



**INSTITUTO FEDERAL DO AMAZONAS
CIÊNCIA E TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
CURSO DE TECNOLOGIA EM SISTEMAS DE
TELECOMUNICAÇÕES**



**RÁDIO DEFINIDO POR SOFTWARE: MANUAL PRÁTICO DA PLATAFORMA
USRP NO *LABVIEW* COM O EQUIPAMENTO ESTUDANTIL NI USRP 2920**

PAULO TELES ANDRADE JÚNIOR

**MANAUS - AM
2019**

PAULO TELES ANDRADE JÚNIOR

**RÁDIO DEFINIDO POR SOFTWARE: MANUAL PRÁTICO DA PLATAFORMA
USRP NO *LABVIEW* COM O EQUIPAMENTO ESTUDANTIL NI USRP 2920**

Monografia apresentada à banca examinadora do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas IFAM, Campus Manaus Distrito industrial CMDI, como requisito para obtenção de aprovação no curso de Tecnologia em Sistemas de Telecomunicações.

Orientador: Prof. MSc. José Geraldo de Pontes e Souza.

MANAUS - AM

2019

Dados Internacionais de Catalogação na Publicação (CIP)

A553r Andrade Júnior, Paulo Teles.

Rádio definido por software: Manual prático da plataforma *USRP* no *LabVIEW* com o equipamento estudantil NI USRP 2920. / Paulo Teles Andrade Júnior. – Manaus, 2019.

106 f. : il. Color.

Rádio definido por software: Manual prático da plataforma *USRP* no *LabVIEW* com o equipamento estudantil NI USRP 2920 (Tecnologia em Sistemas de Telecomunicações) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, 2019.

Orientador: Prof. MSc. José Geraldo de Pontes e Souza

1. SDR. 2. LabVIEW. 3. NI USRP 2920. 4. Rádio FM. I. Souza, José Geraldo de Pontes e (Orient.) II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 621.382

Elaborada por Fc^a. Amélia Frota, registro n.858 (CRB11)

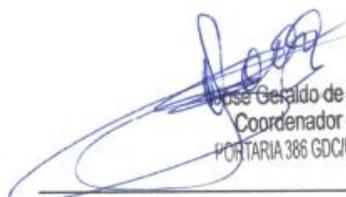
PAULO TELES ANDRADE JÚNIOR

**RÁDIO DEFINIDO POR SOFTWARE: MANUAL PRÁTICO DA PLATAFORMA
USRP NO *LABVIEW* COM O EQUIPAMENTO ESTUDANTIL NI USRP 2920**

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial (IFAM/CMDI), como requisito para obtenção do Título de Tecnólogo em Sistemas de Telecomunicações.

Aprovado em 12 de Dezembro de 2019.

BANCA EXAMINADORA



José Geraldo de Pontes e Souza
Coordenador de Eletrônica
PORTARIA 386 GDC/CMDI/IFAM 8/11/2018

Prof. MSc. José Geraldo de Pontes e Souza

Orientador



Prof. MSc. Nivaldo Rodrigues e Silva

Avaliador



Prof. MSc. Ewerton Andrey Godinho Ribeiro

Avaliador

DEDICATÓRIA

Dedico este projeto a todas as pessoas que contribuíram para minha formação principalmente a Deus, meus pais Paulo Teles e Maria Evanilde e ao professor e orientador José Geraldo.

AGRADECIMENTOS

Agradeço primeiramente à Deus, pois sem ele certamente eu não estaria concluindo esta graduação.

A meus pais Paulo Teles Andrade e Maria Evanilde Pereira de Souza Andrade, pois eles contribuíram para minha formação como homem e cidadão.

Aos meus irmãos Camila Souza Andrade e Santcler Souza Andrade, e familiares.

A minha amada Kethlen de Oliveira Rebouças.

Ao Professor e Orientador José Geraldo.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – Campus Distrito Industrial IFAM – CMDI e a todo corpo docente e funcionários que contribuíram de maneira efetiva na minha formação profissional.

Aos meus amigos e companheiros de sala, Paulo Galvão, Marllon Silva, Walcemir Aguiar, Richardison Moura, Fred Barradas, Roberto Lopes, Rafael Botero, Wellington Pereira, Jhonathas Moutinho, Adriano Viana, Paulo Bacelar, Júlio Cesar, David Allan, Thiago Epifânio e Michell Soares.

A todos que ajudaram nessa caminhada, peço a Deus que abençoe.

RESUMO

Este presente trabalho visa descrever os principais conceitos sobre Rádios Definidos por *Software*, com a finalidade de obter um melhor entendimento desta tecnologia. E, para isso foram feitas pesquisas, análises, o que compõe um SDR (mostrando cada elemento), e o que constitui o *LabVIEW*. Também é descrito uma das aplicabilidades do SDR que é fazer mudanças em toda a estrutura do equipamento, sem a necessidade de modificar o *hardware*, mas apenas a interface gráfica (no *LabVIEW*) ou alterando as linhas do programa, o SDR é um projeto que oferece baixo custo de implementação, *hardware* reduzido, sistema flexível e etc. Suas limitações são os processadores, os conversores Digital-Analógico e Analógico-Digital, os módulos que formam o seu *hardware* (*WIFI*, limite de frequência, e etc.). Apresentando a criação passo a passo do manuseio do equipamento NI USRP-2920, como, também de um circuito de recepção de Rádio FM (demostrando a aplicação combinando o *LabVIEW* e a plataforma USRP).

Palavras-chave: SDR. *LabVIEW*. NI USRP-2920. Rádio FM.

ABSTRACT

This paper aims to describe the main concepts about Software Defined Radios, in order to gain a better understanding of this technology. And, for that were done research, analysis, what makes up an SDR (showing each element), and what constitutes LabVIEW. Also described is one of the applicability of SDR which is to make changes to the entire structure of the equipment without the need to modify the hardware, but only the graphical interface (in LabVIEW) or changing the program lines, SDR is a project that offers low implementation cost, reduced hardware, flexible system and etc. Its limitations are the processors, the Digital-Analog and Digital-Analog converters, the modules that make up your hardware (WIFI, frequency cap, etc.). Introducing the step-by-step creation of handling the NI USRP-2920 equipment as well as an FM Radio receiving circuit (demonstrating the application combining LabVIEW and the USRP platform).

Keywords: SDR, *LabVIEW*, NI USRP-2920, Radio FM.

LISTA DE FIGURAS

Figura 1 - Diagrama em blocos de um receptor de rádio analógico.	16
Figura 2 - Diagrama em blocos de um <i>Software Defined Radio</i> ideal.	16
Figura 3. <i>NI USRP-2920</i>	20
Figura 4. <i>LabVIEW</i> para o RDS	22
Figura 5. Caixa de comando <i>LabVIEW</i>	23
Figura 6. Caixa de funções <i>LabVIEW</i>	24
Figura 7. Interface do <i>GNU Radio</i>	26
Figura 8. Metodologia	27
Figura 9. Especificações <i>USRP 2920</i>	28
Figura 10. Visão geral simplificada da configuração do SDR criada em torno de um <i>NI USRP</i>	29
Figura 11. Diagrama de Blocos do Sistema de Comunicação Digital	30
Figura 12. <i>USRP</i> Diagrama em blocos	32
Figura 13. Componentes	34
Figura 14. Componentes	35
Figura 15. Painel Frontal do <i>NI USRP-2920</i>	35
Figura 16. Radio <i>USRP</i>	38
Figura 17. Tirando proteção de encaixe da antena	38
Figura 18. Conectando antena	39
Figura 19. Antena conectada	39
Figura 20. Conectando cabo <i>Ethernet</i> no Radio	40
Figura 21. Conecte cabo no PC	40
Figura 22. Conectando cabo <i>Ethernet</i> no PC	41
Figura 23. Cabo <i>Ethernet</i> conectado.	41
Figura 24. Conectando fonte no Radio	42
Figura 25. Conectando fonte na tomada	42
Figura 26. Indicações de Instalações e Conexões	43
Figura 27. Arquivo de instalação do programa <i>NI USRP iso</i>	44
Figura 28. Execução do <i>setup</i>	44
Figura 29. Início do processo de instalação do <i>software NI USRP</i>	45
Figura 30. Configurando o <i>host</i>	46
Figura 31. Localização do programa	46
Figura 32. Configurando Endereço do IP do <i>USRP</i>	47
Figura 33. Atualizando <i>Firmware USRP N2xx / NI 29xx</i> e Imagens <i>FPGA</i>	47
Figura 34 - Arquivos de Instalação para o projeto <i>NI USRP</i>	48

Figura 35. Abrindo <i>LabVIEW</i>	50
Figura 36. Selecionando <i>Blank VI</i> e começando a construir o circuito.	50
Figura 37. Selecionando os VI para a montagem do circuito.	51
Figura 38. Selecionando mais VI para a montagem do circuito.....	52
Figura 39. Comando para visualizar nomes dos VIs usados no Projeto.....	53
Figura 40. Criando " <i>Device Name</i> ", onde selecionaremos o IP do Rádio.....	54
Figura 41. Configurando " <i>Device Name</i> ".....	55
Figura 42. Criando " <i>Reset</i> ".	56
Figura 43. Criando " <i>niUSRP Property Node.vi</i> "	56
Figura 44. Criando " <i>Enabled Channel</i> "	57
Figura 45. Criando " <i>Enabled Channel</i> " no <i>Block Diagram</i>	58
Figura 46. Criando " <i>IQ Rate</i> "	59
Figura 47. Criando " <i>Carrier Frequency [Hz]</i> "	60
Figura 48. Configurando " <i>Carrier Frequency</i> " no Painel Frontal.....	61
Figura 49. Criando " <i>Gain (dB)</i> "	62
Figura 50. Criando " <i>Active Antenna</i> "	63
Figura 51. Configurando " <i>Active Antenna</i> " no Bloco de Diagrama	64
Figura 52. Criando " <i>IQ Sampling Rate</i> "	65
Figura 53. Criando " <i>Carrier Frequency</i> "	66
Figura 54. Criando " <i>Gain [dB] (atual)</i> "	67
Figura 55. Criando " <i>Multiply e Seconds per Frame</i> ".....	68
Figura 56. Criando " <i>To Quad Integer</i> "	69
Figura 57. Criando " <i>Initialize Array e Element</i> ".....	70
Figura 58. Criando " <i>Multiply</i> ".....	71
Figura 59. Criando " <i>Reciprocal</i> "	71
Figura 60. Conectando os elementos do circuito.	72
Figura 61. Criando " <i>While Loop</i> " - 1º parte.....	73
Figura 62. Criando " <i>While Loop</i> " - 2º parte.....	74
Figura 63. Criando " <i>While Loop</i> " - 3º parte e Conectando os itens do circuito.	74
Figura 64. Criando " <i>Sample per Frame</i> "	75
Figura 65. Criando " <i>subConvertCluster.vi</i> "	76
Figura 66. Criando " <i>Round Toward - Infinity</i> "	76
Figura 67. Criando " <i>To long Integer</i> "	77
Figura 68. Criando " <i>Reciprocal</i> ", " <i>Initialize Array</i> " e " <i>Build Waveform</i> ".	78
Figura 69. Criando " <i>Sound Output Configure.vi</i> ", " <i>Sound Output Write.vi</i> " e " <i>device ID</i> "	79
Figura 70. Criando " <i>Sound Format</i> " e " <i>Continuous Samples</i> "	80
Figura 71. Criando " <i>Element</i> " e " <i>Unbundle By Name</i> ".....	81

Figura 72. Criando "Sound Output Write.vi", "MT Demodulate FM.vi" e "Resample Waveform (single shot)"	82
Figura 73. Criando "Merge Errors" e "Unbundle By Name"	83
Figura 74. Criando "Build Array" e configurando-o	84
Figura 75. Criando "Build Array" e ativando uma função de adicionar conexão.	85
Figura 76. Criando "reset? (F)"	86
Figura 77. Criando "carrier correction? (F)" e "FM deviation (Hz)"	87
Figura 78. Criando "carrier offset (Hz)" e "remove transient when reset? (F)"	88
Figura 79. Criando "FIR filter specifications" e "open interval? (F)"	89
Figura 80. Fazendo conexões do circuito.....	90
Figura 81. Criando "interpolation mode" e "Compound Arithmetic"	90
Figura 82. Criando botão "STOP"	91
Figura 83. Criando "Build Array", "Sound Output Clear.vi" e "Merge Errors"	92
Figura 84. Criando "error out"	93
Figura 85. Ativando conexões nas extremidades do "While Loop"	93
Figura 86. Criando "Tab Control"	94
Figura 87. Criando "Waveform Graph"	95
Figura 88. Preenchendo informações do "Waveform Graph"	96
Figura 89. "FM Demodulate" criado	97
Figura 90. Demonstração do Projeto Finalizado.	98
Figura 91. Demonstração do Projeto Finalizado	98
Figura 92. Demonstração do Projeto Finalizado	99
Figura 93. Demonstração do Projeto Finalizado	99
Figura 94. Demonstração do Projeto Finalizado	100
Figura 96- Simulação na Frequência 96,9MHz	102
Figura 97 - Simulação na Frequência 95,1MHz	103
Figura 98 - Simulação na Frequência 100,7MHz	103

LISTA DE TABELAS

Tabela 1. Antena Recomendada.....	28
Tabela 2. Conectores do Painel Frontal do Módulo NI USRP-2920.....	36
Tabela 3. LEDs do Módulo NI USRP-2920	37
Tabela 4. Endereços IP Estáticos	45

LISTA DE ABREVIATURAS E SIGLAS

CMDI	– <i>Campus</i> Manaus Distrito Industrial
IFAM	– Instituto Federal de Educação, Ciência e Tecnologia do Amazonas
RDS	– Rádio Definido por <i>Software</i> (<i>Software-Defined Radio</i> ou <i>SDR</i>)
RF	– Rádio Frequência
USRP	– Periférico Universal de Rádio de <i>Software</i> (<i>Universal Software Radio Peripheral</i>)
FM	- Modulação em frequência
OSI	- <i>Open Sistem Interconnection</i> (Interconexão de Sistema Aberto)
CPU	- Unidade Central Processamento
DDC	- Digital <i>Down Converter</i> (Conversor de Frequência Digital)
DUC	- Digital <i>Up Converter</i> (Conversor de Frequência Digital)
FPGA	- Arranjo de Portas Programáveis em Campo
VI	- Instrumentos Virtuais
VHF	- Frequência Muito Alta
UHF	- Frequência Ultra Alta
NI	- <i>National Instrumental</i>
PLL	- Circuito de Bloqueio de Fase
VCO	- Oscilador Controlado por Tensão
LO	- Oscilador Local
ADS-B	- Transmissão Automática de Vigilância Dependente
PC	- <i>Personal Computer</i> (Computador Pessoal)
DCR	- Receptor de Conversão Direta
PPS	- Temporização de Pulso por Segundo
MIMO	- Múltiplas Entradas e Múltiplas Saídas
TX	- Transmissão
RX	- Recepção
ASIC	- Circuitos Integrados de Aplicação Especifica
Hz	- <i>Hertz</i> (Unidade de Medida de Frequência)
AM	- Modulação em Amplitude

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Motivação	17
1.2	Objetivos.....	17
2	REFERENCIAL TEÓRICO	18
2.1	Rádio Definido por <i>Software</i>	18
2.1.1	Plataforma de <i>Hardware</i> - USRP.....	20
2.2	<i>LabVIEW</i>	21
2.2.1	Estrutura do <i>LabVIEW</i>	23
2.2.1.1	Painel Frontal.....	23
2.2.1.2	Diagrama de Blocos.....	23
2.2.1.3	SubVI	24
2.2.1.4	Estruturas de Execução	24
2.3	Frequência Modulada - FM.....	25
2.4	Trabalhos Relacionados	25
3	MÉTODOS EMPREGADOS.....	27
3.1	Especificações USRP 2920.....	28
3.2	Arquitetura USRP	28
4	Práticas laboratoriais	34
4.1	Laboratório 1 – Passo a passo da Instalação	34
4.2	Laboratório 2 – Rádio FM	49
5	ANÁLISE DE RESULTADOS	101
6	CONSIDERAÇÕES FINAIS	105
	REFERÊNCIAS BIBLIOGRÁFICAS.....	106

1 INTRODUÇÃO

O presente Trabalho de Conclusão de Curso tem por finalidade instruir sobre práticas da Plataforma USRP, desde sua montagem até a simulação. Mas, para isso, torna-se necessário os conceitos fundamentais que auxiliam na compreensão do referente estudo, para assim conter o melhor proveito do conhecimento adquirido.

Nas últimas décadas a telecomunicação tem avançado bastante na área da comunicação digital. Com o advento desse avanço e vários estudos envolvidos, foi desenvolvido o processador digital. É um dispositivo que vem integrado a ele, um sistema programável fazendo com que os equipamentos (rádios, *hardware*, e etc.) que o contenham, sejam flexíveis, ou seja, através do *software* que esse dispositivo possui, defino o que quero que o equipamento execute, mas, observando o limite do equipamento. A partir disso, deu-se origem a um novo modelo de rádio chamado *Software Defined Radio* (ou Rádio Definido por *Software* – SDR). Neste modelo foi transferido para o *software* a tarefa de dirigir o comportamento do rádio, deixando para o *hardware* apenas a implementação do *front-end RF*. Assim, através de alterações nas configurações do *software*, podemos modificar a largura de banda, modulação, taxa de codificação, implementar novo elemento ao circuito, e etc., tornando o rádio não mais um *hardware* estático, e sim um elemento dinâmico.

Podemos definir SDR como um *software* implementado em um *hardware* de um rádio capaz de processar o sinal capturado em uma transmissão, ocorrendo a transmissão/processamento dentro de um processador, computador ou *notebook*, havendo somente antena de recepção. Contudo, esta tecnologia mencionada possui limitações que serão abordadas no trabalho mais à frente.

O SDR também pode ser definido por algum algoritmo/*software* que possui dentro do computador ou *notebook*, digamos que “transformando” o computador ou o *notebook* em um FPGA, ou seja, a CPU vindo a ser um microcontrolador para o rádio. Observaremos as figuras abaixo e identificaremos as diferenças entre um rádio analógico e o SDR.

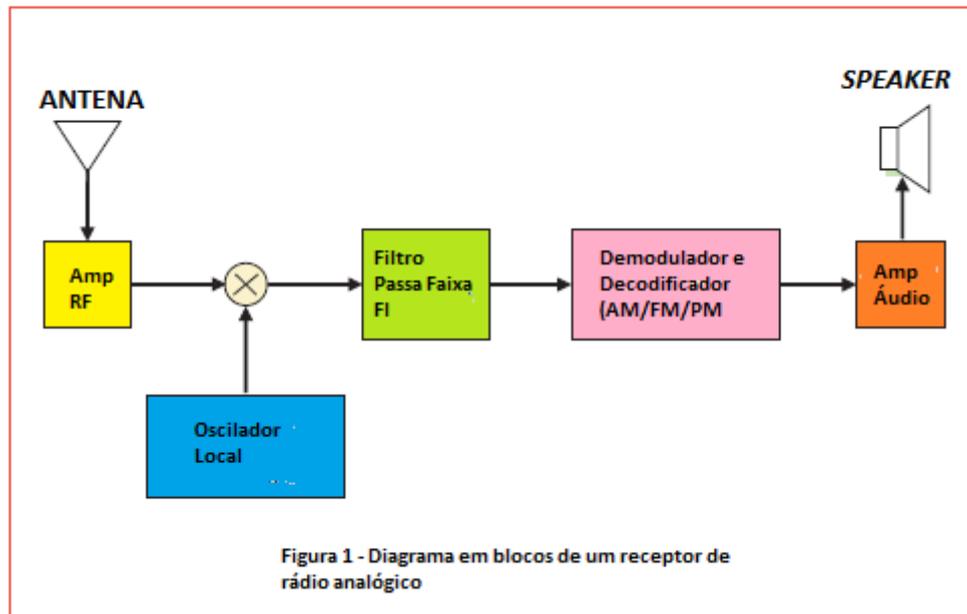


Figura 1 - Diagrama em blocos de um receptor de rádio analógico.

Fonte: <https://www.embarcados.com.br/software-defined-radio-p1/>

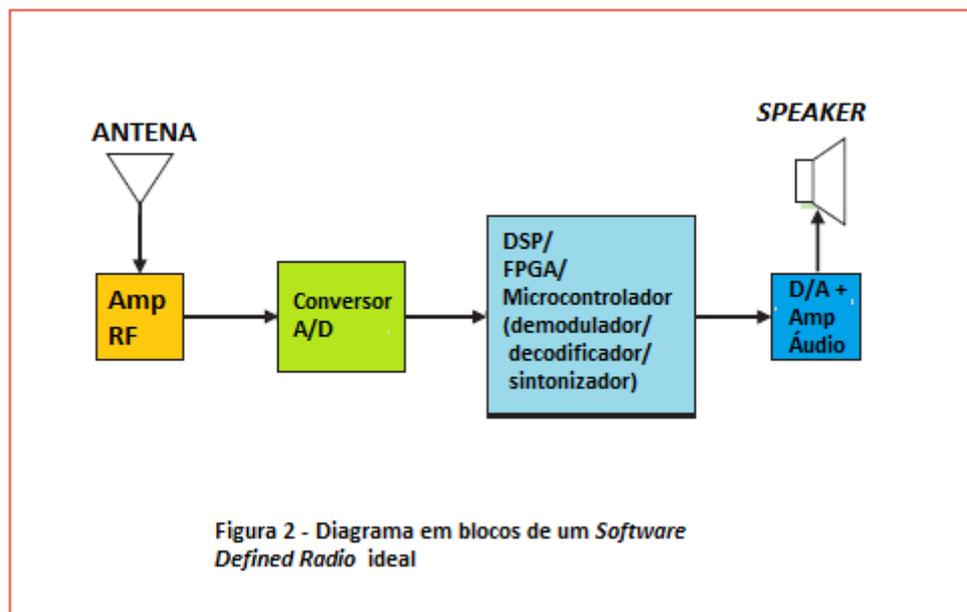


Figura 2 - Diagrama em blocos de um Software Defined Radio ideal.

Fonte: <https://www.embarcados.com.br/software-defined-radio-p1/>

Então observado as figuras percebemos que no SDR todo o processamento foi transferido para uma unidade de processamento externo, deixando para o rádio basicamente as funções de antena e amplificação. A figura 2, seria um SDR ideal, mas não é possível devido as muitas evoluções tecnológicas, que relacionado a este

trabalho, citamos as frequências, onde possuem muitas constâncias tanto na recepção como transmissão do sinal, e com isso muitas pesquisas são feitas para normalizá-las, porém ainda sem tanto êxito.

O SDR apresenta uma grande mudança para os projetos de rádios, como já mencionado o processamento de sinal é programável, podendo alterar as características lógicas, os parâmetros, em tempo real através do *software*. A parte do *hardware* agora ficará mais focada em módulos de transmissão e recepção, deixando as configurações para o *software*.

1.1 Motivação

Conhecer o funcionamento da plataforma USRP, tornou-se algo importante após a aquisição dos *kits* educacionais no Instituto Federal do Amazonas, Campus Manaus Distrito Industrial. E, para futuras aulas, a orientação quanto ao uso da plataforma USRP requer mais material de fácil compreensão, já que manuais ou mesmo simples instruções de utilização se encontram na língua inglesa.

1.2 Objetivos

Objetivo Geral

Formalizar um material de auxílio com passo a passo de práticas laboratoriais, utilizando o equipamento USRP com o *software LabVIEW*, assim proporcionando melhor entendimento sobre o subtema de telecomunicações e Rádio Definido por *Software*, conceito que será lecionado por professores do IFAM CMDI.

Objetivos Específicos

- Estudar o transceptor USRP no *LabVIEW* como objeto de aprendizagem.
- Desenvolver práticas laboratoriais com as instruções adequadas para que os professores possam aplicar conceitos explanados sobre SDR, que englobam telecomunicações.
- Colaborar para que durante a utilização do equipamento USRP, as práticas sejam mais produtivas em sua execução e compreensão do SDR.

Além desta Introdução e dos elementos pré-introdutórios, este trabalho está estruturado da seguinte forma: a Seção 2 apresenta conceitos sobre SDR e USRP; descreve também os mais relevantes trabalhos relacionados, retratando a importância do assunto de estudo; além de falar sobre o *software* utilizado *LabVIEW*. Na Seção 3, temos a metodologia adotada para a realização do estudo, abordando o procedimento da pesquisa com explicações sobre o desenvolvimento das simulações práticas realizadas no laboratório, como configurações, funcionamento, aplicabilidades, etc. Na Seção 4, consta avaliação, análise e discussão dos resultados alcançados. Em seguida, tem-se a Seção 5, com as considerações finais e posteriormente a Seção 6, com as referências bibliográficas.

2 REFERENCIAL TEÓRICO

Este capítulo tem por objetivo contextualizar a proposta deste trabalho, e, com a finalidade de alcançar seus objetivos, apresentamos a fundamentação teórica dos conteúdos que sustentam seu desenvolvimento, além da exposição de algumas abordagens principais e atuais existentes na literatura. Basicamente, foram seguidos os passos a saber: estudou-se os conceitos de SDR e de USRP, assim, como suas implementações através de buscas de tutoriais na internet, Modulação em Frequência – FM, além disso foram pesquisados os trabalhos relacionados com aplicações do USRP. Estes assuntos citados serão apresentados a seguir.

2.1 Rádio Definido por *Software*

O SDR é uma tecnologia que permite a construção de diversas funções de um rádio em *software*, executadas por um dispositivo de processamento programável. O SDR herda as vantagens inerentes de um *software*, principalmente a facilidade de mudanças (MUTTONI, 2017).

Com o objetivo de uniformizar o vocabulário desta tecnologia relativamente recente, o SDR fórum (2007) estabeleceu algumas definições importantes da terminologia no contexto do SDR e do Rádio Cognitivo (MUTTONI, 2017). As principais são:

- *Software* - Instruções modificáveis executadas por um dispositivo de processamento programável;

- Rádio - Tecnologia para se transmitir ou receber informações através do uso de radiação eletromagnética;
- Definido por *Software* - refere-se ao uso do processamento de *software* em um sistema de rádio para implementar funções operacionais;
- Camada física - É a camada mais baixa do modelo OSI, onde ocorrem a codificação de canal e o processamento de sinais de RF, FI2 ou de banda base;
- Rádio Definido por *Software* (SDR) - Rádio no qual algumas ou todas as funções da camada física são definidas por *software*;
- Forma de onda (*waveform*) - Conjunto de transformações aplicadas na informação a ser transmitida e o conjunto correspondente de transformações para converter de volta os sinais recebidos em informação (MUTTONI, 2017).

Radio Definido por *Software* (SDR) é uma tecnologia onde um *hardware* e o *software*, podemos dizer que formam uma bela dupla. No caso teremos um rádio (*hardware*) e o *software* (que é onde as funções que “desejamos” e claro que o *hardware* suporta, são definidas). É através do *software* que definiremos os blocos que um rádio comum possui, para processar no SDR (por exemplo, processamento de sinal, modulação, demodulação, filtragem, detecção, e etc.).

O SDR é uma tecnologia que conta com alto grau de flexibilidade, e também tem um baixo custo de implementação, pois a maioria dos problemas pode ser resolvido através de *software*, ou seja, problemas de *hardware* agora serão problemas de *software*.

Em uma maneira mais simples de dizer, um rádio definido por *software* nada mais é que um transceptor de rádio de cujo funcionamento vem a ser implementado por um *software*, podemos alterar o seu funcionamento apenas atualizando o *software*, sem nenhuma mudança na parte física, o *hardware*.

Um SDR digamos que não ideal, possui amplificadores, filtros, e etc., já um SDR ideal seria constituído de antenas inteligentes, conversores A/D/A (Analógico, Digital, Analógico) e os processadores (*softwares*). Mas, para construirmos um SDR ideal existe muitas limitações; como os conversores ficariam sobrecarregados, a configuração do *software* seria muito complexa, e exigiria muito dos processadores.

2.1.1 Plataforma de *Hardware* - USRP

A plataforma de *hardware* é o dispositivo que fica entre a antena e o computador hospedeiro (que normalmente executa a maior parte do processamento do SDR). A plataforma de *hardware* contém componentes analógicos e digitais e normalmente implementa o processamento primário de sinal do SDR, que geralmente compreende a filtragem, amplificação, conversão de frequência, conversão A/D e/ou D/A, e DDC e/ou DUC. O processamento digital dentro desta plataforma normalmente é executado por um FPGA. A plataforma de *hardware* pode possuir diversos formatos e tamanhos, como ser montável em *rack*, em mesa, ou mesmo ser portátil (MUTTONI, 2017).

Plataforma de *hardware*, objetivamente falando é o rádio, que será usado para executar as instruções a qual desejarmos. Neste rádio encontraremos dispositivos eletrônicos como: conexões, capacitores, indutores, placa mãe, CPU, processadores e etc. assim, esses componentes formam uma plataforma de *hardware*.

Neste trabalho teremos a plataforma USRP (*Universal Software Radio Peripheral*) uma plataforma bastante avançada, com componentes que podem executar quase todas as instruções que desejarmos, dependendo da sua configuração, essa plataforma será controlada pelo *software* que será detalhado mais à frente, chamado *LabVIEW*.

Dentre os diversos formatos, tem-se a plataforma chamada de USRP (*Universal Software Radio Peripheral*), que possibilita a criação de rádios funcionais a partir de modelos de alto nível utilizando computadores pessoais. Assim, mesmo com a transmissão em tempo real do sinal eletromagnético, todo o processamento digital do sinal é feito via *software* (JUNIOR, 2015).



Figura 3. NI USRP-2920

Fonte: <http://www.ni.com/example/14311/en/>

O desenvolvimento das funções de um SDR é feito normalmente em uma plataforma de *software*, uma ferramenta (ou ambiente de programação) executada em um computador hospedeiro que organiza as diversas funções do SDR, permite a sua depuração e execução (MUTTONI, 2017).

2.2 **LabVIEW**

O *software LabVIEW* (Laboratório de Instrumentação Virtual de Engenharia em Bancada) é uma linguagem de programação gráfica, desenvolvida pela *National Instruments*, em 1986. Possui uma interface com efeitos interativos, utilizando ícones e conexões gráficas, muito diferente de outras linguagens de programação que utilizam textos para estabelecer fluxos, operações e relações lógicas do programa que será criado. O *LabVIEW* pode ser trabalhado em diversas plataformas como por exemplo: *Windows*, *Linux* e etc.

O *LabVIEW* é uma plataforma de desenvolvimento proprietária para a programação gráfica, desenvolvida pela *National Instruments*. Para o SDR, tem função e uso similares ao *Matlab/Simulink*, onde as funções de SDR são organizadas em blocos e podem ser programadas graficamente e/ou por linhas de código. A Figura 2 apresenta uma tela de exemplo do *LabVIEW* sendo utilizado para a programação de um SDR (MUTTONI, 2017).

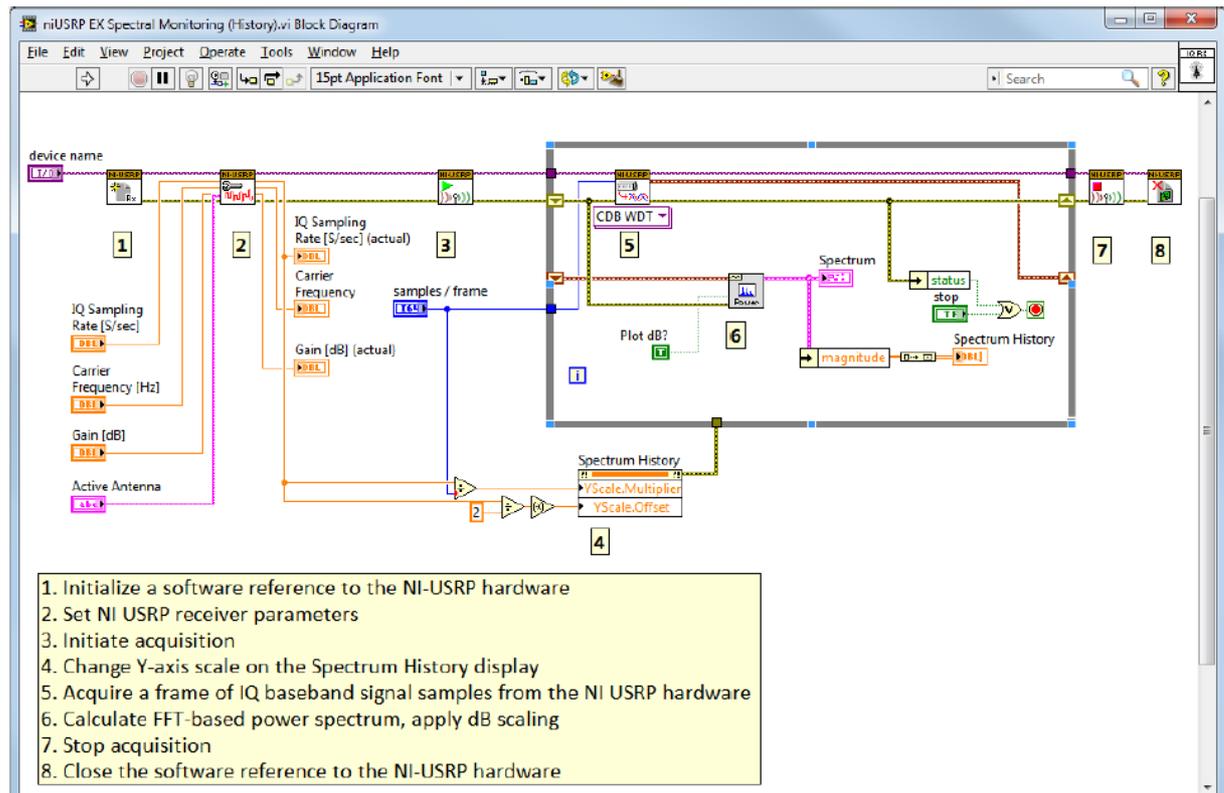


Figura 4. LabVIEW para o SDR

Fonte: Rutgers University School of Engineering (s. d.)

A linguagem de programação G (*graphics*) é a utilizada pelo *LabVIEW* e, é manipulada baseando-se em fluxos de dados, de uma maneira bem intuitiva, estabelecendo entrada de dados, interligados por fios, e assim fluindo o programa, havendo processamento/transformações que passam por funções, representadas por caixas, e por fim saem, apresentando-se aos usuários.

Os programas criados no *LabVIEW* são classificados com VIs (Instrumentos Virtuais). Contendo painel frontal, que possui interface, e o diagrama de blocos, que possui a parte gráfica do programa. O programa criado não é interpretado e sim compilado, podendo ser comparado a uma linguagem de programação de alto nível.

Então no *LabVIEW*, podemos fazer simulações de circuitos elétricos, simulações de funções matemáticas, simulações de análises estatísticas de dados, permite também o processamento de sinais em tempo real, com a integração de *hardware* físico que é o que será detalhado mais à frente neste trabalho *USRP – Rádio FM* ouvindo uma estação de Rádio.

2.2.1 Estrutura do LabVIEW

2.2.1.1 Painel Frontal

Painel Frontal é nele que existe a interação do programa com o usuário, onde temos botões, gráficos, régua e etc., que foram criados pelos programadores. A entrada de dados se dá por cada controle e a saída por cada indicador, definido pelo programador, o processamento dos dados imputados pelo usuário percorre todo o circuito do diagrama de blocos para enfim ser mostrado no painel frontal.

Na figura 5, apresenta alguns dos comandos a serem selecionados pelo programador, possuem comandos de controle e também indicadores que são divididos por categoria. No entanto, cada programa que for ser criado possui uma lógica a ser seguida, existem as caixas de controle e indicadores que são obrigatórios para aquele programa, ou seja, que condiz ao programa.

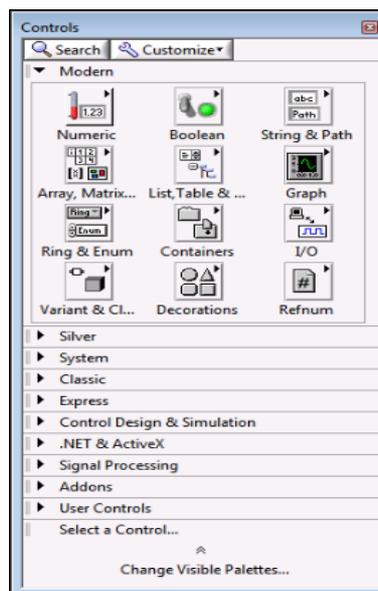


Figura 5. Caixa de comando LabVIEW

Fonte: <http://www.ni.com>

2.2.1.2 Diagrama de Blocos

Ao modo que o programador vai selecionando os controles e indicadores de seu programa, é criado ícones interativos no diagrama de blocos, que por definição são códigos em linguagem G. Os ícones que aparecem no diagrama de blocos, são

representantes de funções, estruturas de execução ou subVI's, que possuem entradas ou saídas de acordo com o programa, tendo ligações entre eles, para executar as operações desejadas.

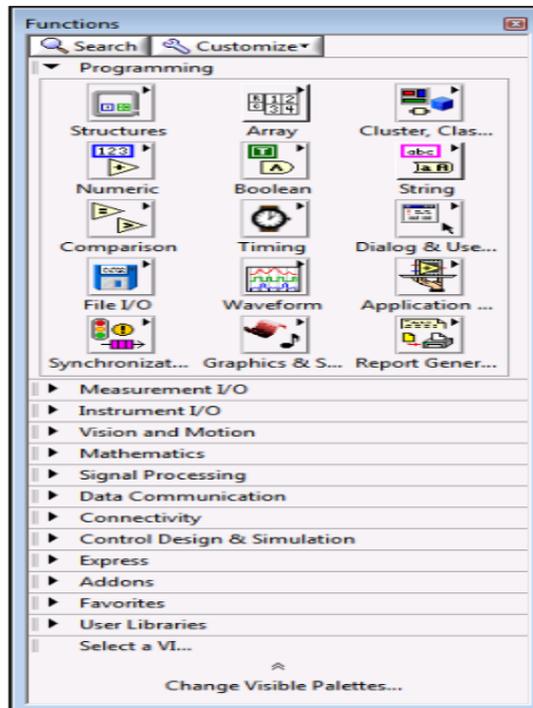


Figura 6. Caixa de funções LabVIEW

Fonte: <http://www.ni.com>

2.2.1.3 SubVI

Um subVI é um VI criado que pode ser implementado em um outro VI, assim denominado subVI. Isso ajuda a diminuir o tamanho do VI que está sendo criado.

2.2.1.4 Estruturas de Execução

As estruturas de execução são os loops, podemos citar o *While*, o *For*, e o *Case*, que são criadas para determinar quando e como uma parte ou todo o programa deverá ser executado, podendo também ser repetido várias vezes.

2.3 Frequência Modulada - FM

A modulação em frequência é conhecida como FM e, tem como definição a variação da frequência da portadora que leva a informação, mantendo a amplitude do sinal. A modulação em frequência é aplicada em radares, telemetria, medições de abalos sísmicos, e até em alguns sistemas de transmissão de vídeo, mas as maiores utilizações da modulação em frequência estão nas transmissões de música, voz e rádios bidirecionais.

O Rádio FM, foi projetado por Edwin Armstrong, no ano de 1933 e, é um serviço de radiodifusão (ex.: serviços de áudio), em frequências que variam entre 88Hz a 108 MHz, usando a modulação em frequência. Mas foi no ano de 1939, que Armstrong colocou a primeira transmissão FM em operação nos Estados Unidos, mais precisamente, em Alpine, Nova Jersey. A primeira emissora de rádio FM no Brasil, foi a “Rádio Imprensa”, fundada por Anna Khoury, ano de 1955. Neste tempo, utilizavam a frequência modulada para fazer a ligação entre o estúdio e o transmissor de rádio.

Uma grande vantagem da FM é o cancelamento de ruídos. A modulação em frequência por possuir uma alta qualidade de transmissão, por isso foi a escolhida para transmitir o áudio da TV aberta, tanto em *VHF* (canais 2 a 13) quanto em *UHF* (canais 14 a 69).

2.4 Trabalhos Relacionados

Os estudos voltados para o uso do rádio *NI USRP-2920*, na grande maioria estão descritos usando outras plataformas de comunicação como por exemplo: *MATLAB* e *GNU Radio*. Porém observou-se que tinham o objetivo de disseminar a caracterização e utilização do rádio *NI USRP-2920*. E, trabalhos relacionados com materiais específicos sobre a programação no *LabVIEW* imputando ações ao Rádio *NI USRP-2920*, não foi possível encontrar. Contudo, os presentes trabalhos relacionados, conservam a ideia central proposta neste trabalho, somando e auxiliando no decorrer deste projeto. A seguir, exemplos mais próximos a este tema serão abordados.

O trabalho de JANSON et al. (2012), é apresentado uma descrição a tecnologia e o que compõe o SDR, exemplificando cada elemento. Demonstra também um exemplo de aplicação utilizando o GNU Radio (é uma plataforma *open source*, ou seja, plataforma livre, gratuita, que pode ser manuseada tanto em linguagem de

programação, como em interface gráfica) e a plataforma USRP. O exemplo prático de seu trabalho foi usando dois SDRs um para transmissão AM e outro para recepção AM, na frequência 432,15 MHz.

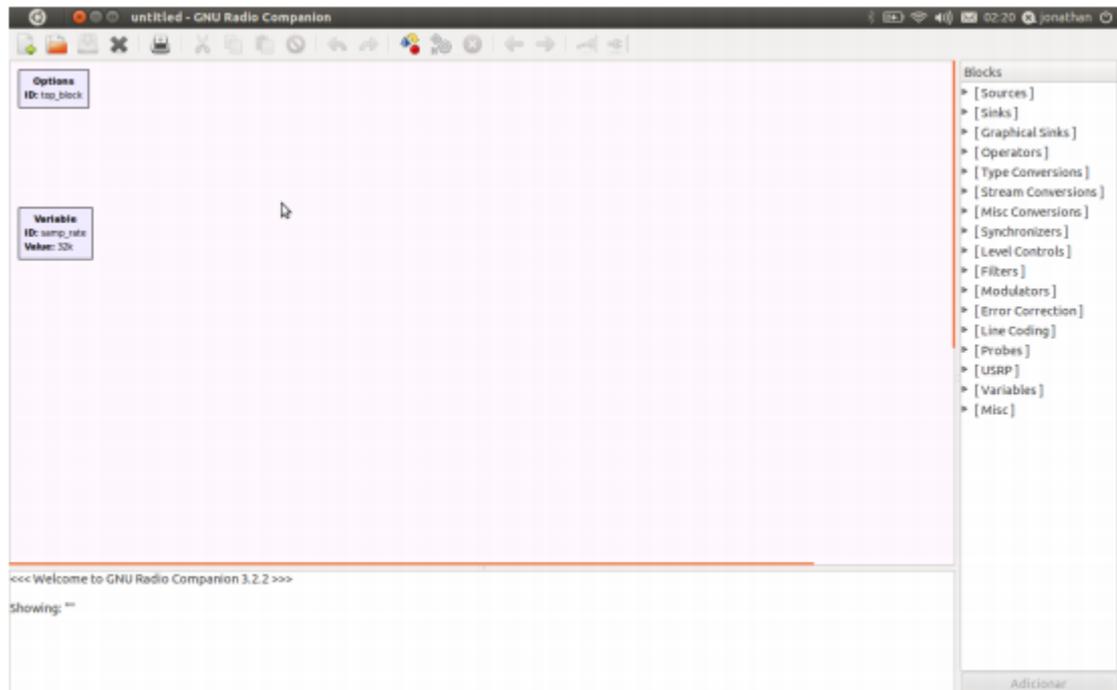


Figura 7. Interface do GNU Radio

Fonte: Jonathan Janson. 2012.

Outro estudo feito MACHADO (2018), é proposto a implementação de uma plataforma de comunicação capaz de transmitir e receber dados, utilizando o protocolo ADS-B (*Automatic Dependent Surveillance Broadcast*), usado em aeronaves como sistema de vigilância. O *hardware* manuseado foi baseado em SDR (*Software Defined Radio*) que substituem *hardware* tradicionais, sendo operado via *software*. Dois SDRs do tipo USRP foram operados, um para transmissão e outro para recepção de sinais ADS-B. Na transmissão ADS-B continha informações de voo de aeronaves, como a velocidade e a altitude. O resultado na recepção vinha como um gráfico em mapa. Então verificou-se que está pode ser uma ferramenta de base para possíveis estudos que envolvam o sistema de comunicação ADS-B, ou roteiro prático inicial do SDR do tipo USRP (*Universal Software Radio Pheripheral*), em aplicações de Rádio Frequência (RF).

3 MÉTODOS EMPREGADOS

Os métodos de abordagens se baseiam nos procedimentos utilizados na investigação da pesquisa descritas por GERHARDT e SILVEIRA (2009). Com base nisso, tem-se as seguintes classificações:

- Quanto aos métodos - indutivo.
- Quanto à natureza - pesquisa aplicada.
- Quanto aos objetivos - descritiva.
- Quanto a abordagem - qualitativa.
- Quanto aos procedimentos - experimental.



Figura 8. Metodologia

Fonte: Autoria Própria

3.1 Especificações USRP 2920

NI USRP

Interface frontal do transceptor RF sintonizável

- Faixa de Frequencia
50 MHz – 2.2 GHz (NI-2920)
2.4 GHz & 5.5 GHz (NI-2921)

Processamento de sinais e síntese

- NI LabVIEW para projetar e explorar algoritmos
- NI Modulation Toolkit e add-ons do LabVIEW para simular ou processar sinais ao vivo

Aplicações

- RÁDIO FM
- TV
- GPS
- GSM
- Zigbee
- Sistema de rádio
- OFDM
- Radar passivo
- Dynamic Spectrum Access

Conectividade ethernet de 1 Gigabit

- Plug-and-play
- Streaming IQ de até 25 MS/s banda base IQ



Figura 9. Especificações USRP 2920

Fonte: Langenaken, 2011

Tabela 1. Antena Recomendada

Fonte: National Instruments, 2014

Dispositivo	Antena	Número de peça da NI
NI USRP- 2920/2922/2930/2932/ 2940R/2942R/2950R/2952R	VERT 400—144 MHz,400 MHz, e 1,200 MHz antena vertical tribanda	781915-01

3.2 Arquitetura USRP

O SDR refere-se à tecnologia em que os módulos de *software* em execução de uma plataforma de *hardware* genérica são usados para implementar funções de rádio (NI, 2013).

Combine o *hardware NI USRP* com o *software LabVIEW* pela flexibilidade e funcionalidade para fornecer uma plataforma para prototipagem rápida envolvendo projeto de camada física, gravação e reprodução de sinal sem fio, inteligência de sinal, validação de algoritmo e muito mais (NI, 2013).

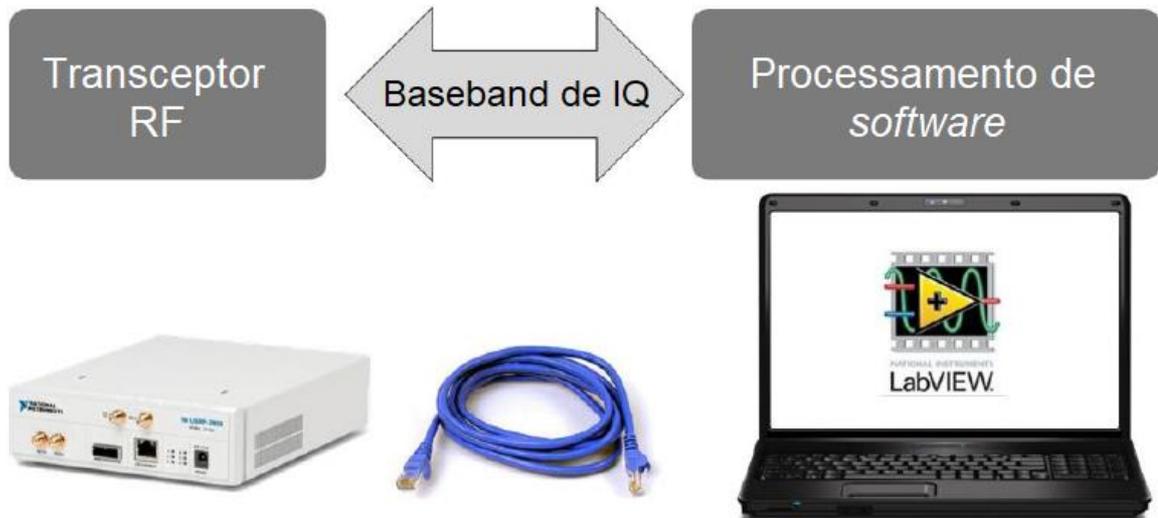


Figura 10. Visão geral simplificada da configuração do SDR criada em torno de um NI USRP

Fonte: National Instruments, 2013

Os sinais recebidos pela USRP-2920 são amplificados, convertidos, filtrados, digitalizados e dizimados antes de serem passados para o computador *host*. Os sinais transmitidos pela USRP-2920 são ampliados, reconstruídos, filtrados, convertidos e amplificados antes de serem transmitidos (NI, 2017).

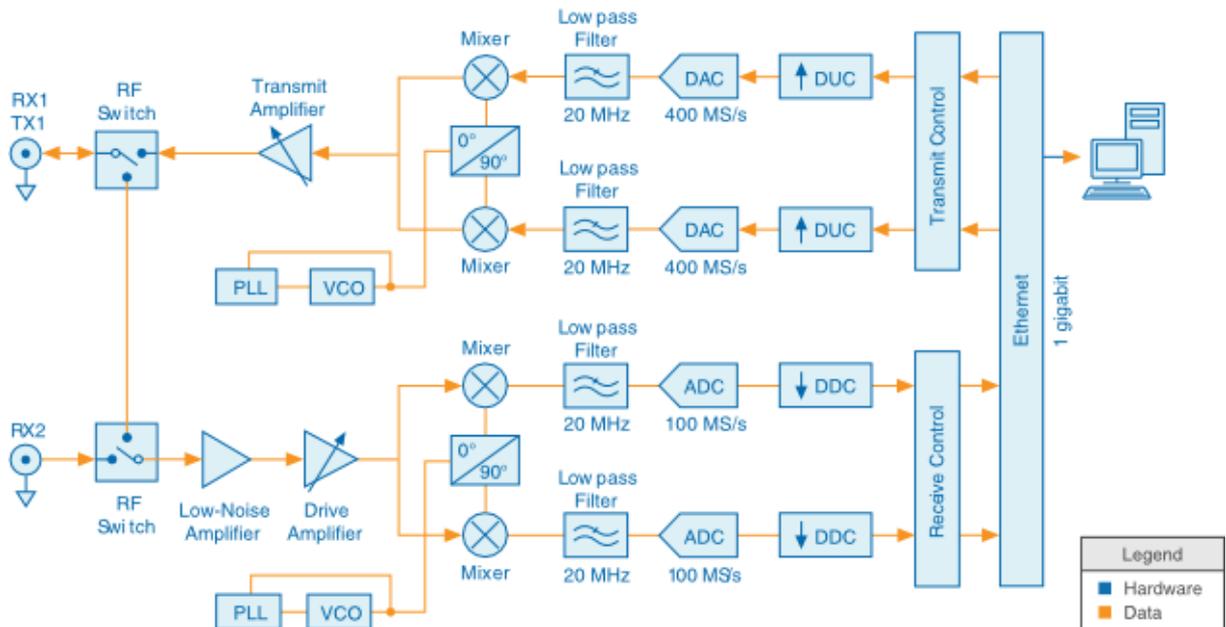


Figura 11. Diagrama de Blocos do Sistema de Comunicação Digital

Fonte: <http://www.ni.com/documentation/en/usrp-software-defined-radio-device/latest/usrp-2920/block-diagram/>

O comutador de RF permite que as operações de transmissão e recepção ocorram na mesma antena compartilhada. Na USRP-2920, uma antena é designada somente para recebimento (NI, 2017).

Caminho de recebimento (NI, 2017):

- O amplificador de baixo ruído e amplificador de unidade amplificam o sinal de entrada.
- O circuito de bloqueio de fase (PLL) controla o oscilador controlado por tensão (VCO) de modo que os relógios do dispositivo e o oscilador local (LO) possam ser bloqueados em frequência para um sinal de referência.
- O *mixer* reduz os sinais para os componentes de fase (I) e de fase de quadratura (Q).
- O filtro *lowpass* reduz os componentes de ruído e alta frequência no sinal.
- O conversor analógico-digital (ADC) digitaliza os dados I e Q.
- O *downconverter* digital (DDC) mistura, filtra e decima o sinal para uma taxa especificada pelo usuário.
- As amostras convertidas para baixo são passadas para o computador *host* através de uma conexão *Ethernet gigabit* padrão.

Caminho de transmissão (NI, 2017):

- O computador *host* sintetiza os sinais de banda base I / Q e transmite os sinais para o dispositivo através de uma conexão *Ethernet gigabit* padrão.
- O *upconverter* digital (DUC) mistura, filtra e interpola o sinal para 400 MS/s.
- O conversor digital-analógico (DAC) converte o sinal para analógico.
- O filtro *lowpass* reduz os componentes de ruído e alta frequência no sinal.
- O *mixer* converte os sinais para uma frequência de RF especificada pelo usuário.
- O PLL controla o VCO para que os relógios do dispositivo e LO possam ser bloqueados em frequência para um sinal de referência.
- O amplificador de transmissão amplifica o sinal e transmite o sinal através da antena.

O NI USRP conecta-se a um PC *host* para atuar como um rádio definido por software. Sinais de entrada ligados ao conector SMA padrão são mixados usando um receptor de conversão direta (DCR) para componentes de banda base I / Q, que são amostrados por um conversor de analógico para digital de 2 canais, 100 MS / s, 14 bits (ADC). Os dados digitalizados de I / Q seguem caminhos paralelos através de um processo de *downconversion* digital (DDC) que mistura, filtra e dizima o sinal de entrada de 100 MS / s para uma taxa especificada pelo usuário. As amostras convertidas para baixo, quando representadas como números de 32 bits (16 bits cada para I e Q), são passadas para o computador *host* em até 20 MS / s através de uma conexão *Ethernet Gigabit* padrão (NI, 2013).

Para transmissão, as amostras de sinal de banda I / Q são sintetizadas pelo computador *host* e alimentadas na USRP-2920 até 20 MS / s através de *Gigabit Ethernet* quando representadas com 32 bits (16 bits cada para os componentes I e Q). O *hardware* USRP interpola o sinal de entrada para 400 MS / s usando um processo de *upconversion* digital (DUC) e, em seguida, converte o sinal para analógico com um conversor de digital para analógico (DAC) de 16 bits e canal duplo. O sinal analógico resultante é então misturado até a frequência da portadora especificada (NI, 2013).

Um modo de 8 bits disponível, no qual o total de 16 bits é usado para representar os valores I e Q de uma amostra convertida ou de uma amostra a ser convertida, pode permitir uma taxa de transferência de até 40 MS / s através da conexão *Gigabit Ethernet*

entre o PC *host* e o USRP (NI, 2013).

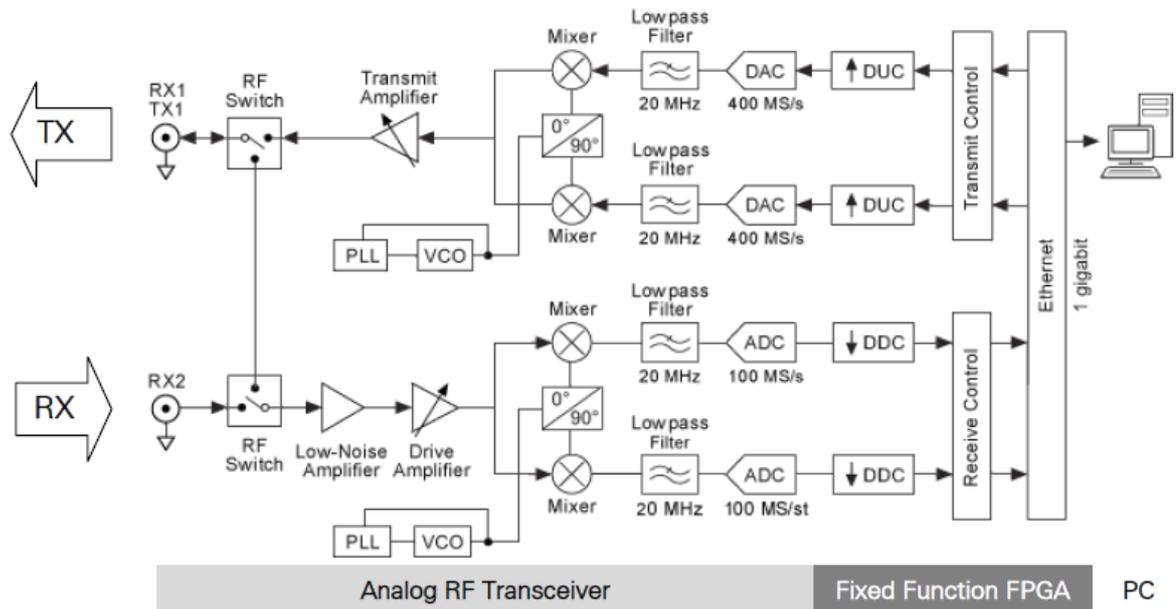


Figura 12. USRP Diagrama em blocos

Fonte: National Instruments, 2013

Neste capítulo será descrito sobre a arquitetura de uma SDR, mais precisamente o módulo *Front End*, parte de processamento digital conhecido como Rádio *back end*, e os processadores de *software* de uma SDR.

A arquitetura SDR possui dois módulos principais que são o *Front End* (se responsabiliza pela transmissão e recepção dos sinais), e o *Rádiorback end* (se responsabiliza pelo processamento dos sinais). O que nos levar a pensar que o Rádio é o elemento majoritário na arquitetura, e isso é uma forma errônea de pensar, porque o tema Rádio Definido por *Software* vem para trazer uma nova ideia de comunicação de Rádio, no qual o *Software* domina a configuração a que o usuário desejar ter (transmissão, recepção e etc.), dentro dos limites de seu Rádio (*hardware*) é claro, ou seja, problemas de *hardware* agora serão problemas de *software*.

Em um modelo de SDR ideal em que temos antenas ligadas a conversores e processamento de sinais, não poderíamos aplicar as tecnologias de hoje, pelo fato dos conversores analógico-digital e digital-analógico possuírem uma tecnologia limitada, tendo que fazer a separação do *front end* onde temos o tratamento do sinal, dos conversores A/D e D/A.

As limitações dos conversores que podemos citar são a taxa de amostragem, o

range dinâmico, o *jitter* (atraso na entrega dos dados em uma rede) de abertura, e alguns ruídos e distorções ligadas aos próprios conversores, por exemplo ruído térmico, ruído de quantização, e distorções não lineares. A partir desses pontos de vista, o *front end* se torna um modulo, onde tem a recepção do sinal preparando-o para o conversor A/D e conseguinte processamento ajustando para o conversor D/A, podendo ser configurado para transmissão, para recepção, ou Recepção/Transmissão. Ao longo deste percurso encontramos amplificação do sinal, ganho do sinal, deslocamento para uma frequência intermediária (para caso de recepção), ou deslocamento para frequência original (para caso de transmissão), filtro passa baixa. Assim, resolvendo as limitações dos conversores e podendo configurar conversores de baixo custo.

4 PRÁTICAS LABORATORIAIS

4.1 Laboratório 1 – Passo a passo da Instalação

NI USRP 2920 hardware e seus componentes

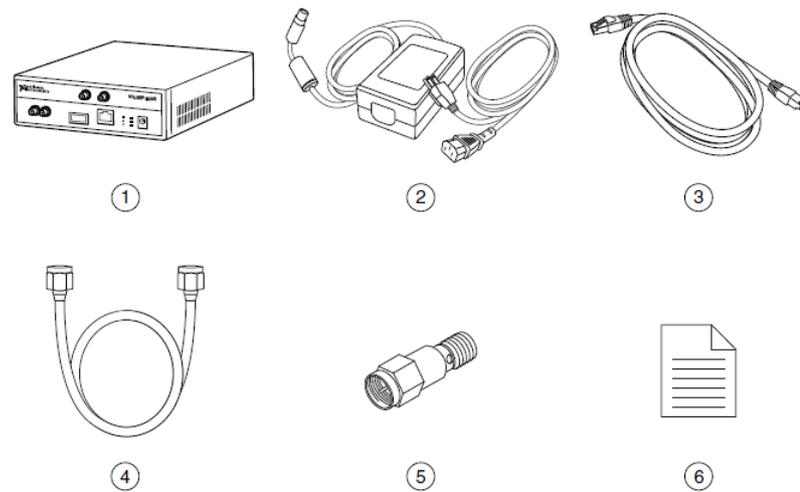


Figura 13. Componentes

Fonte: National Instruments, 2014

- 1 – Rádio
- 2 – Cabo de Alimentação
- 3 – Cabo *Ethernet*
- 4 – Cabo “SMA”
- 5 – Conector de Atenuação
- 6 – Manual



Figura 14. Componentes

Fonte: Autoria Própria.

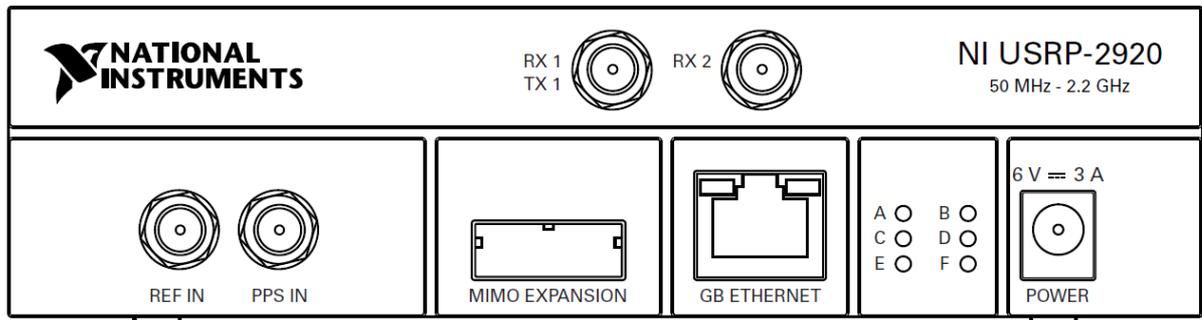


Figura 15. Painel Frontal do NI USRP-2920

Fonte: National Instruments, 2015

Tabela 2. Conectores do Painel Frontal do Módulo NI USRP-2920

Fonte: National Instruments, 2015

Conector	Uso
RX 1 TX 1	Terminal de entrada e saída para o sinal de RF. RX 1 TX 1 é um conector SMA (f) com uma impedância de 50 Ω e é um canal de entrada ou saída de terminação única.
RX 2	Terminal de entrada para o sinal de RF. RX 2 é um conector SMA (f) com uma impedância de 50 Ω e é um canal de entrada <i>single-ended</i> .
REF IN	Terminal de entrada para um sinal de referência externo para o oscilador local (LO) no dispositivo. REF IN é um conector SMA (f) com uma impedância de 50 Ω e é uma entrada de referência de terminação única. REF IN aceita um sinal de 10 MHz com uma potência de entrada mínima de 0 dBm (0,632 Vpk-pk) e uma potência de entrada máxima de 15 dBm (3,56 Vpk-pk) para uma onda quadrada ou onda senoidal.
PPS IN	Terminal de entrada para a referência de temporização de pulso por segundo (PPS). O PPS IN é um conector SMA (f) com uma impedância de 50 Ω e é uma entrada de terminação única. PPS IN aceita sinais TTL 0 V a 3.3 V e TTL 0 V a 5 V.
MIMO EXPANSION	A porta de interface MIMO <i>EXPANSION</i> conecta dois dispositivos USRP usando um cabo MIMO compatível.
GB ETHERNET	A porta <i>Ethernet gigabit</i> aceita um conector RJ-45 e um cabo compatível com <i>Ethernet gigabit</i> (Categoria 5, Categoria 5e ou Categoria 6).
POWER	A entrada de energia aceita um conector de alimentação DC externo de 6 V, 3 A.

Tabela 3. LEDs do Módulo NI USRP-2920

Fonte: National Instruments, 2015

LED	Indicação
A	Indica o status de transmissão do módulo: <i>OFF</i> - O módulo não está transmitindo dados. VERDE — O módulo está transmitindo dados.
B	Indica o status do link do cabo MIMO físico: <i>OFF</i> - Os módulos não estão conectados usando o cabo MIMO. VERDE - Os módulos são conectados usando o cabo MIMO.
C	Indica o status de recebimento do módulo: <i>OFF</i> - O módulo não está recebendo dados. VERDE — O módulo está recebendo dados.
D	Indica o status do <i>firmware</i> do módulo: <i>OFF</i> - O <i>firmware</i> não está carregado. VERDE — O <i>firmware</i> está carregado.
E	Indica o status de bloqueio de referência do LO no módulo: <i>OFF</i> - Não há sinal de referência, ou o LO não está bloqueado para um sinal de referência. PISCANDO - O LO não está bloqueado para um sinal de referência. VERDE - O LO está bloqueado para um sinal de referência.
F	Indica o status de energia do módulo: <i>OFF</i> - O módulo está desligado. VERDE — O módulo está ligado.

Passo a passo da instalação

1 – Rádio



Figura 16. Radio USRP

Fonte: Autoria Própria

2 – Tire o protetor de conexão do Rx1-Tx1.



Figura 17. Tirando proteção de encaixe da antena

Fonte: Autoria Própria

3 – Conecte a antena



Figura 18. Conectando antena

Fonte: Autoria Própria



Figura 19. Antena conectada

Fonte: Autoria Própria

4 – Conecte o cabo *Ethernet*



Figura 20. Conectando cabo *Ethernet* no Rádio

Fonte: Autoria Própria

5 – Conecte o cabo de *Ethernet* no PC

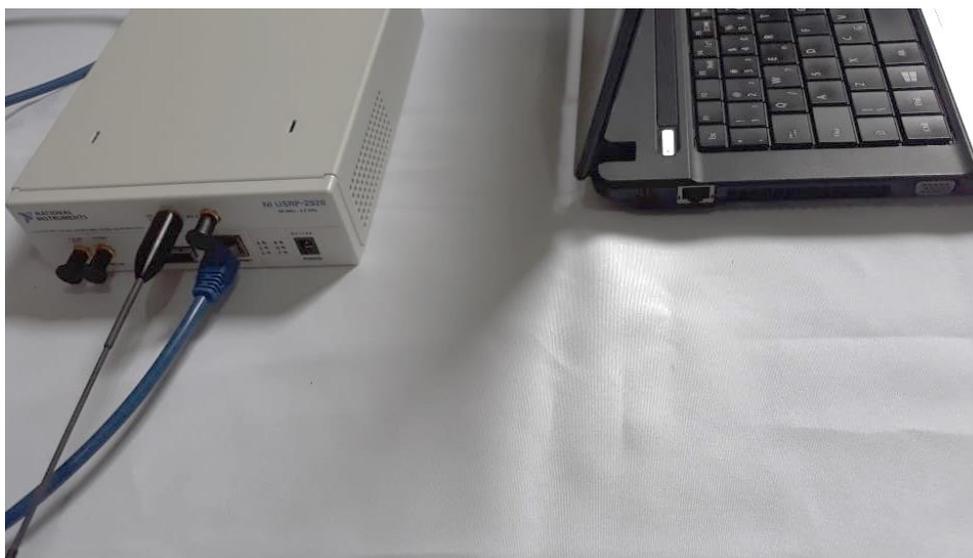


Figura 21. Conecte cabo no PC

Fonte: Autoria Própria



Figura 22. Conectando cabo Ethernet no PC

Fonte: Aatoria Própria



Figura 23. Cabo Ethernet conectado.

Fonte: Aatoria Própria

6 – Conecte o cabo de alimentação no Rádio e posteriormente plugar na tomada.

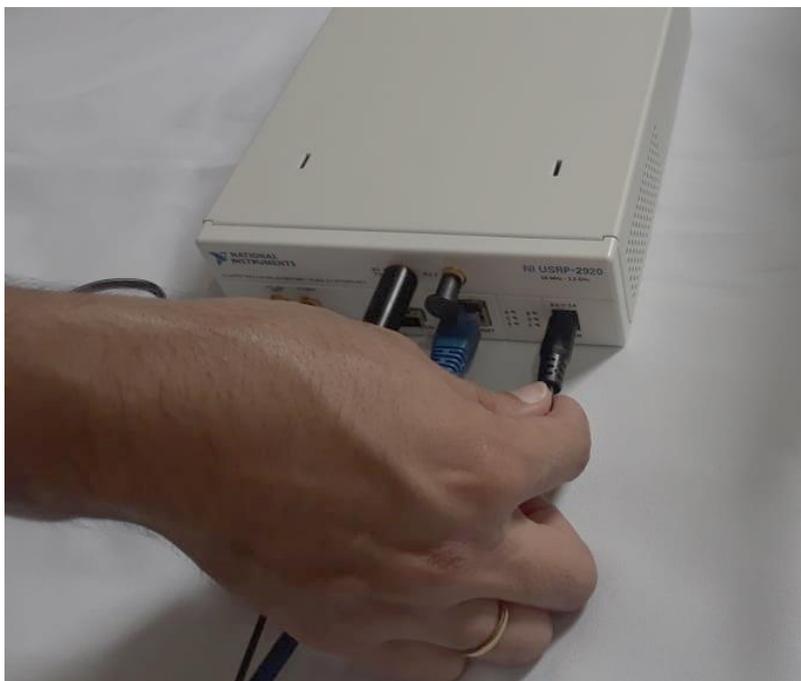


Figura 24. Conectando fonte no Rádio

Fonte: Autoria Própria



Figura 25. Conectando fonte na tomada

Fonte: Autoria Própria

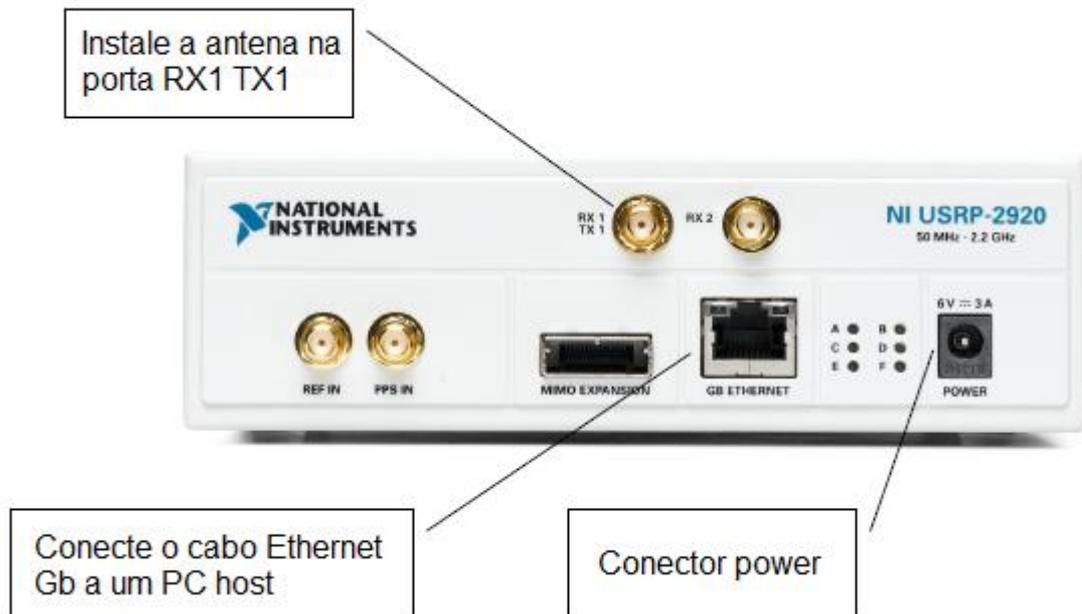


Figura 26. Indicações de Instalações e Conexões

Fonte: www.ni.com

LabVIEW software e seus módulos

LabVIEW 64bits

<http://www.ni.com/download/labview-development-system-2018/7413/en/>

LabVIEW 32bits

<http://www.ni.com/download/labview-development-system-2018/7406/en/>

NI USRP 2920 software

É possível baixar o programa diretamente no site da *National Instruments* <http://www.ni.com/pt-br/support/downloads/drivers/download.ni-usrp.html> onde foi utilizado a versão 18.5.

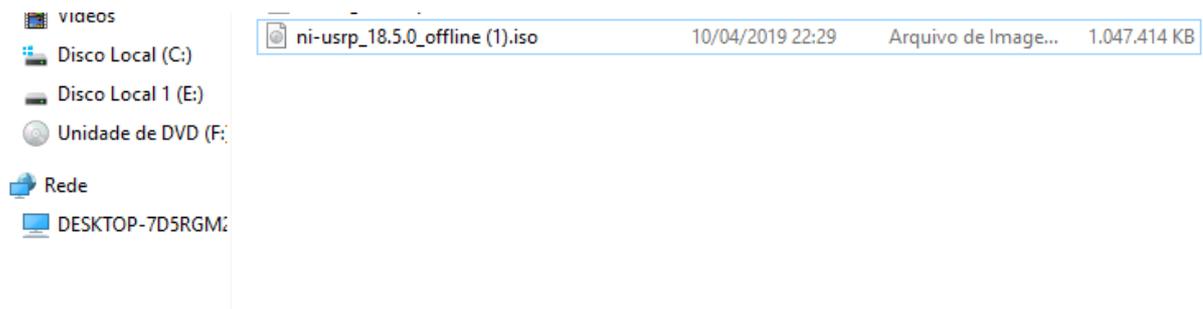


Figura 27. Arquivo de instalação do programa NI USRP iso

Fonte: Autoria Própria

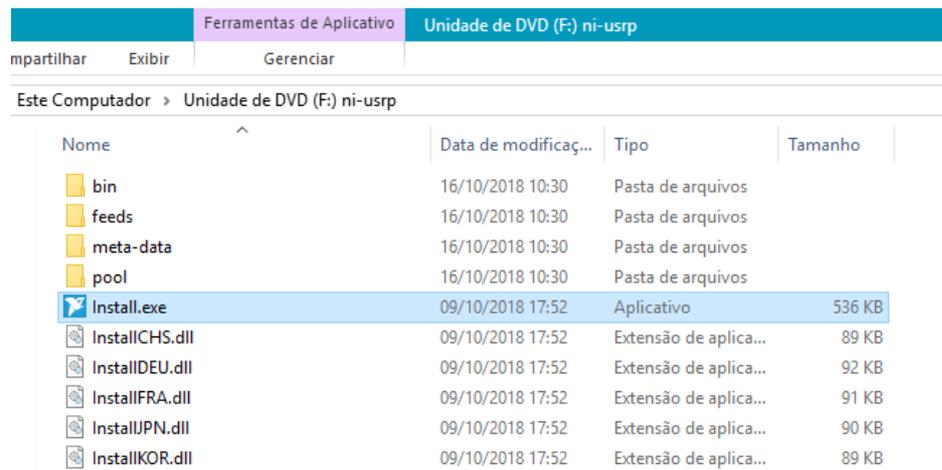


Figura 28. Execução do setup

Fonte: Autoria Própria

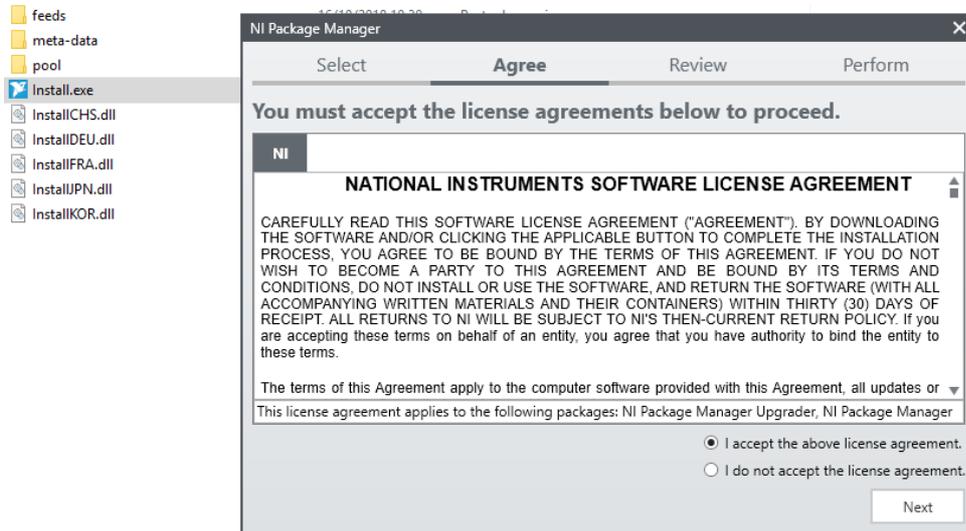


Figura 29. Início do processo de instalação do software NI USRP

Fonte: Autoria Própria

Definições para o correto funcionamento do software de acordo com a tabela 4.

Tabela 4. Endereços IP Estáticos

Fonte: National Instruments, 2015

Componente	Endereço
Endereço IP estático da interface <i>Ethernet</i> do <i>host</i>	192.168.10.1
Máscara de sub-rede da interface <i>Ethernet</i> do <i>host</i>	255.255.255.0
Endereço IP do dispositivo <i>USRP</i> padrão	192.168.10.2

Ip do host

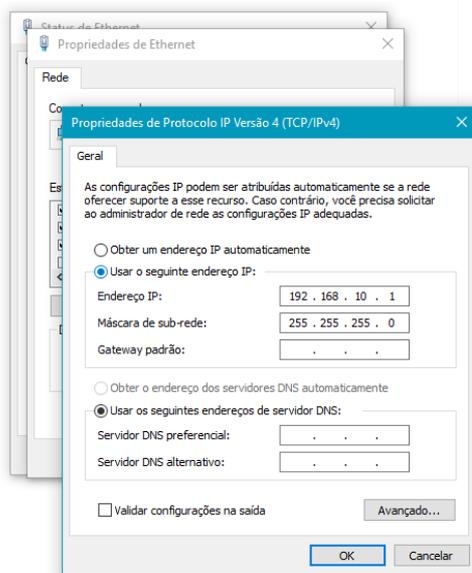


Figura 30. Configurando o host

Fonte: Autorial Própria

Ip do USRP

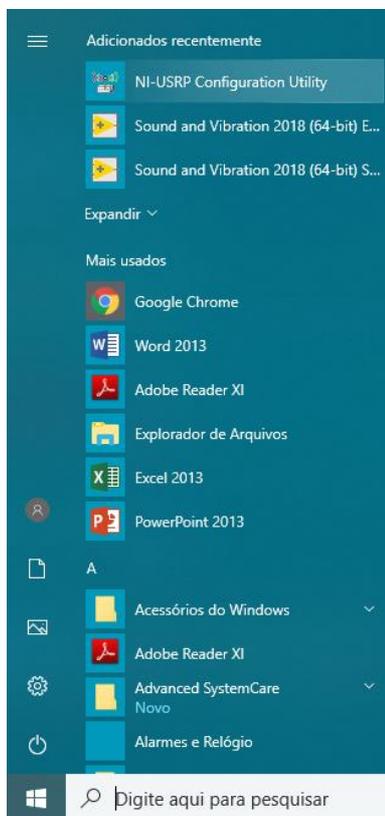


Figura 31. Localização do programa

Fonte: Autorial própria

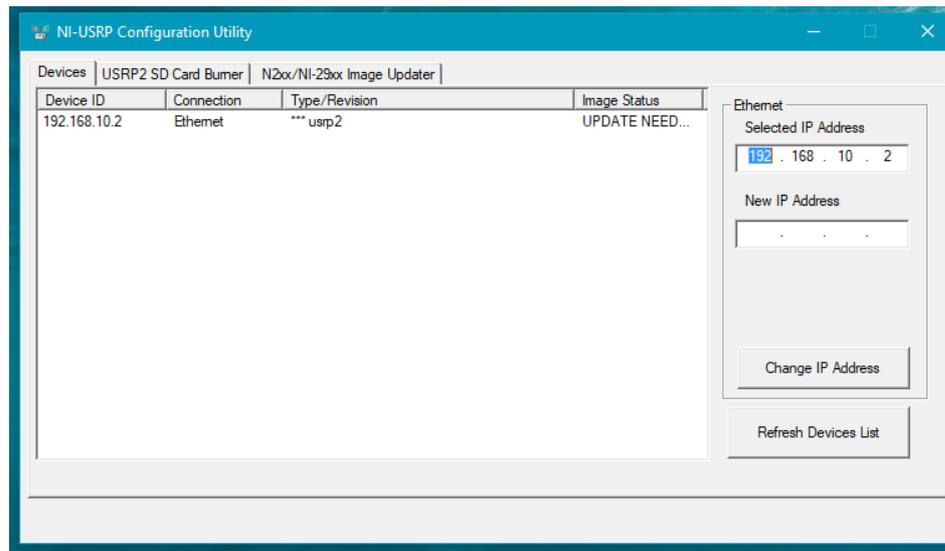


Figura 32. Configurando Endereço do IP do USRP

Fonte: Autoria Própria

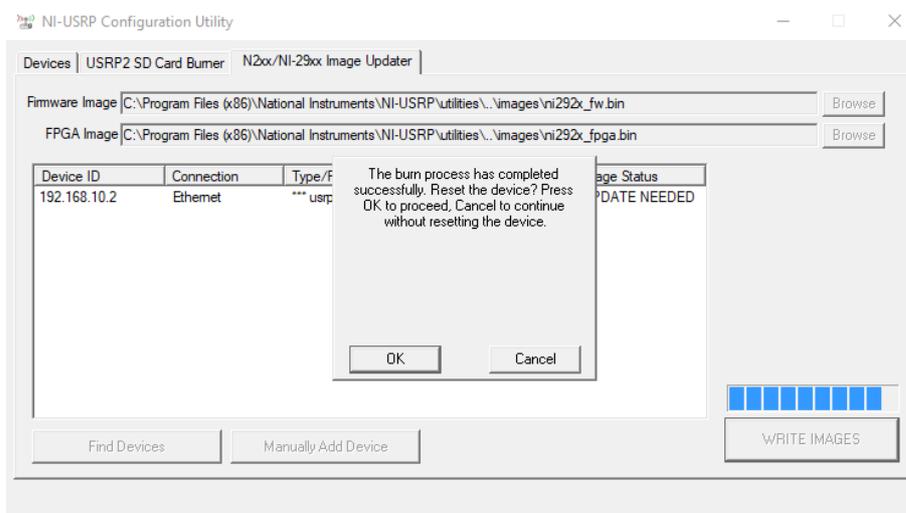


Figura 33. Atualizando Firmware USRP N2xx / NI 29xx e Imagens FPGA

Fonte: Autoria Própria

Observamos na figura abaixo os programas que devem ser instalados no PC ou notebook, para o êxito do projeto:

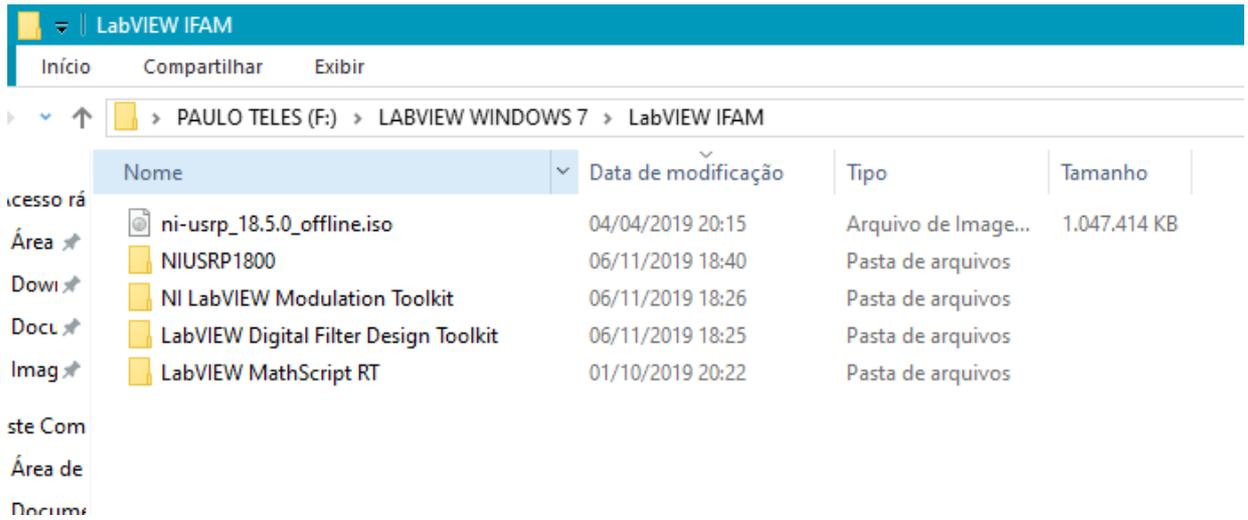


Figura 34 - Arquivos de Instalação para o projeto NI USRP.

Fonte: Autoria Própria.

4.2 Laboratório 2 – Rádio FM

4.2.1 A camada física - *Front End*

A filosofia básica por trás da camada física é fazer com que o processamento de forma de ondas seja realizado na CPU de um computador. Todas as operações de alta velocidade como elevação de frequência (*upconversion*) e redução de frequência (*downconversion*), decimação e interpolação, são realizadas em uma matriz de portas de campo programáveis (do inglês, *Field Programmable Gate Array – FPGA*) ou em um circuito integrado de aplicação específica (do inglês, *Application Specific Integrated Circuits – ASIC*) do *front end* (OLIVEIRA, LEITE, et al., 2012).

Implementação de uma Modulação em Frequência

Em um diagrama básico de um transmissor FM (do inglês, *Frequency Modulation*), o sinal de entrada é gerado por um microfone e que, depois de passar por um ramo de amplificação, é introduzido num misturador de frequências. Chegando no misturador que ocorre a modulação propriamente dita, pois o sinal de informação é modulado pela portadora que foi gerada no oscilador local. Em seguida a onda FM passa por um dobrador de frequência e então segue para o amplificador de potência e para a antena. No receptor FM, em que o tratamento do sinal é feito no sentido inverso ao do transmissor. Primeiramente o sinal FM passa pelo amplificador de RF e entra em um misturador no qual a frequência do sinal é reduzida com base no oscilador local para a frequência intermediária – FI, utilizada nos próximos estágios da recepção. Assim, passa por mais um estágio de amplificação, nesse momento o sinal é limitado em amplitude, eliminando, assim, os ruídos da transmissão. O discriminador, extrai a portadora e deixa apenas o sinal de voz, que é novamente amplificado para reprodução nos autôfalantes.

4.2.2 Passo a passo no *LabVIEW* de recepção de Rádio FM



Figura 35. Abrindo LabVIEW

Fonte: Autoria própria

1º Passo - Clique no aplicativo do NI *LabVIEW* para abri-lo. Assim que ele (*LabVIEW*) abrir procure *Create Project* e clique para iniciar seu projeto.

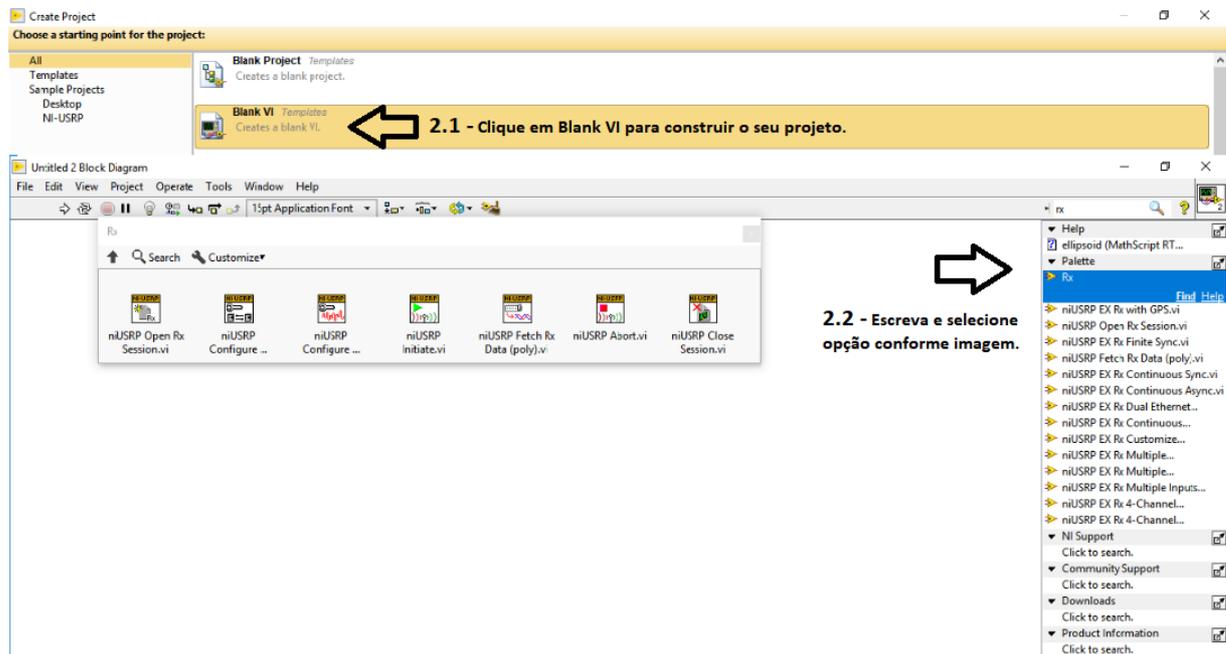


Figura 36. Selecionando Blank VI e começando a construir o circuito.

Fonte: Autoria Própria

2º Passo – Selecione a opção *Blank VI* para criar o seu programa. Após passo anterior, na opção pesquisar digite conforme imagem “RX”.

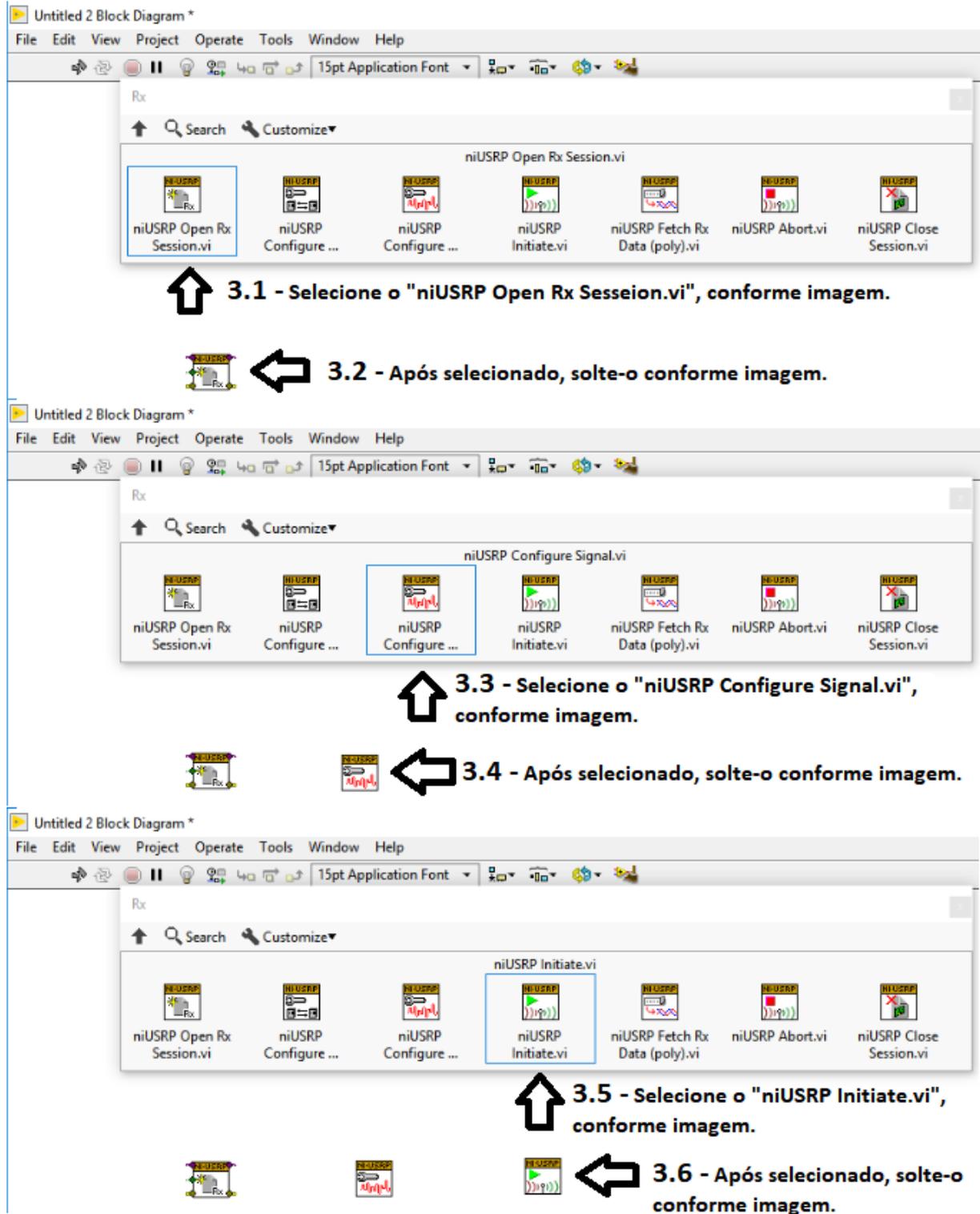


Figura 37. Selecionando os VI para a montagem do circuito.

Fonte: Autoria Própria

3º Passo – Selecione os 3 VIs conforme imagem, do passo a passo.

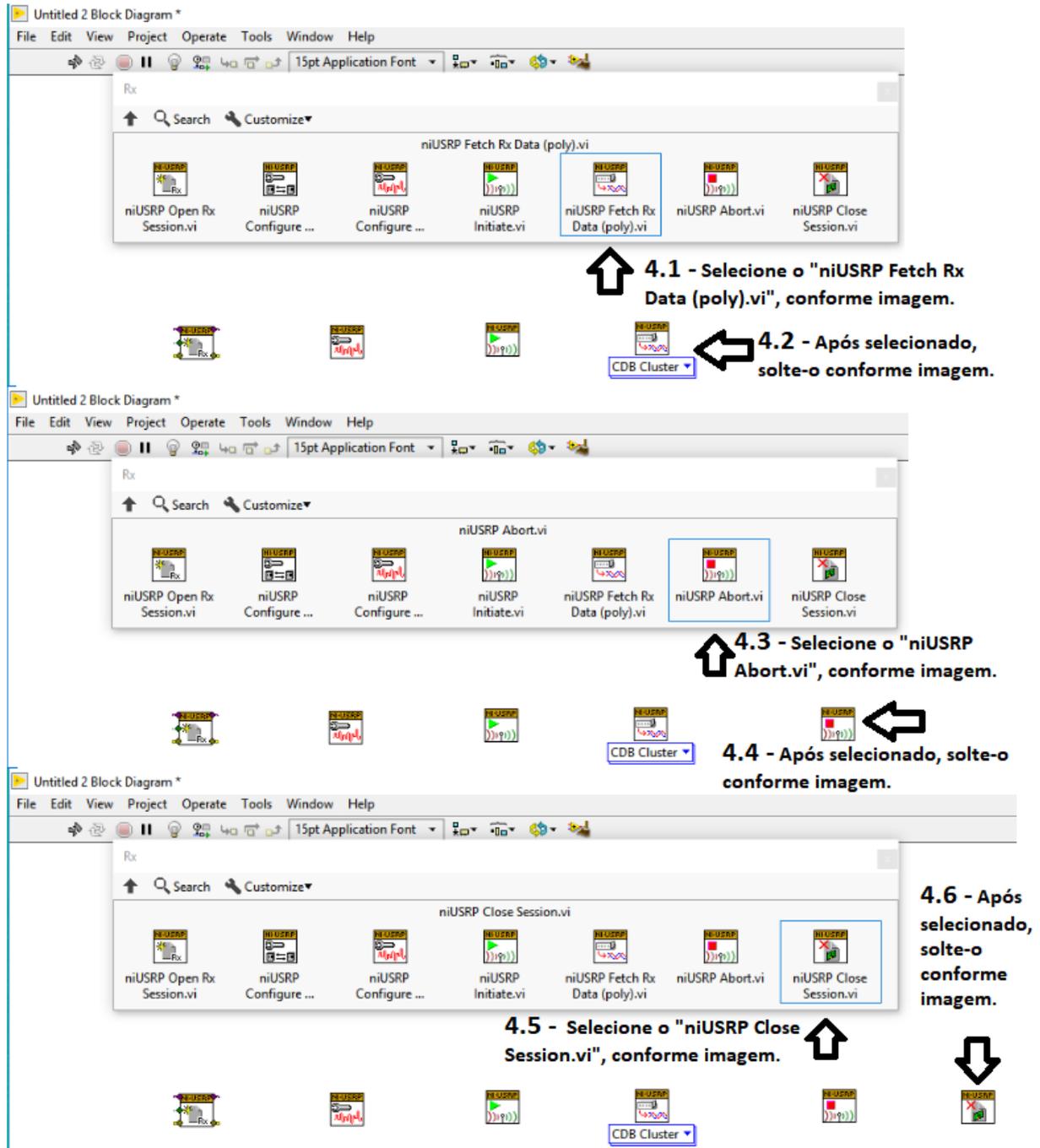


Figura 38. Selecionando mais VI para a montagem do circuito.

Fonte: Autoria Própria.

4º Passo – Selecione os próximos 3 VIs que compõem o circuito básico deste projeto em questão.

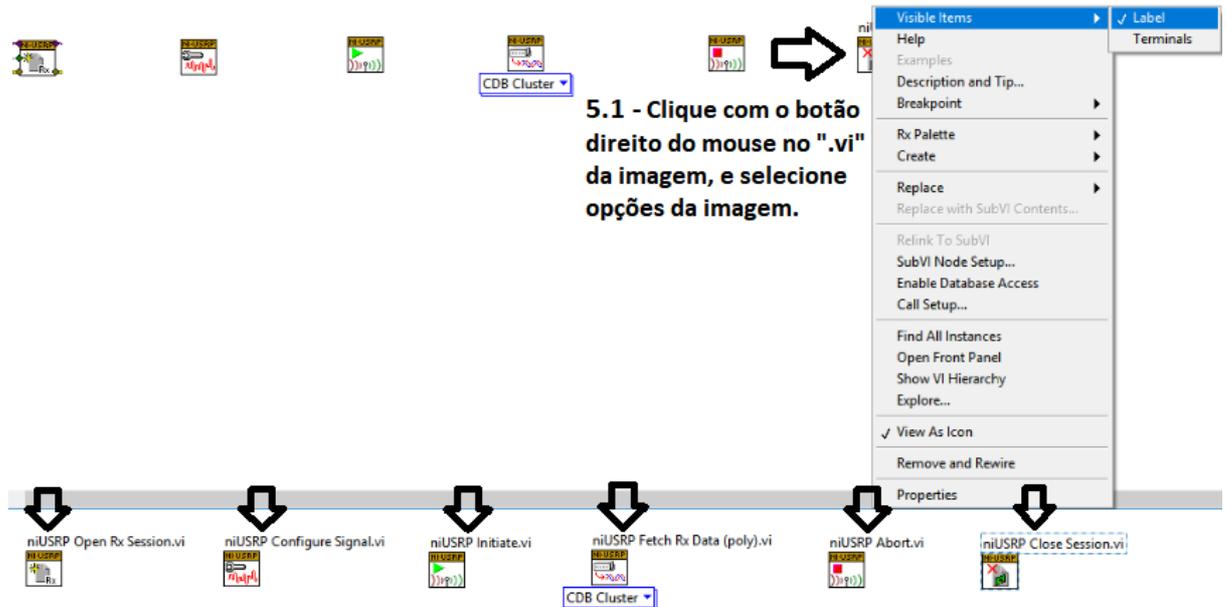


Figura 39. Comando para visualizar nomes dos VIs usados no Projeto.

Fonte: Autoria Própria.

5º Passo – Clicando com o botão direito do mouse, poderá ver o nome de cada VI usado em seu projeto.

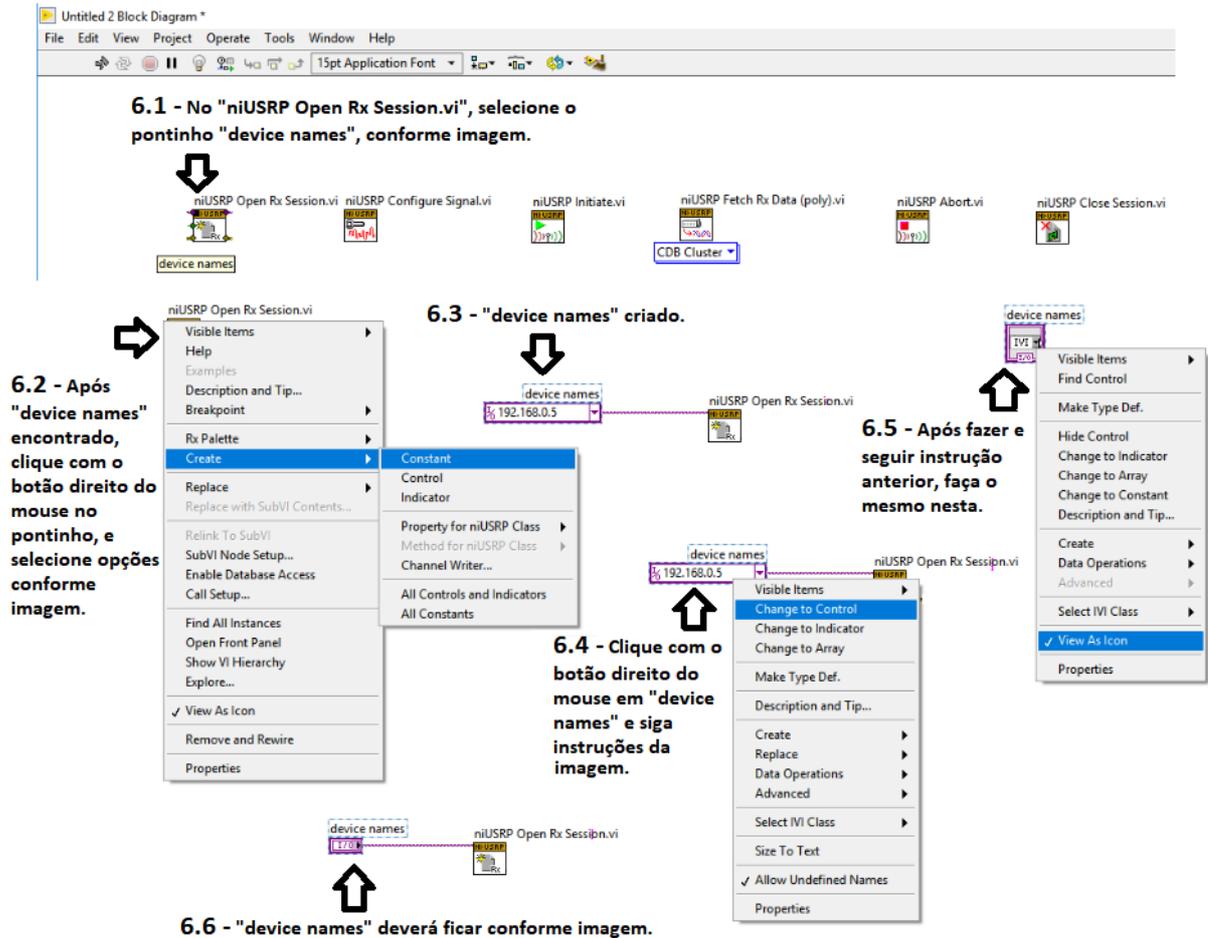


Figura 40. Criando "Device Name", onde selecionaremos o IP do Rádio.

Fonte: Autoria Própria.

6º Passo – Criação do *Device Names*, nele será imputado o IP de *Ethernet* que será o mesmo do Rádio. Siga as instruções da imagem, conforme sequência informada. Clicando com o botão direito do mouse para fazer as devidas configurações.

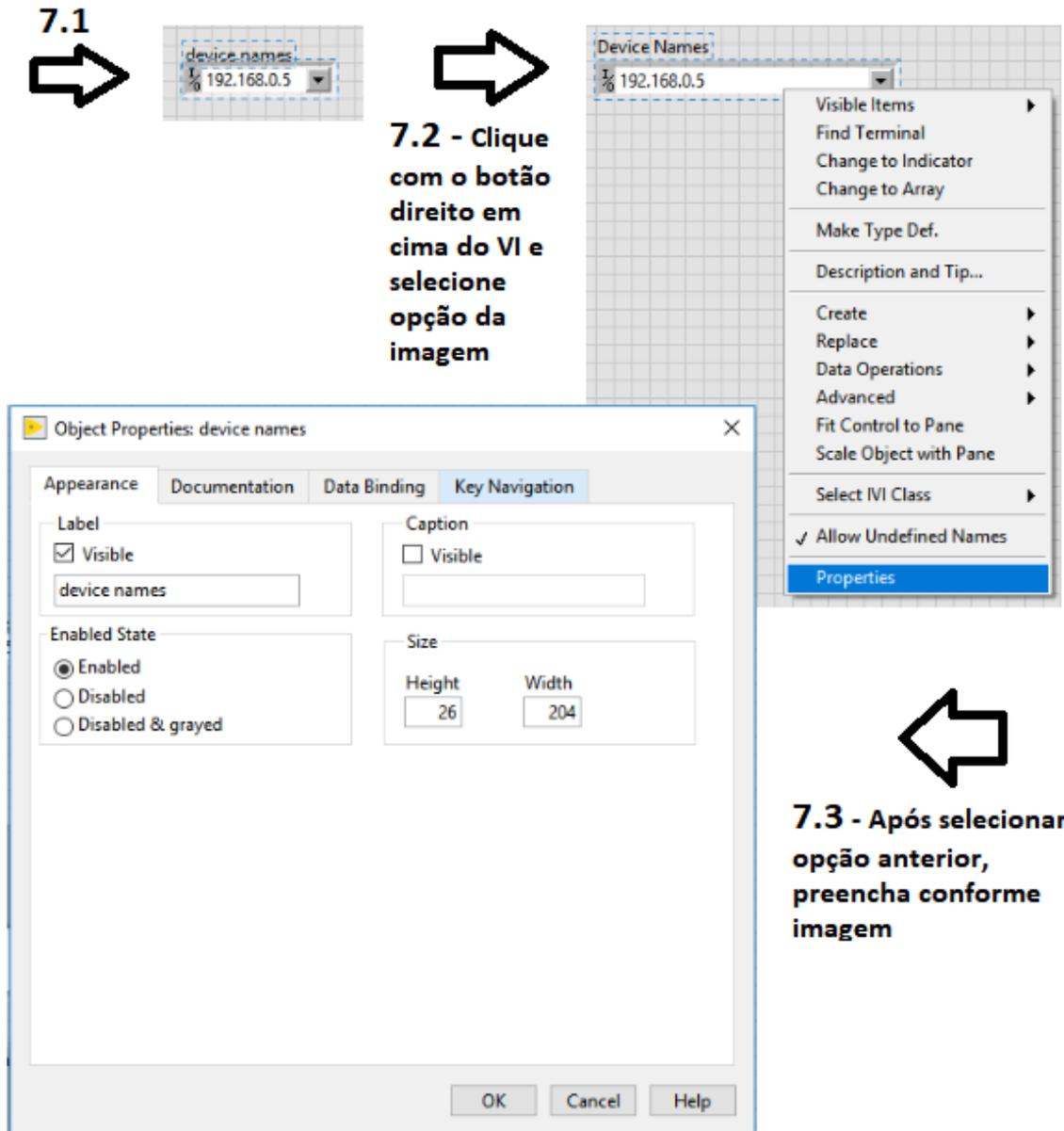


Figura 41. Configurando "Device Name"

Fonte: Autoria Própria.

7º Passo – No Painel Frontal, siga os passos conforme imagem. Clicando com o botão direito do mouse. E seu *Device Names* foi criado com sucesso.

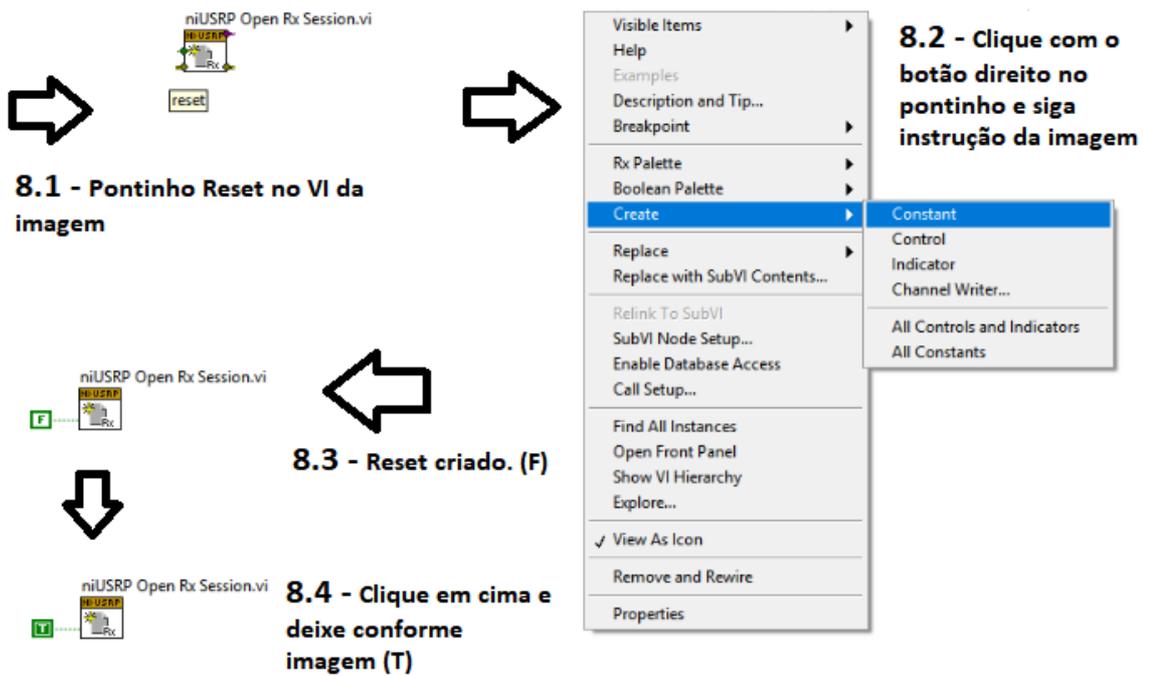


Figura 42. Criando "Reset".

Fonte: Autoria Própria.

8º Passo – Criando “Reset”, clicando com o botão direito do mouse e seguindo imagem.

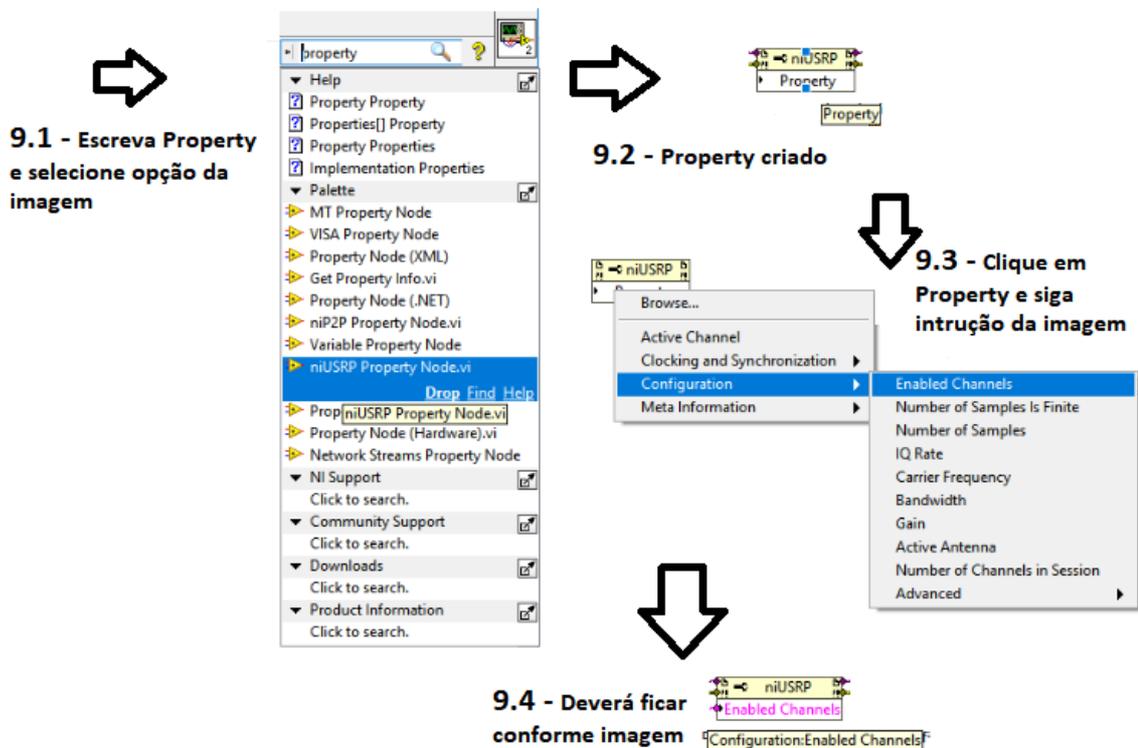


Figura 43. Criando "niUSRP Property Node.vi"

Fonte: Autoria Própria.

9º Passo – Criando “Property Node VI”.

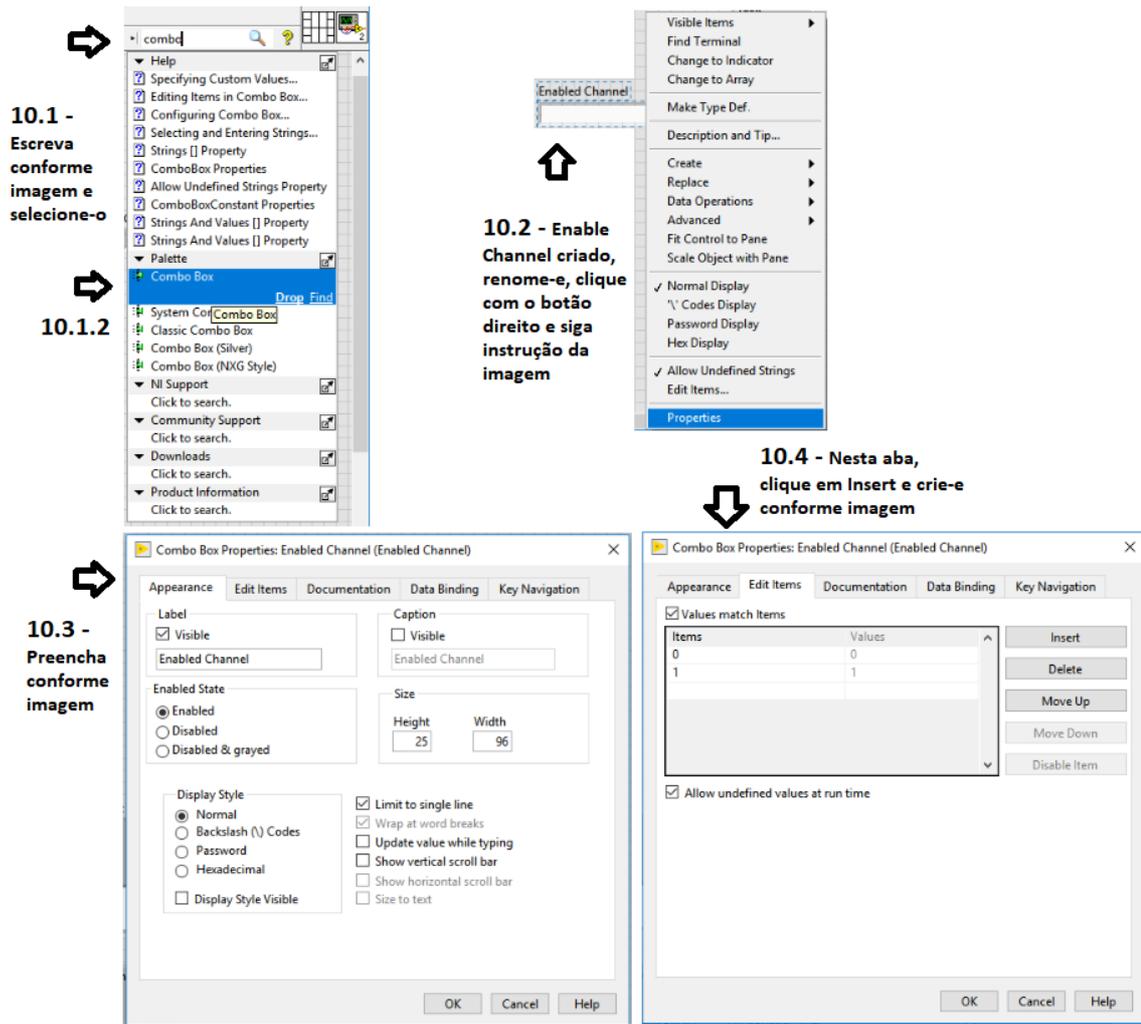


Figura 44. Criando "Enabled Channel".

Fonte: Autoria Própria.

10º Passo – Criando "Enabled Channel" (Canal que a antena está conectada ao rádio), clicando com o botão direito do mouse e seguindo instruções da imagem.

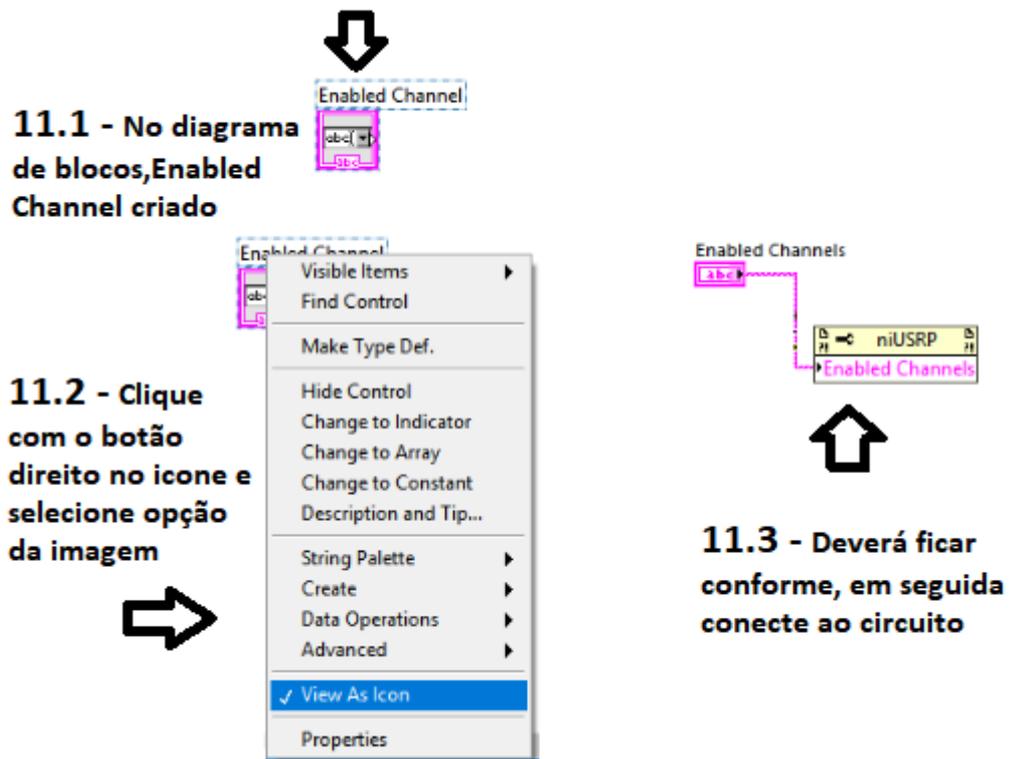


Figura 45. Criando "Enabled Channel" no Block Diagram.

Fonte: Autoria Própria.

11º Passo – Criando "Enabled Channel" no Diagrama de blocos, clicando com o botão direito do mouse e seguindo instruções da imagem.

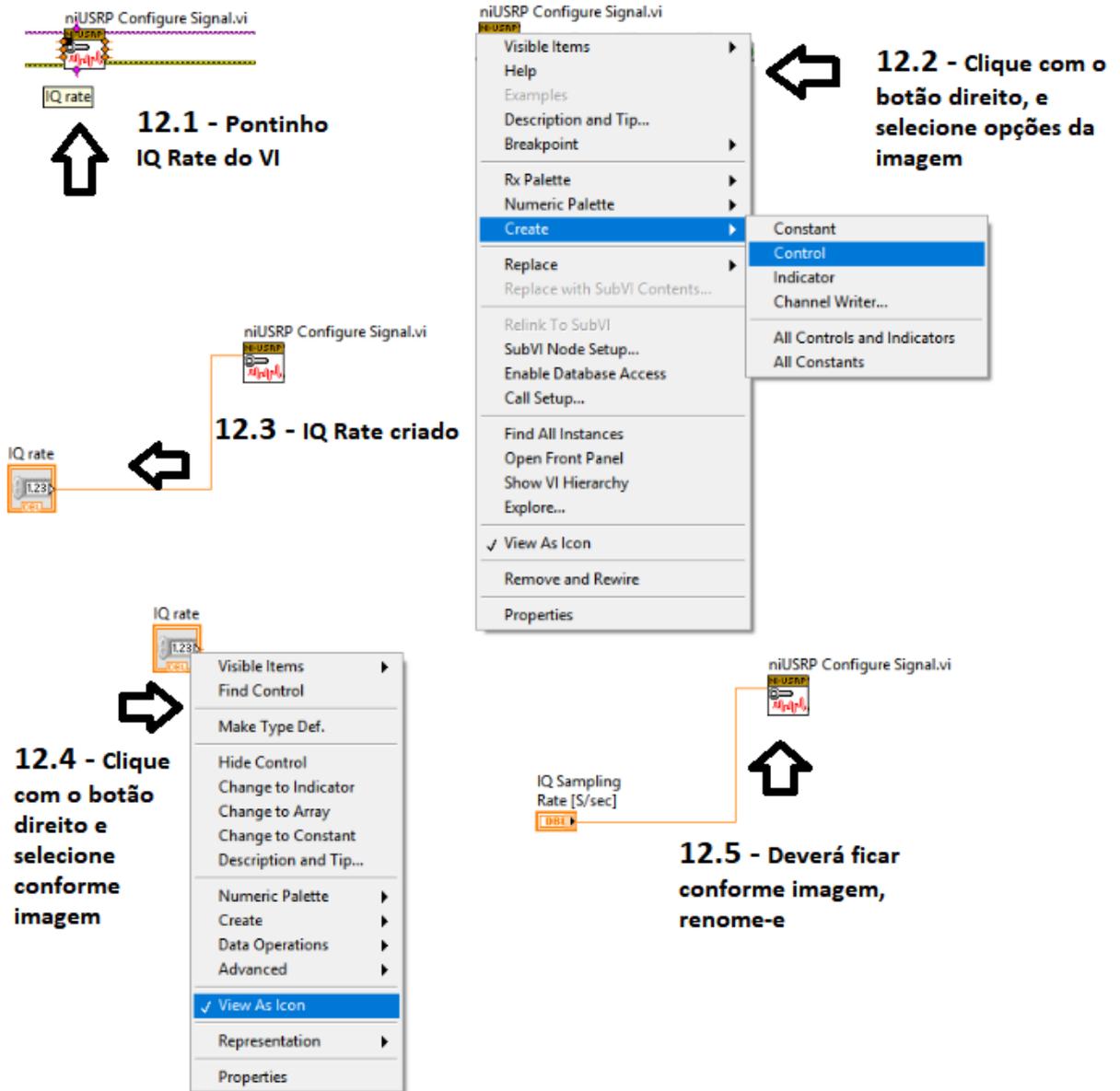
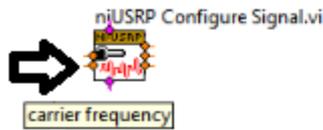


Figura 46. Criando "IQ Rate"

Fonte: Autoria Própria.

12º Passo – Criando "IQ Rate", clicando com o botão direito do mouse e seguindo instruções da imagem.

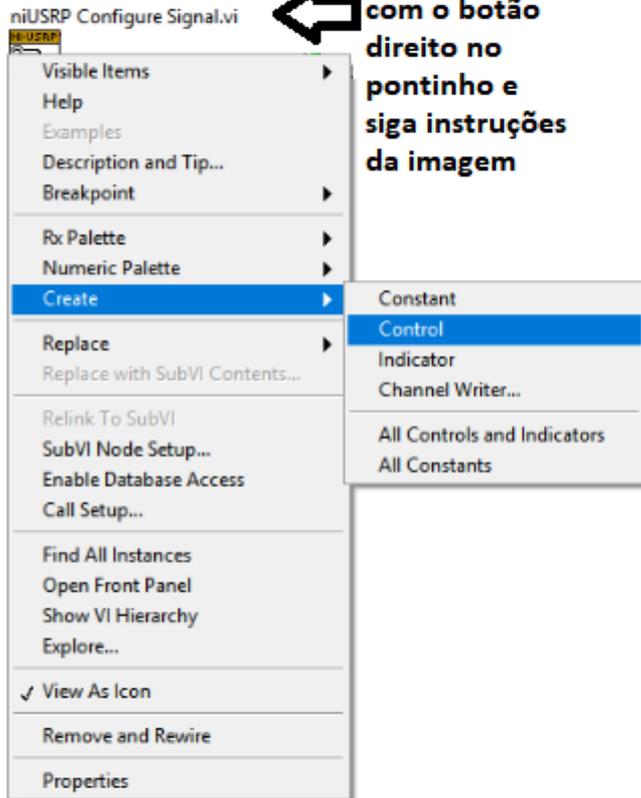
**13.1 - Pontinho
Carrier Frequency**



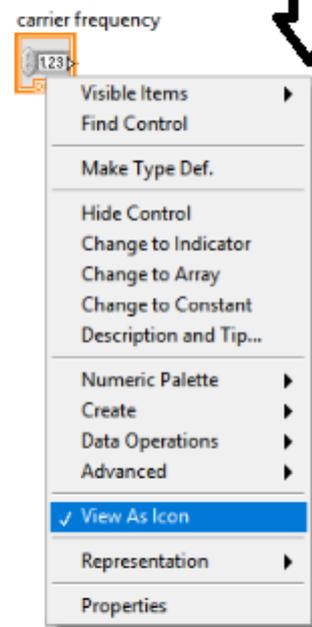
**13.3 - Carrier
Frequency criado**



**13.2 - Clique
com o botão
direito no
pontinho e
siga instruções
da imagem**



**13.4 - Clique com o botão
direito do mouse, siga
instrução, selecionando
opção da imagem**



**13.5 - Carrier
Frequency deverá ficar
conforme imagem**

Figura 47. Criando "Carrier Frequency [Hz]"

Fonte: Autoria Própria.

13º Passo – Criando "Carrier Frequency [Hz]", clicando com o botão direito do mouse e seguindo instruções da imagem.

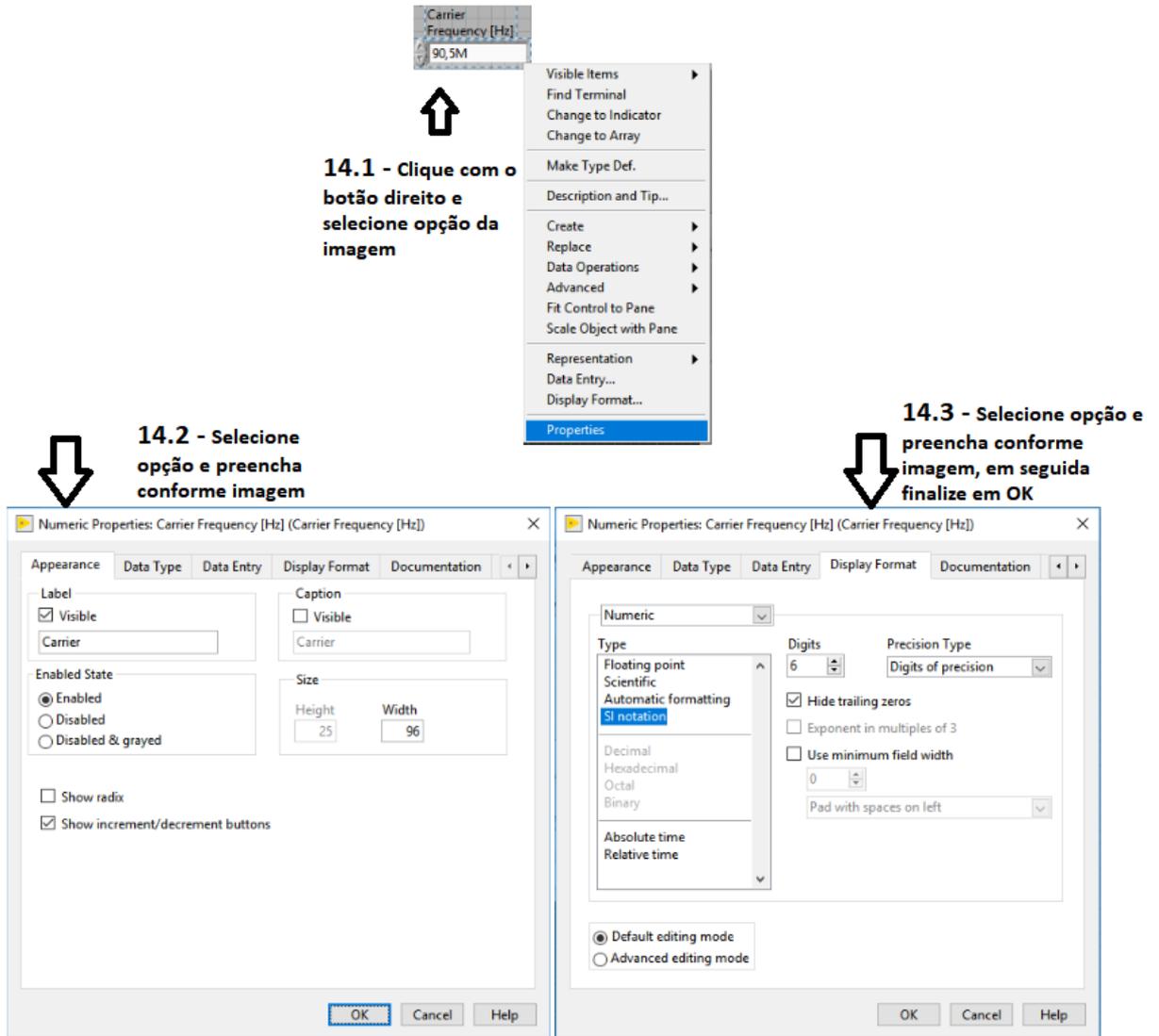
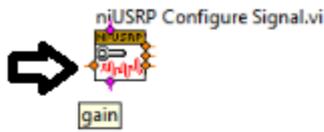


Figura 48. Configurando "Carrier Frequency" no Painel Frontal

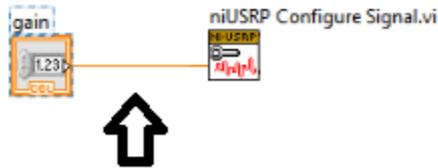
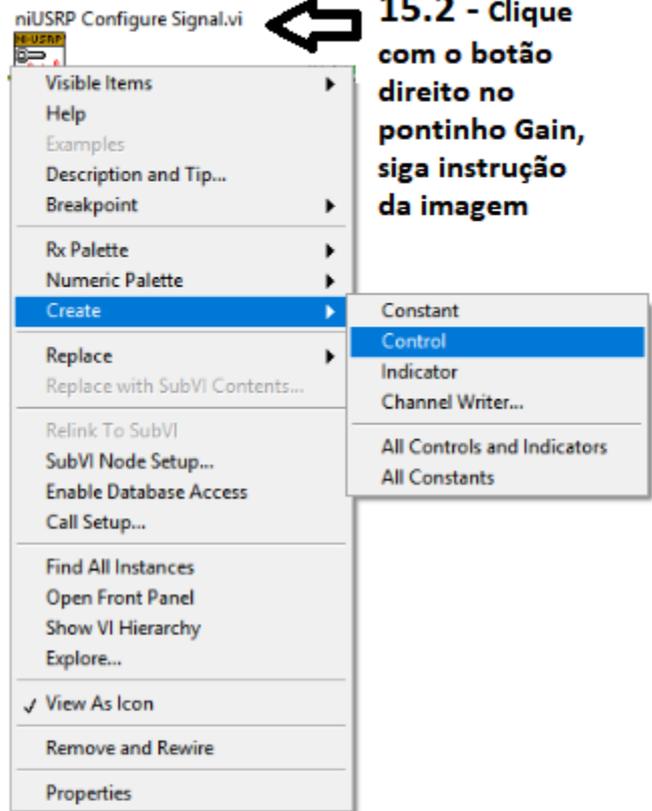
Fonte: Autoria Própria.

14º Passo – Carrier Frequency no Painel Frontal.

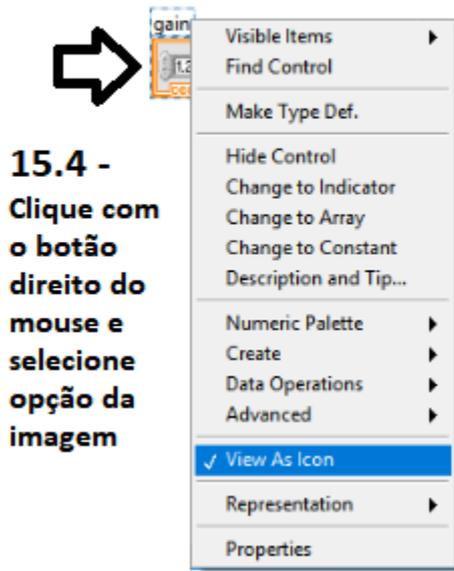
15.1 - Criando Gain (dB)



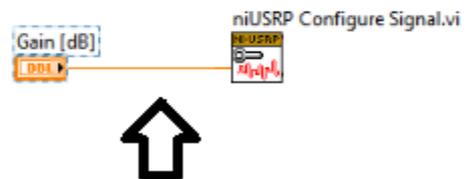
15.2 - Clique com o botão direito no pontinho Gain, siga instrução da imagem



15.3 - Gain (dB) criado



15.4 - Clique com o botão direito do mouse e selecione opção da imagem



15.5 - Gain (dB) deverá ficar conforme imagem

Figura 49. Criando "Gain (dB)"

Fonte: Autoria Própria

15º Passo – Criando Gain (dB).

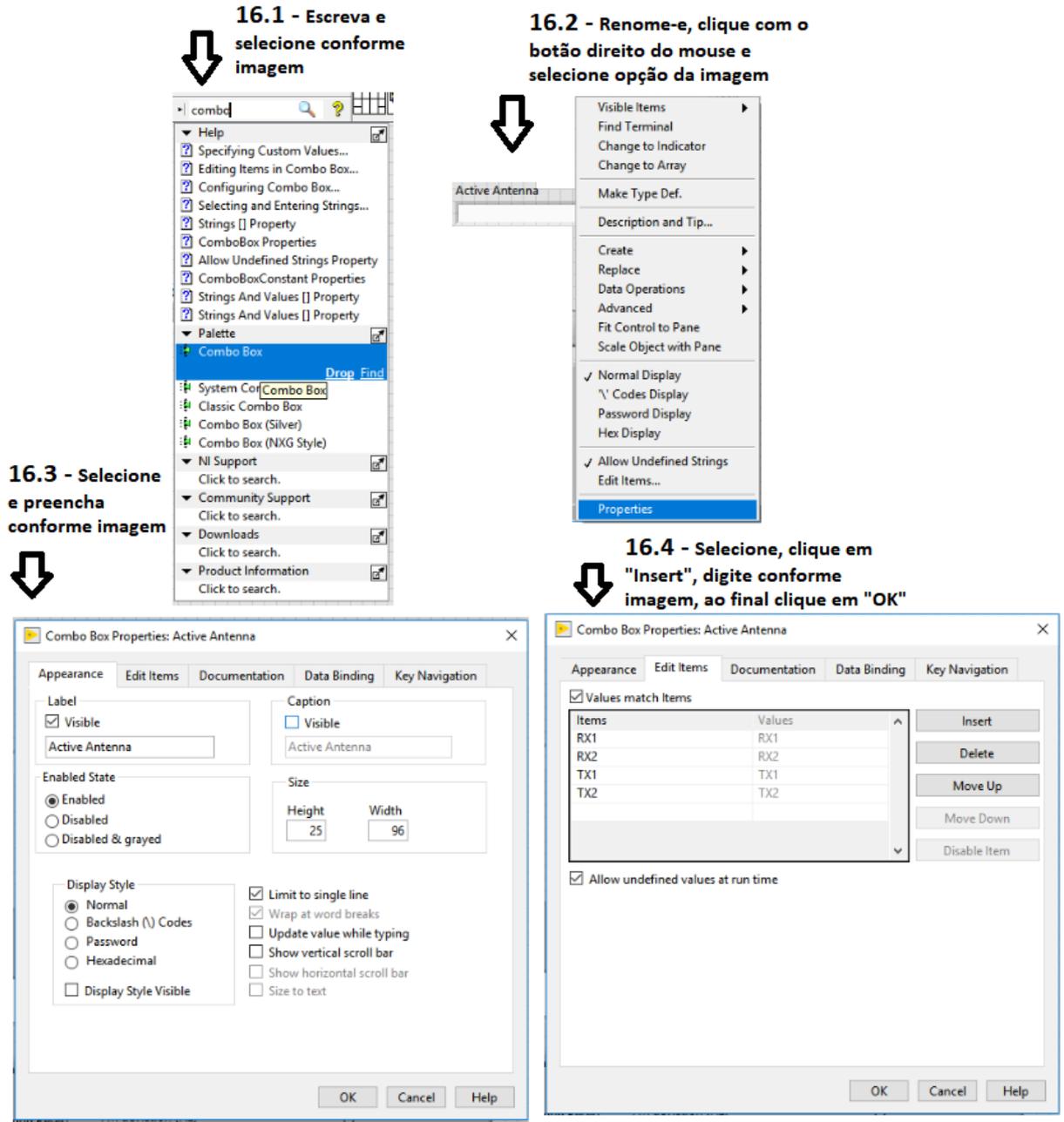
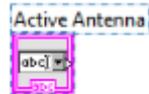


Figura 50. Criando "Active Antenna"

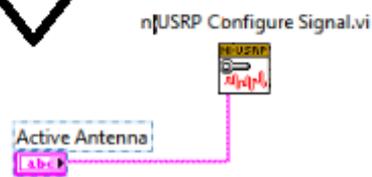
Fonte: Autoria Própria.

16º Passo – Criando Active Antenna.

17.1 - Active Antenna criado no Diagrama de blocos



17.3 - "Active Antenna" deverá ficar conforme imagem, conecte ao circuito



17.2 - Clique com o botão direito do mouse em cima, selecione opção da imagem

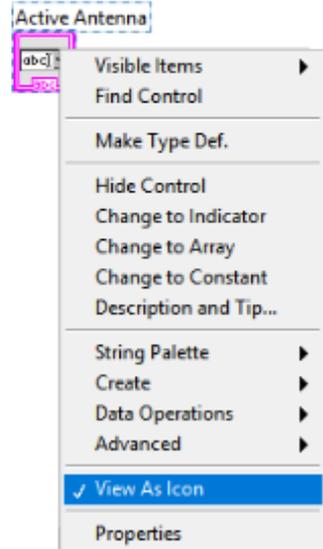
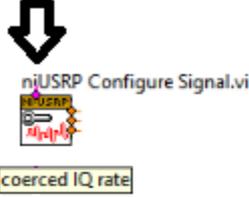


Figura 51. Configurando "Active Antenna" no Bloco de Diagrama

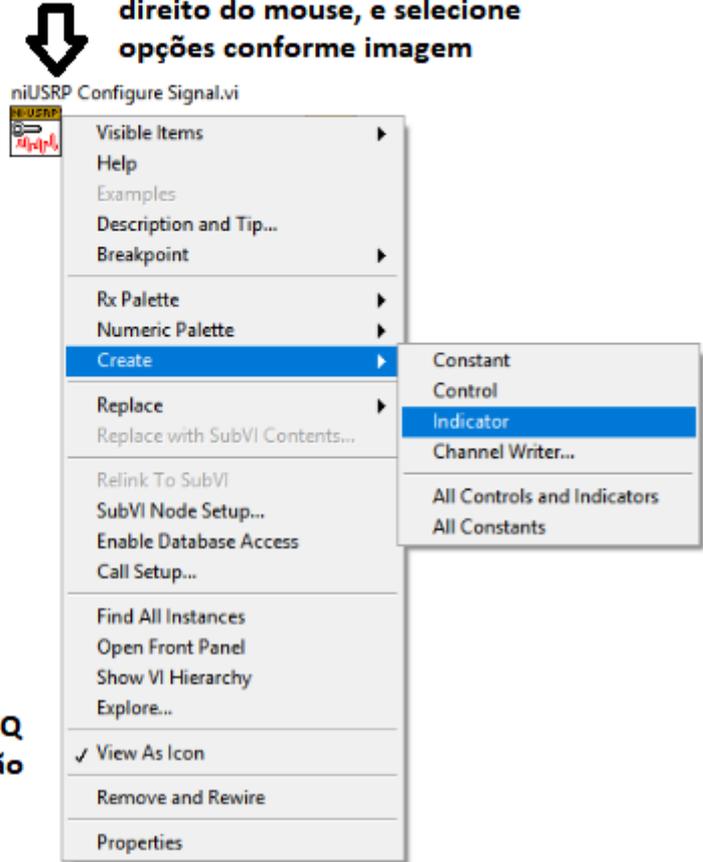
Fonte: Autoria Própria.

17º Passo – No Bloco de diagrama, formate e conecte ao circuito, de acordo com instruções.

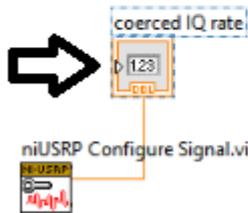
18.1 - Criando "IQ Rate"
cursor do mouse em cima do pontinho do "IQ Rate" no VI da imagem



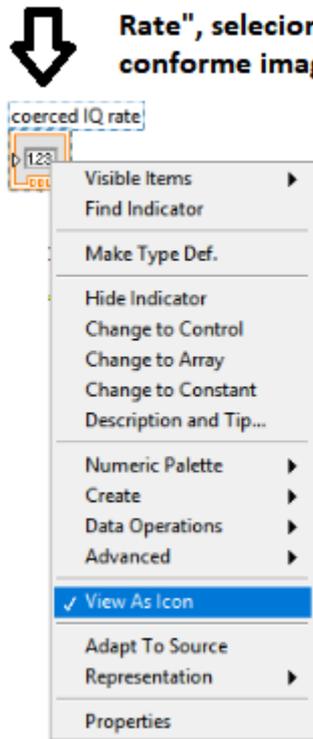
18.2 - Clique com o botão direito do mouse, e selecione opções conforme imagem



18.3 - "IQ Rate" criado



18.4 - Clique com o botão direito do mouse em cima do "IQ Rate", selecione opção conforme imagem



18.5 - Renome-e e conecte ao circuito

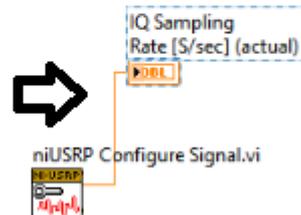
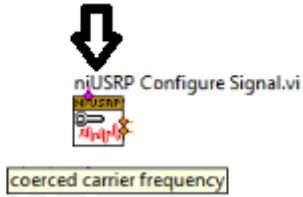


Figura 52. Criando "IQ Sampling Rate"

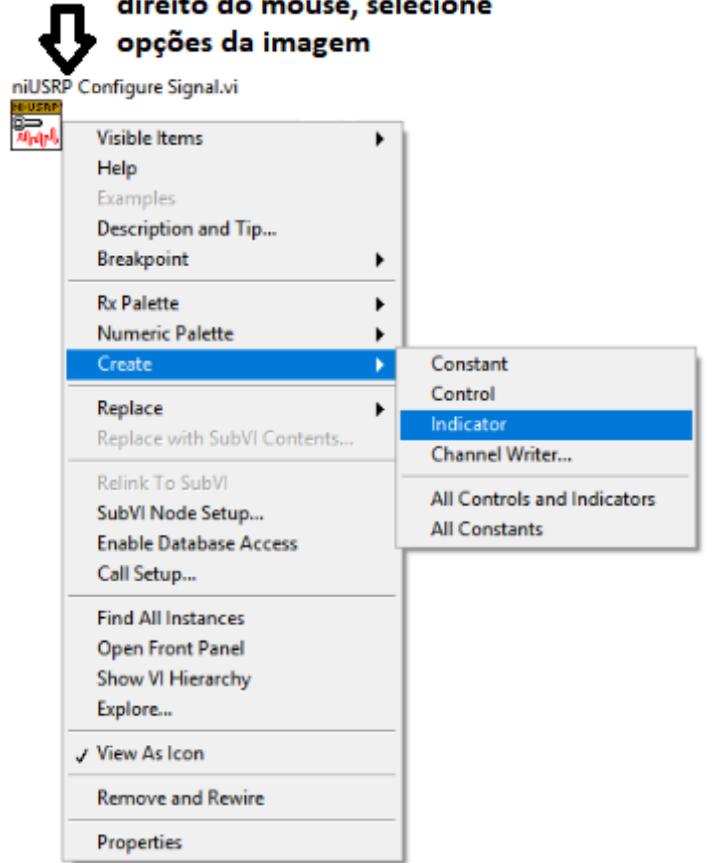
Fonte: Autoria Própria.

18º Passo – Criando IQ Sampling Rate.

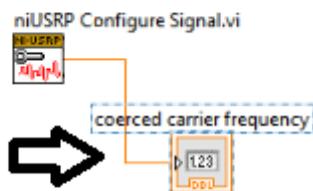
19.1 - Criando "Carrier Frequency", cursor do mouse em cima do pontinho do "Carrier Frequency" no VI da imagem



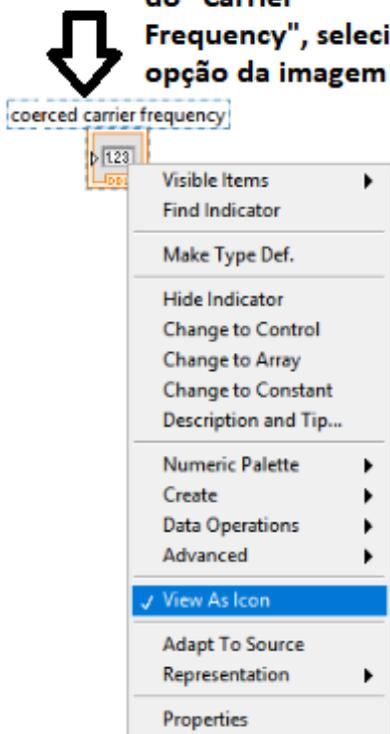
19.2 - Clique com o botão direito do mouse, selecione opções da imagem



19.3 - "Carrier Frequency" criado



19.4 - Clique com o botão direito em cima do "Carrier Frequency", selecione opção da imagem



19.5 - Renome-e e conecte ao circuito.

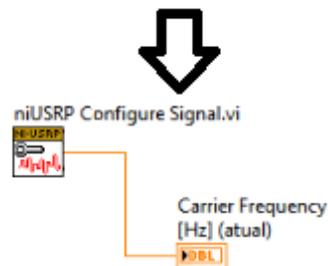
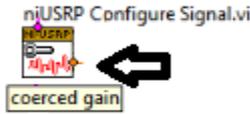


Figura 53. Criando "Carrier Frequency"

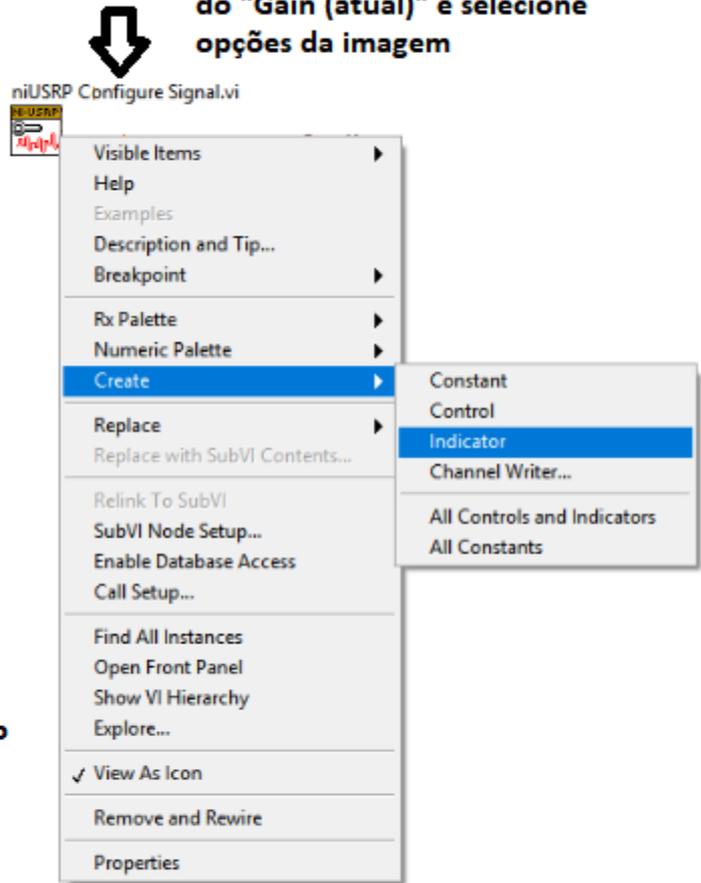
Fonte: Autoria Própria.

19º Passo – Criando Carrier Frequency.

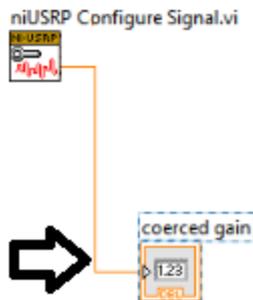
20.1 - Criando "Gain (atual)", cursor do mouse em cima do pontinho do "Gain (atual)" no VI da imagem.



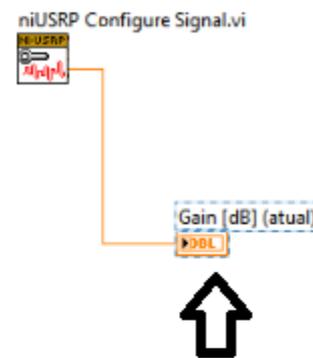
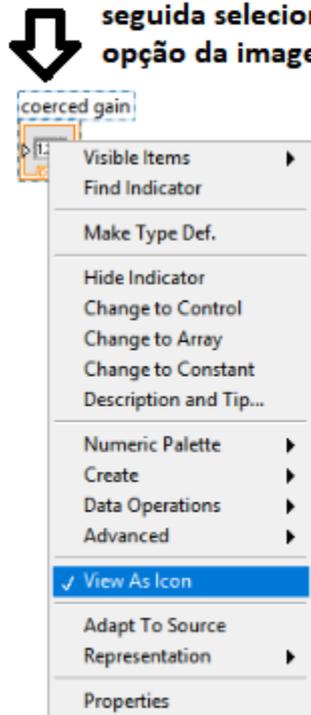
20.2 - Clique com o botão direito do mouse no pontinho do "Gain (atual)" e selecione opções da imagem



20.3 - "Gain (atual)" criado



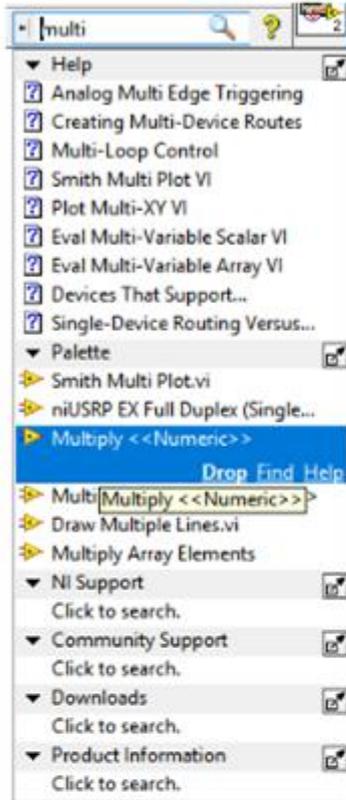
20.4 - Clique com o botão direito do mouse em cima do "Gain (atual)", em seguida selecione opção da imagem



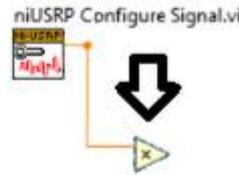
20.5 - Renome-e e conecte ao circuito.

Figura 54. Criando "Gain [dB] (atual)"

21.1 - Escreva, selecione conforme imagem



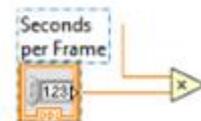
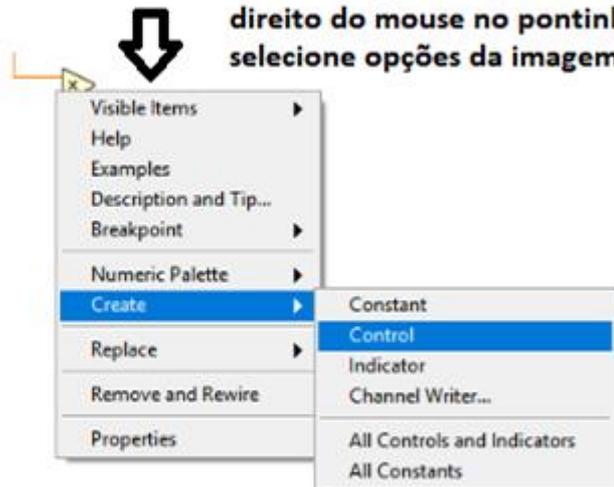
21.2 - "Multiply (Numeric)" criado e conectado ao circuito



21.3 - Cursor do mouse no pontinho de Y no "Multiply (Numeric)"

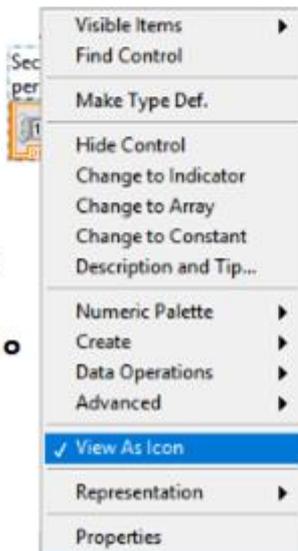


21.4 - Clique com o botão direito do mouse no pontinho, e selecione opções da imagem.



21.5 - "Seconds per Frame" criado.

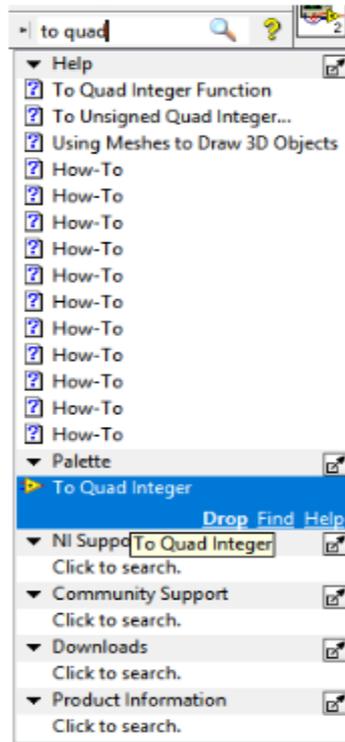
21.6 - Clique com o botão direito do mouse e selecione opção da imagem.



21.7 - "Seconds per Frame" deverá ficar conforme imagem.

Figura 55. Criando "Multiply e Seconds per Frame"

**22.1 - Criando "To Quad Integer",
digite e selecione opção da imagem.**



1164



22.2 - "To Quad Integer" criado



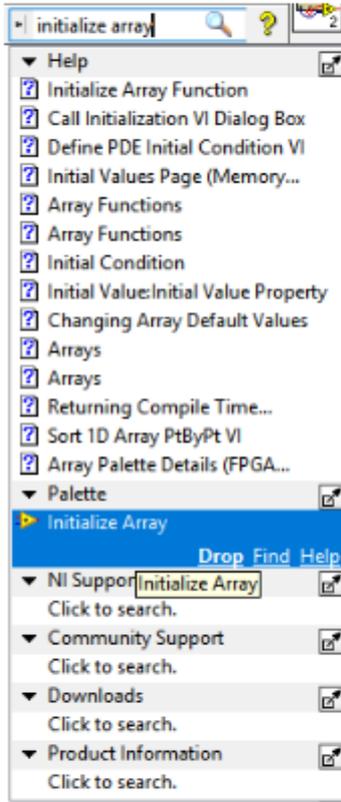
**22.3 - Renome-e e
conecte ao circuito.**

Figura 56. Criando "To Quad Integer"

22º Passo – Criando *To Quad Integer*.

Fonte: Autoria Própria.

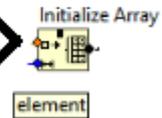
23.1 - Criando "Initialize Array", digite e selecione opção da imagem.



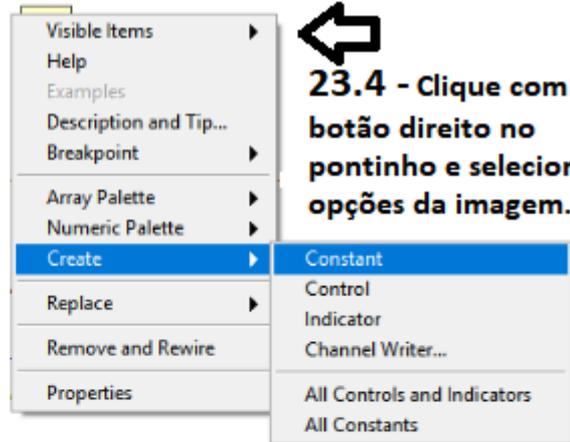
23.2 - "Inicializa Array" criado.



23.3 - Cursor do mouse no pontinho "element".



23.4 - Clique com o botão direito no pontinho e selecione opções da imagem.



23.5 - "Element" criado

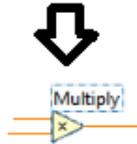


Figura 57. Criando "Initialize Array e Element"

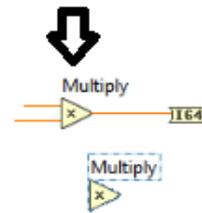
Fonte: Autoria Própria.

23º Passo – Criando Initialize Array e Element que o compõe.

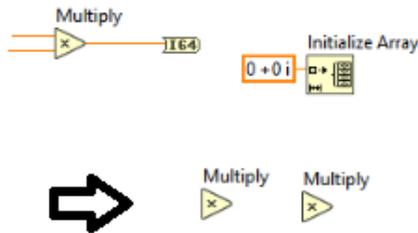
24.1 - Selecione o "Multiply" da imagem



24.2 - Após selecionado clique Ctrl + C, e depois Ctrl + V



24.3 - Repita o procedimento anterior



24.4 - Deverá ficar conforme imagem, conectados.

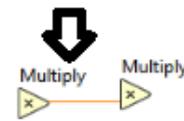
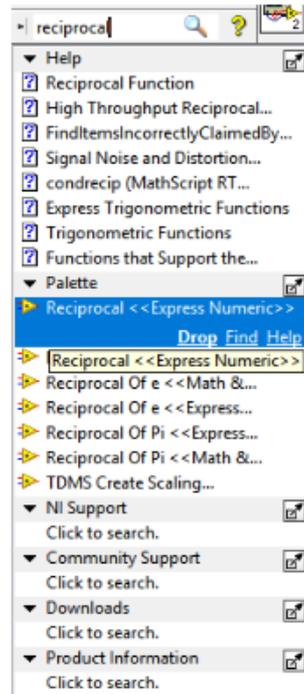


Figura 58. Criando "Multiply"

Fonte: Autoria Própria.

24º Passo – Criando outros Multiply.

25.1 - Escreva, e selecione conforme imagem



25.2 - "Reciprocal" criado



25.3 - Faça as devidas conexões conforme imagem

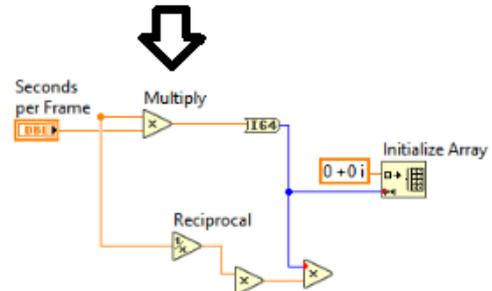


Figura 59. Criando "Reciprocal"

Fonte: Autoria Própria.

25º Passo – Criando "Reciprocal" e fazendo as devidas conexões do circuito.

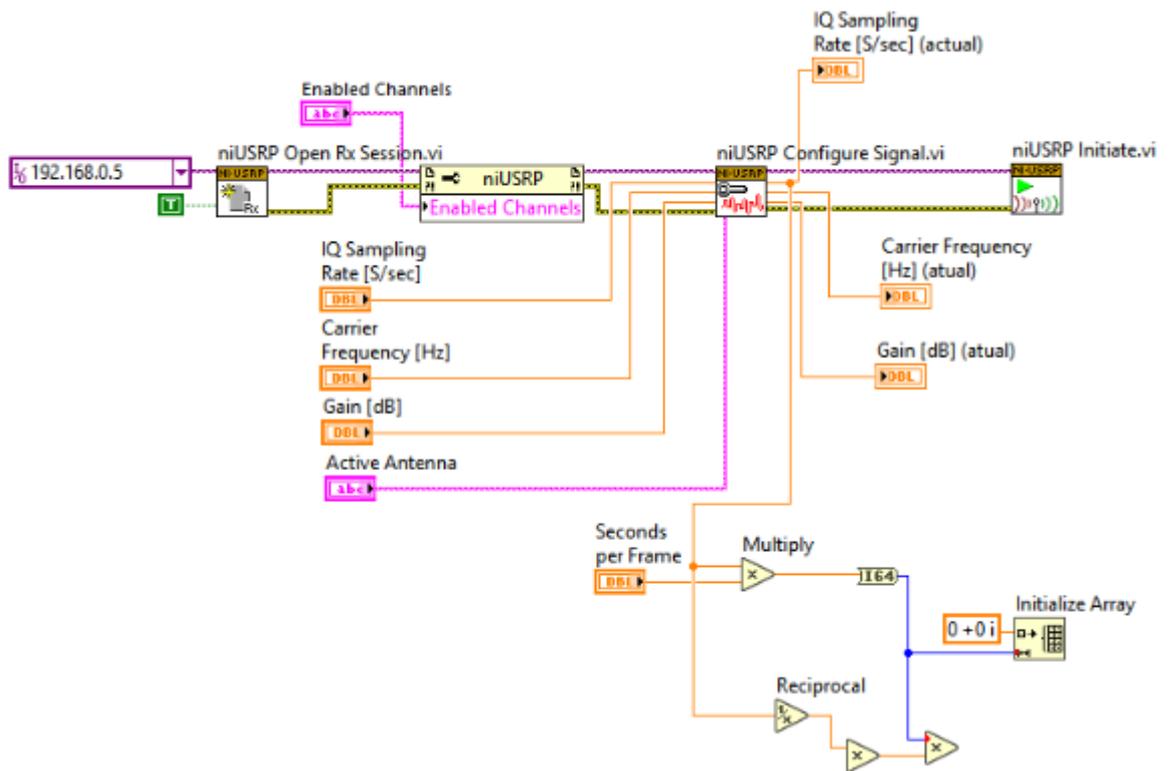
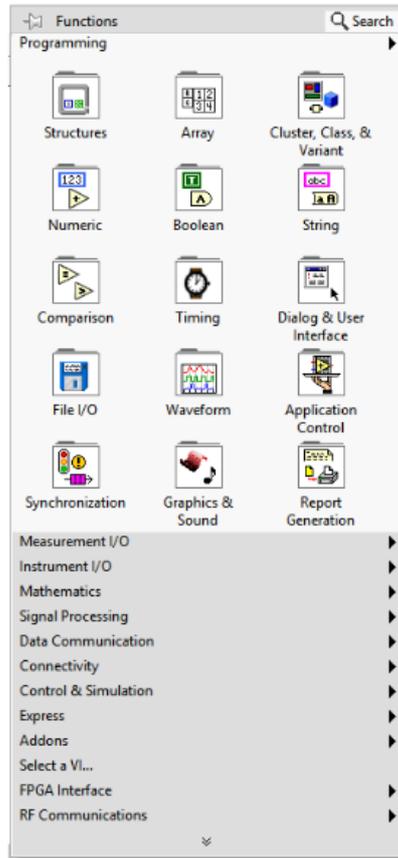


Figura 60. Conectando os elementos do circuito.

26º Passo – Faça as devidas conexões.

Fonte: Autoria Própria.

27.1 - Clique com o botão direito do mouse em qualquer area dentro circuito, exceção no itens criados, e aparecerá esta tela.



27.2 - Selecione opções conforme imagem

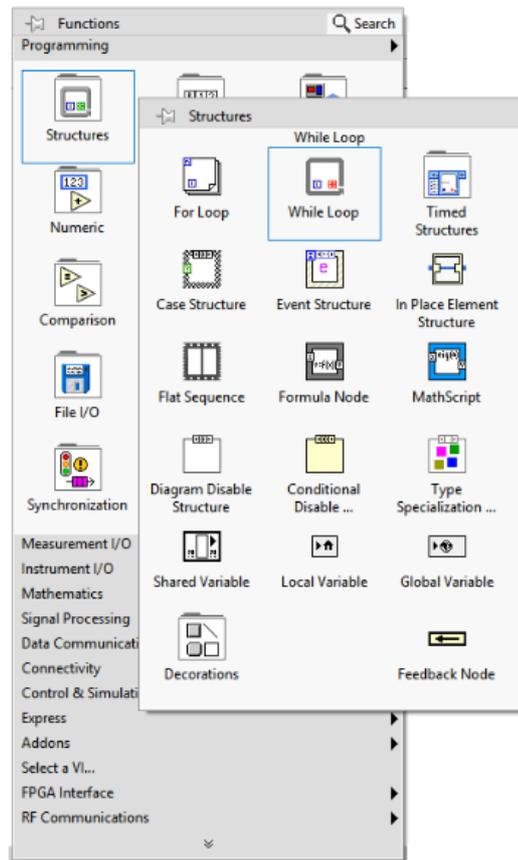


Figura 61. Criando "While Loop" - 1º parte

Fonte: Autoria Própria.



27.3 - Deverá ficar conforme imagem, mas o tamanho é definido conforme desejar.



Figura 62. Criando "While Loop" - 2º parte

Fonte: Autoria Própria.

27.4 - Em seguida, faça a devida ligação.

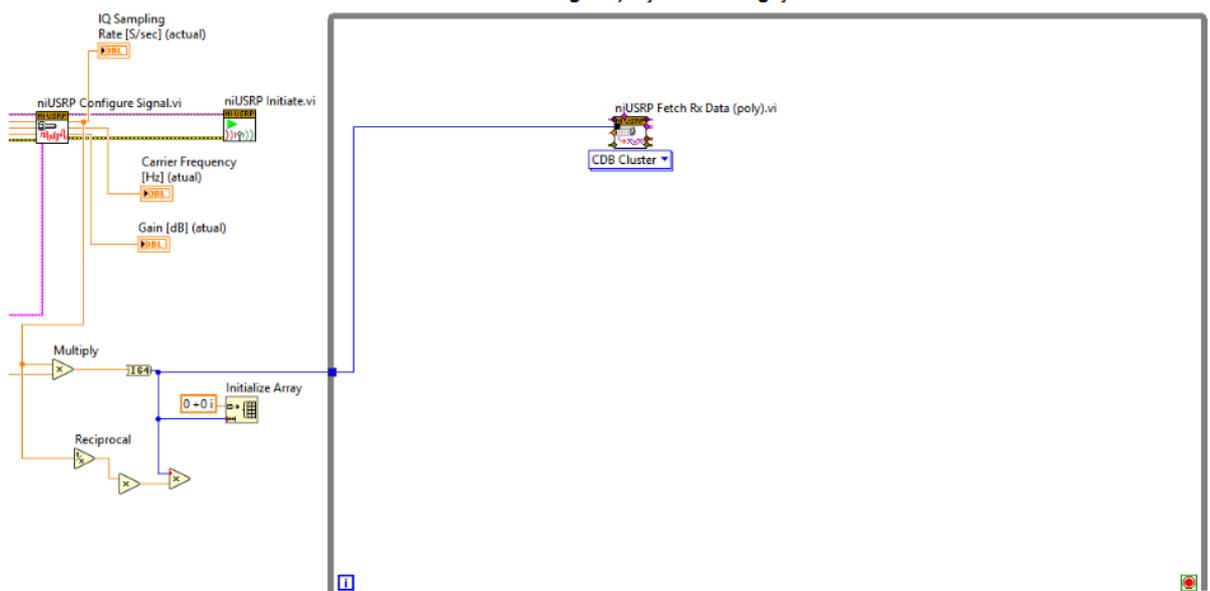


Figura 63. Criando "While Loop" - 3º parte e Conectando os itens do circuito.

Fonte: Autoria Própria.

27º Passo – Criando "While Loop" e fazendo as necessárias ligações.

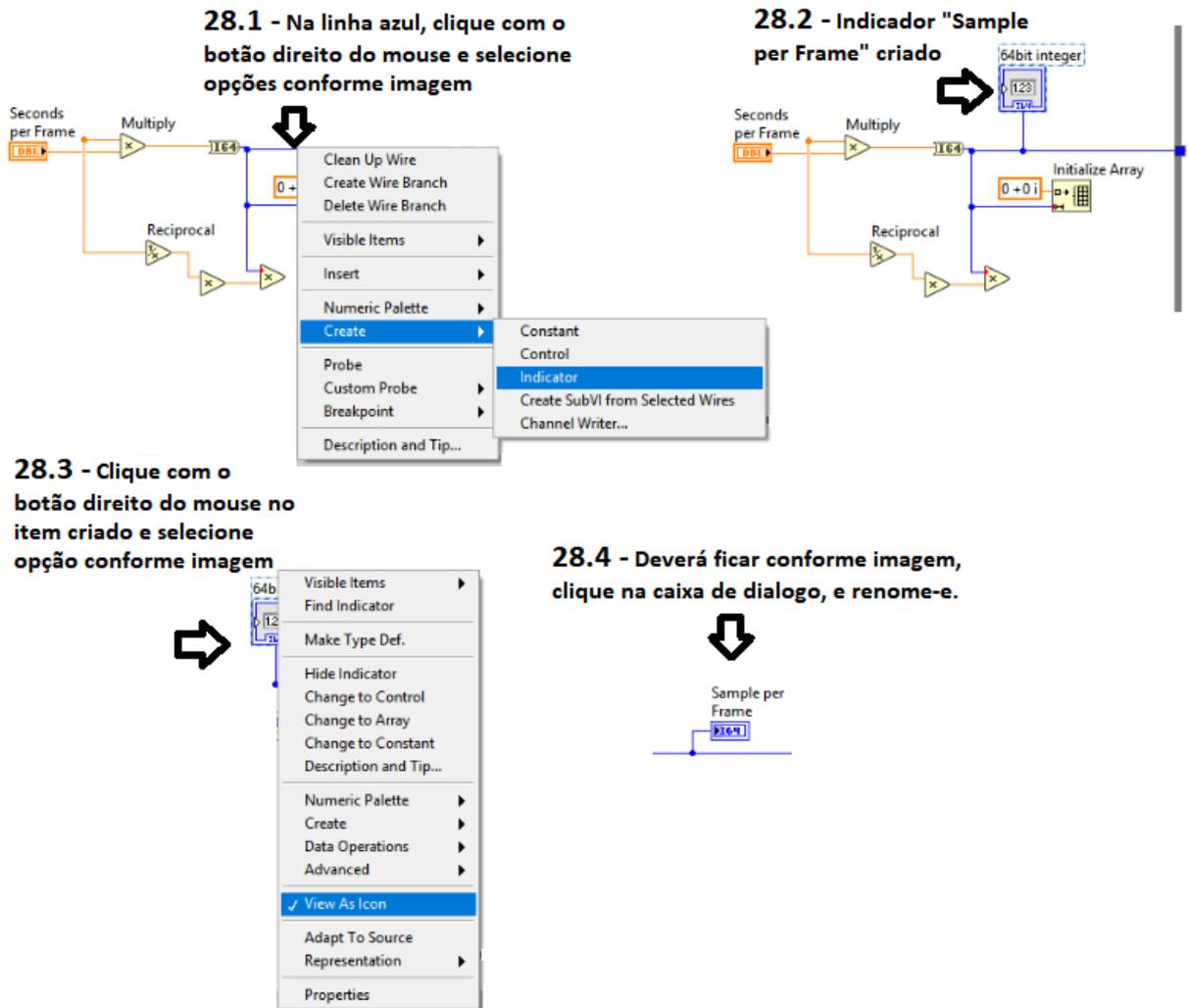
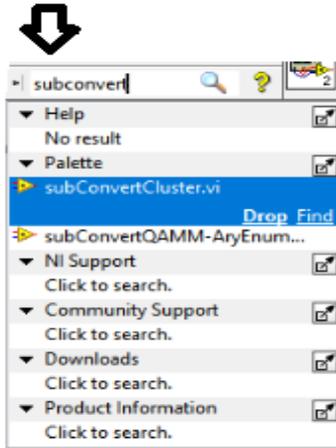


Figura 64. Criando "Sample per Frame"

Fonte: Autoria Própria.

28º Passo – Criando "Sample per Frame".

29.1 - Criando "subConvertCluster.vi", escreva de acordo com imagem.



29.2 - "subConvertCluster.vi" criado e conectado ao circuito.

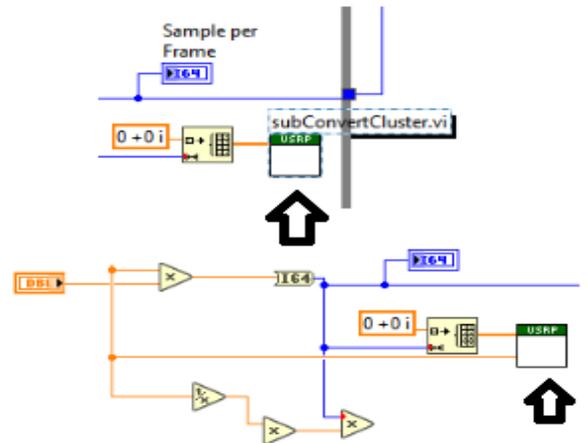
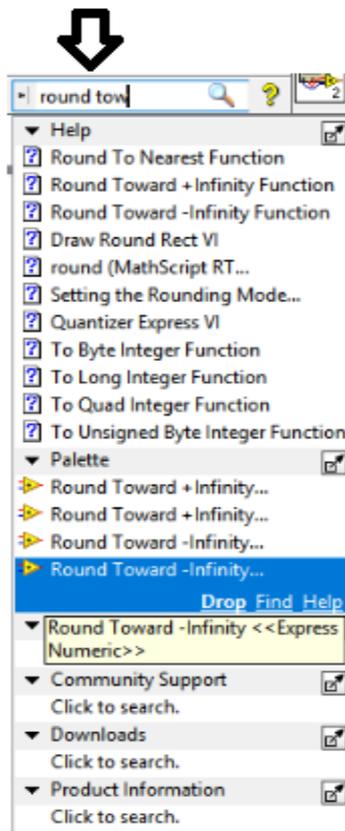


Figura 65. Criando "subConvertCluster.vi"

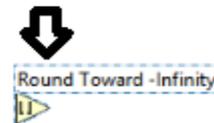
Fonte: Autoria Própria.

29º Passo – Criando "subConvertCluster.vi".

30.1 - Criando "Round Toward - Infinity" escrevendo e selecionando conforme imagem.



30.2 - "Round Toward - Infinity" criado



30.3 - Conecte ao circuito conforme imagem.

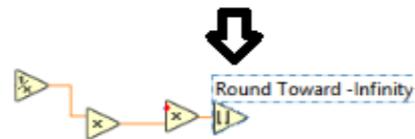
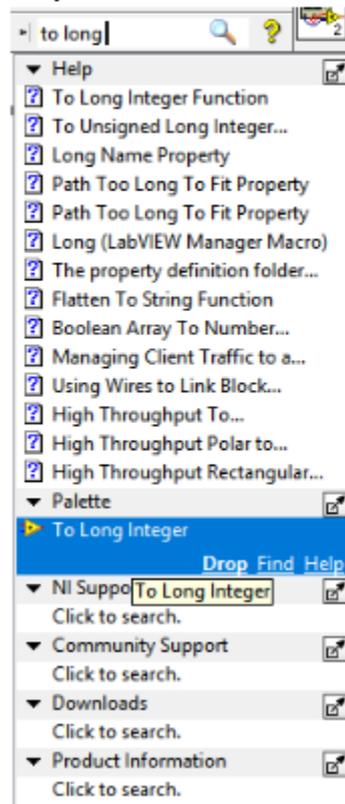


Figura 66. Criando "Round Toward - Infinity"

Fonte: Autoria Própria.

30º Passo – Criando "Round Toward - Infinity".

31.1 - Criando "To Long Integer", escreva e selecione conforme imagem



31.2 - "To Long Integer" criado e conectado ao circuito

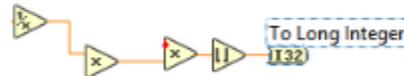
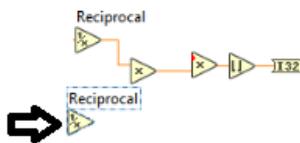


Figura 67. Criando "To long Integer"

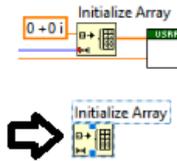
Fonte: Autorial Própria.

31º Passo – Criando "To Long Integer".

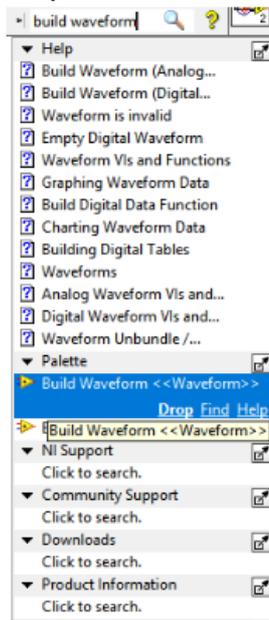
32.1 - Selecione o "Reciprocal" antes criado, e clique Ctrl + C e depois Ctrl + V, novo "Reciprocal" criado



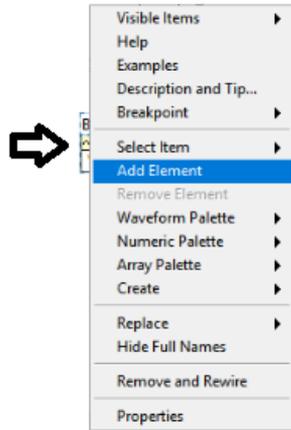
32.2 - Selecione o "Initialize Array" antes criado, e clique Ctrl + C e depois Ctrl + V, novo "Initialize Array" criado.



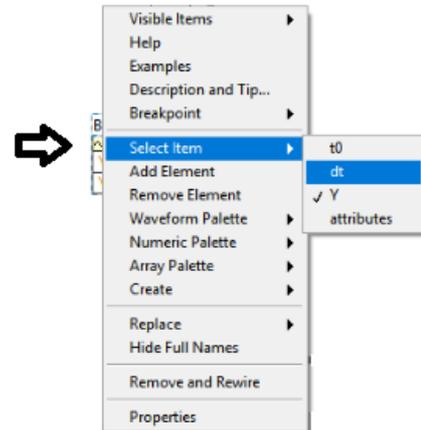
32.3 - Criando "Build Waveform", escrevendo e selecionando conforme imagem.



32.5 - Clique com o botão direito em "Build Waveform" e selecione opção da imagem



32.6 - Em seguida, selecione o primeiro Y, clique com o botão direito do mouse e siga instruções da imagem



32.7 - Deverá ficar conforme imagem, conecte os itens.

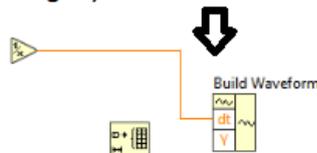
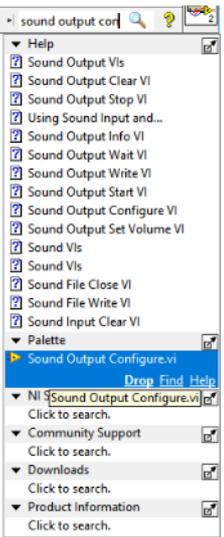


Figura 68. Criando "Reciprocal", "Initialize Array" e "Build Waveform".

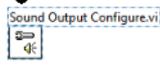
Fonte: Autoria Própria.

32º Passo – Criando itens necessários para o circuito.

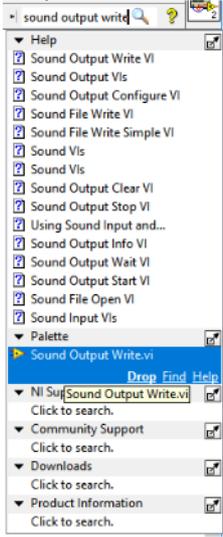
33.1 - Criando "Sound Output Configure.vi", escreva e selecione opção da imagem



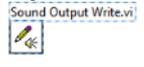
33.2 - "Sound Output Configure.vi" criado



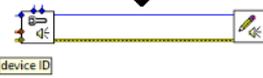
33.3 - Criando "Sound Output Write.vi", escreva e selecione opção da imagem



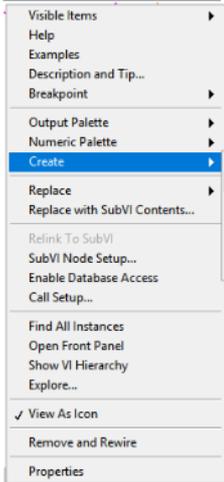
33.4 - "Sound Output Write.vi", criado.



33.5 - Conecte-os, e identifique "device ID", conforme imagem.



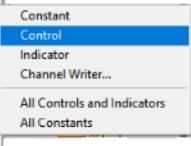
33.6 - Após identificar, conforme item anterior, clique com o botão direito do mouse e selecione opções da imagem



33.7 - "device ID" criado



33.8 - Clique com o botão direito em cima, do "device ID" e selecione conforme imagem



33.9 - Deverá ficar conforme imagem.

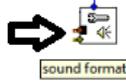


Figura 69. Criando "Sound Output Configure.vi", "Sound Output Write.vi" e "device ID"

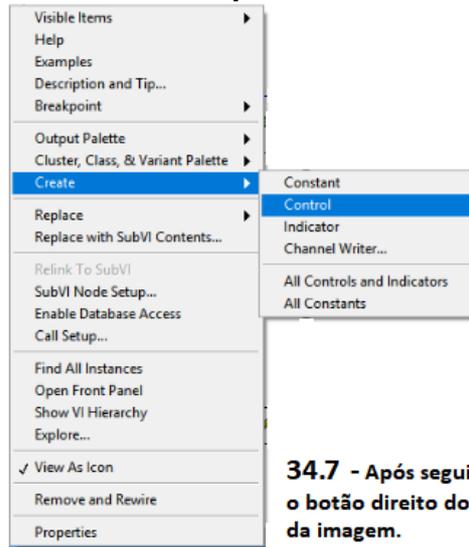
Fonte: Autoria Própria.

33º Passo – Criando itens para o desenvolvimento do circuito.

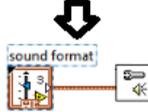
34.1 - Em "Sound Output Configure.vi" com o ícone do mouse pesquise "sound format", conforme imagem.



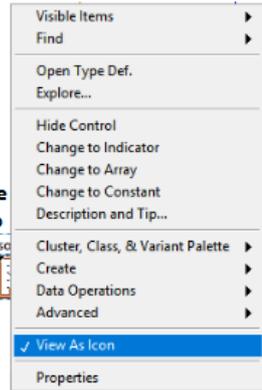
34.2 - Após seguir item anterior, clique com o botão direito do mouse e selecione opções da imagem.



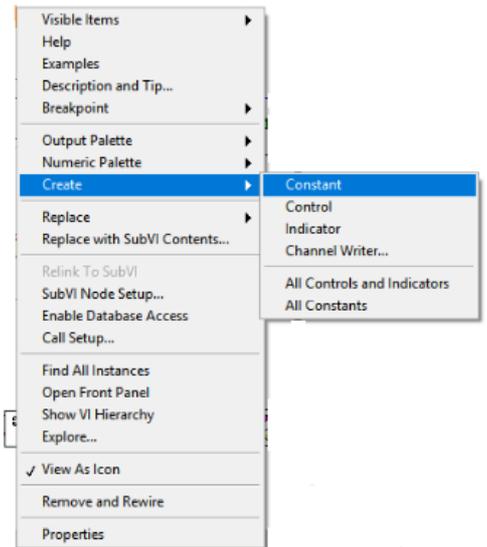
34.3 - "Sound Format" criado



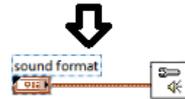
34.4 - Clique com o botão direito do mouse em cima do item criado anteriormente e selecione opção da imagem.



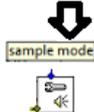
34.7 - Após seguir item anterior, clique com o botão direito do mouse e selecione opções da imagem.



34.5 - "Sound Format" deverá ficar conforme imagem



34.6 - Em "Sound Output Configure.vi" com o ícone do mouse pesquise "Sample Mode", conforme imagem.



34.8 - "Continuous Samples" deverá ficar conforme imagem.

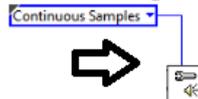
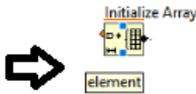


Figura 70. Criando "Sound Format" e "Continuous Samples"

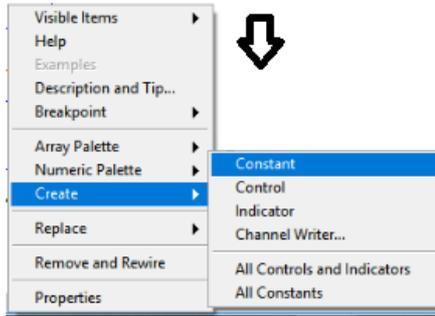
Fonte: Autoria Própria.

34º Passo – Criando "Sound Format" e "Continuous Sample".

35.1 - Em "Initialize Array" com o ícone do mouse pesquise "Element", conforme imagem.



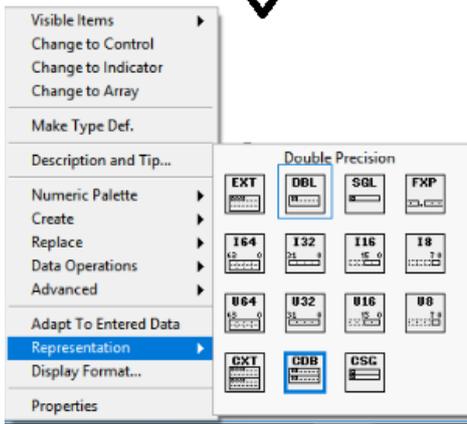
35.2 - Após seguir item anterior, clique com o botão direito do mouse no pontinho "Element" e selecione opções conforme imagem.



35.3 - "Element" criado



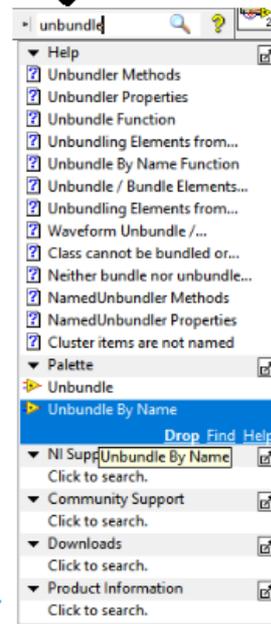
35.4 - Clique com o botão direito do mouse em cima de "Element" e selecione opções conforme imagem.



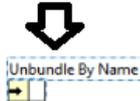
35.5 - "Element" deverá ficar conforme imagem.



35.6 - Criando "Unbundle By Name", escreva e selecione conforme imagem.



35.7 - "Unbundle By Name" criado



35.8 - Conecte os itens do circuito conforme imagem.

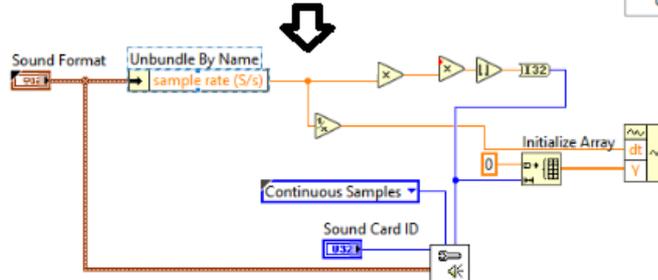
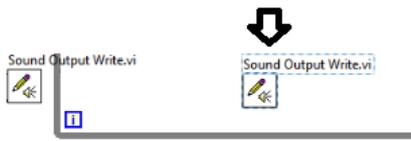


Figura 71. Criando "Element" e "Unbundle By Name"

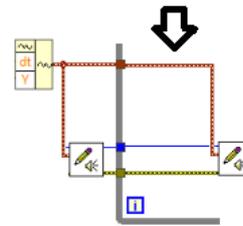
Fonte: Autoria Própria.

35º Passo – Criando "Element", "Unbundle By Name" e conectando os itens, circuito em formação.

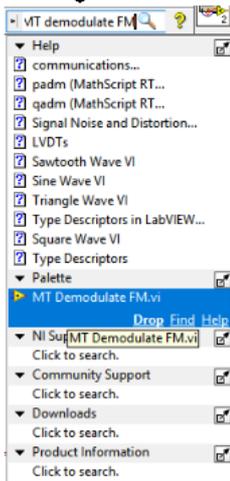
36.1 - Criando um "Sound Output Write.vi", comando Ctrl + C e depois Ctrl + V.



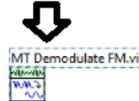
36.2 - Faça as devidas conexões.



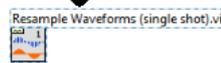
36.3 - Criando "MT Demodulate FM.vi"



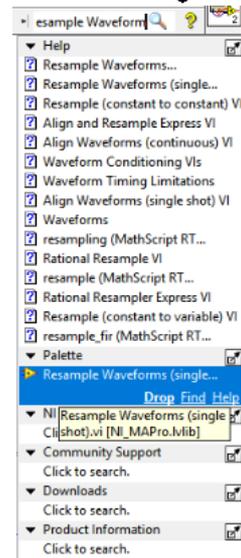
36.4 - "MT Demodulate FM.vi" criado.



36.6 - "Resample Waveforms (single shot).vi" criado.



36.5 - Criando "Resample Waveforms (single shot).vi"



36.7 - Conecte os itens conforme imagem.

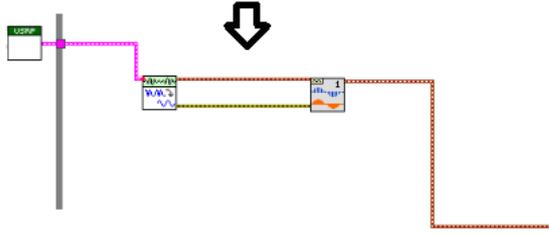
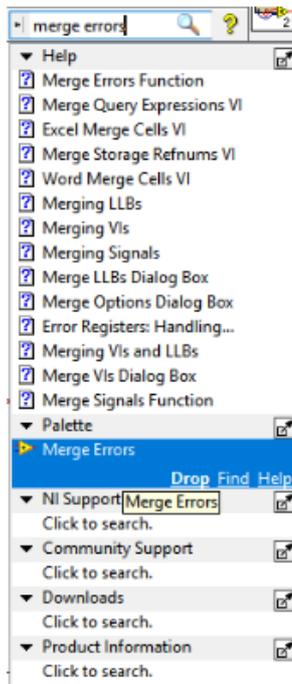


Figura 72. Criando "Sound Output Write.vi", "MT Demodulate FM.vi" e "Resample Waveform (single shot)"

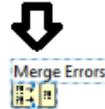
Fonte: Autoria Própria.

36º Passo – Criando e conectando os itens do circuito.

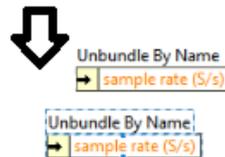
**37.1 - Criando "Merge Errors",
escrevendo e selecionando
conforme imagem.**



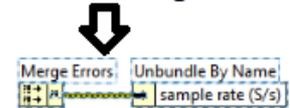
37.2 - "Merge Errors" criado.



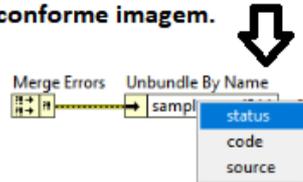
**37.3 - Criando "Unbundle By Name",
Ctrl + C e depois Ctrl + V, novo
"Unbundle By Name" criado.**



**37.4 - Conecte ao circuito
conforme imagem.**



**37.5 - Clique dentro do item
"Unbundle By Name" e selecione
conforme imagem.**



**37.6 - "Unbundle By Name" deverá
ficar conforme imagem.**

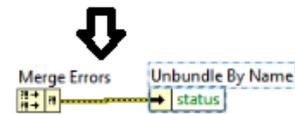
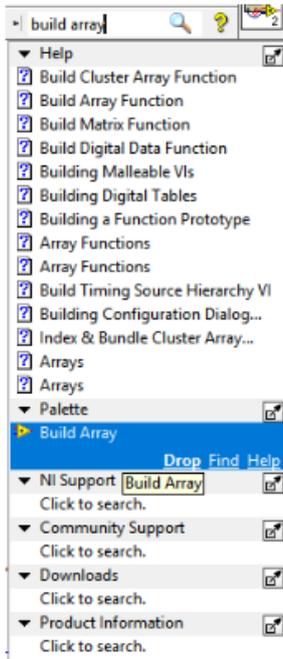


Figura 73. Criando "Merge Errors" e "Unbundle By Name"

Fonte: Autoria Própria.

37º Passo – Criando "Merge Errors" e "Unbundle By Name", conectando-os.

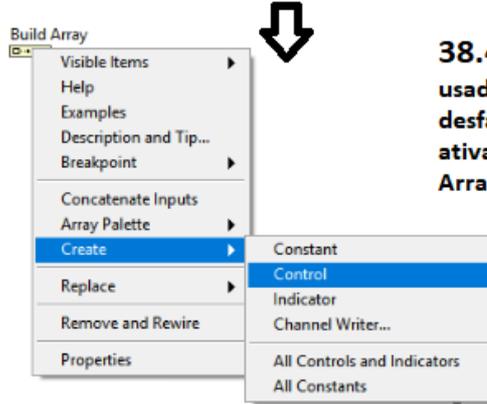
38.1 - Criando "Build Array", escreva e selecione conforme imagem.



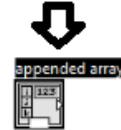
38.2 - "Build Array" criado.



38.3 - Clique com o botão direito do mouse em "Build Array" e siga instruções da imagem. Este procedimento é para ativar uma função chamada "Add Input", caso ainda não estiver ativa.



38.4 - Criou um controle mas não será usado, clique o comando "Ctrl + Z" para desfazer item. Procedimento feito para ativar função de "Add Input" em "Build Array".

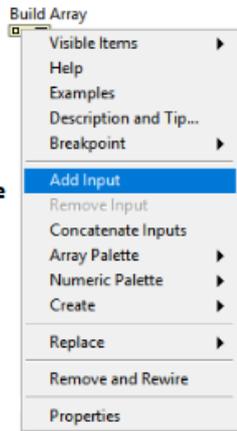


38.6 - "Build Array" deverá ficar conforme imagem.



38.7 - Repita o passo 38.5

38.5 - Clique com o botão direito do mouse em "Build Array" e selecione opção da imagem.



38.8 - "Build Array"

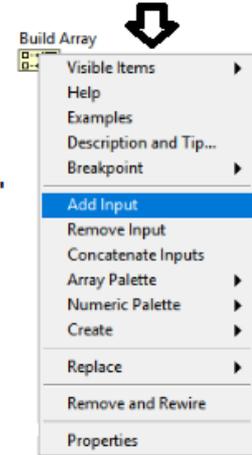


Figura 74. Criando "Build Array" e configurando-o.

Fonte: Autoria Própria

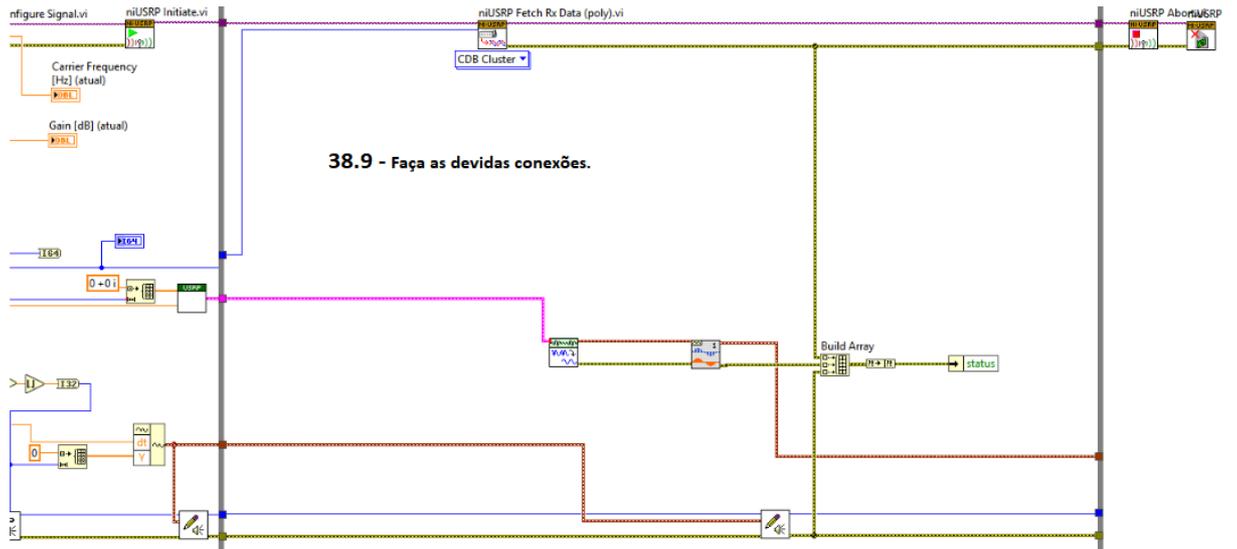


Figura 75. Criando "Build Array" e ativando uma função de adicionar conexão.

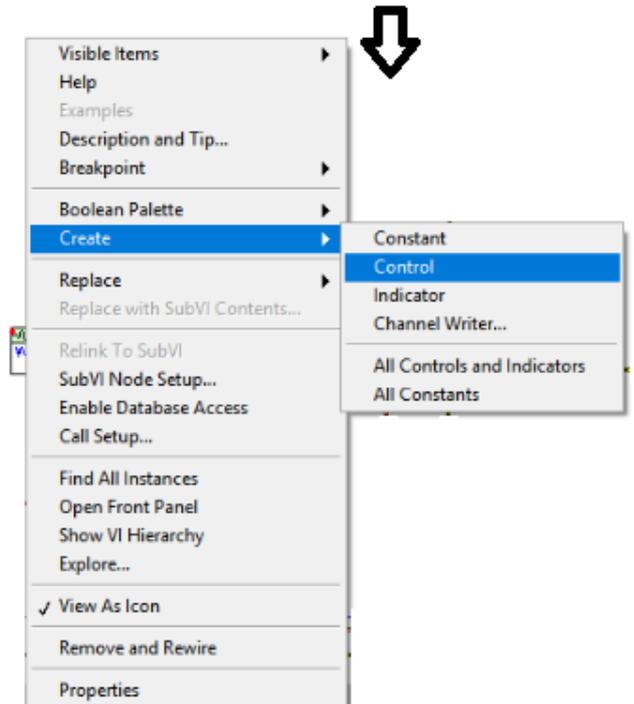
Fonte: Autoria Própria.

38º Passo – Criando "Build Array" e ativando uma função de adicionar mais uma conexão. Fazendo as conexões do circuito.

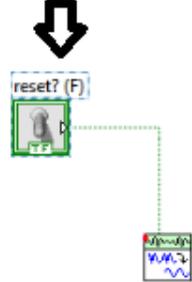
39.1 - Com o icone do mouse, pesquise no VI da imagem, o pontinho do "reset? (F)".



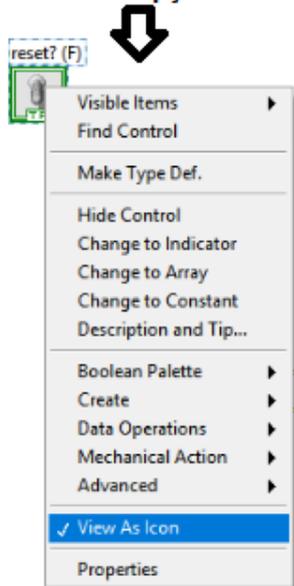
39.2 - Clique com o botão direito do mouse no VI, selecione opções da imagem.



39.3 - "reset? (F) criado.



39.4 - Clique com o botão direito do mouse no item "reset? (F)", selecione opção da imagem



39.5 - "reset? (F), deverá ficar conforme imagem.

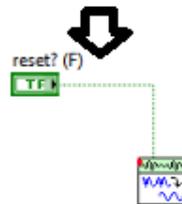
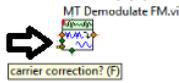


Figura 76. Criando "reset? (F)"

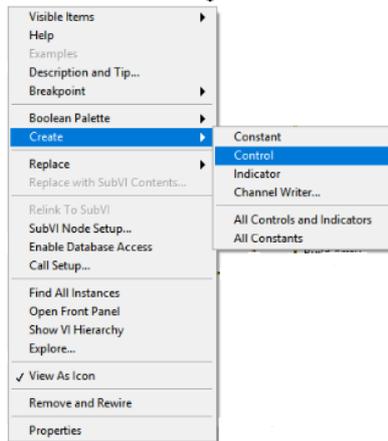
Fonte: Autoria Própria.

39º Passo – Criando "reset? (F)".

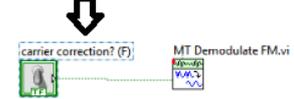
40.1 - No "MT Demodulate FM.vi" com o ícone do mouse pesquise o pontinho "carrier correction? (F)", conforme imagem.



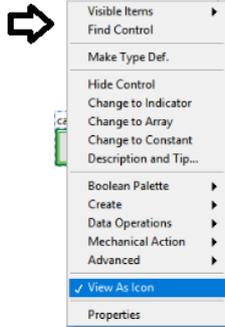
40.2 - Clique com o botão direito do mouse no pontinho do "carrier correction? (F)" e selecione opções da imagem.



40.3 - "carrier correction? (F) criado.



40.4 - Clique com o botão direito do mouse, no "carrier correction? (F)", selecione opção da imagem

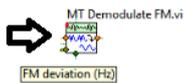


40.5 - "carrier correction? (F)" deverá ficar conforme imagem.

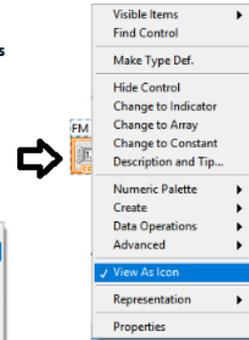
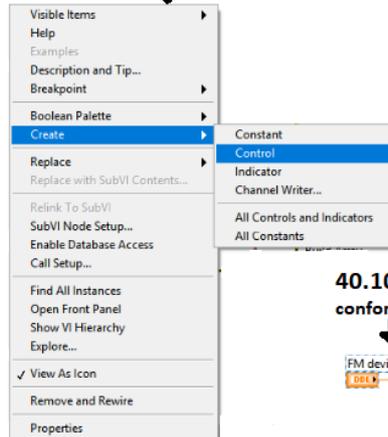


40.9 - Clique com o botão direito do mouse no "FM deviation (Hz)", selecione opção da imagem.

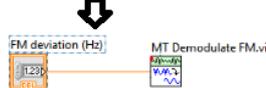
40.6 - No "MT Demodulate FM.vi" com o ícone do mouse pesquise o pontinho "FM deviation (Hz)", conforme imagem.



40.7 - Clique com o botão direito do mouse no pontinho do "FM deviation (Hz)", selecione opções da imagem.



40.8 - "FM deviation (Hz) criado.



40.10 - "FM deviation (Hz)" deverá ficar conforme imagem.

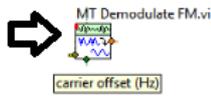


Figura 77. Criando "carrier correction? (F)" e "FM deviation (Hz)"

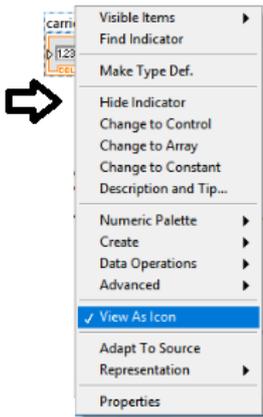
Fonte: Autoria Própria

40º Passo – Criando "carrier correction? (F)" e "FM deviation (Hz)"

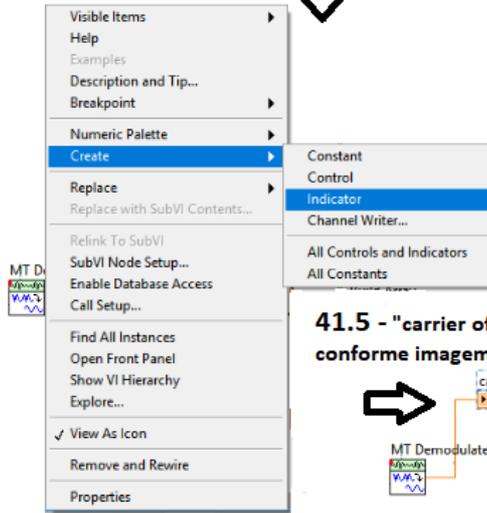
41.1 - Criando "carrier offset (Hz)", com o icone do mouse, selecione o pontinho conforme imagem.



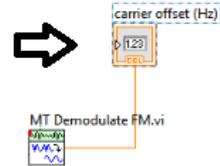
41.4 - Clique com o botão direito do mouse no item "carrier offset (Hz)", e selecione opções da imagem.



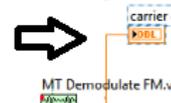
41.2 - Clique com o botão direito do mouse no pontinho encontrado, e selecione opções conforme imagem.



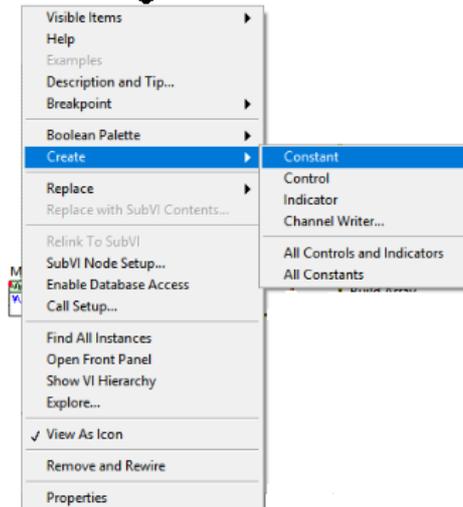
41.3 - "carrier offset (Hz)" criado.



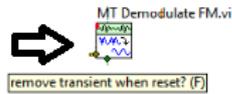
41.5 - "carrier offset (Hz)" deverá ficar conforme imagem.



41.7 - Clique com o botão direito do mouse no pontinho encontrado, e selecione opções conforme imagem.



41.6 - Criando "remove transient when reset? (F)" com o icone do mouse, pesquise o pontinho no .vi, conforme imagem.



41.8 - "remove transient when reset? (F)", criado



Figura 78. Criando "carrier offset (Hz)" e "remove transient when reset? (F)"

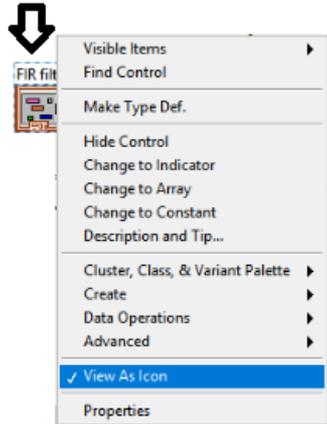
Fonte: Aatoria Própria

41º Passo – Criando "carrier offset (Hz)" e "remove transient when reset? (F)"

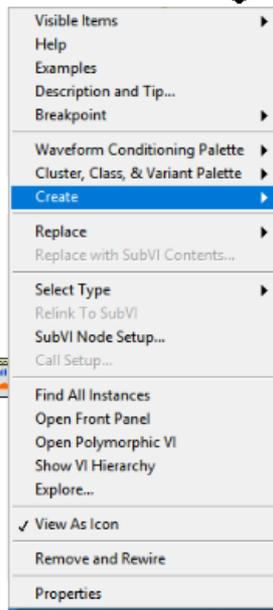
42.1 - Criando "FIR filter specifications", com o icone do mouse pesquise pelo pontinho, conforme imagem.



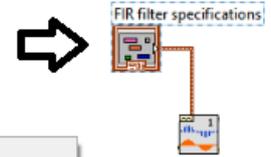
42.4 - Clique com o botão direito do mouse no item "FIR filter specifications", e selecione opções da imagem.



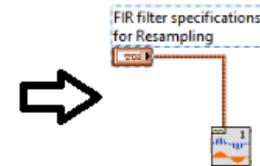
42.2 - Clique com o botão direito do mouse no pontinho encontrado, e selecione opções da imagem.



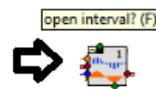
42.3 - "FIR filter specifications" criado.



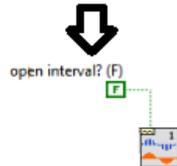
42.5 - "FIR filter specifications", deverá ficar conforme imagem.



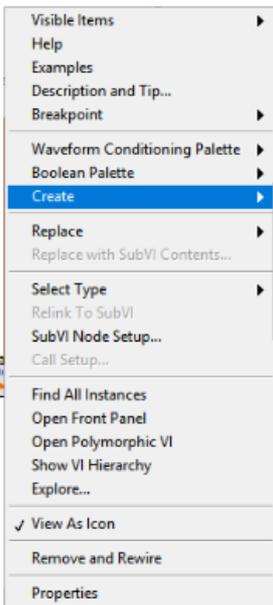
42.6 - Criando "open interval? (F)", com o icone do mouse pesquise pelo pontinho, conforme imagem.



42.8 - "open interval? (F)" criado.



42.7 - Clique com o botão direito do mouse no pontinho encontrado, e selecione opções da imagem.



42.9 - Clique em "open interval? (F)" e deverá ficar conforme imagem.



Figura 79. Criando "FIR filter specifications" e "open interval? (F)"

Fonte: Autoria Própria.

42º Passo – "FIR filter specifications" e "open interval? (F)"

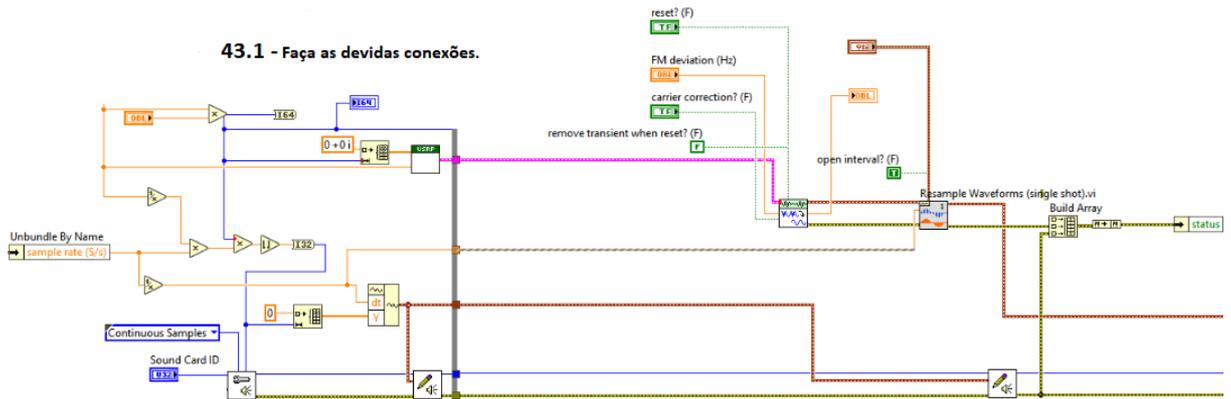
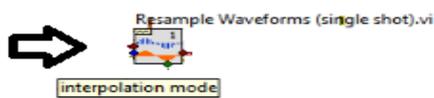


Figura 80. Fazendo conexões do circuito

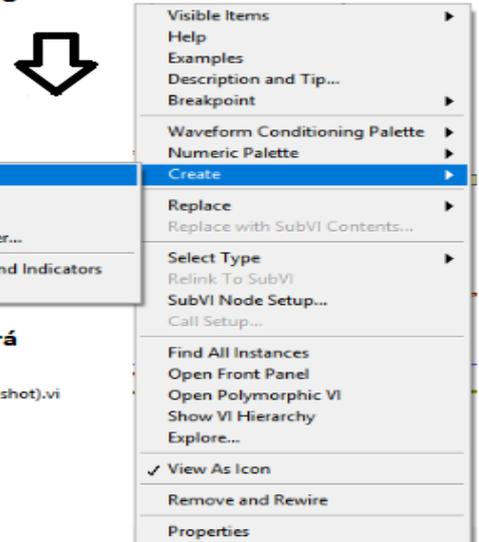
Fonte: Autoria Própria.

43º Passo – Faça as devidas conexões.

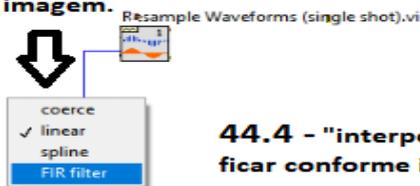
44.1 - Criando "interpolation mode" com o icone do mouse pelo pontinho, conforme imagem.



44.2 - Clique com o botão direito do mouse no pontinho encontrado, e selecione opções da imagem.



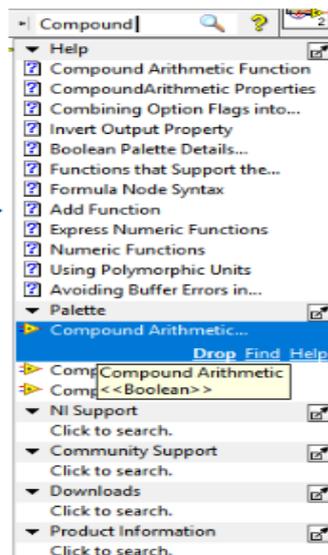
44.3 - "interpolation mode" criado, após clique nele e selecione opção da imagem.



44.4 - "interpolation mode" deverá ficar conforme imagem.



44.5 - Criando "Compound Arithmetic", escrevendo e selecionando conforme imagem.



44.6 - "Compound Arithmetic" criado.

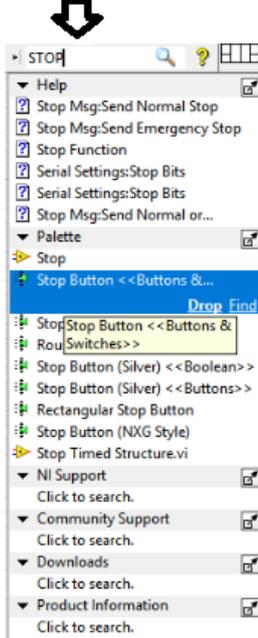


Figura 81. Criando "interpolation mode" e "Compound Arithmetic"

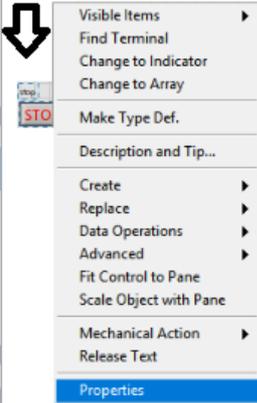
Fonte: Autoria Própria.

44º Passo – Criando "interpolation mode" e "Compound Arithmetic".

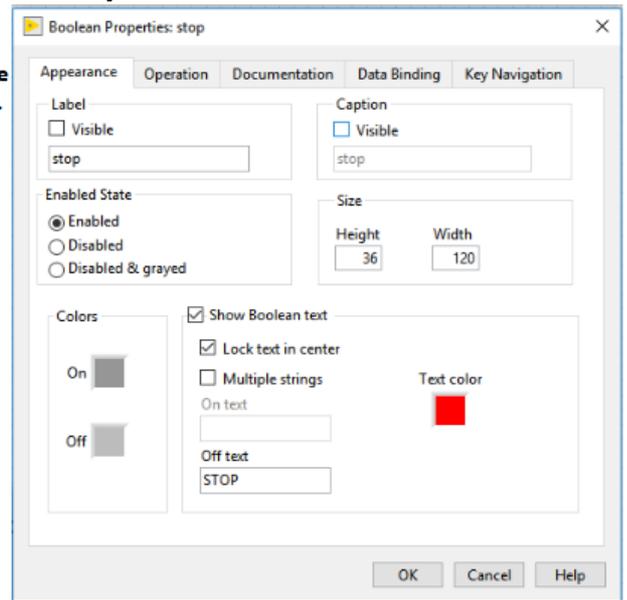
45.1 - No "Front Panel", crie o "STOP", escreva e selecione conforme imagem.



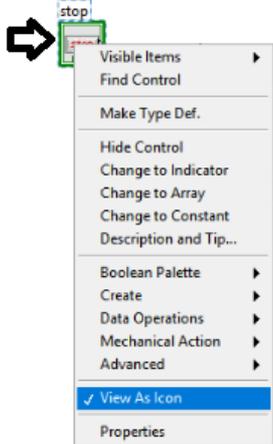
45.2 - "STOP" criado, clique com o botão direito do mouse e selecione opção da imagem.



45.3 - Nesta janela, preencha conforme imagem.



45.4 - No "Block Diagram" clique com o botão direito do mouse no "STOP" e selecione opção da imagem.



45.5 - "STOP" deverá ficar conforme imagem.

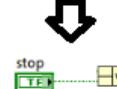
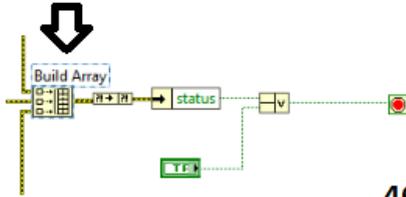


Figura 82. Criando botão "STOP"

45º Passo – Criando botão "STOP".

Fonte: Autoria Própria.

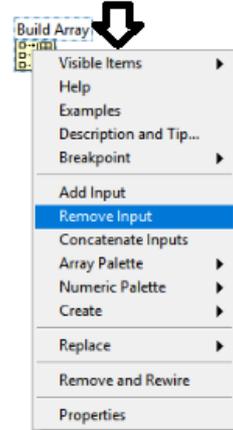
46.1 - Criando outro "Build Array" copiando o já existente no circuito, ou seja, Ctrl + C e depois Ctrl + V.



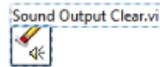
46.2 - "Build Array" criado.



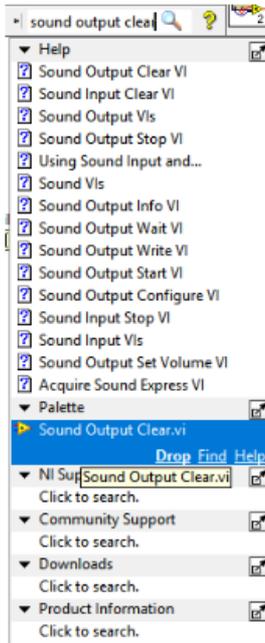
46.3 - Clique com o botão direito do mouse em "Build Array" e selecione opção da imagem.



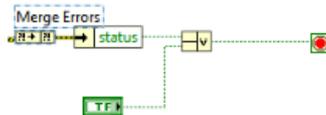
46.5 - "Sound Output Clear.vi" criado.



46.4 - Criando "Sound Output Clear.vi", escreva e selecione opção conforme imagem.



46.6 - Criando outro "Merge Errors", copiando o já existente no circuito, ou seja, Ctrl + C e depois Ctrl + V.



46.7 - "Merge Errors" criado.



46.8 - Faça as conexões conforme imagem. "Build Array" + "Merge Errors".



Figura 83. Criando "Build Array", "Sound Output Clear.vi" e "Merge Errors"

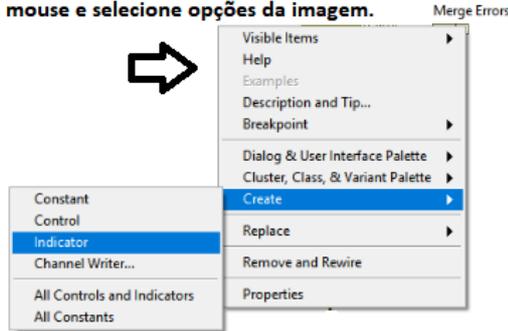
Fonte: Autoria Própria.

46º Passo – Criando "Build Array", "Sound Output Clear.vi" e "Merge Errors".

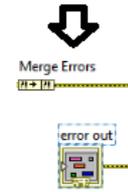
47.1 - Criando "error out", em "Merge Error" com o icone do mouse seleccione o pontinho "error out", conforme imagem.



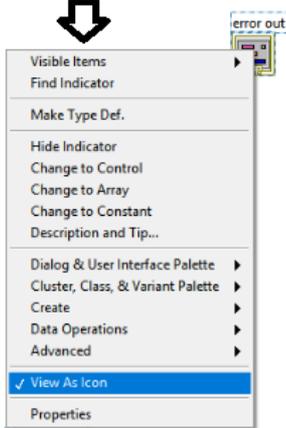
47.2 - Após encontrar o pontinho "error out" em "Merge Errors" clique com o botão direito do mouse e seleccione opções da imagem.



47.3 - "Error out" criado.



47.4 - Clique com o botão direito do mouse no "error out" e seleccione opção conforme imagem.



47.5 - "Error out" deverá ficar conforme imagem.

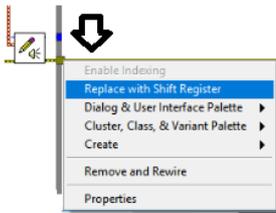


Figura 84. Criando "error out"

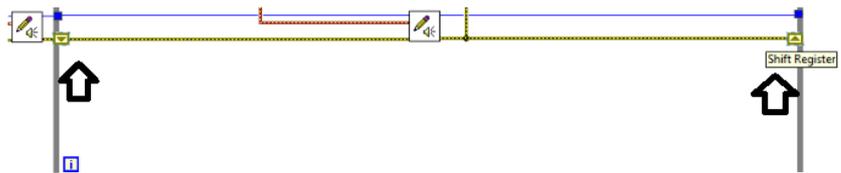
Fonte: Autoria Própria.

47º Passo – Criando "error out" em "Merge Errors".

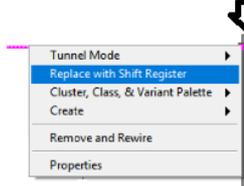
48.1 - Clique com o botão direito do mouse no ponto quadrado amarelo e seleccione opção conforme imagem.



48.2 - Quadrados nas extremas deverá ter esses "triângulos" na linha de conexão amarela.



48.3 - Clique com o botão direito do mouse no ponto quadrado rosa e seleccione opção conforme imagem.



48.4 - Quadrados nas extremas deverá ter esses "triângulos" na linha de conexão rosa.

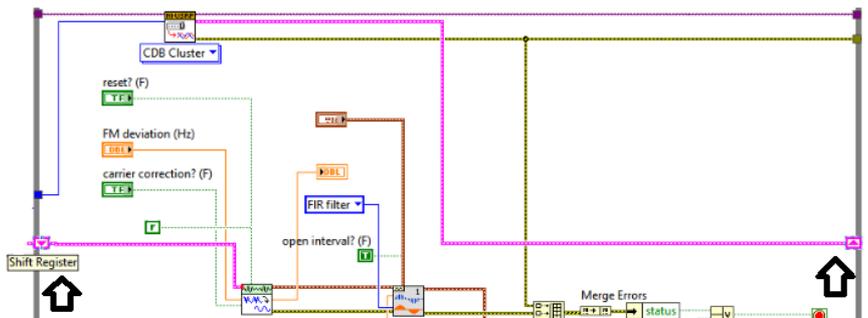


Figura 85. Ativando conexões nas extremidades do "While Loop"

Fonte: Autoria Própria.

48º Passo – Ativando conexões dentro do Loop.

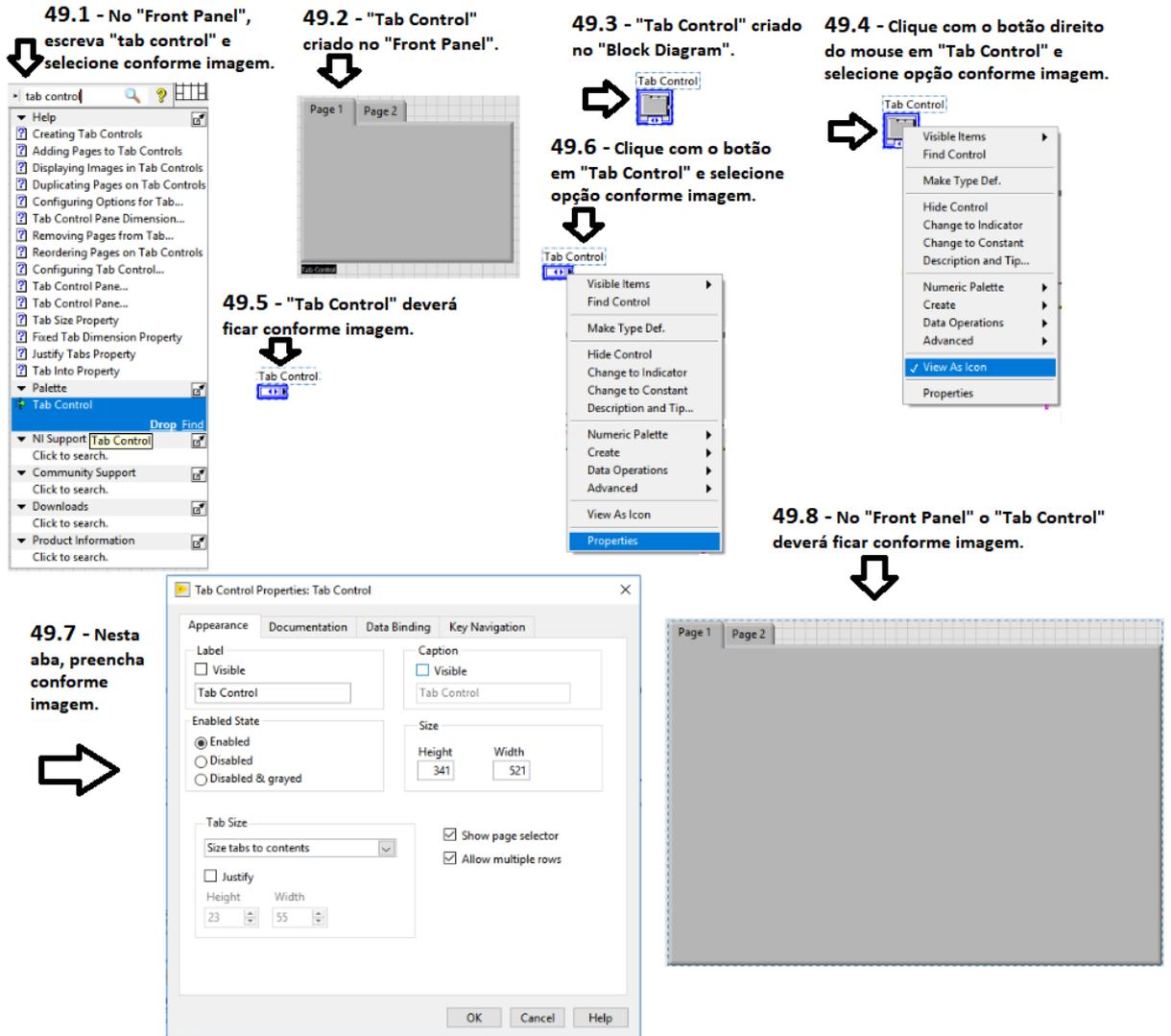
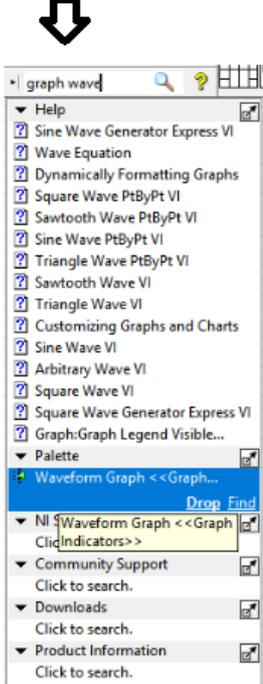


Figura 86. Criando "Tab Control"

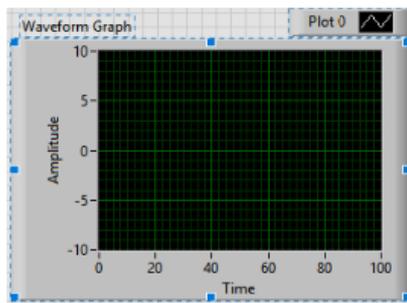
49º Passo – Criando "Tab Control".

Fonte: Autoria Própria.

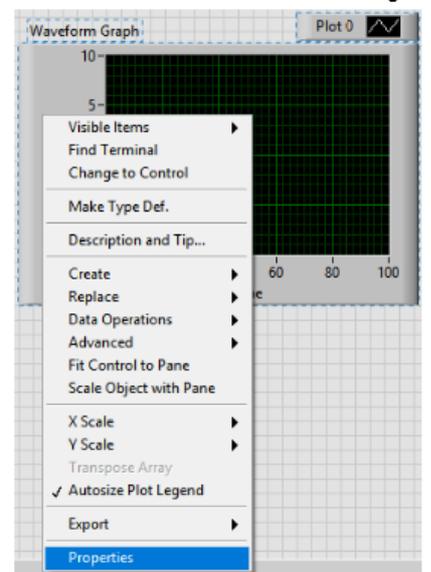
50.1 - Criando "Waveform Graph" no "Front Panel", escreva e selecione conforme imagem.



50.2 - "Waveform Graph" criado.



50.3 - Clique com o botão direito do mouse no "Waveform Graph" e selecione opções da imagem.



50.4 - Nesta aba, preencha conforme imagem.

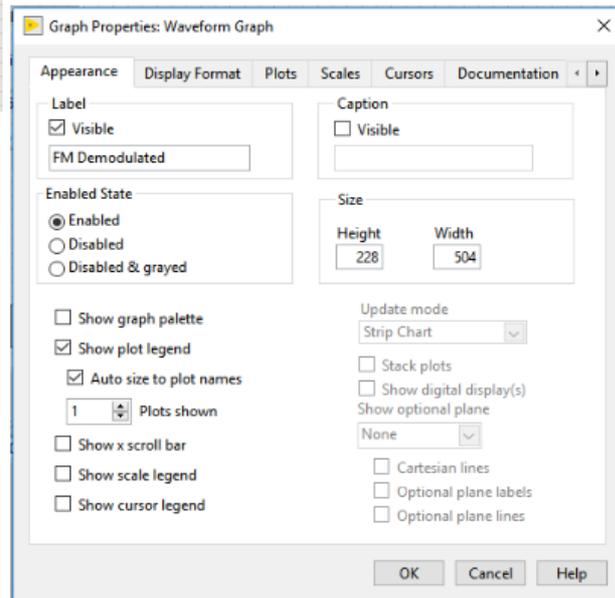
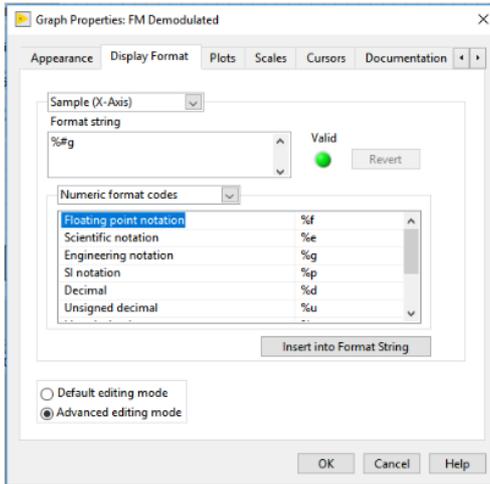


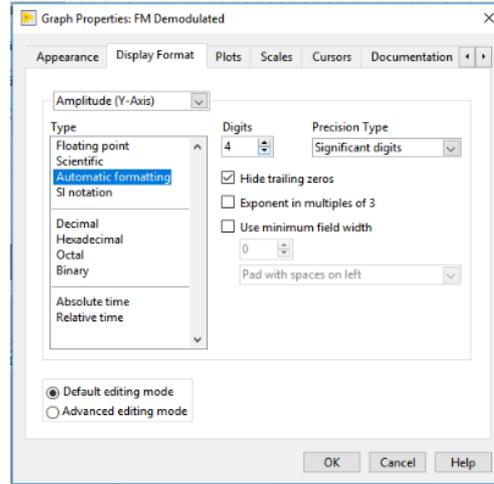
Figura 87. Criando "Waveform Graph"

Fonte: Autoria Própria.

50.5 - Aba "Display Format" preencha o "Sample (X-Axis)" conforme imagem.



50.6 - Aba "Display Format" preencha o "Amplitude (Y-Axis)" conforme imagem.



50.8 - Na aba "Scales" em "Sample (X-Axis)" preencha conforme imagem.



50.7 - Na aba "Plots" preencha conforme imagem.

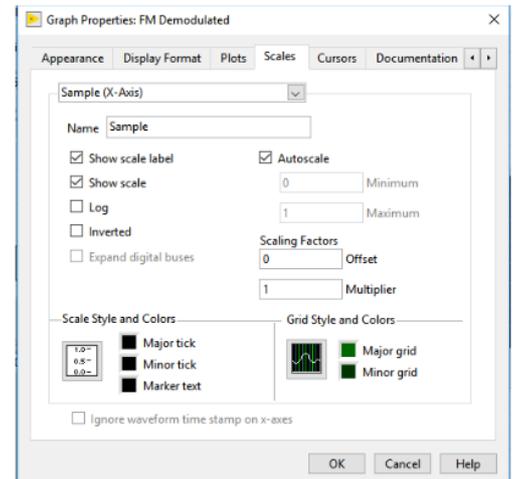
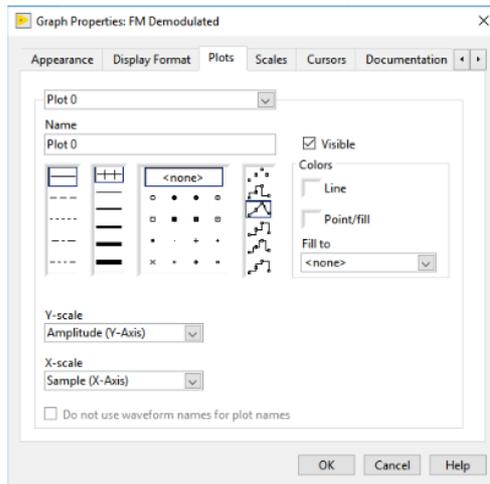
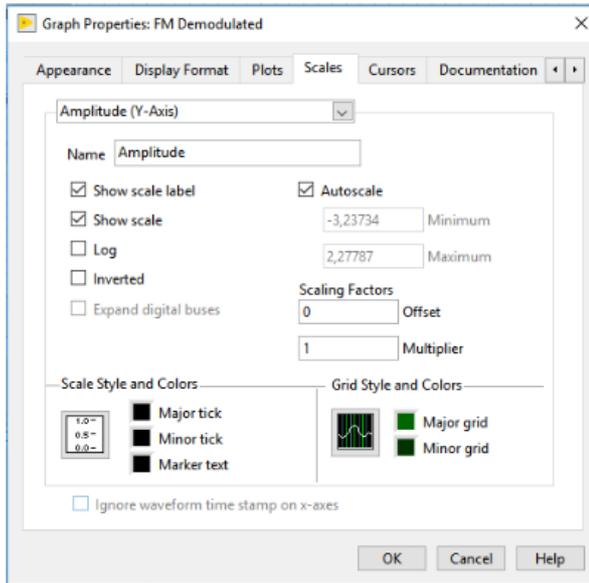


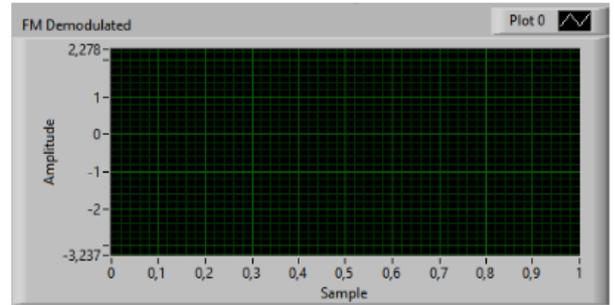
Figura 88. Preenchendo informações do "Waveform Graph"

Fonte: Autoria Própria.

50.9 - Na aba "Scales" em "Amplitude (Y-Axis)" preencha conforme imagem. E clique em "OK" para finalizar.



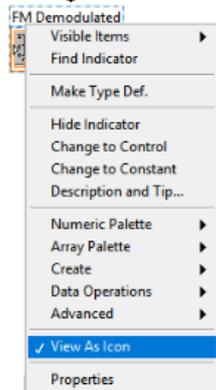
50.10 - "FM Demodulated" criado. No "Front Panel".



50.11 - "FM Demodulate" criado. No "Block Diagram".



50.12 - Clique com o botão direito do mouse em "FM Demodulated" e selecione opção da imagem.



50.13 - Conecte "FM Demodulate" conforme imagem.

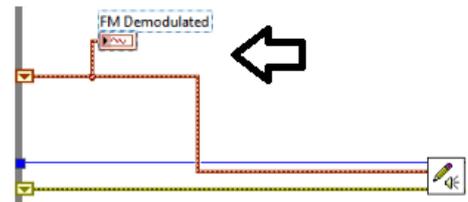


Figura 89. "FM Demodulate" criado

Fonte: Autoria Própria.

50º Passo – Criando "FM Demodulate".

51.1 - No "Front Panel", as configurações deverão ficar conforme imagens.



Figura 90. Demonstração do Projeto Finalizado.

Fonte: Autoria Própria

51.2 - No "Block Diagram", o circuito e as suas devidas conexões e configurações deverão está conforme imagem.

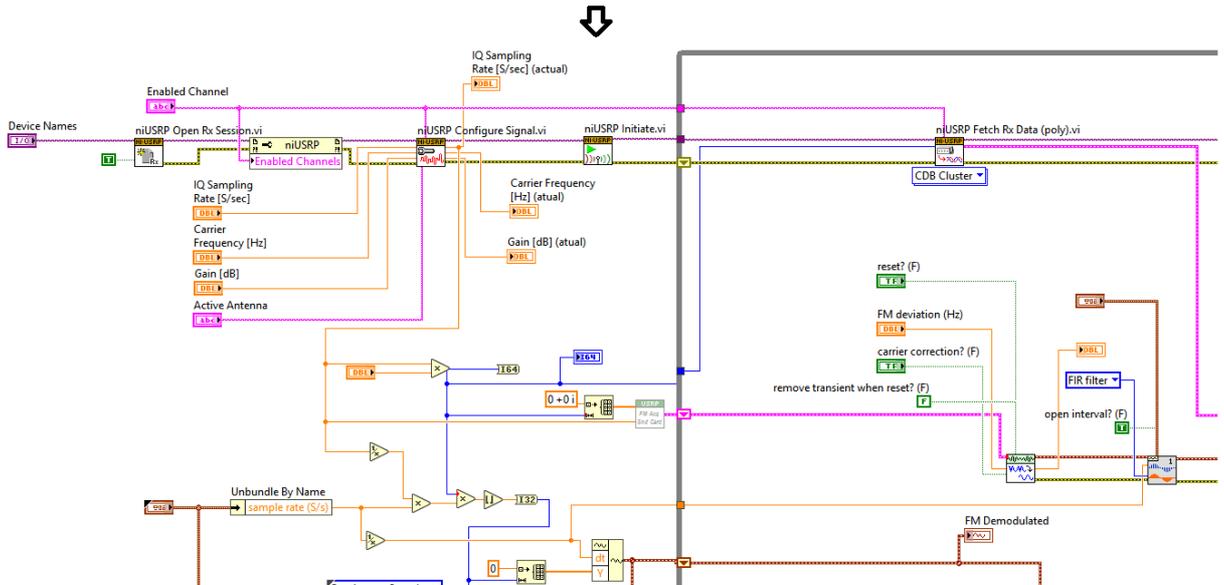


Figura 91. Demonstração do Projeto Finalizado

Fonte: Autoria Própria.

51.3 - No "Block Diagram", o circuito e as suas devidas conexões e configurações deverão está conforme imagem.

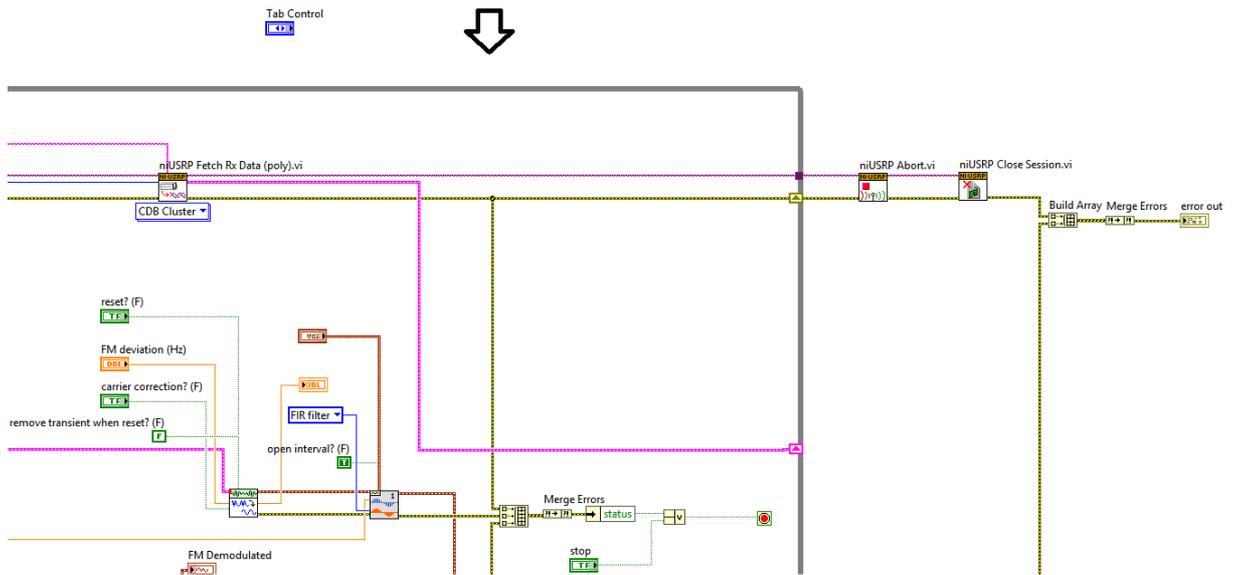


Figura 92. Demonstração do Projeto Finalizado

Fonte: Autoria Própria.

51.4 - No "Block Diagram", o circuito e suas devidas conexões e configurações deverão está conforme imagem.

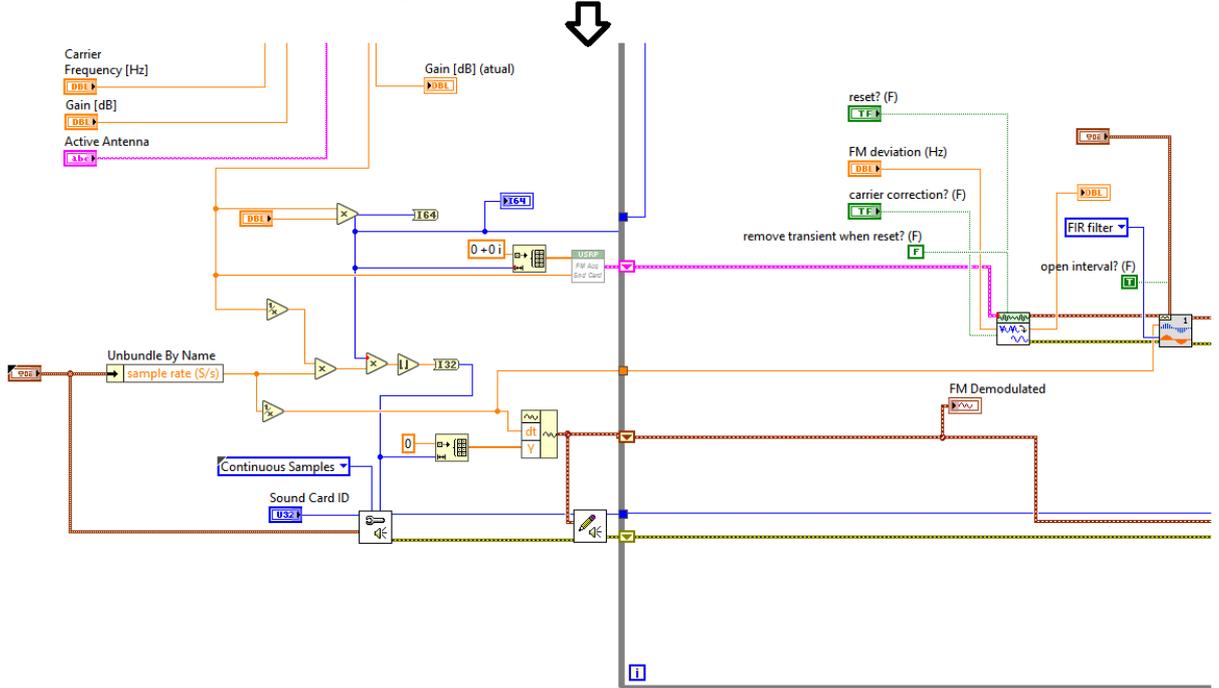


Figura 93. Demonstração do Projeto Finalizado

Fonte: Autoria Própria.

51.5 - No "Block Diagram", o circuito e suas devidas conexões e configurações deverão está conforme imagem.

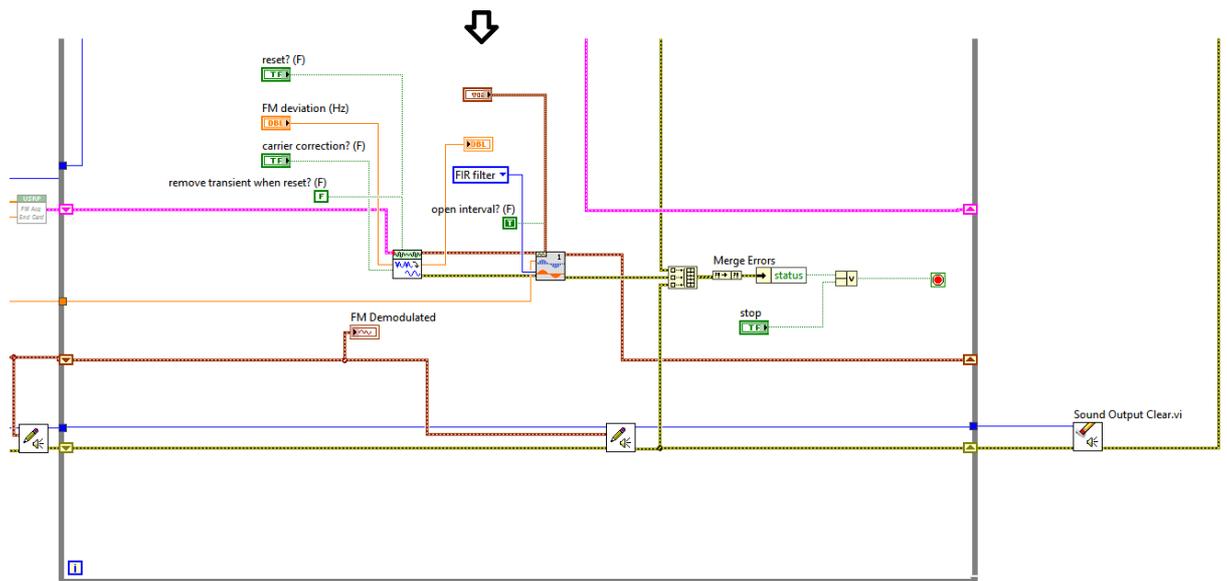


Figura 94. Demonstração do Projeto Finalizado

Fonte: Autoria Própria.

51º Passo – Circuito, Conexões e Gráficos deverão ficar conforme imagens mostradas.

5 ANÁLISE DE RESULTADOS

Considerando o empenho e o esforço aplicado a este trabalho, resultados significativos são alcançados, vale ressaltar que ainda existem muitas melhorias neste objeto de estudo (o rádio), porém o primeiro passo já foi dado, pois em se tratando de rádio a melhoria deve ser contínua, devido as várias constantes que existem, exemplo de variáveis são o clima [em tempos de chuva, sinais de rádio tendem a variar suas componentes (alcance de sinal, e etc.) comparado ao tempo/clima com céu “limpo”], espaço (árvores, prédios, casas), e etc.

Um equipamento como o rádio para um perfeito funcionamento, requer atenção tanto de quem o utiliza, quanto de quem o analisa tecnicamente, essas duas variáveis são essenciais para se obter um equipamento em perfeitas condições, seja para testes ou mesmo algo avançado como uma rádio amadora estudantil, que dependendo da instituição, podendo ter um uso constante.

Os resultados obtidos nos mostram a instalação/conexão dos componentes do kit estudantil NI USRP 2920, e apresenta um exemplo simples e funcional de rádios definidos em software, implementado em *LabVIEW* que é um Rádio FM.

5.1 Instalação/Conexão dos componentes

Observando que para a conexão dos componentes do equipamento estudantil NI USRP 2920, existe uma ordem a seguir (figura 16 até a figura 25) e ao final dela temos a figura 32, podemos observar que apareceu o IP (endereço host) do rádio, indicando que ele está conectado ao PC ou notebook do usuário e quase pronto para uso, ou seja, para receber os comandos do usuário, esse “quase”, significa que o processo indicado pela figura 33 mostra que o rádio estará apto a ser dirigido pelo software. O não seguimento da ordem de conexão dos componentes pode causar problemas futuros ao equipamento ou mesmo ao projeto executado.

5.2 Rádio FM

A partir do passo a passo criado e descrito neste trabalho, verificou-se que tudo que um *hardware* de rádio nos proporciona, foi construído em *software*, objetivamente, é recepção de sinal FM e transmissão de voz a qual a frequência está transmitindo. Descrito e observado nas figuras abaixo.

Na figura abaixo observamos que o Painel Frontal do *LabVIEW* no PC, conectado ao equipamento NI USRP 2920 em simulação, setando a frequência em 96,9 MHz notamos as formas de onda dessa frequência e ouvimos a transmissão de voz desta frequência, informações de modo geral, ou seja, uma música, notícias, ruídos, e etc.

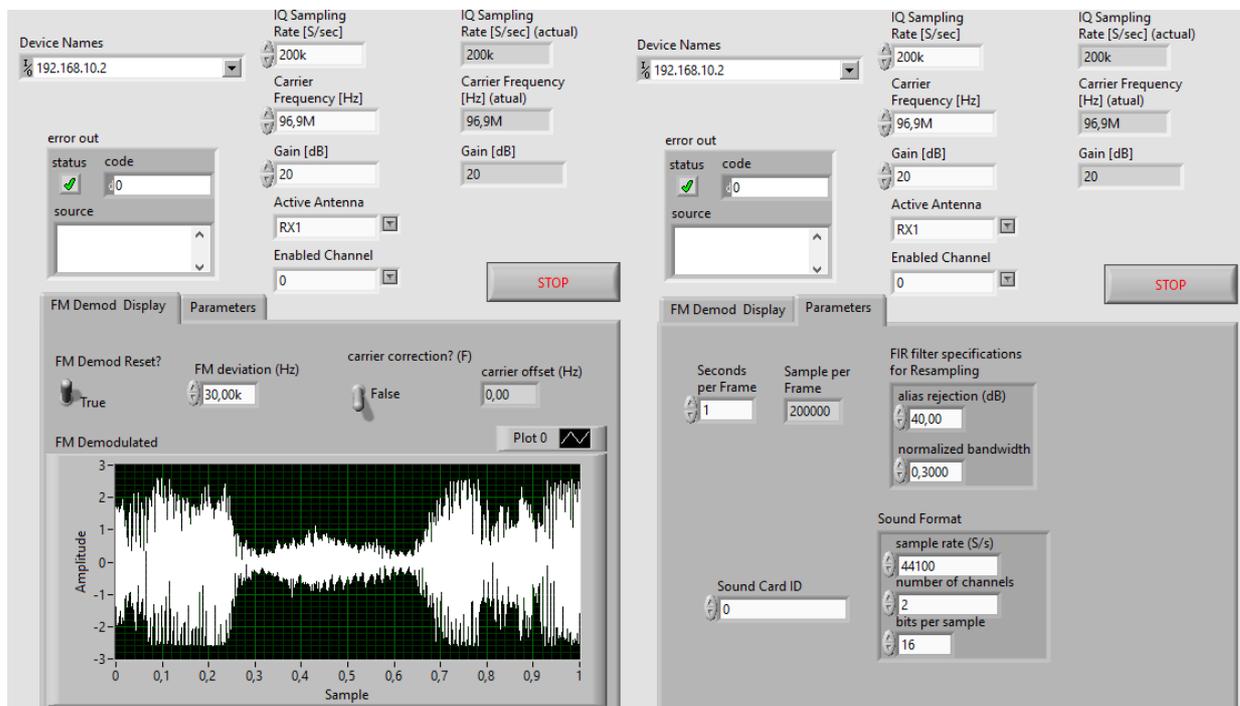


Figura 95- Simulação na Frequência 96,9MHz

Fonte: Autoria Própria.

Posteriormente em simulação, setando a frequência em 95,1 MHz é observado as formas de onda dessa frequência e também é ouvido a transmissão de voz desta frequência, como citado anteriormente, informações de modo geral, ou seja, uma música, notícias, ruídos, e etc.

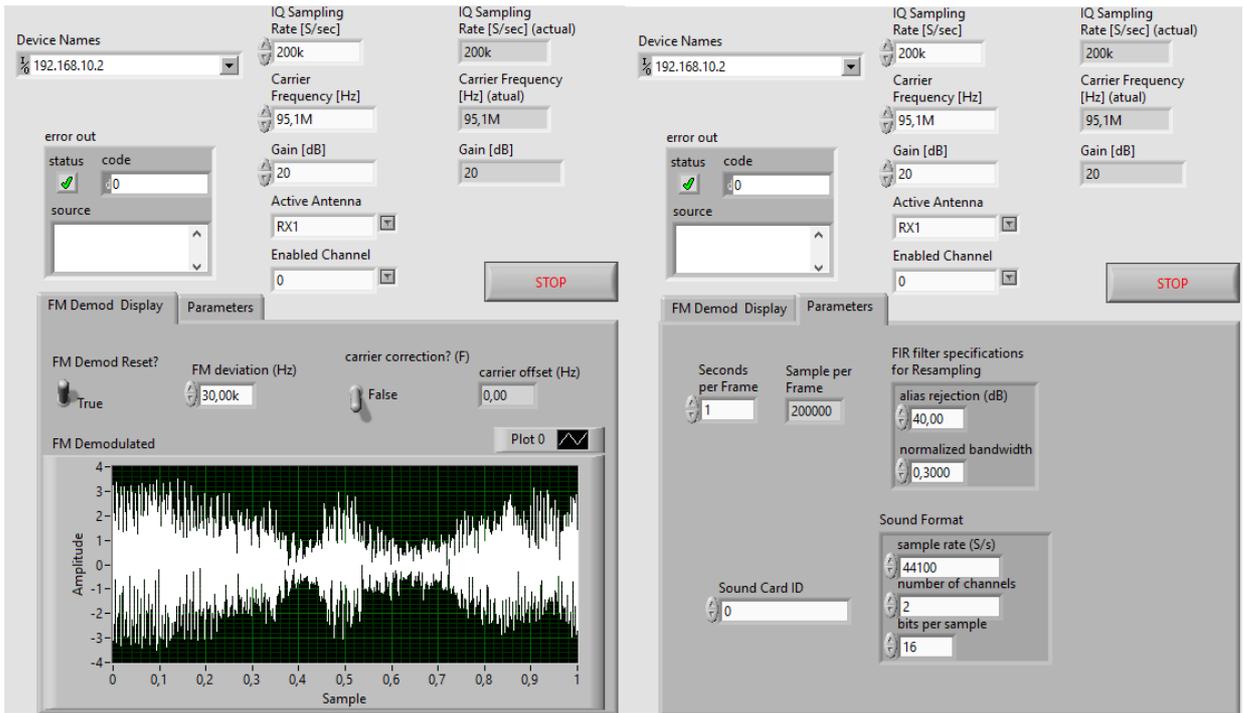


Figura 96 - Simulação na Frequência 95,1MHz

Fonte: Autoria Própria.

E a última simulação, setando a frequência em 100,7 MHz é analisado as formas de onda dessa frequência e também percebemos a transmissão de voz desta frequência, como citado nas duas simulações anteriores, informações de modo geral, ou seja, uma música, notícias, ruídos, e etc.

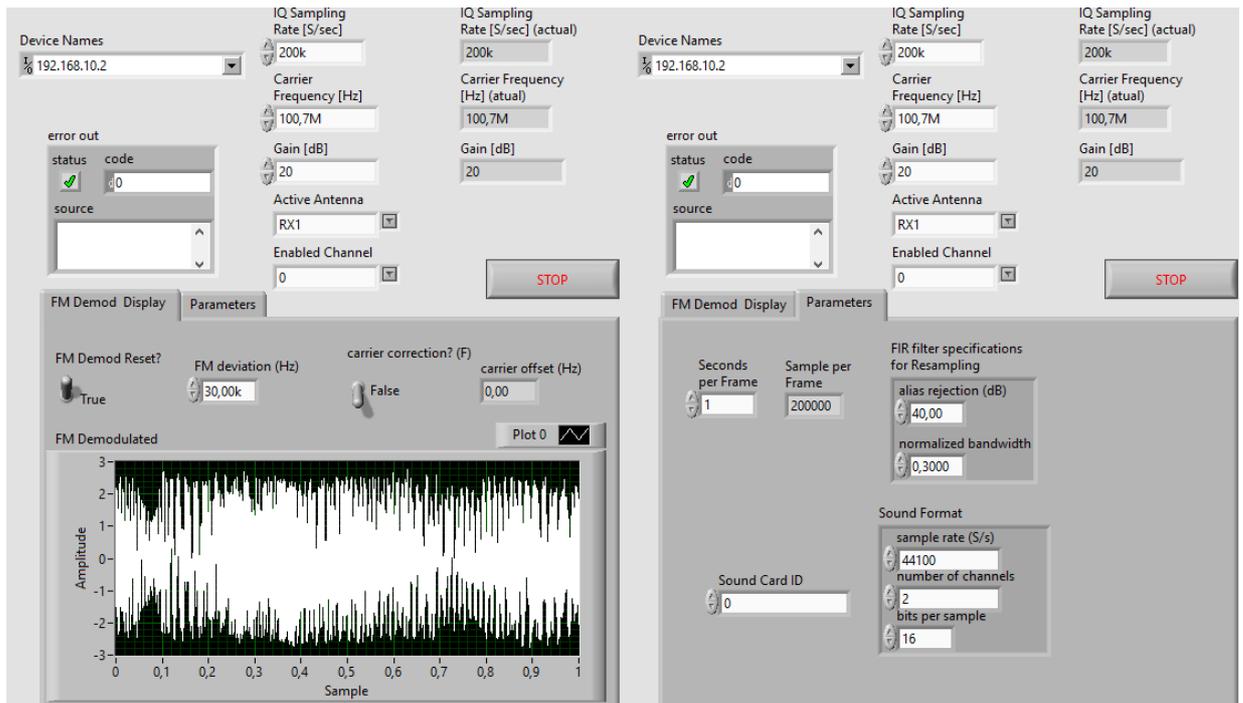


Figura 97 - Simulação na Frequência 100,7MHz

Fonte: Autoria Própria.

Os resultados apresentados neste trabalho atenderam o objetivo principal da pesquisa, que é a criação do passo a passo da utilização do equipamento estudantil NI USRP 2920 em comunicação com o *LabVIEW* proporcionando melhor entendimento de telecomunicações e Rádio Definido por *Software*. Um atento a observar, a antena conectada ao rádio possui um ganho muito baixo na unidade IFAM CMDI, e foi posto a mão na antena, segurando-a, aumentando a permeabilidade e conseqüente o ganho, e assim diminuindo o ruído na transmissão do sinal de voz.

6 CONSIDERAÇÕES FINAIS

A exploração ao grande potencial dos protótipos de rádios definidos por *software* está apenas começando, mas já há grande interesse entre pesquisadores e empresas da área, tanto que algumas plataformas comerciais já foram desenvolvidas, e diversos produtos incorporam SDR mesmo que parcialmente.

O rádio definido por *software* é uma área de pesquisa extremamente ampla e que está em constante mudança e desenvolvimento, sendo assim, seria impossível abordar todos os tópicos relativos a esta tecnologia. Essa não era a ideia do trabalho. O pensamento deste trabalho era mostrar, analisar os módulos da arquitetura de um rádio definido por *software* de maneira didática, de modo que pessoas leigas no assunto conseguissem aprender e entender pelos menos o básico de uma tecnologia tão ampla e usual nos dias atuais.

Neste trabalho foi apresentado uma visão dos aspectos teóricos e práticos, no desenvolvimento de pesquisa em SDR. A parte teórica, apresentamos o uso de SDR, porém não existe um conceito formado para esta tecnologia, foram usados vários autores citados e analisados de uma plataforma popular na pesquisa em telecomunicações e os fundamentos de transmissão digital. Dentro deste mesmo capítulo uma amostra geral da arquitetura de um SDR real foi abordada. Seguindo esta arquitetura de transceptor real do SDR. Na parte prática, focamos na plataforma *LabVIEW*. Mostrando como utilizar o *LabVIEW*, seus módulos e ferramentas. Também demonstramos o passo a passo da criação de um Rádio FM.

Os Rádios definidos por *software*, considera-se que existe um campo amplo para o desenvolvimento de novos projetos de pesquisa, seja como ferramenta para o desenvolvimento de novos padrões e protocolos de comunicação, seja como alvo de estudos sobre novas soluções de implementação. Certamente, os trabalhos mais interessantes na área ainda virão. Podendo ser trabalhado nas tecnologias emergentes comunicações móveis, TV digital, VoIP, e etc.

REFERÊNCIAS BIBLIOGRÁFICAS

GERHARDT, T.E.; SILVEIRA, D.T. **Métodos de Pesquisa**. 2009.

JUNIOR, V. A. S. **Construção de kits didáticos e prototipagem de soluções usando GNU Radio e USRP**. 2015.

MUTTONI, L. **Desenvolvimento de um Receptor ISDB-T 1-Seg através de Rádio Definido por Software**. UFU COPEL. 2017.

NI. National Instruments. **An Introduction To Software Defined Radio: With NI LabVIEW and NI USRP**. 2013.

NI. National Instruments. **Getting Started Guide Ni USRP and LabVIEW Communications System Design Suite Universal Software Radio Peripheral**. 2014.

NI. National Instruments. **Getting Started Guide NI USRP-2920/2921/2922 Universal Software Radio Peripheral**. 2015.

NI. National Instruments. **USRP Software Defined Radio Device Manual: USRP-2920 Block Diagram**. Disponível em: <http://www.ni.com/documentation/en/usrp-software-defined-radio-device/latest/usrp-2920/block-diagram/>. 2017.

OROFINO, G. B. A. **Implementação de um analisador de espectro utilizando o conceito rádio definido por software (SDR)**. Departamento de Engenharia de Telecomunicações. 2017.

JANSON, J. **Radio Definido por Software: Estudo e Realização de Teste com uma Plataforma Livre**. Centro Federal de Educação Tecnológica de Santa Catarina. 2012.

MACHADO, E. **Implementação de Sistema de Comunicação ADS-B por meio de dispositivos de Rádio Software**. Universidade Federal do Rio Grande do Sul. 2018.

LANGENAKEN, F. **A Rapid Graphical Programming Approach to SDR Design and Prototyping with LabVIEW and the USRP**. 2011.