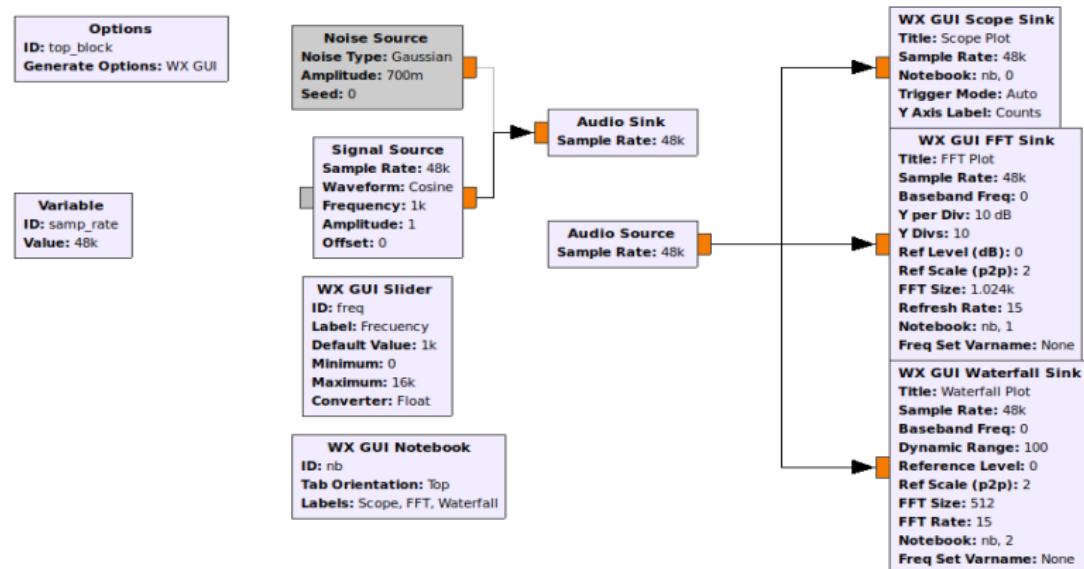


# Audio



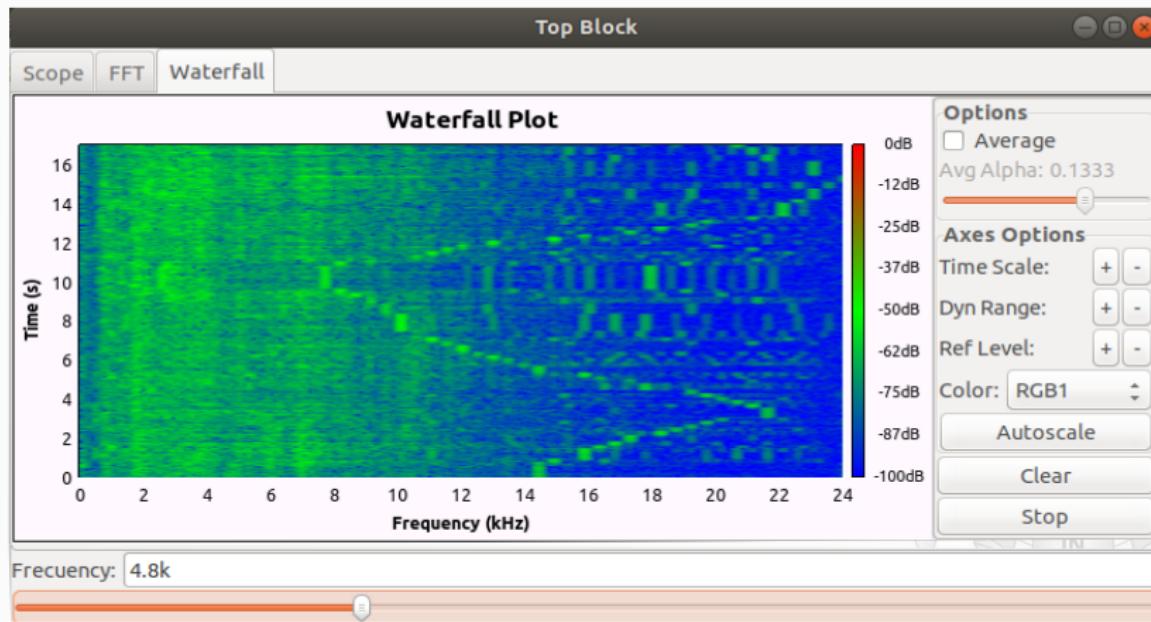
Muestra las diferentes señales presentes en el entorno captadas por la tarjeta de audio de la computadora a través del micrófono.

# Diagrama: prueba con aproximación de “loopback”



Ejecutando el programa generador de onda sinusoidal al mismo tiempo, y cambiando la frecuencia. Se trata de una prueba aproximada de “loopback” en la que el micrófono de la computadora escucha sus altavoces.

# Audio



Con la realimentación de las entradas micrófono-altavoces y generación de señal a través de la tarjeta de audio de la computadora.

# ERRORES DE SINCRONIZACION

Los datos de frecuencias de muestreo a un receptor de audio funcionan a 48000S/s muestras por segundo, aunque no todos los receptores pueden ser manejados en esta frecuencia. Dependiendo de la unidad de audio utilizada y el hardware, algún día se produce un overflow o underflow del búfer, imprimiendo uno de los siguientes códigos de error:

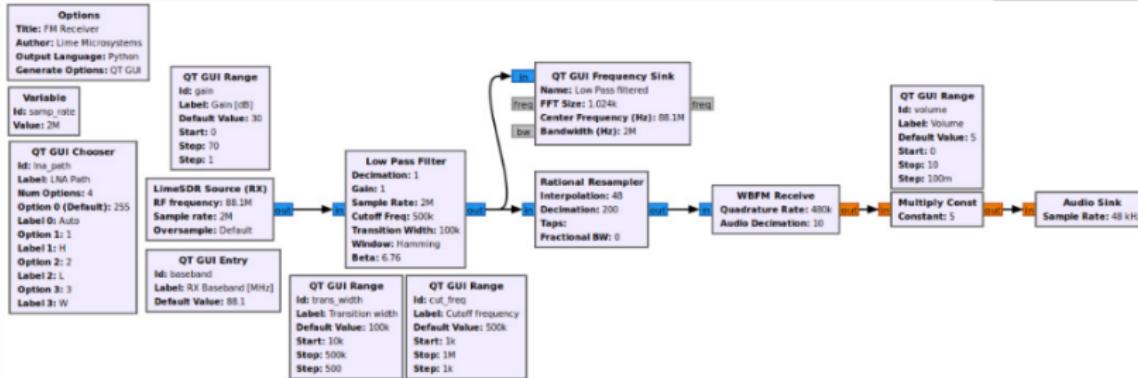
- "**a**" audio (tarjeta de sonido).
- "**O**" overrun "desbordamiento" (la PC no se mantiene al día con los datos recibidos de la tarjeta de audio usrp0).
- "**U**" underrun "subestimada" (la PC no proporciona datos lo suficientemente rápido).

# ERRORES DE SINCRONIZACION

- "**aUaU**" audio underrun (no hay suficientes muestras listas para enviar al sonido USRPsink).
- "**uOuO**" USRP desbordado (las muestras de USRP se descartaron porque no se leyeron a tiempo).
- "**S**" indica un error de número de secuencia en los paquetes de Ethernet que marca una saturación de USRP a la PC como "O".

La arquitectura de audio más común utilizada en Linux es "Arquitectura de sonido avanzada de Linux" (ALSA). Incluye un dispositivo de muestreador de software llamado plughw, que proporciona una amplia adaptación de frecuencia en un rango dado.

# ERRORES DE SINCRONIZACION



```

aUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUa
aUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUaUa
[ibus_Exit] application left some devices open

```

>>> Done

Id	Value
Imports	
Variables	
baseband	<Open Properties>
cut_freq	<Open Properties>
gain	<Open Properties>
ina_path	<Open Properties>
samp_rate	2e6
trans_width	<Open Properties>

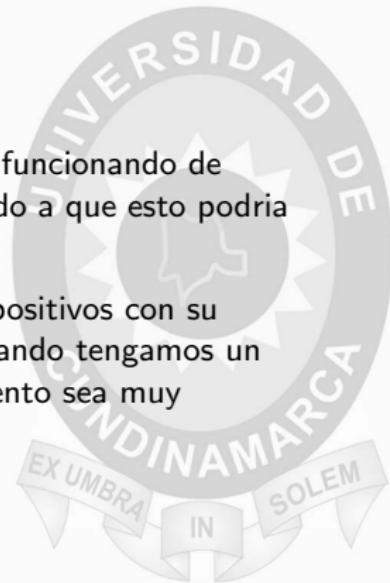
# ERRORES DE SINCRONIZACION

cuando nos aparece este error “aUaU” lo más probable es que sea un “problema de 2 relojes”. El problema es que tiene 2 piezas de hardware con un reloj, y no hay forma de alinearlas o sincronizarlas perfectamente. Aquí hay muchos modos de falla, pero uno muy obvio es el siguiente: imagine que el radio reloj (emisor) es un poco más lento de lo anunciado, y el reloj de audio (receptor) es más rápido. Por lo tanto en algún momento la cantidad de datos subministrados serán muy bajos tanto así que se perderán dichos datos y nos aparecerá dicho error. Además, recuerde que GNU Radio mezcla datos en fragmentos, por lo que puede haber suficientes datos, pero no en el momento preciso en que el receptor de audio los necesita.

# ERRORES DE SINCRONIZACION

una posible solucion es verificar que todos los bloque esten funcionando de forma predeterminada y que no hallan sufrido cambios debido a que esto podria generar una variacion de velocidad en los relojes.

Este error solo ocurrira cuando estemos trabajando dos dispositivos con su respectivo reloj (tarjeta de audio, RTL, etc) o siempre y cuando tengamos un emisor y un receptor y la velocidad de su reloj o procesamiento sea muy diferente.

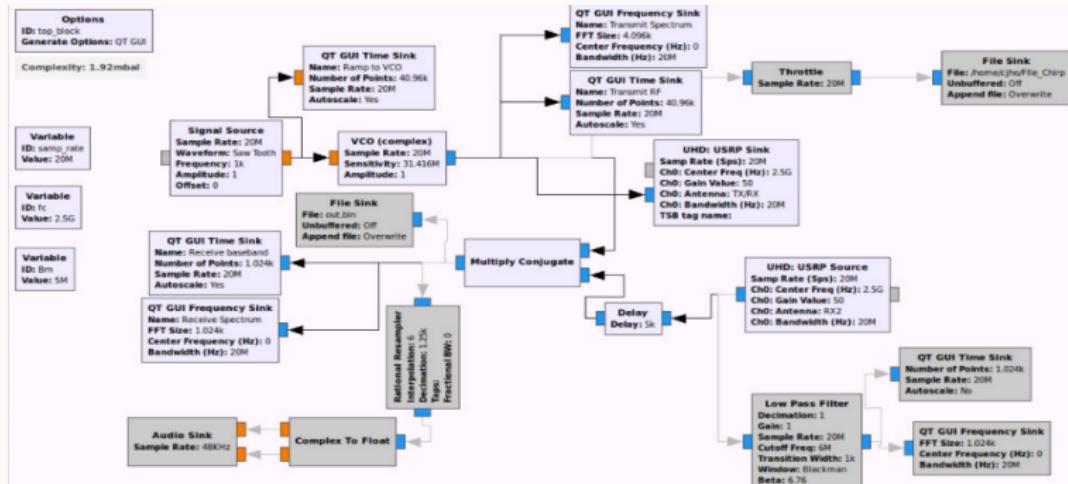


# ERRORES DE SINCRONIZACION

El muestreo a una frecuencia de muestreo alta puede causar errores de desbordamiento, lo que significa que la computadora host no puede procesar las muestras entrantes a la frecuencia que se entregan, lo que nuevamente significa que el USRP tiene que descartar las muestras. Por lo general, esto lo detecta el software de manejo y se le notifica al usuario imprimiendo "O" sutil en la salida estándar.

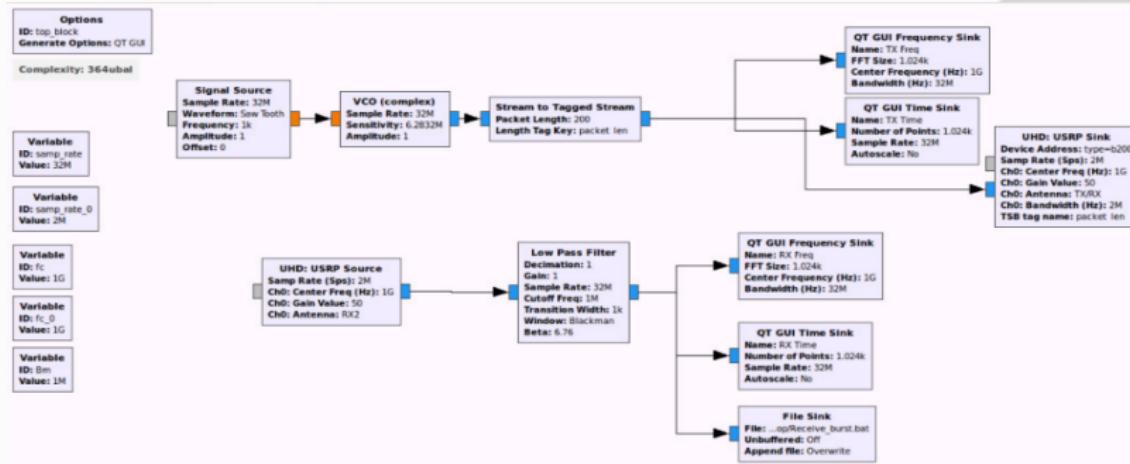
Sin embargo, el autor no los buscó, especialmente porque no se esperaba un desbordamiento para las frecuencias de muestreo a 1 MHz. Debido a la incertidumbre, el autor tuvo que retroceder y medir ambas balizas a diferentes frecuencias de muestreo para evitar por completo la posibilidad de que LA2SHF tuviera problemas de tiempo en su código Morse, y que esto era solo desbordamientos y errores de tiempo durante nuestra recepción cadena.

## ERRORES DE SINCRONIZACION



uOuO = USRP desbordado (las muestras de USRP se descartaron porque no se leyeron a tiempo).

# ERRORES DE SINCRONIZACION



```

Setting master clock rate selection to 'automatic'.
Asking for clock rate 32.000000 MHz...
Actually got clock rate 32.000000 MHz.
Performing timer loopback test...pass
Performing timer loopback test...pass
Using Volk machine: avx2_64_mmxx_orc
300000000000
>>> Done

```

# ERRORES DE SINCRONIZACION

El hecho de que el problema de sincronización se debe a desbordamientos también es muy evidente por el hecho de que el diagrama de flujo de GNU Radio imprimió "OOO" durante la recepción de esta serie de mediciones, lo que indica al menos tres desbordamientos, lo que notamos si hubiéramos seguido la salida estándar de GNU Radio acompañante más diligentemente durante las mediciones del patrón.

Esto no es muy interesante, y es un problema conocido que debe tenerse en cuenta asegurando que la computadora esté al día, que los bloques puedan manejar la transmisión, que la conexión al USRP sea lo suficientemente rápida, etc.

# Lab4

## Modulación ASK en GRC

# Modulación ASK en GRC

- En esta práctica se hace uso de la Modulación por desplazamiento de amplitud (ASK) este es un esquema de modulación digital en el que la amplitud de la onda portadora se cambia con respecto a la señal de información, manteniendo la fase y la frecuencia de constantes. Para el presente laboratorio, se utilizó BASK, el cual tiene el mismo comportamiento, pero utilizando bits. Donde su comportamiento es descrito por la siguiente ecuación:

$$S(t) = Am(t)\cos(wt)$$

- El diagrama de bloques en GRC consiste en la multiplicación de una señal de información con una portadora, que corresponde a una señal coseno de diez veces la frecuencia de la información. La frecuencia de muestreo debe ser mayor que el doble de la frecuencia máxima de la señal de datos.

# Modulación ASK en GRC

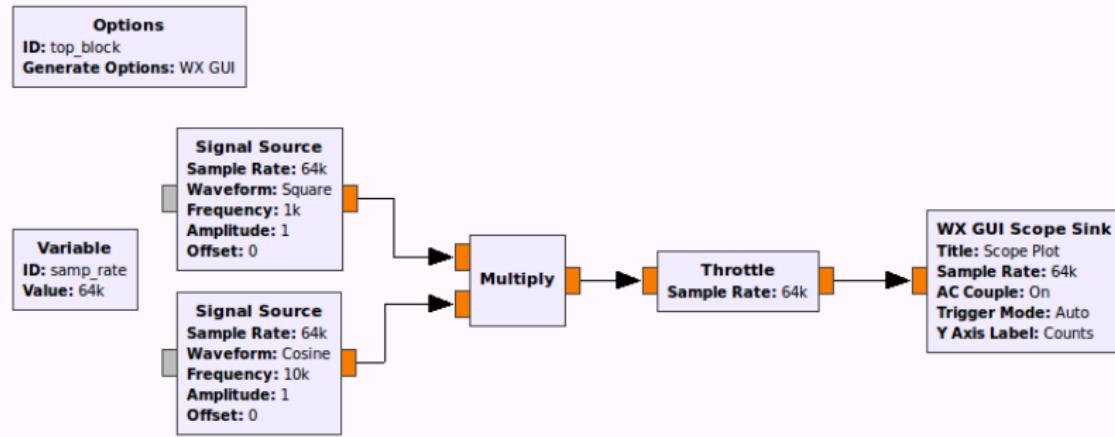
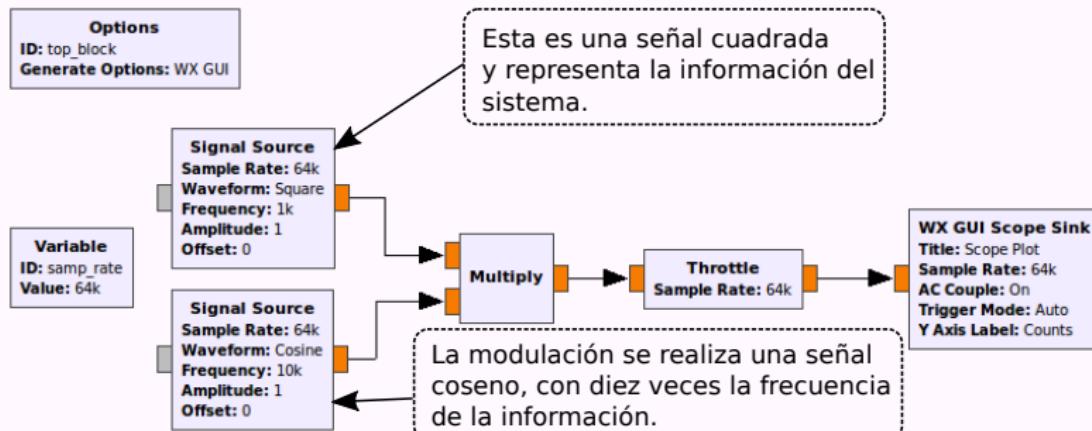
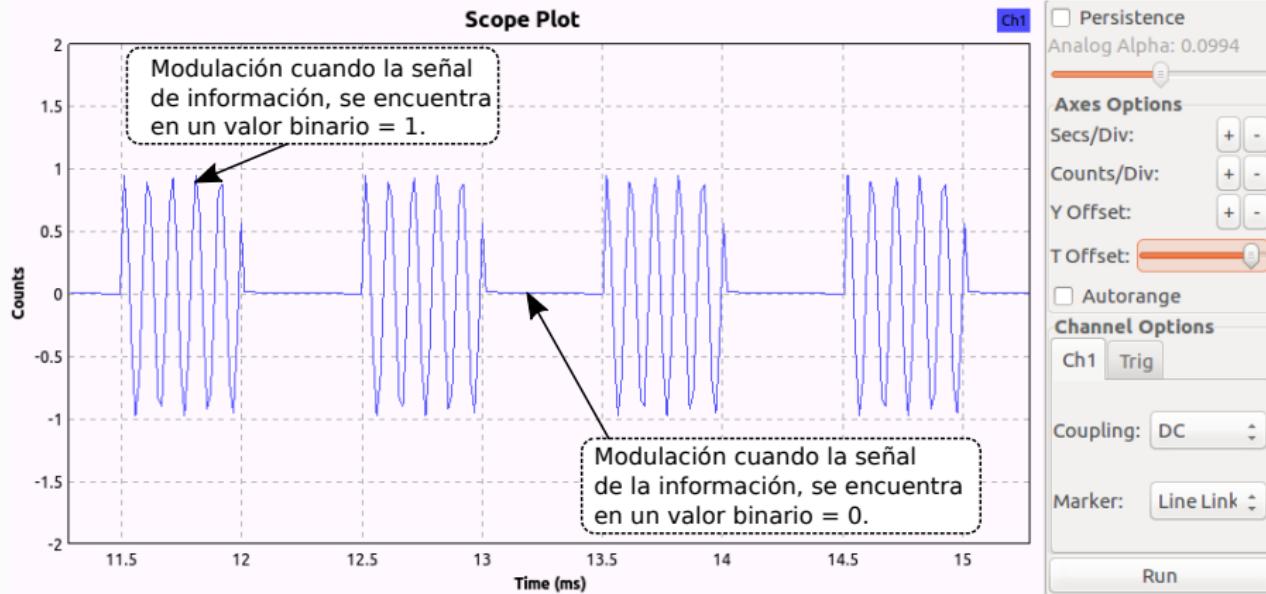


Diagrama de bloques en GNU Radio, para generar una modulación ASK entre dos señales.

# Modulación ASK en GRC



# Modulación ASK en GRC



# Modulación ASK en GRC

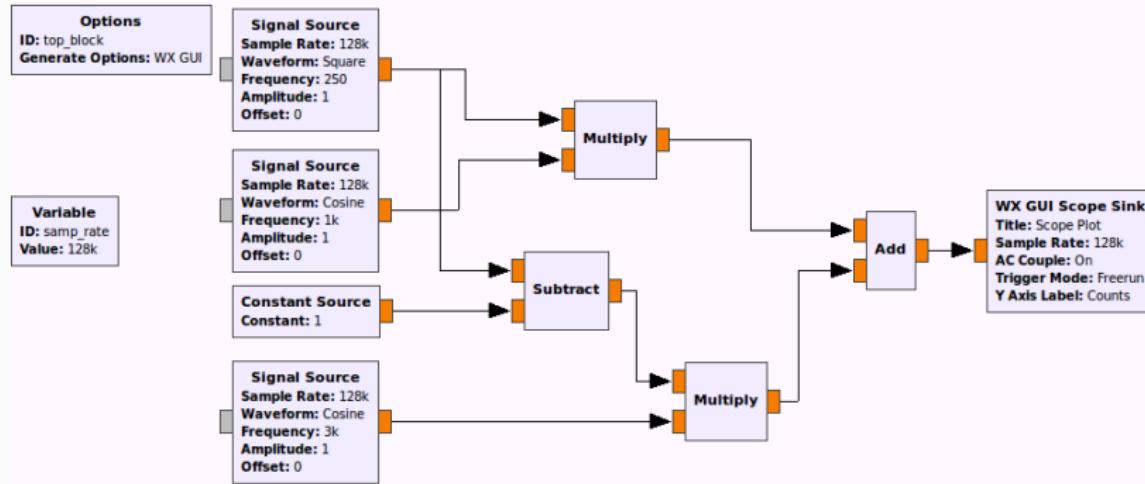
- La Modulación por desplazamiento de frecuencia (FSK) es una técnica de modulación digital, utilizada para la transmisión de datos. Para BFSK que corresponde al mismo proceso, pero con datos binarios, se utilizan dos frecuencias diferentes para transmitir dos señales, es decir 0 y 1. como se puede ver a continuación:

$$S_1(t) = A \cos(\omega_1 t)$$

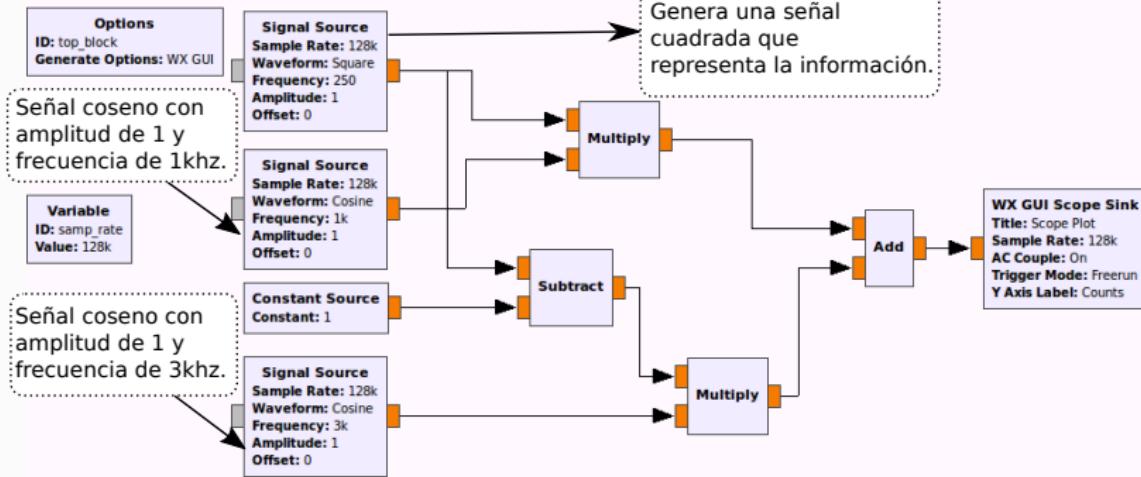
$$S_2(t) = A \cos(\omega_2 t)$$

- El diagrama de boques en GRC se utilizaron dos señales coseno como portadora de amplitud 1, y frecuencias de 1Khz y 10Khz. La señal 1 es representada por una onda cuadrada, y la señal 0 es obtenida restándole 1 a la señal cuadrada. Posteriormente se multiplican las señales 0 y 1 con las portadoras y luego se suman obteniendo la modulación FSK.

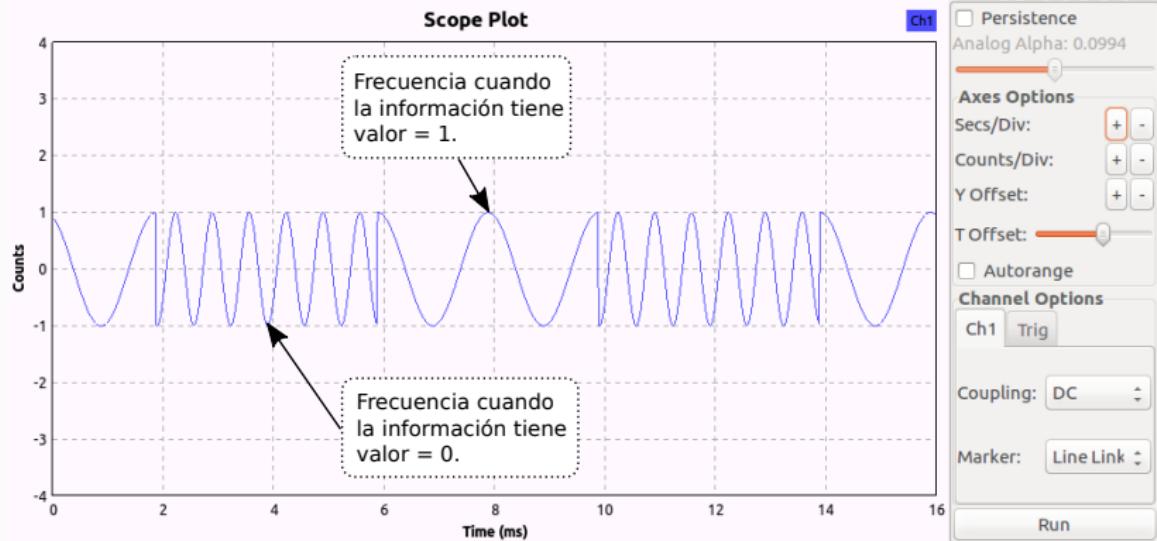
# Modulación ASK en GRC



# Modulación ASK en GRC



# Modulación ASK en GRC



## Modulacion ASK en GRC

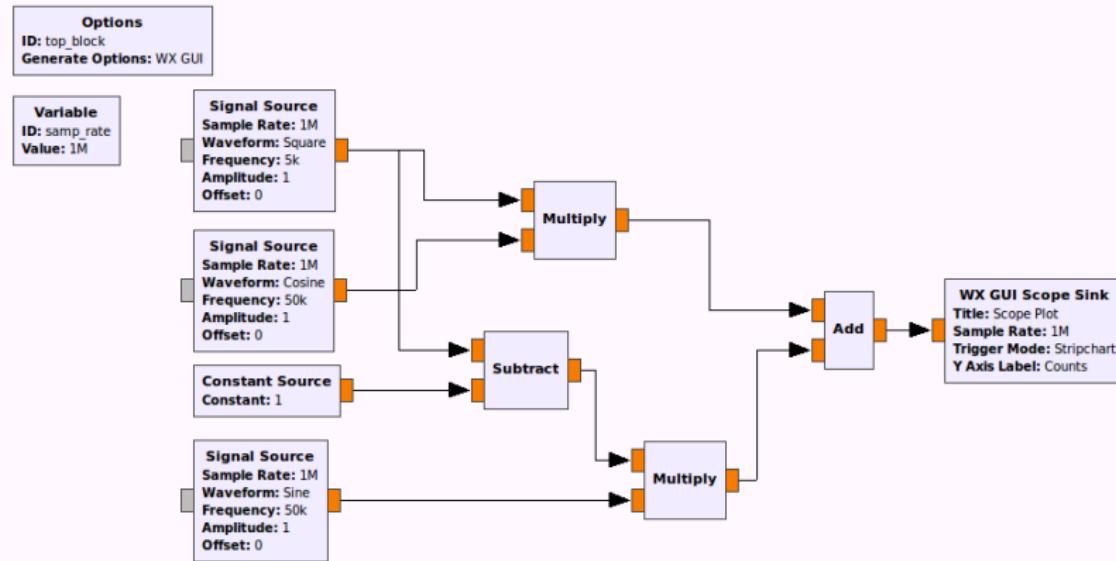
- La Modulación por desplazamiento de fase (PSK) es un esquema de modulación digital donde se varía la fase de la señal, manteniendo la frecuencia y la amplitud constante. Para la modulación PSK se usan dos señales con fases diferentes y frecuencias iguales, estas se multiplican con la señal 0 o 1 de los datos como se muestra a continuación:

$$S_1(t) = A \cos(\omega t)$$

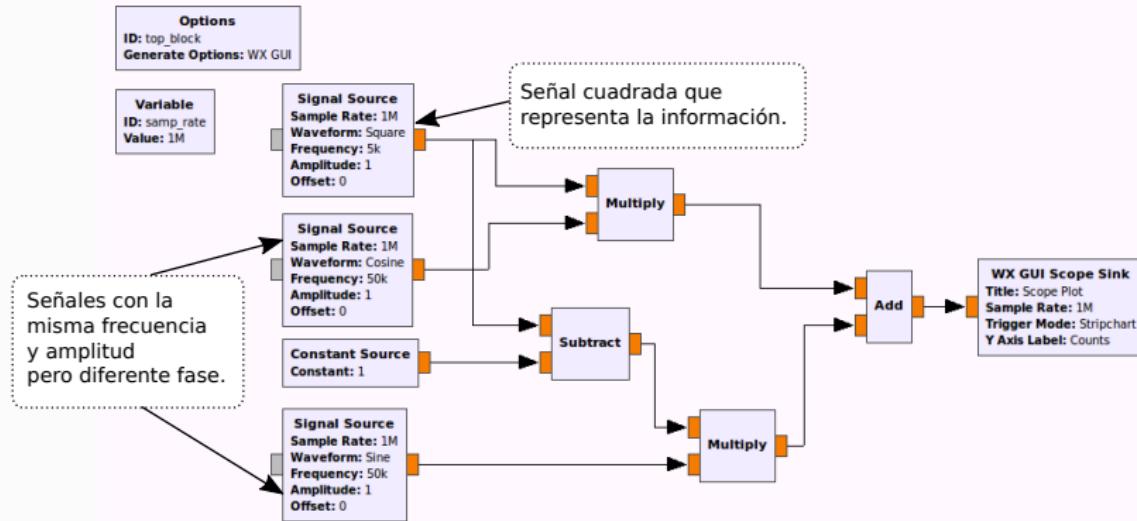
$$S_2(t) = -A \sin(\omega t)$$

- El diagrama de bloques en GRC se utiliza una señal coseno de 50000 Hz y la amplitud pico de 1, y otra onda sinusoidal de igual frecuencia y amplitud. Similar a BFSK se representa la señal 1 con la onda cuadrada de frecuencia 5000 Hz y el 0 por la resta de la constante[2].

# Modulación ASK en GRC

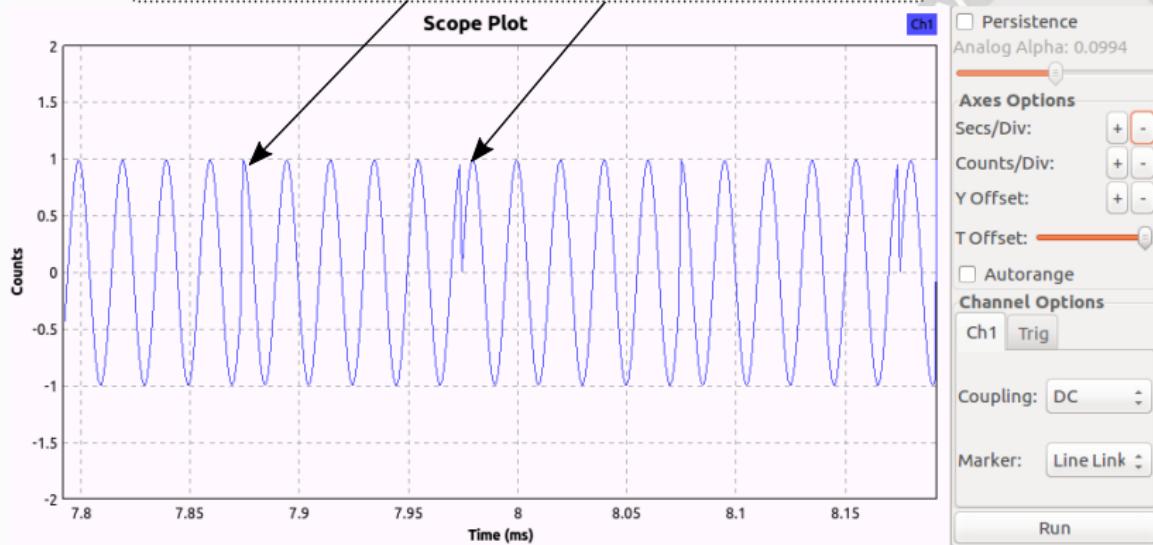


# Modulación ASK en GRC



# Modulación ASK en GRC

Las variaciones de fase en la señal representan el cambio entre 0 y 1 de los datos, en este caso una onda cuadrada con frecuencia de 5KHZ.



## Actividades

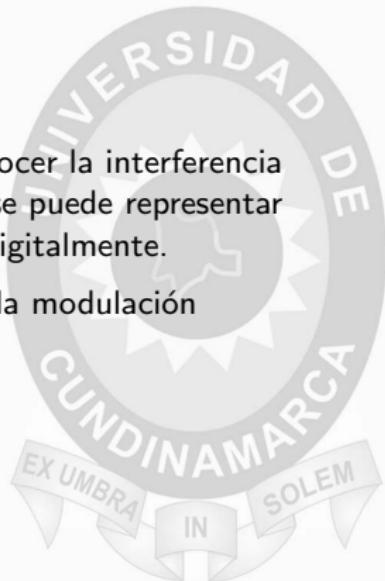
# ¿Comó Observar un diagrama de constelación?

Para visualizar la constelación en Gnu-Radio podemos utilizar los siguientes bloques:

- QT GUI Constellation Sink: Permite mostrar la constelación I-Q de múltiples señales, su receptor admite el trazado de transmisión de datos o mensajes complejos, el puerto de entrada para el mensaje se denomina "in".
- WX GUI Constellation Sink : es la representación gráfica de las señales complejas, es útil para el estudio y visualización de modulaciones digitales como: M-PSK, QAM, FSK, ASK

Los diagramas de constelación se pueden utilizar para reconocer la interferencia y distorsión existente en una señal, por otra parte también se puede representar la relación de amplitud y fase de una portadora modulada digitalmente.

Responda: ¿Se puede ver el diagrama de constelaciones de la modulación ASK? ¿y que se debe observar?

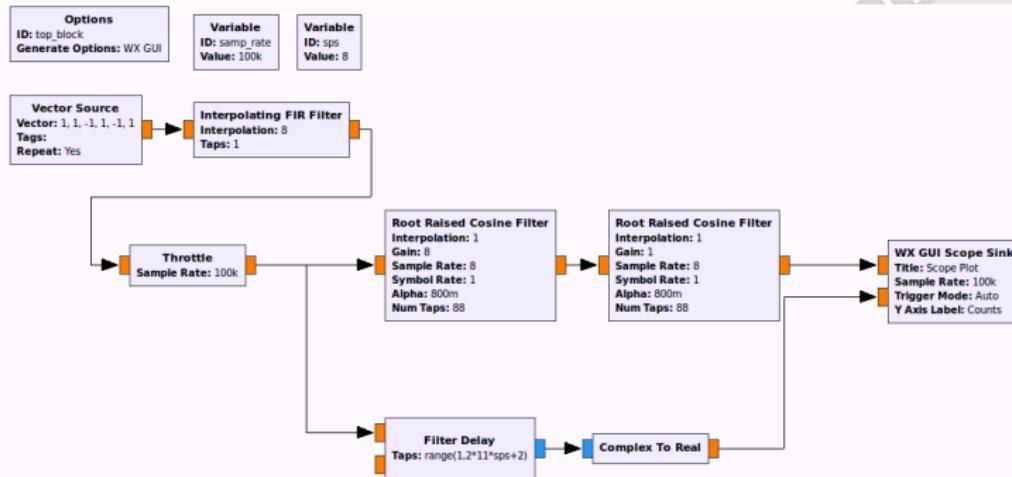


# Lab5

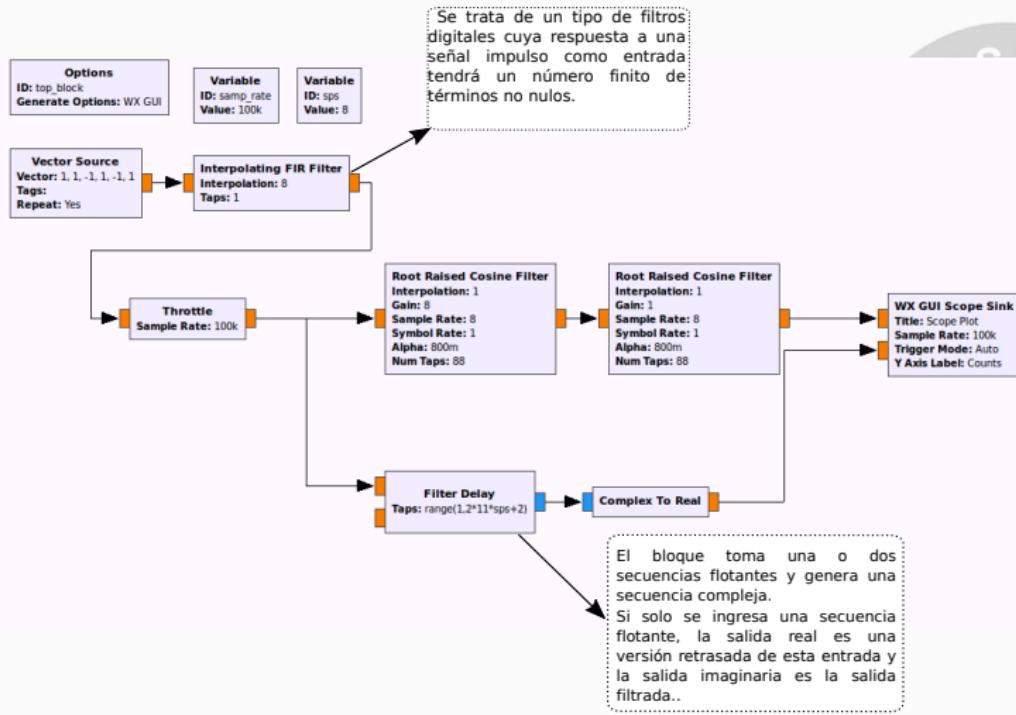
## Flujo de datos digitales BPSK en GRC

# Flujo de datos digitales BPSK

Cómo convertir un flujo de datos digitales en una señal analógica de banda base utilizando un filtro FIR interpolador.[13]



# Flujo de datos digitales BPSK



# Flujo de datos digitales BPSK

## Filtros FIR

Un filtro FIR es aquel que tiene una respuesta finita al impulso y que se caracterizan por ser sistemas no recursivos. Un filtro FIR de orden  $L$  se describe mediante la ecuación en diferencias:

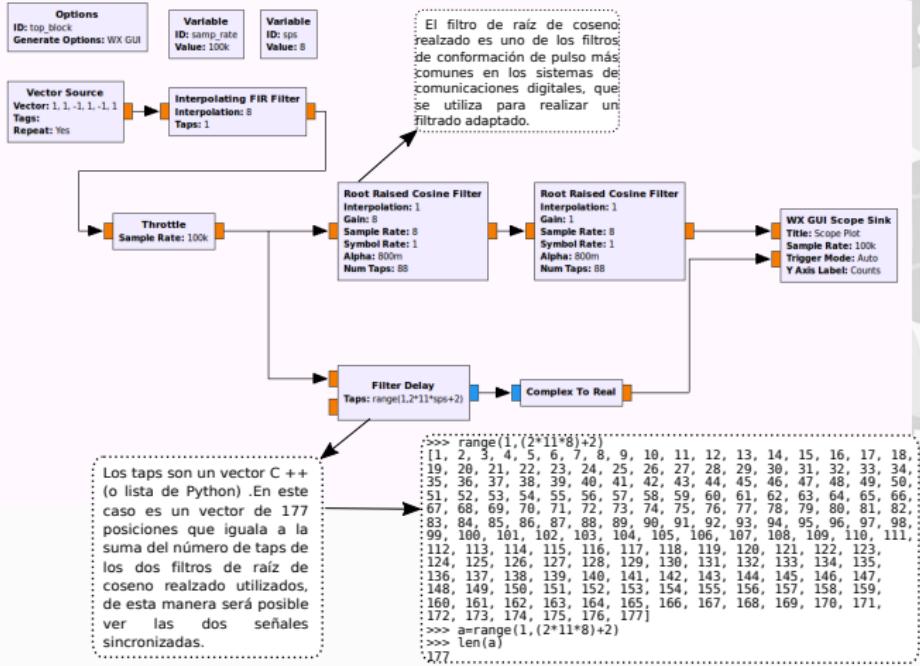
$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) + \dots + a_Lx(n-L)$$

Donde la secuencia  $a_k$  son los coeficientes del filtro. A partir de esta ecuación en diferencias puede obtenerse la función de transferencia del filtro en el dominio de  $Z$ .

$$F(z) = \sum_{k=0}^{L-1} a[k] z^{-k}$$

En este tipo de filtrado no existe retroalimentación. Además, la respuesta al impulso  $H(w)$ , es de duración finita ya que si la entrada se mantiene en cero durante  $L$  periodos consecutivos la salida también será cero.

# Flujo de datos digitales BPSK

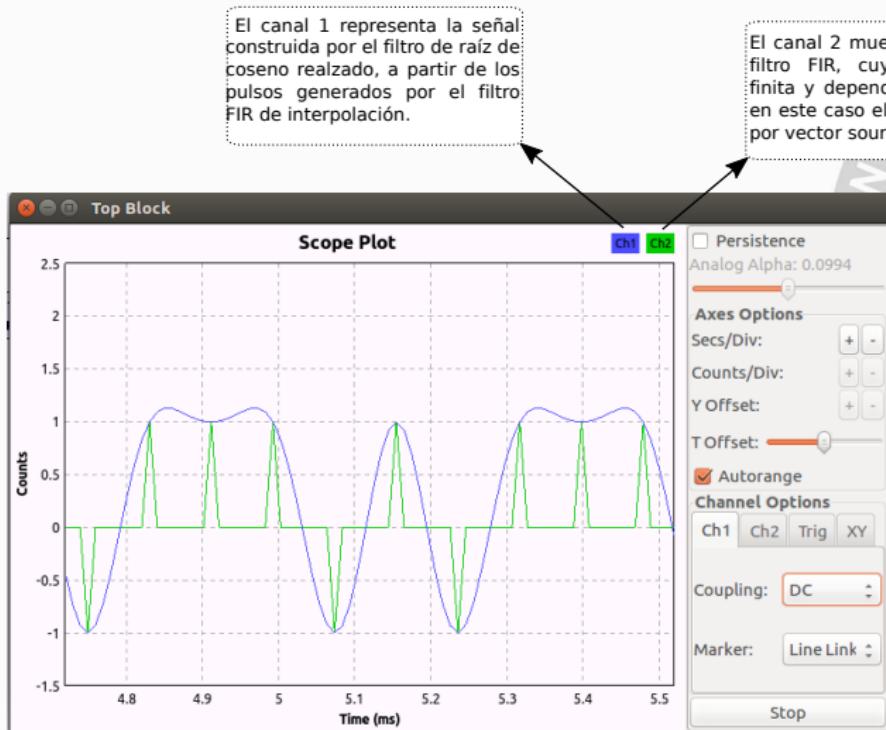


# Flujo de datos digitales BPSK

¿Qué es y para qué se utiliza el Root Raised Cosine Filter?

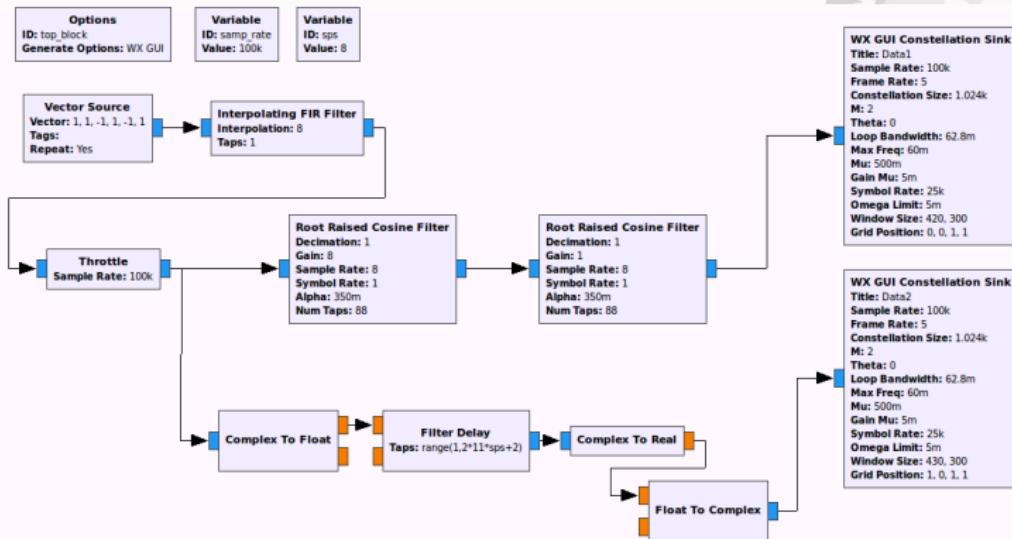
Uno de los principales inconvenientes de todas las formas de onda de la señal es que, aunque pueden controlar muy bien las emisiones de energía dentro del ancho de banda de interés, envían cantidades relativamente altas de energía de esta. Una forma práctica de reducir los lóbulos laterales del espectro de las señales de navegación podría ser usar un filtro de coseno realzado (RCF), ya que tiene un ancho de banda limitado. El filtro de coseno realzado es un caso particular del filtro Nyquist. Los pulsos de Nyquist (filtros) son pulsos que no producen interferencia entre símbolos (ISI) en el momento del muestreo.[14]

# Flujo de datos digitales BPSK

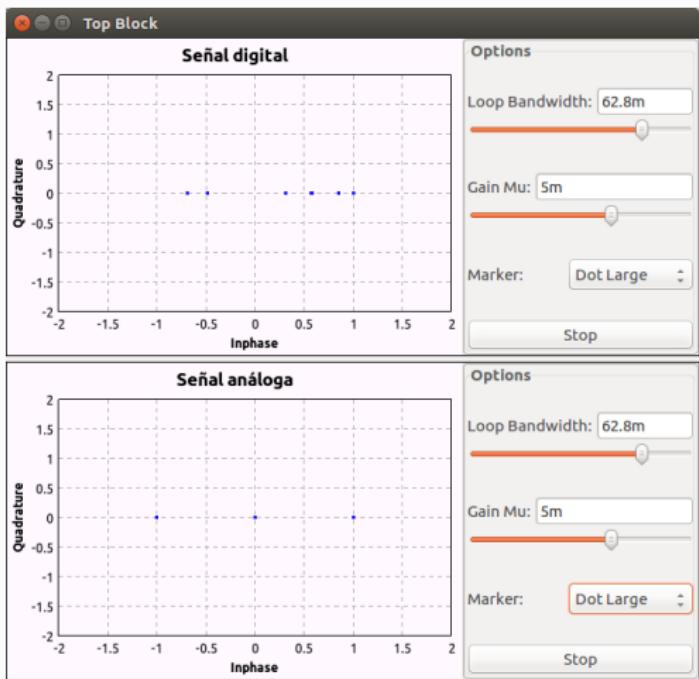


# Flujo de datos digitales BPSK

Puesto que el diagrama de constelación es un método de representación en el plano complejo de los estados de símbolo en términos de amplitud y fase en los esquemas de modulación digital, será necesario para ver este diagrama utilizar un flujo de datos complejos, como se ve a continuación:



# Flujo de datos digitales BPSK



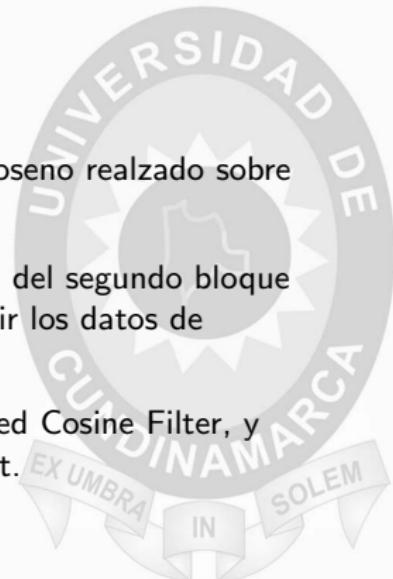
## Actividades

# Flujo de datos digitales BPSK

## Actividad

¿Qué efectos produce el parámetro  $\alpha$  del filtro de raíz de coseno realizado sobre la señal y el ancho de banda?

- Grafique en un bloque de WX GUI FFT Sink la salida del segundo bloque de Root Raised Cosine Filter. Recuerde antes convertir los datos de flotantes a tipo complejo.
- Realice cambios al parámetro  $\alpha$  del bloque Root Raised Cosine Filter, y observe los cambios en el FFT Plot y en el Scope Plot.

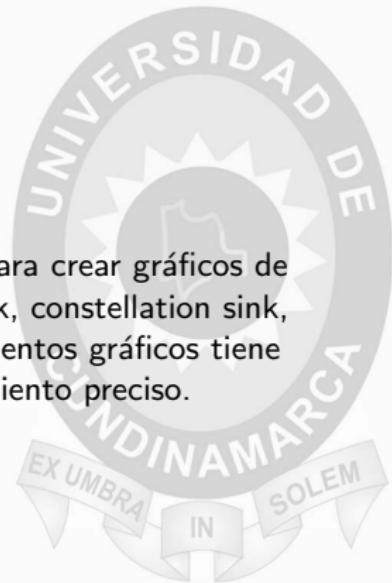


# Interfaces gráficas con WX GUI

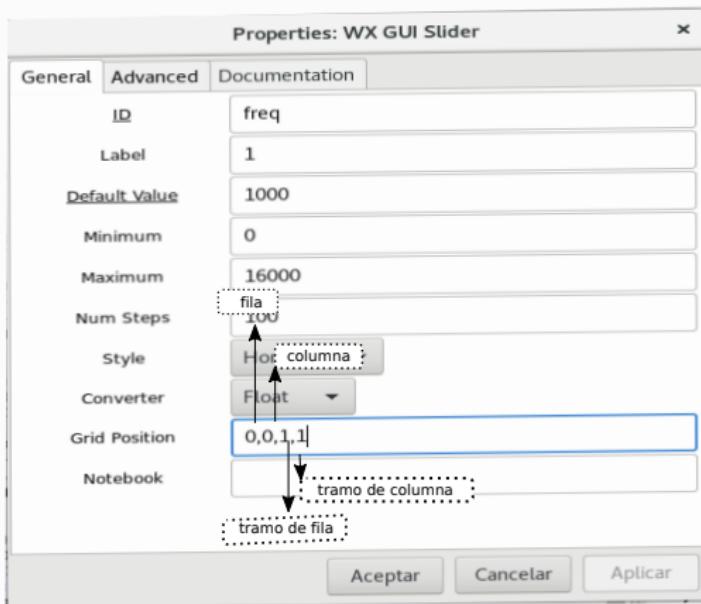
# Grid position

## Posicionamiento de rejilla

GRC ofrece varios receptores gráficos y controles gráficos para crear gráficos de flujo wx-gui.(scope sink, fft sink, number sink, waterfall sink, constellation sink, slider control, and chooser control) Cada uno de estos elementos gráficos tiene un parámetro de posición de cuadrícula para un posicionamiento preciso.



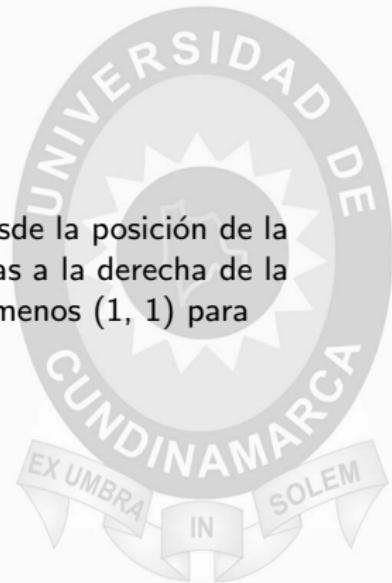
# Grid position



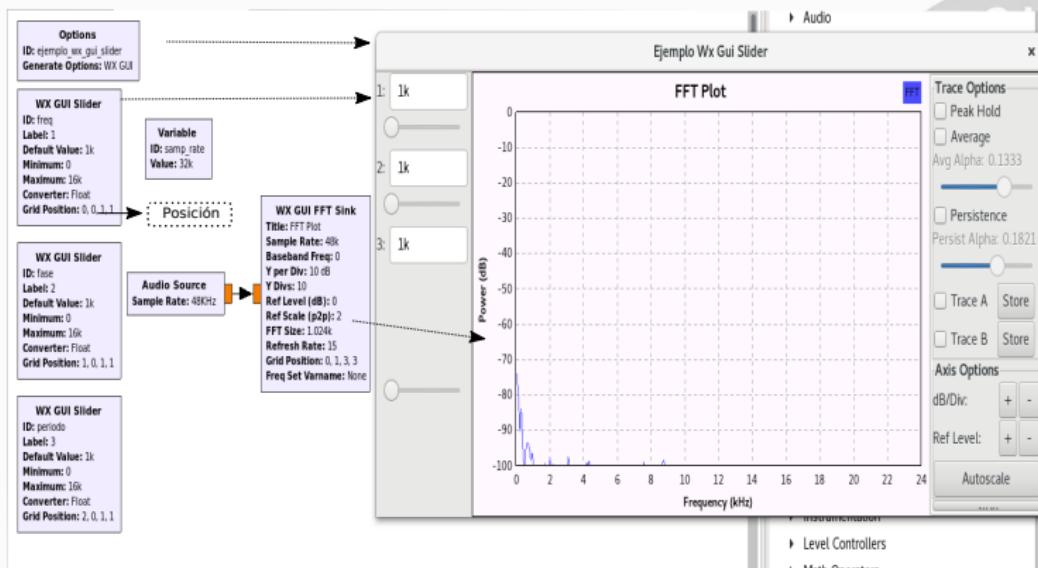
Un parámetro de posición de cuadrícula es una lista de 4 enteros de la forma (fila, columna, tramo de fila, tramo de columna). La fila y la columna especifican la posición de la esquina superior izquierda del elemento gráfico. La posición más pequeña, la posición (0, 0), especifica la esquina superior izquierda de la cuadrícula.

## Grid position

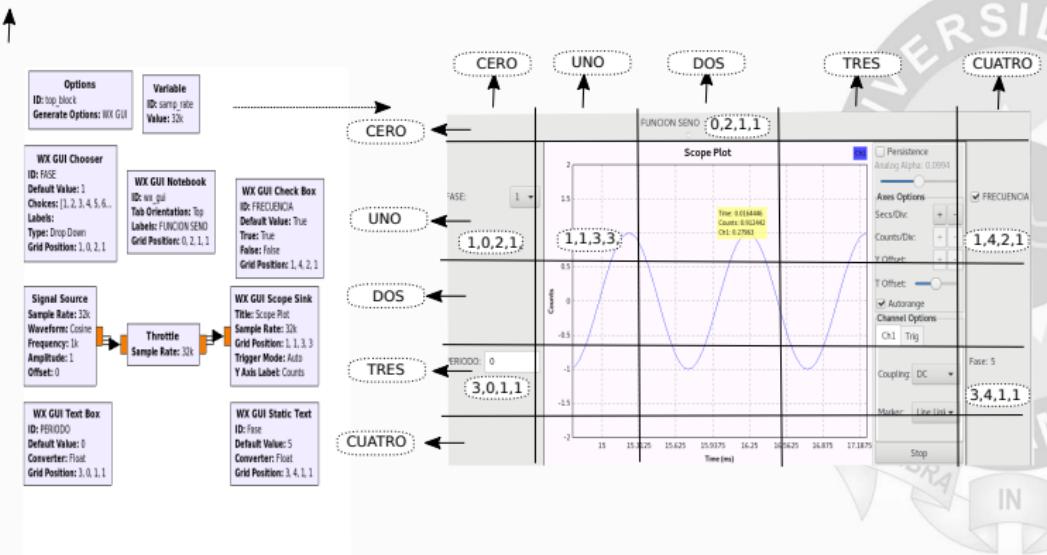
El tramo de fila especifica el número de filas hacia abajo desde la posición de la fila, y el tramo de columna especifica el número de columnas a la derecha de la posición de la columna. Por lo tanto, el tramo debe ser al menos (1, 1) para ocupar el mínimo de 1 celda de cuadrícula.



# Grid position



# Grid position



# CONFIGURACIÓN E INSTALACIÓN DE HARDWARE

## Parte II - Tabla de contenidos

Lab6: RTL

137

Lab7: HackRF One

162

# Lab6

## RTL

## RTL\_SDR

El RTL\_SDR es un dispositivo el cual permite combinar tanto la parte de software como hardware para implementar un sistema con el cual sea posible desarrollar procesos de radiocomunicaciones tanto de recepción como de transmisión de señales.

El RTL se creó principalmente para solucionar los problemas de compilación para equipos con procesamiento de Pentium 4, esta tarjeta está diseñada para poder demodular señales en FM, AM y SSB con diferentes anchos de banda; también tiene la capacidad de analizar más de cien frecuencias en un segundo.

En general el RTL es una gran ayuda a nivel educativo ya que ayuda a hacer una estación receptora si no se tiene una.

# Instalación de paquetes

En la siguiente guía práctica se dará un ejemplo por medio de uso de terminal y órdenes, para la captura de señales de radio FM, escáner de la policía, airband scanner y decodificador de localización.



# Instalación de Paquetes

Para instalar los paquetes desde consola se debe ingresar las siguientes órdenes.

## Paquetes de SOX

- Mediante esta orden, es posible la descarga de las bibliotecas o paquetes necesarios para el correcto y completo funcionamiento de SOX.

```
box@user~$ sudo apt-get install libsox-fmt-alsa
```

- Este paquete contiene la mayoría de las bibliotecas de formatos de audio compatibles con SOX

```
box@user~$ sudo apt-get install libsox-fmt-base
```

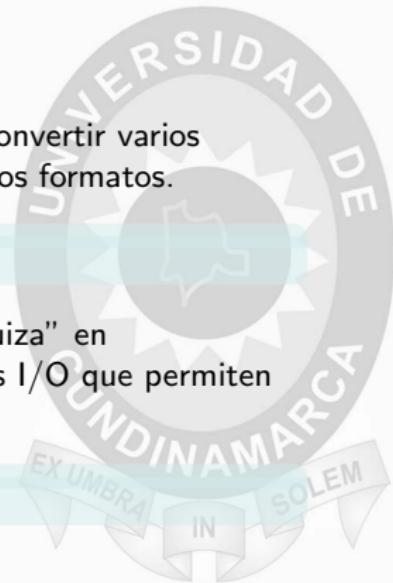
# Instalación de Paquetes

- Este paquete contiene la biblioteca sox que permite convertir varios formatos de archivos de audio de computadora en otros formatos.

```
box@user~$ sudo apt-get install libsox2
```

- SOX es una herramienta conocida como “la navaja suiza” en manipulación de audio, ya que contiene las bibliotecas I/O que permiten el procesamiento del audio en diversos formatos.

```
box@user~$ sudo apt-get install sox
```



# Instalación de Paquetes

- Este paquete contiene una biblioteca compartida.

```
box@user~$ sudo apt-get install  
libgnuradio-osmosdr0.1.4
```

- Este paquete es el software que proporciona el control del hardware USB e independientemente de la API, pasar datos a aplicaciones de radio definidas por software en el host. Tiene la posibilidad de recibir o transmitir información.

```
box@user~$ sudo apt-get install libosmosdr0
```



# Instalación de Paquetes

## Paquetes de RTL-SDR

- Es un software que soporta dispositivos SDR tales como Airspy, funcube Dongles, rtl-sdr, HackRF, y USRP

```
box@user~$ sudo apt-get install gqrx-sdr
```

- Este archivo contiene bibliotecas y archivos de desarrollo que complemetan el funcionamiento de conexión de la RTL.

```
box@user~$ sudo apt-get install librtlsdr-dev
```



# Instalación de Paquetes

- Este paquete contiene una biblioteca compartida.

```
box@user~$ sudo apt-get install librtlsdr0
```

- Este paquete permite el recepcionamiento I/Q para dispositivos basados en la RTL2832.

```
box@user~$ sudo apt-get install rtl-sdr
```



## Probar la RTL

Para iniciar es necesario abrir una nueva terminal, desde allí se realizará el procedimiento para comprobar la conexión del hardware RTL con su computadora. Se debe ingresar la siguiente orden:

```
box@user~$ rtl_test
```

rtl\_test es una herramienta verificadora de la conexión para receptores DVB-T basados en RTL2832.

Si existe una respuesta satisfactoria evidenciará el siguiente mensaje:

```
Info: This tool will continuously read from the device,  
and report if samples get lost. If you observe no  
further output, everything is fine. Reading samples in  
async mode... cb transfer status: 1, canceling...
```

## Probar la RTL

Si el hardware RTL no está conectado evidenciará el siguiente mensaje:

No supported devices found.

Al conectar más de un dispositivo RTL y ejecutar la orden `rtltest`, se visualiza un listado de los dispositivos con sus respectivas referencias, como se muestra a continuación.

Found 2 device(s) :

0: Realtek, RTL2838UHIDIR, SN: 00000001

1: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM

Found Rafael Micro R820T tuner

Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7

12.5 14.4 15.7 16.6 19.7 20.7 22.9 25.4 28.0 29.7 32.8

33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0 49.6

[R82XX] PLL not locked!

Sampling at 2048000 S/s.

## Probar la RTL

Al ser desconectada la RTL mientras se ejecuta el test lo notifica mediante el siguiente mensaje:

```
Reading samples in async mode...
lost at least 84 bytes
cb transfer status: 5, canceling...
Library error -5, exiting...
```



# Receptor de radio pública FM

Para esta práctica se debe realizar desde la terminal, puede utilizar la misma o abrir una nueva. Se ingresa la siguiente orden:

```
box@user~$ rtl_fm -M wbfm -f 103.3M -s 1000k -l 0  
|play -r 32k -t raw -e s -b 16 -c 1 -V1 -
```



## Receptor de radio pública FM

En donde cada uno de los componentes del mando `rtl_fm`, tienen una función específica para poder captar la señal de radio FM.

- `-f 103.3M` : Indica la frecuencia de sintonización, es decir la de la Radio Pública a escuchar, para este caso 103.3MHz.
- `-M wbfm` : Define el tipo de modulación, WBFM (banda ancha).Por defecto es fm (NBFM).Las opciones para -M son, fm, wbfm, raw, am, usb, lsb.
- `-s 1000k` : Indica que se toma una tasa de muestreo de 1MS/s
- `-l 0` : Deshabilita el squelch (actúa para suprimir la salida de audio de un receptor en ausencia de una señal de entrada deseada suficientemente fuerte)

# Receptor de radio pública FM

Para ver más opciones de `rtl_fm`, ejecute lo siguiente:::

```
box@user~$ rtl_fm --help
```

Ahora bien, se utiliza el carácter | el cual es un conector para relacionar la orden `rtl_fm` con la orden `play` para poder escuchar en el computador, la emisora sintonizada.

```
box@user~$ rtl_fm... | play -r 32k -t raw -e s -b 16  
-c 1  
-V1 -
```

## Receptor de radio pública FM

play es un reproductor que lee y escribe archivos de audio en los formatos más populares y opcionalmente puede aplicarles efectos. Puede combinar múltiples fuentes de entrada, sintetizar audio y, en muchos sistemas, actuar como un reproductor de audio de propósito general o un grabador de audio multimedia.

Cada uno de los componentes de play, tiene una función específica, para la reproducción de la señal captada:

- **-r** : Usa números aleatorios por defecto (lo mismo en cada ejecución de SOX)
- **-t** : Tipo de archivo de audio FILETYPE
- **-s** : Muestra el progreso mientras procesa los datos de audio fuerte)
- **-b** : BITS Tamaño de muestra codificado en bits
- **-c** : Factor de compresión para formato de salida
- **-v** : Factor de ajuste del volumen del archivo de entrada (número real)
- **raw** : Enlaza un dispositivo de caracteres brutos de Linux

## Receptor de radio pública FM

Para ver más opciones de play, ejecute lo siguiente:

```
box@user~$ play -help
```

Finalmente, se obtendrá una respuesta, en donde se podrá visualizar la recepción de radio FM y escuchar la emisora sintonizada.



# Instalar GQRX

Para instalar la última versión de gqrx añada backports a su archivo sources.list

```
box@user~$ sudo echo "deb  
http://ftp.ch.debian.org/debian jessie-backports main"  
>> /etc/apt/sources.list
```

Asegúrese de que su índice local esté actualizado y luego instale los paquetes necesarios:

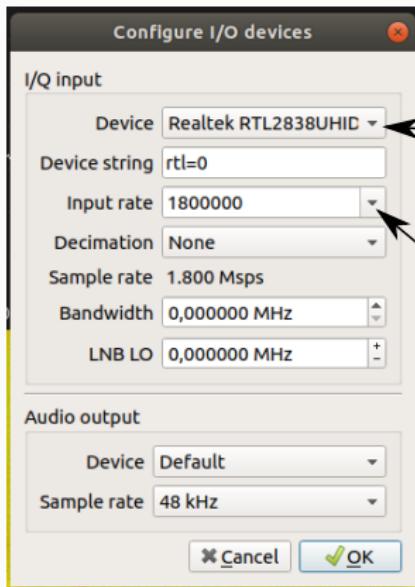
```
box@user~$ sudo apt-get update  
box@user~$ sudo apt-get install -t jessie-backports  
gqrx-sdr rtl-sdr
```

Una vez que los paquetes están instalados, conecte el RTL-SDR a la computadora y luego inicie la GUI gqrx desde un terminal:

```
box@user~$ gqrx
```

# gqrx FM

Una vez se abre gqrx, se le presenta esta ventana

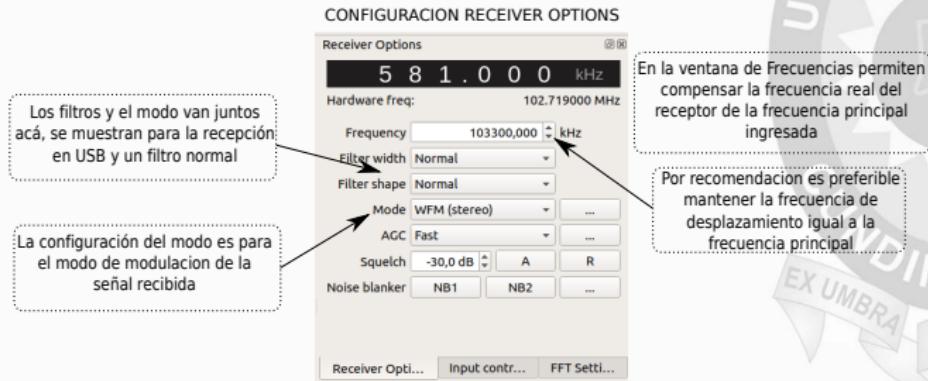


Asegúrese de que la línea del dispositivo muestre Realtek RTL o el dispositivo que está utilizando

Frecuencia de muestreo en la entrada

## gqrx Configuración receptor

Una vez presionamos OK se abre una ventana y en la parte derecha vemos la configuración del demodulador:



# gqrx Configuración filtros

Los filtros pueden ser Normal, Estrecho Ancho o configurado por el usuario.

Filtros estrechos: son útiles contra la interferencia de las estaciones en las frecuencias cercanas.

AGC , se puede configurar en apagado, usuario, lento, medio, rápido.

Se elige el que mejor se ajuste al desvanecimiento de propagación actual  
Por recomendación se tiende a usar el modo AGC Rápido



El silenciamiento se ajusta cuando se desea monitorear una frecuencia fija, por ejemplo VHF selección el botón A para ajustar de forma automática el nivelado del ruido de fondo actual.

NB1 y NB2 activan eliminadores de ruido para atenuar la estática y el ruido de tipo pulsante

# gqrx Configuración filtros

CW-U o CW-L : para la modulacion On-Off uso de la banda lateral superior o inferior dependiendo de la posible interferencia de otra portadora cercana

USB o LBS para la modulacion de amplitud de banda lateral unica, banda lateral superior o inferior

WFM (estéreo o mono) para modulación de frecuencia ancha utilizada por estaciones de difusión amplia

CONFIGURACIÓN DE MODE

Receiver Options

5 8 1 . 0 0 0 kHz

Hardware freq: 102.719000 MHz

Frequency 103300,000 kHz

Filter width Normal

Filter shape Normal

Mode WFM (stereo) **...**

AGC Fast **...**

Squelch -30,0 dB A R

Noise blanker NB1 NB2 **...**

Receiver Opt... Input contr... FFT Setti...

AM : modulacion en amplitud de estilo antiguo, utilizada por las estaciones de radiodifusion

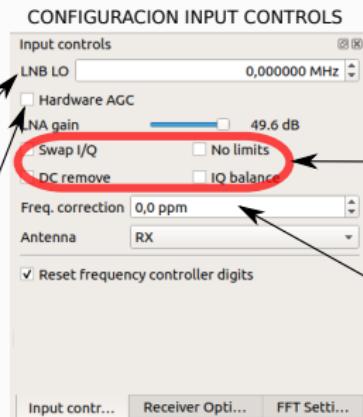
Demod Off: apaga el procesamiento de la señal. GQRX puede ser usado para ver el espectro en tiempo real

FM estrecho: VHF, marino

# gqrx Reducción de interferencia

LNB LO se ingresa una frecuencia de oscilador local.  
Al usar SDR-RTL se mantiene en 0.00000MHz

Hardware AGC en algunos casos que se presente una interferencia grave de una interferencia local, se apaga el AGC y reduce la ganancia LNA y el mezclador y se observa el efecto en el espectro de RF

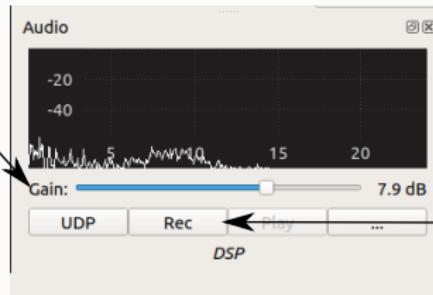


I- I/Q intercambia los canales I y Q normalmente se deja desactivado  
- Sin Limites: solo se usa, si se utiliza la SDR más allá del dominio de frecuencia recomendado.  
- DC rem: Elimina el sesgo de CC que se ve a menudo en los receptores de conversión directa  
- IQ rem: Se utiliza si se presenta imágenes fatastica en el espectro

El corrector de frecuencias funciona cuando se iguala la frecuencia principal a la frecuencia de referencia. En el cual 10 ppm es 10Hz de una señal de 1MHz, 1.44kHz de una señal de 144MHz.

## gqrx Audio

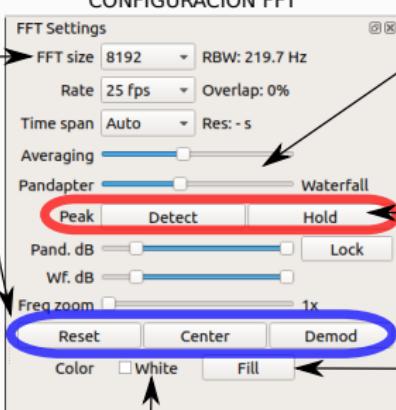
Se puede cambiar la escala del espectro, así mismo el nivel de ganancia aumentando el nivel de salida del altavoz /auricular



Se utiliza para realizar una grabación en la que puede cambiar el directorio en el que GQRX guardará la señal de audio recibida en un archivo .wav

# gqrx Configuración de la gráfica FFT

En la parte inferior derecha encontramos una ventana donde se configura la ganancia del audio



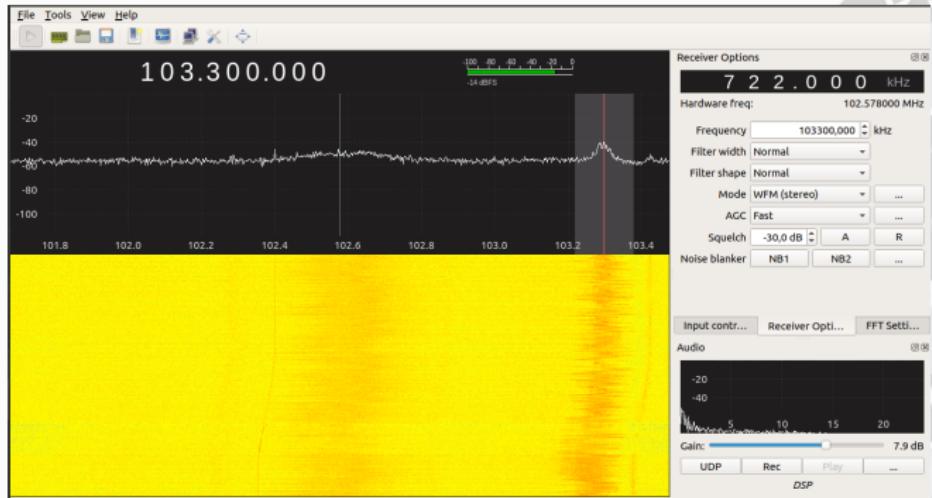
The screenshot shows the 'CONFIGURACION FFT' window with the following settings:

- FFT Settings:**
  - FFT size: 8192
  - RBW: 219.7 Hz
  - Rate: 25 fps
  - Overlap: 0%
  - Time span: Auto
  - Res: - s
- Averaging:** A slider.
- Pandapter:** A slider with options: Peak, Detect, Hold, Lock, and WF. dB.
- Waterfall:** A slider.
- Buttons:** Peak, Detect, Hold, Lock, Pand. dB, WF. dB, Freq. zoom, Reset, Center, Demod.
- Color:** White, Fill.

Annotations with arrows pointing to specific controls:

- Left Annotation:** El promedio de FFT se utiliza para el nivel de ruido. Esto solo afecta a la parcela de Pandapter.
- Center Annotation:** Permite modificar la parte de la pantalla principal asignada al espectro.
- Bottom Annotation:** Cambia el color de la visualización del espectro de RF.
- Right Annotations:**
  - Reset : restablece la visualización del espectro de RF a su valor del 100%
  - Center: centrado la visualización del espectro de RF alrededor de la frecuencia original
  - Demod: centrar la visualización del espectro de RF alrededor de la frecuencia del demodulador
  - El botón Detect unirá pequeños círculos a los picos
  - El botón Hold mostrara débilmente el espectro de RF de la envolvente
  - Llenará/Borrará el espacio debajo de la línea del espectro con un degradado de color

# gqrx Resultados

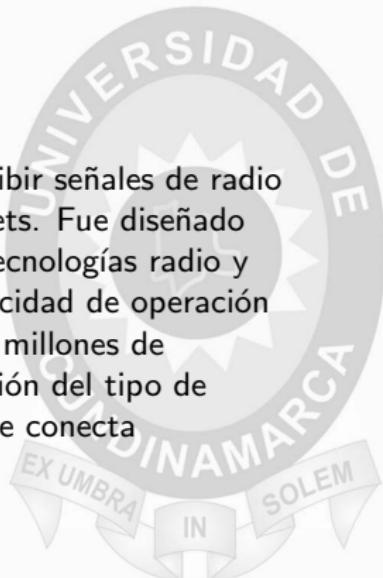


# Lab7

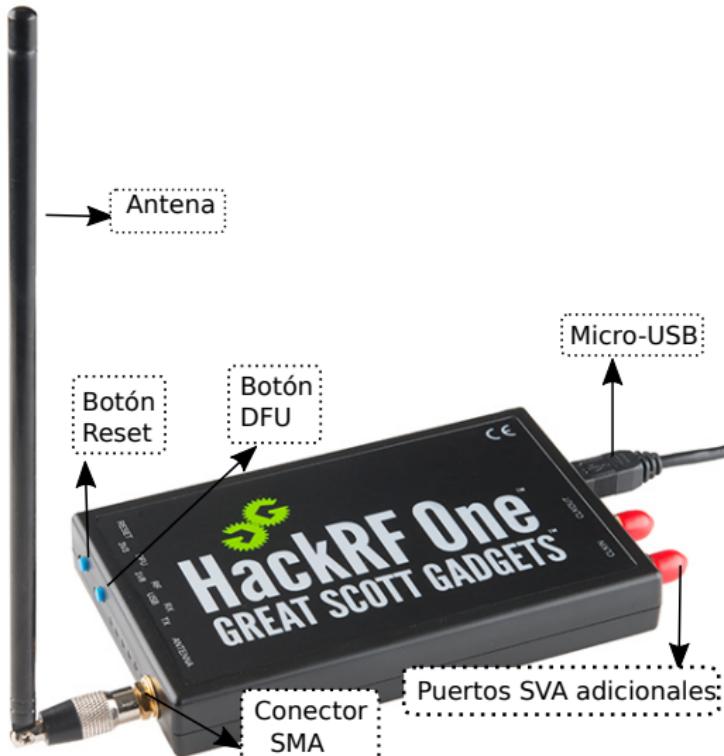
## HackRF One

# HackRF One

HackRF One es un periférico SDR capaz de transmitir o recibir señales de radio desde 1 MHz hasta 6 GHz, fabricado por Great Scott Gadgets. Fue diseñado para facilitar el desarrollo para las nuevas generaciones de tecnologías radio y sus correspondientes protocolos. Es un transceiver con capacidad de operación half-duplex . Tiene una capacidad de muestreo de hasta 20 millones de muestras por segundo, pudiéndose alcanzar las 21,5 en función del tipo de controlador USB 2.0 HS que incluya el computador al que se conecta



## Características



# Instalación de herramientas de la HackRF One

- Conecte el HackRF One a la computadora con el cable Micro-USB a USB. Confirme que los primeros tres LEDs se iluminan para asegurar que el dispositivo esté funcionando.
- Las bibliotecas se pueden instalar con la siguiente orden en la terminal:

```
box@user~$ sudo apt-get install hackrf
```

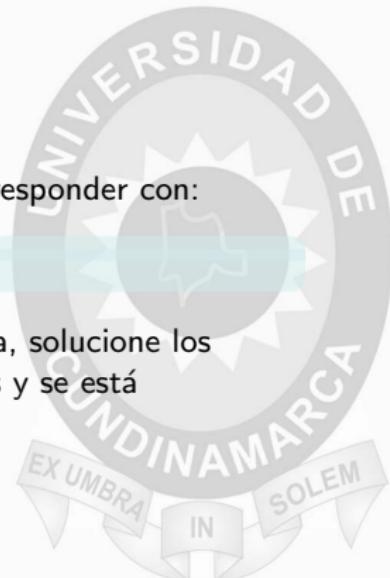
- Despues de la instalación, digite la orden `hackrf_info` para verificar que el dispositivo HRF1 esté conectado.

# Instalación de herramientas de la HackRF One

- Estas son algunas herramientas utilizadas en HackRF One:
  - *hackrf\_info* : permite al usuario probar el dispositivo y mostrar la configuración.
  - *hackrf\_max2837* : permite al usuario controlar el chip max2837
  - *hackrf\_spiflash* : permite al usuario configurar el Flash incorporado.
  - *hackrf\_transfer* : permite al usuario recibir datos de RF y transmitir datos a RF.
  - *hackrf\_si5351c* : permite al usuario controlar el chip si5351c.
  - *hackrf\_cpldjttag* : permite al usuario configurar el CPLD integrado.

# Instalación de herramientas de la HackRF One

- Después de una instalación exitosa, la terminal debe responder con:  
`found hackrf board`
- Si la terminal muestra que la HRF1 no está conectada, solucione los problemas verificando que los cables estén enchufados y se está suministrando potencia al dispositivo



# Actualización del firmware

## Paso 1 Actualizar el firmware de SPI Flash

Para actualizar el firmware en un HackRF One que funcione, use el programa `hackrf_spiflash`:

```
box@user~$ hackrf_spiflash -w hackrf_one_usb.bin
```

Puede encontrar el firmware binario (`hackrf_one_usb.bin`) en el directorio `firmware-bin` del último paquete de versiones o puede compilar el suyo desde la fuente. Para Jawbreaker, use `hackrf_jawbreaker_usb.bin`. Si compila desde el origen, el archivo se llamará `hackrf_usb.bin`.

# Actualización del firmware

## Paso 2 Actualizar el CPLD

Para actualizar a la última imagen CPLD, primero actualice el firmware SPI flash, libhackrf y hackrf-tools. Entonces en la terminal se ingresa lo siguiente:

```
box@user~$ hackrf_cpldjttag -x firmware / cpld /  
sgpio_if / default.xsvf
```

Después de unos segundos, tres LEDs deberían comenzar a parpadear. Esto indica que el CPLD se ha programado con éxito. Restablezca el dispositivo HackRF presionando el botón RESET o desenchufando y enchufando de nuevo.

# Instalación de paquetes para trabajar con GNU Radio: osmocom

A continuación instalamos los paquetes de osmocom.

```
box@user~$ git clone  
git://git.osmocom.org/gr-osmosdr
```

Nos movemos dentro de la carpeta clonada:

```
box@user~$ cd gr-osmosdr
```

Creamos el directorio de construcción, nos movemos dentro y hacemos el cmake.

```
box@user~$ mkdir build && cd build  
box@user~$ cmake .. /
```



# Instalación de paquetes para trabajar con GNU Radio: osmocom

Construimos e instalamos.

```
box@user~$ make  
box@user~$ sudo make install  
box@user~$ sudo ldconfig
```



# LABORATORIOS CON SOFTWARE Y HARDWARE

## Parte III - Tabla de contenidos

Lab8: Receptor FM monofónico	174
Lab9: Transmisor FM monofónico	189
Lab10: Receptor FM estereofónico	201
Lab11: Transmisor FM estereofónicofónico	208
Lab12: Retransmisión emisora mp3 online en FM estéreo	216
Lab13: Recepción de señales de radio FM para Walkie-Talkie UHF	225
Lab14: Transmisión de señales de radio FM para Walkie-Talkie UHF	241
Lab15: Receptor AM	260
Lab16: Transmisor AM	271

# Lab8

## Receptor FM monofónico

## Receptor FM monofónico

En este laboratorio, se realizará la simulación de un receptor FM monofónico en el cual se muestra la función de cada uno de los bloques que conforma dicho receptor, además se muestra el proceso matemático para llegar a 48 KHz de la tasa de muestreo que se va a establecer.



## Modulación de frecuencia

Se refiere a la forma de transmitir información a través de una onda portadora variando su frecuencia. En este tipo de modulación la variación se produce en los saltos de frecuencias.

La modulación de frecuencia se usa comúnmente en las radiofrecuencias de muy alta frecuencia por la alta fidelidad de la radiodifusión de la música y el habla. El sonido de la televisión analógica también se difunde por medio de FM.

La modulación de frecuencia también se utiliza en las frecuencias de audio para sintetizar sonido. Esta técnica, conocida como síntesis FM, fue popularizada a principios de los sintetizadores digitales y se convirtió en una característica estándar para varias generaciones de tarjetas de sonido de computadoras personales.

# Modulación de frecuencia

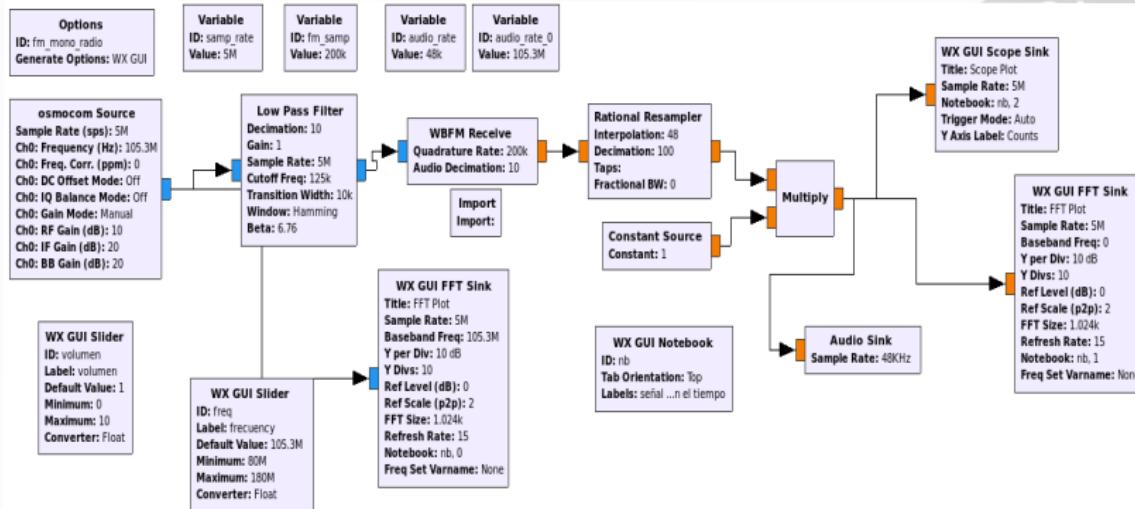
## Aplicaciones en radio

Dentro de las aplicaciones de FM se encuentra la radio, donde los receptores emplean un detector de FM y el sintonizador es capaz de recibir la señal más fuerte de las que transmiten en la misma frecuencia. Otra de las características que presenta FM es la de poder transmitir señales estereofónicas.

## Otras aplicaciones

Entre otras de sus aplicaciones se encuentran la televisión, SECAM: El sistema de televisión en color SECAM modula la información de color en FM. como sub-portadora de sonido; en micrófonos inalámbricos; y como ayuda en navegación aérea, En los sistemas de vídeo analógicos, incluyendo VHS, para registrar la luminancia (blanco y negro) de la señal de video[3].

# Diagrama del receptor FM monofónico



# Receptor FM monofónico

The screenshot shows a GNU Radio block diagram and its properties window for an "osmocom Source" module.

**Properties: osmocom Source**

- General Tab:**
  - ID:** osmosdr\_source\_0
  - Output Type:** Complex float32
  - Sync:** don't sync
  - Num Mboards:** 1
  - Mb0: Clock Source:** Default
  - Mb0: Time Source:** Default
  - Num Channels:** 1
  - Sample Rate (sps):** 2e6
  - Ch0: Frequency (Hz):** 103.3e6
  - Ch0: Freq. Corr. (ppm):** 0
- Advanced Tab:** Not visible in the screenshot.
- Documentation Tab:** Not visible in the screenshot.

**Block Diagram:**

```

graph LR
    Osmocom[osmocom Source] --> Multiplier[Multiply]
    Multiplier --> AudioSink[Audio Sink]
    
```

A callout box points to the "Ch0: Frequency (Hz)" field in the properties window, containing the value "103.3e6". Another callout box points to the "Variable" block in the top left, which has "Value: 2M". A third callout box points to the "WX GUI Slider" in the bottom left, which also has "Value: 2M".

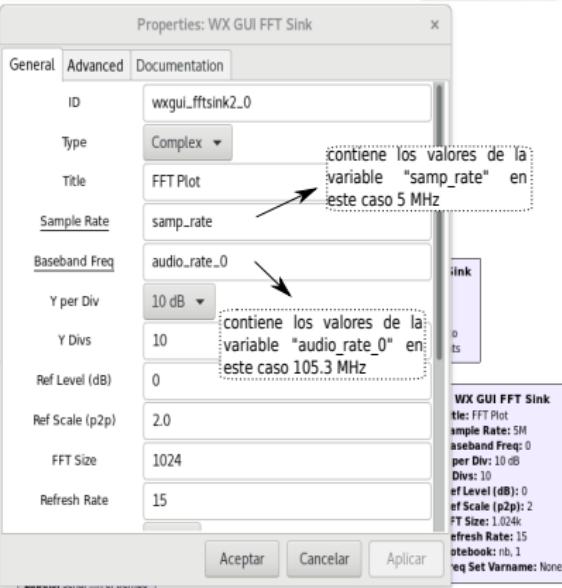
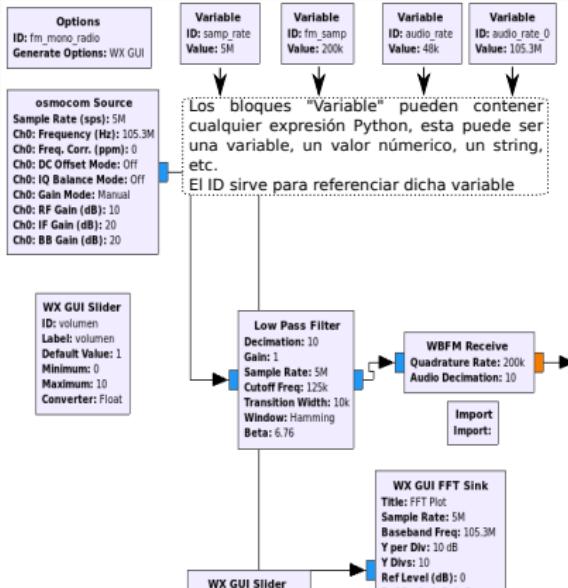
**WX GUI Slider**  
**ID:** volumen  
**Label:** volumen  
**Default Value:** 1  
**Minimum:** 0  
**Maximum:** 10  
**Converter:** Float

**Microondas/audio.grc**  
 >> Done  
 padding: "/home/daniel/receptormono"  
 >> Done

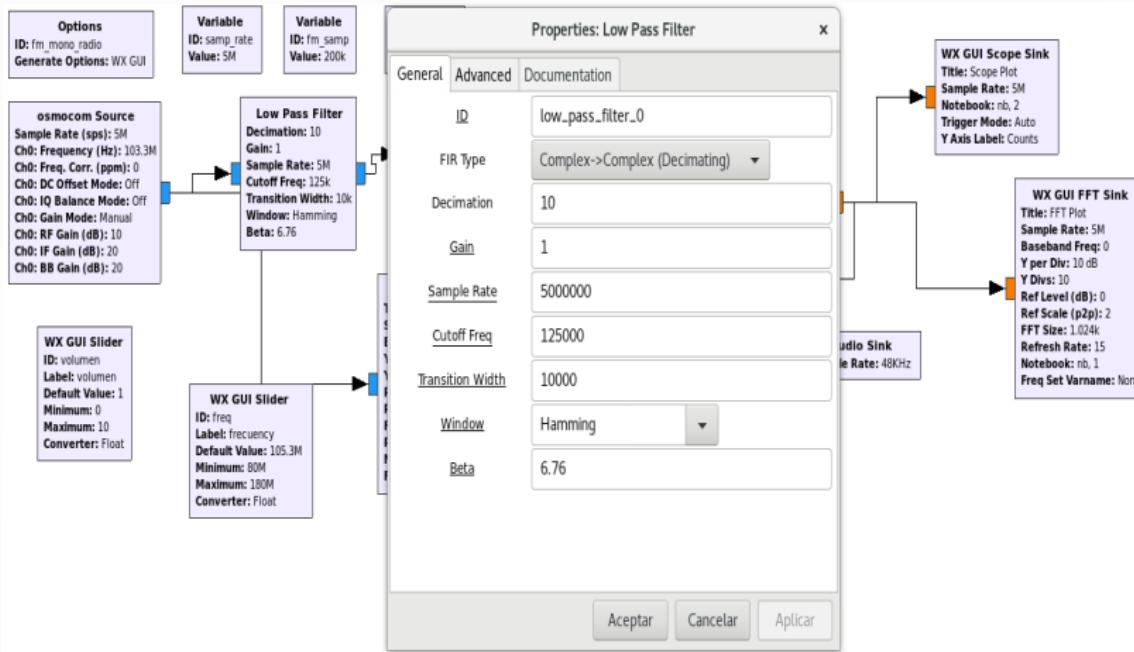
**se agrega la frecuencia ingresada en el bloque "variable" ID: samp\_rate 2 MHz**

**Se sintoniza la frecuencia definida, en este caso 103.3 Mhz**

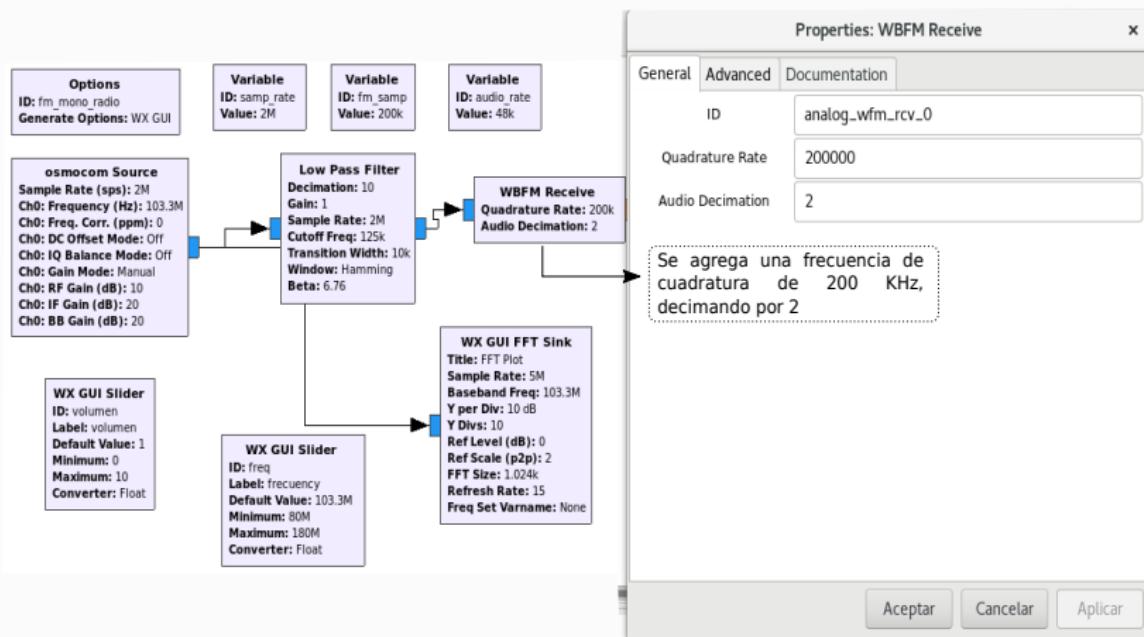
# Receptor FM monofónico



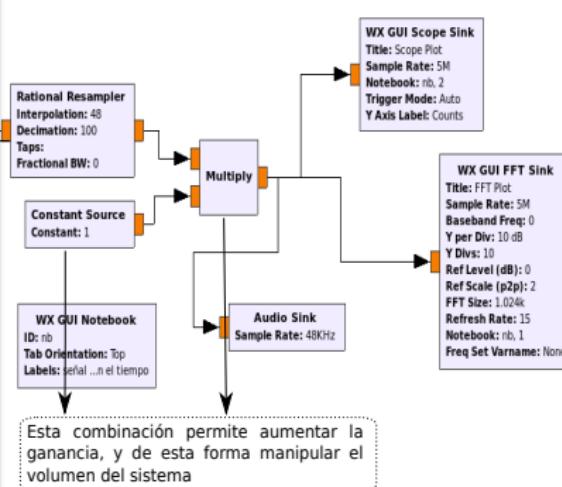
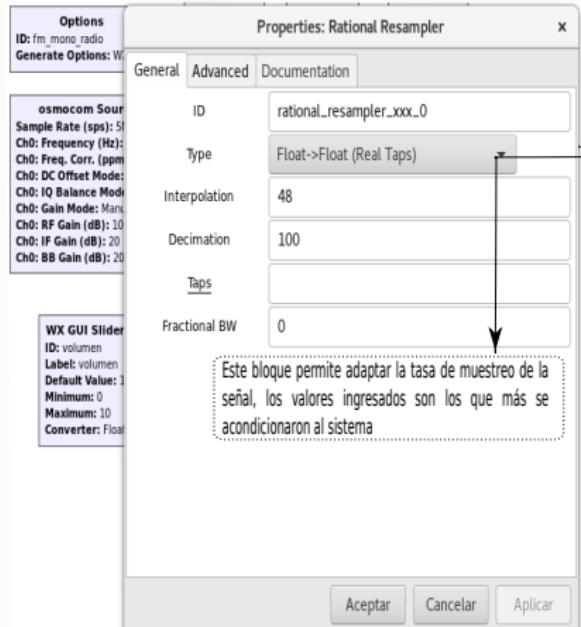
# Receptor FM monofónico



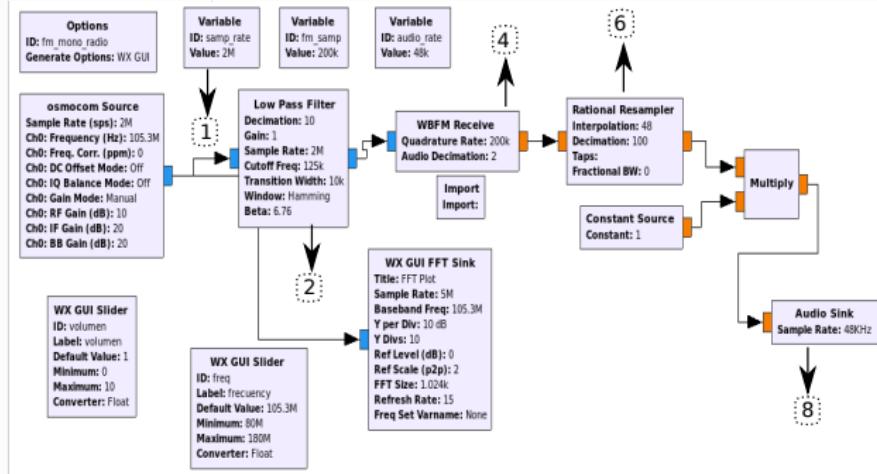
# Receptor FM monofónico



# Receptor FM monofónico



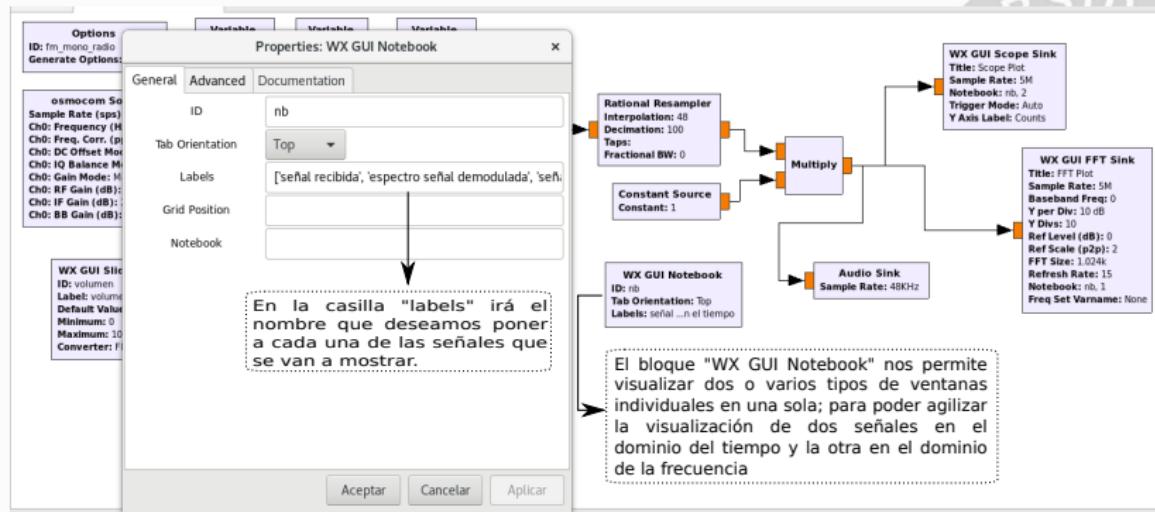
# Proceso matemático



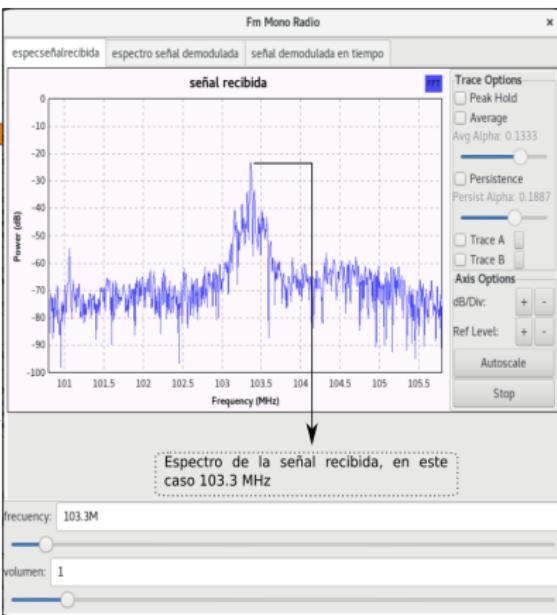
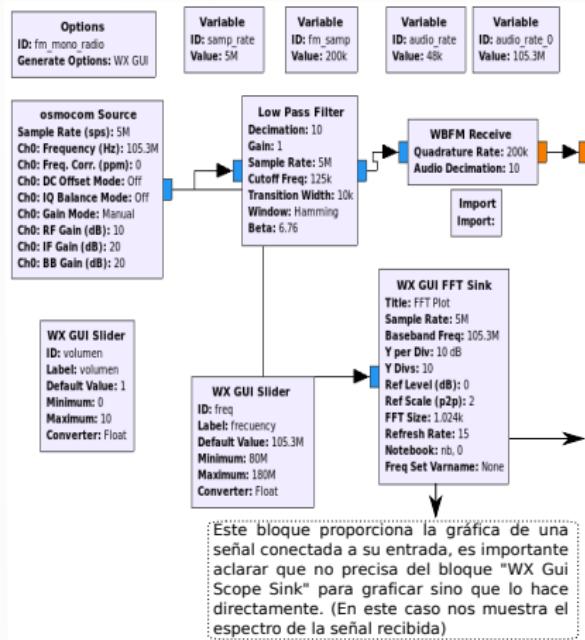
## PROCESO MATEMÁTICO

1. se ingresa una frecuencia de muestreo de 2 MHz
2. en el filtro pasa bajo se usa una decimación de 10
3. el resultado de  $(2 \text{ MHz})/(10) = 200 \text{ KHz}$
4. en el bloque WBFM Receive, se agrega el resultado anterior; frecuencia de cuadratura de 200 KHz, decimado en 2.
5.  $(200 \text{ KHz})/(2) = 100 \text{ KHz}$
6. en el bloque Rational Resample ingresamos una decimación de 100 y una interpolación de 48
7. a los 100 KHz del paso #5 se le decima los 100 del paso #6 y el resultado es 1 KHz.
8. por último se multiplica el 48 del paso #6 por el resultado del paso #7 (1 KHz) el resultado final es 48 KHz

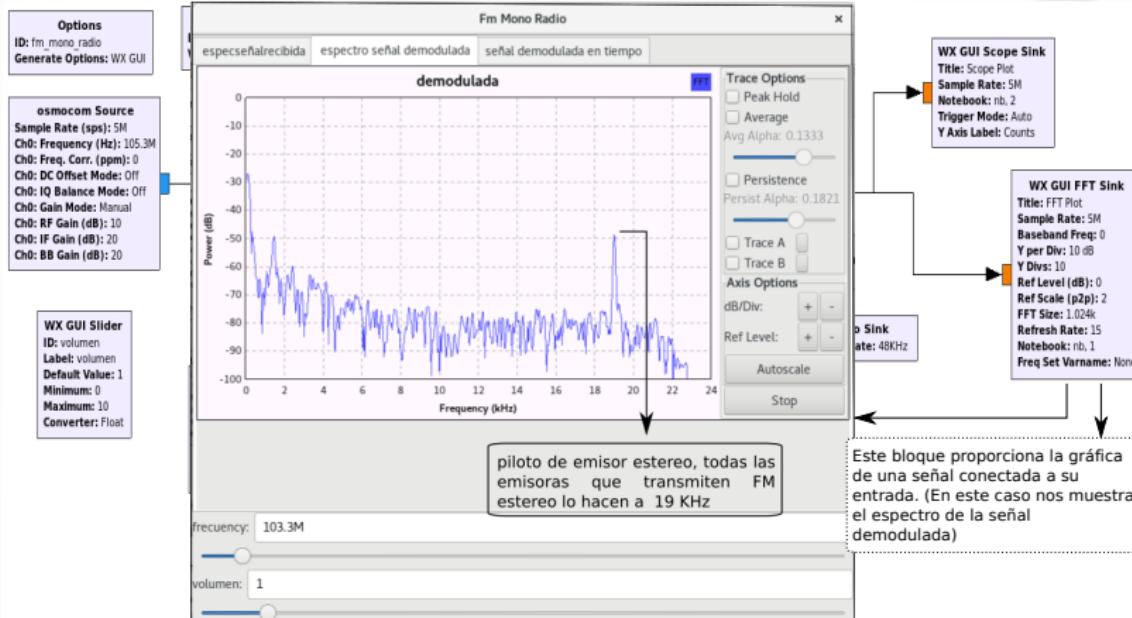
# Receptor FM monofónico



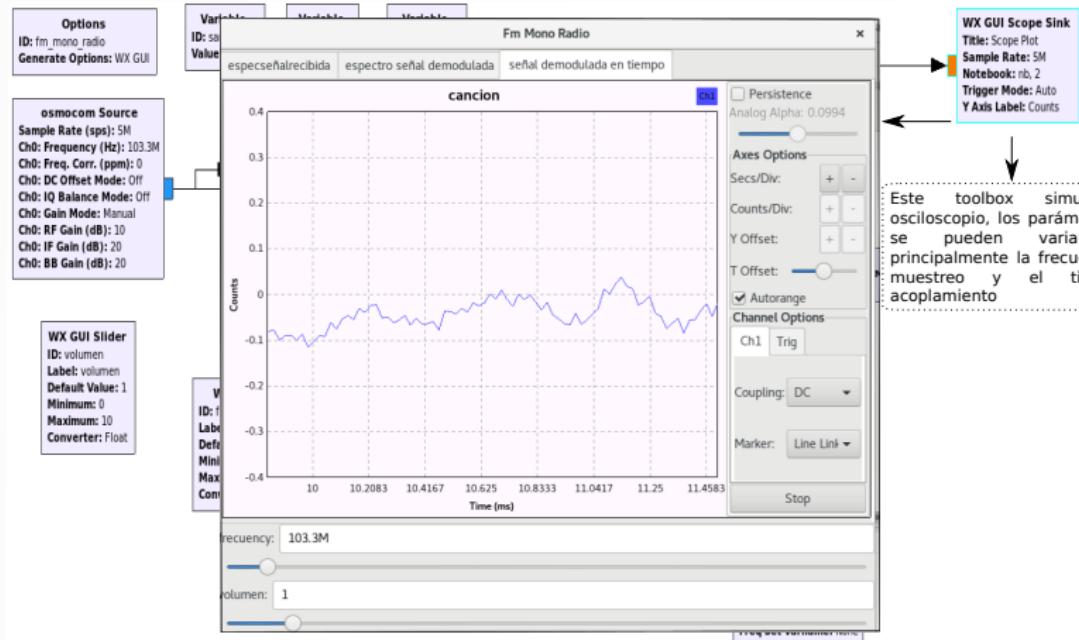
# Espectro señal recibida



# Espectro señal demodulada



# Señal demodulada en el tiempo



# Lab9

## Transmisor FM monofónico

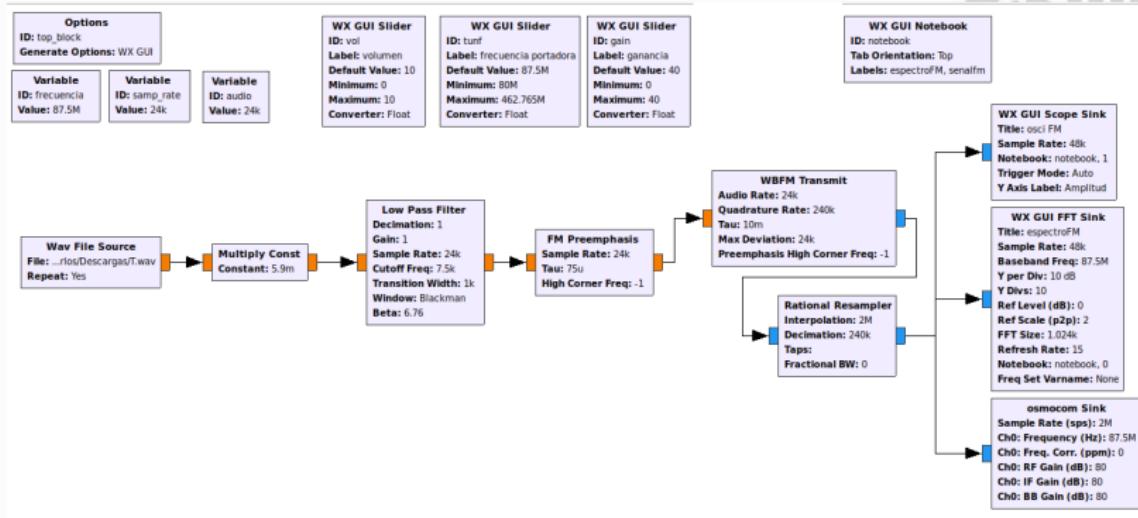
# Transmisor FM monofónico

El transmisor FM monofónico es un dispositivo electrónico que, mediante una antena, irradia ondas electromagnéticas que contienen (o pueden contener) información, como ocurre en el caso de las señales de radio.

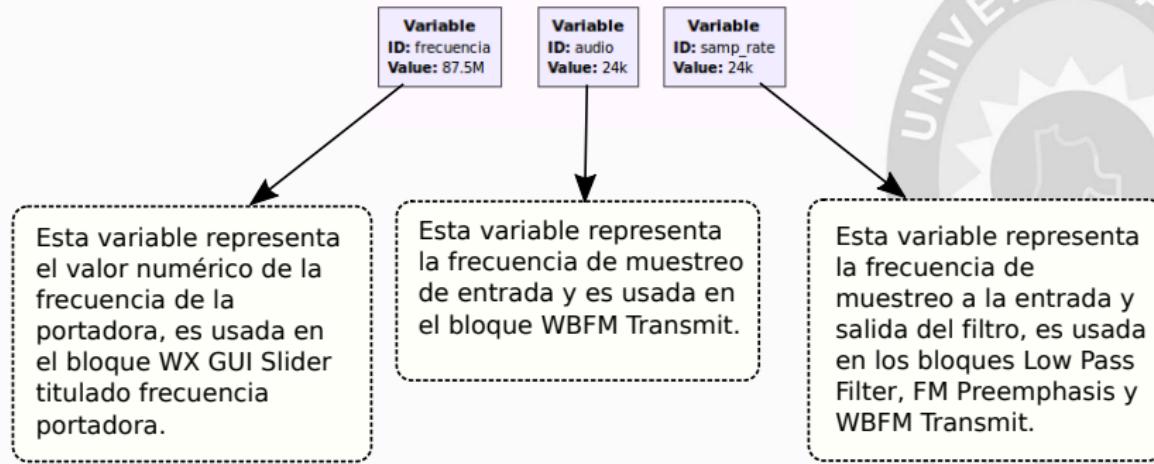
Por medio de la modulación angular se puede hacer que un parámetro de la onda portadora cambie de valor según las variaciones de la señal moduladora, que es la información que queremos transmitir.

Se utiliza esta modulación porque facilita la propagación de la señal de información, ordena el radio espectro, disminuye dimensiones de antenas, optimiza el ancho de banda de cada canal, evita interferencias entre canales y protege la información de las degradaciones por ruido.

# Transmisor FM monofónico



# Transmisor FM monofónico



# Transmisor FM monofónico

**WX GUI Slider**  
**ID:** vol  
**Label:** volumen  
**Default Value:** 10  
**Minimum:** 0  
**Maximum:** 10  
**Converter:** Float

**WX GUI Slider**  
**ID:** tunf  
**Label:** frecuencia portadora  
**Default Value:** 87.5M  
**Minimum:** 80M  
**Maximum:** 462.765M  
**Converter:** Float

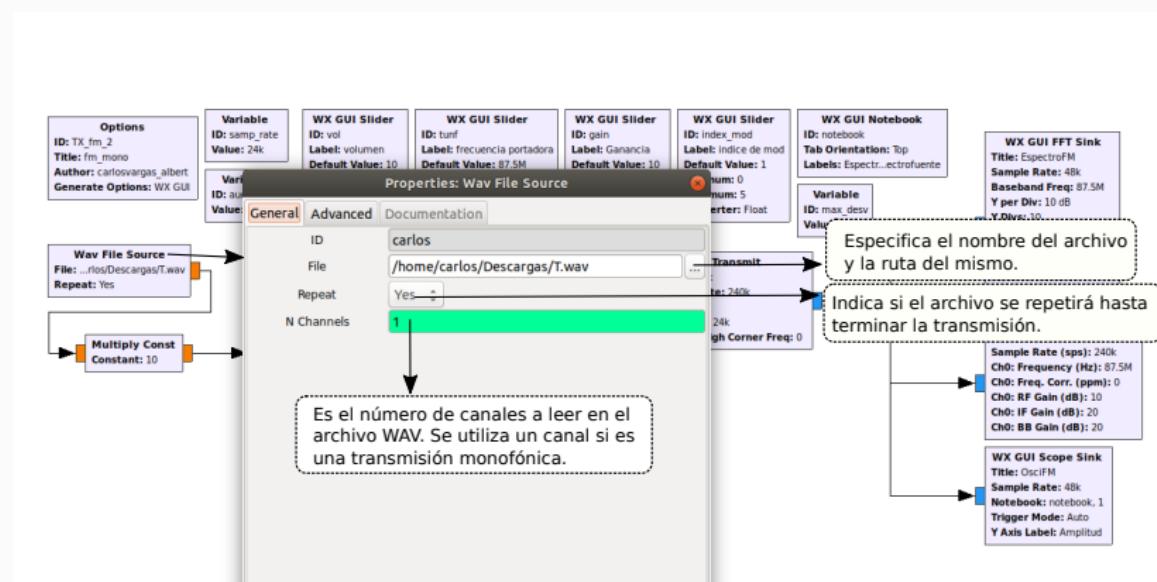
**WX GUI Slider**  
**ID:** gain  
**Label:** ganancia  
**Default Value:** 40  
**Minimum:** 0  
**Maximum:** 40  
**Converter:** Float

Representa numéricamente la variable volumen y sus valores están entre 0 y máximo de 10, es usada en el bloque Multiply Const.

Representa el valor de la frecuencia de la portadora y es usada en el bloque Osmocom sink.

Representa el valor de la ganancia que se le dará a la señal en el bloque Osmocom sink.

# Transmisor FM monofónico



En este transmisor FM monofónico la fuente es un archivo de audio en formato WAV muestreado a 24 kHz.

# Transmisor FM monofónico

Mediante el uso de la orden `file` en la consola, se puede consultar la frecuencia de muestreo de un archivo de audio de la siguiente forma (en este caso formato WAV).

- Ingresando a la consola y mediante la opción `cd` buscar la carpeta donde está ubicado el archivo de audio.
- Una vez ubicado en la carpeta desde el terminal con la opción `ls` se pueden ver los nombres de archivos que se alojan en esa ubicación, de esta forma se verá el nombre del archivo de audio.
- Con el nombre del archivo de audio, usar `file` de la siguiente manera:

```
box@user~$ file nombre_del_archivo.wav
```

## Transmisor FM monofónico

Si la frecuencia de muestreo de un archivo de audio se desea cambiar es posible hacerlo mediante el uso de la orden `sox` así:

- Con la ubicación y el nombre del archivo de audio de formato WAV usar `sox`.
- Como ejemplo, para un archivo que se llame “**audio1.wav**” y que esté muestreado a 44100 Hz y se requiera cambiar a 24000 Hz se usaría la orden así:

```
box@user~$ sox audio1.wav -r 24000 audio2.wav
```

- El nombre se cambia al final ya que lo que se hace es crear un nuevo archivo de audio con una frecuencia de muestreado diferente.

# Transmisor FM monofónico

**Properties: Low Pass Filter**

**General**

ID	low_pass_filter_0
FIR Type	Float->Float (Decimating)
Decimation	1
Gain	1
Sample Rate	24000
Cutoff Freq	7500
Transition Width	1000
Window	Blackman
Beta	6.76

**Documentation**

Modifica la frecuencia de muestreo de la señal. Es 1 porque en realidad la frecuencia de muestreo no cambia.

Es la ganancia del filtro . Es 1 porque la señal tiene la misma amplitud de entrada y salida del filtro.

Es la banda de transición del filtro, este parámetro se fijó a el menor valor permitido por el bloque siendo el valor de 1kHz.

La ventana es de tipo Blackman, la cual presenta una buena atenuación de los lóbulos laterales.

**WX GUI FFT Sink**

- Title: EspectroFM
- Sample Rate: 48k
- Baseband Freq: 87.5M
- Y per Div: 10 dB

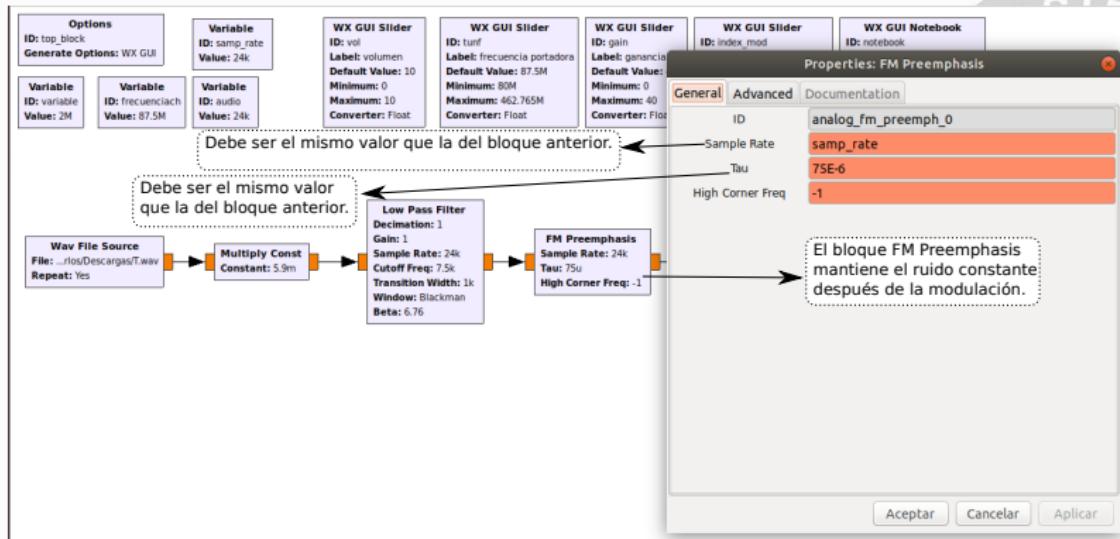
**WX GUI Scope Sink**

- Ch0: Frequency (Hz): 87.5M
- Ch0: Freq. Corr. (ppm): 0
- Ch0: RF Gain (dB): 10
- Ch0: IF Gain (dB): 20
- Ch0: BB Gain (dB): 20

**Frecuencia de muestreo.**

Es la frecuencia de corte, es el ancho de banda máximo de la señal de audio establecido para FM.

# Transmisor FM monofónico



# Transmisor FM monofónico

**WBFM Transmitter Properties:**

- ID: analog\_wfm\_tx
- Audio Rate: audio
- Quadrature Rate: samp\_rate\*10
- Tau: 0.01
- Max Deviation: 24e3
- Preemphasis High Corner: -1.0

**Wav File Source Options:**

- ID: top\_block
- Generate Options: WX GUI
- Variable ID: variable
- Value: 2M

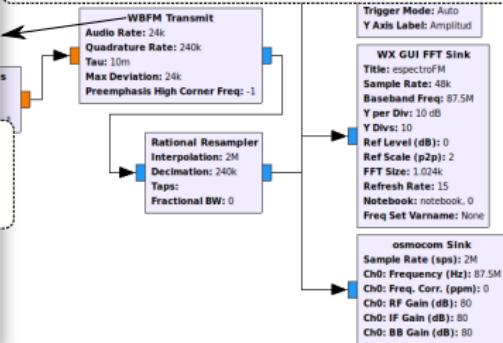
**WX GUI Slider Options:**

- ID: freqencia
- Value: 87.5M

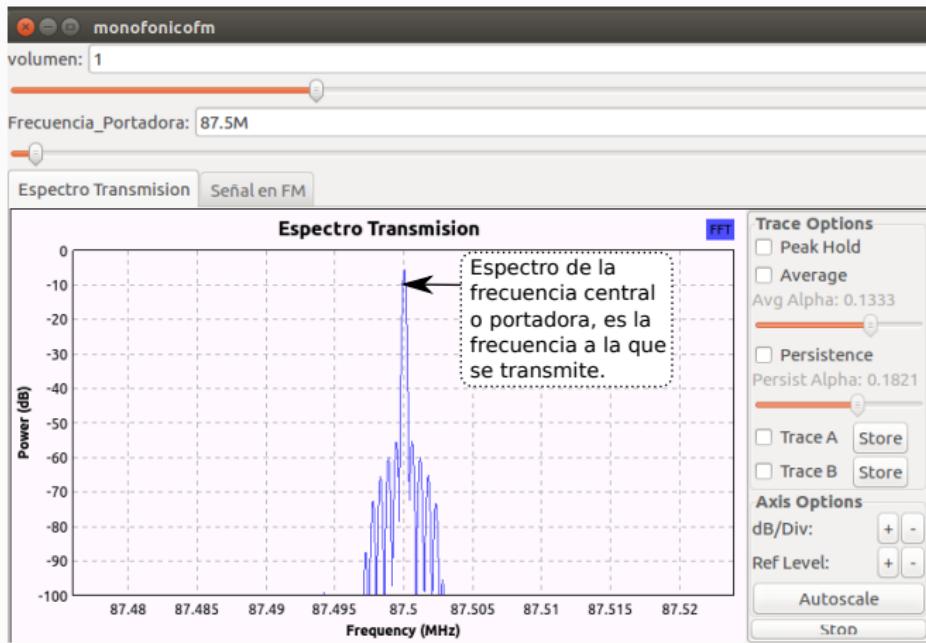
**Buttons:**

- Aceptar
- Cancelar
- Aplicar

Permite que dos señales provenientes de dos fuentes independientes, pero con características de ancho de banda similares, ocupen el mismo ancho de banda de transmisión y se puedan separar en el extremo receptor. Así, si dos señales I y Q, modulan dos señales portadoras de la misma frecuencia, cada señal está desfasada 90° una respecto a la otra.



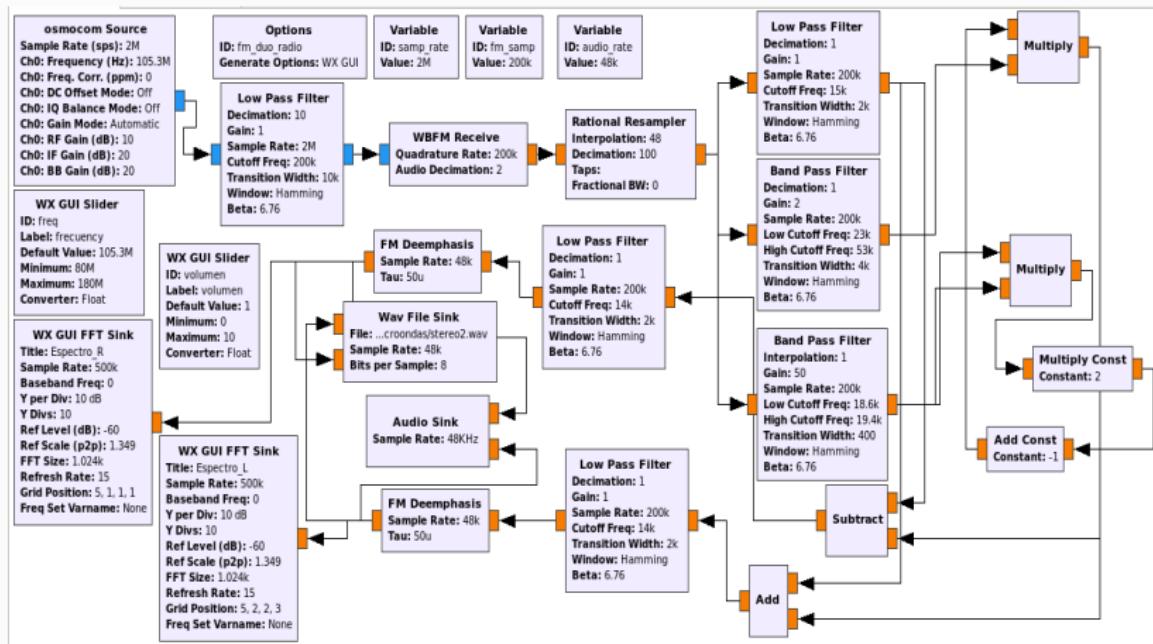
# Transmisor FM monofónico



# Lab10

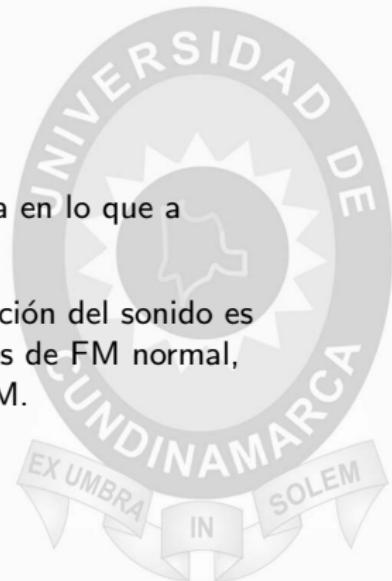
## Receptor FM estereofónico

# Receptor FM estereofónico

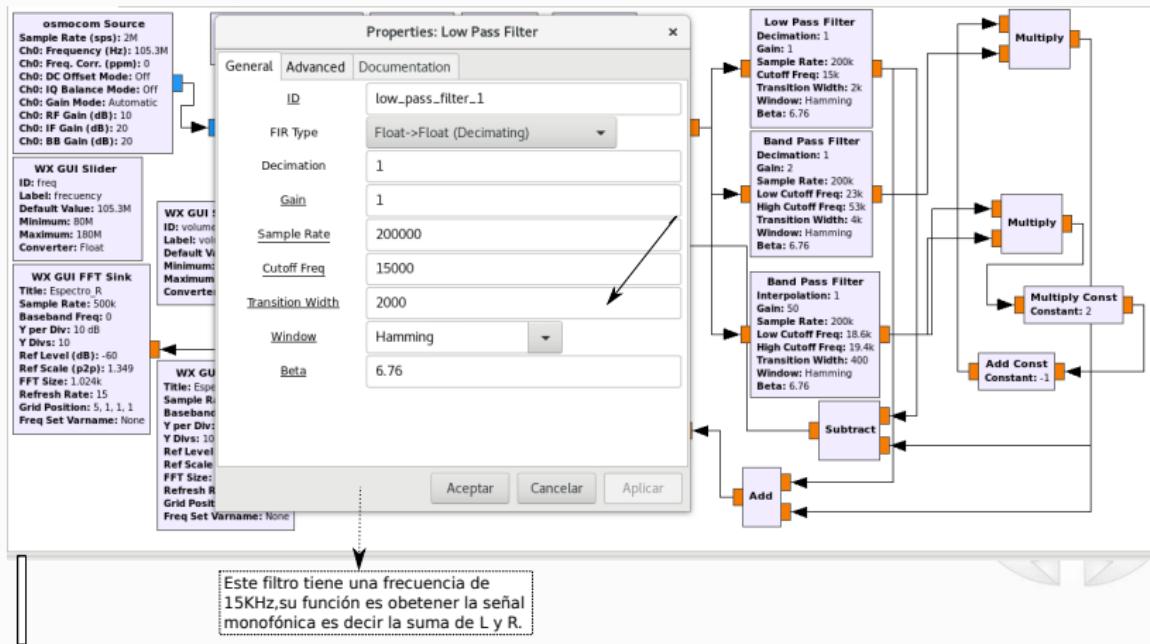


## Receptor FM estereofónico

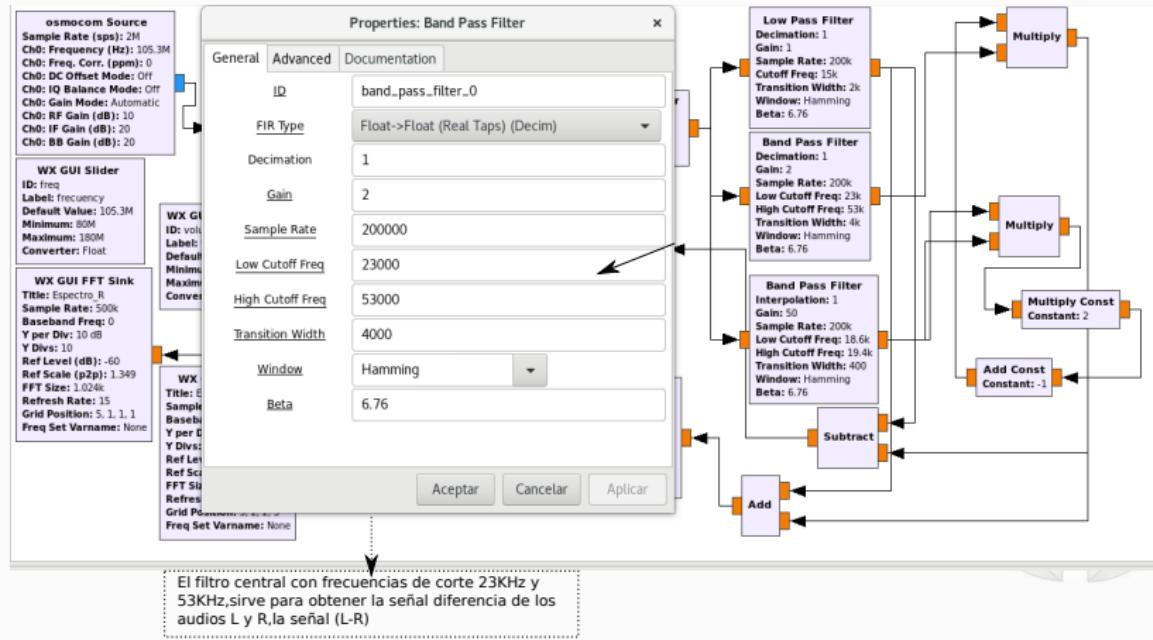
- La FM, vino a cambiar en todo el esquema que existía en lo que a transmisión y recepción de radio se refiere.
- Una de las ventajas de FM estéreo es que la reproducción del sonido es tan buena en los receptores estereofónicos como en los de FM normal, estos lo reproducen como una señal monofónica de FM.



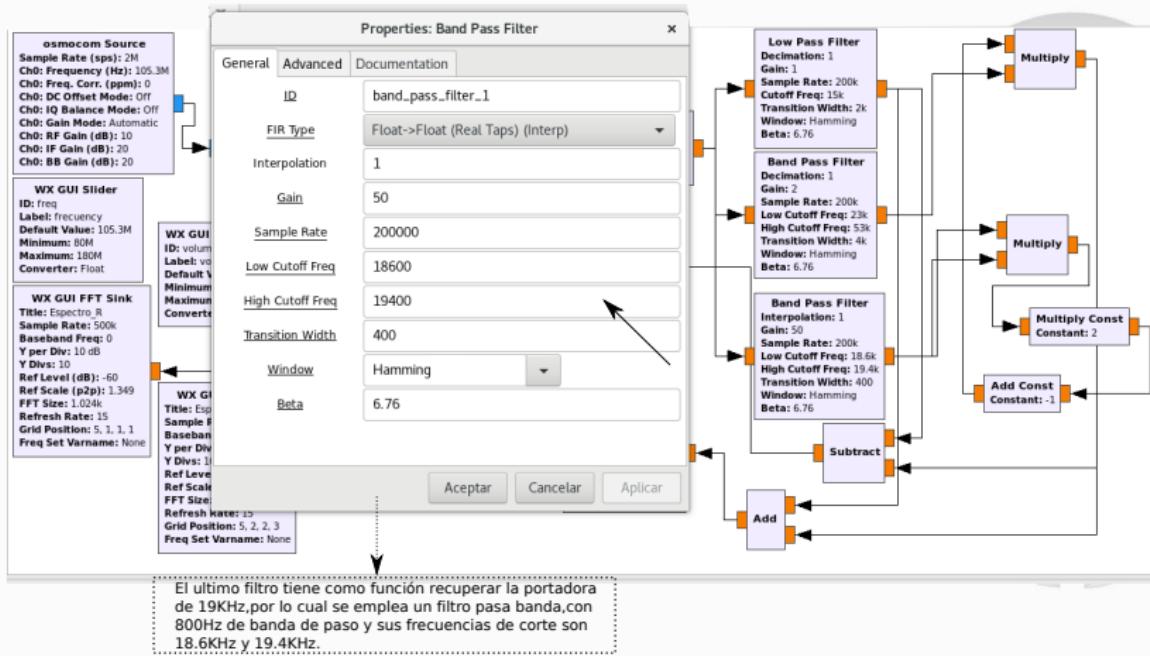
# Receptor FM estereofónico



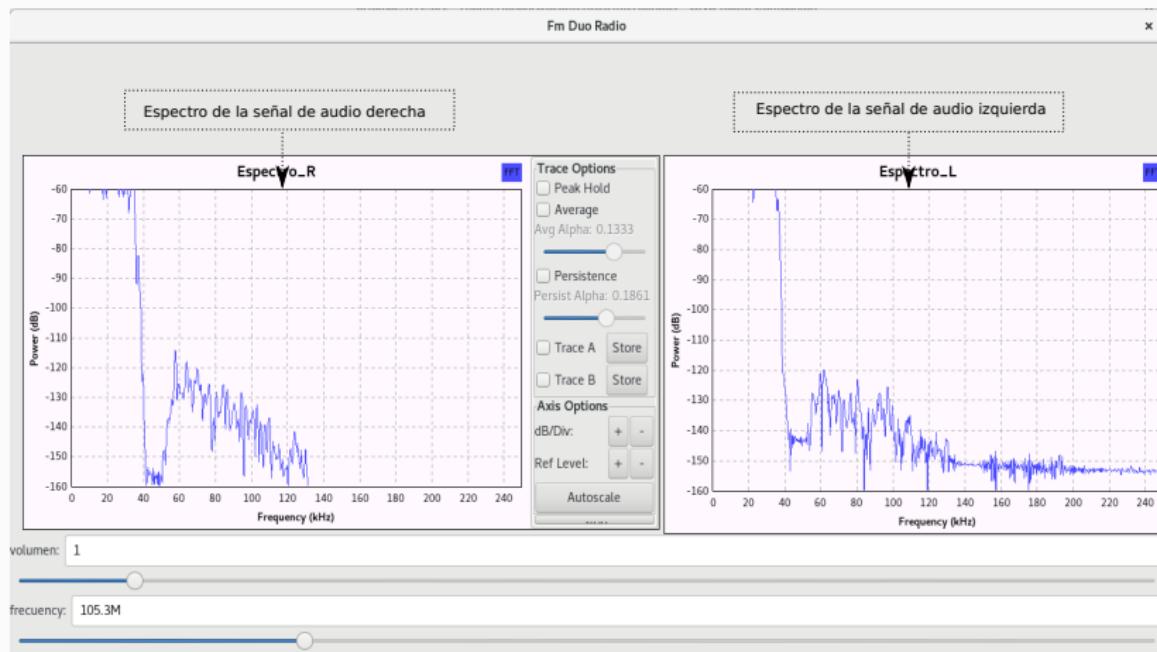
# Receptor FM estereofónico



# Receptor FM estereofónico



# Receptor FM estereofónico

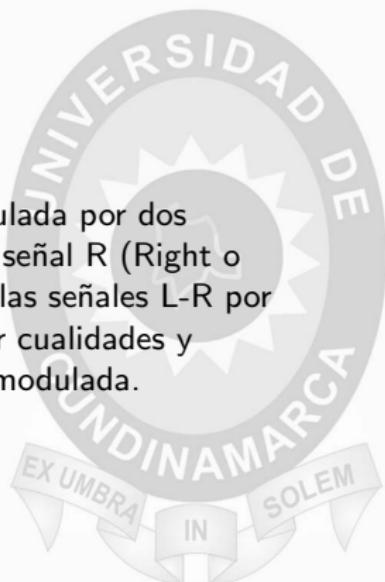


# Lab11

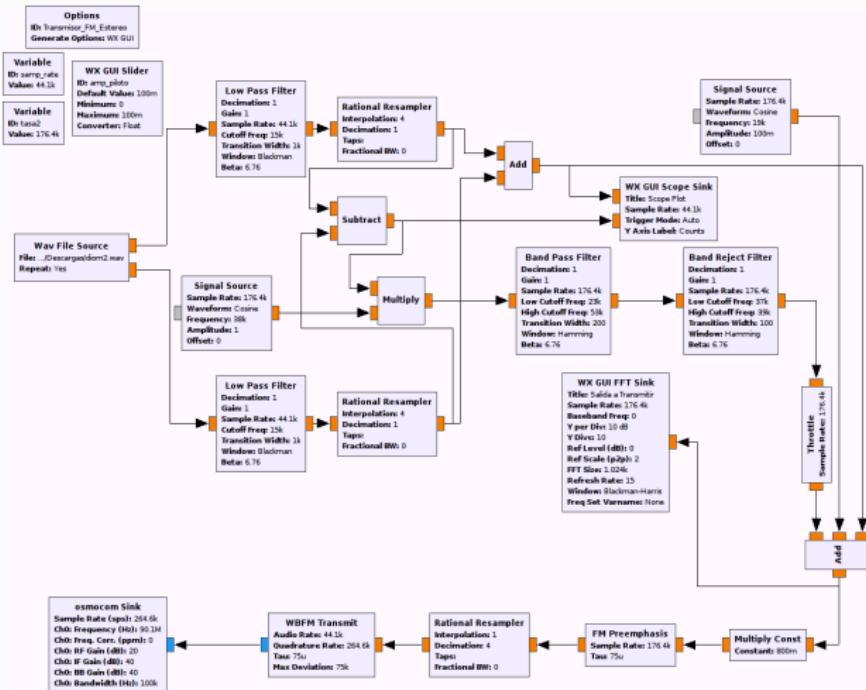
## Transmisor FM estereofónico

# Transmisor FM estereofónica

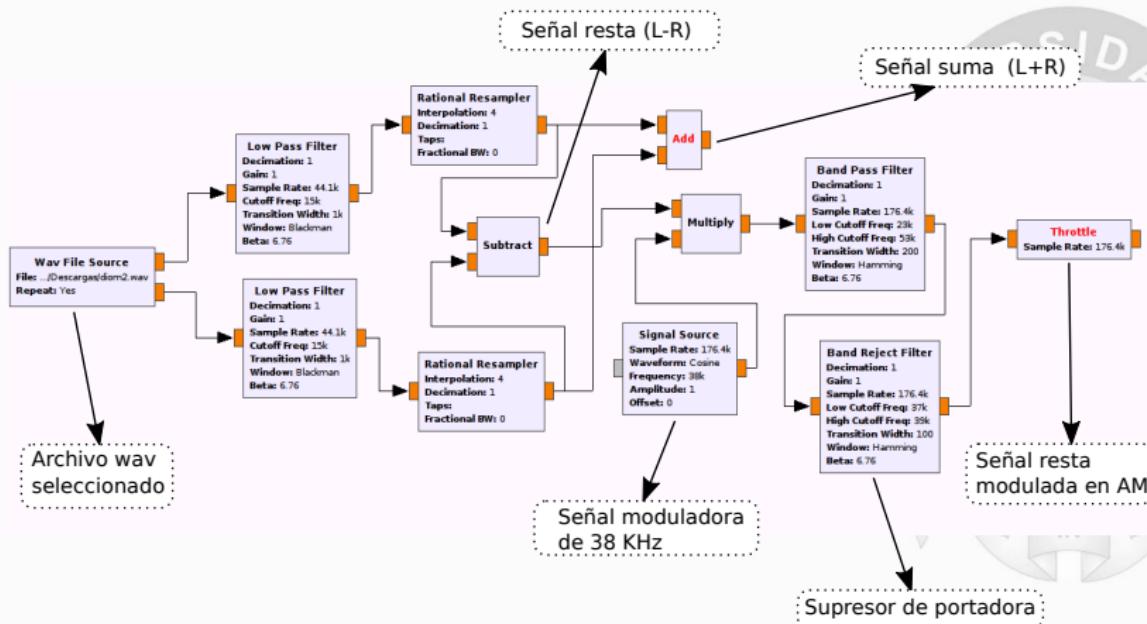
La transmisión FM estéreo es la emisión en frecuencia modulada por dos canales. Para emitir en estéreo no se emite directamente la señal R (Right o Derecha) y la señal L (Left o Izquierda) sino que se emiten las señales L-R por un canal y L+R por el otro. Esto se realiza con el fin de dar cualidades y calidades de sonido estéreo a la transmisión por frecuencia modulada.



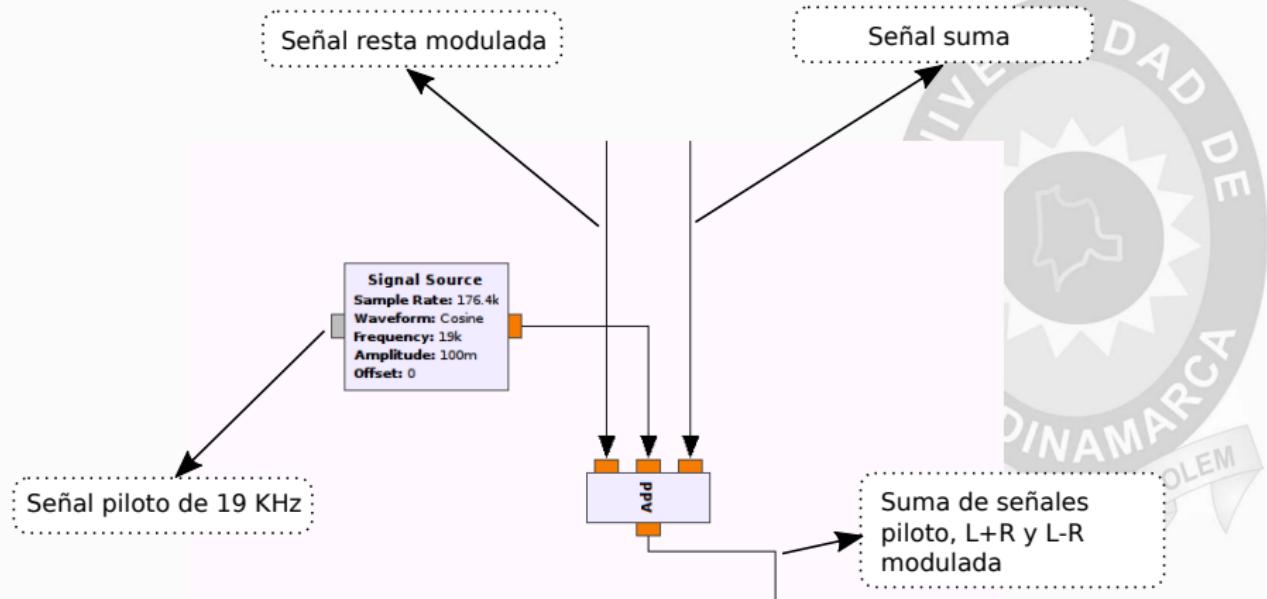
# Transmisor FM estereofónico



# Transmisor FM estereofónico



# Transmisor FM estereofónico

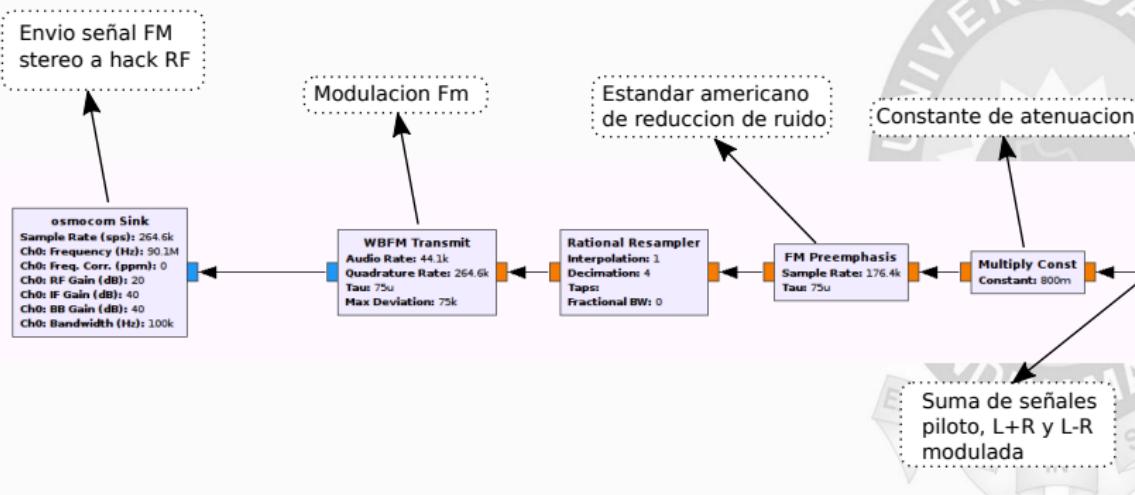


## Señal piloto de 19 KHz

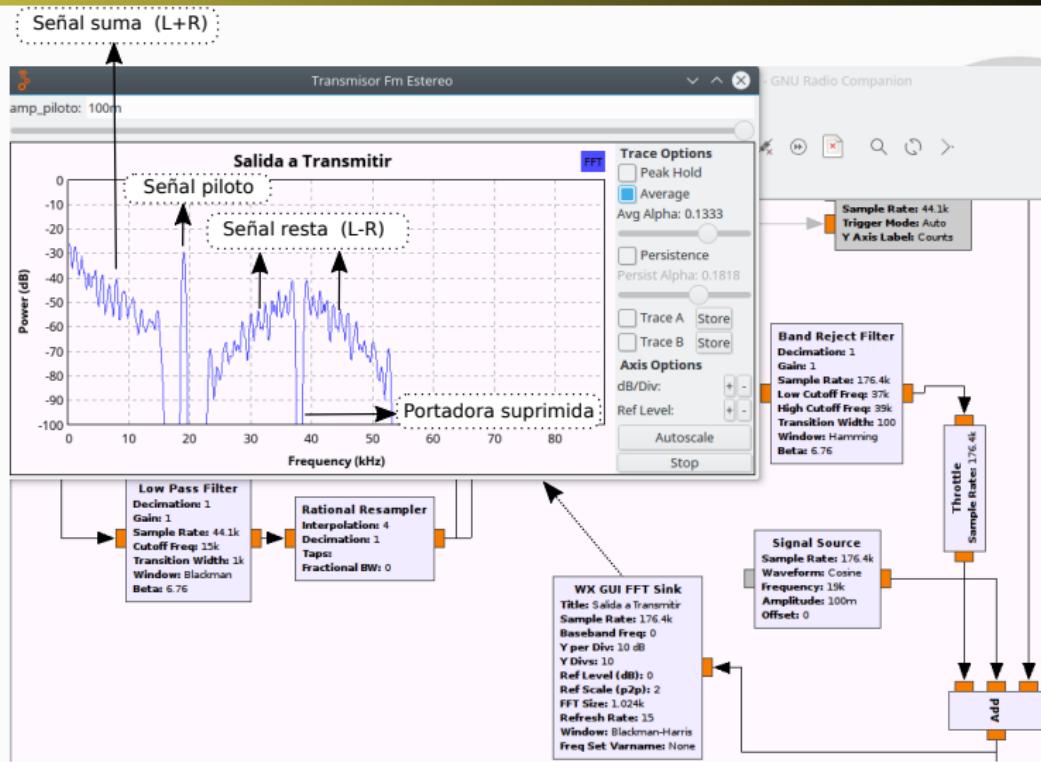
El piloto estéreo es un tono de 19kHz que tiene la misma fase que la portadora de la Señal Resta (que hemos eliminado previamente), y una amplitud de (normalmente) el 10% de la amplitud total de la señal, en caso de no detectarse el piloto es conveniente aumentar la amplitud del mismo hasta ser detectada por el receptor. El piloto estéreo tiene tres funciones principales:

- Informa al receptor de que la emisión es estéreo
- Permite regenerar la subportadora de la Señal Resta a 38kHz que no hemos emitido gracias a modular en DSBSC.
- Permite regenerar la subportadora del RDS a 57kHz que no hemos emitido gracias a modular en DSBSC.

# Transmisor FM estereofónico



# Transmisor FM estereofónico



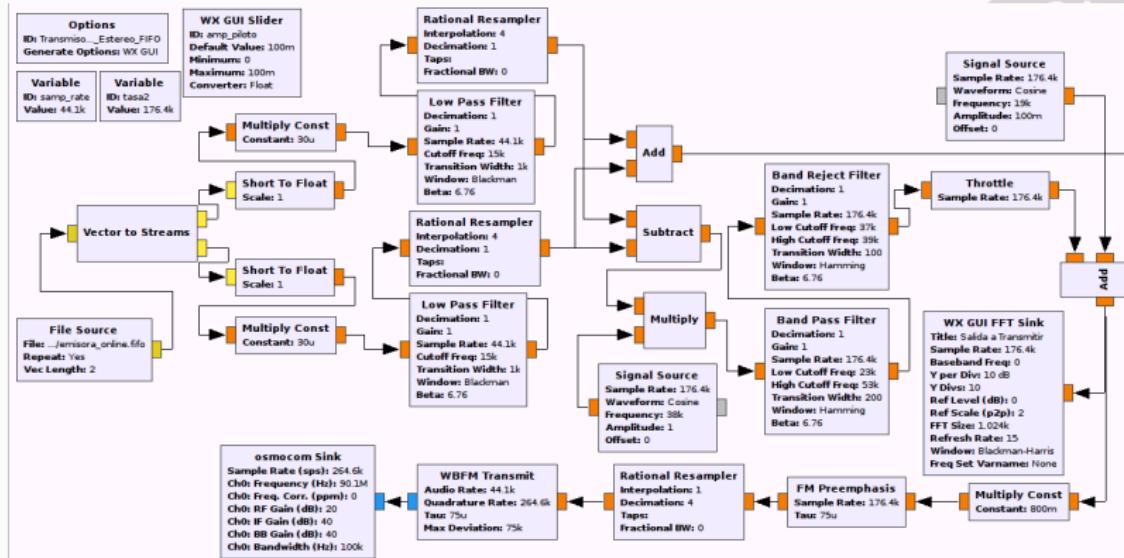
# Lab12

## Retransmisión emisora mp3 online en FM estéreo

# Introducción

Se realizará la retransmisión de una emisora online utilizando el transmisor de FM estéreo construido en la práctica anterior; se aprovechará la característica de un archivo “ducto” o “tubería”, el cual nos permite que procesos separados se comuniquen sin haber sido diseñados para funcionar juntos. Gracias a la herramienta mpg123 que lee uno o más archivos de audio o en este caso una URL que contiene un flujo de bits de audio en formato MPEG, es posible alimentar un archivo FIFO que se comporta como tubería; es decir en la medida que ingresan los bits de información de la emisora online en la “tubería”, así mismo son extraídos por el transmisor de radio FM estéreo y son retransmitidos.[4].

# Esquema general



# Obtención URL emisora online

Como posibilidad de obtención de emisoras mp3 online, Ingrese a [www.internet-radio.com](http://www.internet-radio.com) y escoja la emisora de su preferencia. Estando allí reproduzcalo en una nueva ventana para obtener posteriormente su URL.

The screenshot shows the InternetRadio website interface. On the left, a sidebar lists radio stations: ReggaetonColombia (Kevin Roldan - Tranquilo), 123vallenato.com (http://www.123vallenato.com), and Colombia rumbera Stream (RenÃ© Grand - Sigue la gente [1CSV]). On the right, a main window displays the '123vallenato.com Radio Player - Google Chrome' with the URL <https://www.internet-radio.com/player/?mount=http://radiolatina.info:7087/listen.pls&title=123vallenato.com>. The player shows a play button, a progress bar at 05:14, and the station name 123vallenato.com. A callout box points to the URL in the address bar with the instruction: "Extraiga la URL de la emisora, recuerde que debe ser la dirección del servidor de audio y no la dirección de la página web del la emisora."

## Creación archivo FIFO

Existe un tipo de tubería que puede ser descrita como una tubería “con nombre”, también denominada FIFO que sus siglas traducen “Primero en entrar, primero en salir” y se refiere a la propiedad de que el orden de bytes entrante es el mismo que sale.

- Mediante el uso de consola procedemos a introducir el comando mkfifo, que nos permite crear el archivo FIFO.

```
mkfifo emisora_online fifo
```

- Mediante la herramienta mpg123 se procede a alimentar el archivo FIFO con una emisora online.

```
mpg123 -r44100 --stereo -s
```

```
http://radiolatina.info:7087>emisora_online fifo
```

## Creación archivo FIFO

La opción `--stereo` permite realizar una transmisión de tipo estéreo; utilice `-m` para realizar una transmisión de tipo monofónica.

Con la ayuda de la opción `-r` podemos definir la tasa de muestreo, por ejemplo `-r44100` convierte la frecuencia de muestreo a 44100 muestras por segundo.

Es importante el uso de la orden `-s`, ya que nos permite que las muestras de audio decodificadas se escriban en la salida estándar, es decir, en consola, en lugar de reproducirlas a través del dispositivo de audio; gracias a esta característica se puede llevar a cabo el concepto del uso de tuberías que suelen ser alimentadas a través de pantalla.

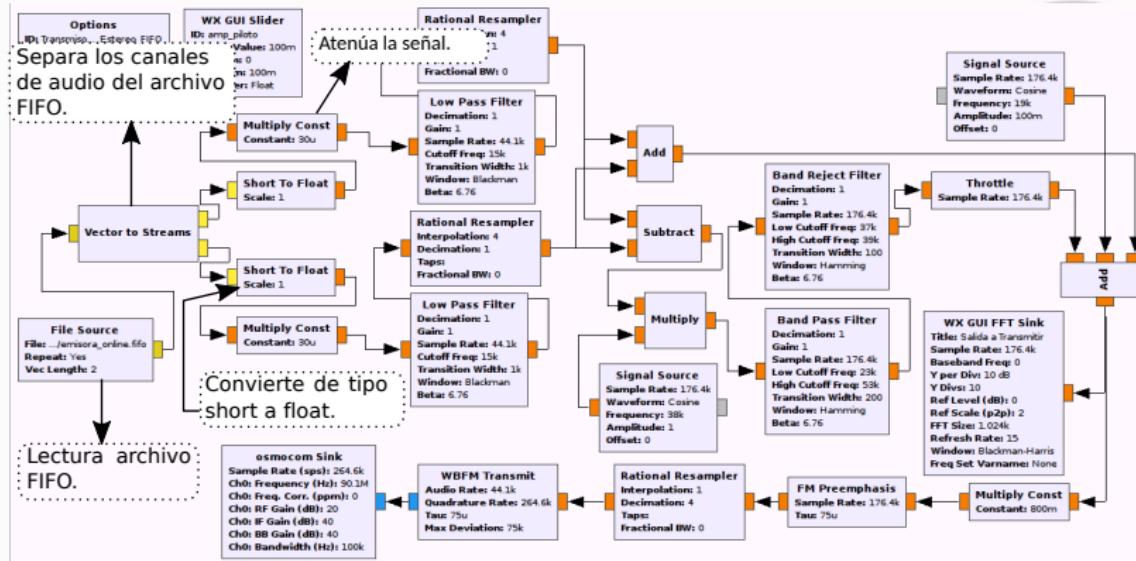
## Creación archivo FIFO

- Si se desea realizar una prueba previa a la transmisión, es posible escuchar la emisora de igual manera mediante la herramienta mpg123; recuerde eliminar la orden -s para que las muestras de audio no se muestren en pantalla y sean reproducidas a través del dispositivo de audio[5].

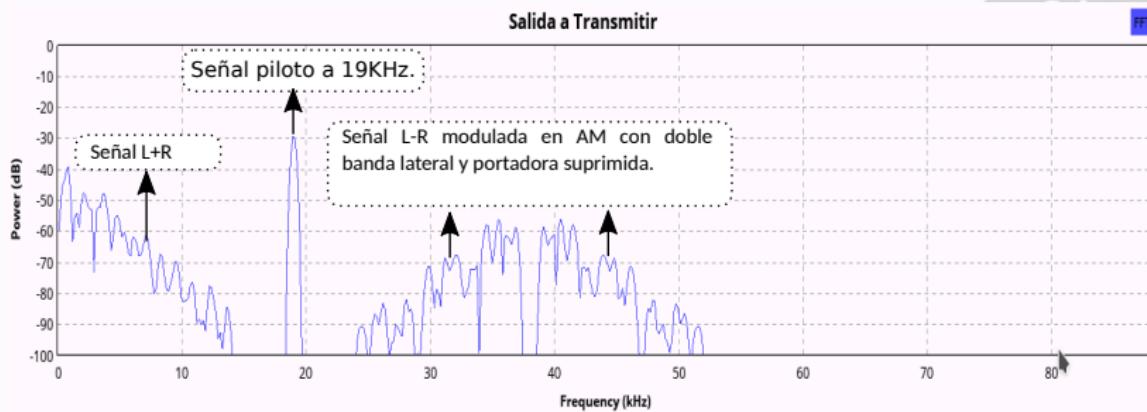
```
mpg123 -r44100 --stereo http://radiolatina.info:7087
```

- Recuerde que para ejecutar el proceso, en primer lugar debe alimentar el archivo FIFO mediante la consola ejecutando la herramienta mpg123; una vez se esté alimentando la tubería puede proceder a iniciar la aplicación del transmisor FM estéreo en GNU Radio. Tenga presente que al momento de detener la aplicación de GNU Radio se detendrá el proceso de la herramienta mpg123 en la consola.

# Explicación bloques extras



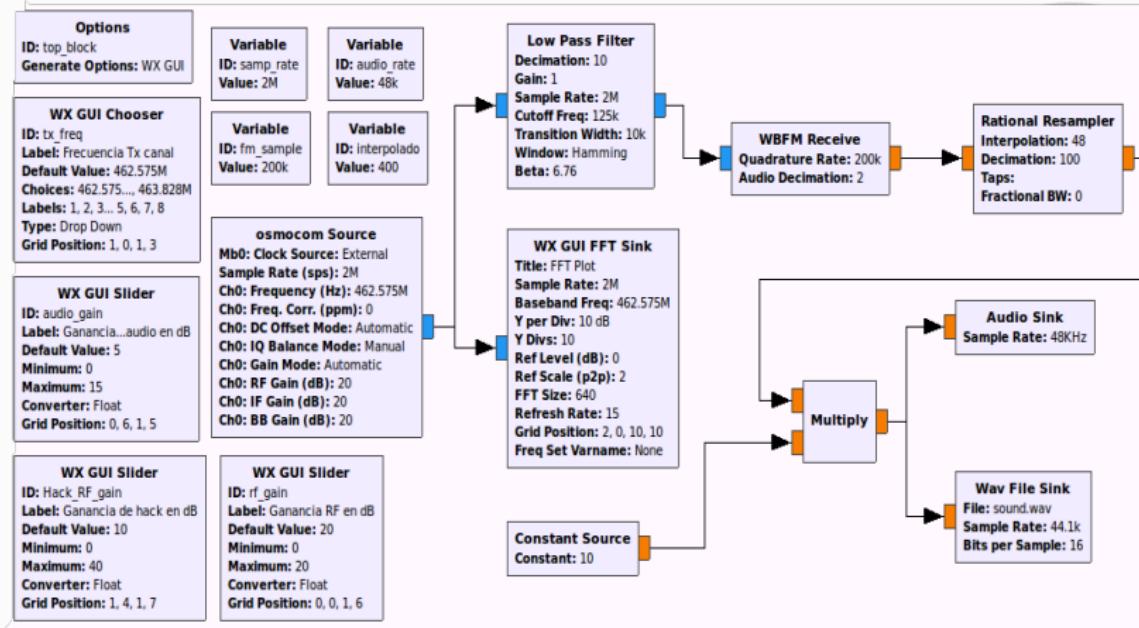
# Espectro de señal a transmitir



# Lab13

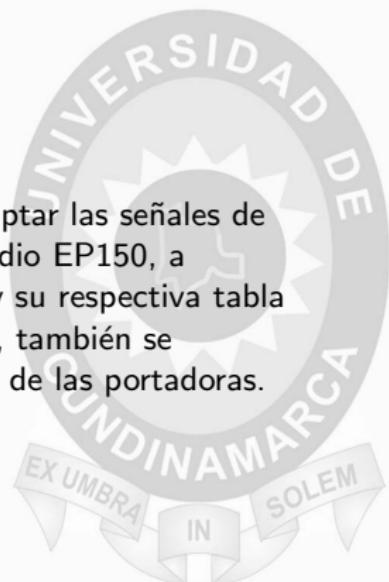
## Recepción de señales de radio FM para Walkie-Talkie UHF

# Recepción de señales de radio FM para Walkie-Talkie UHF



# Recepción de señales de radio FM para Walkie-Talkie UHF

Para esta práctica se realiza el diagrama de bloques para captar las señales de las frecuencias portadoras de cada uno de los canales del radio EP150, a continuación se muestra el radio utilizado en esta práctica y su respectiva tabla de especificaciones, estas son necesarias durante la práctica, también se presenta el diagrama de bloques realizado para la captación de las portadoras.



# Especificaciones del radio

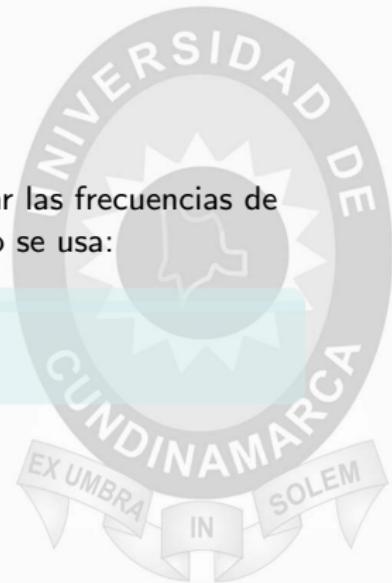
Estas especificaciones serán necesarias con el fin de conocer el rango de frecuencia de operación del radio[6].

Especificaciones generales		
	RU2081BKNBAA	PV2081BKNBAA
Rango de frecuencia	UHF (de 462 a 469 MHz)	VHF (de 151 a 161 MHz)
Salida de audio	2000 mW	
Capacidad de canales	8 canales	
Ancho de banda de canales	12.5/25 kHz	
Dimensiones (Al. x. Anch. x Prof.) cbatería ion-litio estándar	115.6 x 576 x 40.5 mm	
con batería ion-litio de alta capacidad	115.6 x 576 x 45.1 mm	
con batería ion-litio de capacidad ultra alta	115.6 x 576 x 45.1 mm	
Peso con batería ion-litio estándar	244 g	
con batería ion-litio de alta capacidad	293 g	
con batería ion-litio de capacidad ultra alta	293 g	
Duración promedio de la batería a 5/5/90 (con sensor de batería activado)		Hasta 12 horas
con batería ion-litio estándar de 1100 mAH		Hasta 24 horas
con batería ion-litio de alta capacidad de 2200 mAH		Hasta 26 horas
con batería ion-litio de capacidad ultra alta de 2400 mAH		Hasta 26 horas
con accesorio opcional de batería alcalina		
Voltaje de la fuente de alimentación	7.2 voltios de CC (paquete de baterías ion-litio o alcalinas)	
Designación de FCC	AZ489FT4879	AZ489FT3817
ANATEL / COFETEL	1694-07-0506	1693-07-0506

# GQRX

Es necesario instalar el programa gqrx, útil para determinar las frecuencias de trabajo del radio en cada uno de sus canales, para instalarlo se usa:

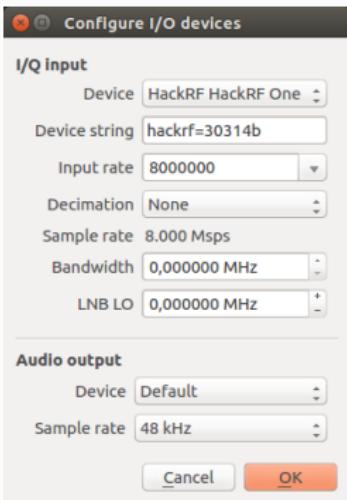
```
sudo apt-get update  
sudo apt-get install gqrx-sdr
```



# HackRF One

Al iniciar el programa se configura el dispositivo, en este caso una Hack RF. Para verificar que esté conectada al PC se puede usar la orden:

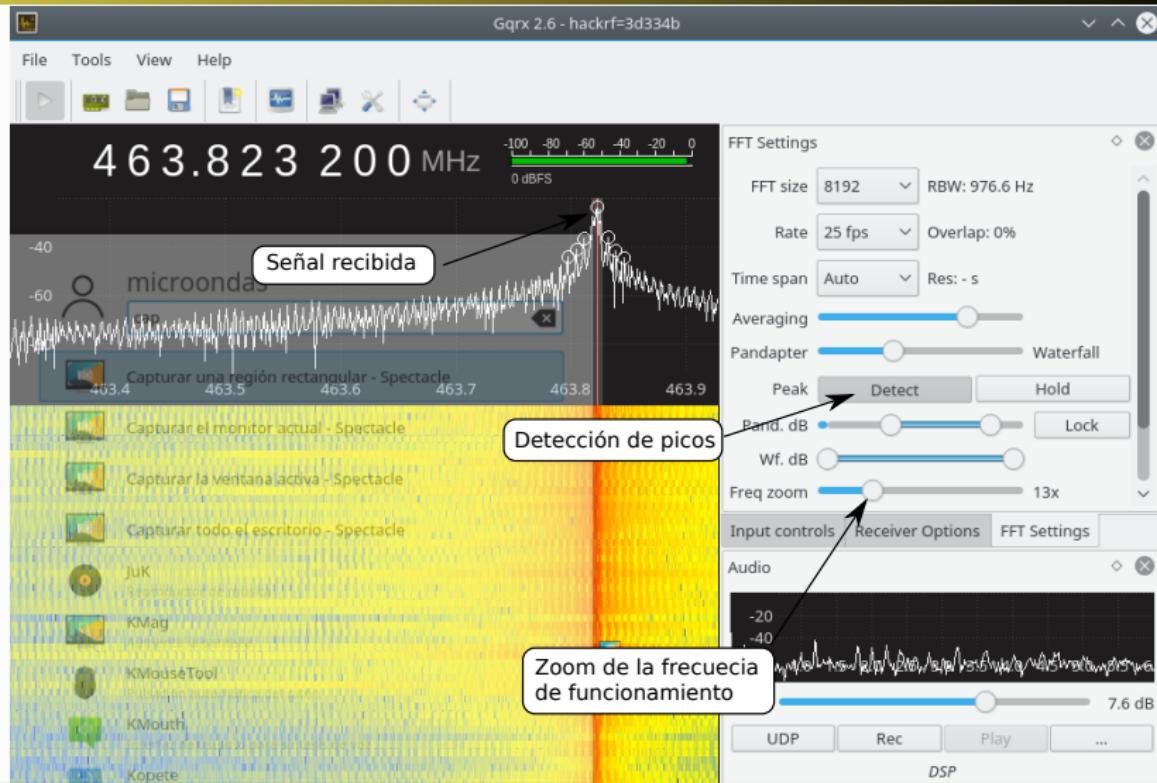
hackrf\_info



# Interfaz del programa



# Detectar frecuencias

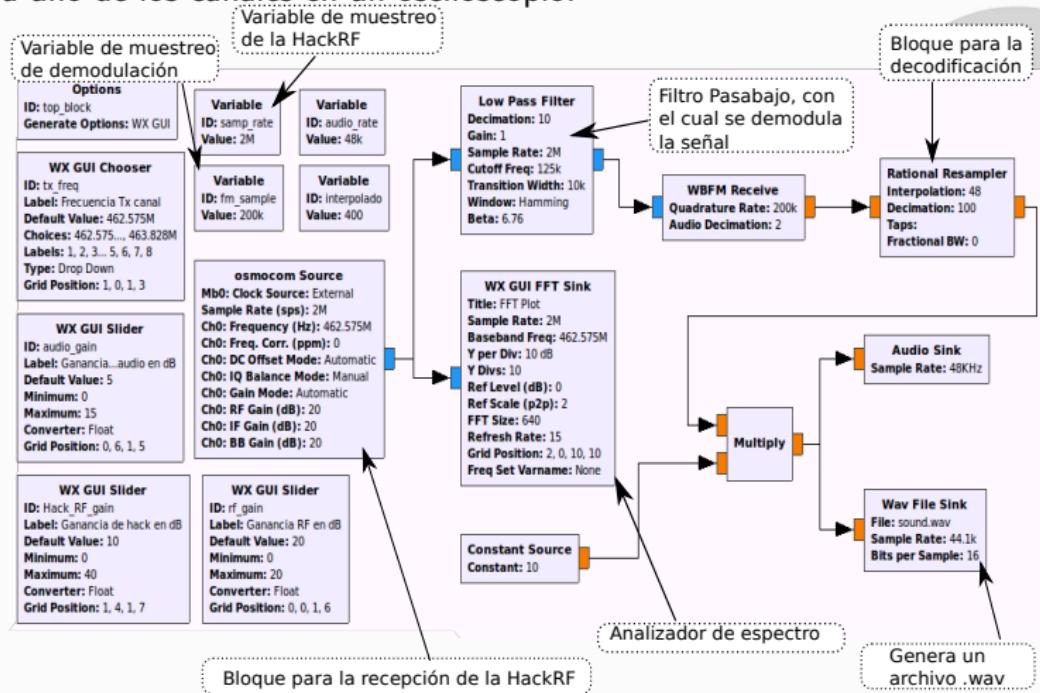


## Tabla de frecuencias

Al variar los 8 canales disponibles del dispositivo, se logran encontrar las frecuencias correspondientes.

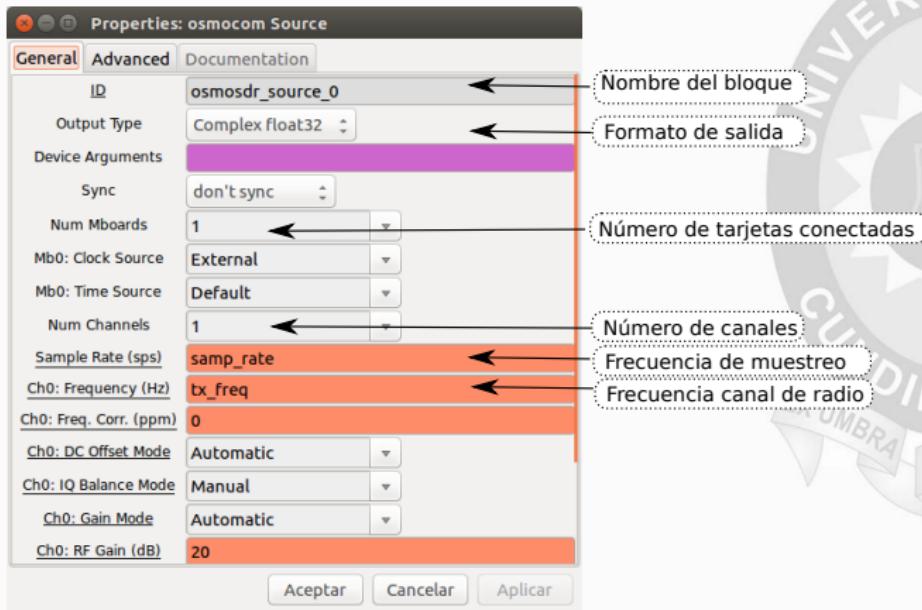
CANAL	FRECUENCIA DE SEÑAL PORTADORA
1	462,5750 MHz
2	462,6250 MHz
3	462,6750 MHz
4	463,5500 MHz
5	463,6250 MHz
6	463,7625 MHz
7	463,7750 MHz
8	463,8250 MHz

Obtenidos los rangos de frecuencias de cada uno de los canales del radio, se realiza el diagrama de bloques con el fin de observar las señales portadoras de cada uno de los canales en un osciloscopio.



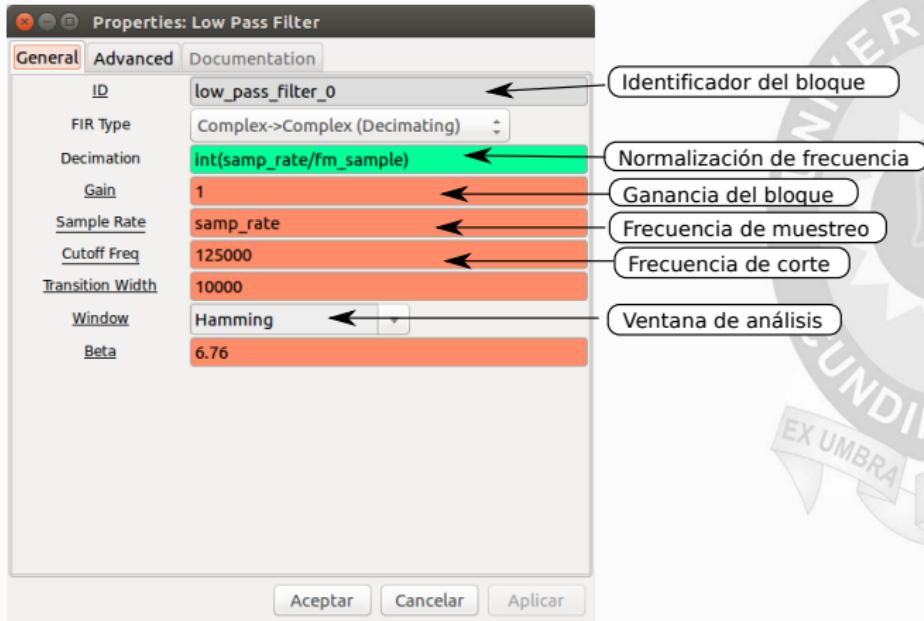
# Configuraciones

Se configura el bloque con el que se recibe la señal con la tarjeta HackRF donde nos permite configurar más de una tarjeta en la misma conexión.



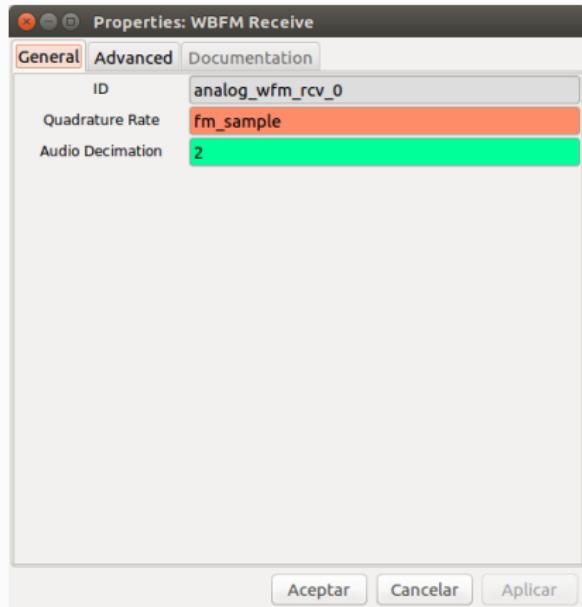
# Configuraciones

Se configura el bloque con el que se demodula la señal con la tarjeta HackRF.



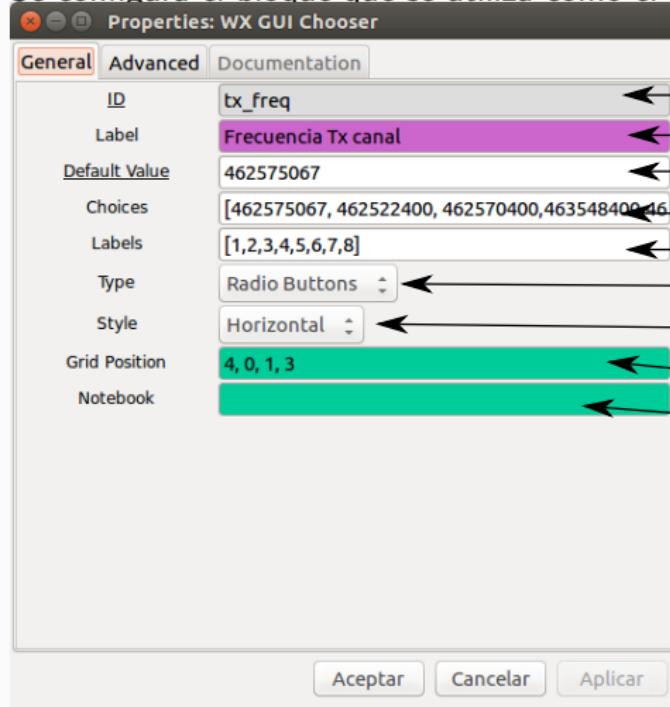
# Configuraciones

Se configura el bloque con el cual se decodifica la información mediante una comparación de frecuencias:



# Configuraciones

Se configura el bloque que se utiliza como el selector de canal:



Identificación del bloque

Nombre del bloque

Frecuencia de referencia

Vector con frecuencias de los canales

Nombre cada canal

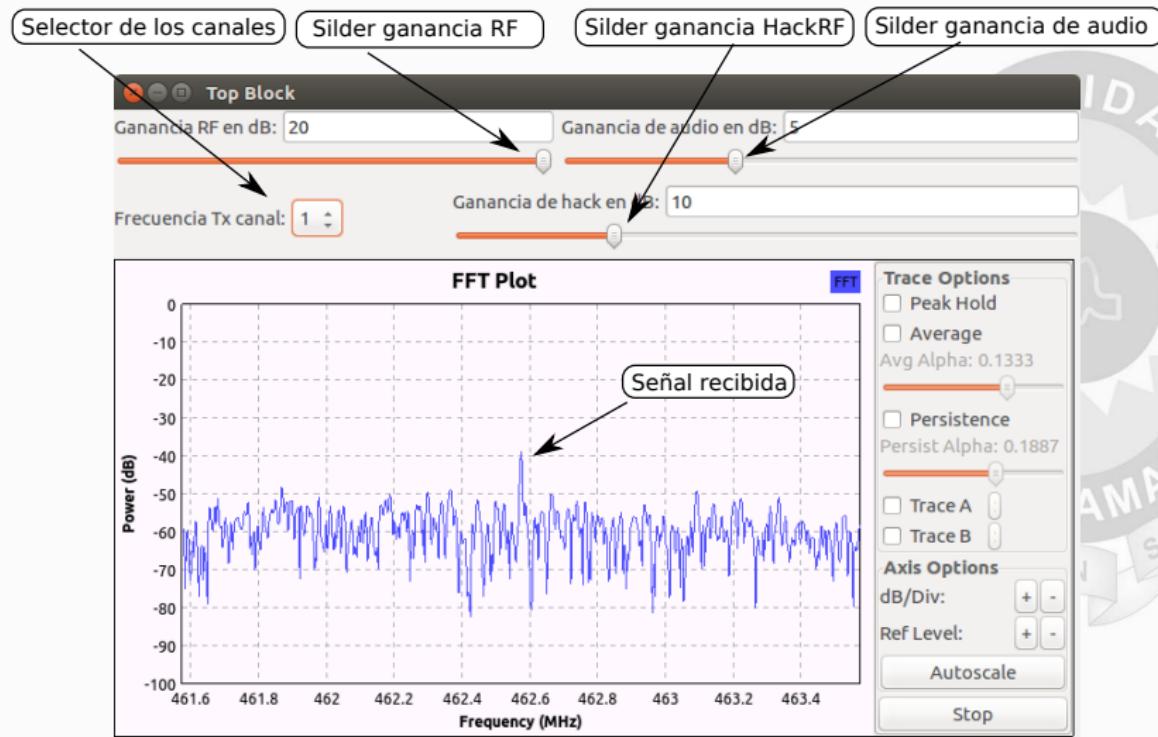
Tipo de selector de canal

Orientación del selector

Posición de la interfaz gráfica

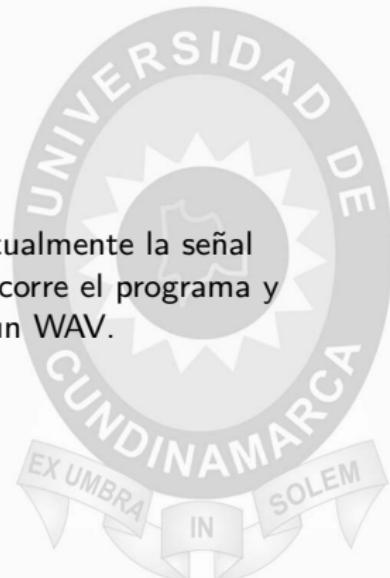
Página de la interfaz

## Interfaz gráfica para la demodulación de la señal



# Conclusión

Se observa y escucha la señal recibida por el EP150 y eventualmente la señal que envía una Hack RF transmisora de FM. El tiempo que corre el programa y esta la interfaz abierta graba y deposita la información en un WAV.

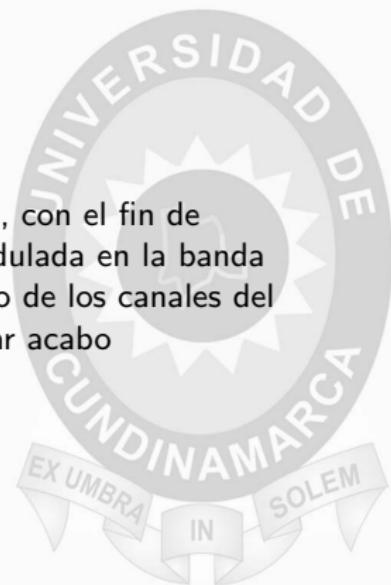


# Lab14

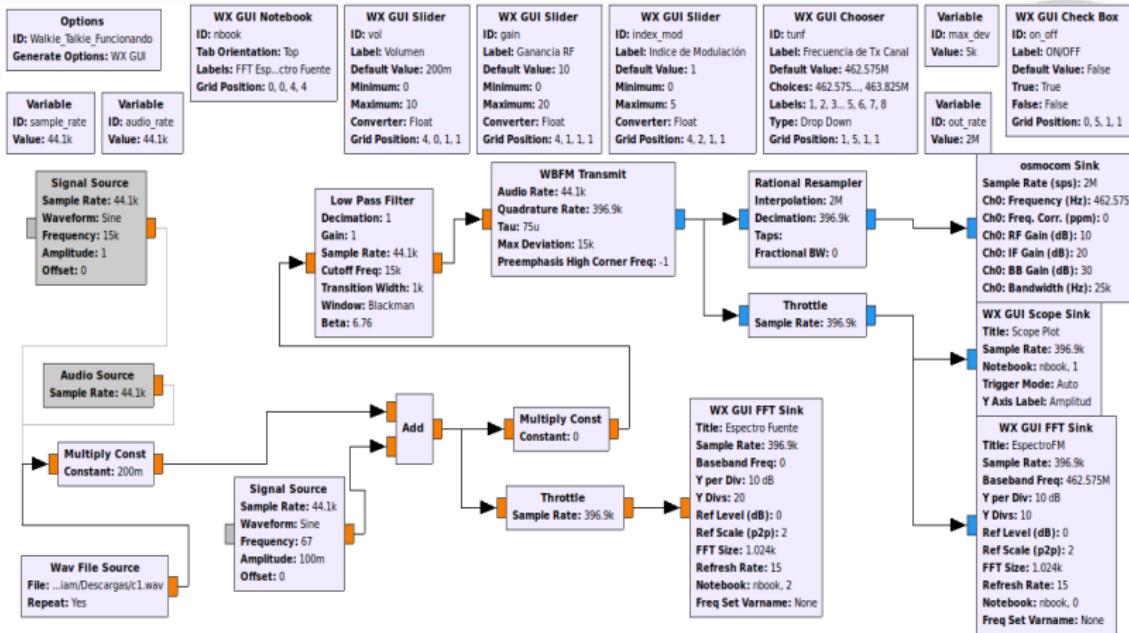
## Transmisión de señales de radio FM para Walkie-Talkie UHF

# Transmisión FM para Walkie-Talkie UHF

Para este laboratorio se utilizará radio definida por software, con el fin de realizar la transmisión de las señales de radiofrecuencia modulada en la banda de UHF en donde se encuentran las frecuencias de cada uno de los canales del dispositivo Motorola EP150, el cual será utilizado para llevar acabo correctamente la práctica.



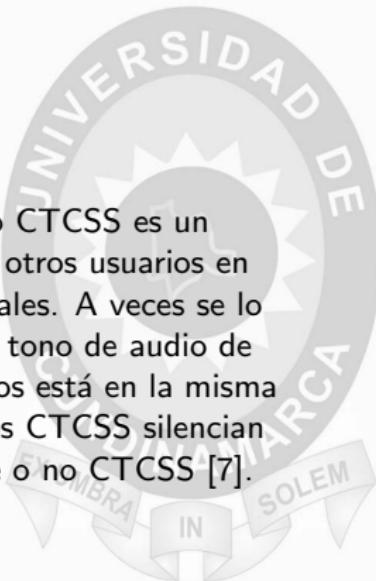
# Diagrama general



# Tonos privados

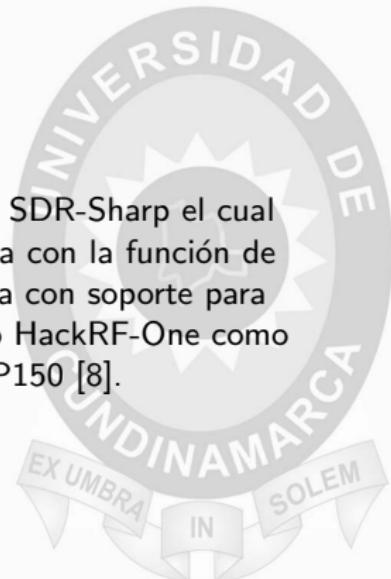
## CTCSS

El sistema continuo de silenciamiento codificado por tonos o CTCSS es un circuito que se utiliza para reducir la molestia de escuchar a otros usuarios en un canal compartido de comunicaciones de radio bidireccionales. A veces se lo conoce como silenciamiento de tono. Lo hace agregando un tono de audio de baja frecuencia a la voz. Cuando más de un grupo de usuarios está en la misma frecuencia de radio (llamados usuarios co-canal), los circuitos CTCSS silencian a los usuarios que están utilizando un tono CTCSS diferente o no CTCSS [7].

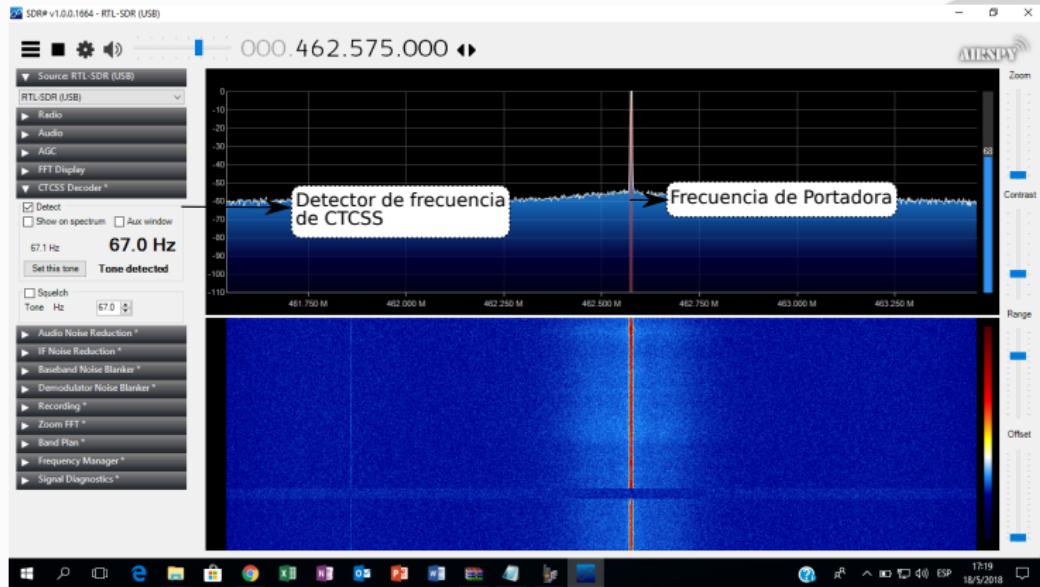


## Detección de portadoras y tonos privados

Para este procedimiento es necesario contar con el software SDR-Sharp el cual es un analizador de espectro de frecuencias y además cuenta con la función de detector de tonos privados CTCSS (actualmente solo cuenta con soporte para Windows), sumado a esto también se utilizará el dispositivo HackRF-One como receptor de señales FM, además del transmisor Motorola EP150 [8].



# Detección de portadoras y tonos privados



## Frecuencias de la señal de portadora

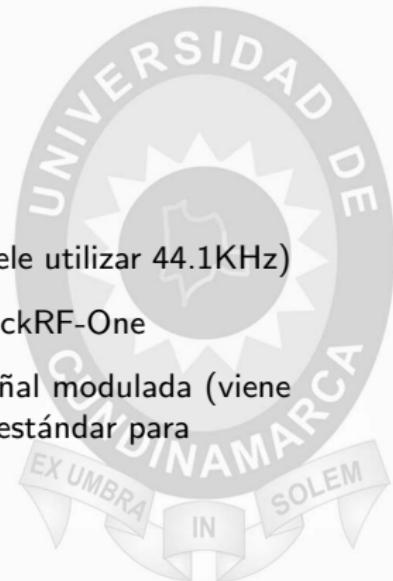
Como se mostró en la figura anterior se pueden observar la frecuencia de las portadoras de cada canal y su respectivo tono CTCSS el cual es 67 Hz como se muestra en la siguiente tabla, dado el caso de que las portadoras halladas y sus tonos no correspondan con la tabla es necesario reiniciar el dispositivo a sus valores de fábrica véase el manual [9].

CANAL	FRECUENCIA DE SEÑAL PORTADORA
1	462,5750 MHz
2	462,6250 MHz
3	462,6750 MHz
4	463,5500 MHz
5	463,6250 MHz
6	463,7625 MHz
7	463,7750 MHz
8	463,8250 MHz

# Desarrollo del diagrama en GNU Radio

Las variables necesarias para el desarrollo del esquema son:

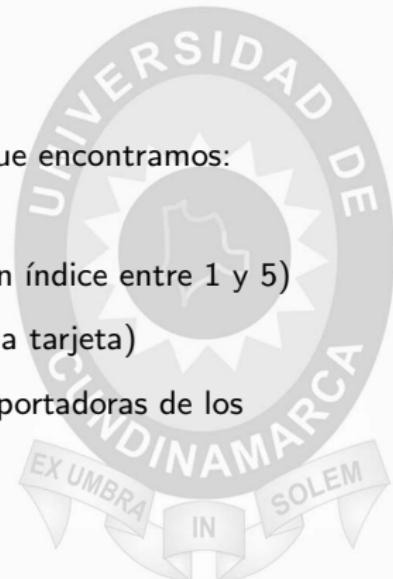
- *sample\_rate* = Frecuencia de muestreo general
- *audio\_rate* = Frecuencia de muestreo del audio (se suele utilizar 44.1KHz)
- *out\_rate* = Frecuencia de muestreo del dispositivo HackRF-One
- *max\_dev* = Desviación máxima en frecuencia de la señal modulada (viene dada por el producto *index\_mod\*15KHz* el cual es el estándar para WBFM)



# Desarrollo del diagrama en GNU Radio

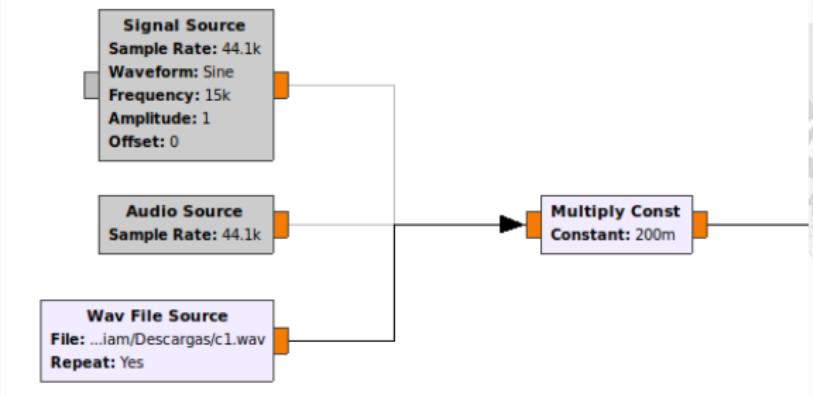
También es necesario utilizar un grupo de slider entre los que encontramos:

- *vol* = Volumen del audio
- *index\_mod* = Índice de modulación (suele utilizarse un índice entre 1 y 5)
- *gain* = Ganancia RF (ganancia prevista por parte de la tarjeta)
- *tunf* = Frecuencia de Tx (frecuencia de las distintas portadoras de los canales)



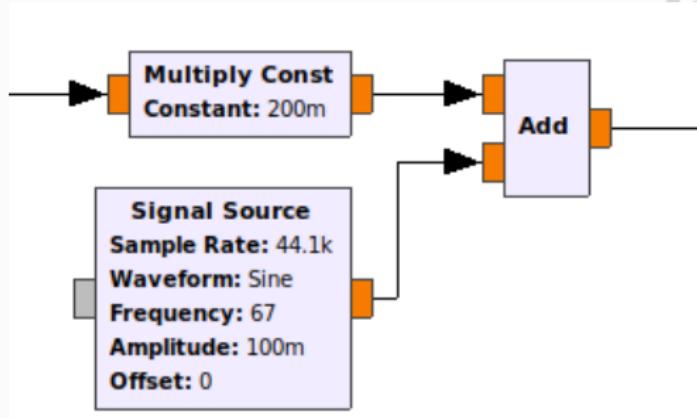
## Desarrollo del diagrama en GNU Radio

Para comenzar con el diagrama de transmisiones es necesario incluir el audio que se desea transmitir esto se puede lograr cargando un archivo WAV (bloque WAV file source), mediante el micrófono del equipo (bloque Audio Source) o finalmente a partir de un generador de señales (bloque Signal Source) seguido a esto se multiplica por una constante, cuyo valor esta dado por el slider vol, con el fin de obtener la amplitud deseada, dicha amplitud no debe exceder los 200m, como se observa a continuación:



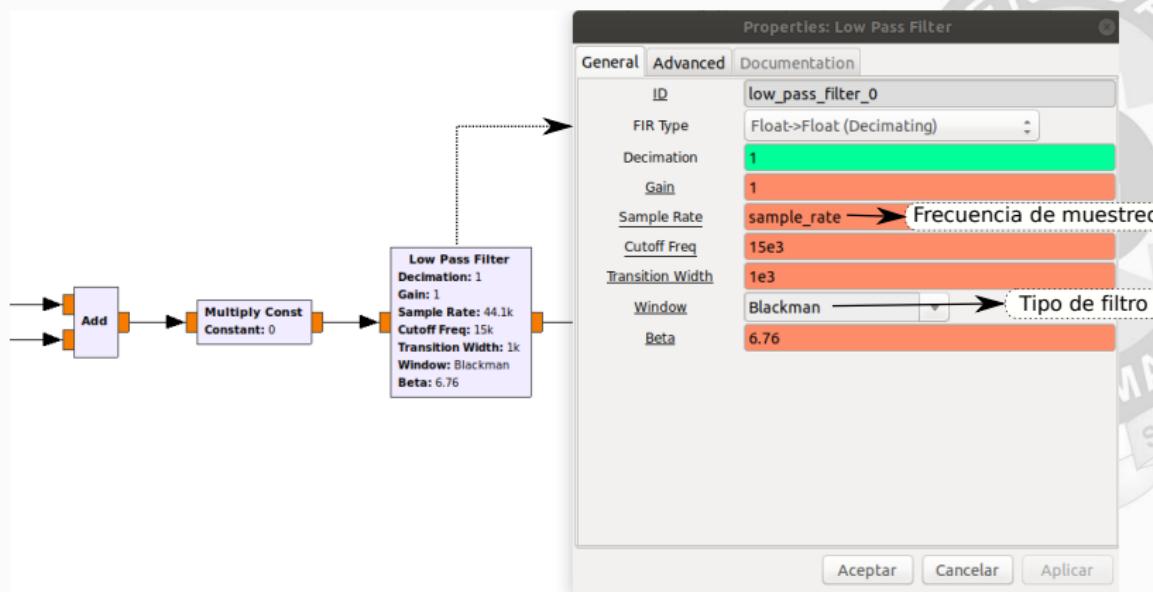
## Desarrollo del diagrama en GNU Radio

Luego se añade el tono CTCSS a la información a transmitir esto con el fin de que la señal transmitida pueda ser escuchada desde el dispositivo Walkie-Talkie como se puede observar:



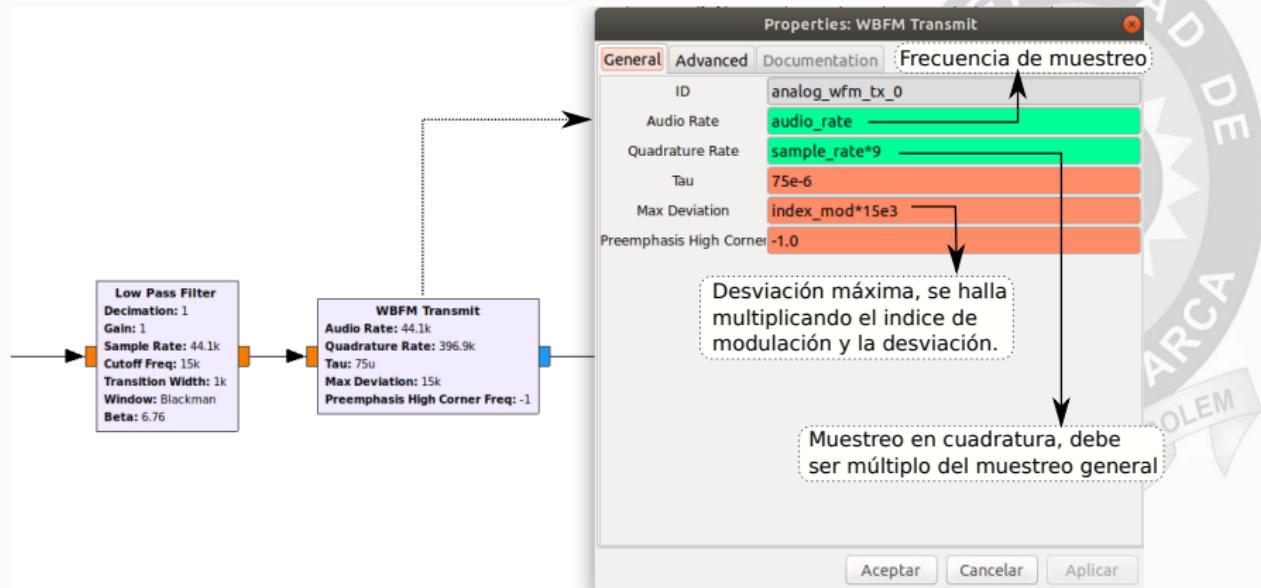
## Desarrollo del diagrama en GNU Radio

En la siguiente etapa se implementa un filtro pasa bajas a la señal evitando frecuencias altas que puedan agregar ruido durante la transmisión.



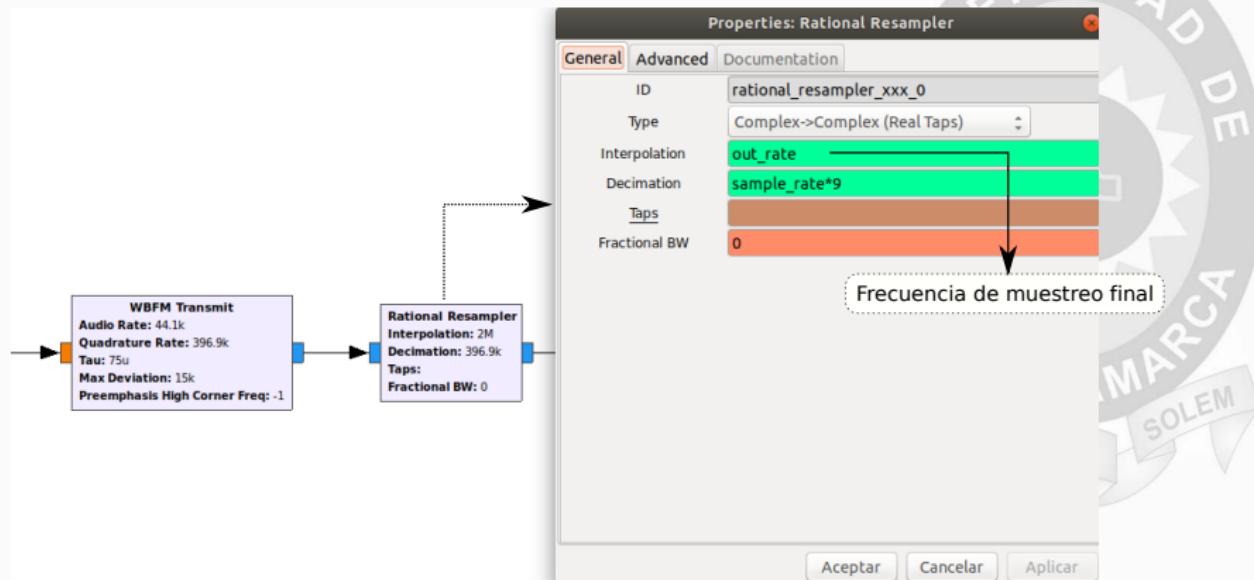
# Desarrollo del diagrama en GNU Radio

Ahora es necesario realizar la modulación de la señal en FM para ello se configuró el bloque WBFM Transmit.



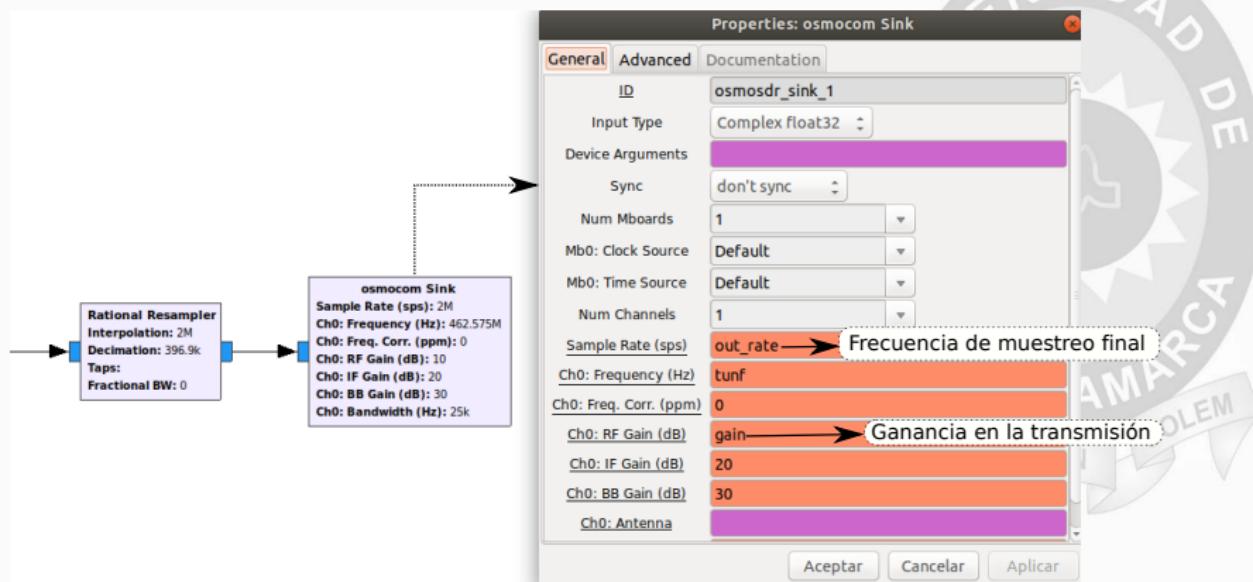
# Desarrollo del diagrama en GNU Radio

Antes de transmitir se realiza un re-muestreo a una frecuencia que permita a la tarjeta enviar el audio sin ningún problema.



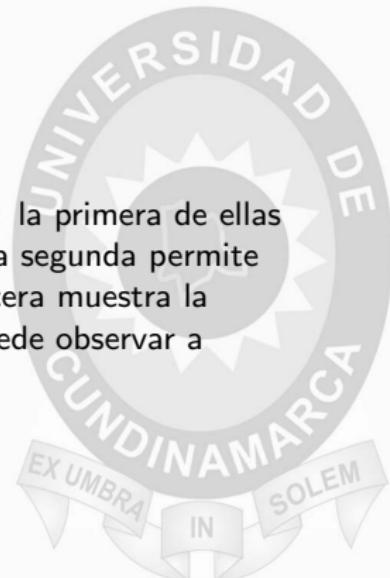
# Desarrollo del diagrama en GNU Radio

Finalmente se realiza la transmisión de la señal utilizando el bloque Osmocom Sink.

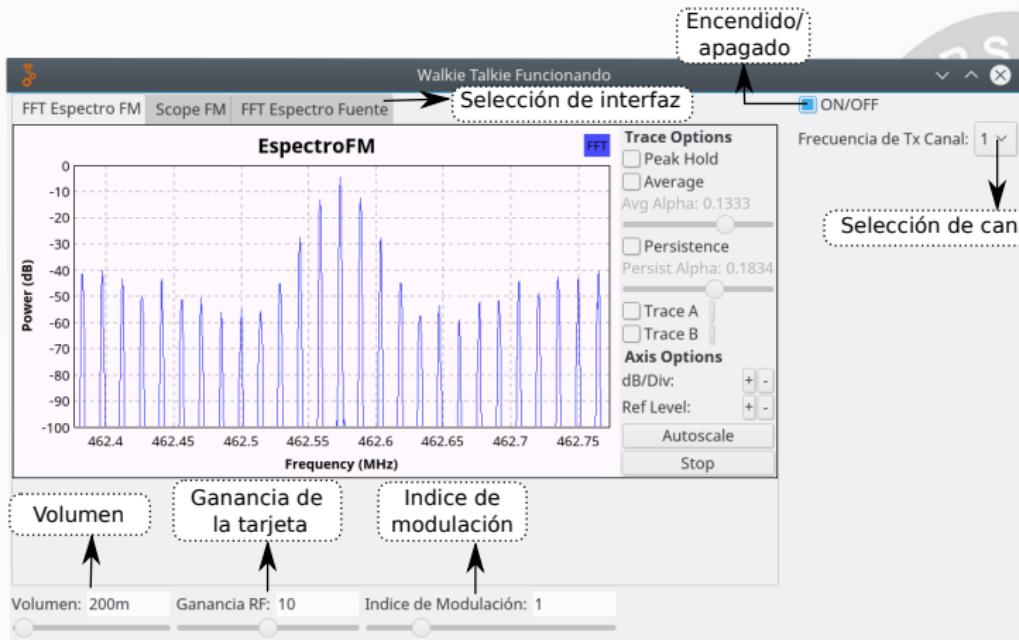


# Interfaz gráfica

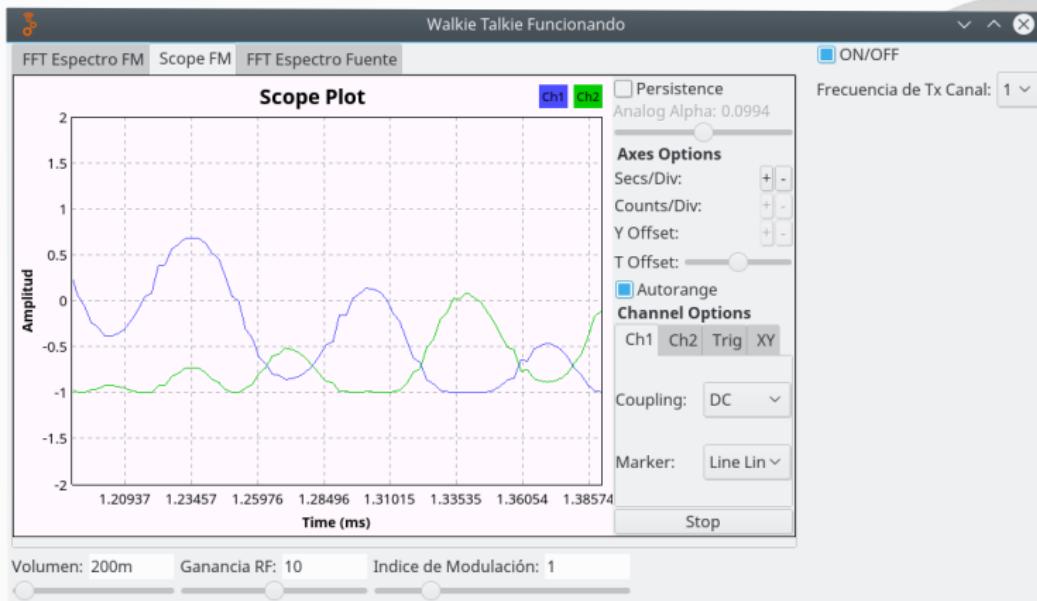
La interfaz gráfica está compuesta por 3 páginas diferentes: la primera de ellas consiste en la transmisión final en términos de frecuencia, la segunda permite ver en la señal a transmitir en términos del tiempo y la tercera muestra la información entrante en términos de frecuencia como se puede observar a continuación:



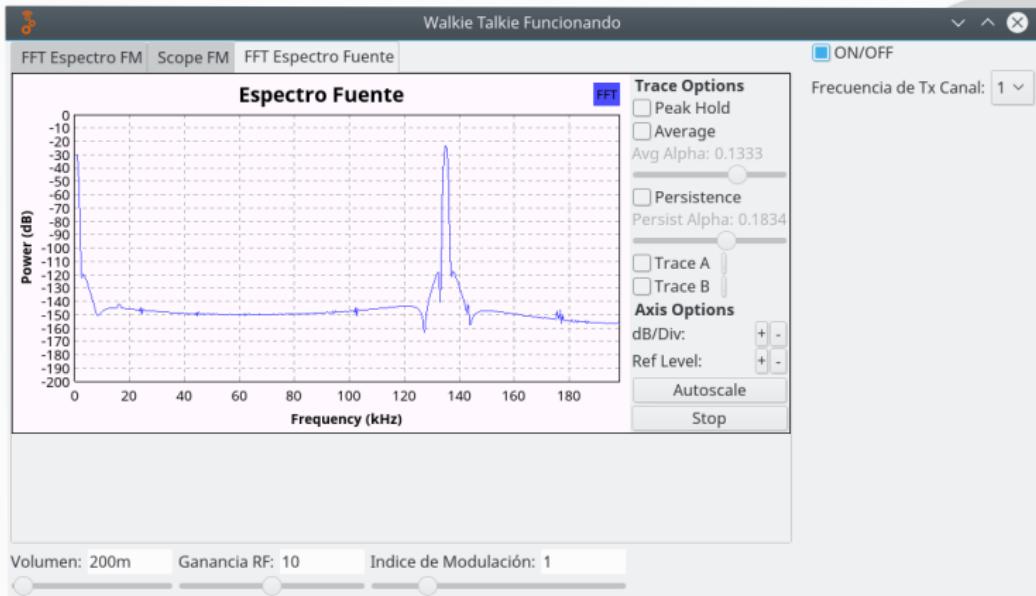
# Interfaz gráfica



# Interfaz gráfica



# Interfaz gráfica



# Lab15

## Receptor AM

## Receptor AM

El funcionamiento de un demodulador IQ se puede explicar representando su señal de entrada de RF  $sRF(t)$  como una combinación de dos portadoras de cuadratura modulada de doble banda lateral:

$$sRF(t) = SI(t) + SQ(t) = I(t)\cos(wRF)t - Q(t)\sin(wRF)t$$

El componente en fase  $I(t)$  y el componente de cuadratura  $Q(t)$  son señales de banda base que se pueden ver como entradas a un modulador IQ ideal que genera  $1sRF(t)$ .

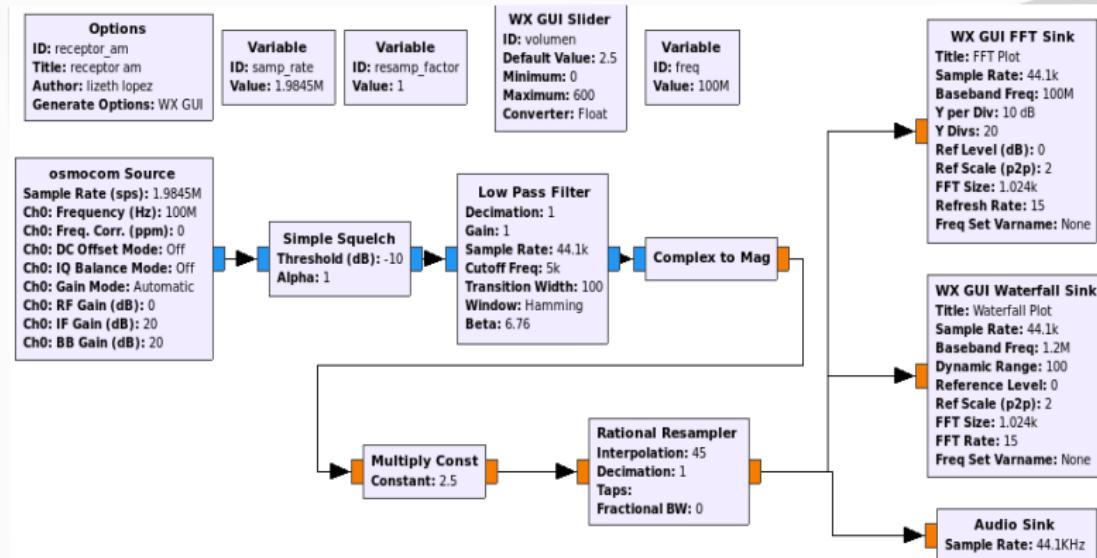
## Receptor AM

A continuación, construiremos un receptor de AM simple para la recepción de banda de transmisión y onda corta. En el receptor se utiliza el hardware HackRF. Además, muestra cómo configurar y mostrar la frecuencia del receptor.

El diseño del receptor AM principalmente consta de:

- *Osmocom source* = Fuente HackRF
- *Low pass filter* = Filtro pasa bajas con frecuencia de corte de 5KHz.
- *Complex to Mag* = Permite realizar la demodulación de una manera más compleja
- *Multiply const* = Para tener cierta ganancia
- *Sumidero de audio Audio Sink* = para poder escuchar la señal demodulada.

# Receptor AM



# Receptor AM

Properties: Options

General	Advanced	Documentation
ID	receptor_am	→ Nombre del archivo Python
Title	receptor am	→ Título de la ventana GUI
Author	microondas	→ Nombre del autor
Description		
Canvas Size	1280, 1024	→ tamaño de lienzo de GRC
Generate Options	WX GUI	→ Tipo de código a generar
Run	Autostart	→ Inicio de diagrama de flujo
Max Number of Output	0	
Realtime Scheduling	Off	▼

Aceptar Cancelar Aplicar

Properties: Variable

General	Advanced	Documentation
ID	samp_rate	
Value	256e3	

Se utiliza esta frecuencia de muestreo y no otra por varios motivos:  
-Cumplir con el Teorema de Nyquist y evitar así la pérdida de la información contenida en la señal.  
-Evitar el oversampling, malgastando ciclos de la CPU  
  
-Facilitar el remuestreo necesario para la utilización de la tarjeta de sonido del PC

Aceptar Cancelar Aplicar

# Receptor AM

The screenshot shows the Gqrx Software interface with a focus on the 'Properties' dialog for an 'osmocom Source' block. The dialog has tabs for General, Advanced, and Documentation, with General selected. The 'osmocom Source' block is highlighted on the left, showing its parameters: Sample Rate (sps): 256k, Ch0: Frequency (Hz): 18M, Ch0: Freq. Corr. (ppm): 0, Ch0: DC Offset Mode: Off, Ch0: IQ Balance Mode: Off, Ch0: Gain Mode: Automatic, Ch0: RF Gain (dB): 10, Ch0: IF Gain (dB): 20, and Ch0: BB Gain (dB): 20.

The 'Properties: osmocom Source' dialog contains the following settings:

- ID:** osmosdr\_source\_0 → Nombre generado en Python
- Output Type:** Complex float32 → Tipo de salida del bloque
- Device Arguments:** (empty)
- Sync:** don't sync
- Num Mboards:** 1
- Mb0: Clock Source:** Default
- Mb0: Time Source:** Default
- Num Channels:** 1
- Sample Rate (sps):** samp\_rate → Frecuencia de muestreo definida en la variable samp\_rate
- Ch0: Frequency (Hz):** 18e6 → Frecuencia captada por la antena
- Ch0: Freq. Corr. (ppm):** 0

At the bottom of the dialog are buttons: Aceptar, Cancelar, and Aplicar.

# Receptor AM

Properties: Low Pass Filter

General Advanced Documentation

ID: low\_pass\_filter\_0 → nombre de la variable generada en Python

FIR Type: Complex->Complex (Decimating) → las transmisiones de entrada y salida son complejas, con la opción de diezmar la salida.

Decimation: resamp\_factor → permite disminuir la tasa de muestreo.

Gain: 1 → ganancia de filtro pasa bajo.

Sample Rate: 44100 → establece la frecuencia de muestreo del filtro, en Hz.

Cutoff Freq: 5e3 → Establece la frecuencia de corte del filtro, en Hz.

Transition Width: 100 → establece el ancho de transición entre la banda de paso y la banda de detención.

Window: Hamming → Especifica la función de ventana que se aplica al filtro FIR. Aplicamos una ventana Hamming.

Beta: 6.76 → Parámetro beta para la ventana de Káiser.

Aceptar Cancelar Aplicar

# Receptor AM

Este bloque permite demodular señales AM detectando el envolvente o magnitud de una señal compleja. A partir de este bloque los datos que fluyen por el grafo son de tipo float y esto obliga, a que los datos de los bloques que se añadan posteriormente sean del mismo tipo.

WX GUI Slider  
ID: volumen  
Default Value: 2.5  
Minimum: 0  
Maximum: 100  
Converter: Float

Properties: Multiply Const

General	Advanced	Documentation
ID blocks_multiply_const_vxx_0	IO Type Float	Constant volumen
Vec Length 1		

Este bloque se añade debido a que los valores de las muestras son demasiados pequeños, y para solucionarlo se añade este bloque que aumente dichos valores.

• "/home/lizeth/Nueva carpeta/transmisoram.nrr"

Aceptar Cancelar Aplicar

# Receptor AM

Rational Resampler

Properties: Rational Resampler

General Advanced Documentation

ID: rational\_resampler\_xxx\_0

Type: Float->Float (Real Taps)

Interpolation: resamp\_factor → Aumenta 1 vez la tasa de muestreo

Decimation: 45 → Divide en 45 la tasa de muestreo

Taps: Key: decim  
Type: int  
Value: 45

Fractional BW: 0

Rational Resampler  
Interpolation: 1  
Decimation: 45  
Taps:  
Fractional BW: 0

Aceptar Cancelar Aplicar

The screenshot shows a 'Properties' dialog box for a 'Rational Resampler' component. The 'Advanced' tab is selected. Under 'Interpolation', the value 'resamp\_factor' is set to 1, with a tooltip 'Aumenta 1 vez la tasa de muestreo'. Under 'Decimation', the value '45' is set, with a tooltip 'Divide en 45 la tasa de muestreo'. The 'Taps' section shows a key 'decim' with type 'int' and value '45'. Below the dialog is a component icon labeled 'Rational Resampler' with parameters: Interpolation: 1, Decimation: 45, Taps: Fractional BW: 0.

# Receptor AM

**WX GUI FFT Sink**  
Title: FFT Plot  
Sample Rate: 44.1k  
Baseband Freq: 100M  
Y per Div: 10 dB  
Y Divs: 20  
Ref Level (dB): 0  
Ref Scale (p2p): 2  
FFT Size: 1.024k  
Refresh Rate: 15  
Freq Set Varname: None

funciona como analizador de espectros realizando una Transformada discreta de Fourier de Tiempo Reducido.

**Properties: WX GUI FFT Sink**

General	Advanced	Documentation
ID: wxgui_fftsink2_0	Type: Float	Title: FFT Plot
Sample Rate: samp_rate/45	Baseband Freq: freq	Y per Div: 10 dB
Y Divs: 20	Ref Level (dB): 0	Ref Scale (p2p): 2.0
Ref Scale (p2p): 2	FFT Size: 1024	Refresh Rate: 15
FFT Size: 1.024k	Refresh Rate:	Freq Set Varname: None

**WX GUI Waterfall Sink**  
Title: Waterfall Plot  
Sample Rate: 44.1k  
Baseband Freq: 1.2M  
Dynamic Range: 100  
Reference Level: 0  
Ref Scale (p2p): 2  
FFT Size: 1.024k  
FFT Rate: 15  
Freq Set Varname: None

Este muestra el espectro de tiempo FFT. Establece el tamaño de FFT EN 2048. La entrada agregada se conecta a la salida.

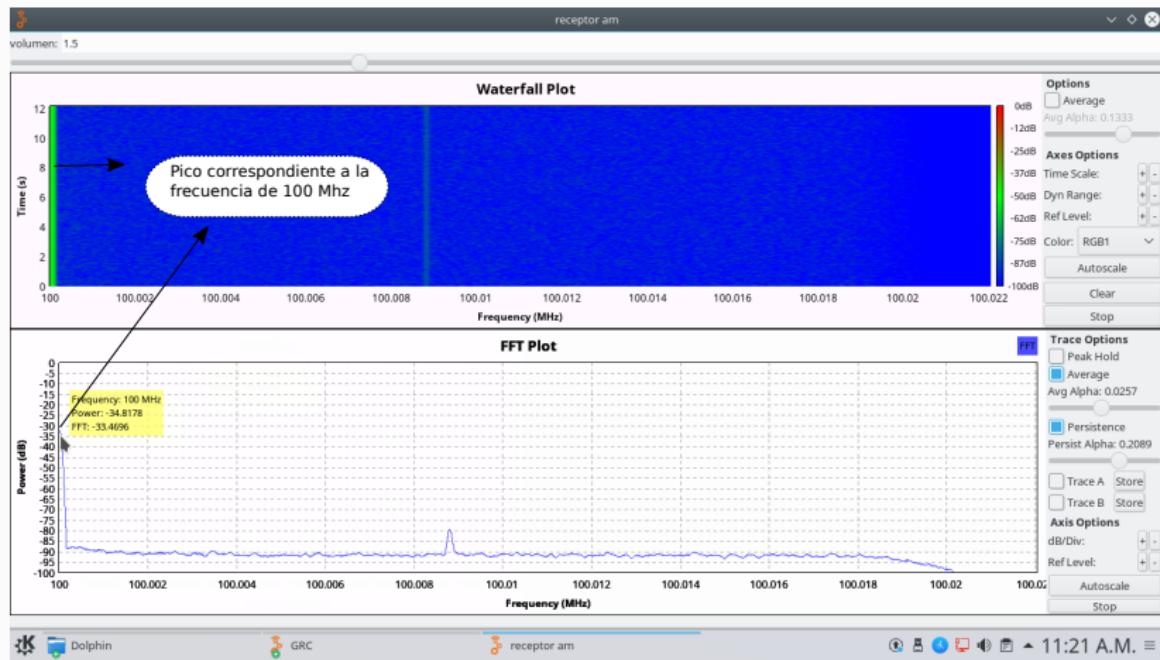
**Properties: WX GUI Waterfall Sink**

General	Advanced	Documentation
ID: wxgui_waterfallsink2_0	Type: Float	Title: Waterfall Plot
Sample Rate: 44100	Baseband Freq: freq	Dynamic Range: 100
Baseband Freq: freq	Reference Level: 0	Ref Scale (p2p): 2.0
Dynami Range: 100	Ref Scale (p2p): 2	FFT Size: 1024
Reference Level: 0	FFT Size: 1024	FFT Rate: 15
Ref Scale (p2p): 2	FFT Rate: 15	Average: Off

**Audio Sink**  
Sample Rate: 44.1KHz

Bloque donde se puede escuchar la señal demodulada, cuya función es dirigir la señal a la tarjeta de sonido del ordenador

# Receptor AM



# Lab16

## Transmisor AM

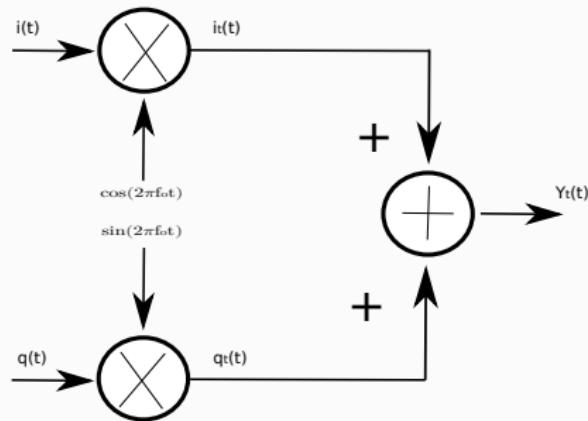
## Modulación de amplitud IQ

La modulación de amplitud (AM) funciona mediante la variación de la amplitud de la señal transmitida de alta frecuencia que varía en proporción con una señal que por naturaleza es de baja frecuencia. La señal de baja frecuencia es la información que se desea transmitir también llamada señal moduladora, usualmente se encuentra en el orden de los KHz y la señal de frecuencia alta se le llama portadora usualmente en el orden de los MHz.

Es una modulación digital que transmite dos mensajes independientes y está conformado por dos canales ortogonales  $i(t)$  y  $q(t)$  que pueden ser transmitidos simultáneamente.

El canal  $i(t)$  es la señal moduladora (entrada) que contiene la información y es la parte real para la transmisión, el canal  $q(t)$  es la señal portadora que debe estar desfasada  $90^\circ$  de la señal moduladora y es la parte imaginaria para la transmisión[10].

# Modulación de amplitud IQ



Matemáticamente se expresa:

$$i_t(t) = i(t)\cos(2\pi f_0 t + 0^\circ)$$

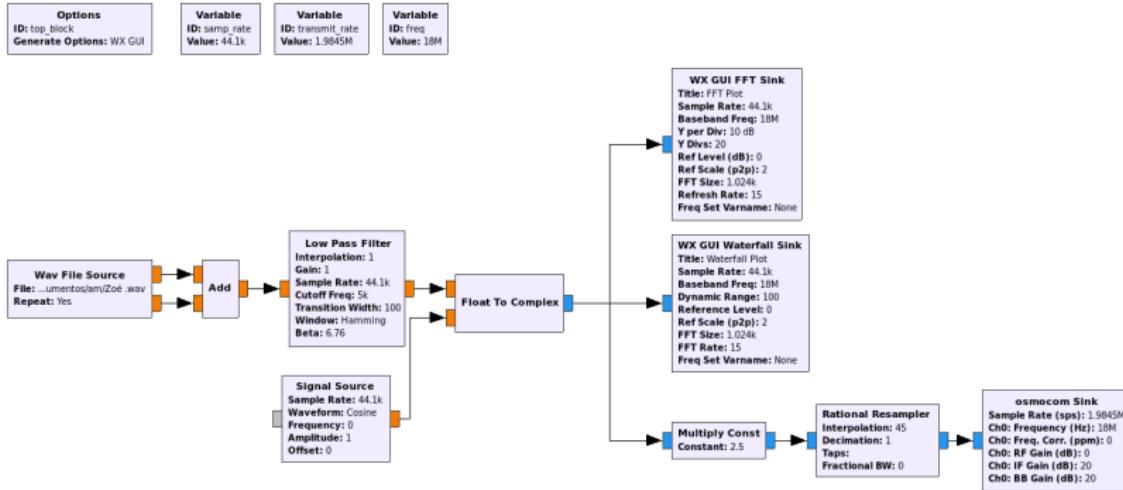
$$q_t(t) = q(t)\cos(2\pi f_0 t + 90^\circ) = \\ q(t)\sin(2\pi f_0 t)$$

$$y_t(t) = \sqrt{i^2(t) + q^2(t)}\cos(2\pi f_0 t + \theta(t))$$

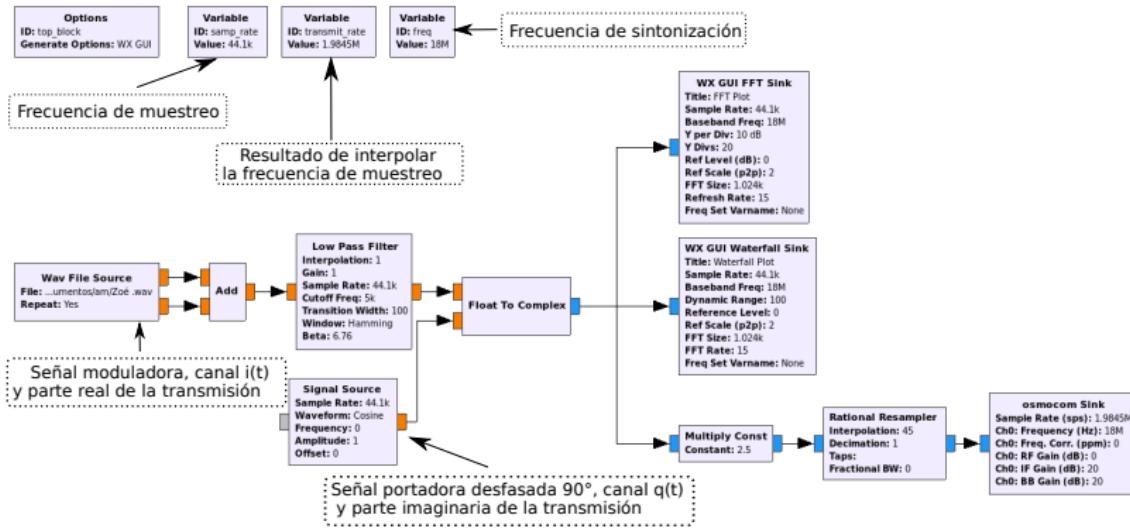
$$\theta(t) = \tan^{-1} \frac{q(t)}{i(t)}$$

$$-180^\circ < \theta < 180^\circ$$

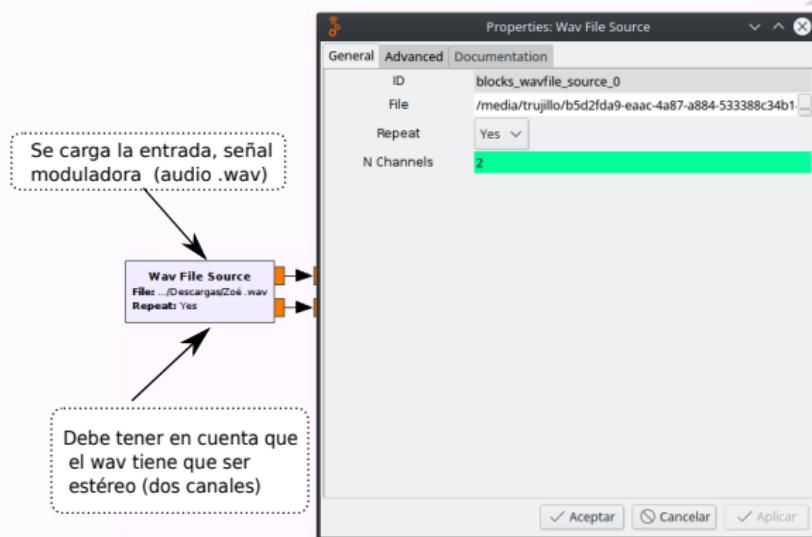
# Modulación de amplitud IQ



# Modulación de amplitud IQ

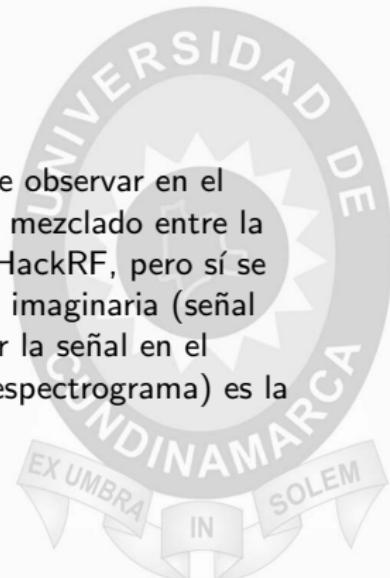


# Modulación de amplitud IQ



# Modulación de amplitud IQ

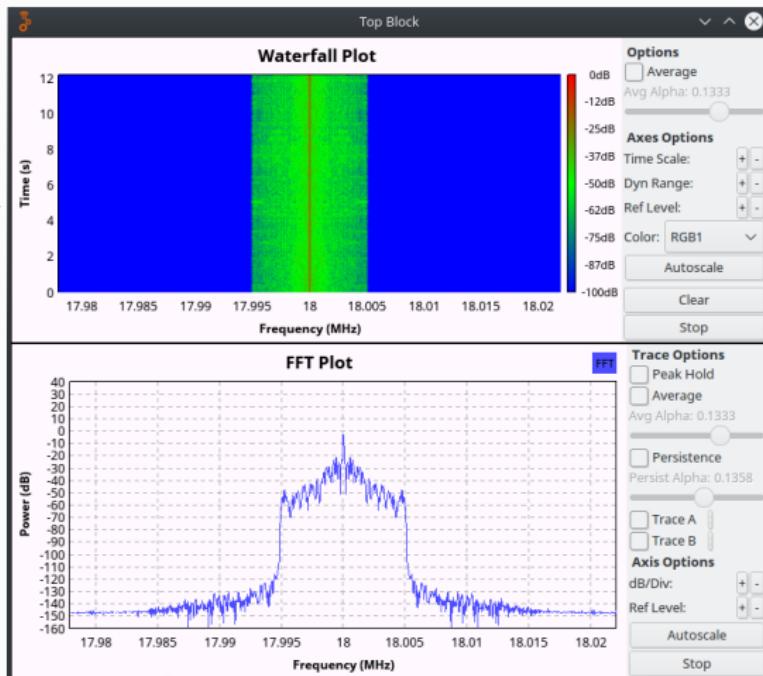
El resultado de la modulación (señal modulada) no se puede observar en el dominio del tiempo ni de la frecuencia ya que el proceso de mezclado entre la señal portadora y moduladora se hace directamente con el HackRF, pero sí se puede observar el resultado de la unión entre la parte real e imaginaria (señal moduladora y portadora). La frecuencia utilizada para mirar la señal en el dominio de la frecuencia (FFT) y el diagrama de cascada (espectrograma) es la sintonizada en el radio.



# Resultado

**WX GUI Waterfall Sink**  
**Title:** Waterfall Plot  
**Sample Rate:** 44.1k  
**Baseband Freq:** 18M  
**Dynamic Range:** 100  
**Reference Level:** 0  
**Ref Scale (p2p):** 2  
**FFT Size:** 1.024k  
**FFT Rate:** 15  
**Freq Set Varname:** None

**WX GUI FFT Sink**  
**Title:** FFT Plot  
**Sample Rate:** 44.1k  
**Baseband Freq:** 18M  
**Y per Div:** 10 dB  
**Y Divs:** 20  
**Ref Level (dB):** 0  
**Ref Scale (p2p):** 2  
**FFT Size:** 1.024k  
**Refresh Rate:** 15  
**Freq Set Varname:** None



# BIBLIOGRAFÍA

## Referencias I

[1] Seeber, Balint.

"GNU Radio Tutorials Labs 1 – 5". [Online]

[https://files.ettus.com/tutorials/labs/Lab\\_1-5.pdf](https://files.ettus.com/tutorials/labs/Lab_1-5.pdf).

[2] Vachhani, Khyati. Gokhruwala, Kenil; Kumar, Jay

"Design Analysis of Digital Modulation Schemes with GNU Radio".

[Descargado de:] [https://www.researchgate.net/publication/281106848\\_Design\\_Analysis\\_of\\_Digital\\_Modulation\\_Schemes\\_with\\_GNU\\_Radio](https://www.researchgate.net/publication/281106848_Design_Analysis_of_Digital_Modulation_Schemes_with_GNU_Radio).

[3] Wikipedia.

"Frecuencia modulada". [Online]

[https://es.wikipedia.org/wiki/Frecuencia\\_modulada](https://es.wikipedia.org/wiki/Frecuencia_modulada).

[4] Vaught, Andy.

"Introduction to Named Pipes". [Online]

<https://www.linuxjournal.com/article/2156>.

## Referencias II

[5] Wiki Opendigitalradio.

"Simple FM transmitter using gnuradio". [Online]

[http://wiki.opendigitalradio.org/Simple\\_FM\\_transmitter\\_using\\_gnuradio](http://wiki.opendigitalradio.org/Simple_FM_transmitter_using_gnuradio).

[6] Electronisys.

"Ficha técnica de radio portátil UHF Motorola EP150". [Descargado de:]

[http://www.electronisys.cl/index.php?route=product/product&product\\_id=50](http://www.electronisys.cl/index.php?route=product/product&product_id=50).

[7] Wikipedia.

"Continuous Tone-Coded Squelch System". [Online]

[https://en.wikipedia.org/wiki/Continuous\\_Tone-Coded\\_Squelch\\_System](https://en.wikipedia.org/wiki/Continuous_Tone-Coded_Squelch_System).

## Referencias III



### [8] Airspy.

"Airspy Redefining the Radio Experience". [Online]  
<https://airspy.com/download/>

### [9] Motorola Solutions.

"Two-Way Radios". [Online]  
<https://www.motorolasolutions.com/content/dam/msi/docs/enxl/products/two-way-radios-business/portable-radios/on-site-smallbusiness/.>

### [10] Documento PDF

"Lecture 9 Analog and Digital I/Q Modulation". [Online]  
<http://web.mit.edu/6.02/www/f2006/handouts/Lec9.pdf.>

## Referencias IV

[11] TOMASI, WAYNE.

"Sistemas de comunicaciones electrónicas". [Descargado de:]

[http://fernandoarciniega.com/books/  
sistemas-de-comunicaciones-electronicas-tomasi-4ta-edicion  
pdf.](http://fernandoarciniega.com/books/sistemas-de-comunicaciones-electronicas-tomasi-4ta-edicion.pdf)

[12] Arévalo Nelson

"Transmisión y recepción de imagen con modulación QPSK usando radio definido por software". [Online] [https://repository.unimilitar.edu.co/handle/10654/16606.](https://repository.unimilitar.edu.co/handle/10654/16606)

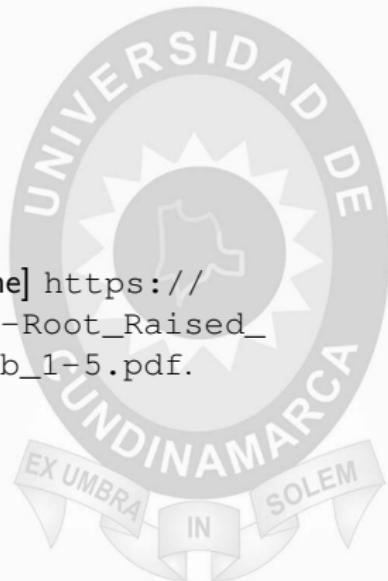
[13] Aaron Scher

"Cómo convertir un flujo de datos digitales en una señal analógica de banda base utilizando un filtro FIR interpolador". [Online] [http://aaronsscher.com/GNU\\_Radio\\_Companion\\_Collection/BPSK\\_modulatorB.html.](http://aaronsscher.com/GNU_Radio_Companion_Collection/BPSK_modulatorB.html)

## Referencias V

### [14] Agencia Nacional Europea

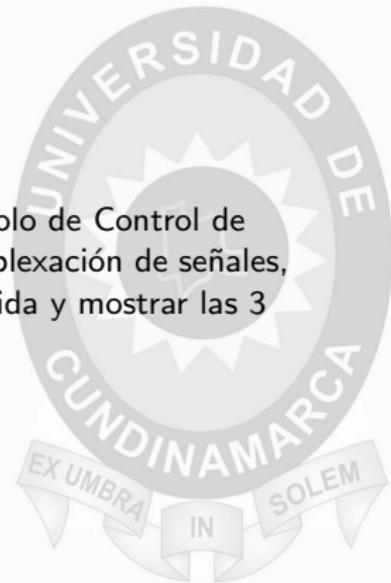
"Señales de coseno realizado de raíz cuadrada ". [Online] [https://gssc.esa.int/navipedia/index.php/Square-Root\\_Raised\\_Cosine\\_Signals\\_\(SRRC\)/tutorials/labs/Lab\\_1-5.pdf](https://gssc.esa.int/navipedia/index.php/Square-Root_Raised_Cosine_Signals_(SRRC)/tutorials/labs/Lab_1-5.pdf).



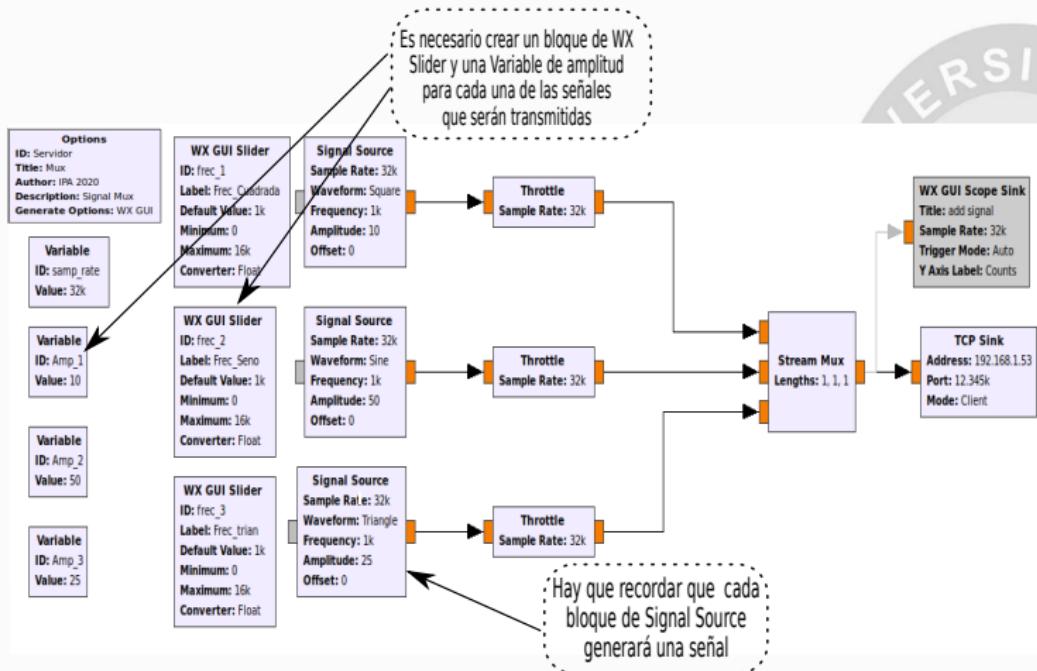
## SOLUCIONES DE ACTIVIDADES Y PREGUNTAS

# Transmisión de señales por multiplexación

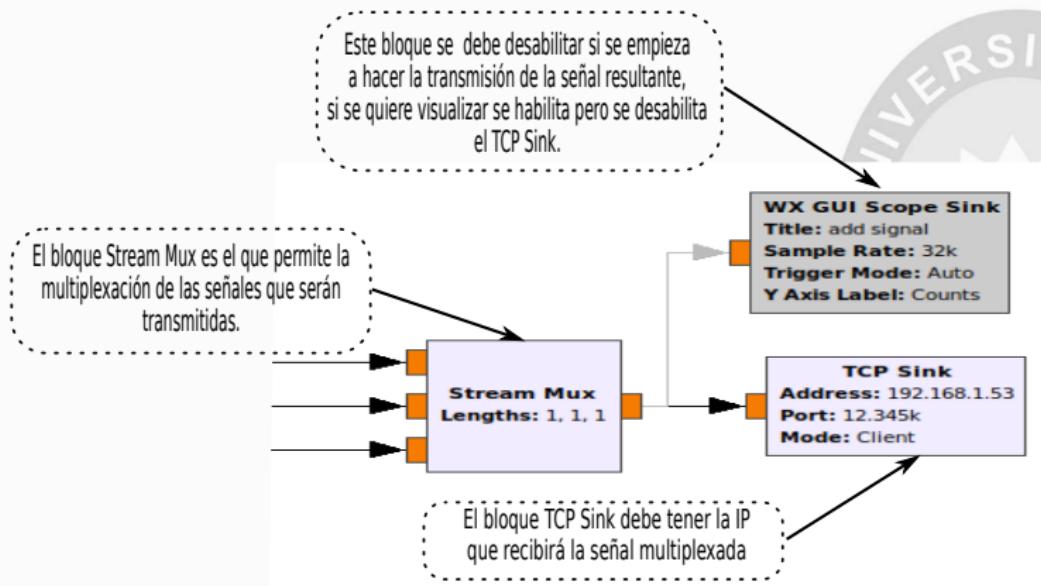
Transmitir 3 señales periódicas por medio del TCP (Protocolo de Control de Transmisión) desde el cliente, mediante el proceso de multiplexación de señales, al servidor, que se encargará de demultiplexar la señal recibida y mostrar las 3 que fueron transmitidas en el Scope Sink.



# Solución 1: Servidor



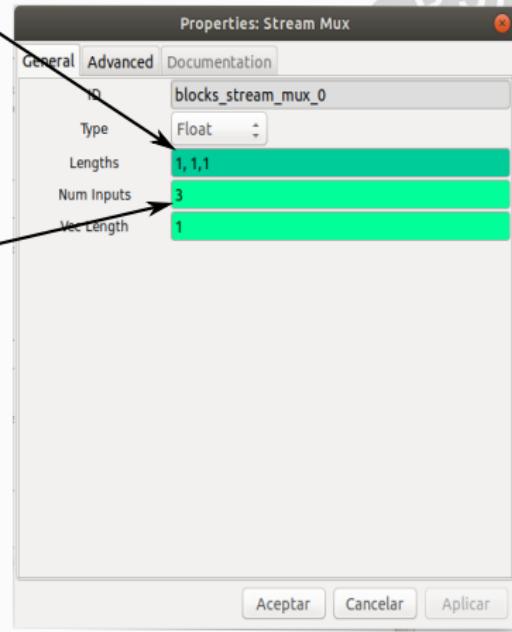
## Solución 1: Servidor



## Solución 1: Servidor

La opción Lengths es una lista que especifica el número de items de cada una de las señales que se van a multiplexar.

También es necesario definir el número de señales que entrarán al bloque



# Solución 1: Cliente

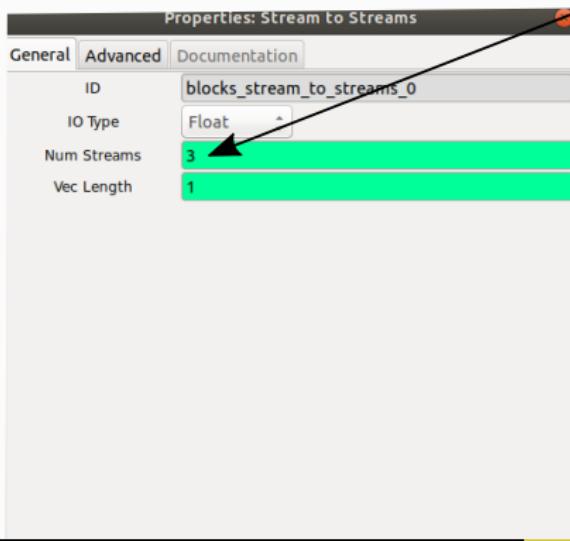
**Options**  
**ID:** Cliente  
**Title:** Demux  
**Author:** IPA 2020  
**Description:** Signal Demux  
**Generate Options:** WX GUI

**Variable**  
**ID:** samp\_rate  
**Value:** 32k



El bloque Stream to Streams permite la demultiplexación de la señal que es recibida por el TCP Source.

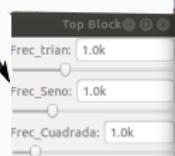
# Solución 1: Cliente



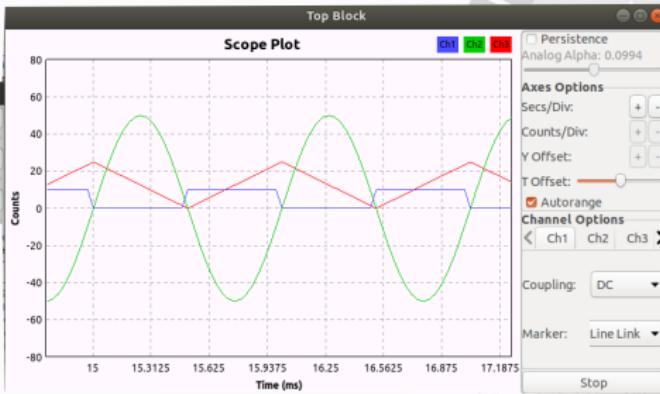
En las propiedades de Stream to Streams, en Num Streams, se hace referencia al número de señales que conforman la señal multiplexada y así las pueda mostrar cada una con su respectiva información

## Solución 1: Cliente

En el Top Block se encuentran los WX GUI Sliders: de cada una de las señales que se transmiten.



El resultado esperado de la actividad sería este:



En el Scope Plot se muestran las señales recibidas y se evidencian los cambios que se les pueden hacer a las frecuencias.

# Solución de la actividad "Modificación de las variables de una señal periódica"

En Options le damos el nombre a la ventana que nos muestra el osciloscopio, ademas se añaden los datos de la simulación

Options  
ID: ventana  
Title: Modifica... la Señal  
Author: Jorge C. Arango - 2020  
Description: la se... pr sink  
Generate Options: WX GUI

Variable  
ID: samp\_rate  
Value: 128k

Es necesario agregar un bloque WX GUI Slider por cada variable que se desea ver en el Osciloscopio

WX GUI Slider  
ID: Fase\_Ref  
Label: Fase  
Default Value: 0  
Minimum: -100  
Maximum: 100  
Converter: Integer  
Grid Position: 1, 4, 1, 1

WX GUI Slider  
ID: Frecuencia\_Ref  
Label: Frecuencia  
Default Value: 0  
Minimum: 0  
Maximum: 10k  
Converter: Float  
Grid Position: 1, 1, 1, 1

WX GUI Slider  
ID: Amplitud\_Ref  
Label: Amplitud  
Default Value: 1  
Minimum: 0  
Maximum: 1k  
Converter: Float  
Grid Position: 1, 2, 1, 1

WX GUI Slider  
ID: Nivel\_dc\_Ref  
Label: Nivel dc  
Default Value: 0  
Minimum: 0  
Maximum: 1k  
Converter: Float  
Grid Position: 1, 3, 1, 1

WX GUI Slider  
ID: Ref\_Offset  
Label: Ref\_Offset  
Default Value: 0  
Minimum: -100  
Maximum: 100  
Converter: Integer  
Grid Position: 1, 4, 1, 1

WX GUI Slider  
ID: Amplitude\_Ref  
Label: Amplitude\_Ref  
Default Value: 0  
Minimum: 0  
Maximum: 1k  
Converter: Float  
Grid Position: 2, 1, 1, 1

WX GUI Slider  
ID: Delay\_Ref  
Label: Delay\_Ref  
Default Value: 0  
Minimum: 0  
Maximum: 1k  
Converter: Integer  
Grid Position: 2, 2, 1, 1

WX GUI Scope Sink  
Title: Señal Modificada  
Sample Rate: 128k  
Grid Position: 2, 1, 1, 5  
X Axis Label: Tiempo  
Y Axis Label: Counts

```

graph LR
    SS1[Signal Source] --> D1[Delay]
    SS2[Signal Source] --> D2[Delay]
    D1 --> T1[Throttle]
    D2 --> T2[Throttle]
    T1 --> WGS[Sink]
    T2 --> WGS
  
```

The diagram illustrates the signal flow for modifying periodic signals. It starts with two Signal Source blocks (Cosine waves at 120Hz). These feed into two Delay blocks (delay = 0). The outputs of the delays feed into two Throttle blocks (sample rate = 128k). The outputs of the throttles feed into a single WX GUI Scope Sink block.

Throttle disminuye el flujo de muestras de modo que la velocidad promedio no supere las muestras por segundo

El Signal Source es la fuente de la señal a modificar o de referencia

El Delay se encarga de retrasar la entrada de la señal por un cierto numero de muestras

# Solución de la actividad "Modificación de las variables de una señal periódica"

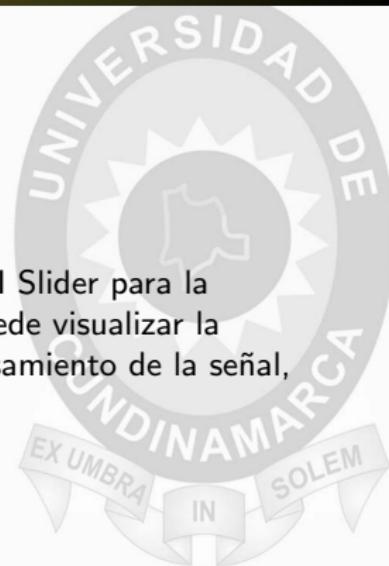
- 1) ¿Cómo se varía la amplitud, frecuencia, nivel DC y fase de una señal periódica en GNU Radio y hacer modificables estos valores por medio de un "Slider", visualizando su variación desde el osciloscopio?



# Solución de la actividad "Modificación de las variables de una señal periódica"

Los pasos son:

1. En la página anterior, se pueden observar los WX GUI Slider para la frecuencia, la amplitud y el nivel DC. A su vez, se puede visualizar la configuración del bloque "Delay" para lograr el desfasamiento de la señal, como se muestra a continuación



# Solución de la actividad "Modificación de las variables de una señal periódica"

The diagram shows two parallel signal paths. Each path starts with a 'Signal Source' block (Sample Rate: 128k, Waveform: Cosine, Frequency: 0, Amplitude: 1, Offset: 0), followed by a 'Delay' block (Delay: 0), and then a 'Throttle' block (Sample Rate: 128k). A callout box points to the first 'Delay' block with the text: 'Delay retrasa la entrada por un cierto número de muestras'. Another callout box points to the second 'Delay' block with the text: 'Los retrasos positivos insertan cero elementos al comienzo de la secuencia. Los retrasos negativos descartan elementos en la secuencia'.

The 'Properties: Delay' dialog is open on the right. It shows the following settings:

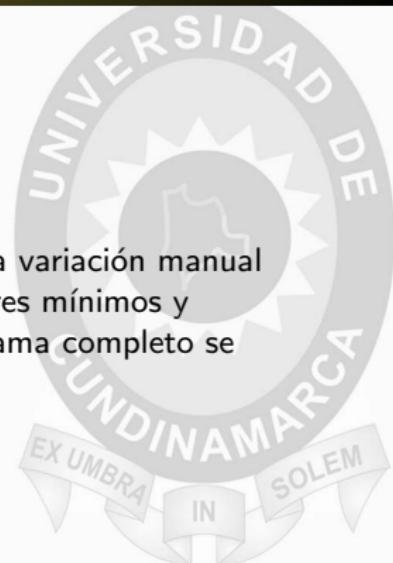
ID	blocks_delay_0
Type	Int
Delay	Fase
Num Ports	1
Vec Length	1

Callout boxes point from the dialog to the text: 'Tipo entero' (Type integer), 'Variable' (Variable), and 'Cantidad de entradas' (Number of inputs).

Buttons at the bottom of the dialog are: Aceptar (Accept), Cancelar (Cancel), and Aplicar (Apply).

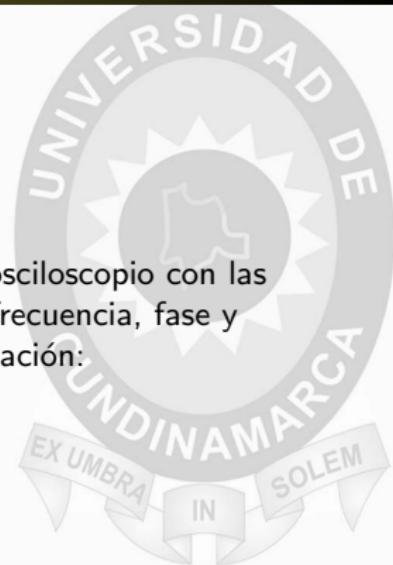
# Solución de la actividad "Modificación de las variables de una señal periódica"

2. Luego añadimos un WX GUI Slider para determinar la variación manual de la fase, donde se establece el ID de fase y los valores mínimos y máximos que puede tomar el retraso en fase, el diagrama completo se muestra a continuación:

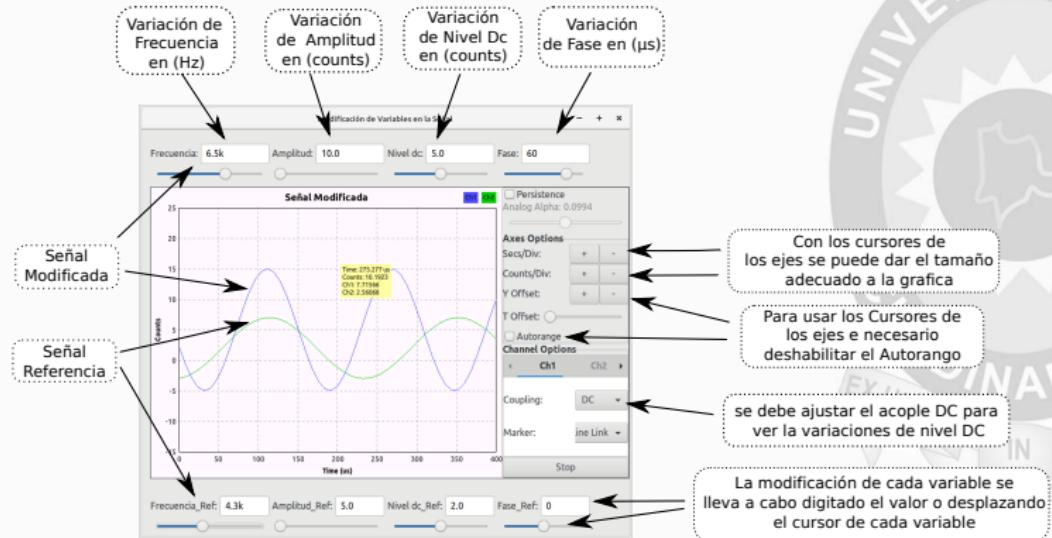


# Solución de la actividad "Modificación de las variables de una señal periódica"

- 3 Lo que obtendremos al compilar el programa será el osciloscopio con las opciones adicionales de fijar los valores de amplitud, frecuencia, fase y nivel DC de la señal, tal y como se muestra a continuación:



# Solución de la actividad "Modificación de las variables de una señal periódica"



*solucion lab 2 actividad 1*

## Respuesta Actividad 1 lab 2

**El ruido Gaussiano** se encuentra asociado con la radiación electromagnética. Ya que no podemos tener comunicación eléctrica sin electrones es imposible evitar el ruido, el ruido Gaussiano muestra una densidad de probabilidad que responde a una distribución normal(o distribución de Gauss).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

**la distribución uniforme** continua es una familia de distribuciones de probabilidad para variables aleatorias continuas, tales que para cada miembro de la familia, todos los intervalos de igual longitud en la distribución en su rango son igualmente probables. El dominio está definido por dos parámetros,  $a$  y  $b$ , que son sus valores mínimo y máximo. La distribución es a menudo escrita en forma abreviada como

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in (a, b) \\ 0 & \text{si } x \notin (a, b) \end{cases}$$

## Respuesta Actividad 1 lab 2

**distribución de Laplace** es una densidad de probabilidad continua, llamada así en honor a Pierre-Simon Laplace. Es también conocida como distribución doble exponencial puesto que puede ser considerada como la relación las densidades de dos distribuciones exponenciales adyacentes. La distribución de Laplace resulta de la diferencia de dos variables exponenciales aleatorias, independientes e idénticamente distribuidas

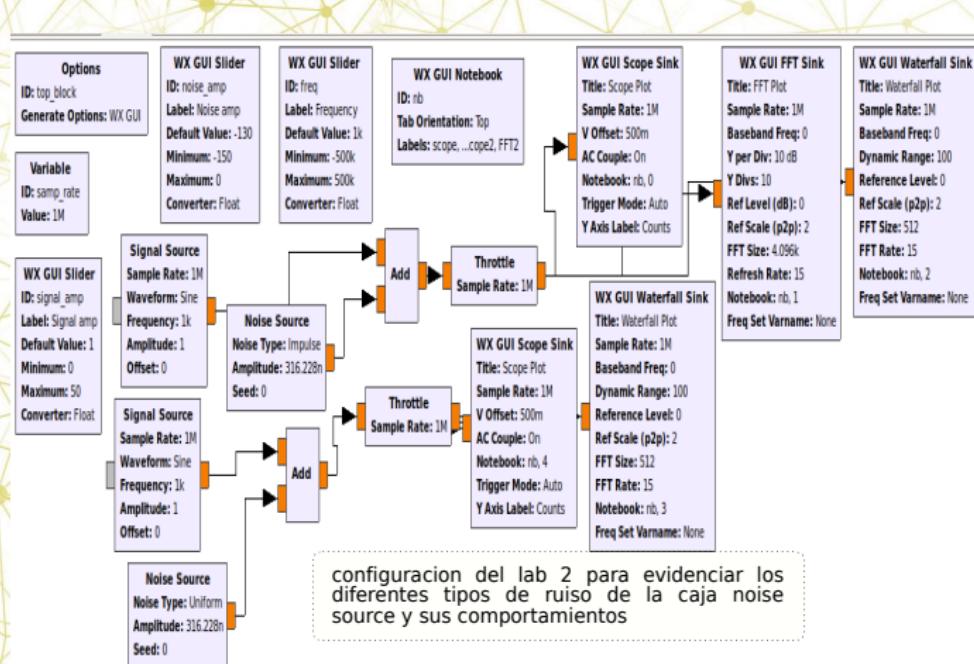
$$f(x) = \begin{cases} e^{(-\frac{\mu-x}{b})} & \text{si } x < \mu \\ e^{(-\frac{x-\mu}{b})} & \text{si } x \geq \mu \end{cases}$$

**El ruido impulsivo** es aquel ruido cuya intensidad aumenta bruscamente durante un impulso. La duración de este impulso es breve en comparación con el tiempo que transcurre entre un impulso y otro. Incide fundamentalmente en la transmisión de los datos, se debe básicamente a fuertes inducciones consecuencias de conmutaciones electromagnéticas

$$\delta(x - c) = \begin{cases} \infty & \text{si } x = c \\ 0 & \text{si } x \neq c \end{cases}$$

*solucion lab 2 actividad 2*

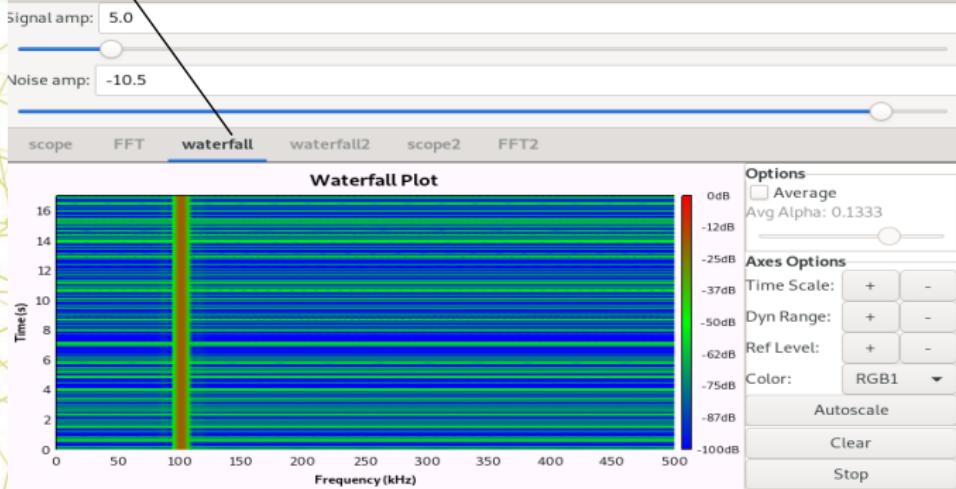
## Respuesta actividad 2 lab 2



## Respuesta actividad 2 lab 2

señal de 100k con ruido de impulso en waterfall 1 se pueden apreciar los impulsos de ruido generados por el bloque noise con valores de ruido cercanos a los -100 dB.

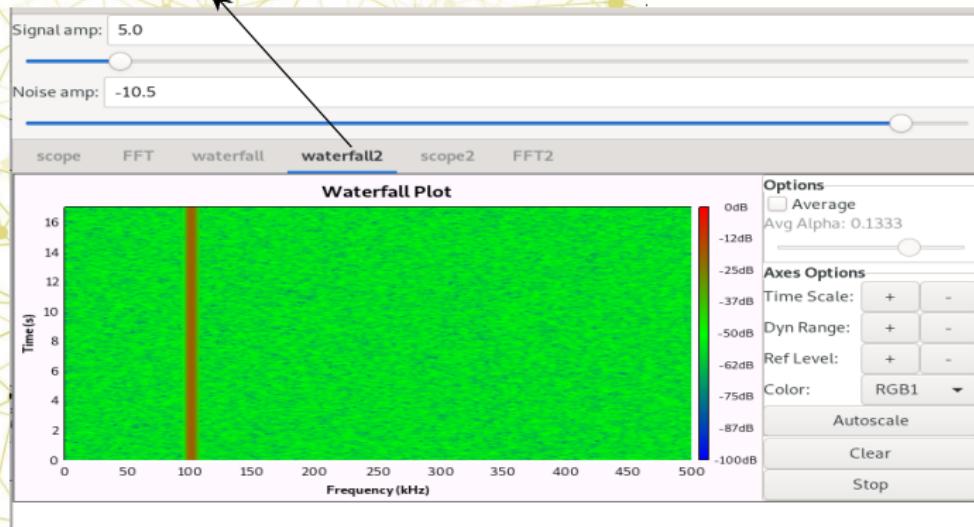
para observar los diferentes tipos de ruido se requiere la misma configuración en los parámetros como se muestra en la imagen.



## Respuesta actividad 2 lab 2

señal de 100k con ruido de tipo uniforme en waterfall 2 se pueden apreciar los impulsos de ruido generados por el bloque noise en una distribución más homogénea con rangos de ruido entre los -30dB

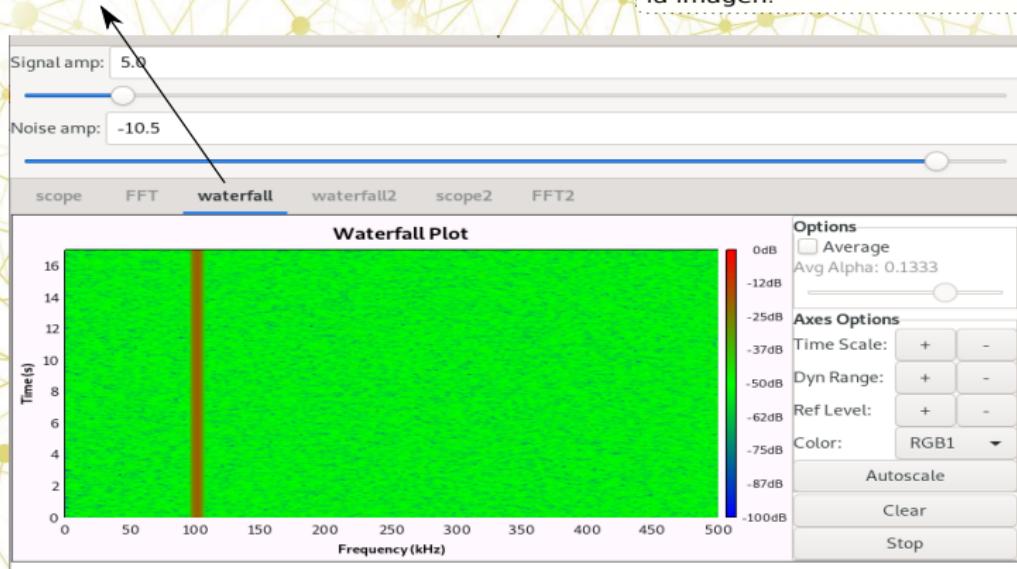
para observar los diferentes tipos de ruido se requiere la misma configuración en los parámetros como se muestra en la imagen.



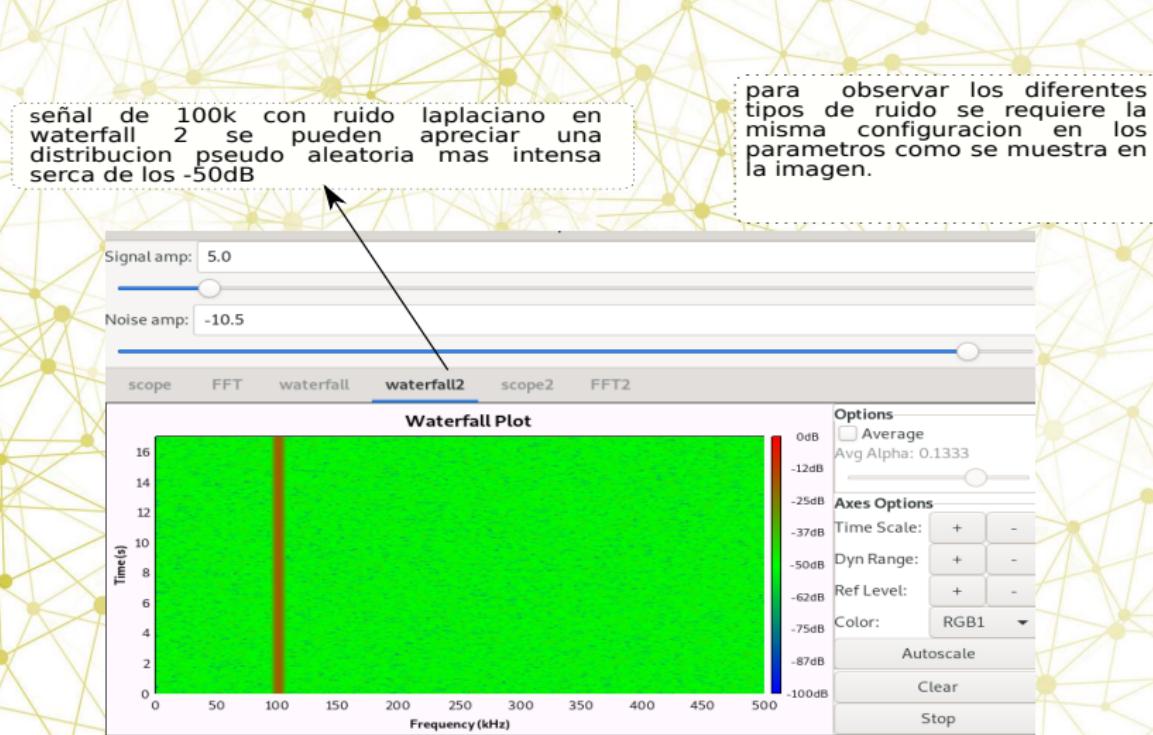
## Respuesta actividad 2 lab 2

señal de 100k con ruido gausiano en waterfall 1 se pueden apreciar los impulsos de ruido generados por el bloque noise en una distribución pseudo aleatoria entre los -37db

para observar los diferentes tipos de ruido se requiere la misma configuración en los parámetros como se muestra en la imagen.



## Respuesta actividad 2 lab 2



*solucion lab 2 actividad 3*

## Respuesta Actividad 3 lab 2

**Curvas de Lissajous** Las curvas de Lissajous fueron descubiertas por el físico francés Julio Antoine Lissajous (1822 a 1880). Fueron utilizadas para determinar las frecuencias de sonidos o de señales de radio, ademas permiten el estudio de los movimientos vibratorios y, particularmente, la comparación de los sonidos dados por dos instrumentos.

# Respuesta Actividad 3 lab 2

Iniciar las variables y las opciones con las que se va a trabajar en este esquema.

A continuación se incluyen los bloques XW GUI Chosser en los cuales de dará la información para generar la imagen dependiendo de la fase que sea escogida, creando botones específicos.

Options  
ID: lab2aC3  
Generate Options: WX GUI

Variable  
ID: samp\_rate  
Value: 3M

WX GUI Chooser  
ID: lab2aC3  
Labels: Fase  
Default Value: 0  
Choices: 250, 10., 100, 250  
Labels:  
Type: Radio Buttons  
Grid Position: 0, 4, 1, 5

Variable  
ID: variable  
Value: 72k

Signal Source  
Sample Rate: 3M  
Waveform: Cosine  
Frequency: 1K  
Amplitude: 500m  
Offset: 0

Signal Source  
Sample Rate: 3M  
Waveform: Cosine  
Frequency: 3K  
Amplitude: 500m  
Offset: 0

Signal Source  
Sample Rate: 3M  
Waveform: Cosine  
Frequency: 2K  
Amplitude: 500m  
Offset: 0

Signal Source  
Sample Rate: 3M  
Waveform: Cosine  
Frequency: 1K  
Amplitude: 500m  
Offset: 0

Signal Source  
Sample Rate: 72k  
Waveform: Cosine  
Frequency: 1K  
Amplitude: 500m  
Offset: 0

Signal Source  
Sample Rate: 72k  
Waveform: Cosine  
Frequency: 3K  
Amplitude: 500m  
Offset: 0

Delay  
Delay: 1

Throttle  
Sample Rate: 3M

Throttle  
Sample Rate: 3M

Throttle  
Sample Rate: 3M

Throttle  
Sample Rate: 3M

Throttle  
Sample Rate: 72k

WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 1M  
XY Mode: On  
Notebook: On  
Trigger Mode: Auto  
Y Axis Label: Counts

WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 1M  
V Scale: 50m  
XY Mode: On  
Notebook: notebook\_0, 0  
Trigger Mode: Auto  
Y Axis Label: Counts

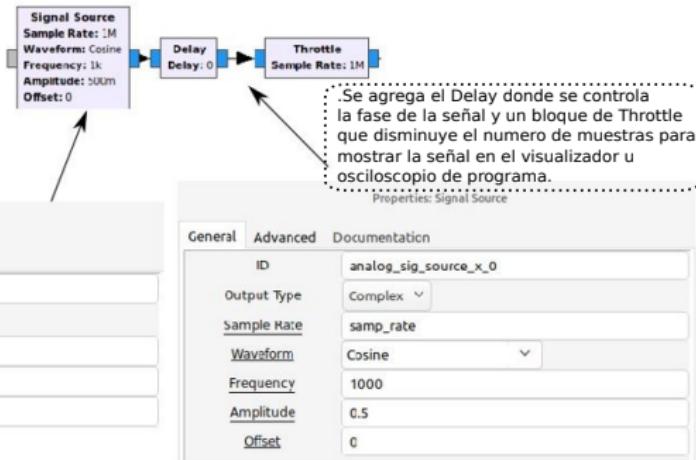
WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 1M  
V Scale: 10m  
XY Mode: On  
Notebook: notebook\_0, 1  
Trigger Mode: Auto  
Y Axis Label: Counts

WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 1M  
V Scale: 50m  
XY Mode: On  
Notebook: notebook\_0, 2  
Trigger Mode: Auto  
Y Axis Label: Counts

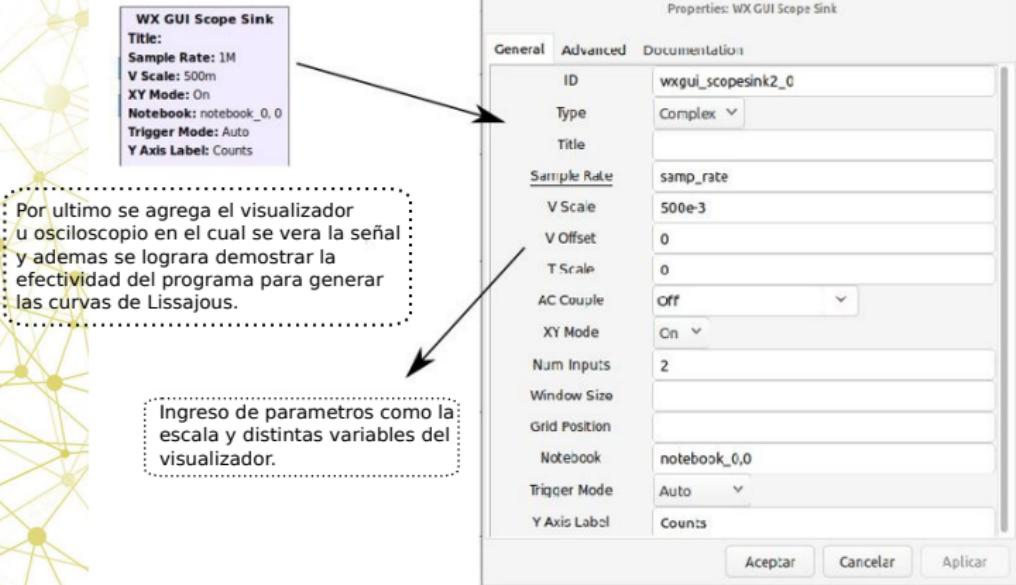
WX GUI Scope Sink  
Title: Scope Plot  
Sample Rate: 72k  
V Scale: 50m  
XY Mode: On  
Notebook: notebook\_0, 3  
Trigger Mode: Auto  
Y Axis Label: Counts

## Respuesta Actividad 3 lab 2

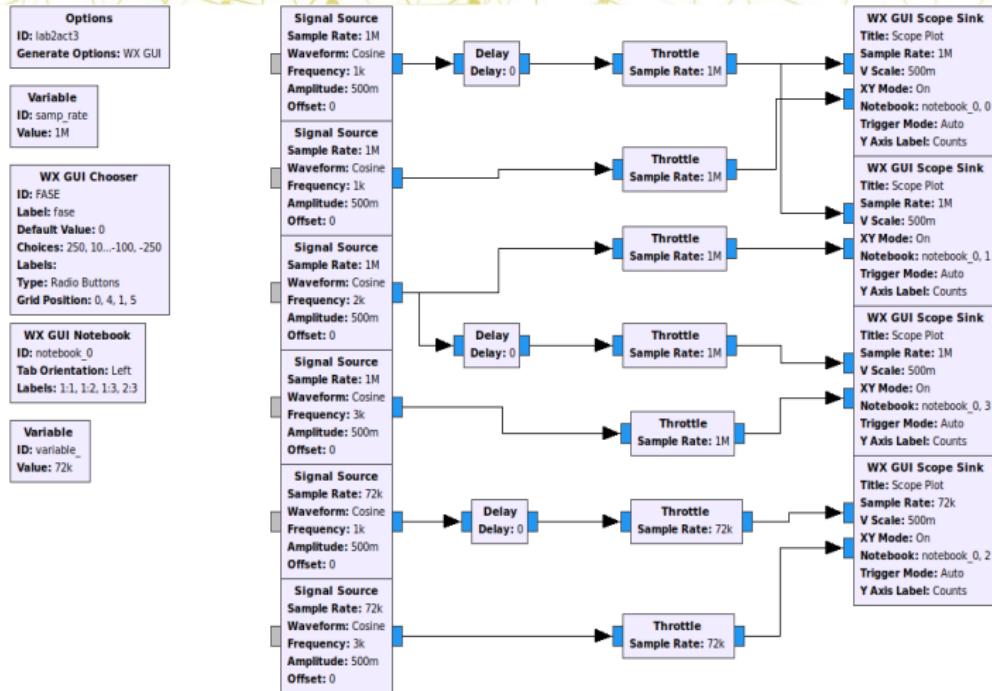
se agregan las cajas de Signal Source en las cuales se dará la información de la señal y del tipo de señal a generar, ademas de los parámetros de frecuencia min y max de la misma.



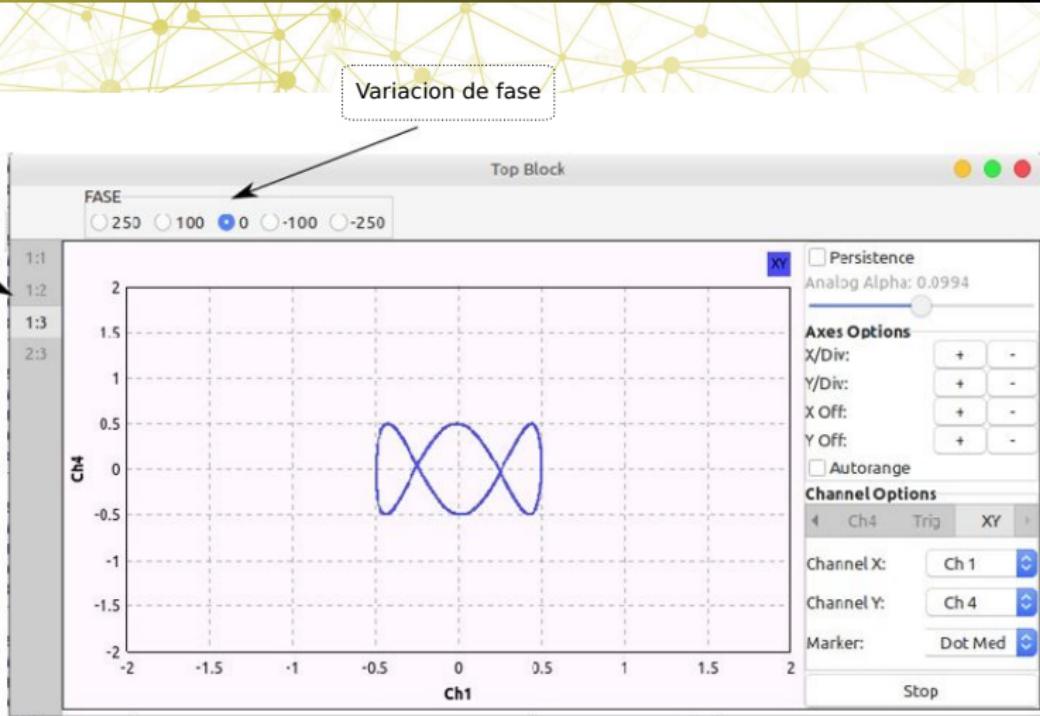
## Respuesta Actividad 3 lab 2



# Respuesta Actividad 3 lab 2



## Respuesta Actividad 3 lab 2



*solucion lab 4 actividad 1*

# "¿Como observar la constelación de una modulación ASK?"

En estos tipos de modulación digital se pueden observar los diagramas de constelación de la siguiente manera:

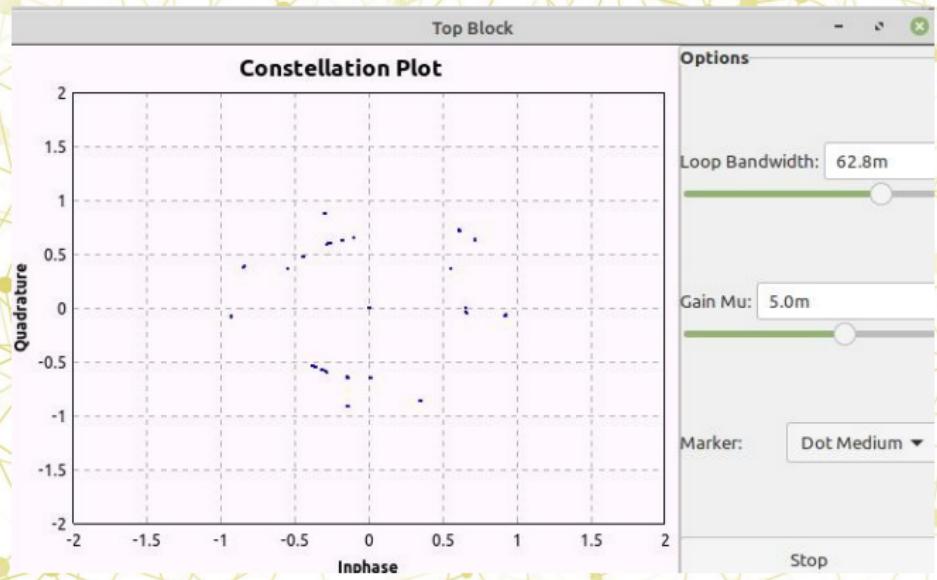
- Modulación ASK
- Modulación FSK
- Modulación PSK

## "Tipos de Modulaciones ASK"

- La Modulación por desplazamiento de amplitud (ASK) es un esquema de modulación digital en el que la amplitud de la onda portadora se cambia con respecto a la señal de información, manteniendo la fase y la frecuencia de constantes.
- La modulación por desplazamiento de frecuencia (FSK) es un tipo de modulación de frecuencia cuya señal modulante es un flujo de pulsos binarios que varía entre valores predeterminados.
- La modulación por desplazamiento de fase (PSK) es un tipo de modulación angular que consiste en hacer variar la fase de la portadora entre un número de valores discretos.

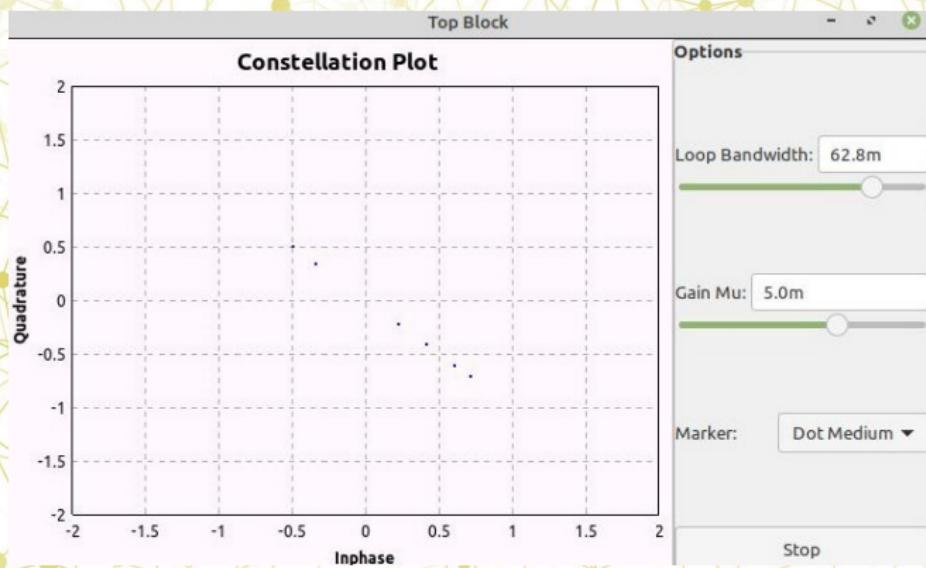
# "Constelación PSK"

Conectamos el wx GUI constellation Sink a la modulacion ASK Y FSK obtendremos la siguiente grafica



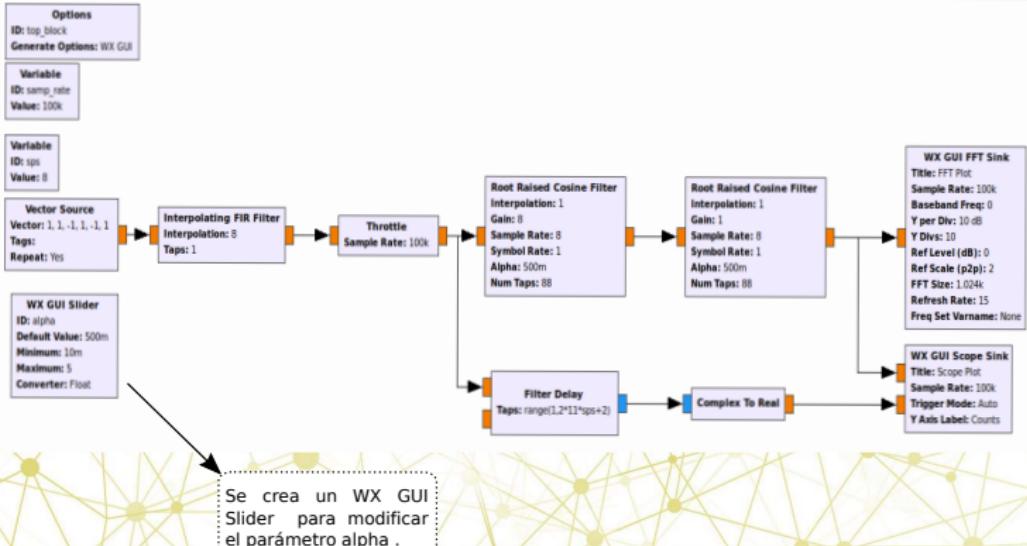
# "Constelación ASK Y FSK"

Conectamos el wx GUI constellation Sink a la modulacion PSK obtendremos la siguiente grafica

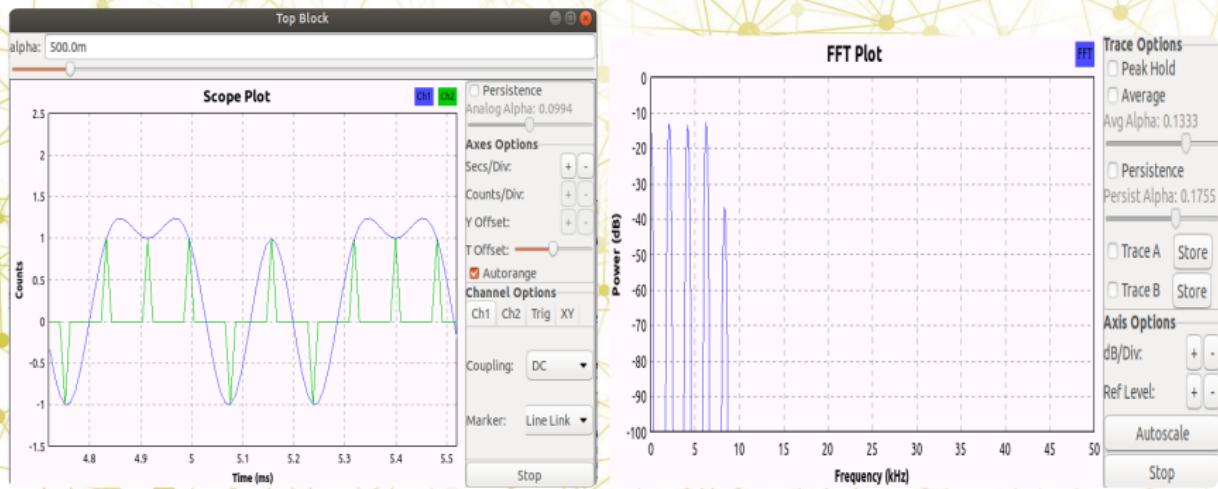


*solucion lab 5 actividad 1*

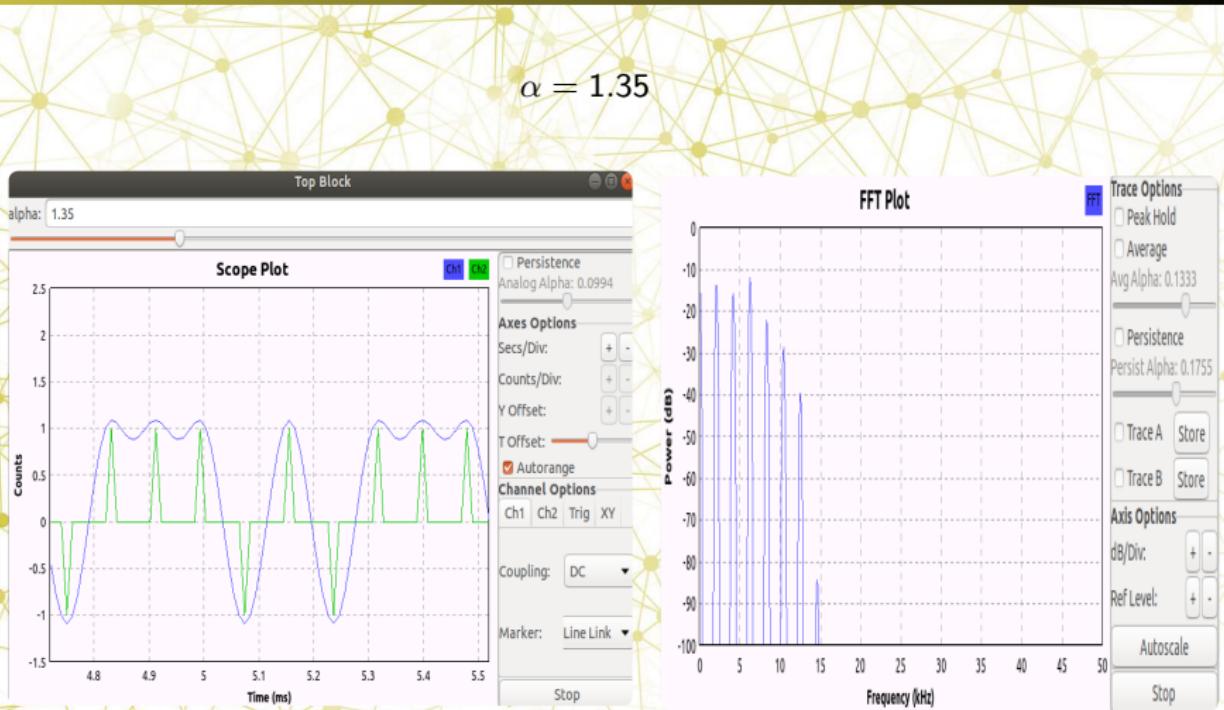
# Solución de la actividad "Modificación del parámetro alpha en un filtro de coseno realizado"



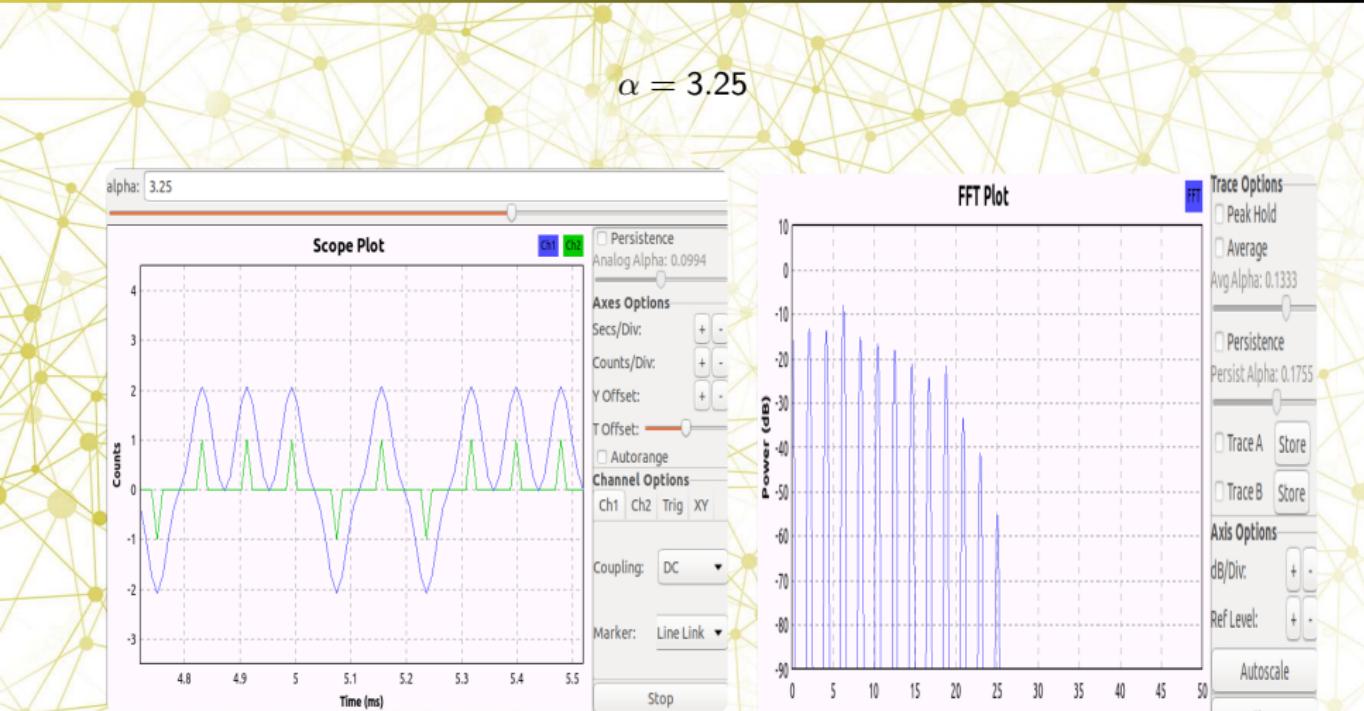
# Solución de la actividad "Modificación del parámetro alpha en un filtro de coseno realizado"



# Solución de la actividad "Modificación del parámetro alpha en un filtro de coseno realizado"



# Solución de la actividad "Modificación del parámetro alpha en un filtro de coseno realizado"



# Solución de la actividad "Modificación del parámetro alpha en un filtro de coseno realizado"

Como se puede observar , cuando el valor de  $\alpha$  sobrepasa 1, la potencia de las bandas laterales supera la de la portadora , causando que la señal aumente la amplitud y la forma de esta se parezca cada vez mas al tren de pulsos representado por el canal 2. Así mismo si se aumenta mucho el parámetro  $\alpha$  se ve afectado el ancho de banda puesto que este aumenta.