

ESTUDIO DE LA TECNOLOGÍA RADIO DEFINIDO POR SOFTWARE

SANDOVAL ROMERO DANNY FABIAN  
ZAMORA JORGE ANDRES

UNIVERSITARIA AGUSTINIANA  
FACULTAD DE INGENIERIAS  
PROGRAMA DE INGENIERIA EN TELECOMUNICACIONES  
BOGOTÁ D.C.  
2017

ESTUDIO DE LA TECNOLOGÍA RADIO DEFINIDO POR SOFTWARE

SANDOVAL ROMERO DANNY FABIAN  
ZAMORA JORGE ANDRES

Asesor de trabajo  
GOMEZ CARLOS ANDRES

Trabajo de grado para optar por el título de profesional en Ingeniería en Telecomunicaciones

UNIVERSITARIA AGUSTINIANA  
FACULTAD DE INGENIERIAS  
PROGRAMA DE INGENIERIA EN TELECOMUNICACIONES  
BOGOTÁ D.C.  
2017

**NOTA DE ACEPTACIÓN**

---

---

---

---

---

---

**PRESIDENTE DEL JURADO**

---

**FIRMA DEL JURADO**

---

**FIRMA DEL JURADO**

**Bogotá D.C., 24 de noviembre de 2017**

## TABLA DE CONTENIDO

TABLA de contenido .....	iv
lista de FIGURAS .....	xiii
LISTA DE TABLAS.....	xx
RESUMEN.....	19
introducción.....	20
1. problema de investigación .....	21
1.1. Tema de Investigación .....	21
1.2. Planteamiento del Problema .....	21
2. Justificación .....	22
3. objetivos.....	23
3.1. obgetivo general.....	23
3.2. objetivos específicos .....	23
4. Marco de referencia .....	24
4.1. Marco teórico.....	24
4.2. MARCO CONCEPTUAL .....	29
4.2.1. Ondas de radio: .....	29
4.2.2. Radio definida por software (SDR) .....	29
4.2.3. Procesamiento Digital de Señal (DSP) .....	30
4.2.4. GNU Radio .....	31
4.2.5. Modulación y Demodulación.....	32
4.3. Marco legal .....	34

4.3.1.	Ley 1341 de 2009 (Julio 30).....	34
4.3.2.	Decreto 93 de 2010 (Enero 19).....	34
4.3.3.	Decreto 4169 de 2011 (Noviembre 3) .....	34
4.3.4.	Ley 1507 de 2012 (Enero 10) .....	35
4.3.5.	Decreto 2453 de 2013 (07 de Noviembre).....	35
5.	metodología del proyecto.....	37
5.1.	enfoque cuantitativo.....	37
6.	Administración de proyecto.....	38
6.1.	Cronograma.....	38
6.2.	Presupuesto .....	39
6.2.1.	Presupuesto de personal.....	39
6.2.2.	Presupuesto de equipos .....	39
6.2.3.	Presupuesto de software.....	39
6.2.4.	Presupuesto de materiales y suministros.....	39
6.2.5.	Presupuesto de eventos científicos y salidas de campo .....	40
6.2.6.	Presupuesto de material bibliográfico.....	40
6.2.7.	Presupuesto de publicaciones .....	40
6.2.8.	Presupuesto de servicio técnico .....	40
6.2.9.	Presupuesto general .....	41
7.	desarrollo de problema de investigación.....	42
7.1.	Estudio de los componentes de un hardware SDR .....	42

7.1.1.	QUE ES EL ESPECTRO RADIOELÉCTRICO .....	42
7.1.2.	Que es DAC-ADC?.....	43
7.1.3.	CONVERTIDOR DIGITAL A ANALOGICO (DAC) CARACTERISTICAS DE FUNCIONAMIENTO.....	44
7.1.4.	CONVERTIDOR EN ESCALERA PONDERADO R/2R (WEIGHTED RESISTOR LADDER) .....	45
7.1.4.1.	Escala completa de salida (Full-Scale Output (FSO)). .....	45
7.1.4.2.	Resolución (Resolution):.....	46
7.1.4.3.	Precisión:.....	46
7.1.4.4.	Linealidad:.....	46
7.1.4.5.	Monotonicidad: .....	46
7.1.5.	CONVERTIDORES ANALÓGICOS A DIGITAL (ADC) MODOS DE FUNCIONAMIENTO. ....	47
7.1.6.	QUE ES LA FRECUENCIA INTERMEDIA (IF) .....	47
7.1.7.	PROCESO DIGITAL DE SEÑAL DSP.....	49
7.1.8.	APLICACIONES.....	50
7.1.9.	CARACTERÍSTICAS DE LOS DSP.....	51
7.1.10.	QUE ES FFT.....	52
7.1.11.	LA RADIO DEFINIDA POR SOFTWARE (SDR).....	55
7.1.12.	COMO FUNCIONA SDR.....	56

7.1.12.1.	Estructura del sistema SDR.....	56
7.1.13.	CONCEPTO DE RADIO DEFINIDO POR SOFTWARE SDR .....	57
7.1.14.	NIVELES DE SDR.....	58
7.1.15.	EVOLUCION DEL SDR.....	60
7.1.15.1.	La evolución de la arquitectura de radio definida por software.....	60
7.1.16.	LA ÚLTIMA ARQUITECTURA SDR.....	60
7.1.17.	ARQUITECTURAS SDR .....	62
7.1.17.1.	La arquitectura RTL-SDR.....	64
7.1.18.	RTL-SDR HARDWARE.....	65
7.1.19.	OPERACIÓN DEL DISPOSITIVO .....	65
7.1.20.	QUE ES UN USRP.....	67
7.1.21.	APLICACIONES DE SDR.....	68
7.1.22.	SDR IMPLEMENTADOS .....	70
7.1.22.1.	Funcube dongle .....	70
7.1.22.2.	PM-SDR.....	71
7.1.22.3.	Finningley 80M.....	71
7.1.22.4.	Excalibur .....	72
7.1.22.5.	Universal Software Radio Reripheral (USRP).....	72
7.1.22.6.	QS1R.....	73
7.2.	Estudio de los Componentes GNU Radio Companion.....	74

8.	realización de practicas de laboratorio con sus guias .....	76
8.1.	Laboratorio 1: Construcción de un receptor de radio FM en GNU Radio Companion. 76	
8.2.	Laboratorio 2: Procesamiento de señales digitales, DSP.....	116
9.	Conclusiones.....	137
10.	Referencias.....	138
11.	ANEXOS .....	143
11.1.	Anexo 1: Documentación de software GNU Radio companion.....	143
11.1.1.	Options (Bloque de Opciones del Workspace).....	145
11.1.2.	Level Controllers (Controladores de nivel) .....	146
11.1.2.1.	AGC .....	146
11.1.2.2.	Feed Forward AGC .....	148
11.1.2.3.	Power Squelch (Silenciador de energía) .....	149
11.1.2.4.	Rail.....	151
11.1.2.5.	Mute .....	152
11.1.2.6.	Sample and Hold (Muestreo y retención) .....	154
11.1.2.7.	Moving Average (Media móvil) .....	155
11.1.2.8.	Threshold (Límite) .....	156
11.1.3.	Waveform Generators (Generadores de forma de onda).....	157
11.1.3.1.	Signal Source (Fuente de señal).....	157
11.1.3.2.	Constant Source (Fuente Constante).....	159

11.1.3.3.	Noise Source (Fuente de ruido).....	160
11.1.3.4.	Random Source (Fuente aleatoria).....	162
11.1.3.5.	VCO .....	163
11.1.4.	Misc (Miscelánea).....	164
11.1.4.1.	Virtual Source (Fuente Virtual) .....	164
11.1.4.2.	Virtual Sink (Receptor Virtual).....	165
11.1.4.3.	Note (Nota).....	166
11.1.4.4.	Import (Importar) .....	167
11.1.4.5.	Selector.....	167
11.1.4.6.	Valve (Valvula).....	169
11.1.4.7.	Throttle (Acelerador) .....	170
11.1.4.8.	Delay (Retrasar) .....	171
11.1.4.9.	Null Source (fuente nula).....	172
11.1.4.10.	Null Sink .....	174
11.1.5.	Variables .....	175
11.1.5.1.	Variable .....	175
11.1.5.2.	Variable Config (Configuración de variables).....	175
11.1.5.3.	Parameter (Parámetro) .....	177
11.1.6.	Audio.....	178
11.1.6.1.	Wav File Source (Fuente de archivos Wav) .....	178

- 11.1.6.2. Wav File Sink (Sink de Archivos Waw)..... 180
- 11.1.6.3. Audio Sink ..... 181
- 11.1.6.4. Audio Source (Fuente de audio) ..... 182
- 11.1.7. Boolean Operators ..... 183
  - 11.1.7.1. And (Y) ..... 183
  - 11.1.7.2. And Const (Y Constante)..... 185
  - 11.1.7.3. Not (No) ..... 185
  - 11.1.7.4. Or (O)..... 186
  - 11.1.7.5. Xor..... 187
- 11.1.8. File Operators..... 188
  - 11.1.8.1. Wav File Source ..... 188
  - 11.1.8.2. Wav File Sink..... 189
  - 11.1.8.3. File Source (Origen del archivo)..... 189
  - 11.1.8.4. File Sink ..... 190
- 11.1.9. Math Operators (Operadores matemáticos) ..... 191
  - 11.1.9.1. Add (Adición) ..... 191
  - 11.1.9.2. Add Const (Adición Constante)..... 193
  - 11.1.9.3. Divide (Dividir)..... 194
  - 11.1.9.4. Multiply (Multiplicar) ..... 195
  - 11.1.9.5. Multiply Const (Multiplicar Constante)..... 196

11.1.9.6.	Subtract (Sustraer).....	197
11.1.10.	Message Tools (Herramientas de mensaje) .....	197
11.1.10.1.	Message Source (Fuente del Mensaje).....	197
11.1.10.2.	Message Sink .....	199
11.1.11.	Stream Operators (Operadores de flujo).....	199
11.1.11.1.	Deinterleave (Desintercalado).....	199
11.1.11.2.	Interleave (Intercalado) .....	200
11.1.11.3.	Repeat (Repetir) .....	202
11.1.12.	Type Converters.....	203
11.1.12.1.	Char To Float .....	203
11.1.12.2.	Char To Short.....	204
11.1.12.3.	Complex To Float .....	204
11.1.12.4.	Complex To Real .....	205
11.1.12.5.	Float To Int.....	206
11.1.12.6.	Float To Short .....	207
11.1.13.	Filters .....	209
11.1.13.1.	Low Pass Filter (Filtro de paso bajo) .....	209
11.1.13.2.	DC Blocker (Bloqueador de DC).....	211
11.1.13.3.	Hilbert .....	212
11.1.13.4.	Rational Resampler (Remuestrador Racional).....	213

11.1.13.5. Frequency Xlating FIR Filter (Filtro FIR de frecuencia Xlating)..... 214

## LISTA DE FIGURAS

Figura 1. Onda de radio (Comin1415cs, s. f.).....	29
Figura 2. SDR (AIRSPY, 2016) .....	30
Figura 3. DSP (Hernandez, 2004).....	31
Figura 4. Interacción niveles GNU Radio. (Pinar & Murillo, 2011) .....	32
Figura 5. Modulación AM y FM (Alfaro, 2011) .....	33
Figura 6. Bandas de Frecuencia .....	42
Figura 7. ADC, Y DAC (turmero, 2015).....	43
Figura 8. Muestreo de una señal digitalización (turmero, 2015) .....	44
Figura 9. Convertidor por Ponderación Binaria (Convertidores Digitales DAC y analogico a Digital ADC, 2017).....	44
Figura 10. Convertidor en escalera R/2R (Convertidores Digitales DAC y analogico a Digital ADC, 2017).....	45
Figura 11. Aproximaciones sucesivas (Convertidores Digitales DAC y analogico a Digital ADC, 2017) .....	47
Figura 12 Receptor de Radio tipo superheterodino con filtro de rechazo a la banda de frecuencia. (2012).....	48
Figura 13. Representaciones numéricas comunes en los DSP comerciales (Salazar).....	50
Figura 14. (En la Parte Superior) Arquitectura Von Neumann. (En la Parte Inferior) Arquitectura Harvard. (Salazar).....	51
Figura 15. Silbato A. (Nave, 2014).....	53
Figura 16. Silbato B (Nave, 2014) .....	54
Figura 17. Silbatos A y B (Nave, 2014).....	54
Figura 18. Diagrama en bloques de un Sistema digital de Comunicaciones (Garcia, 2011).....	56
Figura 19. Figura SDR (Mario, 2012) .....	59
Figura 20. Los componentes del SDR simple, conceptual, "Ultimate" que en teoría digitalizarán el espectro de 0Hz a 2GHz (o FS/2) (Bob Stewart, 2015).....	61
Figura 21. La evolución de SDR: a medida que el índice de muestreo del ADCS aumenta y se acercan más a la antena (a) la radio digital de banda base evoluciona a (b) la radio digital if, y luego a (c) la radio digital RF (cero-IF) (Bob Stewart, 2015) .....	64
Figura 22. (Bob Stewart, 2015).....	66

Figura 23. Diagrama de bloques destacando los procesos principales que en el TX lado de la USRP FPGA y RF placa. (Bob Stewart, 2015) .....	67
Figura 24. Aplicación Amateur SDR (Azurdía, 2016) .....	68
Figura 25. Implementación y Simulación de Llamadas Móviles. ....	69
Figura 26. Detención y Uso de las Señales GPS .....	70
Figura 27. Funcube Dongle (AMSAT-UK, 2011).....	70
Figura 28. PM-SDR. (Kanga., 2011) .....	71
Figura 29. FINNINGLEY 80M (Kanga., 2011) .....	71
Figura 30. Modelo Excalibur .....	72
Figura 31. Universal Software Radio Peripheral (USRP) (Blossom., 2011).....	72
Figura 32. Modelo QS1R (LLC., s.f.).....	73
Figura 33. Paso 0.....	77
Figura 34. Paso 0a.....	77
Figura 35. Paso 1.....	78
Figura 36. Paso 1a.....	78
Figura 37. Paso 2.....	79
Figura 38. Paso 2a.....	80
Figura 39. Paso 3.....	80
Figura 40. Paso 3a.....	81
Figura 41. Paso 3b.....	81
Figura 42. Paso 4.....	82
Figura 43. Paso 5.....	83
Figura 44. Paso 6.....	84
Figura 45. Paso 6a.....	84
Figura 46. Paso 6b.....	85
Figura 47. Paso 7.....	86
Figura 48. Paso 7a.....	86
Figura 49. Paso 8.....	87
Figura 50. Paso 8a.....	87
Figura 51. Paso 8.1.....	88
Figura 52. Paso 8.2.....	88

Figura 53. Paso 9.....	89
Figura 54. Paso 9.1.....	90
Figura 55. Paso 10.....	91
Figura 56. Paso 10a.....	91
Figura 57. Paso 10b.....	92
Figura 58. Paso 10c.....	92
Figura 59. Paso 11.....	93
Figura 60. Paso 12.....	94
Figura 61. Paso 12a.....	94
Figura 62. Paso 12b.....	95
Figura 63. Paso 12.1.....	95
Figura 64. Paso 12.2.....	96
Figura 65. Paso 13.....	97
Figura 66. Paso 13a.....	97
Figura 67. Paso 14.....	98
Figura 68. Paso 15.....	99
Figura 69. Paso 15.1.....	99
Figura 70. Paso 16.....	100
Figura 71. Paso 16a.....	101
Figura 72. Paso 17.....	102
Figura 73. Paso 17a.....	102
Figura 74. Paso 18.....	103
Figura 75. Paso 18a.....	104
Figura 76. Paso 18.1.....	105
Figura 77. Paso 18.1a.....	105
Figura 78. Paso 19.....	106
Figura 79. Paso 19a.....	106
Figura 80. Paso 19.1.....	107
Figura 81. Paso 20.....	108
Figura 82. Paso 20a.....	108
Figura 83. Paso 21.....	109

Figura 84. Paso 21.1.....	110
Figura 85. Paso extra 1.....	111
Figura 86. Paso extra 1a.....	111
Figura 87. Paso extra 1b.....	112
Figura 88. Paso extra 1.1.....	112
Figura 89. Paso extra 1.1a.....	113
Figura 90. Paso extra 2.....	114
Figura 91. Figura de ejemplo.....	115
Figura 92. 2lab Paso 0.....	117
Figura 93. 2lab Paso 1.....	118
Figura 94. 2lab Paso 1a.....	119
Figura 95. 2lab Paso 1b.....	119
Figura 96. 2lab Paso 1c.....	120
Figura 97. 2lab Paso 1d.....	120
Figura 98. 2lab Paso 2.....	121
Figura 99. 2lab Paso 3.....	122
Figura 100. 2lab Paso 3a.....	123
Figura 101. 2lab Paso 4.....	123
Figura 102. 2lab Paso 4a.....	124
Figura 103. 2lab Paso 4b.....	124
Figura 104. 2lab Paso 4c.....	125
Figura 105. 2lab Paso 4d.....	125
Figura 106. 2lab Paso 5.....	126
Figura 107. 2lab Paso 5a.....	126
Figura 108. 2lab Paso 6.....	127
Figura 109. 2lab Paso 6a.....	128
Figura 110. 2lab Paso 7a.....	129
Figura 111. 2lab Paso 7b.....	129
Figura 112. 2lab Paso 7c.....	130
Figura 113. 2lab Paso 7d.....	130
Figura 114. 2lab Paso 7e.....	131

Figura 115. 2lab Paso 7f. ....	131
Figura 116. 2lab Paso 7g.....	132
Figura 117. 2lab Paso 7h.....	132
Figura 118. 2lab Paso 7i.....	133
Figura 119. 2lab Paso 8.....	133
Figura 120. 2lab Paso 8a.....	134
Figura 121. . 2lab Paso 9.....	135
Figura 122. Propiedades bloque Options. (Prieto & Rafael, 2017) .....	146
Figura 123. Bloque AGC y sus propiedades. (Software GRC, 2017) .....	147
Figura 124. Bloque Feed Forward AGC y sus propiedades. (Software GRC, 2017) .....	149
Figura 125. Bloque Power Squelch y sus propiedades. (Software GRC, 2017).....	150
Figura 126. Tipos de datos para Type del bloque Power Squelch. (DocBook, 2013).....	150
Figura 127. Bloque Rail y sus propiedades. (Software GRC, 2017) .....	152
Figura 128. Bloque Mute y sus propiedades. (Software GRC, 2017) .....	153
Figura 129. Tipos de datos para IO Type del bloque Mute. (DocBook, 2013) .....	153
Figura 130. Bloque Sample and Hold y sus propiedades. (Software GRC, 2017).....	154
Figura 131. Bloque Moving Average y sus propiedades. (Software GRC, 2017).....	155
Figura 132. Bloque Threshold y sus propiedades. (Software GRC, 2017).....	156
Figura 133. Bloque Signal Source y sus propiedades. (Software GRC, 2017) .....	157
Figura 134. Tipos de datos para Output Type del bloque Signal Source. (DocBook, 2013) .....	158
Figura 135. Tipos de onda para el bloque Signal Source. (DocBook, 2013) .....	158
Figura 136. Bloque Constant Source y sus propiedades. (Software GRC, 2017) .....	159
Figura 137. Bloque Noise Source y sus propiedades. (Software GRC, 2017) .....	160
Figura 138. Tipos de datos para Output Type del bloque Noise Source. (DocBook, 2013) .....	161
Figura 139. Tipos de distribución estadística del ruido. (DocBook, 2013) .....	161
Figura 140. Bloque Random Source y sus propiedades. (Software GRC, 2017) .....	162
Figura 141. Bloque VCO y sus propiedades. (Software GRC, 2017) .....	163
Figura 142. Bloque Virtual Source y sus propiedades. (Software GRC, 2017) .....	164
Figura 143. Bloque Virtual Sink y sus propiedades. (Software GRC, 2017).....	165
Figura 144. Bloque Note y sus propiedades. (Software GRC, 2017).....	166
Figura 145. Bloque Import y sus propiedades. (Software GRC, 2017).....	167

Figura 146. Bloque Selector y sus propiedades. (Software GRC, 2017).....	168
Figura 147. Tipos de datos para el bloque Selector. (DocBook, 2013) .....	168
Figura 148. Bloque Valve y sus propiedades. (Software GRC, 2017) .....	169
Figura 149. Bloque Throttle de forma junto con sus propiedades. (Software GRC, 2017).....	170
Figura 150. Tipos de datos para el bloque Throttle. (DocBook, 2013) .....	171
Figura 151. Bloque Delay y sus propiedades. (Software GRC, 2017) .....	172
Figura 152. Bloque Null Source y sus propiedades. (Software GRC, 2017).....	173
Figura 153. Tipos de datos para el bloque Null Source. (DocBook, 2013).....	173
Figura 154. Bloque Null Sink y sus propiedades. (Software GRC, 2017) .....	174
Figura 155. Tipos de datos para el bloque Null Sink. (DocBook, 2013).....	174
Figura 156. Bloque Variable. (Software GRC, 2017) .....	175
Figura 157. Bloque Variable Config y sus propiedades. (Software GRC, 2017).....	176
Figura 158. Tipos de datos. (DocBook, 2013).....	176
Figura 159. Tipos de datos para Type en el bloque Parameter. (DocBook, 2013).....	178
Figura 160. Bloque Wav File Source y sus propiedades. (Software GRC, 2017).....	179
Figura 161. Bloque Wav File Sink y sus propiedades. (Software GRC, 2017).....	180
Figura 162. Bloque Audio Sink. (Software GRC, 2017).....	181
Figura 163. Bloque Audio Source. (Software GRC, 2017) .....	182
Figura 164. Figura de verdad Operación AND. (Electrontools, s.f.).....	183
Figura 165. Bloque AND y sus propiedades. (Software GRC, 2017) .....	184
Figura 166. Tipos de datos para IO Type. (DocBook, 2013) .....	184
Figura 167. Bloque And Const y sus propiedades. (Software GRC, 2017).....	185
Figura 168. Bloque Not y sus propiedades. (Software GRC, 2017).....	186
Figura 169. Bloque Or y sus propiedades. (Software GRC, 2017) .....	187
Figura 170. Bloque Xor y sus propiedades. (Software GRC, 2017).....	188
Figura 171. Bloque File Source y sus propiedades. (Software GRC, 2017) .....	189
Figura 172. Bloque File Sink y sus propiedades. (Software GRC, 2017) .....	190
Figura 173. Bloque Add y sus propiedades. (Software GRC, 2017).....	192
Figura 174. Tipos de datos para IO Type bloque Add. (DocBook, 2013).....	192
Figura 175. Bloque Add Const y sus propiedades. (Software GRC, 2017).....	193
Figura 176. Bloque Divide y sus propiedades. (Software GRC, 2017).....	194

Figura 177. Bloque Multiply y sus propiedades. (Software GRC, 2017).....	195
Figura 178. Bloque Multiply Const y sus propiedades. (Software GRC, 2017) .....	196
Figura 179. Bloque Subtrac. (Software GRC, 2017).....	197
Figura 180. Bloque Message Source y sus propiedades. (Software GRC, 2017).....	198
Figura 181. Tipos de datos para Output Type del bloque Message Source. (DocBook, 2013)..	198
Figura 182. Tipos de datos para Input Type del bloque Message Sink. (DocBook, 2013) .....	199
Figura 183. Bloque Deinterleave y sus propiedades. (Software GRC, 2017) .....	200
Figura 184. Bloque Interleave y sus propiedades. (Software GRC, 2017).....	201
Figura 185. Bloque Repeat y sus propiedades. (Software GRC, 2017).....	202
Figura 186. Bloque Char To Float y sus propiedades. (Software GRC, 2017) .....	203
Figura 187. Bloque Char To Short y sus propiedades. (Software GRC, 2017).....	204
Figura 188. Bloque Complex To Float y sus propiedades. (Software GRC, 2017) .....	205
Figura 189. Bloque Complex To Real y sus propiedades. (Software GRC, 2017) .....	206
Figura 190. Bloque Float To Int y sus propiedades. (Software GRC, 2017).....	207
Figura 191. Bloque Float To Short y sus propiedades. (Software GRC, 2017) .....	208
Figura 192. Bloque Low Pass Filter y sus propiedades. (Software GRC, 2017).....	209
Figura 193. Tipos de datos para FIR Type del bloque Low Pass Filter. (DocBook, 2013).....	210
Figura 194. Bloque DC Blocker y sus propiedades. (Software GRC, 2017).....	211
Figura 195. Tipos de datos para Type del bloque DC Blocker. (DocBook, 2013).....	211
Figura 196. Bloque Hilbert y sus propiedades. (Software GRC, 2017) .....	212
Figura 197. Bloque Rational Resampler y sus propiedades. (Software GRC, 2017) .....	213
Figura 198. Tipos de datos para Type del bloque Rational Resampler. (DocBook, 2013) .....	214
Figura 199. Tipos de datos para Type del bloque Frequency Xlating FIR Filter. (DocBook, 2013) .....	215

**LISTA DE TABLAS**

Tabla 1. Cronograma de trabajo.....	38
Tabla 2. Presupuesto de personal.....	39
Tabla 3. Presupuesto de equipos.....	39
Tabla 4. Presupuesto de software.....	39
Tabla 5. Presupuesto de materiales y suministros. ....	39
Tabla 6. Presupuesto de eventos científicos y salidas de campo. ....	40
Tabla 7. Presupuesto de material bibliográfico.....	40
Tabla 8. Presupuesto de publicaciones. ....	40
Tabla 9. Presupuesto de servicio técnico. ....	40
Tabla 10. Presupuesto general. ....	41

## **RESUMEN**

Este trabajo presenta un estudio sobre la tecnología Software Definida por Radio (SDR por su siglas en inglés); se aborda el tema desde sus orígenes, documentando las etapas de desarrollo de la tecnología, luego se estudia el funcionamiento propio de este tipo de equipos. Se realiza un estudio profundo sobre las herramientas de desarrollo por software de funciones sobre equipos SDR, en particular el entorno de desarrollo denominado GCR de GNU Radio, y se finaliza realizando 2 talleres prácticos documentados sobre aplicaciones de SDR.

## INTRODUCCIÓN

La nueva ola de tecnologías inalámbricas ha permitido que la sociedad a nivel mundial se actualice y genere nuevas necesidades, las cuales se dieron inicio en la radiodifusión hasta la necesidad de comunicarse mediante una terminal móvil. El nacimiento de la tecnología SDR obedece a la necesidad de dar mayor flexibilidad a los sistemas de comunicaciones para adoptar hardware de capacidades genéricas y que sea el software quien personalice el uso específico de la tecnología.

De igual manera es de tener en cuenta que la SDR es una tecnología emergente particularmente dentro de la industria inalámbrica que tiene gran potencial para ser utilizado entre amplia gama de usos. En términos de requisitos de comunicación presenta soluciones económicas para los diferentes sectores como militar, civil y comercial. El factor que se hace más evidente en el uso del equipo de radio definido por software, es la capacidad que tiene para desarrollar diferentes escenarios que se contemplan sobre el espectro radioeléctrico.

Es presente proyecto aborda un estudio sobre la tecnología SDR en su evolución histórica y en las herramientas y capacidades actuales. Se aborda una perspectiva práctica en la que se da a conocer mediante dos Talleres de laboratorio que parte del hardware puede ser reconfigurado por software, logrando de esta manera abarcar la descripción de la plataforma que utiliza la tecnología SDR.

## **1. PROBLEMA DE INVESTIGACIÓN**

### **1.1. TEMA DE INVESTIGACIÓN**

Estudio de funcionamiento de la plataforma SDR con las tarjetas HackRF, con el fin de identificar y documentar fortalezas para el aprendizaje de las telecomunicaciones.

### **1.2. PLANTEAMIENTO DEL PROBLEMA**

Las Radios Definidas por Software (SDR) son la respuesta tecnológica a las nuevas necesidades de los sistemas de comunicación (velocidad, fiabilidad, robustez, compatibilidad... etc.)

Cuando se habla de comunicaciones inalámbricas hay una gran diversidad de tecnologías y protocolos, los cuales definen las características de un sistema de comunicaciones en particular. Ingeniosamente, dichas tecnologías eran implementadas en sistemas de comunicación fundamentados solamente en hardware con un propósito específico y ya una vez construidos y desarrollados su funcionalidad permanecía fija y no podía ser modificada fácilmente.

La ventaja que ofrece la plataforma SDR es la capacidad de implementar distintos tipos de tecnologías en un mismo dispositivo, disminuyendo así el costo de implementación. Debido a que la tecnología SDR se encuentra en pleno desarrollo, es necesario el estudio e implementación de distintas tecnologías de comunicación inalámbrica para poder evaluar plenamente el desempeño de una plataforma; de igual manera “las Tarjetas HACK RF tiene como finalidad manejar todo tipo de señales de radio comprendidas entre 30 MHz y 6 GHz desde un mismo periférico conectable al ordenador a través de un puerto USB” según como lo manifiesta (RODRÍGUEZ, 2013).

A pesar de la gran herramienta que significa el SDR, no hay suficiente documentación que compile el desarrollo histórico, con el uso de herramientas actuales, prácticas, y que se encuentre en idioma español.

## 2. JUSTIFICACIÓN

La finalidad de este proyecto tiene como base la creciente importancia de SDR como herramienta de seguridad el cual se refleja en sus ventajas al momento de fortalecer los protocolos de telecomunicaciones inalámbricas y métodos de autenticación que en ellas se presentan, pensando en los posibles ataques que pueden ser llevados a cabo e incluyéndolos en la recopilación de pruebas en procesos de pentesting.

Las plataformas que se encuentran a disposición al momento de ejecutar SDR varían en múltiples factores, como: “rangos de frecuencia, ancho de banda, interfaces disponibles, capacidad de recepción/transmisión, o precisión del oscilador, utilizando equipos de recepción y transmisión de señales, o mediante análisis de ondas con procesadores exclusivos que permiten el ahorro en costo de procesamiento” (BILIĆ, 2014).

HackRF es un proyecto reciente el cual crear una plataforma de código abierto sobre la que implementar un potente sistema de SDR que funcione y sea capaz de manejar gran variedad de señales de radio comprendidas entre 30 MHz y 6 GHz desde un mismo periférico conectable al ordenador a través de un puerto USB.

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Realizar un estudio de la tecnología Radio Definido por Software SDR para identificar su evolución y documentar su uso

#### **3.2. OBJETIVOS ESPECÍFICOS**

- Documentar la evolución de la tecnología SDR y el funcionamiento actual de estos dispositivos.
- Documentar el entorno de desarrollo GRC para la programación de aplicaciones SDR.
- Realizar y documentar prácticas de laboratorio de aplicaciones prácticas de equipos SDR.

## 4. MARCO DE REFERENCIA

### 4.1. MARCO TEÓRICO

Iiehara K. et al, (2002) presenta un trabajo realizado en Japón y trata del diseño de un prototipo de SDR basado en una arquitectura de multiprocesador (WA) para el sistema telefónico personal japonés (PHS) de un sistema móvil 2G y una LAN inalámbrica IEEE802.11, que tiene un ancho de banda mucho más amplio que los sistemas 2G el cual se implementa con éxito. Este diseño evalúa el rendimiento del prototipo con experimentos que incluyen la carga del procesador PHS y las características de retardo de procesamiento y las características de carga y procesamiento del procesador de LAN inalámbrica. Los autores lograron definir que una plataforma de SDR de banda ancha (más de 20 MHz) y altamente flexible, mediante el desarrollo de pre / post-procesadores de tasa flexible (FR-PPP) hacen posible el soporte de sistemas LAN inalámbricos que utilizan el esquema de secuencia directa, espectro ensanchado (DSSS). También lograron demostrar que el SDR permitirá que un solo terminal cubra sistemas móviles de segunda y tercera generación, así como sistemas inalámbricos de mayor velocidad y ancho de banda, como las LAN inalámbricas IEEE802.11, debido a que la tasa de chip de WCDMA de la red 3G Móvil es aproximadamente un tercio de IEEE802.11.

Por otro lado Stewart, y otros (2015) que dice que gracias a las nuevas tecnologías como los software de diseño de comunicación, dispositivos de hardware que reciben y digitalizan señales digitales y tantos dispositivos con entorno SDR como el RTL-SDR de muy bajo costo, permiten la creación y utilización de los SDR para proyectos de estudiantes y profesionales. Por todo esto este documento se baso en la enseñanza del entorno SDR en formas de sistemas compatibles con DSP (procesador de señales digitales), los receptores disponibles de SDR como el RTL-SDR, con soporte en softwares, algunos pagos y otros libres, que permiten que los hardware ahora

puedan interactuar con Matlab y Simulink por ejemplo, permitiendo que hoy por hoy sea más sencillo el desarrollo de SDR desde un simple computador de escritorio. En este proyecto los autores se propusieron enseñar la realización y utilización de SDR, los componentes necesarios y aspectos claves como los receptores QPSK con sincronización y demás sistemas de comunicaciones digitales, así que con este proyecto los estudiantes y profesionales tendrán todas las bases necesarias para que desde casa usando su ordenador personal, el hardware LTR-SDR, software tales como Matlab y Simulink, creen su propio SDR sin necesidad de los costosos analizadores de espectros, generadores de señales, etc., que generalmente se debían usar para encontrar señales de radio que pudieran ser espectralmente vistas, analizadas e incluso descodificadas y que ahora se puede lograr lo mismo con los componentes mencionados desde casa, a un bajo costo y con mayor facilidad.

En otro trabajo se presenta la implementación y prueba de un SDR (Radio definida por software) de bajo costo basado en el Raspberry Pi2 computer (una diminuta placa que contiene un poderoso chip que la convierte en un ordenador de bajo costo), junto con en el receptor de radio Dongle RTL-SDR y el software receptor de radio GNU radio companion. En este documento tratan aspectos clave en la instalación y configuración del software GNU radio companion y de los controladores RTL-SDR en Raspbian Operating System, donde el dispositivo resultante puede demodular tanto señales AM como FM, en el rango de frecuencia desde los 30MHz a los 1.7GHz, y puede cumplir funciones moderadas de un analizador de espectro. Este proyecto se centró en realizar todo el procesamiento en la Raspberry Pi2 computer ejecutando el software libre GNU radio companion y utilizando el RTL-SDR como la fuente de la señal, con estas configuraciones lo que se busco fue la reducción de costos y gracias al software utilizado que cuenta con una interfaz gráfica muy sencilla de usar y que permite muchas

aplicaciones con diferentes medios manipulables, para este tipo de proyectos. Al final se pudo evidenciar que con el software y la computadora Raspberry Pi2, en conjunto fueron lo suficientemente potentes para poder demodular en tiempo real las señales de radio FM visualizando la densidad espectral de potencia FFT, por consiguiente dicho proyecto fue un éxito y logro demostrar que se puede realizar un SDR con equipos de dominio general a muy bajo costo. (Sierra & Ramírez, 2015)

Hoy en día está en gran auge y demanda los sistemas de radio reconfigurable ya que la gran variedad de aplicaciones como la transmisión de datos a través de un canal inalámbrico, que es todo un reto y teniendo en cuenta que cada aplicación requiere un procesamiento de señal diferente, hace que los sistemas de radio tradicional fijos e integrados ya no sean viables para satisfacer las necesidades y los requisitos inmersos en el cambio de las comunicaciones. La radio definida por software o mejor conocida como SDR, es una plataforma de radio basada en software que ofrece flexibilidad para entregar un sistema altamente reconfigurable logrado en conjunto entre hardware y software como USRP y GNU Radio companion respectivamente. Sin embargo, la realización del concepto SDR está inherentemente limitada por los componentes analógicos del hardware que está siendo utilizado y es por ello que este proyecto tuvo por objeto proporcionar una información informal sobre su viabilidad como una plataforma de alta reconfiguración para sistemas de comunicación. En este documento se presenta la SDR-basada en el transceptor de un sistema común de modulación digital mediante GNU Radio y USRP. Donde se logra concluir que USRP y GNU Radio son herramientas muy buenas y fiables en plataforma SDR para emular el transceptor de sistemas de comunicación digital. La flexibilidad que ofrece esta plataforma permite la reconfiguración a nivel de software mientras mantiene el mismo hardware para facilitar varios protocolos o estándares de comunicación. La flexibilidad también permite

establecer puentes entre la simulación y dominios experimentales reales que abre más posibilidades de aplicaciones inteligentes a explorar. (Muslimin, Asnawi, Ismail, & Jusoh, 2016)

Al igual los autores Suzuki, Kawakita, & Ichikawa (2016) propusieron la implementación de un SDR remoto basado en el GNU radio companion de un SDR existente para realizar una plataforma cloud-based de entorno SDR. Ellos aclaran que el entorno SDR es configurado generalmente con un convertidor analógico a digital (ADC) y un ordenador personal (PC), pero que estos dos son configuraciones fijas y para los diferentes proyectos las reconfiguraciones son necesarias para así responder a diferentes fuentes de datos y entornos de computación cuando el protocolo de radio o SDR son indefinidos. Así que para habilitar esta reconfiguración, los autores presentan una plataforma en la nube con recursos de computación escalables y fuentes de datos desarrollados en un área amplia, utilizando un entorno de desarrollo SDR existente. Para poder aplicar este entorno de desarrollo de la plataforma en la nube, se requiere la separación de la interfaz de usuario y la gestión de datos y precisamente en este documento, los autores implementan un entorno de SDR para muchos usuarios a través de la separación de la interfaz de usuario seleccionando a GNU Radio Companion (GRC) como plataforma. Los autores se centraron en la compatibilidad y flexibilidad con respecto a la configuración, con la finalidad de permitir al usuario ajustar los recursos que se consumen en el PC. Los resultados muestran que una implementación particular de la ejecución y desarrollo de SDR remoto es adecuada y además, confirmaron la compatibilidad con GRC en términos de habilidades de usuario y de software, y consiguieron un procesamiento distribuido para asignar recursos entre el cliente y la nube, según la moderación del usuario, dividiendo el procesamiento del SDR en base a la unidad de función.

Por último y como tema adicional, pero no menos importante se puede asegurar que en esta época en la que estamos y vivimos, se ha vuelto tan importante la realización de proyectos con SDR que en este trabajo se presenta un estudio sobre el uso de la modulación FBMC (Filtro de Banco Multi Carrier) alternativa de la modulación OFDM (multiplexación por división de frecuencia ortogonal) en servicios de Radiodifusión Digital Terrestre B (ISDB-TB), desarrollando un modelo completo de radio de software en GNU Radio companion, realizando comparaciones entre el sistema con modulación OFDM y FBMC. El GNU radio ha sido una herramienta importante para simular sistemas de comunicaciones, debido a que es un sistema de software, que permite realizar análisis y mediciones en tiempo real. Este trabajo permitió definir que con la modulación FBMC se dan diversos beneficios para la ISDB-TB en comparación con la modulación actualmente utilizada OFDM, se estableció disminución de las interferencias entre las sub portadoras, al igual mejoro la robustez del sistema en un 12 %, con la utilización de FBMC se mejoró en un 6% la tasa de bits/seg, entre otras. Esto se logró determinar gracias a la utilización de SDR con el software GNU radio companion y demás componentes que ayudaron a que este proyecto llegara a felices términos. (Hengles Almeida, Batista Lopes, & Akamine, 2017)

## 4.2. MARCO CONCEPTUAL

### 4.2.1. Ondas de radio:

Las ondas de radio son ondas electromagnéticas con características específicas que permiten su uso para la transmisión de información. Estas ondas electromagnéticas son la forma de propagación a través del espacio de una radiación electromagnética, además este tipo de ondas no necesitan de un medio físico para su propagación, es decir que estas pueden propagarse a través de la nada, a diferencia de otro tipo de ondas como las mecánicas, por ejemplo, las ondas sonoras que necesitan del aire para su propagación. En la siguiente Figura se observa un gráfico de una onda de radio con sus características. (Vaquerizo & Moreno, 2016).

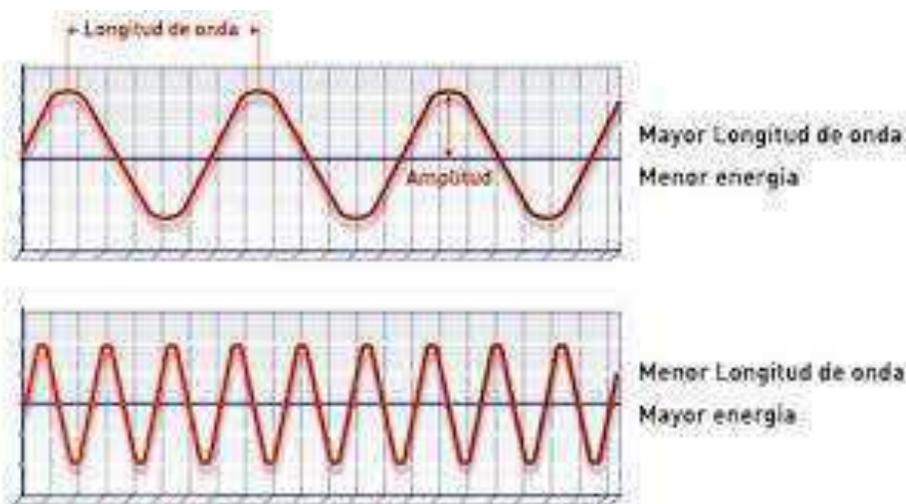


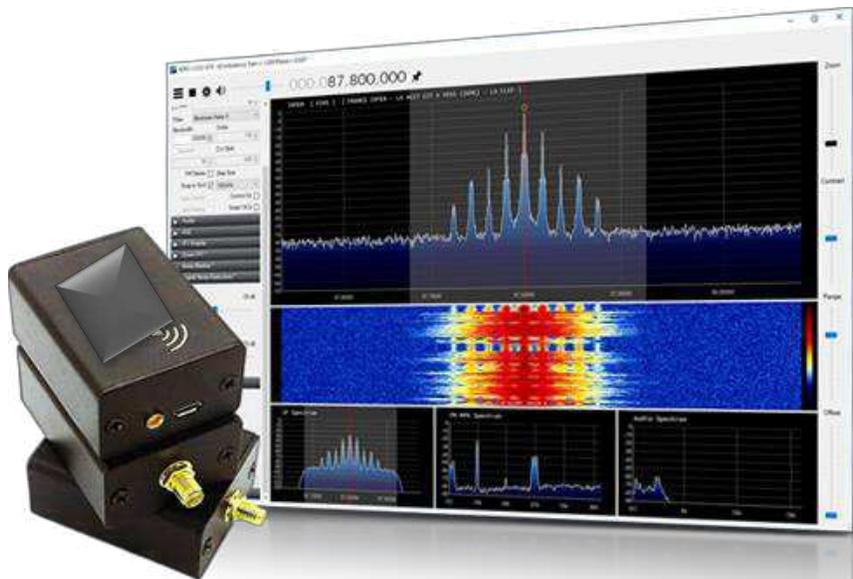
Figura 1. Onda de radio (Comin1415cs, s. f.)

### 4.2.2. Radio definida por software (SDR):

El concepto SDR básicamente hace referencia al hecho de utilizar un software para controlar casi todas las funciones de un dispositivo hardware de comunicaciones o de sensores tipo radar. Esta tecnología SDR es muy flexible y fácil de usar lo que permite y habilita la creación de dispositivos inalámbricos interoperables con diversas ventajas, entre las que se puede resaltar:

- Múltiples modos de operación
- Reconfiguración
- Actualizaciones Over-The-Air
- Menor coste de desarrollo

El concepto SDR básicamente lo que hace es acercar el código lo máximo posible a la antena, convirtiendo así problemas hardware en problemas software, ya que gracias a la tecnología con la que cuenta (tecnología software, tecnología de hardware de procesamiento de señal, tecnología RF) como se muestra en la siguiente Figura, donde la unión entre un hardware y un software hacen realidad la consecución de este concepto. (E&Q Engineering, 2010).

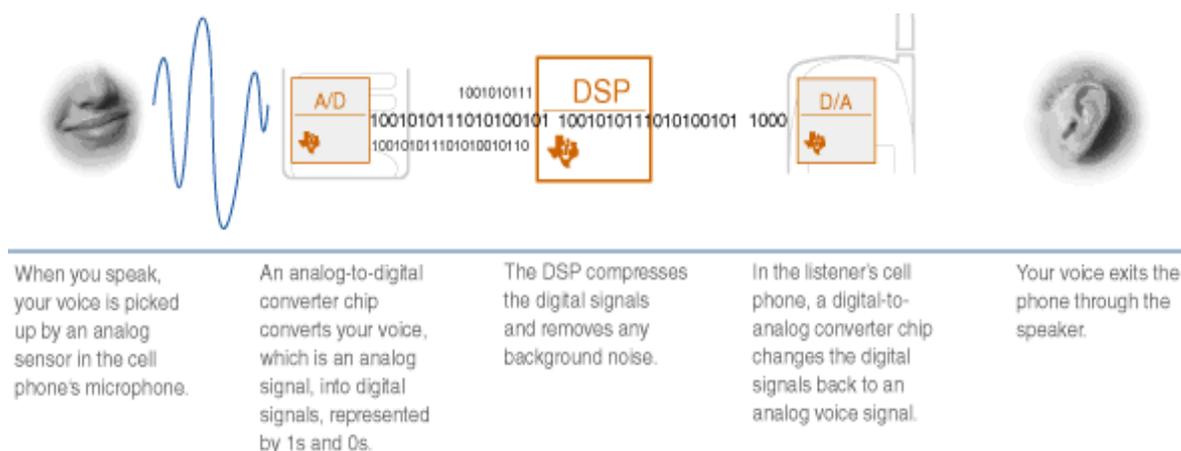


*Figura 2. SDR (AIRSPY, 2016)*

#### **4.2.3. Procesamiento Digital de Señal (DSP):**

Este es un término que se relaciona bastante con el concepto SDR y básicamente es una técnica que toma señales que usualmente están en forma analógica, y las convierte en datos digitales que luego pueden ser analizados. Este análisis es ejecutado en forma digital ya que una señal cuando

ha sido reducida a valores numéricos discretos, sus componentes pueden ser aislados, analizados y reordenados más fácilmente que en su primitiva forma analógica. Prácticamente DSP es la transformación de una señal en un hardware digital según reglas previamente definidas las cuales son introducidas en el hardware a través de un software que puede o no manejar lenguajes tanto de alto como de bajo nivel. En la siguiente Figura se muestra un esquema muy simple de lo que trata DSP. (Hernandez, 2004).



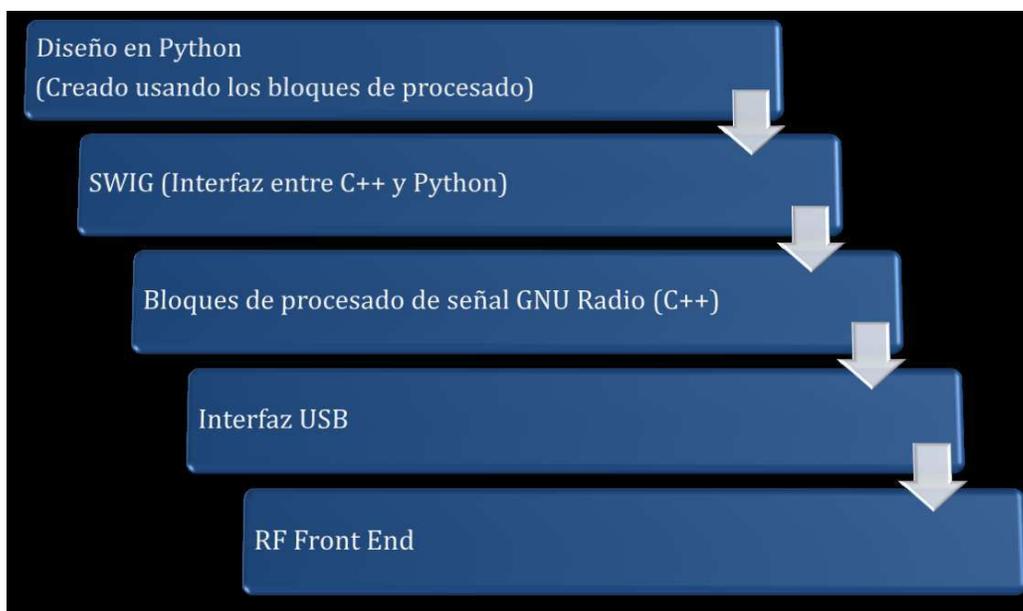
*Figura 3. DSP (Hernandez, 2004)*

#### 4.2.4. GNU Radio:

GNU Radio permite manipular señales de radio por medio de procesamiento digital de señales, esto se debe a que básicamente GNU Radio es un grupo o un conjunto de archivos y aplicaciones pertenecientes a librerías mediante las cuales permiten realizar los diferentes diseños para los sistemas de radio definidos por software (SDR), esto se logra cuando el PC se conecta a un SDR, sin embargo también es posible trabajar con una tarjeta de audio. GNU en primera instancia fue concebido para correr sobre sistemas operativos GNU/Linux como Ubuntu, sin embargo hoy en día ya es posible que GNU corra en sistemas operativos Windows y Mac.

Los diseños que se realizan en GNU Radio son programados en Python para lo cual es necesario crear un grafo donde cada nodo son los bloques de procesamiento de la señal y el

conector entre ellos representa el flujo de los datos. Estos bloques son implementados en C++ y llevados a Python mediante un programa llamado SWING, esto quiere decir que para implementar el diseño se crearan módulos en Python basados en los bloques implementados en C++. En la siguiente Figura se muestra la interacción entre los diferentes niveles de GNU Radio. (Pinar & Murillo, 2011)

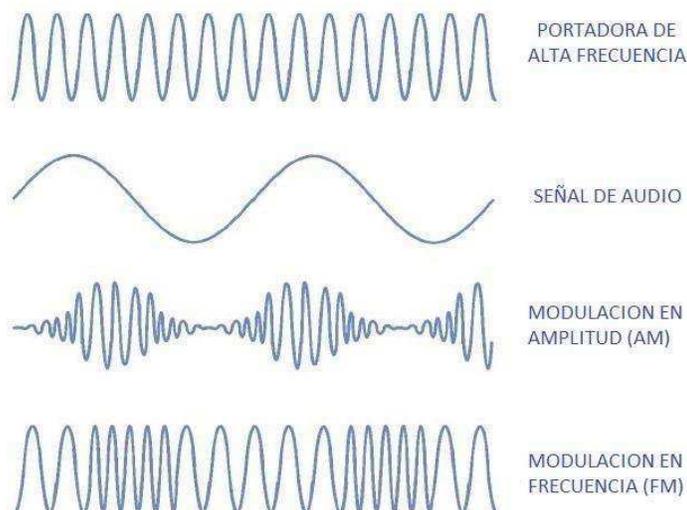


*Figura 4. Interacción niveles GNU Radio. (Pinar & Murillo, 2011)*

#### **4.2.5. Modulación y Demodulación:**

En telecomunicación el término modulación engloba a todo un conjunto de técnicas enfocadas al transporte de información sobre una onda portadora, que típicamente es una onda senoidal. Estas técnicas permiten un aprovechamiento mucho más amplio del canal de comunicación lo que permite transmitir más información simultánea protegiendo aún más la información de posibles interferencias y ruidos. Básicamente, la modulación consiste en hacer que un parámetro de la onda portadora cambie su valor de acuerdo con las variaciones de la señal moduladora, que es la información que queremos transmitir. La demodulación es básicamente realizarle el proceso

inverso a la onda modulada para poder recoger la información que original fue enviada. En las ondas de radio se encuentran tres tipos de modulación analógica: AM, FM y PM y en la modulación digital se encuentran: ASK, FSK, PSK, QPSK, QAM, OFDM, entre otros. En la Figura siguiente se muestra un ejemplo de la modulación analógica en AM y FM. (Gomez, 2012).



*Figura 5. Modulación AM y FM (Alfaro, 2011)*

### **4.3. MARCO LEGAL**

En el año 2009 fue creada la Agencia Nacional del Espectro (ANE) mediante la ley 1341 la cual dice textualmente “Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones –TIC, se crea la Agencia Nacional de Espectro y se dictan otras disposiciones” (Ministerio de Tecnologías de la Información y las Comunicaciones, 2009). Esta ley aclara que la ANE tiene como objeto ofrecer el soporte técnico de todo lo relacionado con la gestión, el control, planeación y vigilancia del espectro radioeléctrico, todo esto lo debe hacer en plena coordinación con las demás entidades que tengan funciones o actividades que se relacionen con lo mismo. A continuación se muestran las diferentes normas que definen las funciones que tiene la ANE:

#### **4.3.1. Ley 1341 de 2009 (Julio 30)**

“Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones – TIC, se crea la Agencia Nacional de Espectro y se dictan otras disposiciones.”

#### **4.3.2. Decreto 93 de 2010 (Enero 19)**

“Por el cual se adopta la estructura de la Agencia Nacional del Espectro - ANE, y se dictan otras disposiciones”

#### **4.3.3. Decreto 4169 de 2011 (Noviembre 3)**

“Por el cual se modifica la naturaleza jurídica de la Agencia Nacional del Espectro y se reasignan funciones entre ella y el Ministerio de Tecnologías de la Información y las Comunicaciones.”

#### **4.3.4. Ley 1507 de 2012 (Enero 10)**

“Por la cual se establece la distribución de competencias entre las entidades del estado en materia de televisión y se dictan otras disposiciones.”

#### **4.3.5. Decreto 2453 de 2013 (07 de Noviembre).**

“Por el cual se modifica el Decreto 2618 de 2012.” Precisa el alcance y las funciones entre el Ministerio de las Tecnologías de la Información y las Comunicaciones y la Agencia Nacional del Espectro. (ANE, 2016)

Otra ley es la 1286 de 2009 que fortalece a Colciencias y al sistema nacional de tecnología e innovación en Colombia; textualmente esta ley dice “Por la cual se modifica la Ley 29 de 1990, se transforma a Colciencias en Departamento Administrativo, se fortalece el Sistema Nacional de Ciencia, Tecnología e Innovación en Colombia y se dictan otras disposiciones” (Congreso de la Republica , 2009). Esta ley lo que pretendió como objetivo general fue el fortalecimiento de Colciencias y del Sistema Nacional de ciencia y tecnología para lograr un modelo productivo fundado en la tecnología, innovación y la ciencia, así como apoyar el desarrollo productivo de una nueva industria nacional y darle un valor agregado a los diferentes productos y servicios de la economía colombiana. Para destacar esta ley promueve el establecimiento de disposiciones que conlleven a fortalecer el desarrollo científico y de innovación para el correcto cumplimiento de dicha ley, así como las bases para poder consolidar una política de estado en la ciencia, tecnología e innovación, lo que lleva a incrementar la capacidad científica, tecnología, innovación y competitividad, buscando así darle un mayor valor agregado a los productos y servicios nacionales e incrementando el bienestar de la población en general. (Congreso de la Republica , 2009)

Se menciona a la ANE ya que este proyecto se basa en el análisis de espectro y es esta entidad la que se encarga de este tema legalmente hablando y por ello es indispensable traer a acotación las normas y leyes que la crearon y la facultan en sus diferentes funciones. Al igual se debe tener muy en cuenta la ley de ciencia y tecnología de Colciencias ya que éste es quien da el marco general para el desarrollo tecnológico del país.

## 5. METODOLOGÍA DEL PROYECTO

### 5.1. ENFOQUE CUANTITATIVO

Se investigaron los componentes electrónicos del SDR para conocer su funcionamiento y posibles usos, así como también los bloques a utilizar de GNU Radio para el desarrollo de soluciones o posibles alternativas que nos brinda el SDR. Implementando un método de información Cuantitativa por Experimentación se realizó el desarrollo de dos laboratorios con la tarjeta Hack RF, donde se hace una recepción de las de diferentes radio frecuencias obteniendo un resultado de captación de la señal muy óptimo, teniendo en cuenta que todo se desarrolla por software.

Se realizó la documentación de las prácticas con la utilización del GNU Radio (Software) y la tarjeta Hack RF y se recopiló la información necesario para su perfecto funcionamiento y puesta en marcha para posibles prácticas de recepción de señales en este caso (Recepción FM)



## 6.2. PRESUPUESTO

### 6.2.1. Presupuesto de personal

Tabla 2. Presupuesto de personal.

<b>Nombre</b>	<b>Horas dedicación semanal</b>	<b>Semanas dedicadas</b>	<b>Valor hora</b>	<b>TOTAL</b>
Danny Fabian Sandoval	4	20	23.000	1'840.000
Andrés Zamora Montañez	4	20	23.000	1'840.000
<b>TOTAL</b>				3'680.000

### 6.2.2. Presupuesto de equipos

Tabla 3. Presupuesto de equipos.

<b>Equipo</b>	<b>Justificación</b>	<b>VALOR</b>
Computador	Instalación Programa GNU Radio	1'900.000
HackRF	Necesaria para la realización del proyecto	1'200.000
<b>TOTAL</b>		3'100.000

### 6.2.3. Presupuesto de software

Tabla 4. Presupuesto de software.

<b>Software</b>	<b>Justificación</b>	<b>VALOR</b>
<b>TOTAL</b>		0

### 6.2.4. Presupuesto de materiales y suministros

Tabla 5. Presupuesto de materiales y suministros.

<b>Materiales**</b>	<b>Justificación</b>	<b>VALOR</b>
<b>TOTAL</b>		0

### 6.2.5. Presupuesto de eventos científicos y salidas de campo

*Tabla 6. Presupuesto de eventos científicos y salidas de campo.*

<b>Lugar</b>	<b>Justificación***</b>	<b>Cantidad</b>	<b>Costo Unitario</b>	<b>VALOR</b>
<b>TOTAL</b>				0

### 6.2.6. Presupuesto de material bibliográfico

*Tabla 7. Presupuesto de material bibliográfico.*

<b>Libro</b>	<b>Justificación</b>	<b>VALOR</b>
<b>TOTAL</b>		0

### 6.2.7. Presupuesto de publicaciones

*Tabla 8. Presupuesto de publicaciones.*

<b>PUBLICACION</b>	<b>VALOR</b>
<b>TOTAL</b>	0

### 6.2.8. Presupuesto de servicio técnico

*Tabla 9. Presupuesto de servicio técnico.*

<b>Tipo de servicio</b>	<b>Justificación</b>	<b>VALOR</b>
<b>TOTAL</b>		0

**6.2.9. Presupuesto general***Tabla 10. Presupuesto general.*

<b>ITEM</b>	<b>TOTAL</b>
Personal	3'680.000
Equipos	3'100.000
Software	0
Materiales y suministros	0
Salidas de campo (locales)	0
Material bibliográfico	0
Publicaciones	0
Servicio técnico	0
<b>TOTAL</b>	<b>6'780.000</b>

## 7. DESARROLLO DE PROBLEMA DE INVESTIGACIÓN

### 7.1. ESTUDIO DE LOS COMPONENTES DE UN HARDWARE SDR

#### 7.1.1. QUE ES EL ESPECTRO RADIOELÉCTRICO

El espectro radioelectrónico da claridad al medio por el cual se trasmite las frecuencias de ondas de radio electromagnéticas de carácter limitado, “las cuales permiten la comunicación de (radio, Televisión, Internet, Telefonía Móvil, etc. Bien que es de dominio público y que se encuentra regulado por el Ministerio de Tecnologías de la Información y las Comunicaciones” (MINTIC, 2016). Por tal causal se convierte en un medio intangible que puede utilizarse para la prestación de diversos servicios de comunicación, conjuntamente o no con medios tangibles como cables, Fibra Óptica, entre otros.

Es clara que la definición de espectro radioeléctrico, tal y como la ha definido la Unión Internacional de Telecomunicaciones (UIT), organismo especializado de las Naciones Unidas con sede en Ginebra (Suiza) es: "Las frecuencias del espectro electromagnético usadas para los servicios de difusión y servicios móviles, de policía, bomberos, radioastronomía, meteorología y fijos." Este "(...) no es un concepto estático, pues a medida que avanza la tecnología se aumentan (o disminuyen) rangos de frecuencia utilizados en comunicaciones, y corresponde al estado de avance tecnológico. (UIT, 2005)



*Figura 6. Bandas de Frecuencia*

El espectro radioeléctrico, el cual se puede apreciar en la figura 6, se divide en bandas de frecuencia que competen a cada servicio que estas ondas electromagnéticas están en capacidad de prestar para las distintas compañías de telecomunicaciones, avaladas y protegidas por las instituciones creadas para tal fin de los estados soberanos.

### 7.1.2. Que es DAC-ADC?

En gran parte la mayoría de los controladores son utilizados en dispositivos digitales como los son, Computadores, Televisores, radios entre otros. Por tal motivo al tratarse de un dispositivo digital es necesario que este cuente con dos dispositivos importantes para su desarrollo como se muestran en la figura 7.



*Figura 7. ADC, Y DAC (turmero, 2015)*

Se pretende buscar una relación única entre una señal analógica y una digital o viceversa.

“Las magnitudes físicas son analógicas y normalmente el procesado de señal se realiza de forma digital, de ahí la necesidad de los convertidores ADC.

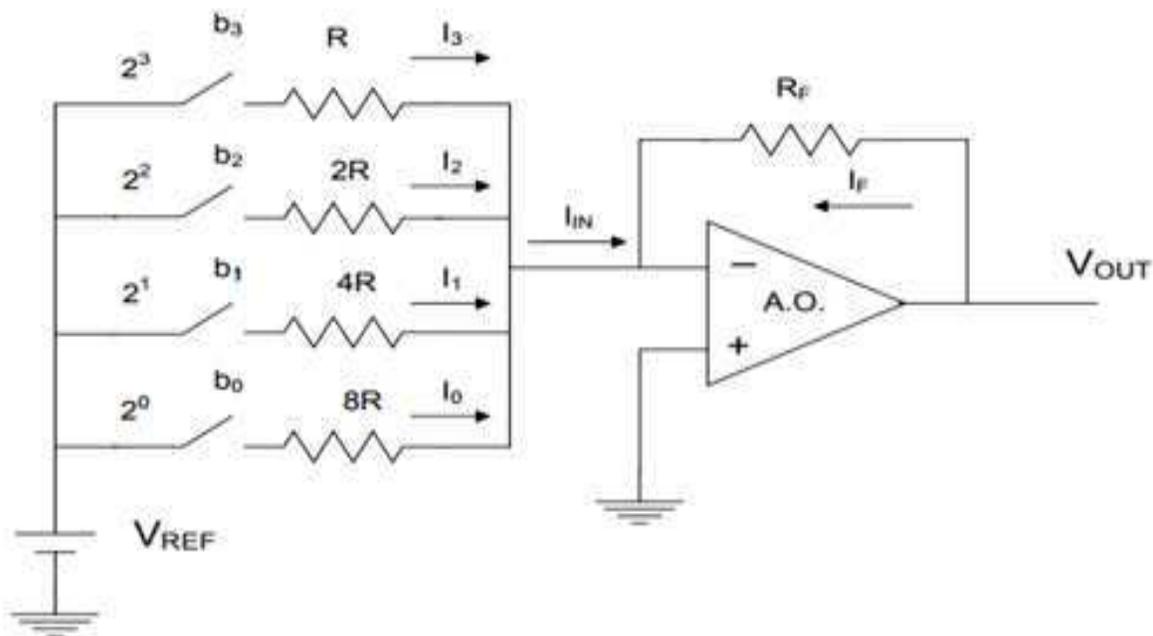
También puede ser necesario actuar sobre un sistema analógico una vez procesada una señal de forma digital por lo que se hace necesario un convertidor DAC” (Escuela Universitaria de Ingeniería , 2017).

Para ingresar datos analógicos a un circuito digital, estos se deben transformar en valores digitales codificados; Evaluar numéricamente la señal en instantes discretos muestreo y obtener una señal digitalizada, es el paso a paso que nos refleja la figura 8.



Figura 8. Muestreo de una señal digitalización (turmero, 2015)

### 7.1.3. CONVERTIDOR DIGITAL A ANALOGICO (DAC) CARACTERISTICAS DE FUNCIONAMIENTO.



$$V_{OUT} = -\frac{V_{REF} R_f}{R} \left( \frac{b_0}{8} + \frac{b_1}{4} + \frac{b_2}{2} + b_3 \right)$$

Figura 9. Convertidor por Ponderación Binaria (Convertidores Digitales DAC y analogico a Digital ADC, 2017)

La Figura 9 es un ejemplo de la manera más simples de convertidores que trabajan en paralelo, “su principal desventaja es el gran número de resistencias de distinto valor que necesita. Así, para un DAC de 10bits, son necesarias 10 resistencias con valores de  $R$  a  $512 R$ , con una muy baja tolerancia para poder mantener la precisión del convertidor”. (Convertidores Digitales DAC y analogico a Digital ADC, 2017)

#### 7.1.4. CONVERTIDOR EN ESCALERA PONDERADO R/2R (WEIGHTED RESISTOR LADDER)

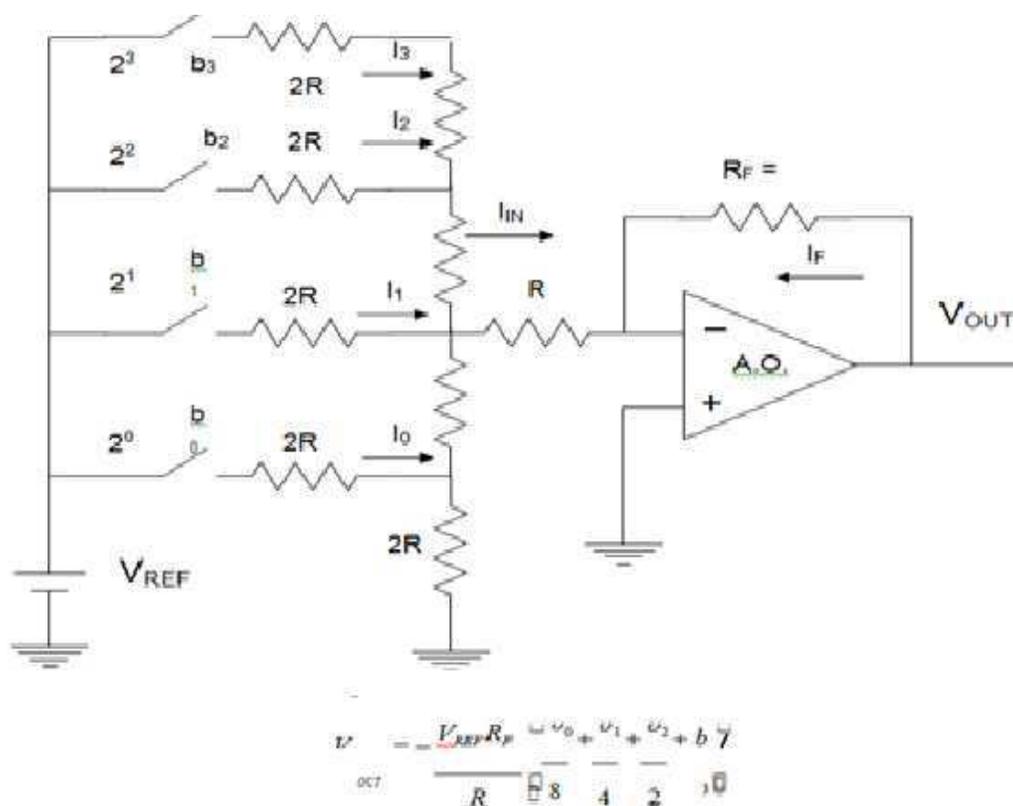


Figura 10. Convertidor en escalera R/2R (Convertidores Digitales DAC y analogico a Digital ADC, 2017)

De acuerdo a la figura 10 los parámetros característicos de un convertidor DAC son:

**7.1.4.1. Escala completa de salida (Full-Scale Output (FSO)).** Se define como el máximo valor analógico de salida posible, es decir, cuando se aplica a la entrada el máximo

valor binario. Así, en el ejemplo de la figura 2., la FSO es 5 v. puesto que cuando se aplica el máximo valor binario de entrada  $(b_3, b_2, b_1, b_0) = (1, 1, 1, 1)$  el valor de salida es 5 v.

**7.1.4.2. Resolución (Resolution):** Se define como la diferencia en voltios que se produce a la salida del convertidor para un cambio sucesivo de su valor binario. Así, en el ejemplo de la figura 7.2.1., la resolución es  $1/3 \text{ v.} = 0,33 \text{ v.}$  También, es típico expresarlo como el recíproco del número de escalones discretos de la salida (expresado en porcentaje). Para un DAC el número total de escalones discretos es  $2^n - 1$ , donde  $n$  es el número de bits. Así, para un DAC de 8 bits, la resolución es:  $[1/(2^8 - 1)] * 100 = 0,3922\%$ . Además, se define la resolución como el número de bits que se convierten y es el parámetro que caracteriza al convertidor (Convertidores Digitales DAC y analogico a Digital ADC, 2017).

**7.1.4.3. Precisión:** Es la comparación entre la salida real de un DAC y la salida esperada. Se expresa como un porcentaje de la tensión de salida máxima. Idealmente, la precisión debería ser como mucho  $\square 1/2$  del bit LSB (bit menos significativo). Así, para un DAC de 8 bits, LSB es  $1/256$  y la precisión es aproximadamente  $\square 0,2 \%$ .

**7.1.4.4. Linealidad:** Es la desviación (error lineal) de la salida ideal. Un caso especial es el error de offset (cuando todos los bits están a cero).

**7.1.4.5. Monotonicidad:** Un DAC es monótono si no produce escalones inversos cuando se le aplica secuencialmente su rango completo de bits de entrada. (FLOYD, 2000).

### 7.1.5. CONVERTIDORES ANALÓGICOS A DIGITAL (ADC) MODOS DE FUNCIONAMIENTO.

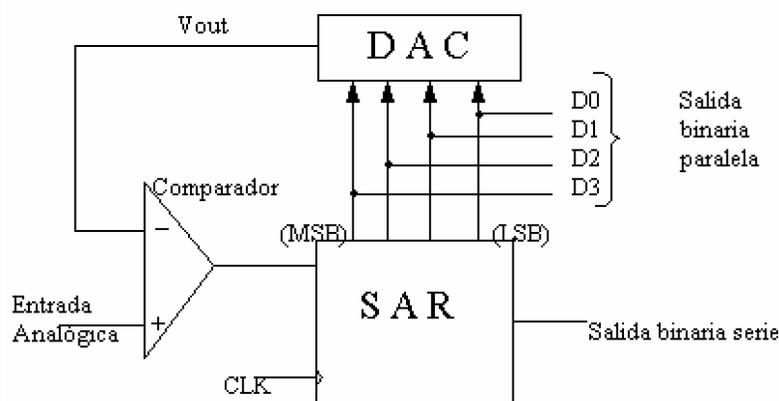


Figura 11. Aproximaciones sucesivas (Convertidores Digitales DAC y analogico a Digital ADC, 2017)

El funcionamiento de un servidor ADC es que los bits de entrada al DAC se habilitan (ponen a 1) cada vez, comenzando por el MSB. Cada vez que se habilita un bit, el comparador produce una salida que indica si la tensión analógica de entrada es mayor o menor que la salida del DAC (Vout). Así, tenemos que:

Si  $V_{out} > V_{analógica}$   $\epsilon$  Comparador = 0  $\epsilon$  MSB de SAR = 0

Si  $V_{out} < V_{analógica}$   $\epsilon$  Comparador = 1  $\epsilon$  MSB de SAR = 1

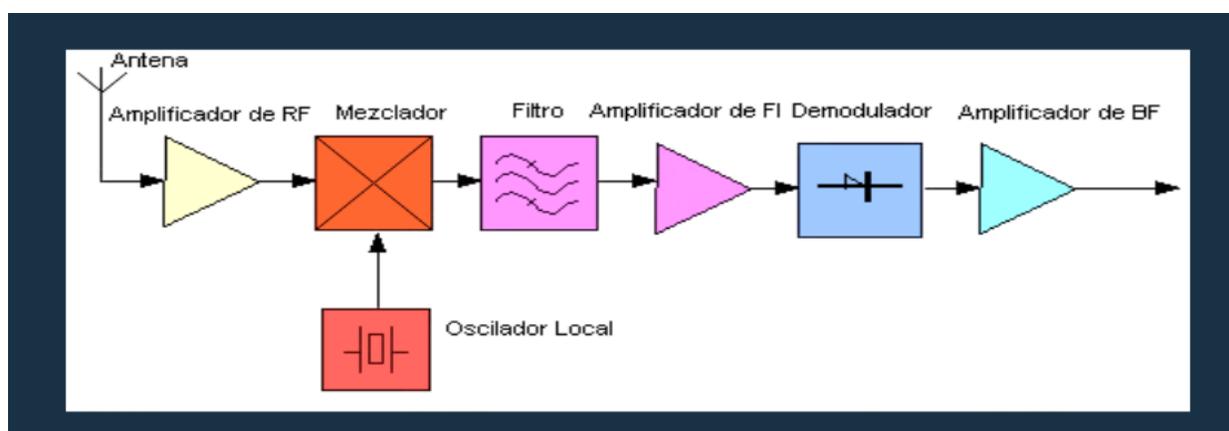
La conversión completa necesita  $n$  ciclos. (Merino, 1989)

### 7.1.6. QUE ES LA FRECUENCIA INTERMEDIA (IF)

El funcionamiento de una frecuencia intermedia radica en que todos los circuitos sintonizados existentes a partir de la etapa en que se efectúa la mezcla, trabajan a una frecuencia fija (la de la FI) y por tanto son más fáciles de ajustar. De este modo se mejora la selectividad y se facilita el diseño de las etapas amplificadoras; En la figura 12 se puede observar el esquema de bloques de un receptor superheterodino.

La señal de antena atraviesa un filtro paso banda ajustado al canal que deseamos recibir y se introduce en un amplificador para mezclarse posteriormente con un tono puro (onda senoidal) generado en un oscilador local.

Tras el mezclador se obtienen 4 señales, dejando pasar, mediante un filtro, únicamente la correspondiente a la frecuencia intermedia (FI o IF) que es amplificada para ser aplicada a un circuito demodulador<sup>1</sup> obteniéndose a la salida de este la señal de audio o en BF que se amplifica para llevarla a un altavoz. (GG, 2015)



*Figura 12 Receptor de Radio tipo superheterodino con filtro de rechazo a la banda de frecuencia. (2012)*

La frecuencia intermedia fue utilizada inicialmente en un receptor de radio de superheterodino, el cual fue inventado y expuesto por primera vez en el año de 1918 por el científico Edwin Armstrong, Miembros de la U.S. Army Signal Corps, construyeron un equipo de búsqueda con el fin de direccionar su radio para rastrear diferentes señales militares alemanas de muy alta frecuencia. Los amplificadores de tubos de vacío tríodos de esta época no amplificaban de forma estable por encima de los 500 kHz, sin embargo, resultaba fácil hacerlos oscilar por encima de esa frecuencia.

De esta manera la posible solución que ofreció Armstrong fue instalar un tubo oscilador que crearía una frecuencia cercana a la frecuencia de la señal sintonizada, y mezclarla con la señal

sintonizada en un tubo de mezcla, creando un 'heterodino' o señal diferencia de frecuencia más baja, la cual podría ser amplificada fácilmente. (Wikipedia, 2017)

Un sintonizador es un subsistema que recibe frecuencia de radio (RF), como las de las emisoras de radio, y convierte la frecuencia portadora seleccionada y su ancho de banda asociado, en una frecuencia fija que es adecuada para su posterior procesamiento. (Wikipedia, 2017)

### **7.1.7. PROCESO DIGITAL DE SEÑAL DSP**

Los DSP o procesadores digitales de señal son microprocesadores exclusivamente diseñados para el procesado digital de señal. Las características más básicas son el formato aritmético, la velocidad, la organización de la memoria o la arquitectura interna hacen que sean o no adecuados para una aplicación en particular, algunas otras características son el coste o la disponibilidad de una extensa gama de herramientas de desarrollo.

El DSP se puede ser utilizado en diferentes chips los cuales puedan trabajar con diferentes señales digitales. En la práctica, el término se refiere a microprocesadores específicamente diseñados para realizar procesado digital de señal. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos intensos implicados en la mayoría de sistemas de procesado de señal en tiempo real. Por ejemplo, las arquitecturas de los DSP incluyen circuitería para ejecutar de forma rápida operaciones de multiplicar y acumular, conocidas como MAC. A menudo poseen arquitecturas de memoria que permiten un acceso múltiple para permitir de forma simultánea cargar varios operados, por ejemplo, una muestra de la señal de entrada y el coeficiente de un filtro simultáneamente en paralelo con la carga de la instrucción. (Salazar)

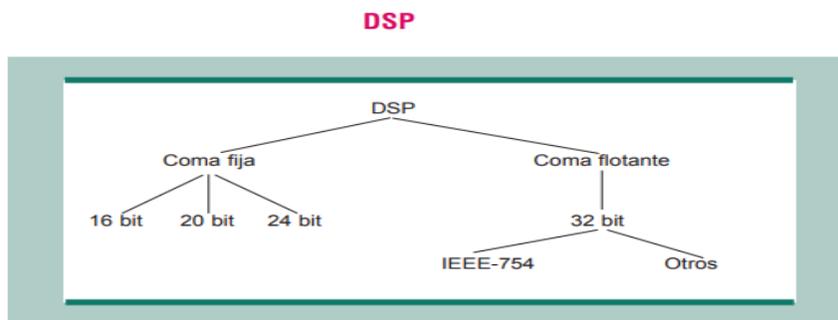


Figura 13. Representaciones numéricas comunes en los DSP comerciales (Salazar).

### 7.1.8. APLICACIONES

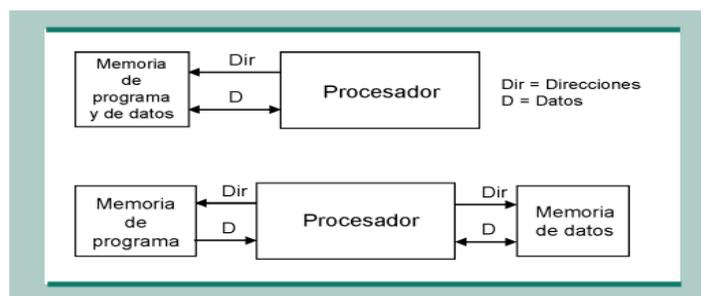
Los DSP se utilizan en variedad de aplicaciones, desde radares hasta la electrónica básica de consumo. Por ello, el diseñador debe elegir muy bien el DSP teniendo en cuenta la importancia relativa de las prestaciones, costo, integración, facilidad de desarrollo, consumo y otros factores para las necesidades de la aplicación en particular. La mayoría de las aplicaciones, en términos de dinero que mueven los productos, se realizan para los sistemas pequeños, económicos y con un gran volumen de producción como los de telefonía celular, modems, en donde el costo y la integración son de la mayor importancia. En sistemas portátiles aquellos que son alimentados por baterías, el consumo es dificultoso. Sin embargo, la facilidad de desarrollo es generalmente en estas aplicaciones menos importante para el diseñador. A pesar de que estas aplicaciones casi siempre implican el desarrollo de hardware y software a medida, el enorme volumen de producción justifica el esfuerzo extra de desarrollo.

La segunda clase de aplicaciones son aquellas que realizan un gran volumen de datos mediante algoritmos complejos. Como consecuencia lo que el diseñador busca un DSP es que este le proporcione las máximas prestaciones, como la facilidad de uso y un soporte para configuraciones multiprocesador. En algunos casos, más que diseñar el propio hardware y

software, el sistema se construye a partir de placas de desarrollo de catálogo y el software a partir de librerías de funciones ya existentes. (Salazar)

### 7.1.9. CARACTERÍSTICAS DE LOS DSP

La selección de un DSP tiene que tener como prioridad que posea de ciertas características condicionadas según la aplicación que se desee destinar. En la figura 14 se puede evidenciar diferentes aspectos característicos de los DSP sin que se pretenda con ello hacer una lista exhaustiva. Dichos aspectos deberán tenerse en cuenta a la hora de su elección para una aplicación en particular.



*Figura 14. (En la Parte Superior) Arquitectura Von Neumann. (En la Parte Inferior) Arquitectura Harvard. (Salazar)*

Por tal motivo y dando claridad a lo antes mencionado los DSP poseen arquitecturas especialmente diseñadas para acelerar los cálculos matemáticos intensos utilizados en la mayoría de sistemas de procesamiento de señal en tiempo real. De igual manera se ha podido evidenciar que el DSP está estrechamente ligado al tipo de aplicación.

Por otro lado, la reciente aparición de compiladores realmente eficaces en extraer el paralelismo de un programa ha propiciado la recuperación de las arquitecturas VLIW y con ello la obtención de DSP de elevadas prestaciones. (Phil Lapsley, 1996)

### 7.1.10. QUE ES FFT

El FFT o también conocido como **Fast Fourier Transform** es un algoritmo para calcular la transformada de Fourier discreta y su inversa, comúnmente se entiende que la transformación de Fourier convierte el tiempo a frecuencia o viceversa. La FFT calcula rápidamente esas transformaciones, como resultado se utiliza en diversas aplicaciones en la ingeniería, la ciencia y las matemáticas. Algunos autores afirman que puede ser el algoritmo numérico más importante de la vida.

Desde el punto de vista algorítmico, se observa que por cada coeficiente se realiza una sumatoria de todos los datos, en términos de operaciones requiere un alto costo. Se concluye que la transformada discreta de Fourier (DFT) de una secuencia de N datos se requieren  $N^2$  mientras que la FFT requiere solo  $N \log_2(N)$  Por ejemplo: con  $N=1024$

$$DFT=1.048.576$$

$$FFT= 10.240$$

Una de las maneras de aplicación se da al momento de analizar el sonido. Por tal motivo es fundamental evaluar la distribución de frecuencias de la energía que transmite un sonido, puesto que en el caso de los seres humanos el oído ejerce tal capacidad en el proceso de audición. La Figura 15 refleja el sonido de un silbato que utiliza normalmente la policía en ambos dominios, del tiempo y de la frecuencia (por medio de la TRF).

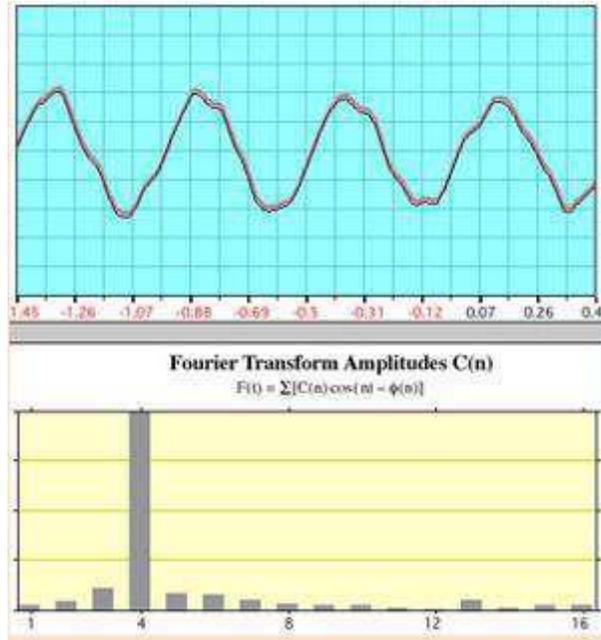


Figura 15. Silbato A. (Nave, 2014)

Llamando a cada Figura como silbato A y B, La figura que se encuentra en la parte superior muestra el sonido del silbato A solo. El gráfico superior que se encuentra representado de color Azul, Refleja la muestra ordinaria de la señal de voltaje respecto del tiempo, proporcionada por un micrófono. El gráfico inferior que se encuentra representado por un diagrama de barras es la transformada rápida de Fourier (TRF) de esa señal. Muestra que la mayor parte de la potencia se da solo a una frecuencia, aproximándose con ello a una onda pura sinusoidal. El hecho de que el pico que muestra la mayor parte de la potencia esté en la posición cuatro, sólo refleja el hecho de que fueron seleccionados cuatro períodos para la muestra FFT. (Nave, 2014)

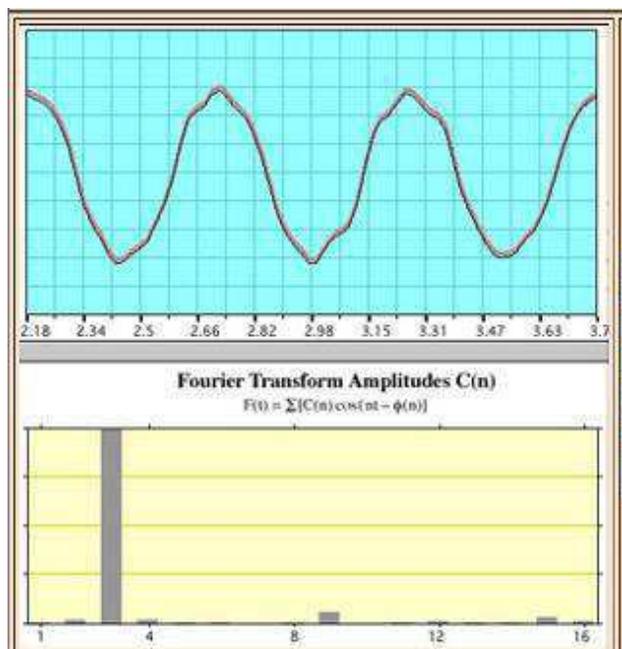


Figura 16. Silbato B (Nave, 2014)

En la figura 16 se muestra la misma clase de muestra. Pero en esta ocasión se seleccionaron 3 periodos para la TRF, arrojando como resultado principal y puntuando la posición 3.

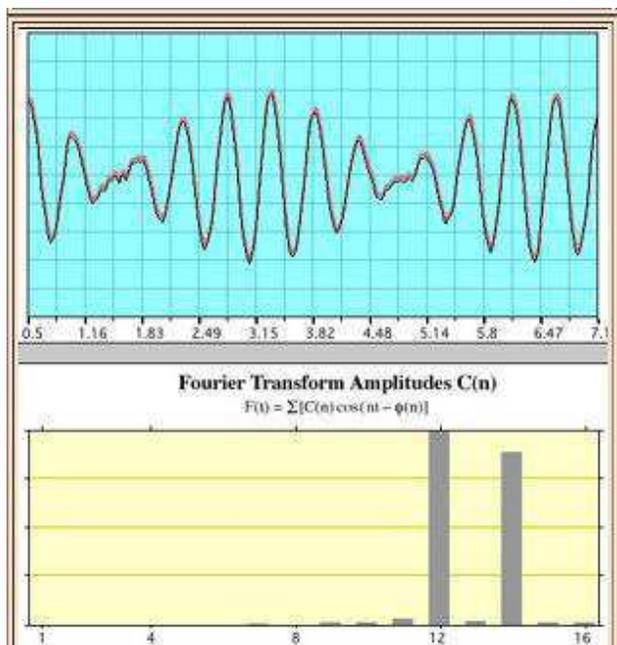


Figura 17. Silbatos A y B (Nave, 2014)

En la figura 17 se toma simultáneamente los silbato A y B y se hacen sonar al mismo tiempo, en la gráfica de color azul se evidencia un patrón de frecuencia Batimiento. La TRF muestra las dos frecuencias distintas de los silbatos individuales.

Con respecto a las diferentes figuras antes mencionadas y explicadas lo que se pretende es mostrar la naturaleza de las TRF. “Para una onda sinusoidal de una frecuencia simple, la TRF consiste en un solo pico. La combinación de dos ondas de sonido, produce un patrón complejo en el dominio del tiempo, pero la TRF muestra claramente que consiste casi enteramente en dos frecuencias” (Nave, 2014)

#### **7.1.11. LA RADIO DEFINIDA POR SOFTWARE (SDR)**

A Principios de la década del 2000 radioaficionados como Gerald Youngblood , iniciaron un proceso investigativo en el cual desarrollarían un nuevo concepto de equipos de radiocomunicaciones, los equipos de radio desarrollados por programa o "radios software", en siglas SDR (Software Defined Radio), en los que la parte hardware sería mínima, y la mayor parte de las funciones que definen un equipo de radio se definen por software (programas) en un ordenador PC o de otro tipo, dotado de tarjeta de sonido (requisito necesario). (Villegas, 2008)

Tal como se indicó, la tecnología de Software Radio (García, 2011) fue instruida por las fuerzas Armadas de EE.UU. para conseguir comunicaciones permanentes en distintas bandas con un solo equipo, asegurar su compatibilidad con la comunicación de diversas ramas como ejército, fuerza aérea, armada Etc., logrando la adaptabilidad frente a nuevas innovaciones de equipos y componentes para otros objetivos

El desarrollo de la tecnología SDR en la creación de sistemas de radio se debe mucho a la devaluación de los microprocesadores, puesto que se reconoció el empleo de procesadores con buenas prestaciones en dispositivos de uso frecuente. El uso de esta tecnología ha facilitado el

surgimiento de cambiar parte de la circuitería de los dispositivos de radio por programas que realicen las mismas funciones y le puedan otorgar un nivel superior mediante un software.

Por otro lado la SDR es una tecnología emergente particularmente dentro de la industria inalámbrica que tiene gran potencial para ser utilizado entre amplia gama de usos. En términos de requisitos de comunicación presenta soluciones económicas para los diferentes sectores como militar, civil y comercial.

## 7.1.12. COMO FUNCIONA SDR

### 7.1.12.1. Estructura del sistema SDR

Para determinar la estructura de un sistema de Radio Definido por Software (SDR) se debiera empezar con los sistemas de comunicación digitales. La figura 18 refleja el diagrama en bloques de un sistema de estas características. Algunos de estos bloques son necesarios y fundamentales para el funcionamiento del sistema de comunicación digital; Otros bloques son accesorios necesarios para un mejor rendimiento y funcionamiento del sistema como por ejemplo la codificación del canal ente otros (Garcia, 2011).

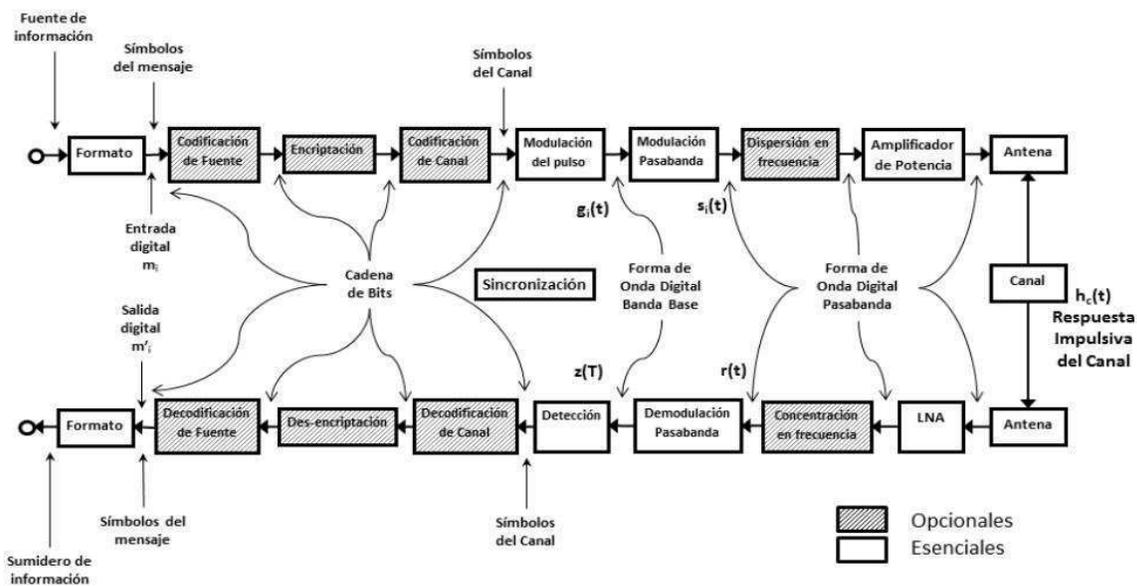


Figura 18. Diagrama en bloques de un Sistema digital de Comunicaciones (Garcia, 2011)

### 7.1.13. CONCEPTO DE RADIO DEFINIDO POR SOFTWARE SDR

Existen diferentes maneras de describir la Radio Definida por Software, una de las más nombradas y que obedece a una visión clara de la tecnología que se manifiesta en Wireless Innovation Forum es la siguiente:

Una radio en la que algunas o todas las funciones de la capa física están definidas por software [...] Una Radio Definido por Software es un radio en el que algunas o todas las funciones de la capa física so definidas por software (Forum, 2017)

Según el colaborador de Wireless Innovation Forum el señor Lee Pucker,

El SDR es una tecnología aplicable a diferentes áreas de la industria inalámbrica, la cual consiste en un sistema de comunicación en la que la mayor parte de sus funciones lo definen como un tipo de modulación y demodulación, banda de emisión y recepción entre otras cosas. Implementándose mediante software en donde el hardware se reduce considerablemente, trabajando con tecnologías de procesamiento programables.

Los autores Sethares & Johnson Jr, (2003) afirman:

Definido por Software significa que los elementos claves del radio están implementados en software; [...] Tomar un enfoque definido por software refleja la tendencia de los diseños modernos donde una parte cada vez mayor del sistema está diseñada y construida utilizando software reconfigurable, en vez de hardware fijo. Esto permite introducir, demostrar y entender los conceptos detrás de la transmisión a través de simulación. Por ejemplo, cuando se hable de traslación en frecuencia de una señal, los procedimientos se presentan matemáticamente a través de ecuaciones, gráficamente usando diagramas en bloques y luego concretamente como programas de MATLAB.

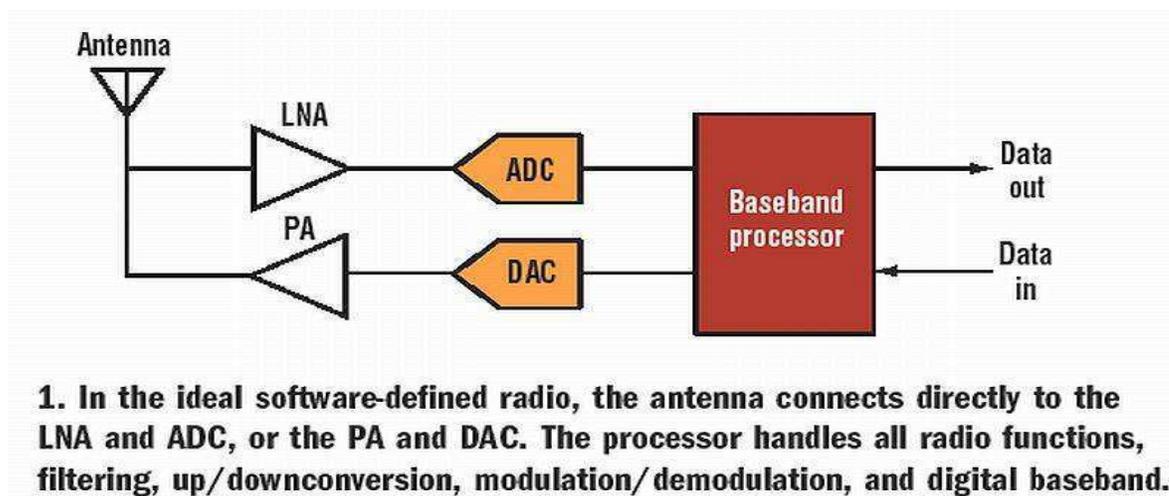
Un concepto de SDR que establezca la definición desde un enfoque de mercado donde muestra la ampliación de los horizontes de SDR: “Radio Definido por Software es un término difuso, es la habilidad de modificar un radio para actuar de un modo diferente, mejorar su comportamiento y ser utilizado para más de un propósito”. (Rubestein, 2010-2011)

#### **7.1.14. NIVELES DE SDR**

Pretendiendo dar una clasificación de los equipos radiales, el *Wireless Innovation Forum* definió 5 niveles de utilización de software dentro del radio para controlar o realizar funciones de la capa física, determinando la frontera entre hardware y software en el equipo (MATLAB, 2011)

- Nivel 0. Radio construido utilizando hardware solamente, no puede cambiarse por software. En este nivel no hay software, ni en el control ni en la realización propia de las funciones de operación del radio.
- Nivel 1: Radio controlado por software con limitaciones en cuanto a funciones controlables. Se controlan algunas como nivel de potencia, interconexiones, pero nunca modo o frecuencia.
- Nivel 2: Una parte significativa del radio se puede configurar mediante software. Existe control de software de ciertos parámetros como frecuencia, modulación, generación/detección de forma de onda, seguridad, etc. La etapa de RF permanece en hardware y no puede ser reconfigurada. Es importante destacar que el software en este tipo de radios sólo controla funciones que están implementadas de modo físico dentro del radio, a hardware únicamente.
- Nivel 3: Este nivel agrupa todos los radios en los que al menos una de sus funciones está definida por software, incluyendo lo que se conoce como Radio Definido por

Software Ideal (ISR) donde la frontera entre la parte configurable y la no configurable se encuentra muy cercana a la antena, y la etapa final de RF es configurable. Se puede decir que el ISR es completamente programable. (Forum, 2017)



*Figura 19. Figura SDR (Mario, 2012)*

La posibilidad que nos brinda el SDR es mostrar de manera gráfica, en un PC, un porcentaje extenso de la banda ancha que se está operando. Si imagináramos que estamos utilizando una banda de 80 Mts, se evidencia en la figura 1 toda la actividad que ocurre, no solo en el sistema tradicional, si no que se evidencia todo un ancho de banda de 100 o 200 Khz, es decir se podrá evidenciar si hay actividad en Telegrafía, en modos digitales, en porciones de DX, si son señales fuertes, débiles, constantes o con desvanecimiento.

Por tal motivo una de las grandes ventajas que tiene el SDR es que permite que las nuevas técnicas radiales tengan la posibilidad de detectar en forma automática “el tipo de modulación que intentamos escuchar, que por lo general se trata de SSB, AM, FM, CW, algún

modo digital (SSTV) y, en algunos casos, señales COFDM (Coded Orthogonal Frequency-Division Multiplexing).” (Mario, 2012)

- Nivel 4: Los equipos pertenecientes a esta categoría reciben la denominación de Ultimate Software Radio (USR) y constituyen una etapa superior al ISR puesto que no solo son completamente programables, sino que además son capaces de soportar un amplio rango de funciones y frecuencias simultáneamente. Dentro de este nivel se encuentran los teléfonos celulares con soporte para varios estándares.

### **7.1.15. EVOLUCION DEL SDR**

#### ***7.1.15.1. La evolución de la arquitectura de radio definida por software.***

A mediados de la década de los 90 el SDR ha presentado posibles soluciones al diseño del futuro para los receptores RF. De igual manera con el pasar del tiempo el SDR ha prometido soluciones por bastantes años pero solo en el año 2014 se convirtió en un costo muy bajo y de fácil acceso a nivel de escritorio. En el pasado el SDR lo asociaban comúnmente con las aplicaciones usadas por las fuerzas militares que tenían como finalidad la investigación de los enemigos y de igual manera se vendía la idea de ser una aplicación con alto costo y de difícil acceso.

De igual manera resulta útil revisar la arquitectura del SDR "definitiva", junto con las implementaciones prácticas más comunes y comprender los factores tecnológicos involucrados. (Bob Stewart, 2015)

### **7.1.16. LA ÚLTIMA ARQUITECTURA SDR**

La estructura y composición más simple del SDR está comprendida de una sección de RF (antena, amplificadores y filtros) y un convertidor analógico a digital de muy alta velocidad (ADC) y convertidor digital a analógico (DAC), par integrado con un procesador potente DSP

y/o sistema informático, como se puede evidenciar en la Figura 14. Las muestras se pasan dentro y fuera de la sección DSP a través de la ADC y DAC, respectivamente. Recordando el teorema de Nyquist (es decir, que es necesario muestrear una señal a mayor de dos veces el ancho de banda de la señal para retener toda la información), muestreando, digamos,  $f_s = 4\text{GHz}$ , produciría un espectro baseband de 0 a 2GHz, o  $FS/2$ . Así, el SDR sería capaz de transmitir y recibir todas las frecuencias hasta 2GHz, con la modulación y demodulación emprendida en el dominio digital.

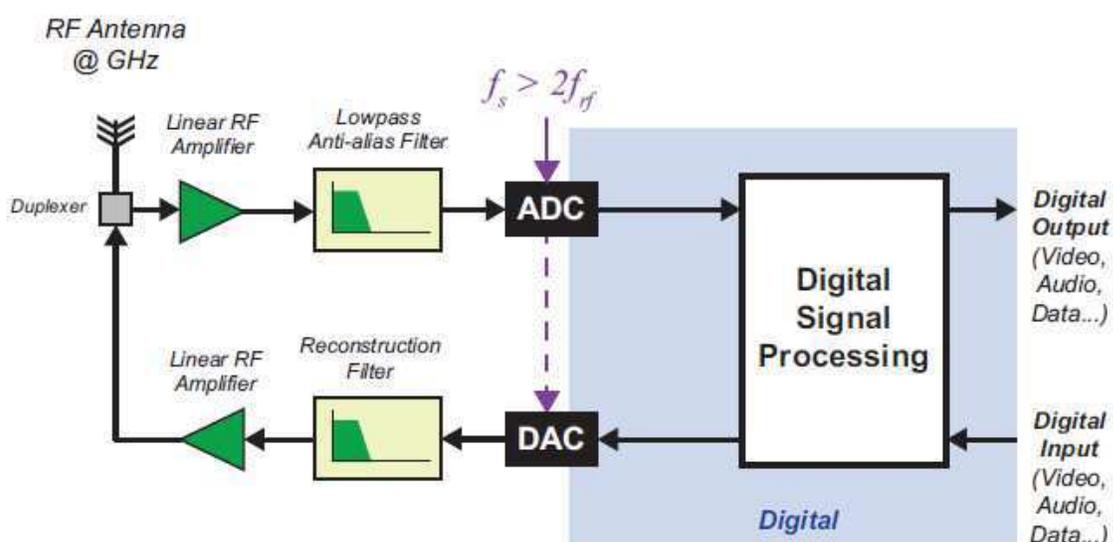


Figura 20. Los componentes del SDR simple, conceptual, "Ultimate" que en teoría digitalizarán el espectro de 0Hz a 2GHz (o  $FS/2$ ) (Bob Stewart, 2015)

Para desarrollar plenamente un Sistema receptor de radio, es necesario contar con las capacidades de la antena RF y de banda ancha, para lograr ilustrarse en asuntos de ingenierías que se encontrarían asociados al funcionamiento de una gama de frecuencia amplia y/o extensa. Por tal motivo la garantía que se utiliza en los componentes analógicos se determinan importantes. Tomando que todo se encuentre en orden, entonces se podría percibir que desde el punto de vista de la teoría de DSP, si se podría digitalizar la señal de RF en estas frecuencias de muestreo, el resto del receptor puede ser implementado puramente en software; dicho software

sería requerido para funcionar en los procesadores de alta velocidad, o quizás incluso en FPGA de gran alcance (eg. para las señales extensas, complejas basadas en estándares tales como LTE). Estos dispositivos realizan toda la aritmética algorítmica de DSP requerida, que son sobre todo multiplicaciones adiciones, según generalmente para DSP.

En algún momento pronto, los ADCS y Dacs de alta velocidad necesarios para la interfaz en las tasas de muestreo de multi-GHz estarán disponibles, y por lo tanto el sistema indicado en la figura 20 será fácilmente realizable. Retomando el proceso de las evoluciones tecnológicas en la década de los 90 específicamente a finales de esta misma, Un DAC/ADC que funcionaba en JS-100KHz con resolución de 16 bits fue la última (y bastante costosa) tecnología. Ahora, en 2015, los dispositivos que funcionan en MHz con la resolución de casi 14 bits están ampliamente disponibles a un costo razonable de empresas como Analog Devices Inc. Los muestreados que operan a frecuencias de GHz están disponibles en una resolución de 8 bits , pero estos tienen un costo bastante alto y aun no poseen un alto consumo. (Bob Stewart, 2015)

#### **7.1.17. ARQUITECTURAS SDR**

La primera generación de ' radios digitales ' re apareció de nuevo a mediados de la década de los 90's. Como se ilustra en la figura 21, la parte analógica de esta arquitectura de radio son señales que se controlan desde su equipo si se utiliza un oscilador local (lo), y luego, utilizando un segundo lo, más aparecen la señal IF a banda base. La señal del banda base fue muestreada y digitalizada usando un ADC (a una tasa de no más de unos pocos 10 ' s de kHz), y el DSP fue utilizado para realizar las etapas finales de procesamiento para recuperar la información transmitida. Los teléfonos móviles de segunda generación de los años 90-probablemente se utilizo esta arquitectura. Es de importancia recordar la primera generación de telefonos móviles a

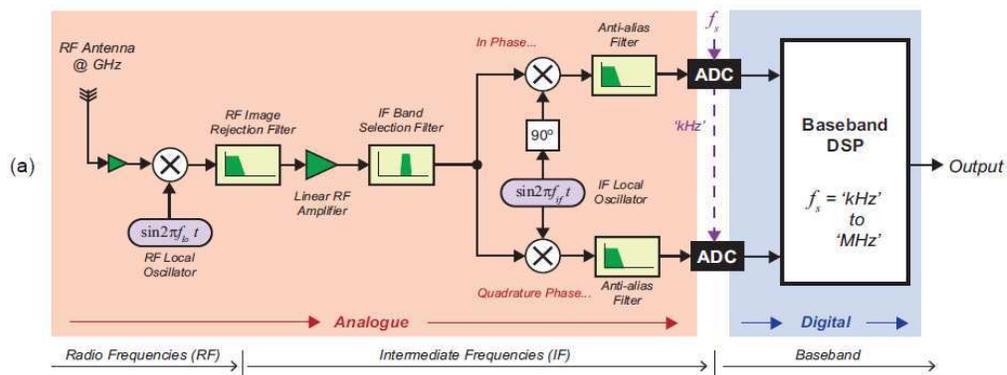
comienzos de los 90 los cuales en su totalidad eran analógicos, utilizando los viejos amperios y estándares de TAC.

En la siguiente generación de radios digitales, los cuales surgieron en la década de 2000, los procesos de muestreo y digitalización comenzaron a realizarse en algunos dispositivos en IFS. Según lo manifiesta en el libro *Software Defined Radio Using MATLAB & Simulink and the RTL-SDR* las pruebas y muestreos hechos se utilizó un IFS de alrededor de 40 MHz (por ejemplo), el cual podría ser apoyado por un ADC que muestreado en, 125 la primera etapa de DSP de esta arquitectura implicada usando un distribuir digital directo (DDC) para cambiar de puesto si señales a banda base usando demodulación y

demodulation filtrado, como se muestra en la figura 15 (b).

El proceso adicional de DSP entonces fue realizado una vez que la señal estaba en banda base. En esta arquitectura, se implementó más funcionalidad en el dominio digital, dando mayor flexibilidad a los SDR.

Por tal motivo lo que se pretende evidenciar es que las señales del RF directamente según lo ilustrado en la figura 15(c), y subconvertir ellos de frecuencias del RF a la banda base en un solo paso, usando DSP. Esto es posible en la actualidad porque ahora se puede evidenciar mediante una muestra en el orden de GHz este es un cambio hacia la arquitectura del SDR mencionada en la última arquitectura del SDR.



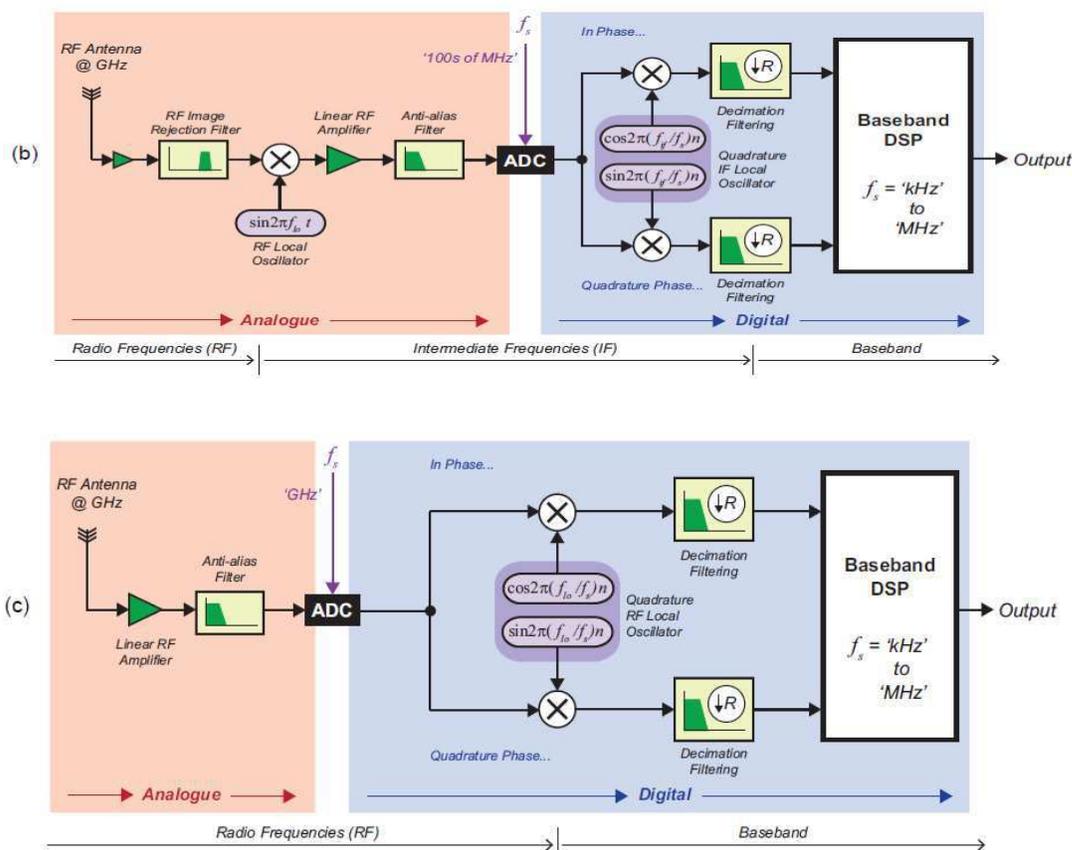


Figura 21. La evolución de SDR: a medida que el índice de muestreo del ADCS aumenta y se acercan más a la antena (a) la radio digital de banda base evoluciona a (b) la radio digital if, y luego a (c) la radio digital RF (cero-IF) (Bob Stewart, 2015)

### 7.1.17.1. La arquitectura RTL-SDR

La arquitectura del RTL-SDR se puede evidenciar en la figura 21 (b). Tiene un proceso de dos etapas de la demodulación: RF a si en hardware análogo; y si a banda base implementado digitalmente. Principalmente, el control sobre el proceso de demodulación puede ser ejercido a través del software, lo que permite seleccionar la banda deseada de frecuencias de RF.

Haciendo relación nuevamente a la figura 21 (b), la salida del RTL-SDR es equivalente a las dos entradas a la salida del bloque de banda base DSP por el RTL-SDR, estas muestras IQ pueden ser traídas a MATLAB y Simulink a través del paquete de soporte de hardware RTL-

SDR, donde se implementa la última etapas del receptor SDR. La salida de las muestras por el RTL-SDR se encuentran en un formato de punto fijo de 8 bits, pero los sistemas diseñados en MATLAB o Simulink se pueden implementar usando aritmética de punto flotante. Una vez hecha la salida por el RTL-SDR, estas muestras IQ pueden ser traídas a MATLAB y Simulink a través del paquete de soporte de hardware RTL-SDR-donde se pueden crear diseños para la sección "base banda DSP" para implementar las etapas final del receptor SDR. La salida de las muestras por el RTL-SDR está en un formato de punto fijo de 8 bits, pero los sistemas diseñados en MATLAB o Simulink se pueden implementar usando aritmética de punto flotante. (Bob Stewart, 2015)

#### **7.1.18. RTL-SDR HARDWARE**

El hardware de USB de RTL-SDR se dio origen en los receptores del grado del consumidor DVB-T, diseñados para permitir que los usuarios vean DTV en una computadora. Estos receptores no fueron diseñados ni concebidos originalmente para que fuesen utilizados como DEG programables genéricos. De igual manera los dispositivos podrían ser colocados en un modo de prueba, que esencialmente omita la etapa de decodificación DVB, lo que permite que el dispositivo para sintonizar la gama 25 a 1,75 GHz, produciendo RAW, de 8 bits de datos IQ de muestras a una velocidad programable de muestreo de banda básica.

#### **7.1.19. OPERACIÓN DEL DISPOSITIVO**

Los dos componentes principales usados en los receptores de RTL-SDR DVB son un sintonizador de DTV (el Rafael micro R820T (Kadman, 2013) es el más comúnmente utilizado, aunque algunos RTL-DEG utilizan el Elonics E4000 (NooElec Inc, s.f.) y el Realtek RTL2832U DVB-T codificado múltiplex de división de frecuencia ortogonal (COFDM) demodulador. De acuerdo con los estudios realizados estos demoduladores podría ser colocado en un "modo de

prueba". En este modo, los dispositivos DVB actúan como si las radios digitales (como se ilustra en la figura 21(B), esencialmente se convierten en ADC de alta velocidad, y la salida de datos en cuadratura RAW con una resolución de 8 bits. Después de este descubrimiento, se acuñó el nombre RTL-SDR, que se refirió al que el RTL basó DVB receivers podría ser utilizado como SDR.

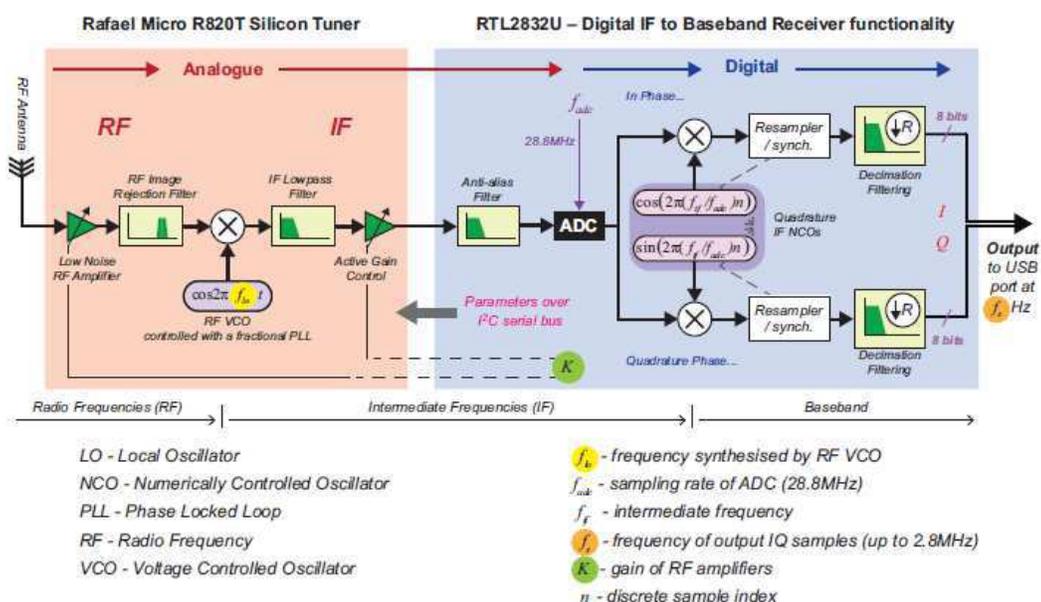


Figura 22. (Bob Stewart, 2015)

En la figura 22 se puede evidenciar un diagrama de flujo de procesamiento de señales de las etapas principales que se llevan a cabo en el RTL-SDR. Las señales de RF que ingresan al sintonizador están abajo convertidores a un bajo-IF se usa un oscilador controlado por voltaje (VCO). El VCO es programable, y es controlado por el RTL2832U sobre un interfaz integrado del circuito (I2C). Después de la etapa de control de ganancia activa (AGC), que ajusta dinámicamente la amplitud de la señal de entrada para adaptarse al rango operativo del dispositivo, la señal de IF entonces requiere ser derribada a banda base. El método clásico de hacer esto es pasar la señal IF a través de un filtro de suavizado, muestra la salida con un ADC, y

el subconvertir a banda base utilizando cuadratura osciladores controlados numéricamente (NCOs). (i.e. un seno y un coseno oscilando en la frecuencia IF).

### 7.1.20. QUE ES UN USRP

El USRP hardware es una gama de los DEG programables basados en FPGA TX/RX realizados por Ettus Research. Con velocidad de hogh (y alta resolución) ADCS y DACS, estos dispositivos de alta calidad toman un enfoque modular, y le permiten utilizar una selección de transmisor, receptor y tranceptor RF placas que se optimizan para trabajar en varias frecuencias diferentes. De forma predeterminada, la FPGA del lado TX de la radio USRP está configurada para realizar la conversión digital (DUC) para convertir las muestras de IQ de banda base (transferidas al dispositivo desde un equipo host) a un if. Las muestras se convierten en señales continuas utilizando un DAC, y luego se mezclan con un portador de RF en cuadratura, realizando la modulación AM-DSB-SC como se muestra en la figura 23. La salida de la señal del RF por el USRP hardware se puede generalizar como sigue.

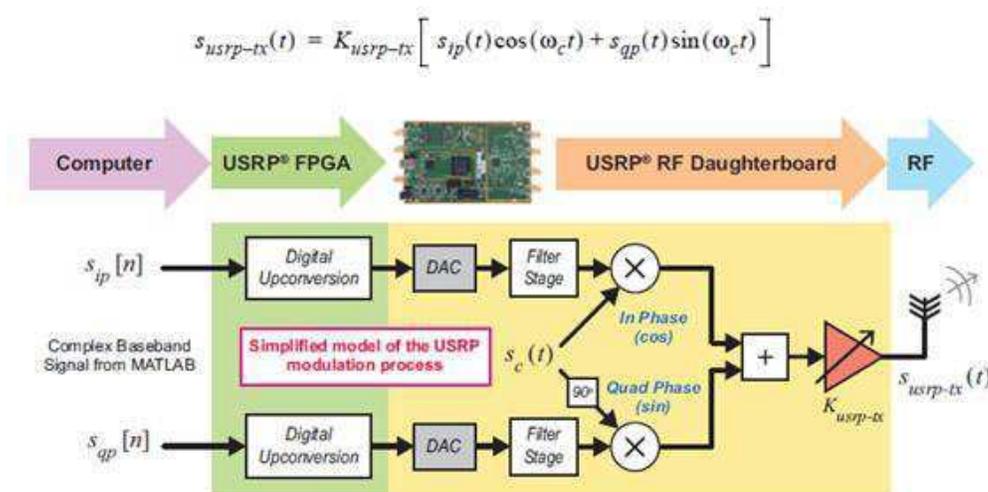


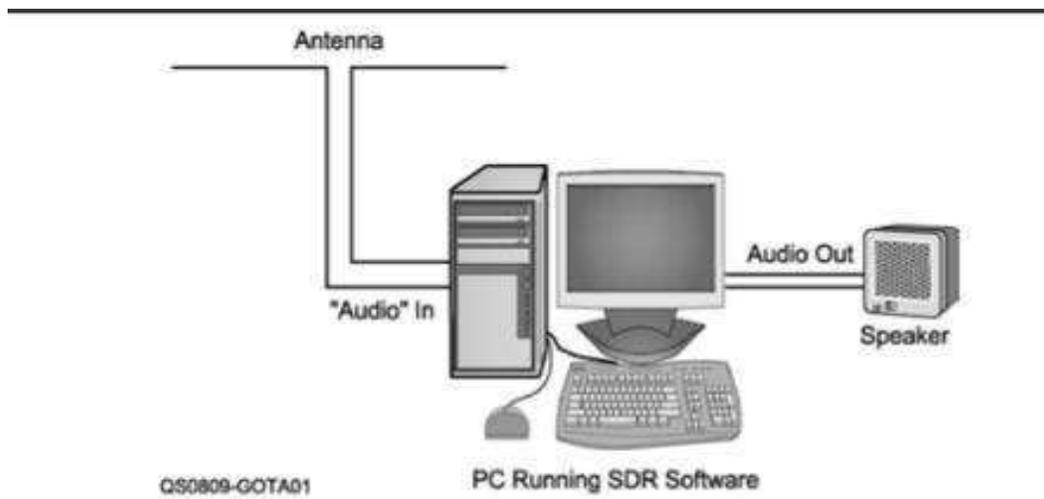
Figura 23. Diagrama de bloques destacando los procesos principales que en el TX lado de la USRP FPGA y RF placa. (Bob Stewart, 2015)

Una de las ventajas de utilizar una radio USRP es que, al igual que el RTL-SDR, es totalmente compatible y soportado con MATHWorks. La descarga e instalación de su paquete de soporte de hardware instalará el controlador USRP hardware (UHD) en el equipo, y habilitará la interfaz con el dispositivo directamente desde MATLAB y Simulink. Este complemento trabaja con un número de los modelos más populares del hardware de USRP, apoyando el USB 3,0 GbE y 10 conexiones del dispositivo de GbE. (Bob Stewart, 2015)

### 7.1.21. APLICACIONES DE SDR

Por el sutil modo en que la idea de SDR se fue introduciendo en la fabricación de equipos de comunicaciones por radio se hace difícil identificar un inicio cronológico del mismo. No obstante, su entrada en el ámbito comercial se define en 3 pasos del tiempo

#### 1. Amateur SDR



*Figura 24. Aplicación Amateur SDR (Azurdia, 2016)*

## 2. Escanear Espectro Electromagnético

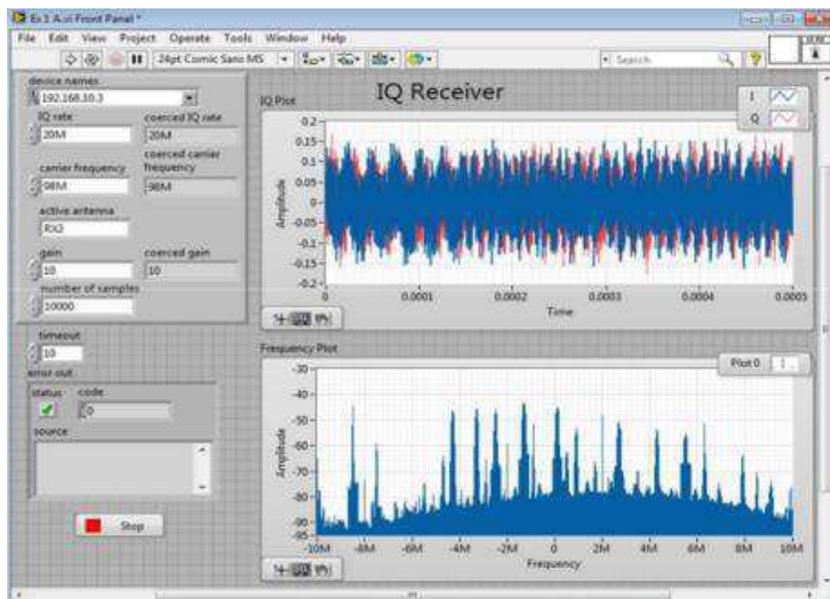


Figura 15. Escaner Espectro Electromagnético.

## 3. Implementación y Simulación de llamadas Móviles

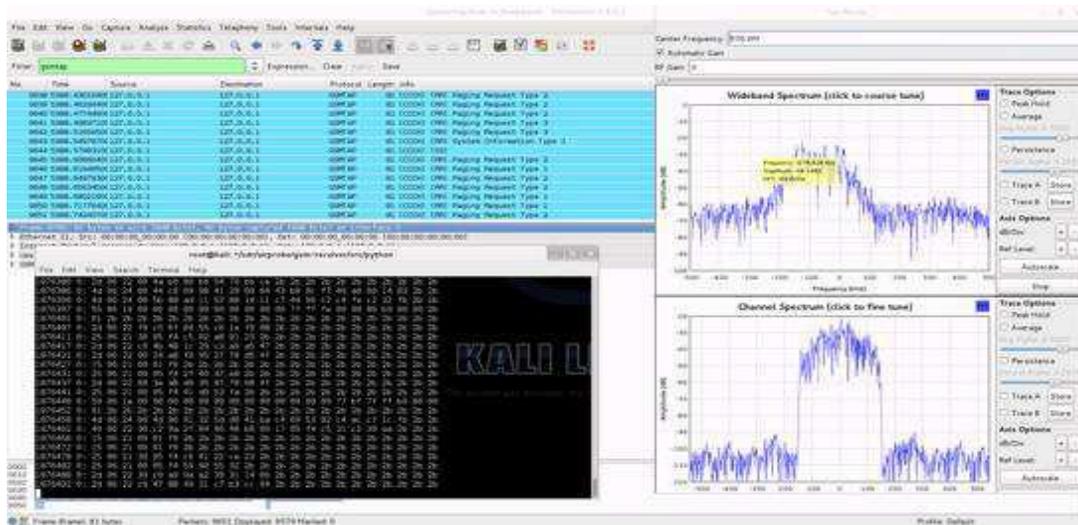


Figura 25. Implementación y Simulación de Llamadas Móviles.

#### 4. Detención y Uso de las Señales GPS



*Figura 26. Detención y Uso de las Señales GPS*

#### 7.1.22. SDR IMPLEMENTADOS

Desde hace unos años se han venido desarrollando algunos radios basados en SDR por parte de aficionados, empresas privadas y desarrolladores libres. Ejemplo de esta afirmación son:

##### **7.1.22.1. Funcube dongle**

Como se puede evidenciar en la figura 27 que el Receptor de radio, o segmento de tierra, desarrollado por AMSAT-UK como parte de un proyecto educativo de colaboración global que sirve como receptor de señales de satélites Funcube alrededor del mundo. Se le puede variar la frecuencia entre 64 y 1,700 MHz. Su precio está cercano a los \$200 USD. (AMSAT-UK, 2011)



*Figura 27. Funcube Dongle (AMSAT-UK, 2011)*

### 7.1.22.2. *PM-SDR*

Como se puede evidenciar en la Figura 28. El PM-SDR fue desarrollado por Martin Pernter, utiliza el software libre WinRad de Jeffrey Pawlan, así como otros programas para funcionar en Linux. Se conecta por USB y puede configurarse en el rango de 100kHz - 50MHz. (Pernter., s.f.)

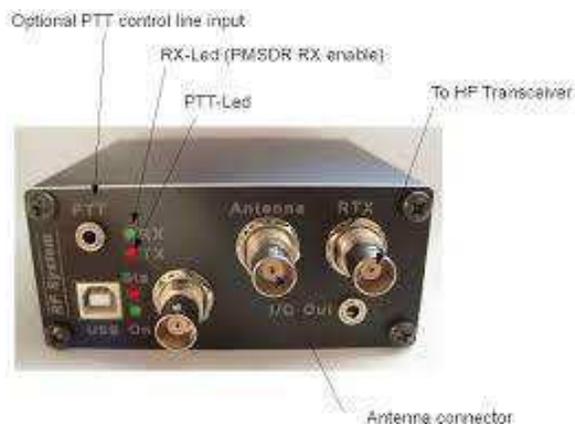


Figura 28. *PM-SDR*. (Kanga., 2011)

### 7.1.22.3. *Finningley 80M*

Como se puede evidenciar en la figura 29. El Finningley 80 M fue creado por Microwave Roundtable y reproducido por Kanga Products. Es un kit para la familiarización con SDR y su precio es de más de \$25 USD. (Kanga., 2011)

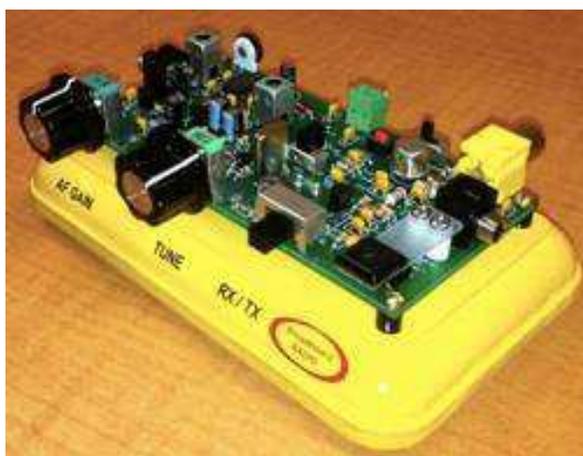


Figura 29. *FINNINGLEY 80M* (Kanga., 2011)

#### 7.1.22.4. *Excalibur*

De acuerdo a la figura 30 el modelo Excalibur fue creado por WiNRADiO, es un receptor de onda corta entre 9KHz y 50 MHz. Incluye un analizador de espectro de tiempo real de 50MHz de ancho de banda y su sensibilidad ronda los 0.35  $\mu$ V. Su interfaz de software es una de las más completas creadas para SDR. [11] (WiNRADiO., 2011)



Figura 30. Modelo Excalibur

#### 7.1.22.5. *Universal Software Radio Reripheral (USRP)*

Tarjeta Madre desarrollada por GNU Radio a la que se le pueden incorporar otras tarjetas para darle funcionalidad. Es capaz de manejar muchas características que pueden ser consultadas en. Tiene una versión mejorada, USRP2. Es uno de los SDR más potentes que se pueden encontrar en la actualidad y emplea software libre. (Blossom., 2011)



Figura 31. Universal Software Radio Peripheral (USRP) (Blossom., 2011)

### 7.1.22.6. QSIR

Creado por QuickSilver, es el receptor SDR más completo que se puede encontrar en el mercado. Abundante en opciones y posibilidades de Figura 31 USRP fabricado por GNURadio. Figura 30 SDR EXCALIBUR de WinRADIO. a) Dispositivo SDR. b) Analizador de espectro de tiempo real del EXCALIBUR. RADIO DEFINIDO POR SOFTWARE 17 configuración, utiliza el rango de frecuencias de 10 KHz a 62.5 MHz, que puede ser ampliado hasta 500 MHz con el empleo de sobre muestreó. (LLC., s.f.)



Figura 32. Modelo QSIR (LLC., s.f.)

## 7.2. ESTUDIO DE LOS COMPONENTES GNU RADIO COMPANION

Según gnuradio.org, (2017) GNU Radio Companion (GRC) es una herramienta y un entorno de desarrollo que facilita el desarrollo de aplicaciones para GNU Radio mediante una interfaz gráfica. Esta herramienta facilita el diseño, ya que está basada en bloques y consiste en la creación de ficheros Python, cabe destacar que para el desarrollo de aplicaciones no es necesario saber programar en este lenguaje (Python) o C++, esto es debido a que GRC genera el código necesario para la aplicación que es construida gráficamente, es decir que GRC es básicamente un lenguaje de programación en entorno visual con el uso de código libre y es utilizado para el procesamiento de señales haciendo uso de las librerías o biblioteca de GNU Radio, varias de las grandes ventajas es en cuanto a que este entorno facilita la corrección de errores, la gestión de pruebas y en cuanto a la ejecución de cada una de las aplicaciones construidas con GRC. (Pinar & Murillo, 2011)

En primera instancia hay que señalar que para la realización de proyectos y creación de módulos utilizando el GRC hay que entender que esta herramienta está basada en interconectar los diferentes bloques usados, de una manera esquematizada y para esto según Pinar & Murillo (2011) los bloques se pueden agrupar de la siguiente forma:

- **Sources** (fuentes): Estos bloques detallan cualquier tipo de fuente, tales como: archivos de audio, generadores de señal, fuentes binarias aleatorias, micrófonos, etc.

- **Bloques de procesado de señal**: En este grupo se encuentran todos aquellos bloques que de alguna manera realizan algún tratamiento a la señal o señales con que se trabajada.

Algunos ejemplos podrían ser: amplificadores, filtros, remuestreadores, moduladores, multiplicadores, etc.

- **Sinks** (sumideros): Los bloques de este grupo son los que hacen parte del final del diagrama que se realiza en GRC, y son los que aseguran que la señal como destino final llegara a un fichero de cualquier formato o a una tarjeta de sonido. En este conjunto también encontramos los diferentes bloques que permiten visualizar las señales (*Graphical sinks*) como *FFT sink* (para visualizar la FFT de la señal en un punto), *Constelation sink* u *Oscilloscope sink* (para representar la señal en tiempo en cualquier lugar del diseño) entre otros.

En el anexo 1 como cumplimiento a uno de los objetivos del proyecto se ecuntra toda la documentación referente a GCR en español con el cual se logra dejar como producto del desarrollo del proyecto documentacion disponible en este idioma para el desarrollo de futuros proyectos.

## 8. REALIZACION DE PRACTICAS DE LABORATORIO CON SUS GUIAS

### 8.1. LABORATORIO 1: CONSTRUCCIÓN DE UN RECEPTOR DE RADIO FM EN GNU RADIO COMPANION.

Esta es la primera guía de las dos generadas para la realización de los laboratorios planteados en esta sección.

Para recordar, la radio definida por software SDR es el uso de procesamiento de señal digital para la implementación de funciones de radio con el uso de un periférico de “radio definido por software” este se conecta a una computadora a través de USB y permite transmitir o recibir señales en una amplia gama de frecuencias. Este curso se llama radio de software definido con RF porque es principalmente sobre la radio definida por software y estamos utilizando Hack RF, en gran parte del curso realmente se va a utilizar la radio GNU y la verdadera naturaleza de la base del curso es realmente acerca de los fundamentos del procesamiento de señales digitales.

En este laboratorio se va a usar GRC para crear una radio definida por software que permitirá escuchar una emisora de radio FM, siguiendo los siguientes pasos:

**Paso 0.** Al abrir el entorno GRC se encuentran dos bloques en espacio para el flujo como se muestra en la Figura 33, allí se debe entrar en las propiedades del bloque *Options* dando clic derecho sobre él y seleccionando propiedades, o simplemente dando doble clic sobre él y directamente nos abre la ventana propiedades, en este punto se pondrá el nombre del proyecto que para este caso será Lab1 y en el parámetro *Generate Options* que es en donde se establece el tipo de flujograma se selecciona **WX GUI** como se observa en la Figura 34, este será el tipo de librerías que se utilizaran para la interfaz gráfica de esta primera práctica. Para recalcar cada vez que se hagan cambios en las propiedades de los diferentes bloques se debe primero dar clic en aplicar y luego en ok.

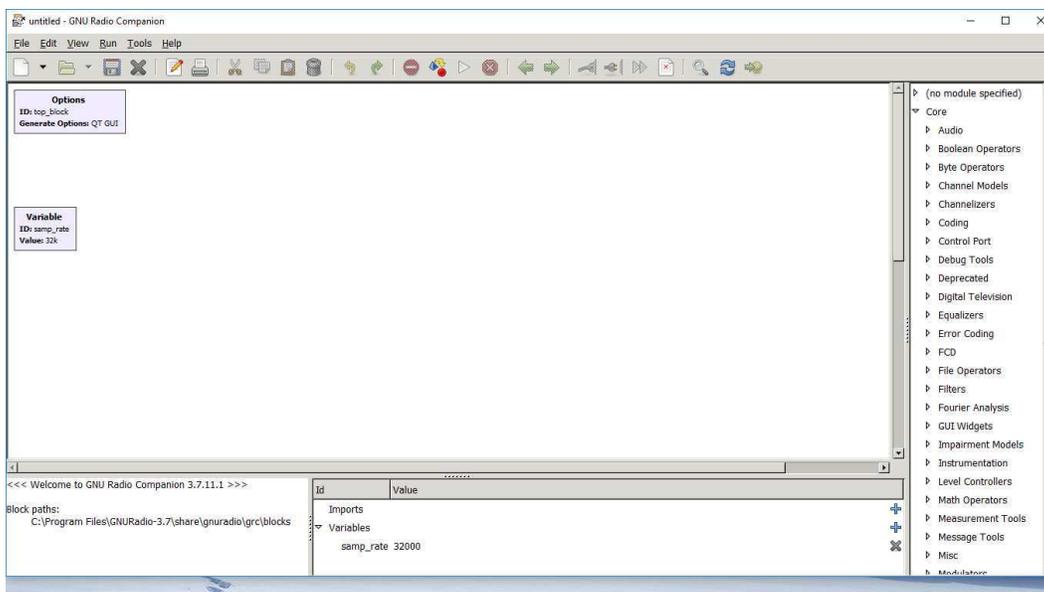


Figura 33. Paso 0.

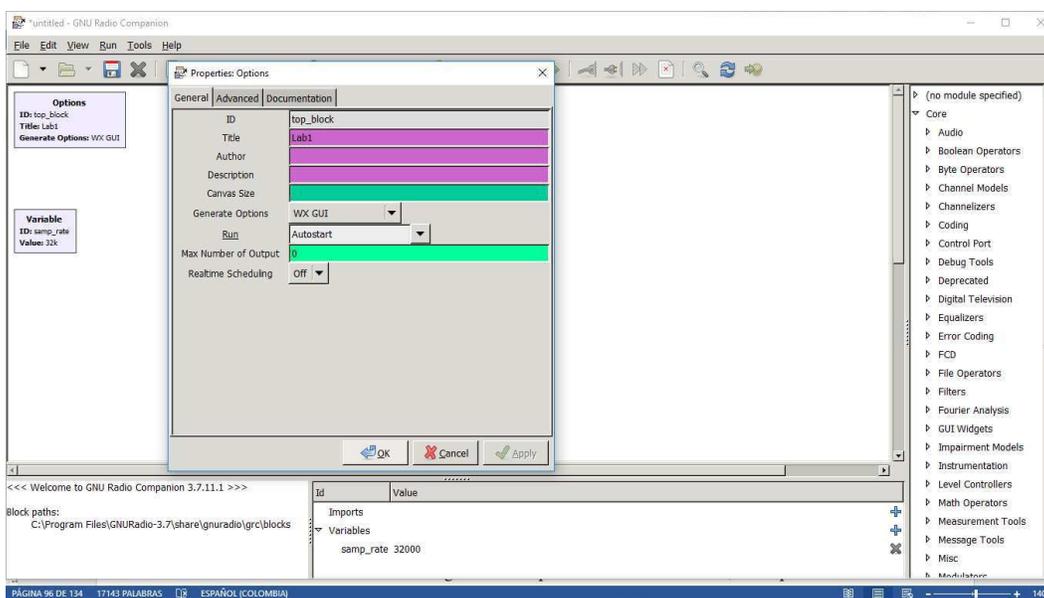


Figura 34. Paso 0a.

**Paso 1.** Ahora si en este paso se entra en el desarrollo del laboratorio como tal, se comienza tomando un bloque de la lista que está a la derecha, se puede realizar la búsqueda del bloque automáticamente dando clic en la figura con la lupa o buscarlo manualmente, el bloque a usar se encuentra dentro del módulo sources (fuentes), el que se utilizara es “*osmocom Source*”, este

bloque es una capa de abstracción permitirá la comunicación entre los diferentes dispositivos del hardware y GRC. En las siguientes figuras 35 y 36 se muestra gráficamente el paso 1.

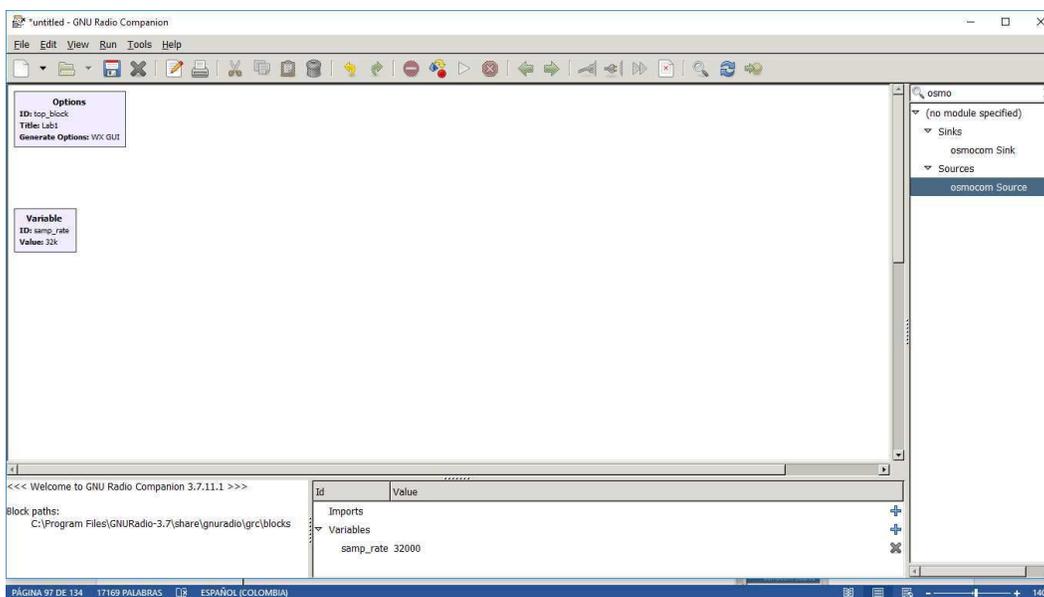


Figura 35. Paso 1.

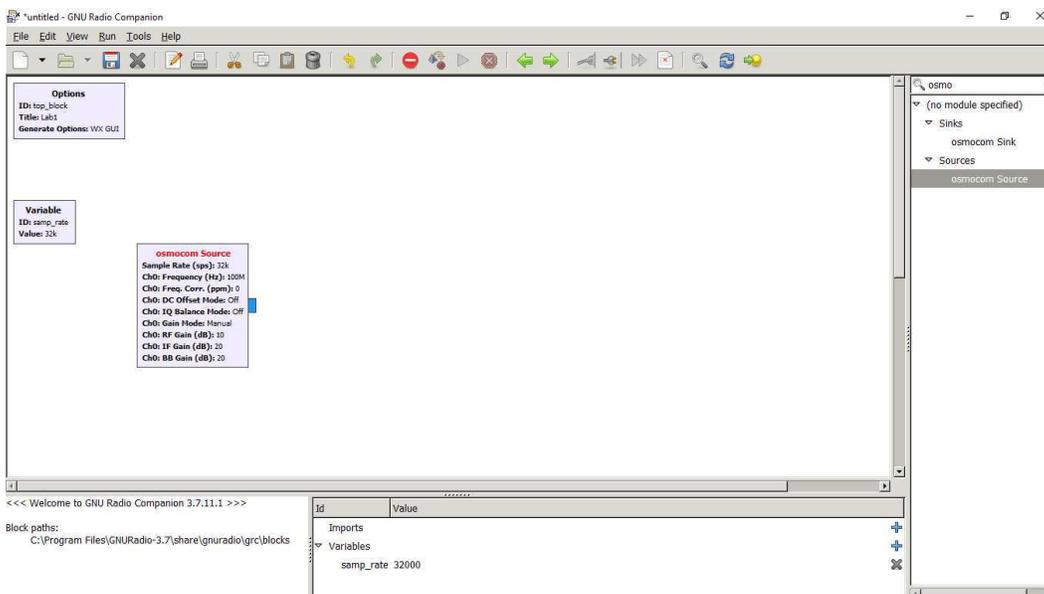


Figura 36. Paso 1a.

Este bloque es una fuente que produce una señal digital la cual será consumida por el próximo bloque del gráfico de flujo y lo que sucede básicamente en este punto es que la fuente le dice a la

Hack RF que debe encender en modo de recepción y que permita que entren las muestras por USB, estas van desde el Hack RF al bloque y luego hacia afuera con salida al siguiente bloque en el gráfico de flujo que se verá en el siguiente paso.

**Paso 2.** El siguiente bloque que se usara será “WX GUI FFT Sink”, este permite visualizar los componentes de frecuencia en una señal, es decir que actúa como una especie de analizador de espectro. En este punto se crea una línea dirigida entre los dos bloques que permitirá su conexión, para esto se da clic en la salida del bloque osmocomb y luego clic en la entrada del wx gui fft. Si se observa antes de interconectarlos estos bloques aparecían con su título en rojo y al realizar la conexión estos cambian a negro, esto quiere decir que sus entradas y salidas han sido satisfechas correctamente, así que ahora se tiene un gráfico de flujo completo el cual podrá ser ejecutado, pero realmente en este punto no será muy útil hasta que se hagan un par de cambios que permitirán corroborar que efectivamente la Hack RF está en modo recepción. En las figuras 37 y 38 se muestra el paso 2 en forma gráfica.

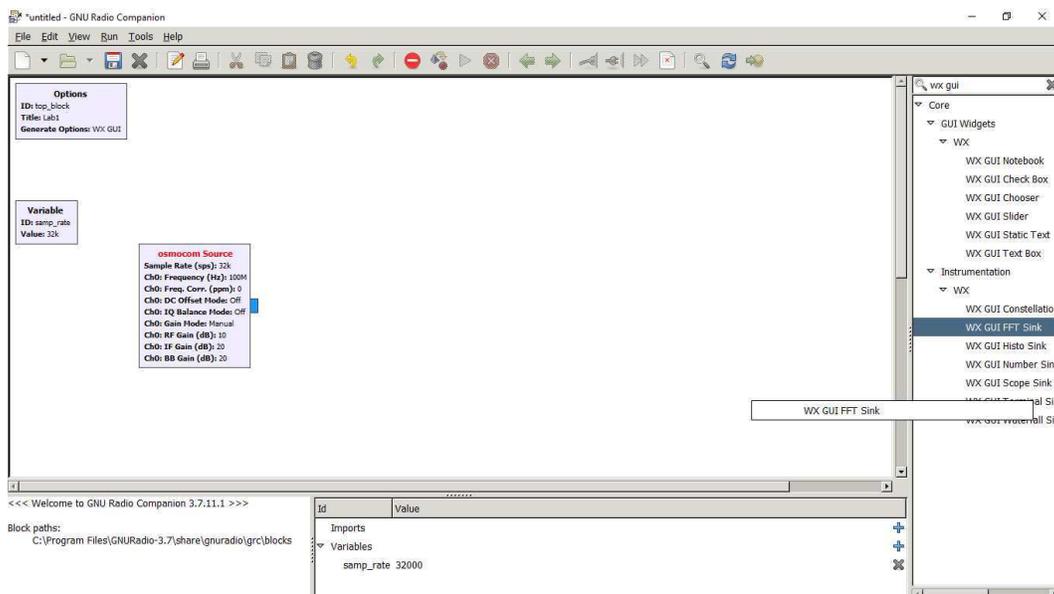


Figura 37. Paso 2.

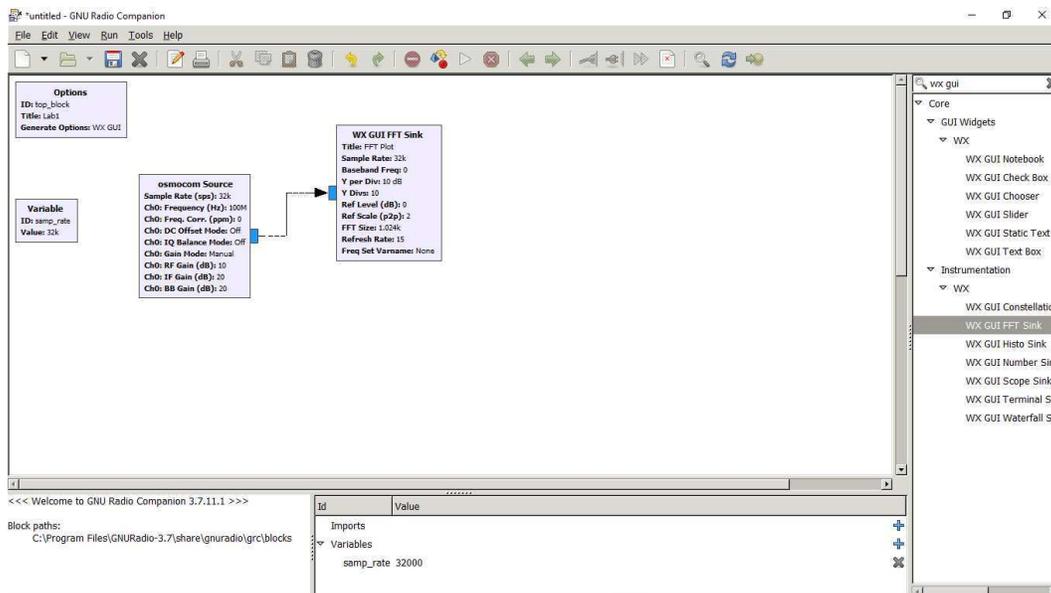


Figura 38. Paso 2a.

**Paso 3.** Ahora lo que se debe hacer es entrar a las propiedades del bloque variable como se muestra en la Figura 39 y en ID escribimos “smp\_rate” y en value ponemos 10’000.000 pero que por practicidad lo definimos como “10e6” muestras por segundo como lo indica la Figura 40. En este punto se puede observar que en los tres bloques se tiene ahora como valor Sample Rate 10 millones tal cual como se muestra en la Figura 41.

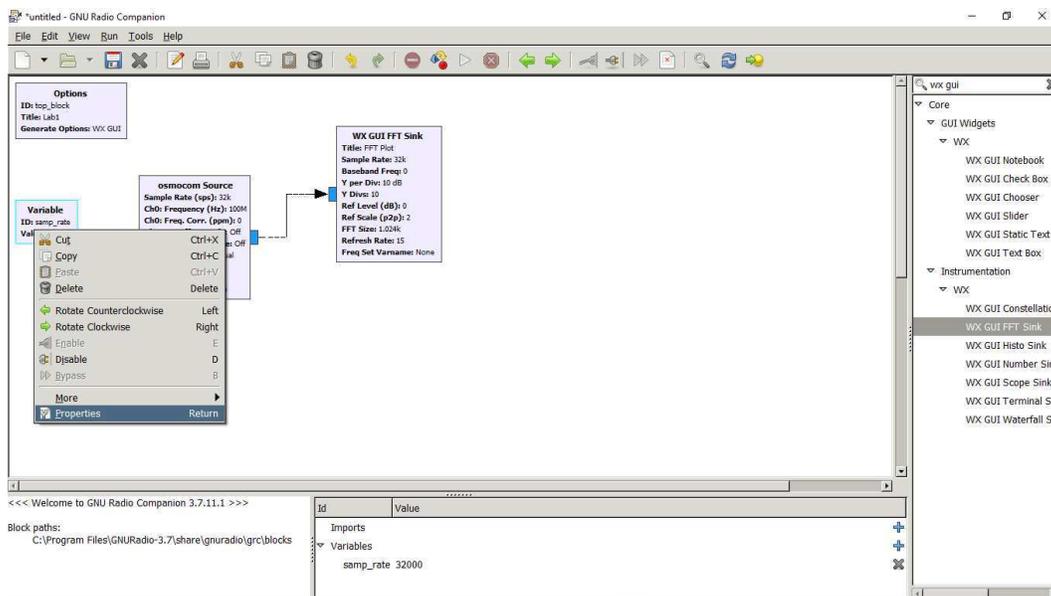


Figura 39. Paso 3.

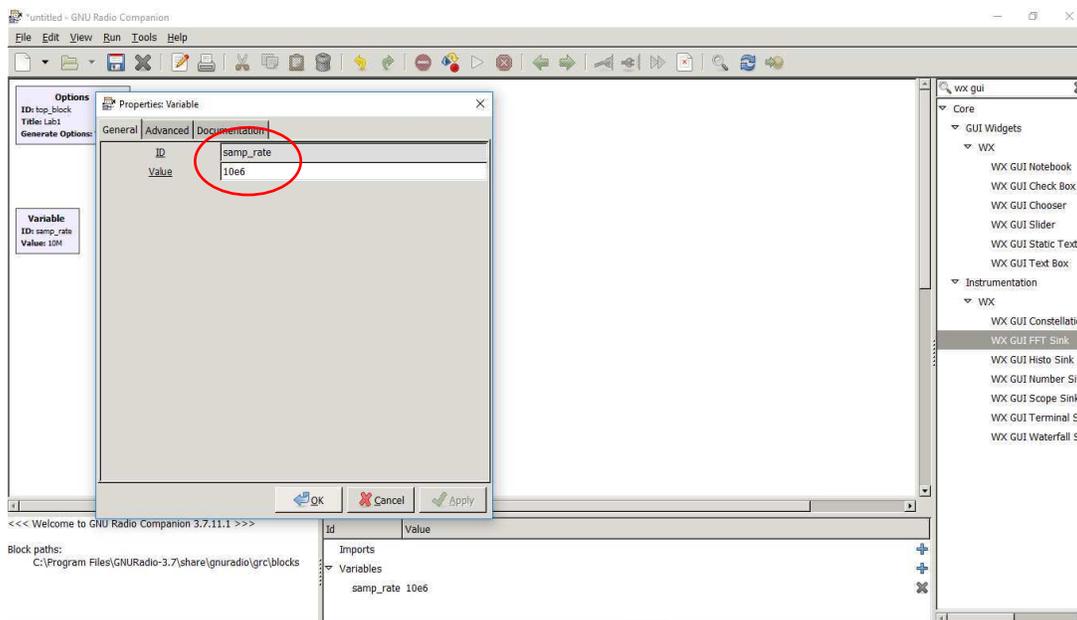


Figura 40. Paso 3a.

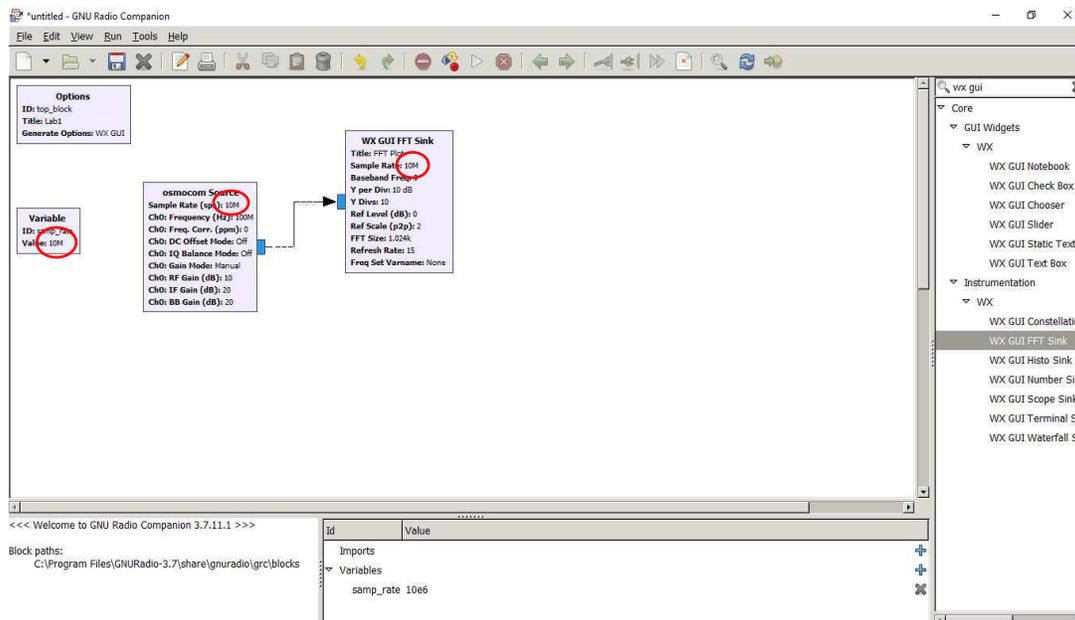


Figura 41. Paso 3b.

Aquí básicamente se cambió en el bloque de variable su tasa de muestreo la cual se configuro para muestrear la tasa de “samp\_rate” a 10 millones de muestras por segundo, es decir, que “samp\_rate” es el nombre de la variable que cambiamos. Esto quiere decir que el bloque variable

permite cambiar múltiples cosas en un solo lugar, cualquier cosa que se refiera como variable separada puede cambiarse simplemente cambiando este bloque.

**Paso 4.** Ahora se deben cambiar algunas otras cosas, vamos a las propiedades del bloque “osmocom Source” y se va a definir en el “Ch0: Frecuency” 102.9e6, esto quiere decir que se le está indicando a la Hack RF que reciba en primera instancia la señal con frecuencia de 102.9Mhz que para la ciudad de Bogotá corresponde a la emisora “Tropicana” y se cambia la ganancia de RF es decir “Ch0: Freq Corr” se establece en 0 para desactivar el amplificador frontal cerca de la antena del Hack RF que probablemente no se necesita, aunque en algunos caso si es necesario principalmente donde se necesita recoger una señal muy débil, pero las emisoras de radio Fm son señales muy fuertes a menos que vivamos lejos de cualquier transmisor. En la figura 42 se evidencia el paso 4.

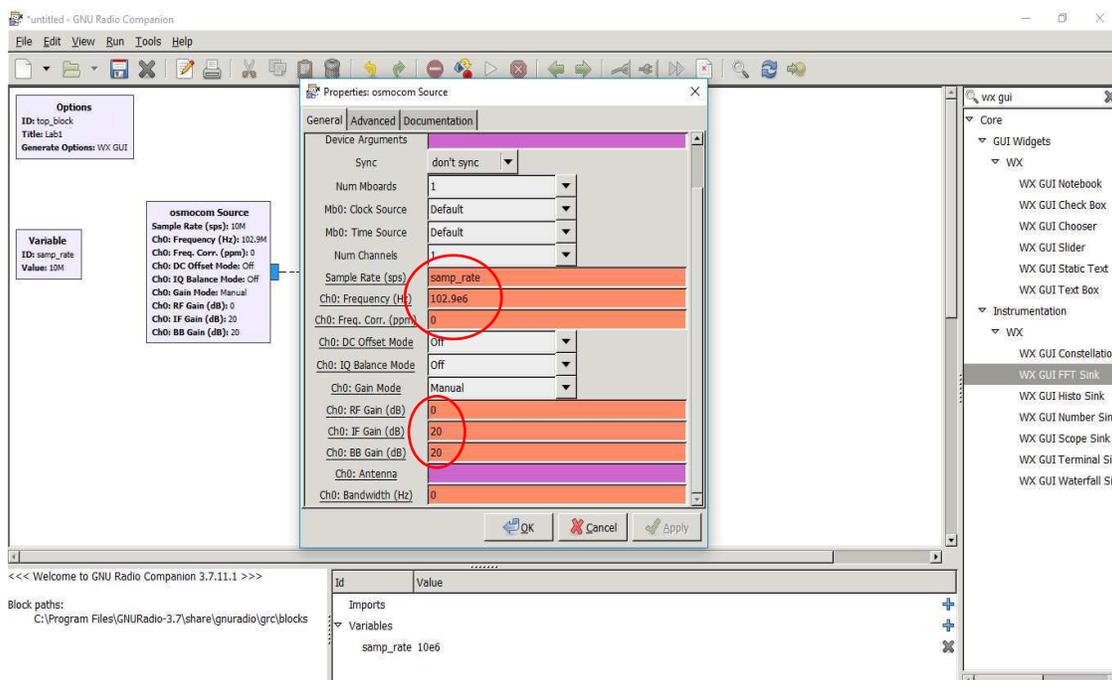


Figura 42. Paso 4.

**Paso 5.** Otro cambio que se debe hacer es en el bloque “FFT Sink”, vamos a propiedades y convertimos el promedio como se observa en la figura 43, lo que hace que sea un poco más fácil ver las estaciones de radio FM individualmente en la trama pfft.

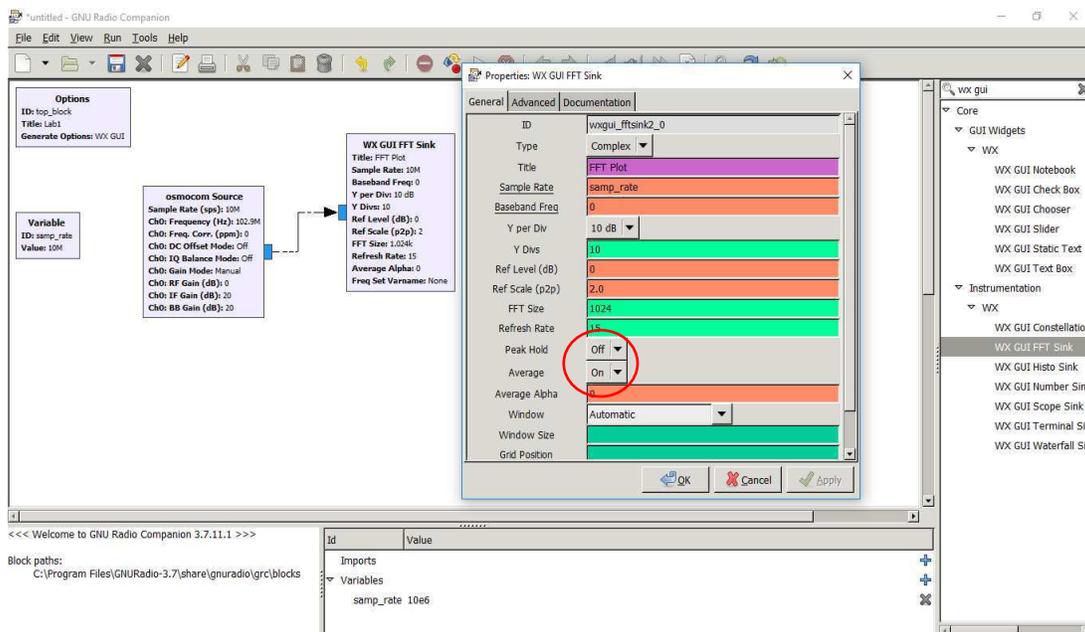


Figura 43. Paso 5.

En este punto se tiene un gráfico de flujo completo, pero aun no permite escuchar la radio FM, sin embargo permite ejecutar el gráfico de flujo para visualizar la señal, ver que algo este sucediendo y validar lo que se ha hecho hasta ahora antes de complicar el diagrama de flujo un poco más.

**Paso 6.** Se hace clic en el botón ejecutar como se muestra en la Figura 44 y lo primero que va a pasar es que va a pedir un nombre de archivo que para este caso se llamara Lab1 y que tiene una extensión holandesa GRC como se puede observar en la Figura 45. Este es un archivo de la representación gráfica del flujo por lo que contiene una lista de los diferentes bloques, su configuración y el cómo están conectados, estos son archivos que la GNU radio companion crea, luego automáticamente se crea un programa en Python el cual va a ser ejecutado, aquí se tiene

una ventana FFT Plot que muestra información de radio en vivo procedente de la antena a través de la USB, esta pasa a través del bloque “osmocom Source” y luego por el bloque “FFT Sink”, en esta ventana se pueden ver varios megahercios de ancho de banda y por lo que cada uno de estos picos principales es una estación de radio FM diferente como se observa en la Figura 46.

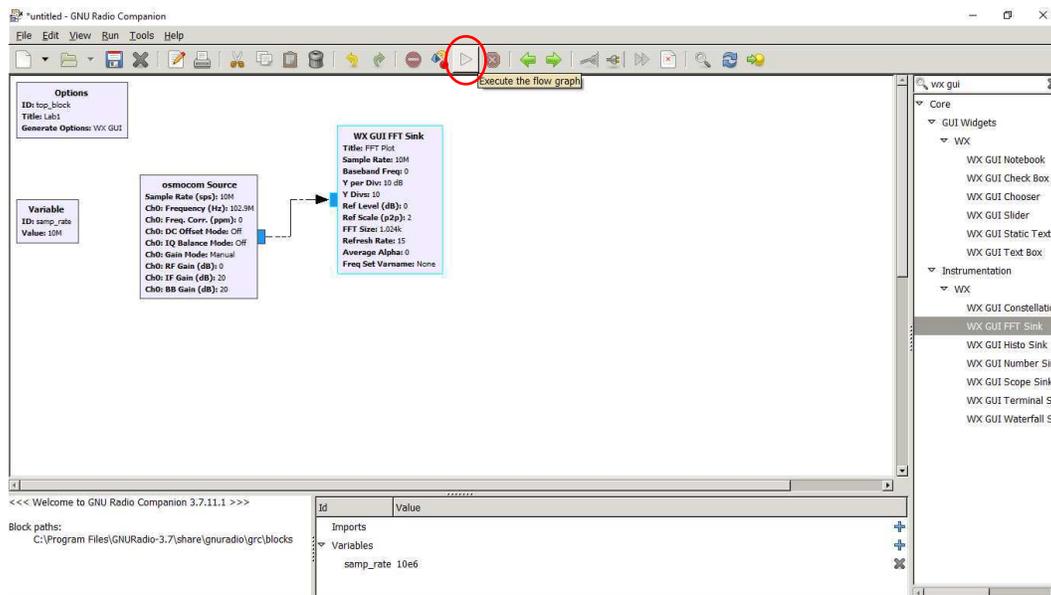


Figura 44. Paso 6.

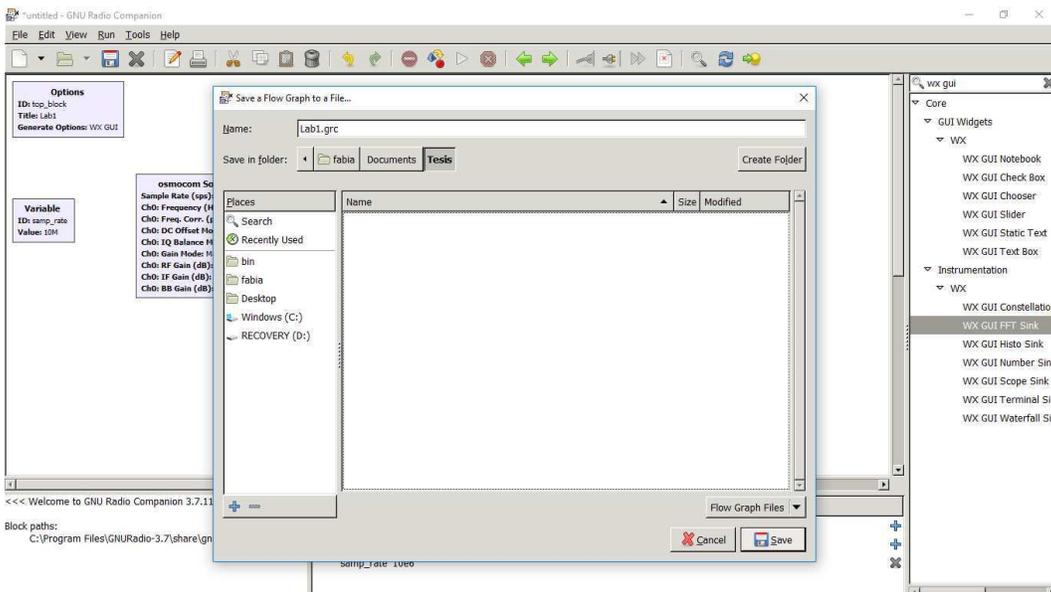


Figura 45. Paso 6a.

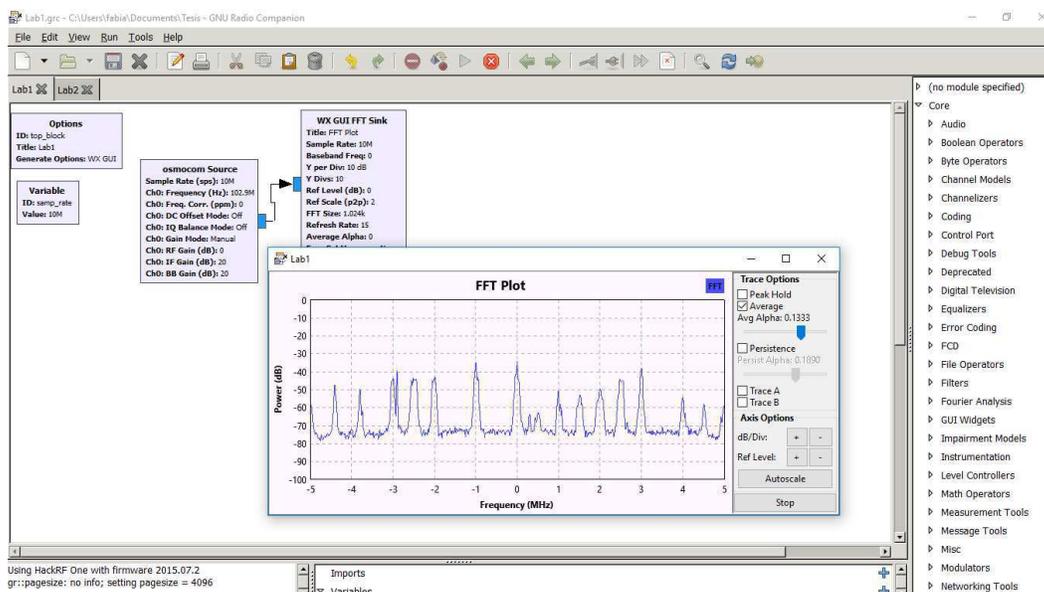


Figura 46. Paso 6b.

En este punto se puede ver que la frecuencia va desde -5 megahertz positivo a 5 megahertz y por ahora se puede pensar que en el punto cero esta la frecuencia central de 102,9 Mhz y así sucesivamente por lo que es sólo ajustar el gráfico de flujo para que lo haga más real.

**Paso 7.** Se finaliza el gráfico de flujo pulsando el botón de detención como se muestra en la Figura 47 en lugar de cerrar la ventana de primer plano, así se asegurará de cerrar la ventana correctamente y que vuelva a su estado predeterminado. Ahora se establece esta frecuencia en banda base aquí mismo y que sea algo distinto de cero en el “FFT Sink” es decir los 102,9Mhz, para esto se podría simplemente escribir la frecuencia en las propiedades del bloque, pero en lugar de tener varios bloques que todos se refieran a 102.9e6, se usara una variable, así que se creara una variable llamada `center_freq` para la frecuencia central. Lo primero es que en las propiedades del bloque “FFT Sink” se escribe `center_freq` como se muestra en la Figura 48, cuando se pulsa Ok se observa que el bloque FFT ha puesto su título en rojo, esto pasa porque la variable `center_freq` no existe aún, así que el siguiente paso es crear dicha variable.

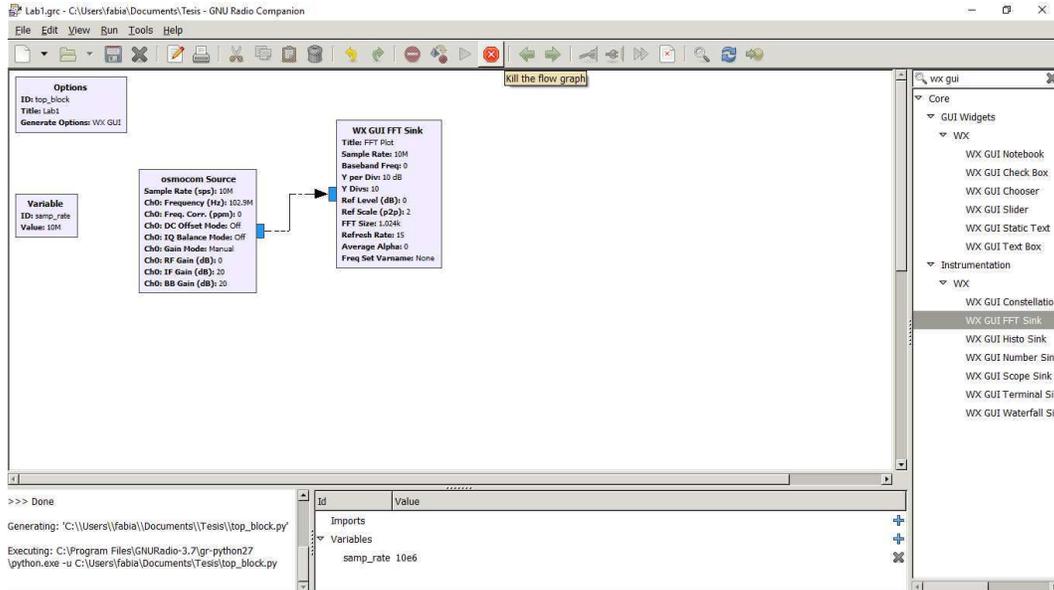


Figura 47. Paso 7.

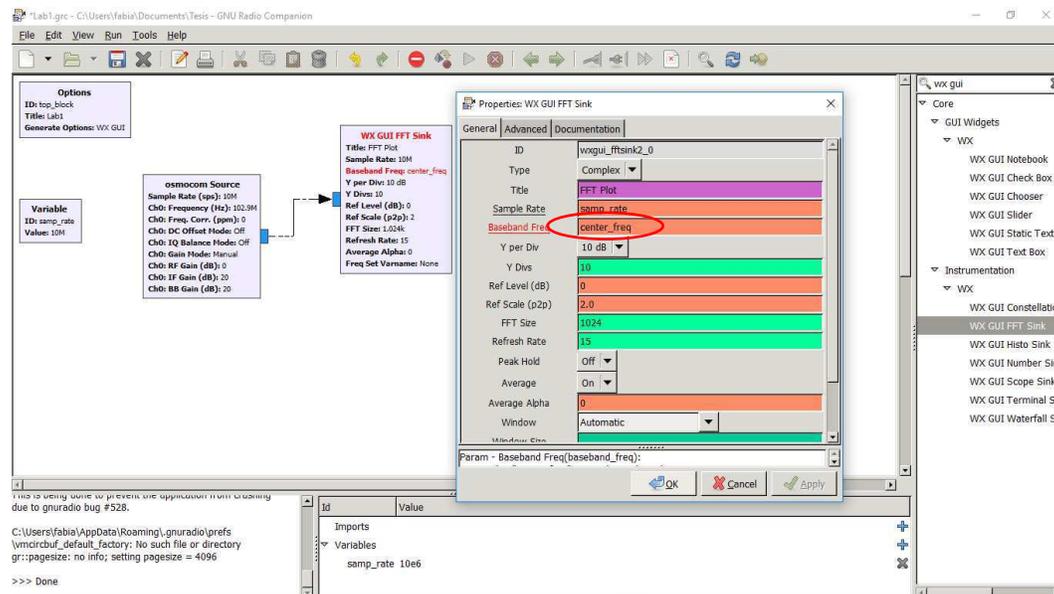


Figura 48. Paso 7a.

**Paso 8.** Se busca el bloque variable en la lista de módulos y lo arrastramos al entorno de trabajo como se muestra en la Figura 49, luego en sus propiedades llamaremos esta variable `center_freq` y en value se pondrá el valor de `102,9e6` como se muestra en la Figura 50.

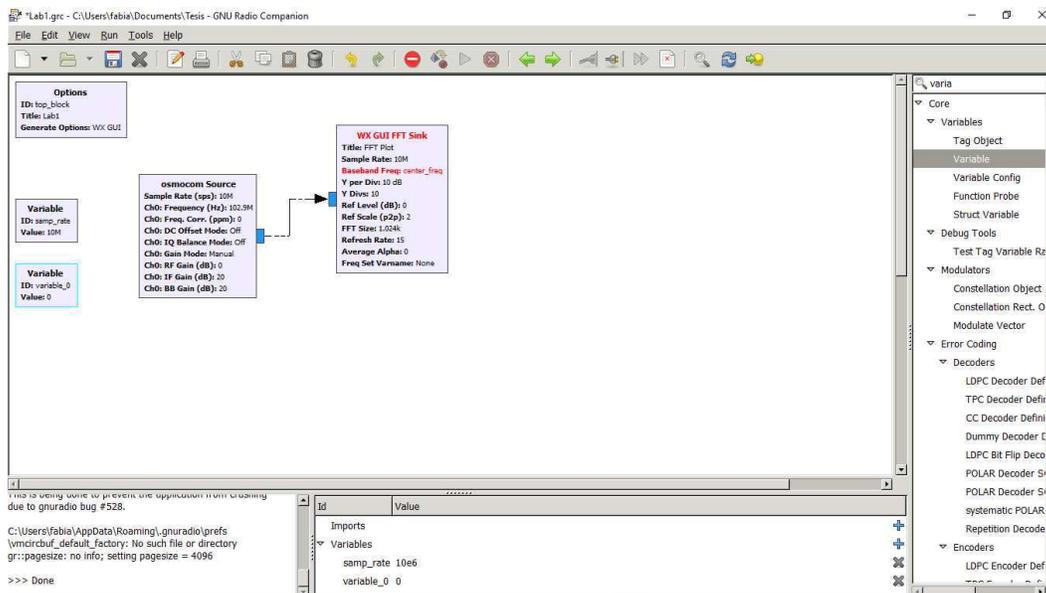


Figura 49. Paso 8.

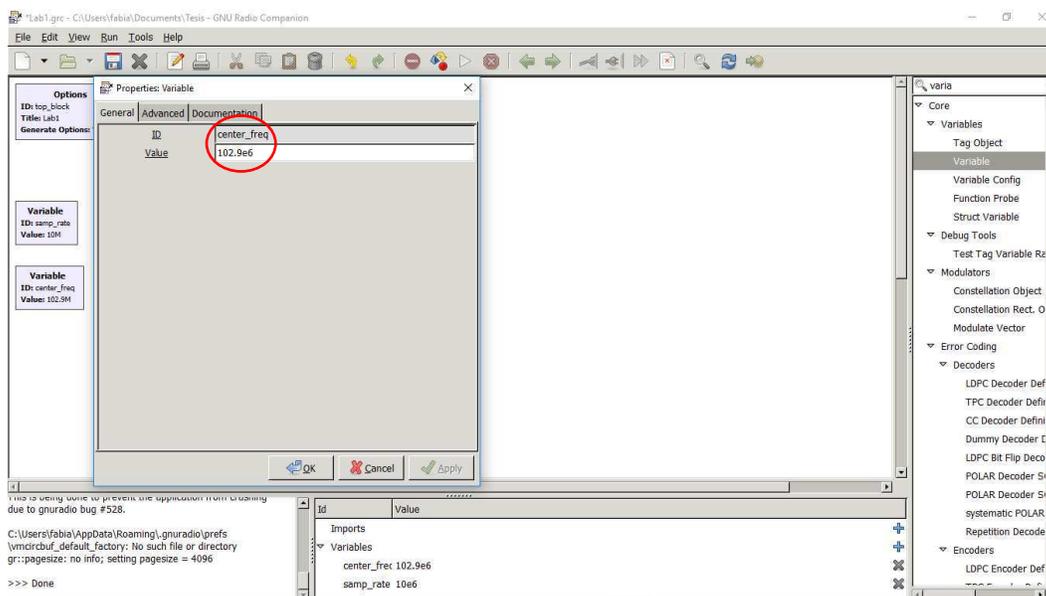


Figura 50. Paso 8a.

**Paso 8.1.** Ahora si observamos el bloque “osmocom Source” este aun dice 102.9M, vamos a propiedades y ponemos center\_freq como se observa en la Figura 51.

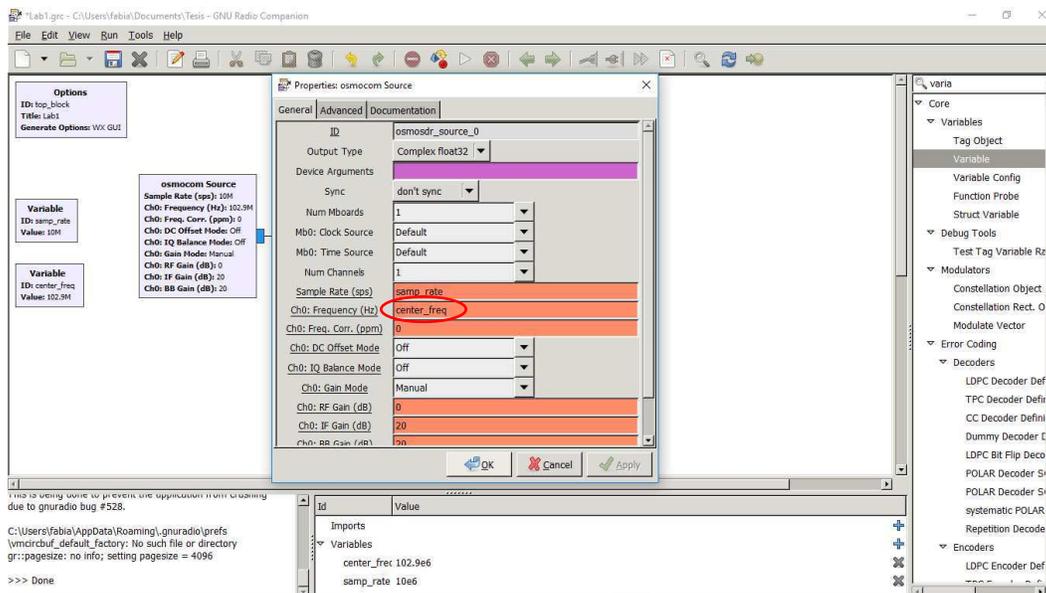


Figura 51. Paso 8.1.

**Paso 8.2.** Al abrir de nuevo la ventana FFT plot, está ahora muestra las frecuencias reales como se observa en la Figura 52 y al mover el cursor se evidencia la frecuencia exacta donde unas son más fuertes que otras.

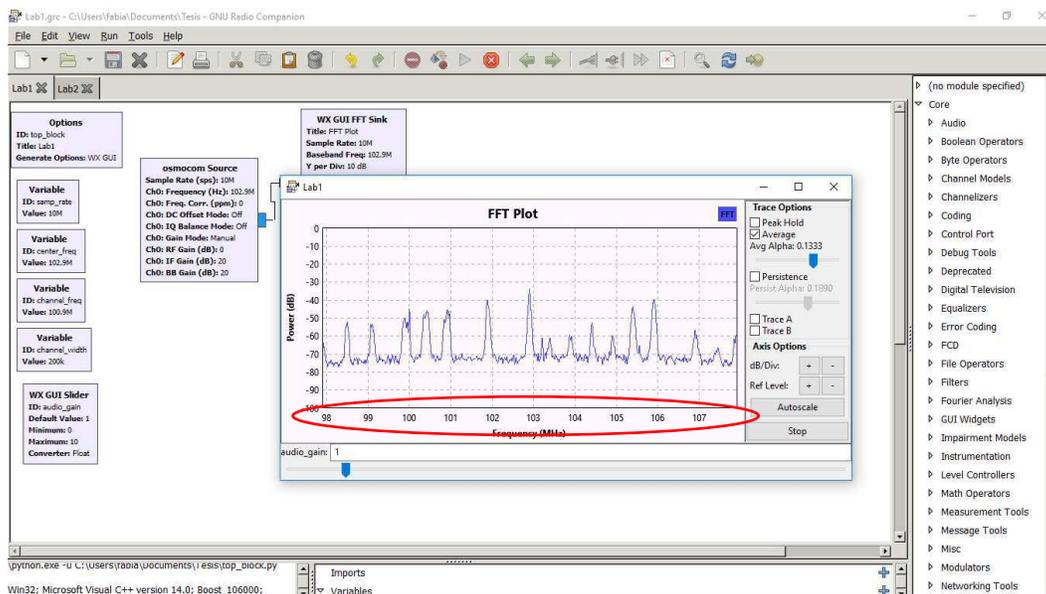


Figura 52. Paso 8.2.

Así que ahora se verá lo realmente necesario para escuchar una de estas estaciones.

**Paso 9.** Lo primero es cambiar una de estas frecuencias a centro, así que el centro de la trama o el área donde anteriormente se llamaba 0 megahertz se va a mover la estación 100.9Mhz que corresponde a “Caracol radio Bogotá”.

Se podría pensar que se cambiara la frecuencia central, pero se hará esto en el software en lugar de ajustar el hardware a una frecuencia diferente, se utilizara un operador de matemáticas llamado “multiply”. Este bloque simplemente hace una multiplicación punto por punto de una señal digital a otra señal digital, se conecta de un bloque a otro como se muestra en la Figura 53.

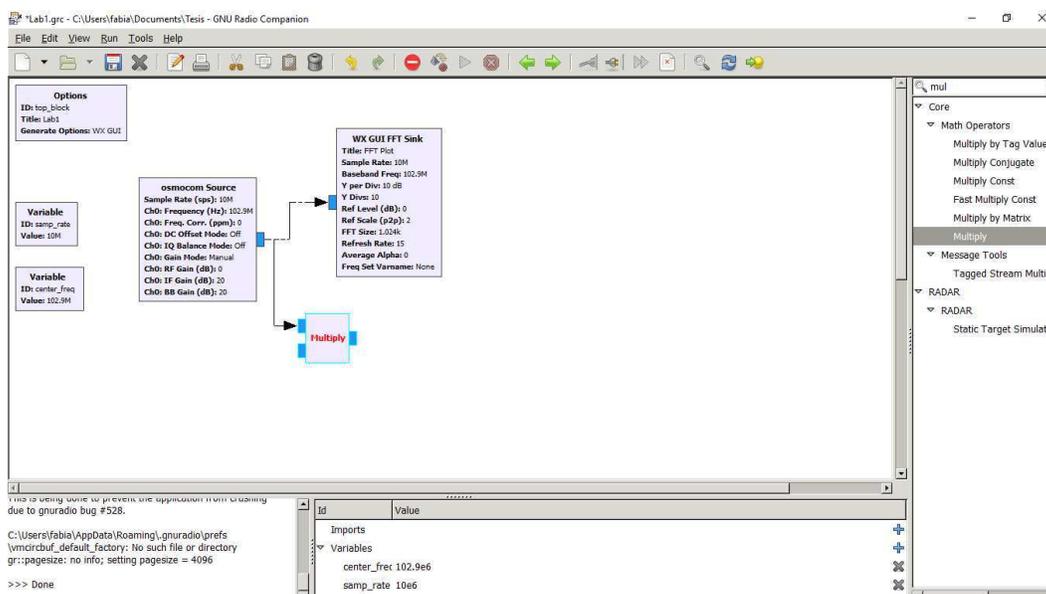


Figura 53. Paso 9.

**Paso 9.1.** Una señal digital es sólo una secuencia de números que va de un bloque a otro, ahora vamos a producir una señal de una sola fuente bajo generadores de forma de onda, se escoge “Signal Source” y la unimos con el “multiply” como se muestra en la Figura 54.

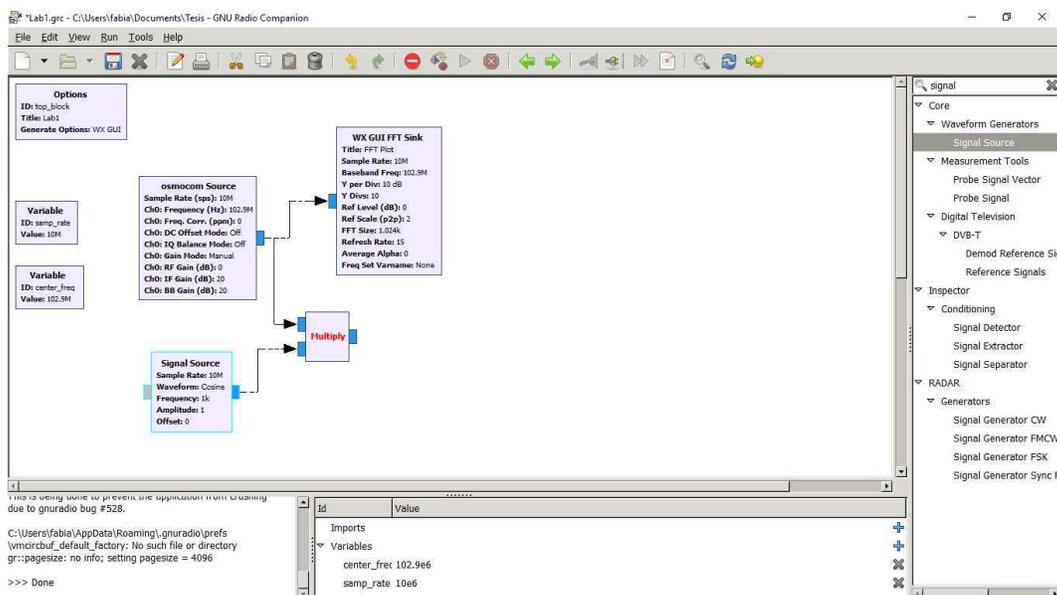


Figura 54. Paso 9.1.

Esta es una señal que se cambia de tamaño y que se está produciendo en el multiplicador de CPU, la señal punto por punto contra la señal que viene de la Hack RF están a una frecuencia que se quiere establecer, esto también es en realidad la cantidad de cambio que se quiere o la diferencia en frecuencia entre la frecuencia central y la frecuencia del canal en la que se tiene interés.

**Paso 10.** Se crea una variable llamada `channel_freq`, para más facilidad simplemente se copia y pega el bloque “variable” como se observa en las Figuras 55 y 56, luego en propiedades se cambia el Id con `channel_freq` y en value por `100.9e6` y ok tal cual como las Figuras 57 y 58.

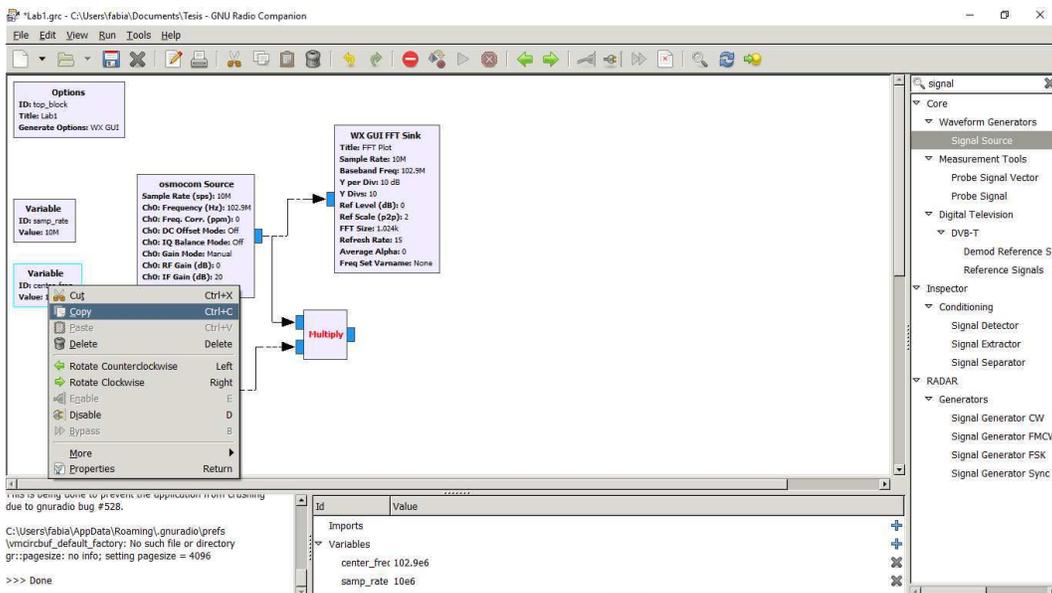


Figura 55. Paso 10.

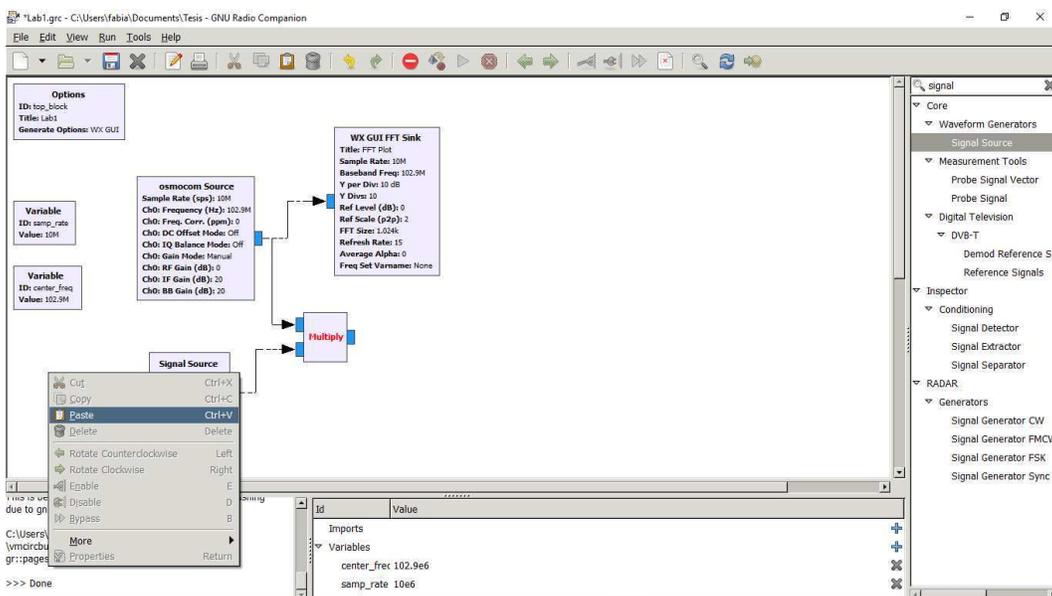


Figura 56. Paso 10a.

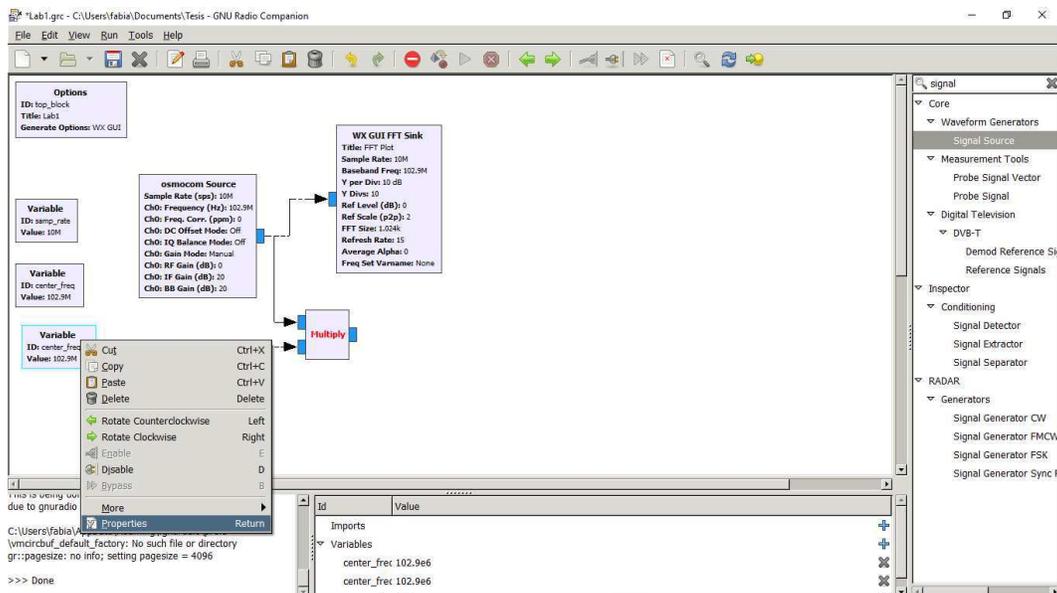


Figura 57. Paso 10b.

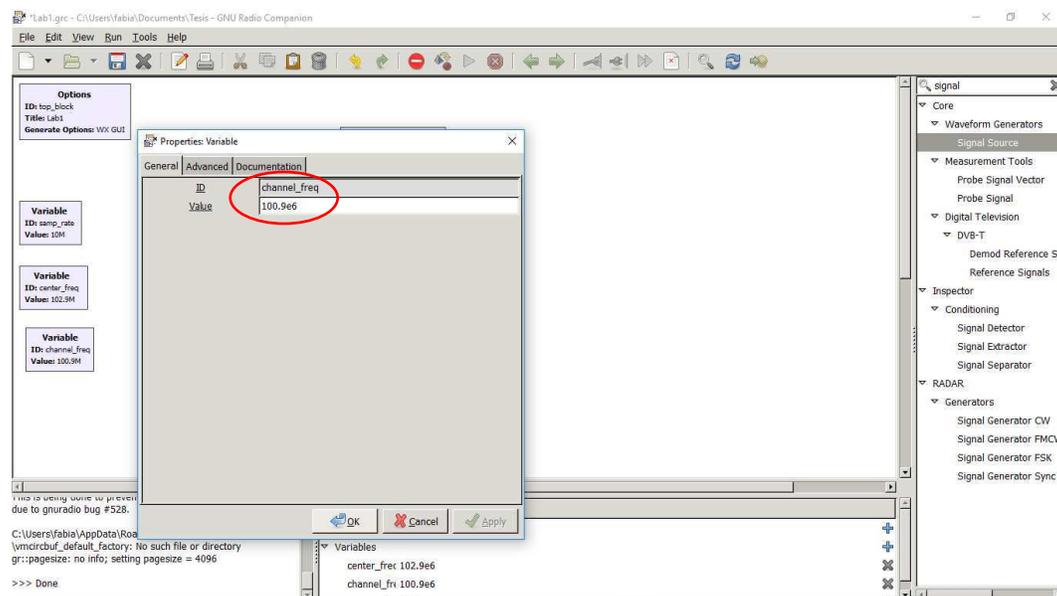


Figura 58. Paso 10c.

**Paso 11.** Ahora en Signal Source, vamos a hacer que la frecuencia sea un `center_freq` menos (-) `channel_freq` en sus propiedades, como se ve en la Figura 59 se puede escribir una expresión aquí usando múltiples variables por lo que en realidad se puede escribir cualquier expresión de

Python. Esta resta de frecuencias da como resultado 2M como se puede observar en el bloque luego de establecer dicha operación.

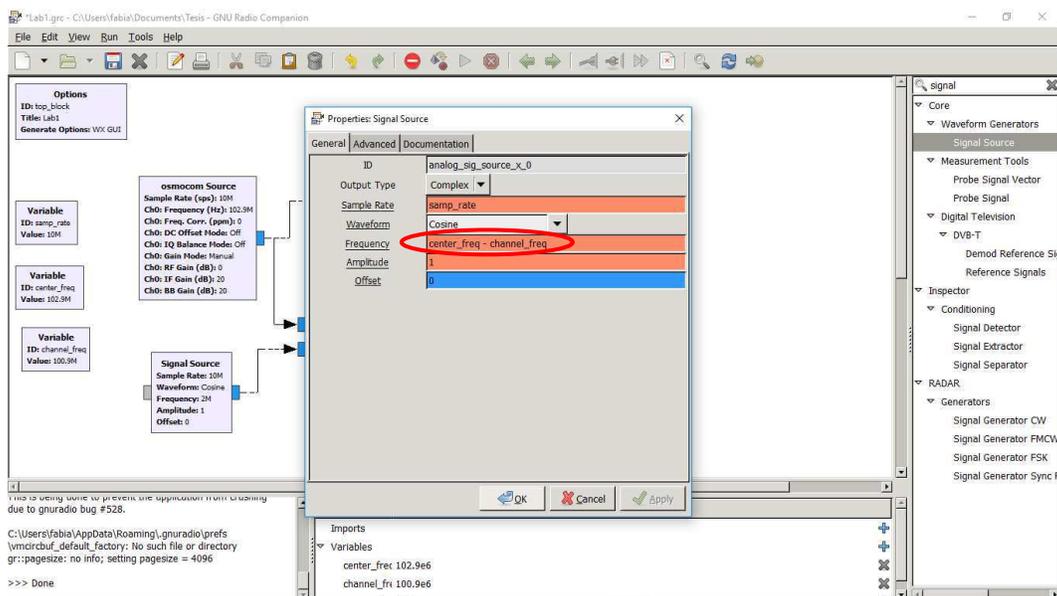


Figura 59. Paso 11.

Para recordar, el GRC está produciendo el código de Python para nosotros así que podemos influenciar la manera que lo hace haciendo cosas como el uso de variables en las expresiones de Python en algunos de los parámetros de estos bloques, en el siguiente paso lo que se hará es verificar que el multiply esté haciendo lo que se quiere.

**Paso 12.** Se tomara el FFT Sink para copiarlo y pegarlo y poder realizar la conexión como se muestra en las figuras 60, 61 y 62.

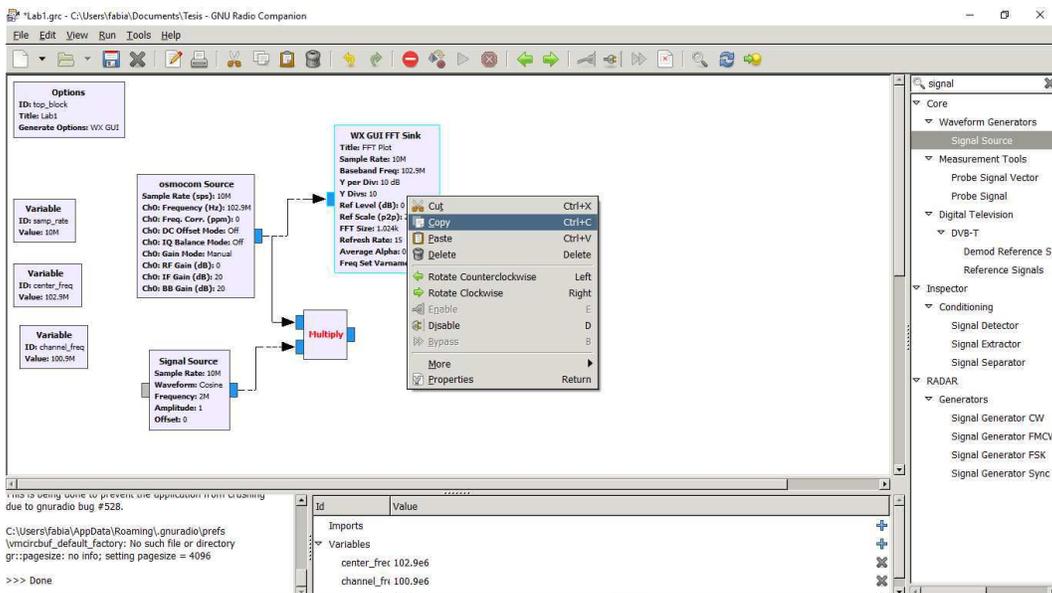


Figura 60. Paso 12.

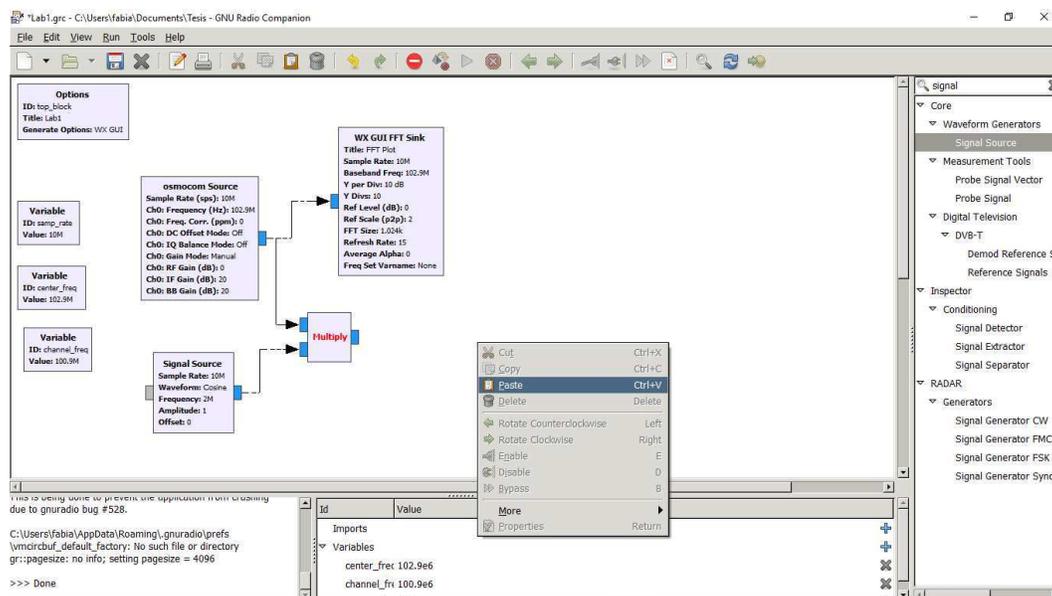


Figura 61. Paso 12a.

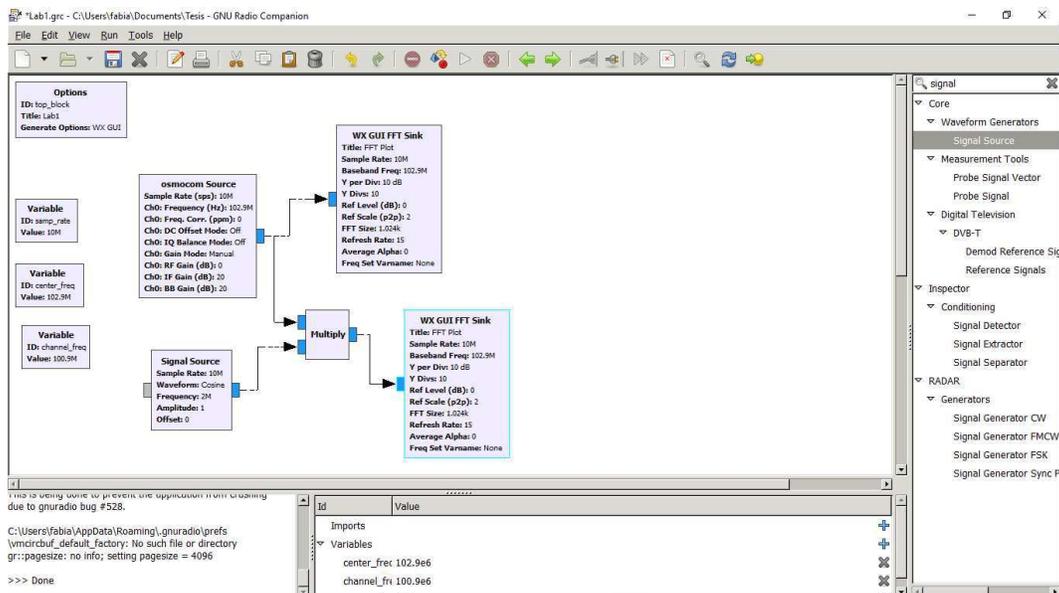


Figura 62. Paso 12b.

**Paso 12.1.** Ahora la frecuencia central de este debe establecerse, se cambiarán los parámetros de la copia FFT sink en sus propiedades tal y como se muestran en la Figura 63.

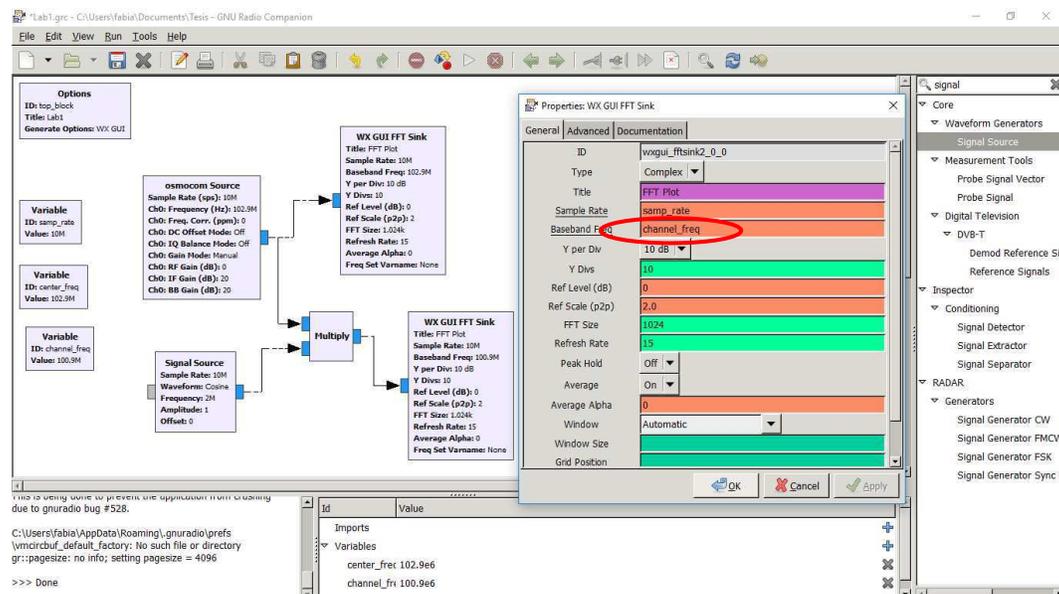


Figura 63. Paso 12.1.

**Paso 12.2.** Ahora se ejecuta y comprobamos si se ve la forma que se quería tal cual como la figura 64.

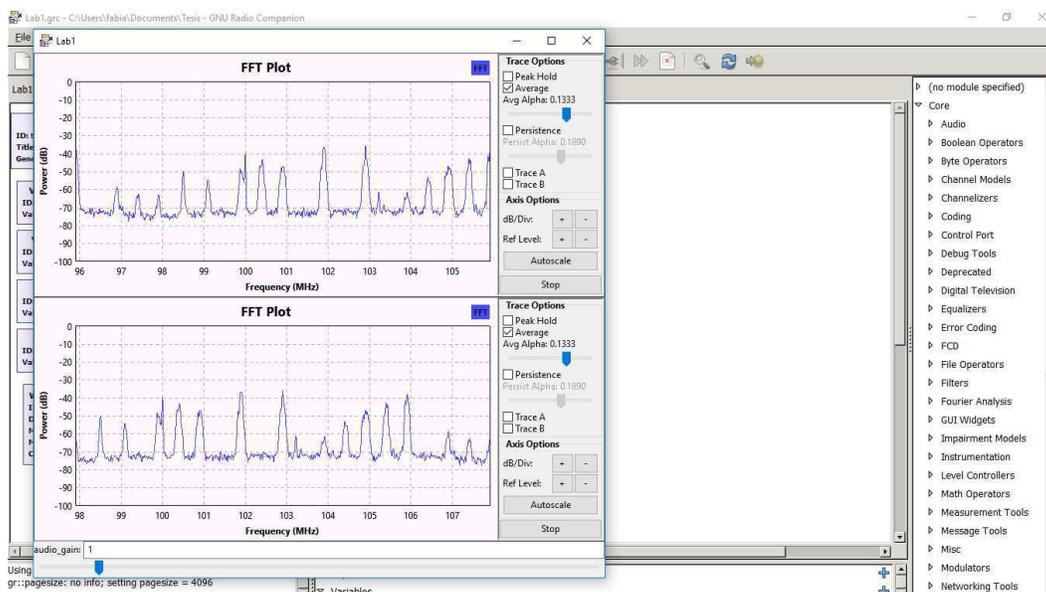


Figura 64. Paso 12.2.

Ahora hay dos FFT plot, el que está en la parte inferior es el que se observó antes y se puede ver que los 100.9Mhz están a la izquierda, pero el que está en la parte superior todo en él se ha desplazado 2Mhz a la derecha y ahora los 100.9Mhz están centrados, es decir que con lo que se acaba de hacer en los pasos anteriores se buscaba establecer esta frecuencia de banda base a 0, es decir que con esto se debían ver los 100.9Mhz en este punto y es justo donde se observa, ahora vamos a hacer algunas cosas para estar listos para demodular la señal de esta estación de radio y convertirla en audio, lo primero que se hará con el siguiente paso es filtrar.

**Paso 13.** Ahora vamos a los filtros y seleccionamos un filtro de paso bajo como se observa en la Figura 65, luego este se conecta al bloque multiply y en propiedades estableceremos una frecuencia de corte de  $75e3$  (75 kilohertz) y una transición con 25 kilohertz como se muestra en la Figura 66, ahora estará la reducción que se quiere la cual está relacionada con la cantidad de ancho de banda de la señal que nos interesa y, obviamente, se tiene mucho más ancho de banda

de lo que realmente se necesita ya que estamos recogiendo múltiples estaciones de radio en muchos megahercios de ancho de banda y simplemente se va a elegir 1.

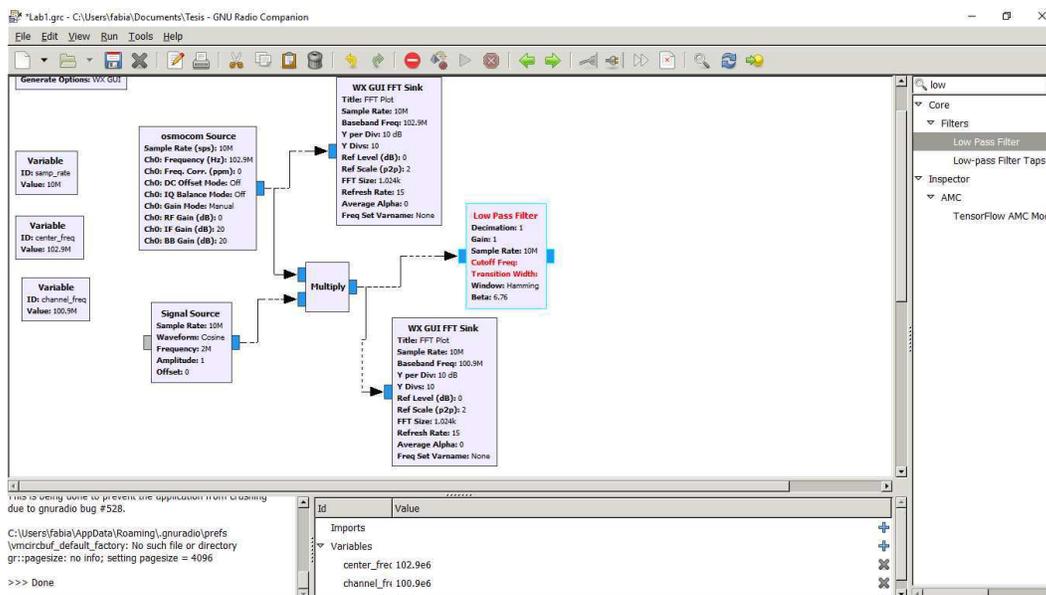


Figura 65. Paso 13.

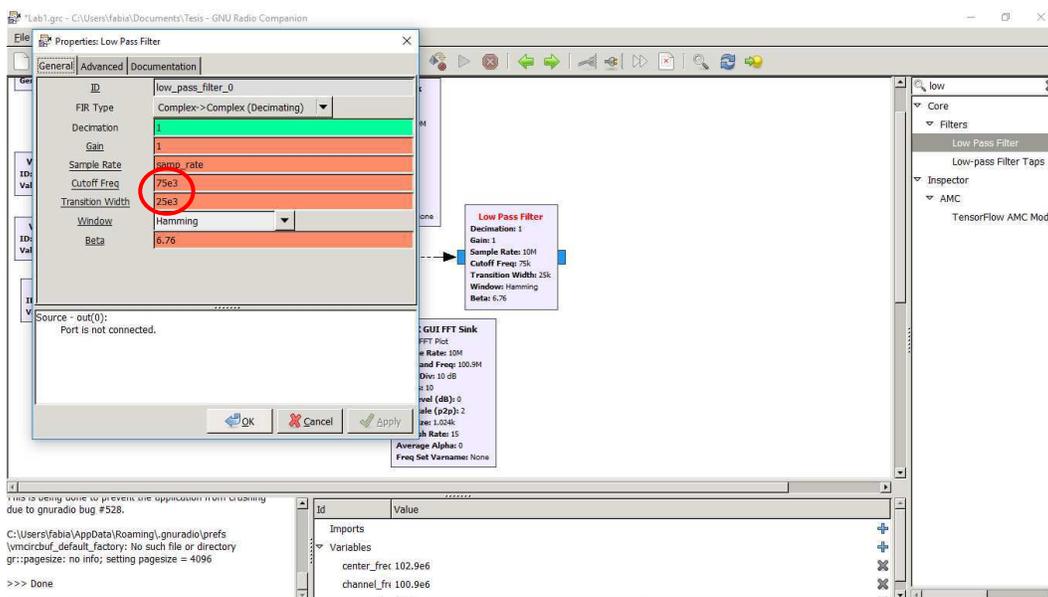


Figura 66. Paso 13a.

**Paso 14.** Vamos a crear una variable copiando y pegando de nuevo el bloque “variable” y se llamar `channel_width` y como el ancho de banda de las estaciones de radio FM es de 200khz, se fijara ese canal con 200e3 como se puede observar en la Figura 67.

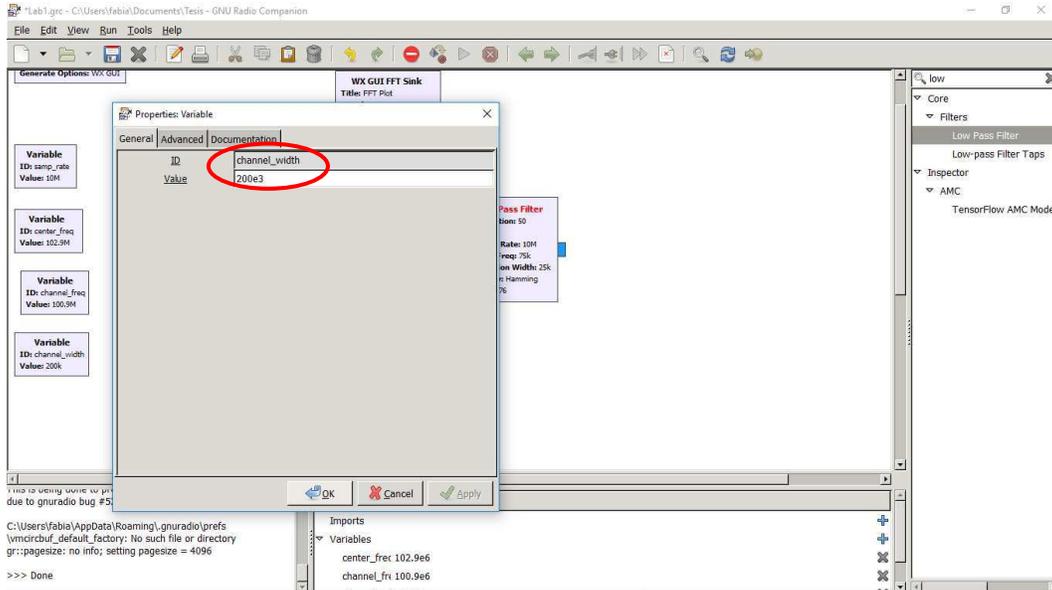


Figura 67. Paso 14.

**Paso 15.** Ahora vamos a establecer el destino del filtro sobre la base de ese canal en sus propiedades, lo cual es el proceso de reducción de la tasa de muestreo, se comenzara con una alta tasa de muestreo entonces se dividirá esa tasa de muestreo en la cantidad que la diezmará “Decimation”, se llamara esta,  $\text{samp\_rate}/\text{channel\_width}$  que será 10 millones / 200.000 como se muestra en la Figura 68. Al hacer clic en ok se pone en rojo el título del bloque del filtro, esto se debe a que este bloque en particular sólo sabe que cómo Decimation solo puede tener un número entero en Python.