

UNIVERSIDADE FEDERAL DO PARANÁ

HELDER ESCOBAR SALES

ACESSO REMOTO DE DISPOSITIVOS DE RÁDIO DEFINIDO POR SOFTWARE PARA
USO EDUCACIONAL

CURITIBA

2023

HELDER ESCOBAR SALES

ACESSO REMOTO DE DISPOSITIVOS DE RÁDIO DEFINIDO POR SOFTWARE PARA
USO EDUCACIONAL

Dissertação apresentada ao curso de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Ph.D. André Augusto Mariano
Coorientador: Prof. Dr. Evelio Martín García Fernández

CURITIBA

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA CIÊNCIA E TECNOLOGIA

Sales, Helder Escobar.

Acesso remoto de dispositivos de rádio definido por software para uso educacional. / Helder Escobar Sales. – Curitiba, 2023.

1 recurso on-line : PDF.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Tecnologia Programa de Pós-Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. André Augusto Mariano.

Coorientador: Prof. Dr. Evelio Martín García Fernández.

1. Sistemas de telecomunicação. 2. Rádio. 3. Radiodifusão digital. 4. Software - Educação. I. Mariano, André Augusto. II. García Fernández, Evelio Martín. III. Universidade Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

Bibliotecário: Nilson Carlos Vieira Júnior CRB-9/1797

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **HELDER ESCOBAR SALES** intitulada: **ACESSO REMOTO DE DISPOSITIVOS DE RÁDIO DEFINIDO POR SOFTWARE PARA USO EDUCACIONAL**, sob orientação do Prof. Dr. ANDRÉ AUGUSTO MARIANO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 30 de Janeiro de 2023.

Assinatura Eletrônica
31/01/2023 14:27:47.0
ANDRÉ AUGUSTO MARIANO
Presidente da Banca Examinadora

Assinatura Eletrônica
01/02/2023 11:58:43.0
LUIS HENRIQUE ASSUMPÇÃO LOLIS
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
31/01/2023 13:24:02.0
SAMUEL BARALDI MAFRA
Avaliador Externo (INSTITUTO NACIONAL DE TELECOMUNICAÇÕES)

Assinatura Eletrônica
31/01/2023 14:48:23.0
ANDREI CAMPONOGARA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Em memória de Victor Romano Pereira Lourenço

Agradecimentos

Agradeço a Deus pela vida e criação do nosso fascinante e misterioso Universo.

Agradeço aos meu pais Ezequiel e Antonia por terem me proporcionado educação, carinho e amor.

À minha irmã Kelly pelo companheirismo, apoio e carinho.

Ao meu professor orientador André Augusto Mariano e professor coorientador Evelio Martín García Fernandez pela oportunidade, confiança e conhecimentos transmitidos.

Ao meu amigo Lucas Tokarski pela amizade e pelo apoio em momentos difíceis.

Ao meu amigo Ricardo Borba cuja amizade, confiança e companheirismo são inestimáveis.

Às pessoas do departamento de Engenharia Elétrica da UFPR e a todos os professores que contribuíram para a minha formação como Mestre em Engenharia Elétrica.

*“A sabedoria com as coisas da vida não consiste,
ao que me parece, em saber o que é preciso fazer,
mas em saber o que é preciso fazer antes
e o que fazer depois.”
(Liev Nikoláievich Tolstói)*

RESUMO

Com a rápida evolução de sistemas de telecomunicações, muitos padrões são estudados e implementados, fazendo com que a demanda por dispositivos de radiofrequência configuráveis aumente. Estes dispositivos especiais em que é possível manipular parâmetros como taxa de amostragem, frequência, ganho, entre outros recursos são chamados de *dispositivos de rádio definido por software*, úteis em validações e testes de novas tecnologias. Porém, devido ao alto custo que apresentam, é relativamente raro o emprego de equipamentos que são capazes de transmitir e receber dados ao mesmo tempo, sobrando apenas algumas alternativas de baixo custo limitadas à recepção de sinais, sendo um desafio o emprego desses equipamentos no ensino em universidades. Sendo assim, o objetivo deste trabalho é configurar e implementar um sistema de acesso remoto de dispositivos de rádio definido por software, para uso universitário, de modo que seja possível controlar remotamente equipamentos de rádio de custo elevado que normalmente não teriam seu acesso facilitado aos estudantes. Com o acesso através do protocolo *Secure Shell*, realiza-se uma conexão com um servidor executando o software de processamento de sinais digitais GNU Radio, em que a partir deste são enviados comandos para transmitir e receber sinais por meio de hardwares de radiofrequência. Através de estudos de caso, foi possível validar o sistema de acesso remoto por meio de computadores destinados aos professores e alunos, com diferentes arranjos de transmissores e receptores espalhados fisicamente no ambiente de testes dentro da universidade.

Palavras-chave: Dispositivos de rádio definido por software. GNU Radio. Processamento digital de sinais. Radiofrequência. USRP. RTL-SDR.

ABSTRACT

With the rapid evolution of telecommunication systems, many standards are studied and deployed, increasing the demand for configurable radiofrequency devices. This kind of special device in which parameters like sampling rate, frequency, gain, and other resources are configurable is called software defined radio, extremely useful for testing and validation of new technologies. However, due to their higher cost, it is relatively rare the employment of equipment that is capable of both transmission and reception of signals, leaving only a few low-cost alternatives limited to reception only, challenging its use in the universities. Thus, the objective of this work is to configure and implement a remote access system consisting of software-defined radios, for university use, in which it is possible to remotely control expensive radio equipment that normally would be out of reach from the students. With the remote access done by the Secure Shell protocol, a connection with the server is established, executing the digital signal processing software GNU Radio from which commands are sent to transmit and receive signals across radiofrequency hardware. Through study cases, the remote access system was validated from computers destined to be used by students and instructors, with different transmitters and receiver arrangements physically scattered in the test ambients inside the university.

Keywords: Radio defined software. GNU Radio. Digital signal processing. Radiofrequency. USRP. RTL-SDR.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Cenário onde os alunos (ou professores) acessam equipamentos de radiofrequência a partir de seus computadores | 20 |
| Figura 2 – Cenário onde os alunos (ou professores) acessam um equipamento de radiofrequência a partir de um servidor conectado à rede | 21 |
| Figura 3 – Cenário onde os alunos (ou professores) acessam um ou mais equipamentos de radiofrequência a partir de um servidor conectado à rede | 22 |
| Figura 4 – Cenário onde os alunos acessam um equipamento de radiofrequência a partir de seus computadores, em que um professor controla um equipamento transmissor | 22 |
| Figura 5 – Diagrama simplificado de um dispositivo de rádio definido por software | 24 |
| Figura 6 – USRP N210 | 25 |
| Figura 7 – RTL-SDR | 26 |
| Figura 8 – Funcionamento do protocolo SSH | 29 |
| Figura 9 – Diagrama de blocos do sistema exemplificando a metodologia de acesso remoto | 30 |
| Figura 10 – Diagrama de blocos do sistema em que ambos aluno e professor acessam remotamente o servidor | 31 |
| Figura 11 – Servidor com o USRP N210 conectado em sua porta <i>ethernet</i> | 33 |
| Figura 12 – Notebook executando o GNU Radio com o RTL-SDR conectado em sua porta USB | 34 |
| Figura 13 – Comandos do terminal para realizar conexão SSH com o servidor | 35 |
| Figura 14 – Comandos do terminal para executar o gnuradio em modo gráfico | 35 |
| Figura 15 – Tela inicial do programa GNU Radio | 36 |
| Figura 16 – Fluxograma do transmissor FM executado pelo servidor | 37 |
| Figura 17 – Fluxograma do receptor FM executado pelo notebook | 37 |
| Figura 18 – Espectro de frequência do sinal recebido | 38 |
| Figura 19 – Forma de onda dos sinais de áudio | 38 |
| Figura 20 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR | 39 |
| Figura 21 – Forma de onda dos sinais de áudio | 39 |
| Figura 22 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR | 40 |
| Figura 23 – Forma de onda dos sinais de áudio | 40 |
| Figura 24 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR | 41 |

| | |
|---|----|
| Figura 25 – Comparativo entre as formas de onda da diferença entre o áudio original e os áudios recebidos pelo RTL-SDR | 41 |
| Figura 26 – Comandos do terminal para realizar conexão SSH com o servidor | 44 |
| Figura 27 – Comandos do terminal para executar o gnuradio em modo gráfico | 44 |
| Figura 28 – Fluxograma do transmissor QPSK simulado | 46 |
| Figura 29 – Fluxograma do receptor QPSK simulado | 46 |
| Figura 30 – Diagrama de constelação QPSK em canal de comunicação simulado | 47 |
| Figura 31 – Diagrama de olho com a modulação QPSK em canal de comunicação simulado | 47 |
| Figura 32 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres não estão sincronizados | 48 |
| Figura 33 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres estão sincronizados | 48 |
| Figura 34 – Fluxograma do transmissor QPSK | 49 |
| Figura 35 – Fluxograma do receptor QPSK | 50 |
| Figura 36 – Diagrama de constelação | 50 |
| Figura 37 – Diagrama de olho | 51 |
| Figura 38 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres não estão sincronizados | 51 |
| Figura 39 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres estão sincronizados | 52 |
| Figura 40 – Diagrama de constelação | 52 |
| Figura 41 – Diagrama de olho | 53 |
| Figura 42 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres estão sincronizados | 53 |
| Figura 43 – Diagrama de constelação | 54 |
| Figura 44 – Diagrama de olho | 54 |
| Figura 45 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres estão sincronizados | 55 |
| Figura 46 – Página inicial do programa GNU Radio | 62 |
| Figura 47 – Guia para abrir um novo fluxograma | 63 |
| Figura 48 – Inserção de um elemento ao fluxograma | 63 |
| Figura 49 – Fluxograma com blocos conectados entre si | 64 |
| Figura 50 – Configurações avançadas do bloco Variable | 64 |
| Figura 51 – Configurações avançadas do bloco Signal Source | 65 |
| Figura 52 – Configurações avançadas do bloco Variable | 65 |
| Figura 53 – Sinal senoidal representado no tempo e na frequência | 66 |
| Figura 54 – Fluxograma com blocos possuindo tipos de variáveis diferentes em conflito (float e complex) | 67 |

| | |
|---|----|
| Figura 55 – Fluxograma com blocos possuindo variáveis do mesmo tipo (float) . . . | 68 |
| Figura 56 – Sinal senoidal representado no tempo e na frequência, com o sinal de entrada do tipo float | 69 |
| Figura 57 – Fluxograma com bloco Multiply | 70 |
| Figura 58 – Janela de configurações do bloco QT Gui Frequency Sink | 71 |
| Figura 59 – Fluxograma completo | 72 |
| Figura 60 – Sinal senoidal representado no tempo e na frequência, com um dos sinais multiplicado por 2 | 73 |
| Figura 61 – Janela inicial do programa | 77 |
| Figura 62 – Janela de popup solicitando ao usuário para selecionar as placas de som | 78 |
| Figura 63 – Janela de configurações do programa | 78 |
| Figura 64 – Áudio original e com ruído adicionado | 79 |
| Figura 65 – Janela principal do programa | 80 |
| Figura 66 – Ruído extraído pelo <i>DiffMaker</i> | 81 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Principais características do USRP | 25 |
| Tabela 2 – Principais características do RTL-SDR | 27 |
| Tabela 3 – Comparativo entre o áudio original e os áudios recebidos pelo RTL-SDR | 42 |

Lista de abreviaturas e siglas

| | |
|--------------|---|
| <i>ADC</i> | <i>Analog-Digital Converter</i> |
| <i>ADSB</i> | <i>Automatic Dependent Surveillance-Broadcast</i> |
| <i>AM</i> | <i>Amplitude Modulation</i> |
| <i>DAC</i> | <i>Digital-Analog Converter</i> |
| <i>CC</i> | <i>Corrente Direta</i> |
| <i>DELT</i> | <i>Departamento de Engenharia Elétrica</i> |
| <i>DVB-T</i> | <i>Digital Video Broadcast - Terrestrial</i> |
| <i>EAD</i> | <i>Ensino à Distância</i> |
| <i>5G</i> | <i>5th Generation of Mobile Telecommunications Technology</i> |
| <i>FM</i> | <i>Frequency Modulation</i> |
| <i>GFSK</i> | <i>Gaussian Frequency Shift Keying</i> |
| <i>GICS</i> | <i>Group of Intergrated Circuits and Systems</i> |
| <i>GPS</i> | <i>Global Positioning System</i> |
| <i>IoT</i> | <i>Internet of Things</i> |
| <i>IIoT</i> | <i>Industrial Internet of Things</i> |
| <i>IP</i> | <i>Internet Protocol</i> |
| <i>ISM</i> | <i>Industrial, Scientific and Medical Band</i> |
| <i>LTE</i> | <i>Long Term Evolution</i> |
| <i>SSH</i> | <i>Secure SHell</i> |
| <i>PM</i> | <i>Phase Modulation</i> |
| <i>PSNR</i> | <i>Peak Signal Noise Ratio</i> |
| <i>QAM</i> | <i>Quadrature Amplitude Modulation</i> |
| <i>QPSK</i> | <i>Quadrature Phase Shift Keying</i> |

| | |
|--------------|--|
| <i>TCP</i> | <i>Transport Layer Protocol</i> |
| <i>TETRA</i> | <i>Terrestrial Trunked Radio Access</i> |
| <i>UFPR</i> | <i>Universidade Federal do Paraná</i> |
| <i>USB</i> | <i>Universal Serial Bus</i> |
| <i>USRP</i> | <i>Universal Software Radio Peripheral</i> |
| <i>V2X</i> | <i>Vehicle-to-everything</i> |
| <i>VPN</i> | <i>Virtual Private Network</i> |

Sumário

| | | |
|------------|--|-----------|
| 1 | INTRODUÇÃO | 15 |
| 2 | FORMULAÇÃO DO PROBLEMA | 20 |
| 3 | MATERIAIS E MÉTODOS | 23 |
| 3.1 | Equipamentos de rádio definido por software | 23 |
| 3.1.1 | USRP N210 | 24 |
| 3.1.2 | RTL-SDR | 26 |
| 3.2 | GNU Radio | 27 |
| 3.3 | CentOS 7 /Anaconda | 28 |
| 3.4 | SSH | 28 |
| 3.5 | Metodologia de acesso remoto aos dispositivos de rádio definido por software | 29 |
| 3.5.1 | Acesso remoto a uma rede interna dentro da universidade | 30 |
| 3.5.2 | Acesso remoto a uma rede externa à universidade | 30 |
| 4 | VALIDAÇÃO DO ACESSO REMOTO À EQUIPAMENTOS DE RÁDIO DEFINIDO POR SOFTWARE | 32 |
| 4.1 | Acesso remoto a equipamentos de rádio definido por software em uma rede local na universidade | 32 |
| 4.1.1 | Teste de conectividade SSH em rede interna com o servidor | 34 |
| 4.1.2 | Transmissão e recepção de áudio em modulação FM | 36 |
| 4.2 | Acesso remoto a equipamentos de rádio definido por software em uma rede externa à universidade | 43 |
| 4.2.1 | Teste de conectividade SSH em rede externa com o servidor | 43 |
| 4.2.2 | Transmissão e recepção de sinais QPSK | 44 |
| 4.2.2.1 | Experimento realizado através de canal de comunicação simulado ideal | 45 |
| 4.2.2.2 | Experimento realizado através de canal de comunicação sem fio utilizando equipamentos de rádio definido por software | 49 |
| 5 | CONCLUSÃO E SUGESTÕES | 56 |
| | REFERÊNCIAS | 58 |

| | | |
|------------|---|-----------|
| | APÊNDICES | 61 |
| | APÊNDICE A – TUTORIAL DE UTILIZAÇÃO DO GNU RADIO . | 62 |
| | APÊNDICE B – CONFIGURAÇÃO DO SERVIDOR | 74 |
| B.1 | Instalação do GNU Radio | 74 |
| B.2 | Conexão dos equipamentos de rádio | 76 |
| | APÊNDICE C – TUTORIAL DE UTILIZAÇÃO DO DIFFMAKER . | 77 |

1 INTRODUÇÃO

As comunicações sem fio estão em constante evolução, gerando novos desafios e novas aplicações em que se faz necessário a condução de testes e validações para o desenvolvimento e aprimoramento de diversas tecnologias. Com o desenvolvimento de novas redes, com destaque para a implantação do 5G, oportunidades surgiram no desenvolvimento e aplicações envolvendo internet das coisas (IoT - do inglês, *Internet of Things*), como a internet industrial das coisas (IIoT - do inglês, *Industrial Internet of Things*) com foco em indústrias onde são estudados os requerimentos e as relações entre redes 5G móveis e as demandas industriais (VARGA et al., 2020); no ramo agrícola com estudos de 5G em aplicações envolvendo operação de drones para realizar manutenções no campo, realidade aumentada para obtenção de informações e tomada de decisões, robôs autônomos com aplicações de inteligência artificial, sensoriamento remoto e outras aplicações (TANG et al., 2021); desenvolvimento de redes 5G *vehicle-to-everything* (V2X) para controle de tráfego, obtenção de dados de meio-ambiente, posicionamento de veículos, sistemas de entretenimento entre outros (STORCK; DUARTE-FIGUEIREDO, 2019); aplicações em cidades inteligentes tais como monitoramento de parâmetros de ambiente para análise de qualidade do ar, sensores de estacionamento para indicar disponibilidade de vagas, monitoramento de densidade de trânsito para otimizar rotas e/ou gerenciar parâmetros de semáforos, controle de irrigação em jardins para manutenção de parques e áreas verdes, gestão de resíduos para organização de lixo e otimizar a coleta seletiva, participação de usuários em reportar ou receber relatórios de eventos e/ou incidentes, etc. (BAUER; SANCHEZ; SONG, 2021); entre outras comunicações visando automação, monitoramento e segurança. Devido à grande quantidade de padrões de comunicações sem fio, para a realização de experimentos há o alto custo relacionado à necessidade de se ter vários equipamentos diferentes, visto que normalmente os equipamentos operam com parâmetros fixos que servem apenas a um propósito específico para o qual este fora construído. No entanto, existem dispositivos de rádio especiais que podem ser reconfigurados através de software, em que é possível controlar diversos parâmetros como frequência, taxa de amostragem, ganho, e assim por diante. Tais dispositivos são chamados de rádio definido por software.

Um dos dispositivos mais comumente utilizado para o desenvolvimento dos mais diversos tipos de trabalhos é o USRP (do inglês, *Universal Software Radio Peripheral*), sigla para *Universal Software Radio Peripheral*. Como exemplos, há autores que o utiliza para analisar sinais de radiofrequência com modulação digital tais como 16-QAM e 64-QAM (do inglês, *Quadrature Amplitude Modulation*) e também analisar o impacto de métodos de janelamento em filtros (BODDU, 2019); análise e geração de sinais de geolocalização

via satélite para a análise de performance e robustez de algoritmos de posicionamento em que são requeridos *testbeds* de sistemas de rádio configuráveis (DIOUF et al., 2021); estudos acerca da coleta de energia em tempo real através de ondas de radiofrequência, utilizadas para minimizar a dependência de baterias em dispositivos IoT de baixo consumo, com a análise de redes 5G e LTE (do inglês, *Long Term Evolution*) (GAUTAM et al., 2021); detecção de padrões respiratórios em pacientes para diagnóstico e triagem de infectados pela doença do coronavírus (REHMAN et al., 2021); entre diversos outros trabalhos. Um problema, todavia, consiste no alto custo que os equipamentos de rádio definido por software apresentam, dificultando o acesso destes aos pesquisadores. Algumas pesquisas vêm sendo feitas na área a fim de descobrir alternativas de baixo custo para a experimentação e uso, como em (SALES; MARIANO; FERNANDEZ, 2022) em que se utiliza um Raspberry Pi para atuar como transmissor além de um RTL-SDR conectado a este para receber os sinais de radiofrequência. Similarmente, em (FRANCISCO et al., 2019) é usado um Raspberry Pi em conjunto com um RTL-SDR (SDR vem do inglês *Software Defined Radio*, ou rádio definido por software, e RTL é pelo fato do dispositivo possuir um *chip Realtek RTL2832*) para a análise de comunicação sem fio, porém, ainda requerendo o uso de um USRP para a realização da transmissão de sinais. Em (RHARRAS; SABER, 2018), os autores utilizaram uma configuração parecida com a de (FRANCISCO et al., 2019) em conjunto com o software de processamento de sinais MATLAB, para a análise de sinais FM (do inglês, *Frequency Modulation*). Os autores (GOWDA et al., 2021) desenvolveram um dispositivo capaz de receber e transmitir sinais utilizando o conjunto Raspberry Pi com RTL-SDR, utilizando os softwares MATLAB e Simulink para implementar a recepção e transmissão de sinais de áudio utilizando a modulação FM. No trabalho proposto por (MESHRAM; KOLHARE, 2019), foi desenvolvido um sistema baseado em rádio definido por software utilizando um RTL-SDR e um Raspberry Pi, os quais são usados para receber sinais FM, em que os autores fizeram a análise dos sinais através do GNU Radio (GNU vem da sigla recursiva em inglês *GNU's not UNIX*, em que UNIX é um acrônimo para *UNiplexed Information Computing System*).

Alguns trabalhos também visam a mobilidade, como em (TOMAR; BHATIA, 2015), no qual é usado uma configuração que combina o uso de um Raspberry Pi com um RTL-SDR para fazer um transceptor com o intuito de ser usado em locais atingidos por desastres, de modo a apresentar baixo custo e boa mobilidade. Outros trabalhos, como em (BALL, 2020) e (PEROTONI; SILVA; VIEIRA, 2021), propuseram um equipamento de rádio definido por software portátil e de baixo custo para estudos de propagação de radiofrequência, onde é possível utilizar o hardware em várias localidades para realizar medições pontuais, com o auxílio de GPS (do inglês, *Global Positioning System*). Porém, os dispositivos gravam as informações e o acesso é feito posteriormente, existindo a oportunidade de implementar um acesso remoto ao equipamento. Como o sistema usa apenas o RTL-SDR, não há a possibilidade de transmitir sinais de rádio. Em (HELBET et al., 2021), com o objetivo

de receber dados de rádiofrequência remotamente foi desenvolvido um sensor baseado em rádio definido por software que utiliza o protocolo **SSH** (do inglês, *Secure Shell*) em conjunto com uma rede virtual privada (**VPN**, do inglês *Virtual Private Network*) para monitorar sistemas de rádio **TETRA** (do inglês, *Terrestrial Trunked Radio Access*, com maiores informações em (INSTITUTE, 2000)), no qual é utilizado um Raspberry Pi com um ADALM Pluto (ADALM vem do inglês *Analog Devices Active Learning Module*) e Hack RF One (com RF do inglês *Radio Frequency*) para receber os sinais e enviá-los para a rede com auxílio do software GNU Radio. Ainda que tenham sido usados equipamentos de rádio capazes de realizar a transmissão de sinais, o potencial foi utilizado apenas para a recepção de sinais **TETRA**.

Com a diversidade de aplicações e técnicas de experimentação, softwares de processamento digital de sinais são extensivamente utilizados para testar e validar hipóteses levantadas em estudos de caso, capazes de gerarem sinais e análises em tempo real (BRAGA, 2011). Alguns autores utilizaram o software GNU Radio na tentativa de ministrar aulas virtuais durante a pandemia, como em (FUHRMANN; FARMBAUER; NIEMETZ, 2021), cujo acesso às salas de aula fora restrito no período. As simulações gráficas aliadas aos diagramas de blocos auxiliam na criação de aulas virtuais, onde os alunos podem interagir com as sessões de aula realizando experimentos. A utilização de ferramentas de simulação melhora a experiência de aprendizado e a motivação pela busca de conhecimento dos estudantes. Porém, como o ambiente é simulado, alguns experimentos podem não condizer com aplicações práticas do que está sendo estudado, pois nem sempre é possível modelar todos os tipos de efeitos de canal de comunicação que o ambiente pode introduzir através das simulações. Assim, uma solução ao problema consiste em utilizar equipamentos de rádio definido por software conjuntamente com programas de processamento digital de sinais para conseguir simular e realizar experimentos considerando canais de comunicação sem fio reais. Os softwares atualmente contam com o suporte para se utilizar equipamentos de rádiofrequência, sendo possível realizar análises mais precisas levando em consideração fatores presentes nos ambientes físicos do que a realização de cenários puramente simulados, que utilizados em salas de aula fornecem um apoio para que estudantes tenham contato com dispositivos de rádio definido por software e as ferramentas de manipulação digital de sinais (CÂNDIDO, 2020). Em (ALMALEH, 2009), o autor realiza a integração de um dispositivo de rádio definido por software LimeSDR (com SDR do inglês *Software Defined Radio*) ao programa de processamento digital de sinais GNU Radio, no qual foram analisadas transmissões com modulação **GFSK** (do inglês, *Gaussian Frequency Shift Keying*) e **QPSK** (do inglês, *Quadrature Frequency Shift Keying*) para explorar as possibilidades de aplicação do LimeSDR num ambiente de experimentos. Em (DINIZ, 2013), a autora utiliza um **USRP** para realizar análise de sinais aplicados ao ensino em engenharia, com simulações realizadas no software de processamento digital de sinais GNU Radio, expandindo posteriormente em (DINIZ, 2019) à utilização do **USRP** em aplicações

de sala de aula invertida (em que o aluno aprende os conceitos em casa antes da aula e, posteriormente, discute os conhecimentos adquiridos junto ao professor durante a aula), também utilizando o GNU Radio.

Diante das linhas de estudos mencionadas, pode-se dizer que o uso de rádios definidos por software possuem amplas aplicações, sendo possível construir sistemas com diferentes arranjos de equipamentos e operá-los com softwares de processamento digital de sinais, inclusive sendo aplicados em sala de aula onde estudantes podem usufruir de experimentos que fazem utilização de dispositivos capazes de se adaptarem a diversas faixas de frequência e velocidades de transmissão e recepção. Portanto, para enriquecer as simulações realizadas em sala de aula em experimentos voltados às telecomunicações e tornar acessível o uso de equipamentos de rádio definido por software que normalmente têm seu custo elevado, o presente trabalho foca em estabelecer uma plataforma fixa em uma universidade cujo acesso aos equipamentos é feito de forma remota, aumentando a mobilidade e facilitando o uso compartilhado, permitindo assim que estudantes e pesquisadores possam usufruir de equipamentos de alto desempenho através do compartilhamento de recursos à distância.

Objetivos

O objetivo geral deste trabalho é desenvolver uma metodologia que permita o acesso remoto e controle de hardwares de rádio definido por software para o posterior uso acadêmico, visando tornar a metodologia de ensino e experimentação simples e fácil.

Os objetivos específicos são:

- Definir equipamentos de rádio definido por software adequados para o acesso remoto no sistema;
- Definir um protocolo de comunicação para acesso remoto;
- Definir software que permita a manipulação dos dispositivos de rádio via rede;
- Realizar estudos de caso que:
 - Envolvam diferentes configurações de localização dos dispositivos transmissores e receptores;
 - Envolvam diferentes modulações.

Trabalhos publicados

Em (SALES; MARIANO; FERNANDEZ, 2022), foi desenvolvido uma plataforma baseada em rádio definido por software de baixo custo que tem como diferencial a comunicação remota via TCP (do inglês, *Transmission Control Protocol*).

Organização da Dissertação

O [segundo capítulo](#) aborda a formulação do problema referente ao alto custo de equipamentos de radiofrequência e o acesso remoto destes, descrevendo-se em maiores detalhes quais os pontos a serem atacados e o que é necessário para a sua solução.

O [terceiro capítulo](#) apresenta os materiais necessários para a solução do problema, tanto na parte de hardware quanto em software utilizados para a realização do acesso remoto aos equipamentos. Na conclusão do capítulo, são comentados os métodos a serem aplicados para que em conjunto com os materiais se atinja a solução idealizada no capítulo anterior.

O [quarto capítulo](#) expõe os testes e validações da metodologia aplicada, mostrando dois casos de estudo em que o acesso remoto a equipamentos de rádio definido por software são utilizados para a transferência de um arquivo de áudio com modulação [FM](#) e transferência de arquivo de texto utilizando modulação [QPSK](#), observando-se os efeitos de um canal real de comunicação sem fio.

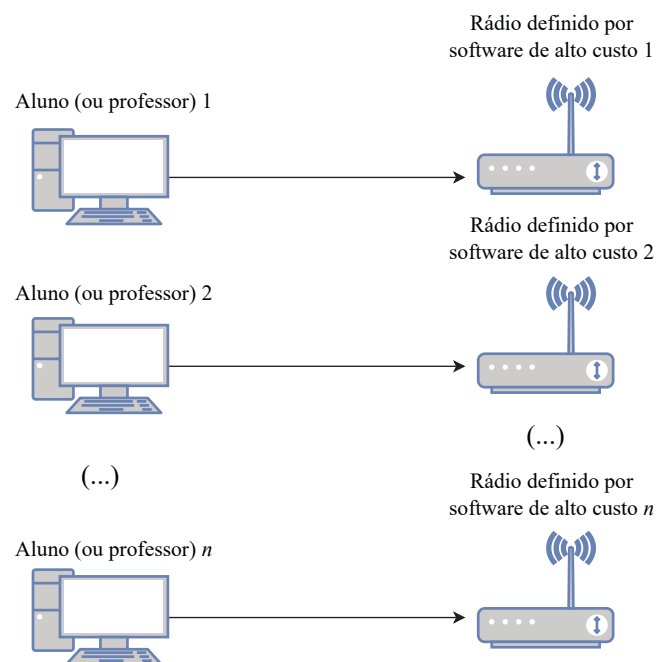
Por fim, o [quinto e último capítulo](#) possui o objetivo de concluir o trabalho, ressaltando os resultados obtidos pelas validações do sistema proposto e apresentar sugestões para expandir as funcionalidades e melhorar a metodologia.

2 FORMULAÇÃO DO PROBLEMA

Neste capítulo serão abordados a descrição do problema, que envolve o alto custo de equipamentos e a capacidade de poder operá-los à distância, e uma possível forma de solucionar através de programas de computador, protocolos de comunicação e equipamentos de rádiofrequência disponíveis, que interligados provê o compartilhamento de recursos entre estudantes e professores para a realização de experimentos.

Conforme discutido no [Capítulo 1](#), existe uma variedade de aplicações envolvendo sinais de radiofrequência, fazendo com que seja interessante o emprego de softwares de processamento digital de sinais para realizar experimentos em disciplinas de universidades. Tais programas permitem com que alunos consigam realizar experimentos simulando diversas tecnologias e realizando estudos de caso para validação de conceitos aprendidos em sala de aula, aumentando-se o engajamento e grau de aprendizado. Porém, como os cenários são simulados, é difícil modelar alguns parâmetros que dependem de fatores do ambiente físico. Sendo assim, o emprego de rádios definido por software nos experimentos solucionaria esse problema, com a [Figura 1](#) mostrando um cenário típico onde os alunos (ou professores) em uma sala de aula acessam dispositivos de radiofrequência para a realização de experimentos, cada um com acesso à um equipamento.

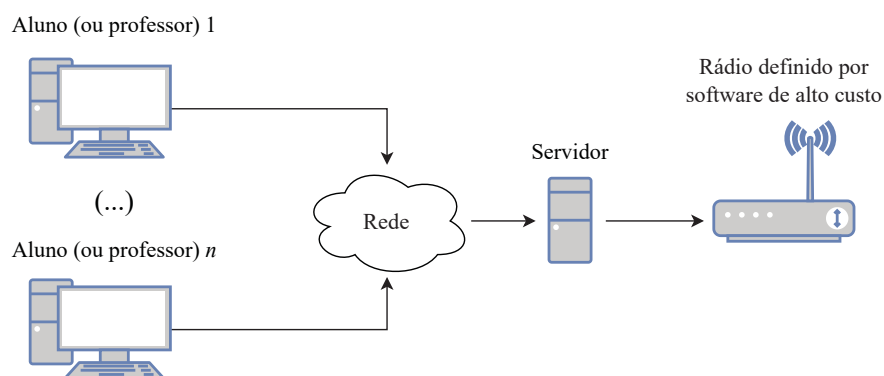
Figura 1 – Cenário onde os alunos (ou professores) acessam equipamentos de radiofrequência a partir de seus computadores



Fonte: O autor (2023).

No entanto, esses equipamentos possuem custo elevado, sendo difícil a aplicação de vários dispositivos para suprir as demandas de utilização por vários alunos e professores em uma universidade. Assim, o compartilhamento de dispositivos pela rede é uma solução possível para eliminar a necessidade de aquisição e operação de vários equipamentos, cortando-se os custos e otimizando o uso de recursos, como mostrado na [Figura 2](#), em que vários alunos (ou professores) acessam pela rede um servidor e por meio deste os equipamentos de rádio definido por software.

Figura 2 – Cenário onde os alunos (ou professores) acessam um equipamento de radio-frequência a partir de um servidor conectado à rede

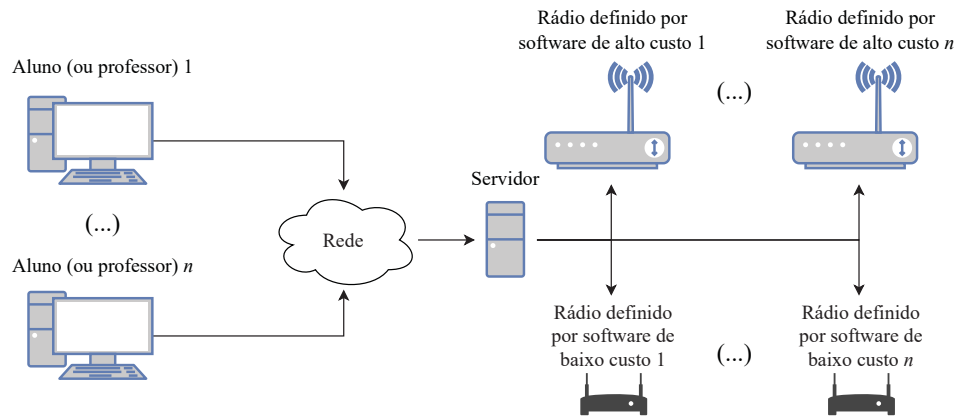


Fonte: O autor (2023).

Para realizar a configuração de um sistema de compartilhamento via rede usa-se um servidor no qual é conectado dispositivos de rádio definido por software. Neste servidor é instalado um software de processamento digital de sinais capaz de controlar diferentes dispositivos de radiofrequência, particularmente os que foram conectados. Para que seja possível a troca de informações entre o servidor e outros computadores através da rede, utiliza-se um protocolo de comunicação que permita o acesso remoto da área de trabalho do servidor, em que posteriormente alunos e professores irão acessar o programa de processamento digital de sinais para realizar experimentos com os dispositivos de rádio definido por software.

Uma limitação do cenário apresentado anteriormente pela [Figura 2](#) é de o acesso não poder ocorrer com mais de um usuário em simultâneo. Caso dois ou mais usuários tentem acessar o dispositivo de radiofrequência, há a possibilidade de os comandos enviados por um usuário serem sobrescritos por outro usuário, sendo portanto necessário a divisão de acesso por meio de acordos em que determinado horário do dia apenas um usuário terá o acesso ao dispositivo de rádio definido por software para uso em experimentos. Para solucionar o problema de múltiplos acessos em simultâneo dos dispositivos, é possível que pela rede sejam conectados mais de um equipamento de radiofrequência no servidor, como mostrado na [Figura 3](#), em que cada usuário poderá controlar um dispositivo que não esteja em uso.

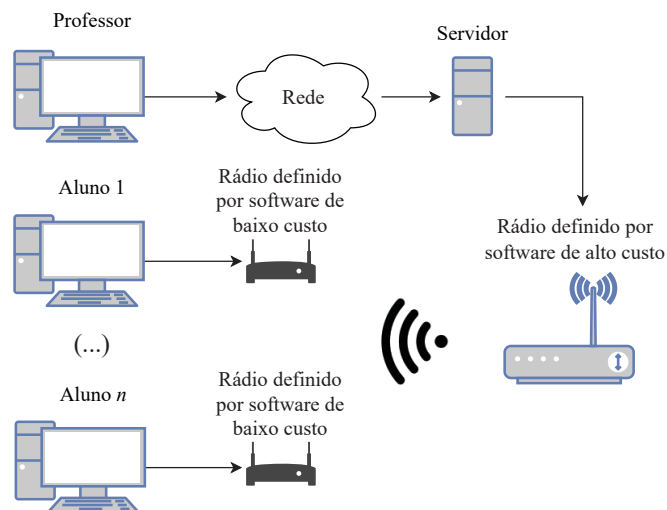
Figura 3 – Cenário onde os alunos (ou professores) acessam um ou mais equipamentos de radiofrequência a partir de um servidor conectado à rede



Fonte: O autor (2023).

Também é possível realizar um acesso misto, com o equipamento de alto custo disponível para controle pela rede como um transmissor, enquanto que computadores locais utilizam dispositivos receptores de baixo custo para receber os sinais transmitidos pelo equipamento mais caro. Uma ilustração deste caso é realizado na [Figura 4](#), onde um professor controla um transmissor enquanto que alunos recebem os sinais através de equipamentos de baixo custo instalados em suas máquinas.

Figura 4 – Cenário onde os alunos acessam um equipamento de radiofrequência a partir de seus computadores, em que um professor controla um equipamento transmissor



Fonte: O autor (2023).

No capítulo a seguir são apresentados os materiais necessários e a metodologia de utilização que visa tornar possível realizar experimentos à distância com o controle de equipamentos de rádio definido por software pela rede e realizar o processamento digital de sinais observando-se os efeitos dos canais de comunicação pelo ar.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais necessários e a metodologia para a realização do acesso remoto a equipamentos de radiofrequência por meio de softwares de processamento digital de sinais, visando o uso educacional e compartilhamento de recursos.

Para que seja possível realizar experimentos e constatar os efeitos de um canal real de comunicação (por exemplo, pelo ar), é necessário a utilização de equipamentos especiais em que pode-se manipular seus parâmetros (tais como ganho, taxa de amostragem, faixa de frequências, entre outros), de modo a integrar em programas destinados à manipulação e processamento digital de sinais para a obtenção das métricas relacionadas aos estudos realizados. Também, o software de processamento de sinais precisa oferecer suporte à operação remota pela internet e permitir a interface com um equipamento de radiofrequência.

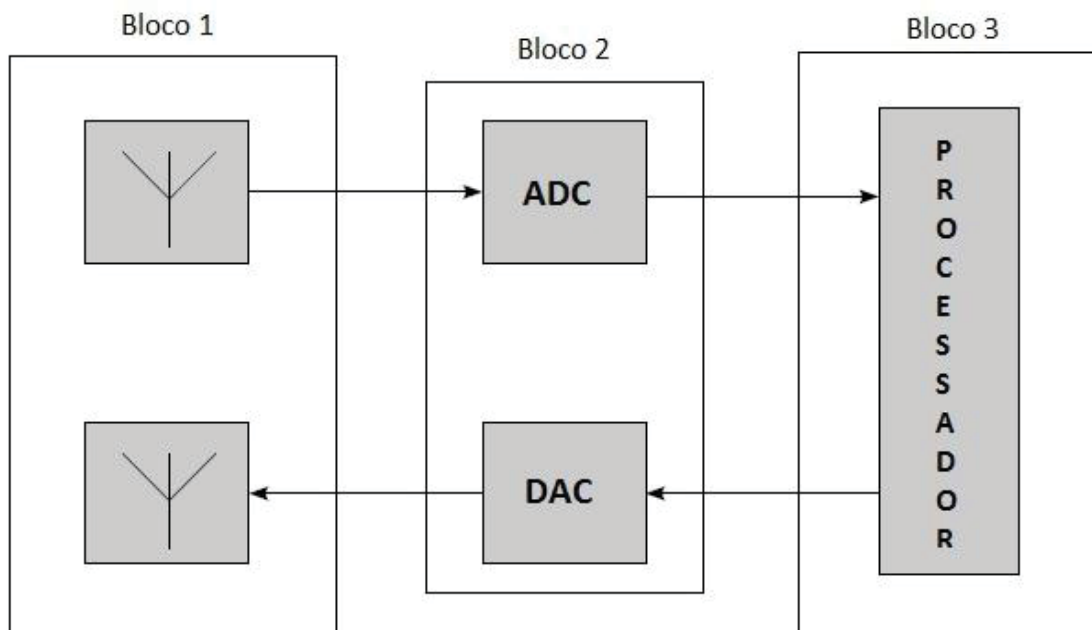
Na seção a seguir, é descrito o funcionamento dos equipamentos de rádio definido por software, particularmente nas subseções seguintes, o [USRP N210](#) e o RTL-SDR. A escolha destes equipamentos é baseada tanto na popularidade e capacidades de customização dos parâmetros quanto na disponibilidade dos equipamentos pela Universidade Federal do Paraná ([UFPR](#)). O [USRP N210](#) possui a vantagem de ter um hardware robusto, altamente configurável e de alta performance, com capacidades de transmissão desde nível [CC](#) (corrente direta) até em torno de 6 GHz, podendo cobrir uma ampla gama de aplicações graças à sua flexibilidade com relação à faixa de frequências suportada. Além disto, pode funcionar como um transceptor, eliminando-se a necessidade de dois dispositivos para a realização de experimentos. Já o RTL-SDR é um receptor que possui uma grande popularidade devido ao seu baixo custo e relativa disponibilidade. A faixa de frequências cobre desde 25 MHz (em alguns modelos é suportado desde 500 kHz) até 1766 MHz, suficiente para a realização de diversos experimentos, podendo ser usado em conjunto com o [USRP N210](#).

3.1 Equipamentos de rádio definido por software

Os equipamentos de rádio definido por software, como o próprio nome diz, são dispositivos em que é possível controlar os parâmetros de comunicação via software, tais como ganho, frequência, taxa de amostragem, etc. A [Figura 5](#) mostra um diagrama simplificado de um dispositivo de rádio definido por software. Na recepção de sinais, o sistema capta o sinal no bloco 1 em uma antena que passa por um [ADC](#) (do inglês, *Analog-Digital Converter*) no bloco 2, o qual converte os sinais analógicos em digitais

e então é computado no bloco 3 por um processador para a realização de filtragem e o processamento digital dos sinais. Para a transmissão de sinais, o sinal digital é gerado através do processador do bloco 3 e convertido para analógico através do DAC (do inglês, *Digital-Analog Converter*) do bloco 2, que é então transmitido pela antena do bloco 1.

Figura 5 – Diagrama simplificado de um dispositivo de rádio definido por software



Fonte: Janson (2012).

Nas próximas subseções são mostrados com maiores detalhes os equipamentos de rádio definido por software utilizados neste trabalho.

3.1.1 USRP N210

O **USRP N210** é um equipamento de rádio definido por software, desenvolvido pela Ettus Research. Consiste de uma placa com um FPGA (do inglês, *Field Programmable Gate Array*), dois ADCs de 100 MS/s, dois DACs de 400 MS/s e uma porta Ethernet Gigabit para transmitir dados para o processador hospedeiro. Pode operar desde CC até na faixa de 6 GHz (RESEARCH, 2022b), a depender de um transceptor externo que pode ser acoplado à placa mãe, o qual é chamado de *daughterboard*. A Figura 6 ilustra uma foto do dispositivo.

Figura 6 – USRP N210



Fonte: [Research \(2022c\)](#).

Devido a seu alto desempenho e flexibilidade de operação em frequência, é um dispositivo bastante usado para a realização de experimentos. A taxa de amostragem alta permite a análise de sinais de banda larga, a precisão do [ADC/DAC](#) melhora a quantidade de pontos da forma de onda obtida/transmitida e o intervalo de frequência de operação permite que possam ser analisados mais tipos de aplicações, como por exemplo, desde análise de rádios [FM](#) (do inglês, *frequency modulation* na faixa de centenas de megahertz até comunicações WiFi de 5 GHz. A [Tabela 1](#) lista as principais características do [USRPN](#), em que os parâmetros listados definem o limite de que se pode alcançar com relação à quantidade de tecnologias em que é possível realizar experimentos. Como exemplos, a máxima frequência de operação define se o equipamento poderá operar em bandas de WiFi de alta frequência, ou a taxa de amostragem define a máxima frequência atingível de uma fonte de dados, entre outros fatores.

Tabela 1 – Principais características do [USRPN](#)

| Parâmetros | Valores |
|---|-------------------------------------|
| Intervalo de frequência | DC - 6 GHz (depende do transceptor) |
| Taxa de amostragem do ADC | 100 MS/s |
| Precisão do ADC | 14 bits |
| Taxa de amostragem do DAC | 400 MS/s |
| Precisão do DAC | 16 bits |
| Velocidade da porta Ethernet | 1 Gbps |
| Máxima taxa de dados de stream via porta Ethernet | 50 MS/s |
| Acurácia de frequência TCXO | 2,5 ppm |

Fonte: [Research \(2022a\)](#)

O [USRPN](#) utilizado neste trabalho possui um transceptor chamado WBX que é capaz de operar em radiofrequência entre 50 a 2200 MHz, com 16 bits de quantização empregando-se uma largura de banda de 25 MHz ou 8 bits de quantização quando utilizado

50 MHz.

3.1.2 RTL-SDR

RTL é uma abreviação de RTL2832U, um chipset fabricado pela Realtek feito para a transmissão broadcast de vídeo digital (DVB-T, do inglês *Digital Video Broadcast - Terrestrial*). Através de esforços conjuntos de Antti Palosaari, Eric Fry e Osmocom (Steve Markgraf em particular), foi descoberto que as amostras de fase e quadratura (I/Q) do chipset RTL2832U poderiam ser acessadas diretamente, de modo que através de um driver customizado desenvolvido por Steve Markgraf, é possível converter um adaptador DVB-T em um dispositivo de rádio definido por software (RTL-SDR, 2022a).

Assim, foi desenvolvido um dispositivo USB (do inglês, *Universal Serial Bus*) cujo circuito eletrônico possui o chipset RTL2832U, uma entrada para antena e circuitos auxiliares para fazer com que possa ser utilizado como um receptor de rádio definido por software, além de encapsulamento de metal para proteção contra ruídos. A Figura 7 mostra um dispositivo RTL-SDR V3 fabricado pelo RTL-SDR Blog.

Figura 7 – RTL-SDR



Fonte: Elektrologi (2022).

Por ser de baixo custo e de relativamente fácil obtenção, além de ter um desempenho satisfatório para a realização de experimentos em conjunto com o USRP, escolheu-se o RTL-SDR para atuar como o receptor dos sinais de radiofrequência deste trabalho. O intervalo de frequências do dispositivo permite que diversos tipos de aplicações possam ser analisadas, tais como controles remotos, rastreamento de aviões através do sistema ADSB (do inglês, *Automatic Dependent Surveillance-Broadcast*), estações de rádio AM

(do inglês, *amplitude modulation*)/FM, entre outros. O cristal oscilador com compensação de temperatura usado no projeto da placa aumenta a acurácia de frequência mesmo com grandes variações na temperatura de operação. A [Tabela 2](#) lista de forma resumida as principais características do RTL-SDR V3, com os parâmetros listados definindo a máxima capacidade com relação ao número de tecnologias possíveis de se testar, entre outros fatores, conforme discutido previamente com o caso do [USRP](#) na [Subseção 3.1.1](#).

Tabela 2 – Principais características do RTL-SDR

| Parâmetros | Valores |
|--|-------------------|
| Intervalo de frequência | 24 MHz - 1766 MHz |
| Intervalo de frequência em modo de amostragem direto | 500 kHz - 24 MHz |
| Taxa de amostragem do ADC | 2,56 MS/s |
| Precisão do ADC | 8 bits |
| Acurácia de frequência TCXO | 2 ppm |

Fonte: [RTL-SDR \(2022b\)](#) e [RTL-SDR \(2022a\)](#)

Na seção seguinte são apresentados as ferramentas de software utilizados neste trabalho, tais como o sistema operacional Linux, o protocolo de comunicação [SSH](#) e o software de processamento digital de sinais GNU Radio. A escolha do Linux vem do fato de ser um sistema operacional de código livre, podendo ser encontradas várias distribuições gratuitas disponíveis para download na internet. Já o protocolo [SSH](#) foi utilizado pela facilidade de se estabelecer uma conexão remota entre cliente e servidor, além de o protocolo oferecer segurança na conectividade e dispensar softwares pagos para a realização da conexão. Em relação ao software para o desenvolvimento de experimentos, escolheu-se o GNU Radio por possuir código livre e sem custos, além de sua interface intuitiva auxiliar na didática para a utilização nas universidades.

3.2 GNU Radio

O software GNU Radio é um software de desenvolvimento baseado em C++ e Python que possui uma interface gráfica onde através de diagramas de blocos realiza-se o processamento digital de sinais, podendo ser implementados e simulados diversos tipos de aplicações de rádio. Um guia que mostra o passo-a-passo de como realizar simulações básicas no programa está apresentado no [Apêndice A](#). Além das simulações, é possível integrar com equipamentos de radiofrequência de modo a controlar um rádio definido por software ([PROJECT, 2022](#)), particularmente útil em testes de validações de novas tecnologias.

O programa também oferece a possibilidade de criar blocos escritos em C++ ou Python, estendendo-se as funcionalidades já oferecidas e abrindo caminho para testes e

validações de novos conceitos ou elaboração de provas e/ou exercícios para aplicações em usos educacionais.

3.3 CentOS 7 /Anaconda

O servidor utilizado neste trabalho possui o sistema operacional Linux CentOS 7, uma versão do Linux para empresas. A escolha deste sistema deve-se à necessidade do servidor executar o software Cadence Virtuoso, o qual os alunos de graduação e pós graduação usam para a análise e desenvolvimento de circuitos integrados. É possível utilizar o software GNU Radio para a simulação e realização de experimentos com rádio definido por software através de uma distribuição do Python chamada Anaconda, disponível para várias plataformas de sistemas operacionais. Sendo assim, mesmo que eventualmente haja a necessidade de mudar o sistema operacional, desde que suportado a distribuição Anaconda, é possível utilizar outras distribuições de Linux, tais como o popular Ubuntu, ou outros sistemas como o Windows, da Microsoft.

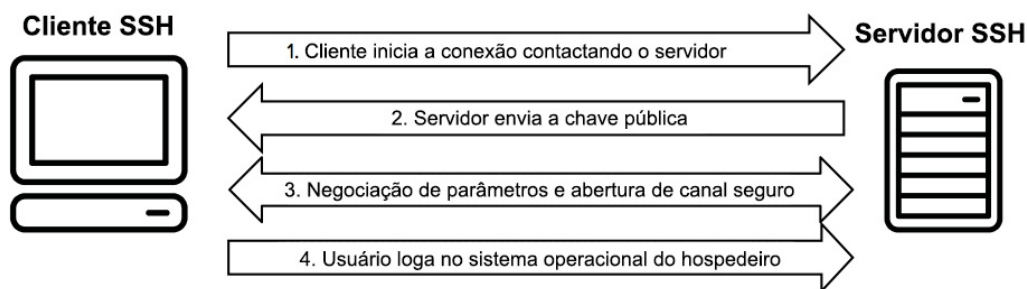
Os lançamentos do CentOS são construídos a partir de códigos-fonte *open-source* publicamente disponíveis providenciados pela *Red Hat* (CENTOS, 2022). O CentOS possui total conformidade com as políticas de redistribuição *Red Hat* e almeja ser funcionalmente compatível com o *Red Hat Enterprise Linux* (CENTOS, 2022).

3.4 SSH

O protocolo SSH utiliza como protocolo base o TCP, um protocolo orientado para conexão que provê um serviço full-duplex e ponto-a-ponto, com o lado cliente e hospedeiro trocando dados entre si para estabelecer os parâmetros de transferência de dados (KUROSE; ROSS, 2014).

Além das características herdadas do TCP, o protocolo SSH possui autenticação entre os lados transmissor e receptor e conexão criptografada de chave pública, de modo a tornar seguro o canal de comunicação (SSH, 2022). A Figura 8 mostra o funcionamento do protocolo SSH, onde o cliente inicia a conexão, o servidor responde com uma chave pública, ambos cliente e servidor negociam parâmetros de conexão (velocidade, dados de autenticação, etc) e por fim o usuário loga no sistema, de onde passa a ter controle do sistema operacional hospedeiro.

Figura 8 – Funcionamento do protocolo SSH



Fonte: SSH (2022).

Na seção seguinte, é apresentada a metodologia de acesso remoto que visa permitir o controle à distância de equipamentos de rádio definido por software, utilizando-se os materiais e tecnologias citadas anteriormente.

3.5 Metodologia de acesso remoto aos dispositivos de rádio definido por software

Nesta seção são abordados os procedimentos realizados para a implantação da metodologia que visa tornar possível o acesso remoto a equipamentos de rádio definido por software e o compartilhamento de recursos entre professores e alunos.

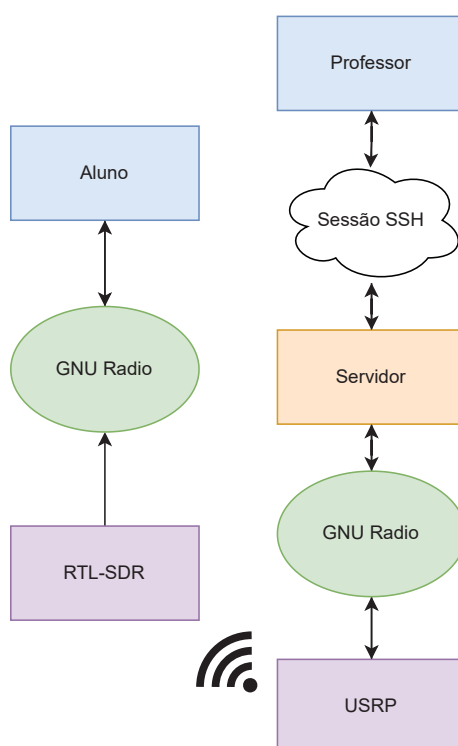
São realizados dois estudos de caso, sendo um deles envolvendo experimentos realizados dentro da rede interna da universidade com o controle dos equipamentos de radiofrequência, no qual o **USRP** é conectado ao servidor e o RTL-SDR conectado a um computador local em uma sala de aula. O outro estudo de caso aborda o acesso em rede externa à universidade em que os dois equipamentos de radiofrequência (**USRP** e RTL-SDR) são conectados a um servidor e este é acessado externamente por um endereço **IP** (do inglês, *Internet Protocol*) público através do protocolo **SSH** com um usuário criado para ser acessado pelo aluno. Antes de se iniciar cada estudo de caso, uma pré-validação é realizada testando-se a conectividade **SSH** com o servidor, para assegurar que será possível controlar remotamente os dispositivos de rádio definido por software através do GNU Radio.

No primeiro estudo de caso, é transmitido um sinal de áudio modulado em **FM**, variando-se a potência do transmissor e então verificados os efeitos na forma de onda recebida. Já no segundo caso de estudo, é transmitido um texto simples utilizando modulação digital **QPSK** (do inglês, *Quadrature Phase Shift Keying*), em que será realizado a sincronização manual dos bits recebidos e variado a potência do transmissor, observando-se os efeitos através do texto recebido e dos diagramas de constelações e diagramas de olho.

3.5.1 Acesso remoto a uma rede interna dentro da universidade

Dentro da universidade, o **USRP** é acessível via **SSH** pelo professor, posicionado em uma sala específica pronto para a transmissão dos sinais, enquanto que o aluno utiliza um RTL-SDR para captar os sinais transmitidos do **USRP** na sala de aula. O professor realiza o login no servidor por meio de **SSH** e executa o software GNU Radio, criando um fluxograma que é usado para controlar o **USRP** e enviar um sinal de áudio em loop através da modulação **FM**. Já o aluno executa em sua máquina na sala de aula uma instância do GNU Radio com o fluxograma configurado para receber os sinais de áudio pelo RTL-SDR. A **Figura 9** mostra o diagrama de blocos que exemplifica a metodologia abordada no primeiro caso estudado.

Figura 9 – Diagrama de blocos do sistema exemplificando a metodologia de acesso remoto



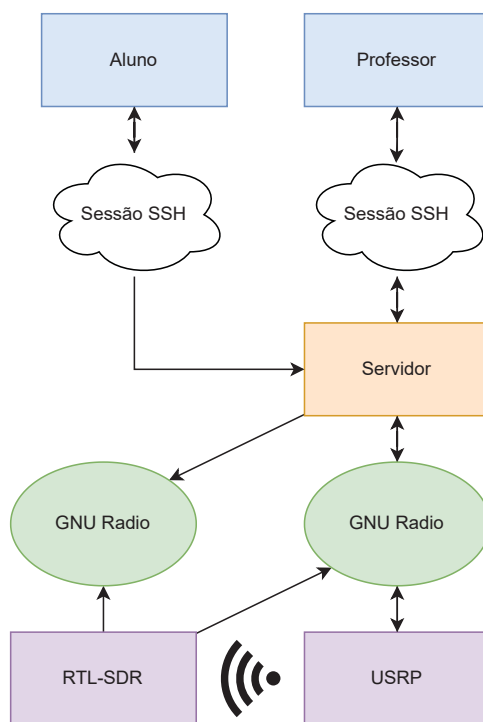
Fonte: O autor (2022).

3.5.2 Acesso remoto a uma rede externa à universidade

Em uma rede externa a da universidade, o professor monta um fluxograma no GNU Radio e salva em uma pasta do servidor que somente seu usuário tem acesso (para não ser acessado pelos alunos). O fluxograma contém uma fonte de dados de texto que é enviado para o **USRP** utilizando-se a modulação **QPSK**. Os alunos podem acessar sua própria instância do GNU Radio no servidor, criar os fluxogramas em seus ambientes de trabalho e interagir com os dados provenientes do fluxograma do professor através RTL-SDR que

está instalado no servidor e receber os sinais do **USRP**. A **Figura 10** exemplifica por meio de diagrama de blocos os passos citados anteriormente.

Figura 10 – Diagrama de blocos do sistema em que ambos aluno e professor acessam remotamente o servidor



Fonte: O autor (2022).

Com os dois estudos de caso, podem-se analisar as capacidades de aplicação da metodologia tanto para acessar remotamente equipamentos dentro da universidade, podendo ser utilizado nas salas de aula, quanto a possibilidade de acesso remoto em uma rede externa à universidade, sendo útil para os casos em que os alunos precisem realizar alguma atividade em casa ou fora da universidade.

No próximo capítulo, são feitas validações acerca da metodologia proposta, através de experimentos com modulação **FM** envolvendo a análise de arquivos de áudio transmitidos e recebidos realizando-se uma comparação entre eles, e experimentos com modulação **QPSK** através de envio de texto e verificando-se a recepção das mensagens ocorreu de forma correta.

4 VALIDAÇÃO DO ACESSO REMOTO À EQUIPAMENTOS DE RÁDIO DEFINIDO POR SOFTWARE

Neste capítulo, são mostradas as validações do acesso remoto a dispositivos de rádio controlados por software, executando-se as simulações baseadas nos fluxogramas apresentados no [Capítulo 3](#) e variando-se os parâmetros do equipamento de radiofrequência para observar os efeitos na comunicação.

Conforme o roteiro apresentado na [Seção 3.5](#), o primeiro caso a ser validado é realizado em uma rede local na universidade, onde o transmissor está instalado no servidor e o receptor está conectado a um computador localizado em uma sala de aula, com ambos servidor e computador na mesma rede interna. É, então, realizada a transmissão de um áudio em modulação [FM](#) no qual varia-se o ganho do transmissor ([USRP](#)) e é observado e comparado a forma de onda do áudio transmitido com o áudio recebido. No segundo caso a ser validado, o acesso é feito de forma externa à rede da universidade. Um arquivo de texto é enviado pelo [USRP](#) com modulação [QPSK](#) e recebido pelo RTL-SDR, variando-se o ganho do transmissor e observando-se os efeitos no diagrama de constelação, diagrama de olho e texto recebidos.

4.1 Acesso remoto a equipamentos de rádio definido por software em uma rede local na universidade

O acesso remoto do servidor em rede local aconteceu nas dependências da Universidade Federal do Paraná, na sala do grupo de pesquisa [GICS](#) (do inglês, *Group of Intergrated Circuits and Systems*). Este servidor, chamado *Enterprise*, executa o Linux CentOS 7, o qual possui um [USRP](#) N210 que é acessível via [SSH](#) pela rede e está configurado conforme descrito no [Apêndice B](#) para transmitir sinais de radiofrequência. Para controle de equipamentos de radiofrequência e simulações, utiliza-se o software GNU Radio. A [Figura 11](#) mostra o servidor com o [USRP](#) N210 conectado em sua porta *ethernet*.

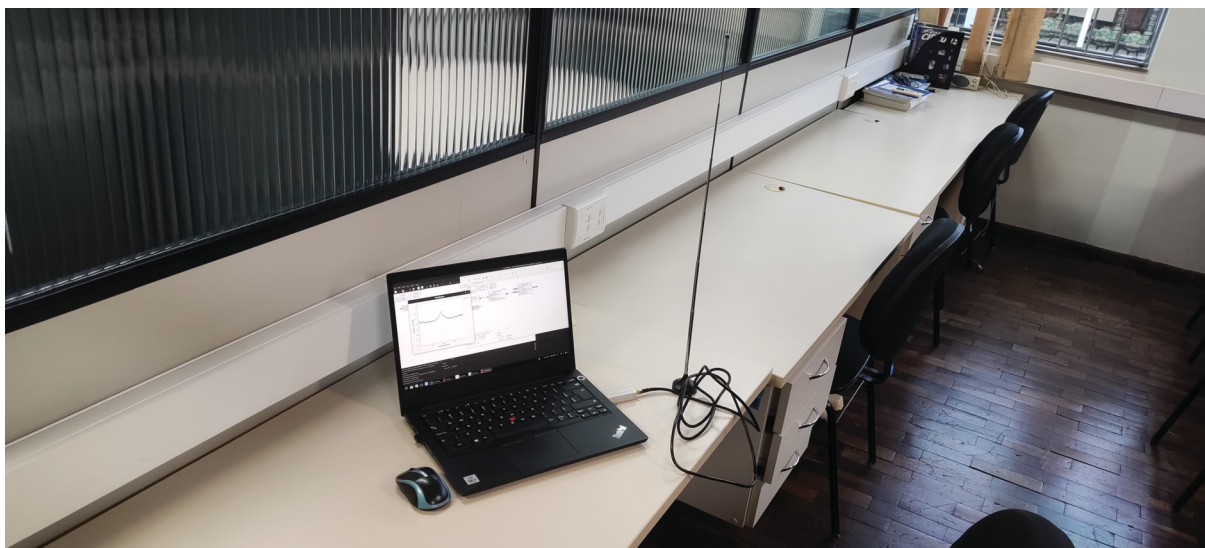
Figura 11 – Servidor com o USRP N210 conectado em sua porta *ethernet*



Fonte: O autor (2022).

Na mesma sala, mas em um computador diferente, conectou-se um RTL-SDR da Nooelec na porta **USB**, conforme mostra a [Figura 12](#), cujo notebook foi configurado da mesma maneira que o servidor seguindo-se os passos descritos na [Apêndice B](#), em que neste caso é executado o GNU Radio para receber os sinais provenientes do **USRP N210** instalado no servidor.

Figura 12 – Notebook executando o GNU Radio com o RTL-SDR conectado em sua porta USB



Fonte: O autor (2022).

Na próxima seção é realizada um teste de conectividade [SSH](#) para validar o acesso remoto ao servidor de forma a poder controlar os dispositivos de rádio definido por software de forma remota, utilizando o programa GNU Radio.

4.1.1 Teste de conectividade [SSH](#) em rede interna com o servidor

Para se testar a conectividade remota via [SSH](#) com o servidor dentro da rede da universidade foi utilizado um computador executando o sistema operacional Ubuntu e realizado o seguinte comando:

```
$ ssh -X helder@192.168.14.75
```

Onde:

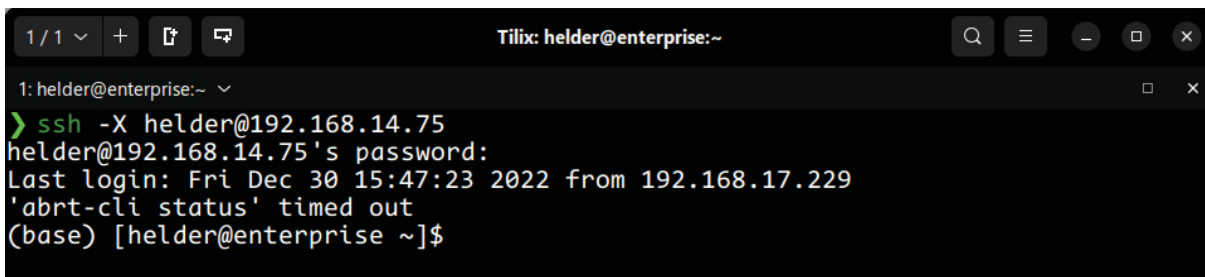
- `-X` é um comando para permitir que se possam executar aplicações gráficas pela conexão via terminal
- `helder` é o nome do usuário a ser acessado no servidor;
- `192.168.14.75` é o endereço [IP](#) do servidor, acessível pela rede interna da universidade

Ao se realizar o comando, o sistema se conecta ao servidor como mostra a [Figura 13](#). Após a conexão, testa-se a abertura do GNU Radio pelos seguintes comandos:

```
$ conda activate /home/conda_shared_envs/gnuradio
$ gnuradio-companion
```

O primeiro comando habilita o ambiente gnuradio criado pela ferramenta Anaconda e o segundo inicializa o software de processamento digital de sinais GNU Radio em modo gráfico, cujo estado de execução do programa é mostrado na [Figura 14](#). Após a execução destes dois comandos, uma janela do GNU Radio irá abrir conforme ilustra a [Figura 15](#), mostrando que a conexão e acesso ao programa no servidor foram bem-sucedidas.

Figura 13 – Comandos do terminal para realizar conexão SSH com o servidor



```
1/1 | + | [share icon] | [refresh icon] | Tmux: helder@enterprise:~ | [search icon] | [menu icon] | [resize icon] | [close icon]
1: helder@enterprise:~ | [close icon]
> ssh -X helder@192.168.14.75
helder@192.168.14.75's password:
Last login: Fri Dec 30 15:47:23 2022 from 192.168.17.229
'abrt-cli status' timed out
(base) [helder@enterprise ~]$
```

Fonte: O autor (2023).

Figura 14 – Comandos do terminal para executar o gnuradio em modo gráfico

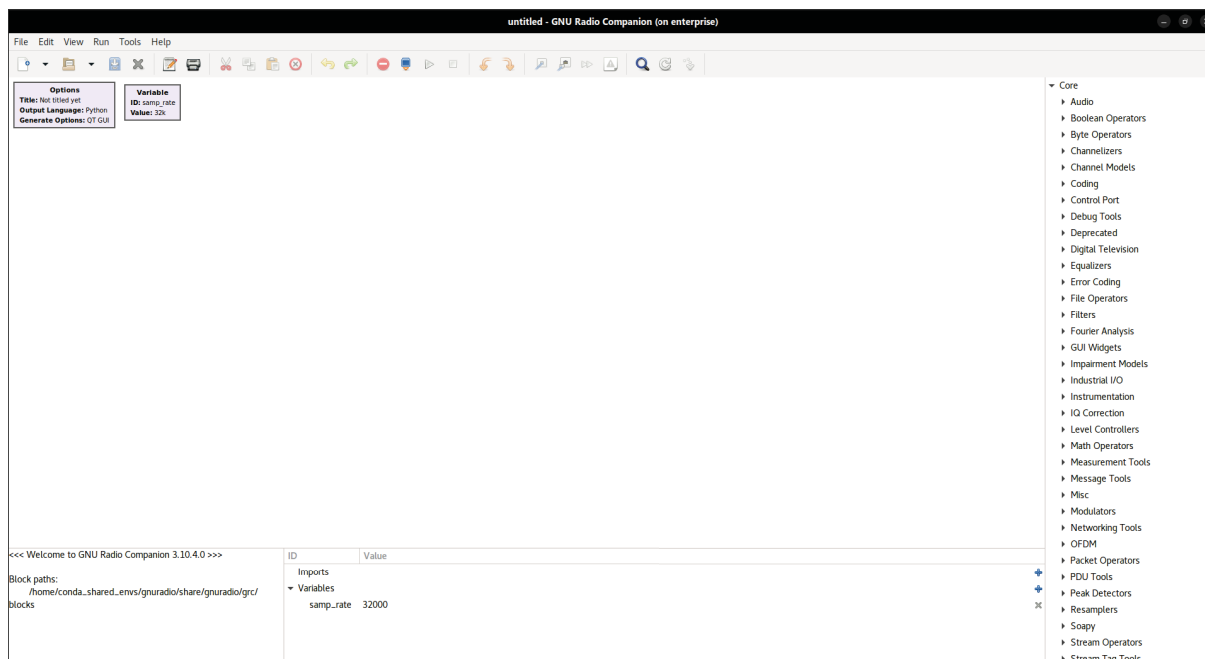


```
1/1 | + | [share icon] | [refresh icon] | Tmux: helder@enterprise:~ | [search icon] | [menu icon] | [resize icon] | [close icon]
1: helder@enterprise:~ | [close icon]
> ssh -X helder@192.168.14.75
helder@192.168.14.75's password:
Last login: Fri Dec 30 15:47:23 2022 from 192.168.17.229
'abrt-cli status' timed out
(base) [helder@enterprise ~]$ conda activate /home/conda_shared_envs/gnuradio
(/home/conda_shared_envs/gnuradio) [helder@enterprise ~]$ gnuradio-companion
>>> Warning: vocoder_codec2_decode_ps - option_attributes are for enums only, ignoring
>>> Warning: vocoder_codec2_encode_sp - option_attributes are for enums only, ignoring
<<< Welcome to GNU Radio Companion 3.10.4.0 >>>

Block paths:
  /home/conda_shared_envs/gnuradio/share/gnuradio/grc/blocks
```

Fonte: O autor (2023).

Figura 15 – Tela inicial do programa GNU Radio



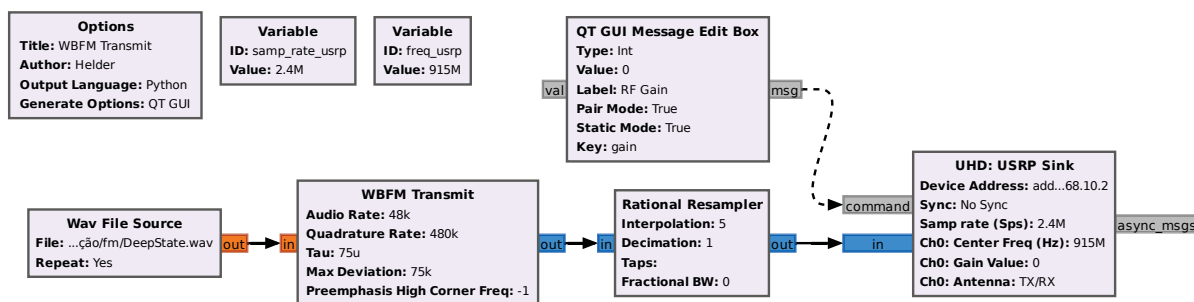
Fonte: O autor (2023).

Assim, prossegue-se com a validação através do estudos de caso da subseção a seguir, onde o **USRP** é controlado remotamente e realizado experimentos de envio de áudio com modulação angular **FM**, variando-se o ganho do **USRP**. A análise do conteúdo recebido é feita pelo RTL-SDR, onde é salvo num arquivo e comparado com o áudio original transmitido.

4.1.2 Transmissão e recepção de áudio em modulação **FM**

Para a realização da transmissão e recepção de sinais **FM**, foi utilizado o GNU Radio com o fluxograma apresentado na **Figura 16**. Um arquivo de áudio chamado *Deep State* (**JAPAN, 2020**), livre de direitos autorais e que foi convertido em um canal de áudio (*mono*) com a extensão *WAV* (Waveform Audio File Format), é usado pelo bloco *Wav File Source*. Este áudio passa pelo bloco *WBFM Transmit* e é interpolado para ser transmitido pelo **USRP** com uma taxa de amostragem de 2,4 MSps na frequência de 915 MHz, frequência escolhida com base nas bandas **ISM** (do inglês, *Industrial, Scientific and Medical*) disponíveis no Brasil que não precisam de licenciamento para seu uso e minimizando o risco de se utilizar frequências que possam interferir com serviços essenciais (tais como estações de polícia, corpo de bombeiros, etc). Para realizar a variação do ganho de transmissão em tempo real do **USRP** durante o experimento, colocou-se um bloco chamado *QT GUI Message Edit Box* conectado à entrada *command* do **USRP**, que permite a alteração do ganho enquanto é transmitido o arquivo de áudio, através da alteração do valor no campo *Value* em que o padrão é 0.

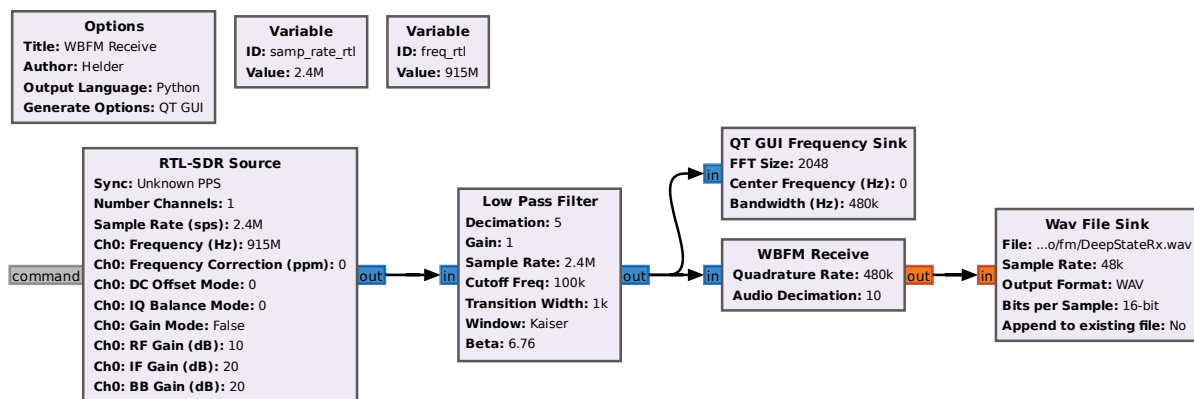
Figura 16 – Fluxograma do transmissor FM executado pelo servidor



Fonte: O autor (2022).

Para receber o áudio, no notebook em que está conectado o RTL-SDR é executado o fluxograma mostrado na Figura 17. Através do bloco *RTL-SDR Source* é recebido o sinal de áudio captado pelo rádio definido por software e então filtrado pelo bloco *Low Pass Filter* para a remoção de ruídos indesejados, decimado e então demodulado pelo bloco *WBFM Receive* e finalmente salvo o áudio resultante em um arquivo com formato *WAV*.

Figura 17 – Fluxograma do receptor FM executado pelo notebook

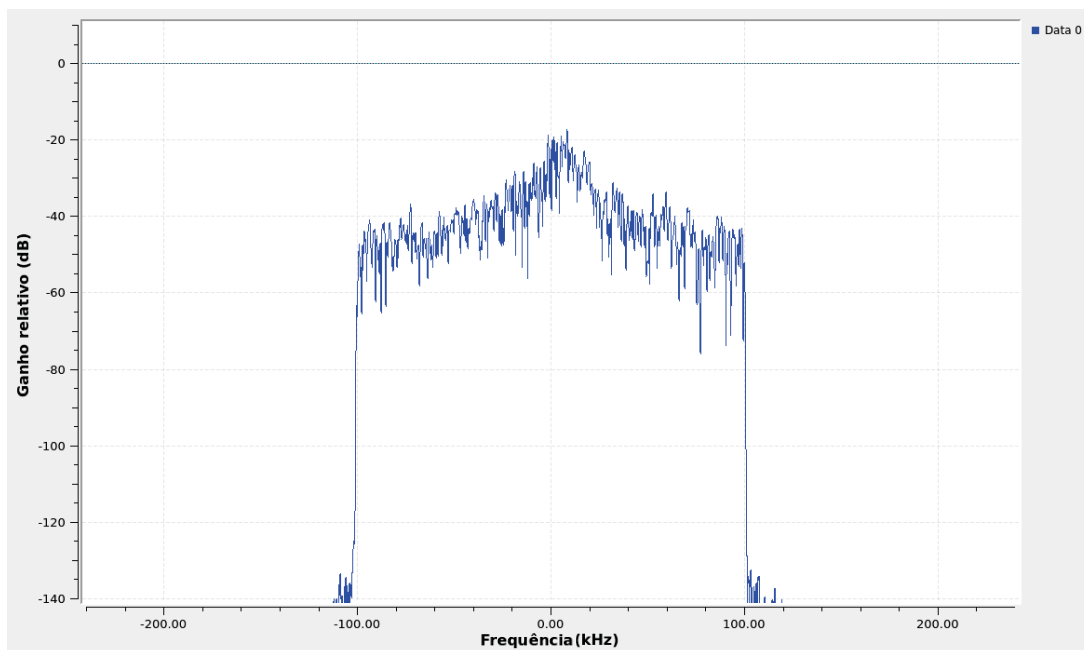


Fonte: O autor (2022).

Para realizar a comparação qualitativa dos arquivos de áudio, utilizou-se o software *DiffMaker*, em que é gerado um arquivo de áudio que contém a diferença entre o áudio original e o áudio recebido, podendo ser ouvido ou comparado graficamente os sinais de áudio em suas amplitudes, com maiores detalhes mostrados no Apêndice C. Foram analisados três experimentos, um com o transmissor configurado com ganho absoluto de 0 dB, outro com ganho de 15 dB e, finalmente, o último caso transmitindo com ganho de 30 dB.

Ao se transmitir o áudio com ganho absoluto de 0 dB no transmissor, o sinal recebido pelo RTL-SDR possui o espectro de frequência apresentado na Figura 18, em que é mostrado em termos de um ganho relativo (não calibrado), expresso em decibels.

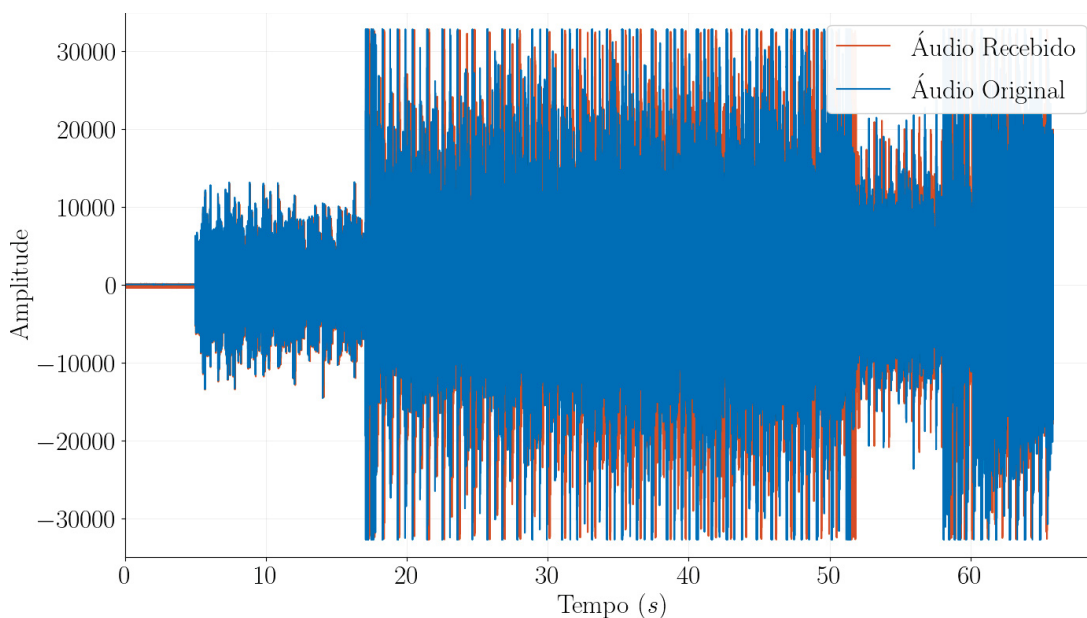
Figura 18 – Espectro de frequência do sinal recebido



Fonte: O autor (2022).

O arquivo de áudio recebido possui a forma de onda apresentada na Figura 19.

Figura 19 – Forma de onda dos sinais de áudio

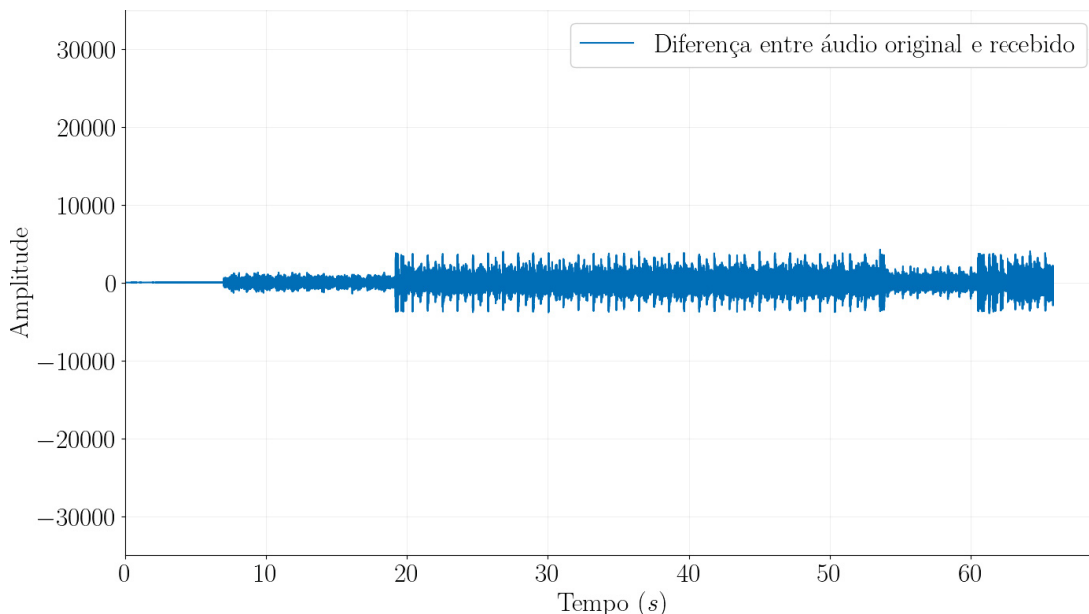


Fonte: O autor (2022).

Usou-se o programa *DiffMaker* para comparar o áudio recebido do original. Se os áudios são iguais, então a diferença entre eles gera apenas um arquivo de áudio contendo silêncio. Se existe diferença, então nos locais de divergência surge uma forma de onda não nula contendo o equivalente à subtração do áudio original com o inverso do áudio recebido,

podendo ser reproduzido por um tocador de mídia. A diferença entre o áudio transmitido e recebido para o transmissor com ganho absoluto de 0 dB é apresentada na Figura 20.

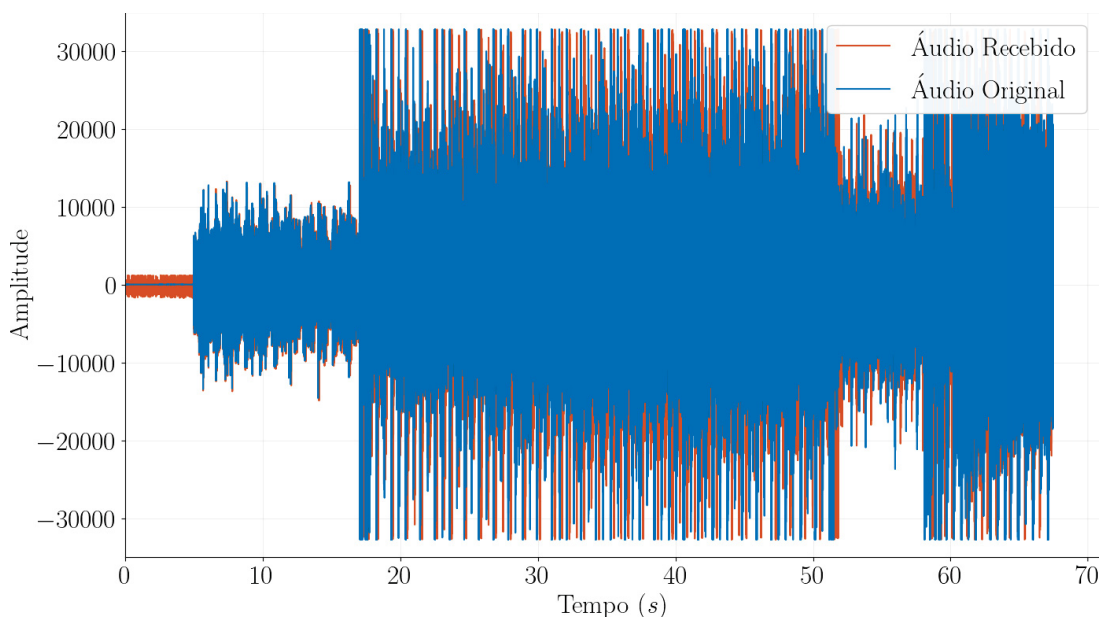
Figura 20 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR



Fonte: O autor (2022).

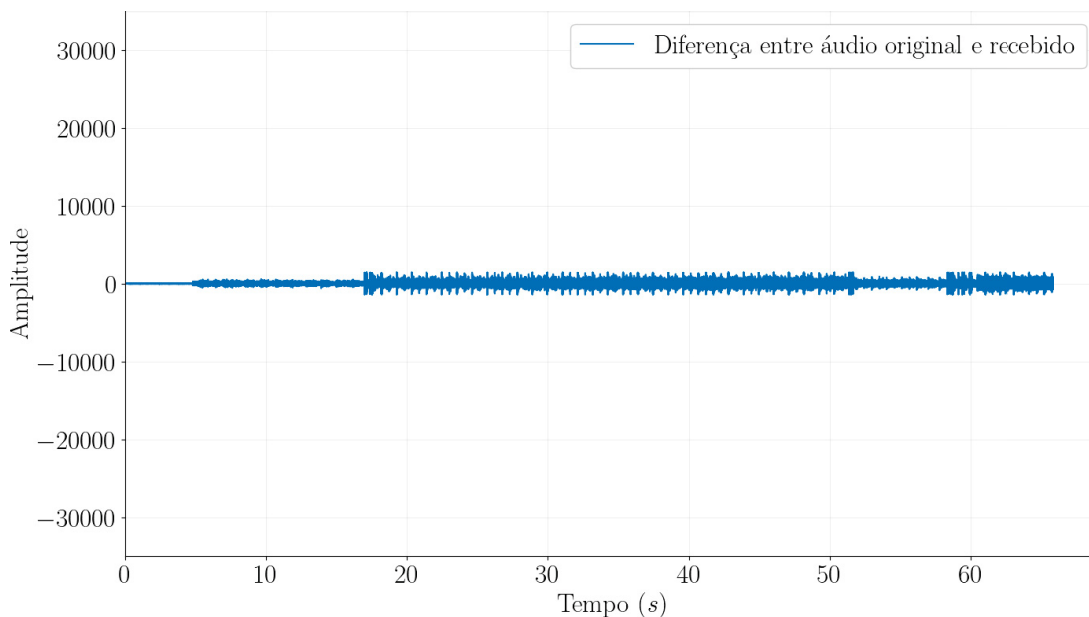
Variando-se o ganho absoluto no transmissor para 15 dB, tem-se a forma de onda apresentada na Figura 21. Comparando-se o áudio recebido com o áudio original, tem-se a forma de onda resultante apresentada na Figura 22.

Figura 21 – Forma de onda dos sinais de áudio



Fonte: O autor (2022).

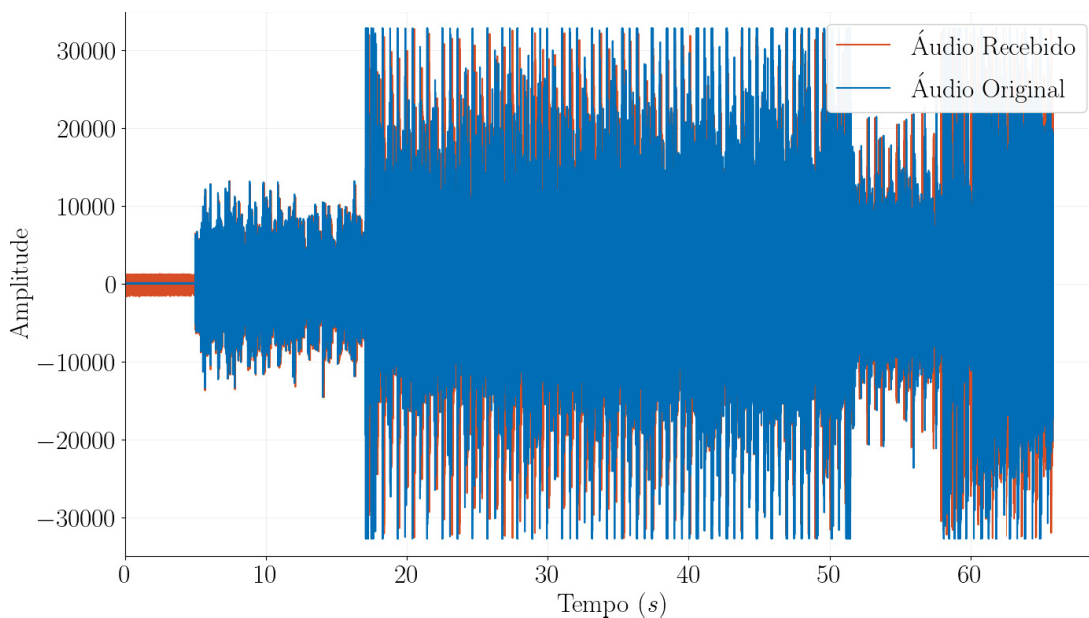
Figura 22 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR



Fonte: O autor (2022).

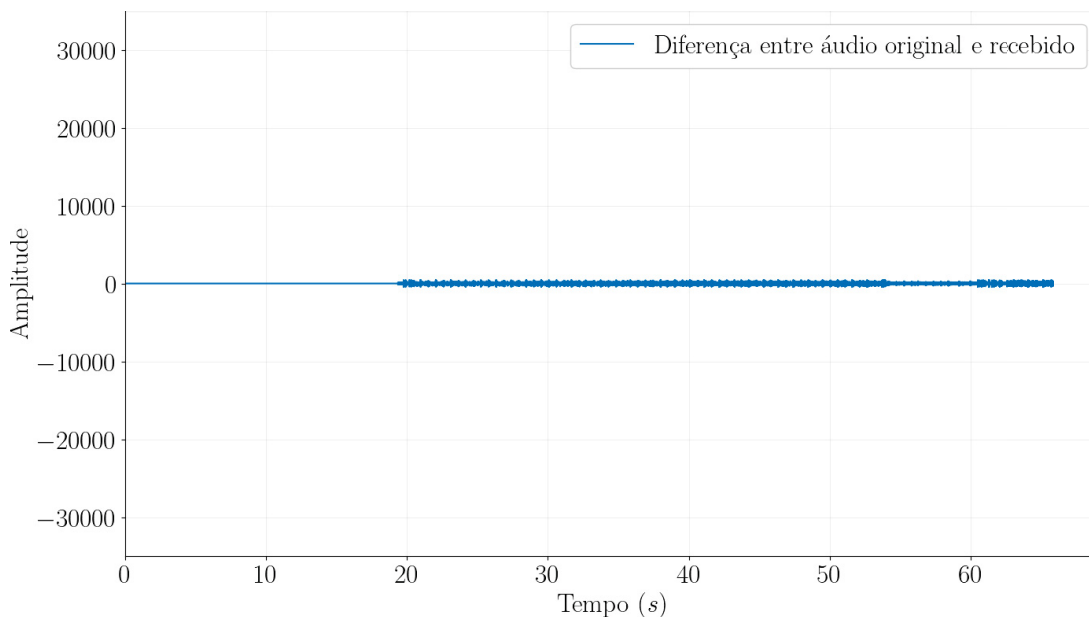
Por fim, ao configurar o ganho absoluto do transmissor para 30 dB, tem-se a forma de onda apresentada na Figura 23. A forma de onda resultante da diferença entre o áudio recebido e transmitido está ilustrado na Figura 24.

Figura 23 – Forma de onda dos sinais de áudio



Fonte: O autor (2022).

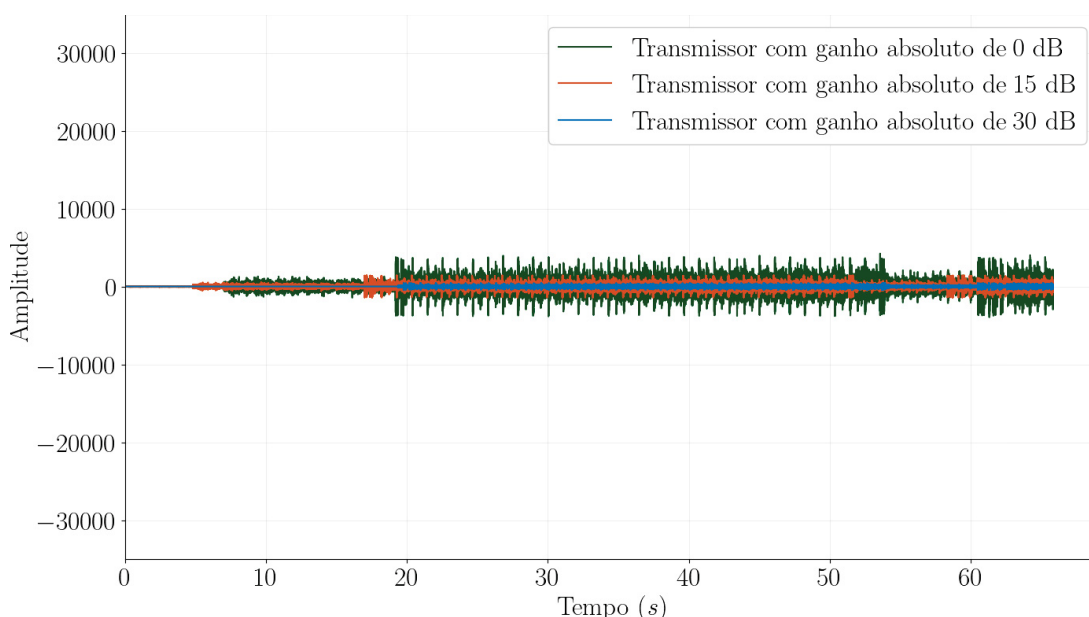
Figura 24 – Forma de onda da diferença entre o áudio original e o recebido pelo RTL-SDR



Fonte: O autor (2022).

Para melhor ilustrar a diferença do áudio recebido nos três casos de transmissão (0 dB, 15 dB e 30 dB de ganho absoluto no transmissor), juntou-se as formas de onda em uma única figura, como apresentado na [Figura 25](#). Pode-se observar que para o menor ganho, a diferença entre o áudio original e recebido pelo RTL-SDR é maior, enquanto que para um ganho maior o áudio recebido tende a ficar mais próximo ao original (ou seja, o sinal de diferença entre os áudios tende a zero).

Figura 25 – Comparativo entre as formas de onda da diferença entre o áudio original e os áudios recebidos pelo RTL-SDR



Fonte: O autor (2022).

Uma métrica para analisar a qualidade do sinal recebido é o **PSNR** (do inglês, *Peak Signal Noise Ratio*), que é a medida da relação sinal-ruído de pico, dado pela [Equação 4.1](#). Além desta métrica, o programa *Diffmaker* fornece uma medida chamada profundidade de correlação, que mostra o quanto duas amostras de áudio são diferentes entre si. Quanto mais alto o valor, mais parecido as amostras de áudio. Como exemplo, se existem dois arquivos de áudio, um sendo cópia do outro, o programa irá apontar um valor de 300 dB de profundidade de correlação. A [Tabela 3](#) mostra os valores de **PSNR** e profundidade de correlação obtidos através dos três casos de ganhos de transmissão.

$$PSNR = 20 \times \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (4.1)$$

Onde:

- *PSNR* é a relação sinal-ruído de pico;
- *MAX_I* é o valor numérico máximo admissível pela amostra;
- *MSE* é o erro médio quadrático entre duas amostras.

Tabela 3 – Comparativo entre o áudio original e os áudios recebidos pelo RTL-SDR

| Ganho do transmissor | Profundidade de Correlação | PSNR |
|----------------------|----------------------------|-------|
| 0 dB | 67,7 dB | 34 dB |
| 15 dB | 73,7 dB | 42 dB |
| 30 dB | 85,0 dB | 51 dB |

Fonte: O autor (2023).

Como pode-se observar, é possível manipular equipamentos de rádio definido por software remotamente (neste caso o **USRP**) e realizar experimentos utilizando um canal real de comunicação sem fio. Ao se variar remotamente o ganho do **USRP**, foi possível analisar no computador com o RTL-SDR os diferentes casos de transmissão através da diferença entre os áudios transmitido e recebido.

Na próxima seção, é validado o acesso remoto em uma rede externa à universidade, através do acesso ao servidor pelo protocolo **SSH** que permite o controle à distância do sistema operacional. São realizados então experimentos envolvendo a modulação passa-faixa digital **QPSK** com transmissão de um arquivo de texto em *loop*.

4.2 Acesso remoto a equipamentos de rádio definido por software em uma rede externa à universidade

O acesso remoto do servidor em uma rede externa deu-se por meio do protocolo [SSH](#) em um computador com o Linux instalado de acordo com o passo-a-passo descrito no [Apêndice B](#). No servidor estão conectados os equipamentos de rádio definido por software [USRP](#) por meio da porta *ethernet* e o RTL-SDR por meio da porta [USB](#). O professor, acessando em seu usuário do sistema uma instância do GNU Radio no servidor, executa um fluxograma que está configurado para transmitir um texto usando modulação [QPSK](#), do qual os alunos terão de montar um fluxograma para recebê-lo. Neste experimento, são simulados dois cenários, um com o canal de comunicação ideal com os dados trafegando diretamente pelo próprio servidor entre o transmissor e receptor (sem passar por equipamentos de rádio definido por software), e outro com os dados trafegando entre o [USRP](#) e o RTL-SDR. Neste último cenário, varia-se o ganho do transmissor e são observados os efeitos no receptor, além de em ambos os casos, na recepção, ser realizado um sincronismo manual dos bits dos caracteres através de atrasos inseridos na recepção destes bits.

Antes de serem realizados os experimentos, um teste de conectividade é feito para validar o funcionamento do acesso remoto via [SSH](#) a partir de um computador que está em uma rede externa à rede da universidade de forma a poder controlar os dispositivos de rádio definido por software utilizando o programa GNU Radio.

4.2.1 Teste de conectividade [SSH](#) em rede externa com o servidor

Para se testar a conectividade remota via [SSH](#) com o servidor, foi utilizado um computador executando o sistema operacional Ubuntu e realizado o seguinte comando:

```
$ ssh -X <usuario>@<ip_do_servidor> -p 2275
```

Onde:

- `-X` é um comando para permitir que se possam executar aplicações gráficas pela conexão via terminal;
- `helder` é o nome do usuário a ser acessado no servidor;
- `200.17.220.151` é o endereço [IP](#) público do servidor;
- `-p 2275` é a porta [TCP](#) para realizar a conexão [SSH](#).

Ao se realizar o comando, o sistema se conecta ao servidor como mostra a [Figura 26](#). Após a conexão, testa-se a abertura do GNU Radio pelos seguintes comandos:

```
$ conda activate /home/conda_shared_envs/gnuradio
$ gnuradio-companion
```

O primeiro comando habilita o ambiente gnuradio criado pela ferramenta Anaconda e o segundo inicializa o GNU Radio em modo gráfico, cujo estado de execução do programa é mostrado na [Figura 27](#). Após a execução destes dois comandos, uma janela do GNU Radio irá abrir da mesma forma ilustrada na [Figura 15](#), mostrando que a conexão e acesso ao programa no servidor foram bem-sucedidas.

Figura 26 – Comandos do terminal para realizar conexão SSH com o servidor

```
helder@enterprise:~
> ssh -X helder@200.17.220.151 -p 2275
helder@200.17.220.151's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Fri Dec 30 14:25:09 2022 from 177.43.130.155
'abrt-cli status' timed out
(base) [helder@enterprise ~]$
```

Fonte: O autor (2023).

Figura 27 – Comandos do terminal para executar o gnuradio em modo gráfico

```
helder@enterprise:~
> ssh -X helder@200.17.220.151 -p 2275
helder@200.17.220.151's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Fri Dec 30 14:25:09 2022 from 177.43.130.155
'abrt-cli status' timed out
(base) [helder@enterprise ~]$ conda activate /home/conda_shared_envs/gnuradio
(/home/conda_shared_envs/gnuradio) [helder@enterprise ~]$ gnuradio-companion
>>> Warning: vocoder_codec2_decode_ps - option_attributes are for enums only, ignoring
>>> Warning: vocoder_codec2_encode_sp - option_attributes are for enums only, ignoring
<<< Welcome to GNU Radio Companion 3.10.4.0 >>>

Block paths:
/home/conda_shared_envs/gnuradio/share/gnuradio/grc/blocks
```

Fonte: O autor (2023).

Assim, prossegue-se com a validação através dos dois estudos de caso das subseções seguintes, onde serão controlados os dispositivos de rádio definido por software remotamente e realizados experimentos de envio de texto utilizando modulação passa-faixa digital QPSK.

4.2.2 Transmissão e recepção de sinais QPSK

Para os experimentos a seguir, primeiro é analisado o caso em que o canal de comunicação é ideal, utilizando-se no lugar dos equipamentos de rádio definido por

software blocos de comunicação chamados *ZMQ Pub Sink* e *ZMQ Sub Source*, em que os dados são trafegados pela rede interna utilizando o protocolo [TCP](#). Se necessário, é corrigido o assincronismo dos bits dos caracteres recebidos manualmente através de blocos de atrasos na recepção. Observa-se os resultados através da análise do texto recebido, do diagrama de constelação e do diagrama de olho.

Por último, analisa-se o caso em que o canal de comunicação é real utilizando-se os equipamentos de radiofrequência [USRP](#) e [RTL-SDR](#). Varia-se o ganho do transmissor para três casos, com ganho absoluto de transmissão de 0 dB, 15 dB e 30 dB. Para cada um dos casos, são observados os efeitos através da análise do texto recebido e da observação dos diagramas de constelação e de olho. Assim como no caso do canal de comunicação ideal, se necessário são corrigidos os assincronismos de bit dos caracteres através de blocos de atrasos na recepção.

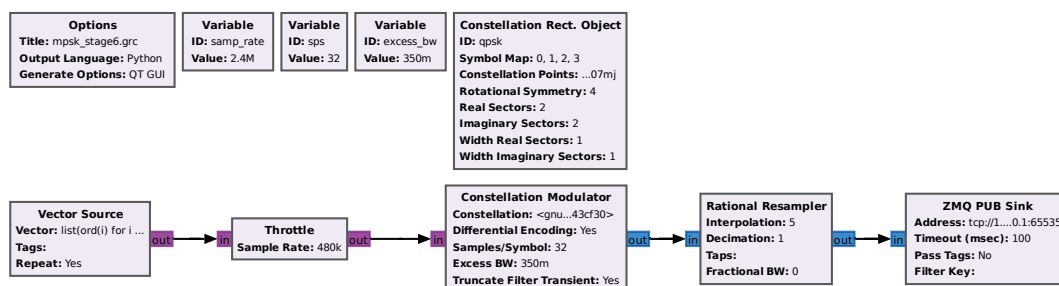
4.2.2.1 Experimento realizado através de canal de comunicação simulado ideal

Uma transmissão passa-faixa digital [QPSK](#) foi realizada conforme o fluxograma da [Figura 28](#), o qual foi adaptada do exemplo *mpsk_stage6* que vem junto com a instalação do GNU Radio, no diretório `<local_de_instalacao_gnuradio>/examples/digital`. O fluxograma é executado no servidor através do usuário *professor*. Um texto contendo a palavra *TESTE_QPSK_*, gerado continuamente em *loop*, é modulado com constelação [QPSK](#) através do bloco *Constellation Modulator*. Os pontos de constelação, mapeados para os símbolos 0, 1, 2 e 3 são:

- Símbolo 0: $0.707 + 0.707j$.
- Símbolo 1: $-0.707 + 0.707j$.
- Símbolo 2: $-0.707 - 0.707j$.
- Símbolo 3: $0.707 - 0.707j$.

Após modulada a informação, esta é então transmitida pelo bloco *ZMQ PUB Sink* no endereço interno do servidor, utilizando uma porta [TCP](#) 65535 (escolhido arbitrariamente) para realizar a conexão com um receptor.

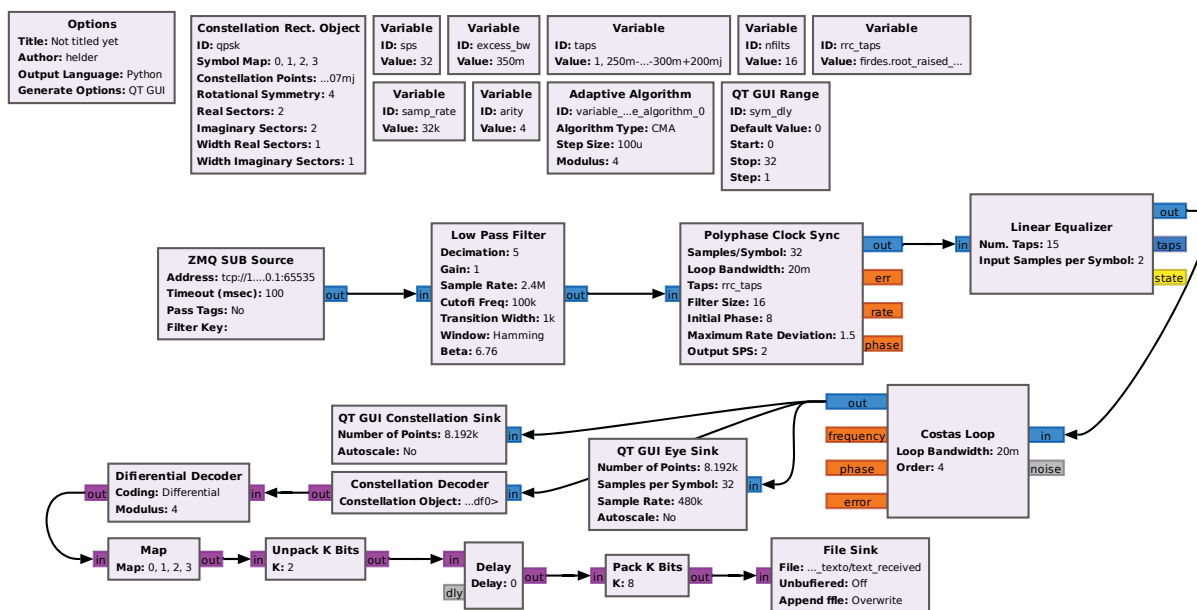
Figura 28 – Fluxograma do transmissor QPSK simulado



Fonte: O autor (2022).

Em uma outra instância do GNU Radio com o usuário *aluno*, é executado o fluxograma da Figura 29, em que o sinal de radiofrequência é captado pelo bloco *ZMQ SUB Source* na porta *TCP* especificada no transmissor.

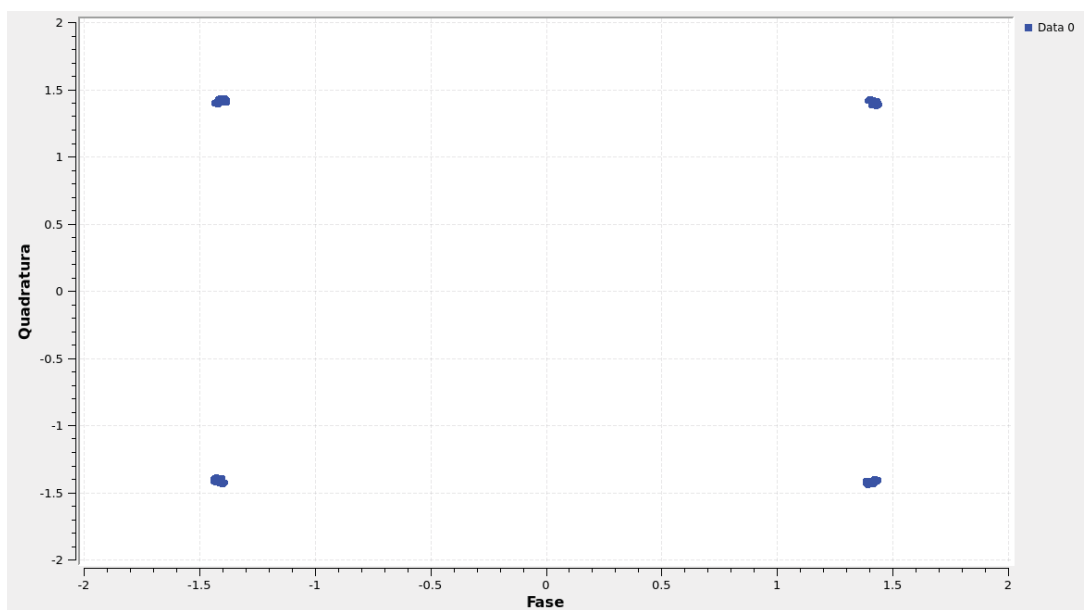
Figura 29 – Fluxograma do receptor QPSK simulado



Fonte: O autor (2022).

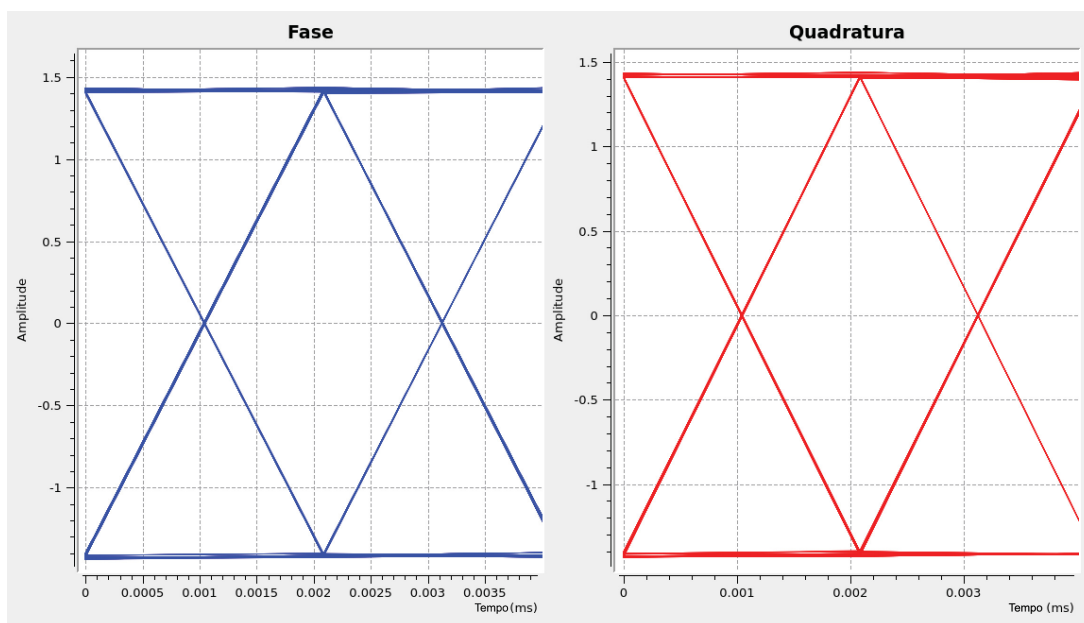
O sinal é processado por um filtro passa baixa, passa pelo bloco *Polyphase Clock Sync* responsável por sincronizar no tempo os sinais, é equalizado pelo bloco *Linear Equalizer* e então sincronizado para QPSK através do bloco *Costas Loop*. A partir deste bloco, é feita a extração da mensagem e sincronizado os bits dos caracteres manualmente por meio do bloco *Delay* e salvo num arquivo. A Figura 30 mostra o diagrama de constelação do sinal QPSK em que mostra os dados recebidos nas quatro constelações, e a Figura 31 mostra o diagrama de olho do sinal em que é possível observar que a recepção possui muito pouco ou não possui efeitos indesejados de ruídos, pois o olho está bem aberto, que no instante próximo de 2 microssegundos o sistema consegue discernir se o sinal é alto ou baixo.

Figura 30 – Diagrama de constelação QPSK em canal de comunicação simulado



Fonte: O autor (2022).

Figura 31 – Diagrama de olho com a modulação QPSK em canal de comunicação simulado



Fonte: O autor (2022).

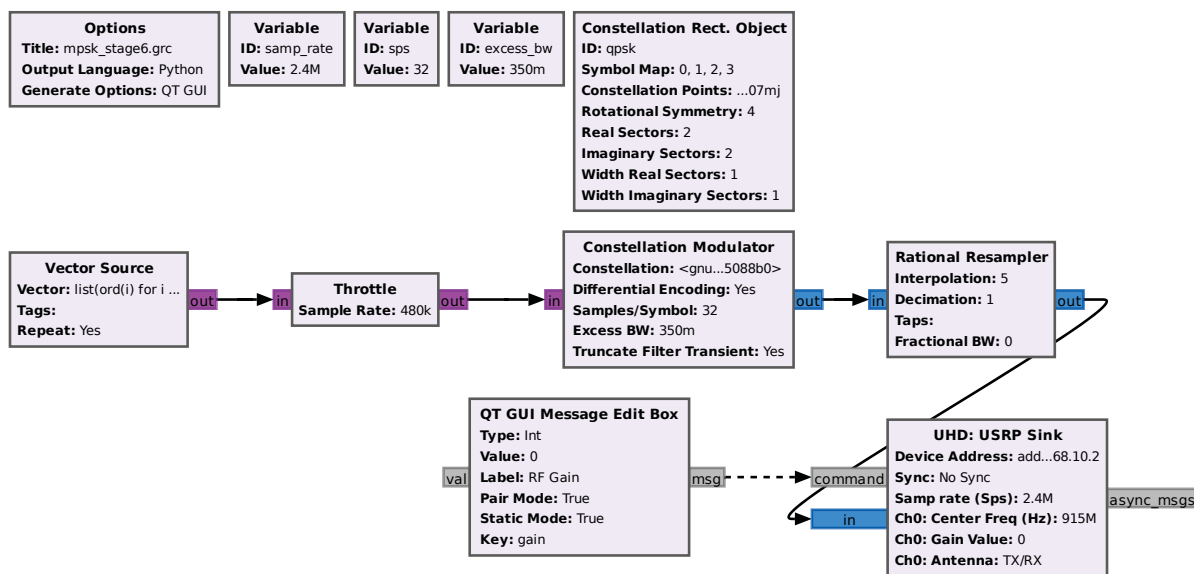
Mesmo com o experimento sendo executado em um canal ideal, por falta de sincronismo dos bits dos caracteres recebidos (visto que o receptor executa em um momento diferente do transmissor), o texto foi recebido de forma errônea, como mostra a [Figura 32](#), indicando que houve algum tipo de deslocamento destes bits que compõem a mensagem original. Assim, ajustou-se o atraso dos bits dos caracteres durante a simulação de um em um através do bloco *Delay* até que na saída fosse visto o texto correto ao se analisar o

Na próxima subseção, os blocos *ZMQ Pub Sink* e *ZMQ Sub Source* são substituídos pelos blocos *UHD: USRP Sink* e *RTL-SDR Source*, respectivamente. São analisados três casos, nos quais o **USRP** transmite com ganho absoluto de 0 dB, 15 dB e 30 dB. Por fim, é feita uma análise através de diagramas de constelação, diagramas de olho e observação do texto recebido para inferir os efeitos de um canal real de comunicação sem fio com equipamentos de rádio definido por software.

4.2.2.2 Experimento realizado através de canal de comunicação sem fio utilizando equipamentos de rádio definido por software

O fluxograma da **Figura 34** mostra uma transmissão passa-faixa digital **QPSK** que é executada no servidor através do usuário *professor*. Um texto contendo a palavra *TESTE_QPSK_*, gerado continuamente em *loop*, é modulado com constelação **QPSK** através do bloco *Constellation Modulator* e então transmitida pelo **USRP** na frequência de 915 MHz obedecendo a faixa de frequências **ISM** definidas no Brasil. Para realizar a variação do ganho de transmissão em tempo real do **USRP** durante o experimento, colocou-se um bloco chamado *QT GUI Message Edit Box* conectado à entrada *command* do **USRP**.

Figura 34 – Fluxograma do transmissor **QPSK**

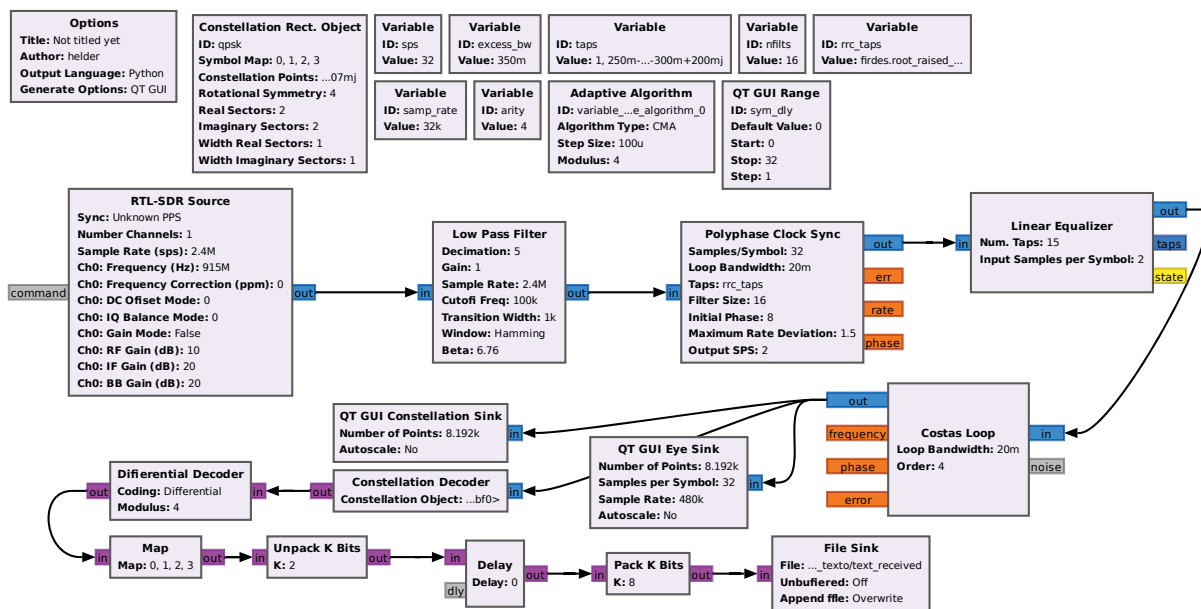


Fonte: O autor (2022).

Em uma outra instância do GNU Radio com o usuário *aluno*, é executado o fluxograma da **Figura 35**, em que o sinal de radiofrequência é captado pelo **RTL-SDR**. O sinal então é processado por filtros e blocos de equalização e sincronismo de tempo (os mesmos da **Subseção 4.2.2.1**). A **Figura 36** mostra o diagrama de constelação do sinal **QPSK** para o transmissor configurado com ganho absoluto de 0 dB e a **Figura 37** mostra o diagrama de olho do sinal recebido. É observado que os dados espalham-se para mais

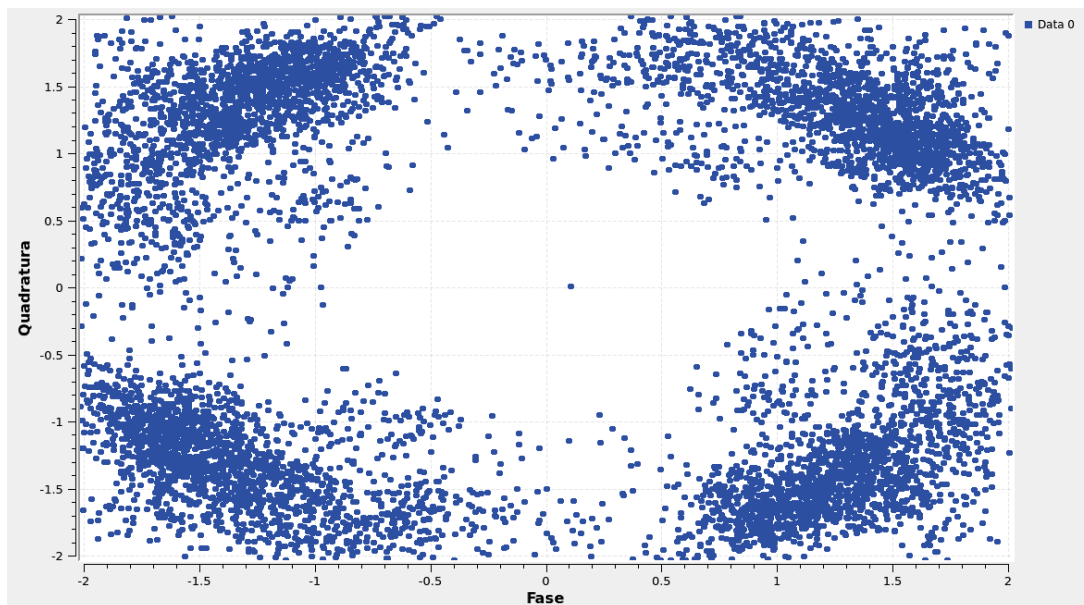
longe dos 4 pontos da constelação, pois agora o sistema não opera em um canal ideal de comunicação, que como consequência o diagrama de olho fecha-se mais e os dados recebidos passam a ter um maior número de erros.

Figura 35 – Fluxograma do receptor QPSK



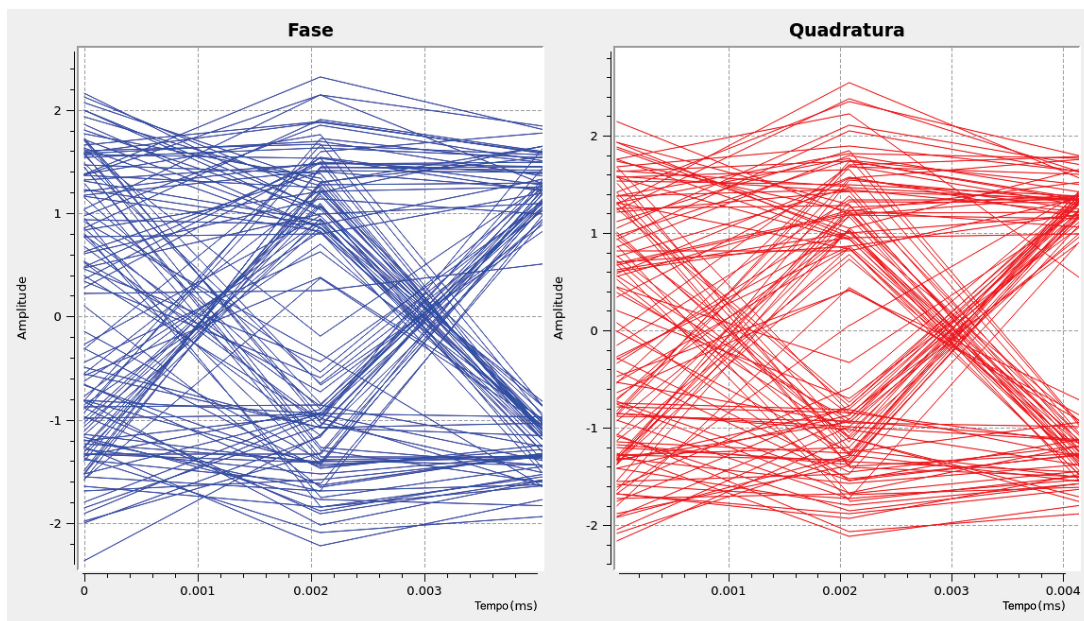
Fonte: O autor (2022).

Figura 36 – Diagrama de constelação



Fonte: O autor (2022).

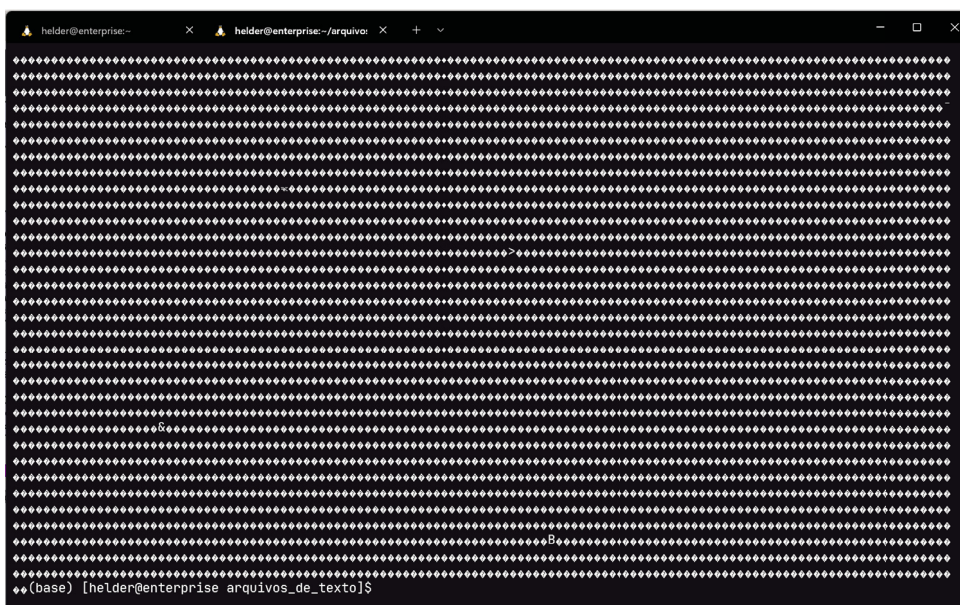
Figura 37 – Diagrama de olho



Fonte: O autor (2022).

Da mesma forma que na [Subseção 4.2.2.1](#), é preciso sincronizar os bits dos caracteres recebidos, pois o receptor desconhece onde começa cada conjunto de bits dos caracteres da mensagem recebida, tendo como resultado um arquivo de texto ilegível conforme ilustra a [Figura 38](#). Assim, faz-se o sincronismo variando o valor do bloco *Delay* até que o texto recebido tenha legibilidade como mostrado na [Figura 39](#).

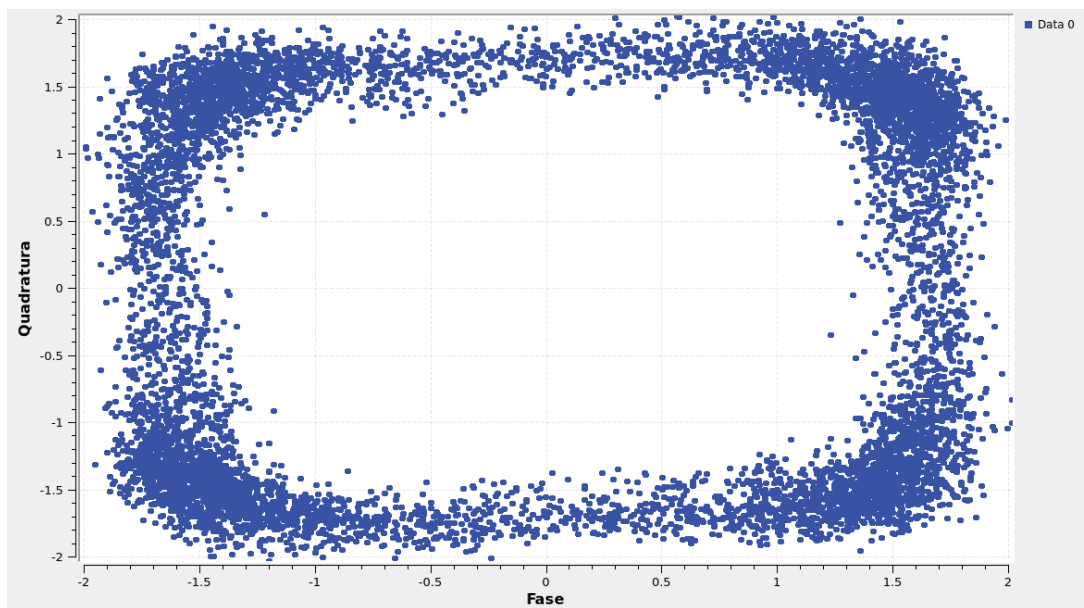
Figura 38 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres não estão sincronizados



Fonte: O autor (2022).

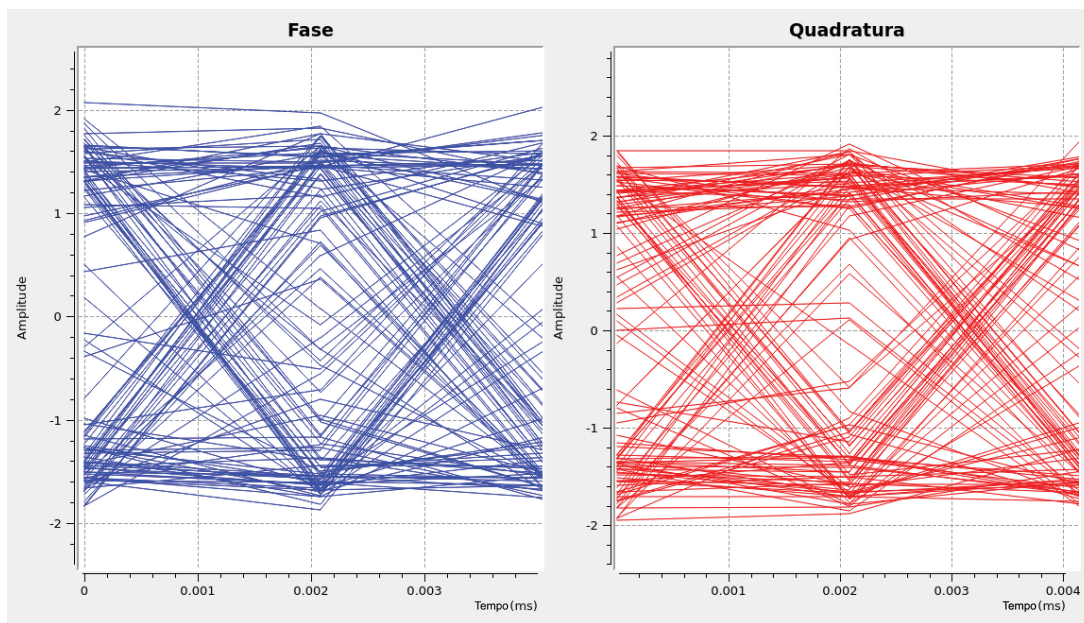
Figura 45.

Figura 43 – Diagrama de constelação



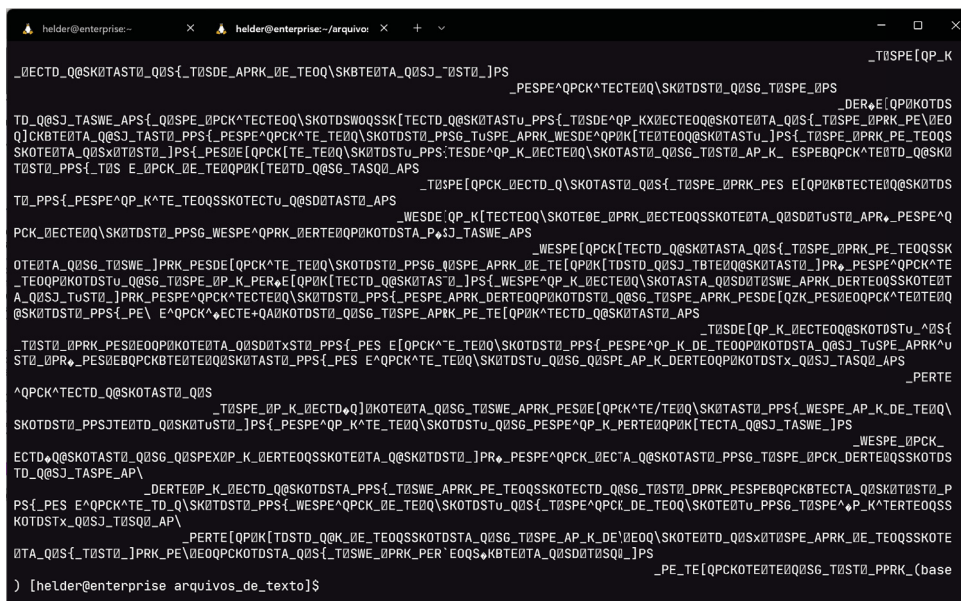
Fonte: O autor (2022).

Figura 44 – Diagrama de olho



Fonte: O autor (2022).

Figura 45 – Conteúdo do arquivo de texto recebido quando os bits dos caracteres estão sincronizados



Fonte: O autor (2022).

Dos experimentos realizados, não somente o canal de comunicação sem fio tem impacto quando se usam dispositivos de rádio definido por software, mas os próprios dispositivos possuem limitações quanto às capacidades de transmissão e recepção de dados. Diferentemente do cenário ideal simulado, usar dispositivos de radiofrequência permite com que se tenha uma maior fidelidade na hora de se considerar os possíveis problemas que possam surgir durante um cenário de aplicação de alguma tecnologia.

Neste capítulo foram abordados e validados o acesso remoto dos equipamentos de rádio definido por software pela rede interna da universidade e por uma rede externa através do acesso SSH ao IP público fornecido pelo servidor. Mesmo operando-se à distância, é possível controlar os dispositivos de radiofrequência e realizar experimentos observando-se os efeitos físicos do ambiente. Por fim, no próximo capítulo conclui-se o trabalho recapitulando-se o que foi abordado e com sugestões de melhoria na metodologia de acesso remoto dos equipamentos de rádio definido por software.

5 CONCLUSÃO E SUGESTÕES

Este trabalho teve como objetivo e contribuição o acesso de dispositivos de rádio definido por software de forma remota, permitindo o compartilhamento de recursos e o uso por estudantes e professores no âmbito das universidades. Foram realizadas configurações abordando diferentes cenários de transmissão e recepção de sinais, com a manipulação dos dispositivos através do GNU Radio acessado via protocolo [SSH](#), além da utilização do GNU Radio para a manipulação dos sinais.

Com a utilização do Anaconda, uma distribuição do Python voltada à computação científica, a instalação do GNU Radio independe do sistema operacional usado, pois o Anaconda é um gerenciador de pacotes multi-plataforma. Além disto, o gerenciador permite a criação de ambientes auto-contidos, onde é possível instalar várias instâncias do GNU Radio em cada ambiente sem que uma instalação interfira com a outra, ou com o próprio sistema operacional, apresentando uma vantagem para atender demandas de diferentes usuários com instalações customizadas.

Ao se utilizar o protocolo [SSH](#), é possível executar programas com interface gráfica, mesmo que o acesso ao sistema seja por um terminal, pois o [SSH](#) oferece suporte a aplicativos gráficos. O sistema é hospedado em um servidor que está instalado nas dependências da Universidade Federal do Paraná, Centro Politécnico, no departamento de Engenharia Elétrica ([DELT](#)), em que os professores podem utilizá-lo para implantar experimentos em aulas, seja presencialmente nas salas de aula ou através de exercícios para os alunos fazerem desde suas residências.

Para testar a metodologia proposta, foram realizados dois estudos de caso. No primeiro, um dispositivo [USRP](#) foi utilizado como transmissor conectado a um servidor, com um receptor RTL-SDR posicionado em uma outra máquina na mesma rede local da universidade. Através do software GNU Radio, no lado do transmissor realizou-se a comunicação remota com o protocolo [SSH](#) para transmitir sinais de áudio pelo [USRP](#), sendo estes recebidos pelo RTL-SDR com a outra máquina executando localmente uma instância do programa GNU Radio, em que são analisados os efeitos do canal de comunicação comparando-se o sinal de áudio recebido com o transmitido pelo programa *DiffMaker*, para diferentes ganhos de transmissão. Este caso mostra que o método é eficaz para a elaboração de um ambiente que faz o uso de um equipamento de custo elevado ([USRP](#)), acessível pela rede, e que nas salas de aula os alunos consigam captar os sinais através de dispositivos RTL-SDR que são de baixo custo.

No segundo estudo de caso, foram conectados ambos [USRP](#) e RTL-SDR no servidor, com o acesso ao GNU Radio realizado por uma rede externa à universidade utilizando o

protocolo [SSH](#), em que foram mostrados os efeitos do canal de comunicação no ar através de experimentos com transmissão de texto utilizando a modulação [QPSK](#). Por meio de diagramas de constelação, diagramas de olho e análise do texto recebido, foi possível ver diferentes efeitos quando variado o ganho absoluto da transmissão, com todo o controle do sistema e obtenção dos resultados acontecendo de maneira remota. Assim, o método se mostra apto para a aplicação em exercícios dos quais os alunos podem resolver em casa, ou em criação de conteúdos voltados para ensino à distância ([EAD](#)), com o problema do alto custo resolvido através de compartilhamento do [USRP](#) e [RTL-SDR](#) remotamente, podendo ser expandido para usar outros dispositivos de radiofrequência.

Por fim, os dois estudos de caso mostram que a metodologia de acesso remoto de equipamentos de rádio definido por software permite cortar custos com a otimização da utilização dos recursos, além de adicionar flexibilidade na margem para criação de disciplinas 100% à distância com experimentos práticos. Sugere-se, no entanto, que para a realização de experimentos, dispensando a necessidade de operar os dispositivos presencialmente e como bônus abrindo realização de experimentos o usuário tenha uma boa conexão à internet, pois podem ser enfrentadas dificuldades em caso de conexão lenta ou quedas. Além disto, a performance do sistema é limitada ao computador hospedeiro, que neste trabalho fora utilizado o servidor *Enterprise* do [GICS](#), sendo que se outros usuários sobrecarregarem o uso de memória ou de processador, os experimentos realizados no GNU Radio podem ficar comprometidos.

Para trabalhos futuros, sugere-se a implementação de acesso seletivo ao transmissor pelos alunos através do login seletivo pelo professor, em que este irá definir as condições de uso (tais como horários de acesso e as limitações das configurações do equipamento transmissor); o uso de blocos customizados pelos alunos para comparação com blocos prontos fornecidos pelo GNU Radio, e; o acesso [TCP](#) diretamente no GNU Radio através de *sockets* entre os hardwares do servidor e os computadores que estejam externos à rede, com estudos e validações da máxima largura de banda que possa trafegar pela rede externa, pois os dados partem da máquina local do usuário percorrendo um caminho maior até chegar ao servidor, tendo impacto direto na performance do sistema com fatores como latência e velocidade de conexão podendo afetar negativamente a experiência de utilização (no caso do [SSH](#) a largura de banda não interfere nos experimentos pois os dados que estão no GNU Radio trafegam pela rede interna do hospedeiro).

Referências

- ALMALEH, R. H. Avaliação do Rádio Definido por Software LimeSDR utilizando a plataforma GNURadio. Alegre, Porto, 2009. Citado na página 17.
- BALL, D. P. W. a. E. A. Highly portable, low-cost sdr instrument for rf propagation studies. *IEEE Transactions on Instrumentation and Measurement*, Institute of Electrical and Electronics Engineers Inc., v. 69, p. 5446–5457, 8 2020. ISSN 15579662. Citado na página 16.
- BAUER, M.; SANCHEZ, L.; SONG, J. Iot-enabled smart cities: Evolution and outlook. *Sensors*, MDPI AG, v. 21, 7 2021. ISSN 14248220. Citado na página 15.
- BODDU, R. D. Experimental validation of spectrum sensing on various transceivers using software defined radio. *Wireless Personal Communications*, Springer New York LLC, v. 109, p. 1615–1630, 12 2019. ISSN 1572834X. Citado na página 15.
- BRAGA, D. D. DESENVOLVIMENTO DE SOFTWARE PARA ANÁLISE GRÁFICA EM TEMPO REAL DE SISTEMAS DE COMUNICAÇÃO DIGITAL DEFINIDOS POR SOFTWARE. 2011. Citado na página 17.
- CÂNDIDO, J. d. S. *Implementações práticas de DSP e RF com GNURadio e HackRF One*. 95 p. Monografia (Trabalho de conclusão de curso) — Universidade Federal de Uberlândia, Porto Alegre, Dezembro 2020. Citado na página 17.
- CENTOS. *Frequently Asked Questions about CentOS in general*. 2022. Disponível em: <https://wiki.centos.org/FAQ/General#What_is_CentOS_Linux.3F>. Acesso em: 22 out 2022. Citado na página 28.
- DINIZ, P. C. A. Utilização de rádio definido por software para análise de sinais aplicados ao ensino em engenharia. 2013. Citado na página 17.
- DINIZ, P. C. A. *Utilização de rádio definido por software e metodologias ativas no ensino em engenharia elétrica*. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal de Uberlândia, 2019. Disponível em: <<https://repositorio.ufu.br/handle/123456789/26912>>. Citado na página 17.
- DIOUF, C. et al. A usrp-based testbed for wideband ranging and positioning signal acquisition. *IEEE Transactions on Instrumentation and Measurement*, Institute of Electrical and Electronics Engineers Inc., v. 70, 2021. ISSN 15579662. Citado na página 16.
- ELEKTROLOGI. *RTL-SDR Dongle V3*. 2022. Disponível em: <https://elektrologi.iptek.web.id/wp-content/uploads/2018/06/rtl-sdr-v3-casing-IMG_1270.jpg>. Acesso em: 22 dez 2022. Citado na página 26.
- FRANCISCO, J. et al. Raspbed: a low cost experimentation platform based on raspberry pi for wireless communication studies. 2019. Citado na página 16.

- FUHRMANN, T.; FARMBAUER, M.; NIEMETZ, M. Integrating GNU Radio into a Virtual Course about Communication Systems. In: *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. Opatija, Croatia: IEEE, 2021. p. 1565–1570. ISBN 978-953-233-101-1. Disponível em: <<https://ieeexplore.ieee.org/document/9596700/>>. Citado na página 17.
- GAUTAM, S. et al. Experimental evaluation of rf waveform designs for wireless power transfer using software defined radio. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 9, p. 132609–132622, 2021. ISSN 21693536. Citado na página 16.
- GOWDA, C. P. M. et al. Design implementation of fm transceiver using raspberry pi and sdr. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 156–161. ISBN 9781665420877. Citado na página 16.
- HELBET, R. et al. Low-cost sensor based on SDR platforms for TETRA signals monitoring. *Sensors*, v. 21, n. 9, maio 2021. ISSN 14248220. Publisher: MDPI AG. Citado na página 16.
- INSTITUTE, E. T. S. *Terrestrial Trunked Radio (TETRA); Conformance testing specification; Part 1: Radio*. 2000. Disponível em: <https://www.etsi.org/deliver/etsi_ts/100300_100399/10039401/02.03.01_60/ts_10039401v020301p.pdf>. Acesso em: 20 fev 2023. Citado na página 17.
- JANSON, J. Radio Definido por Software: Estudo e Realização de Teste com uma Plataforma Livre. 2012. Citado na página 24.
- JAPAN, V. i. *Vans in Japan / Deep State*. 2020. Disponível em: <<https://www.youtube.com/watch?v=1mmsPXWGKuk>>. Acesso em: 30 dez 2022. Citado 2 vezes nas páginas 36 e 78.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 6. ed. São Paulo: Pearson Education do Brasil, 2014. 65–70 p. Citado na página 28.
- MESHAM, S.; KOLHARE, N. The advent software defined radio: Fm receiver with rtl sdr and gnu radio. In: *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. [S.l.: s.n.], 2019. p. 230–235. Citado na página 16.
- PEROTONI, M. B.; SILVA, F. A. A.; VIEIRA, M. S. Radio-propagation measurement based on a low-cost software defined radio. *Progress In Electromagnetics Research Letters*, v. 101, p. 1–10, 2021. Citado na página 16.
- PROJECT, G. R. *About GNU Radio*. 2022. Disponível em: <<https://www.gnuradio.org/about/>>. Acesso em: 22 out 2022. Citado na página 27.
- REHMAN, M. et al. Rf sensing based breathing patterns detection leveraging usrp devices. *Sensors*, MDPI AG, v. 21, 6 2021. ISSN 14248220. Citado na página 16.
- RESEARCH, E. *USRP N200/N210 Networked Series*. 2022. Disponível em: <https://www.ettus.com/wp-content/uploads/2019/01/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf>. Acesso em: 22 out 2022. Citado na página 25.
- RESEARCH, E. *USRP N210*. 2022. Disponível em: <<https://www.ettus.com/all-products/un210-kit/>>. Acesso em: 22 out 2022. Citado na página 24.

- RESEARCH, E. *USRP N210 Software Defined Radio (SDR) - Ettus Research*. 2022. Disponível em: <https://www.ettus.com/wp-content/uploads/2019/01/N210_Top-Large_2.jpg>. Acesso em: 22 out 2022. Citado na página 25.
- RHARRAS, R. S. A. E.; SABER, H. K. A. M. Raspberry Pi and RTL-SDR for Spectrum Sensing based on FM Real Signals. 2018. Citado na página 16.
- RTL-SDR. *ABOUT RTL-SDR*. 2022. Disponível em: <<https://www.rtl-sdr.com/about-rtl-sdr/>>. Acesso em: 22 out 2022. Citado 2 vezes nas páginas 26 e 27.
- RTL-SDR. *BUY RTL-SDR DONGLES (RTL2832U)*. 2022. Disponível em: <<https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/>>. Acesso em: 22 out 2022. Citado na página 27.
- SALES, H. E.; MARIANO, A. A.; FERNANDEZ, E. Plataforma baseada em SDR com Baixo Custo e Comunicação Remota via TCP. In: *Anais do XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Sociedade Brasileira de Telecomunicações, 2022. Disponível em: <<https://biblioteca.sbtr.org.br/articles/3674>>. Citado 2 vezes nas páginas 16 e 18.
- SSH. *SSH (Secure Shell) Home Page*. 2022. Disponível em: <<https://www.ssh.com/academy/ssh>>. Acesso em: 22 out 2022. Citado 2 vezes nas páginas 28 e 29.
- STORCK, C. R.; DUARTE-FIGUEIREDO, F. A 5g v2x ecosystem providing internet of vehicles. *Sensors (Switzerland)*, MDPI AG, v. 19, 2 2019. ISSN 14248220. Citado na página 15.
- TANG, Y. et al. A survey on the 5g network and its impact on agriculture: Challenges and opportunities. *Computers and Electronics in Agriculture*, Elsevier B.V., v. 180, 1 2021. ISSN 01681699. Citado na página 15.
- TOMAR, V. S.; BHATIA, V. Low Cost and Power Software Defined Radio Using Raspberry Pi for Disaster Effected Regions. In: *Procedia Computer Science*. [S.l.]: Elsevier, 2015. v. 58, p. 401–407. ISSN: 18770509. Citado na página 16.
- VARGA, P. et al. 5g support for industrial iot applications – challenges, solutions, and research gaps. *Sensors (Switzerland)*, MDPI AG, v. 20, 2 2020. ISSN 14248220. Citado na página 15.
- WASLO, B. Detecting Changes in Audio Signals by Digital Differencing. 2008. Citado na página 77.

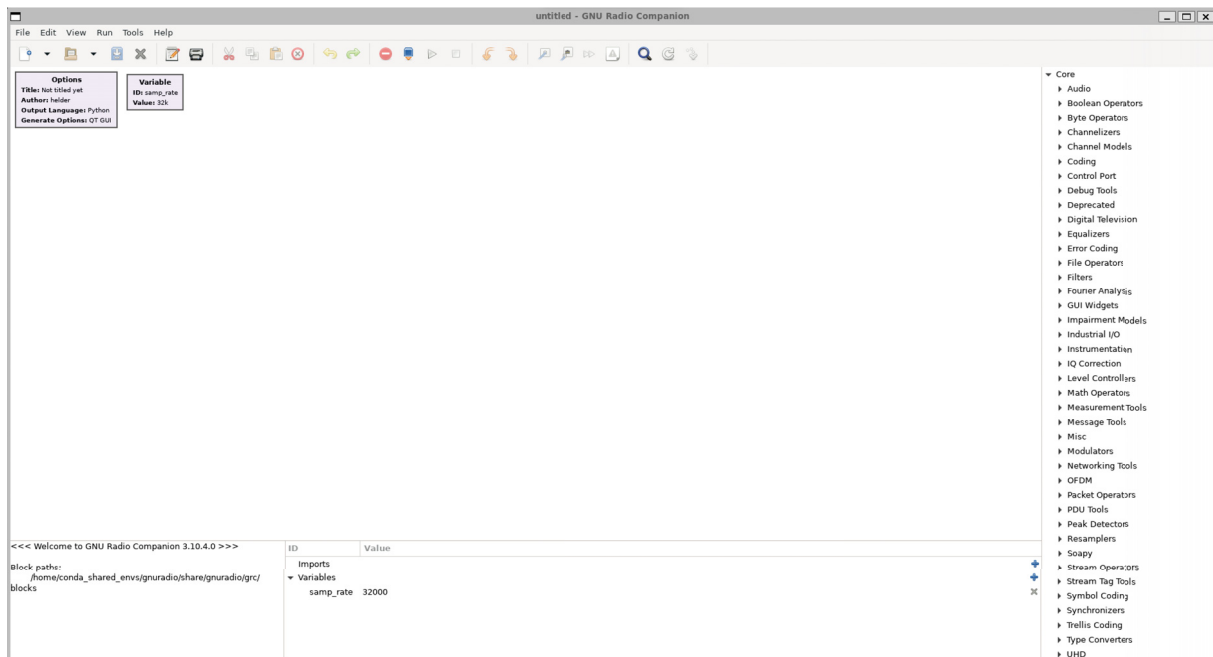
Apêndices

APÊNDICE A – TUTORIAL DE UTILIZAÇÃO DO GNU RADIO

Neste apêndice será apresentado um passo-a-passo de como utilizar o software de processamento digital de sinais GNU Radio. O tutorial consiste de uma simulação simples de um sinal cossenoidal no qual o usuário poderá ver na tela o sinal plotado no tempo e na frequência, com um bloco multiplicador para modificar a forma de onda.

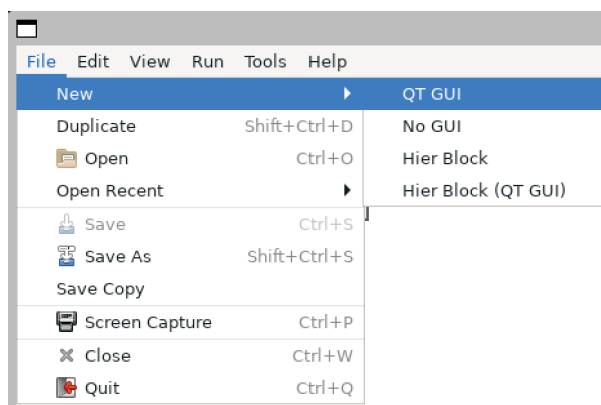
Digite **conda activate /home/conda_shared_envs/gnuradio** no terminal do Linux e em seguida **gnuradio-companion** para abrir o programa, em que aparecerá uma tela com um fluxograma em branco e dois blocos padrão, mostrado na [Figura 46](#). Caso o fluxograma não esteja em branco, poderá ser criado um novo clicando na guia superior em **File > New > QT GUI**, como mostrado na [Figura 47](#).

Figura 46 – Página inicial do programa GNU Radio



Fonte: O autor (2023).

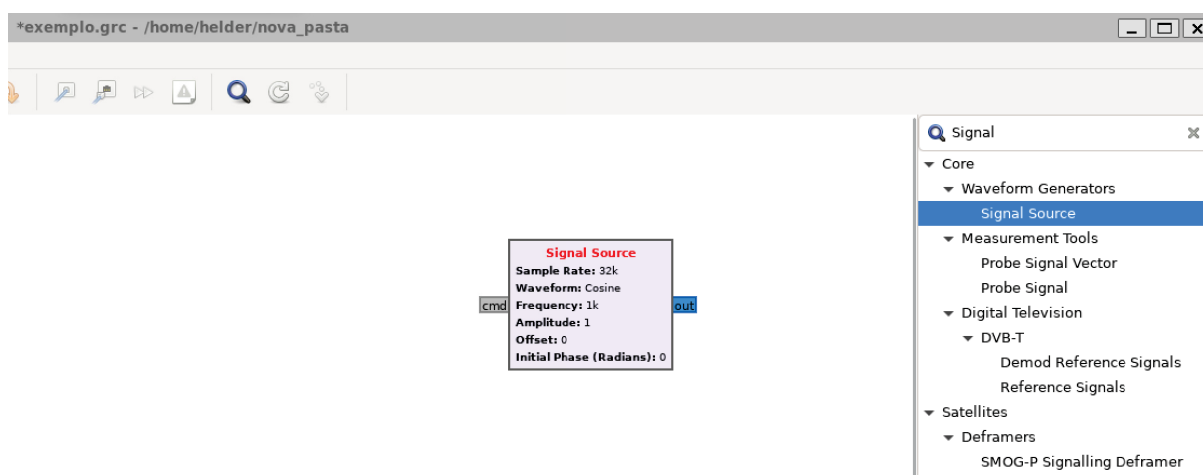
Figura 47 – Guia para abrir um novo fluxograma



Fonte: O autor (2023).

Para adicionar um elemento ao fluxograma, é utilizado o menu lateral direito, conforme ilustra a [Figura 48](#). Para facilitar a localização do bloco desejado, é possível clicar no botão com uma lupa e pesquisar o elemento a ser utilizado no fluxograma. Para inserir o elemento, clica-se duas vezes com o botão esquerdo do mouse ou segura-se o clique e arrasta a palavra para dentro do fluxograma. A [Figura 48](#) mostra um bloco **Signal Source** sendo inserido ao fluxograma.

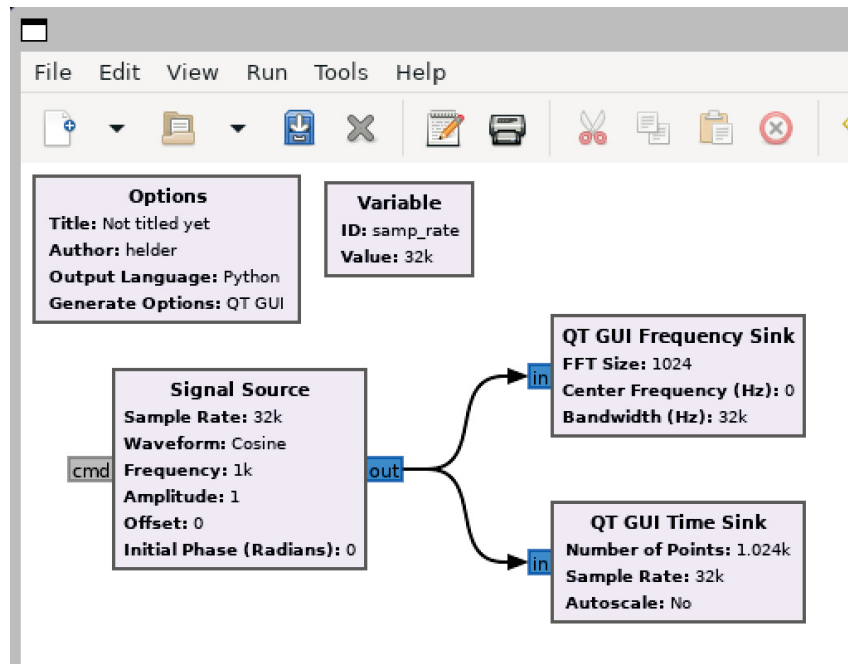
Figura 48 – Inserção de um elemento ao fluxograma



Fonte: O autor (2023).

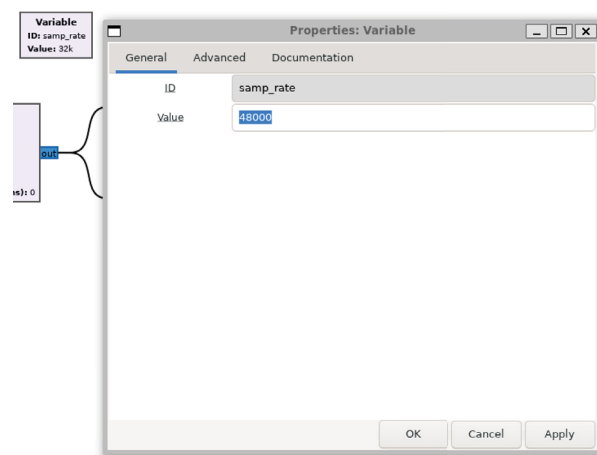
Para poder simular e observar a forma de onda, criam-se dois blocos adicionais chamados **QT GUI Frequency Sink** para plotar um gráfico da entrada em função da frequência e **QT GUI Time Sink** para plotar a entrada em função do tempo. Em seguida, conectam-se as entradas e saídas dos blocos presentes, clicando e segurando o clique do mouse nos quadrados azuis escrito **out** e arrastando até os quadrados azuis rotulados como **in**, resultando no fluxograma mostrado na [Figura 49](#).

Figura 49 – Fluxograma com blocos conectados entre si



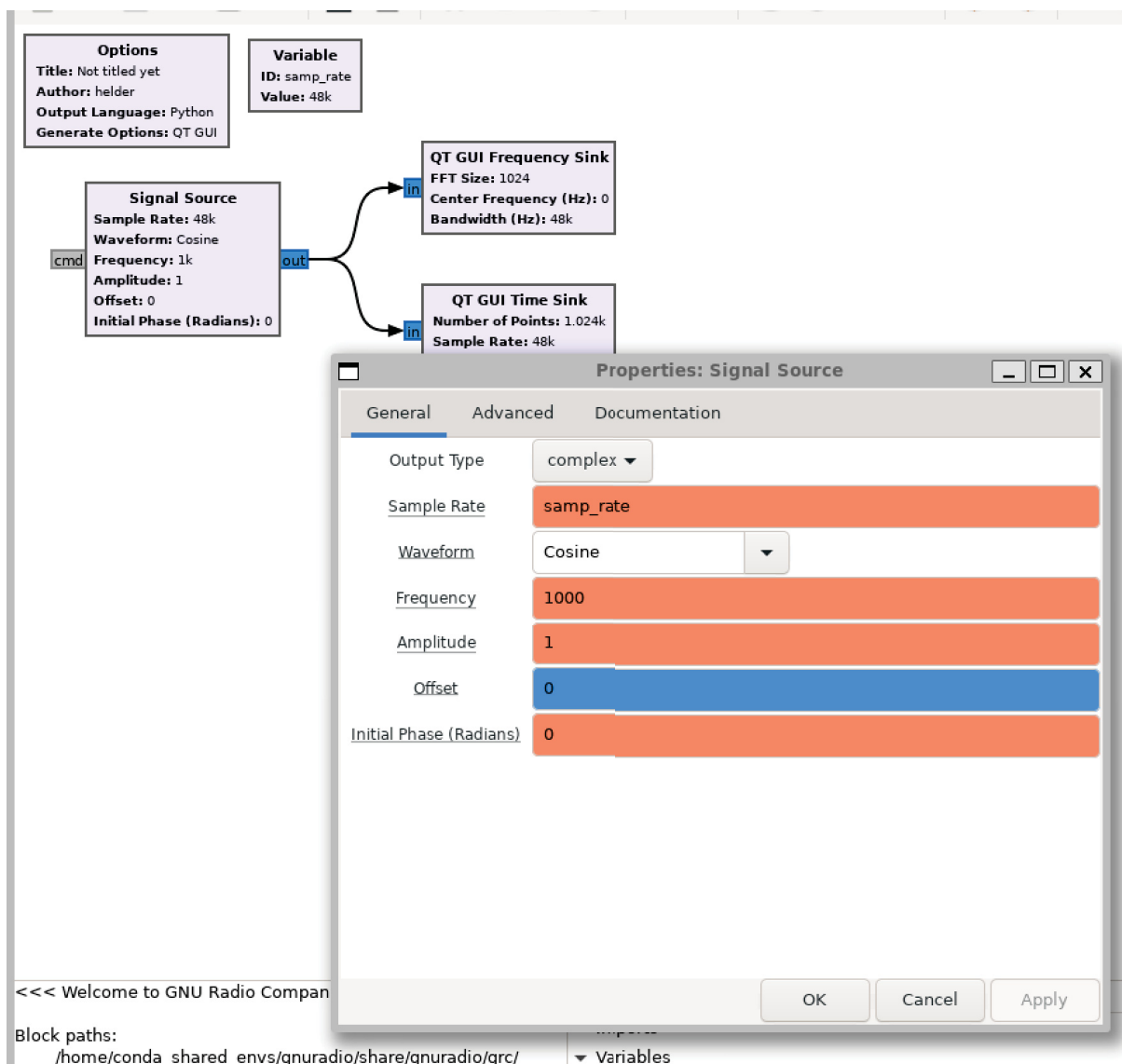
Fonte: O autor (2023).

Antes de iniciar a simulação, clica-se duas vezes com o botão esquerdo no bloco **Variable** com o campo **ID: samp_rate** e altera-se o valor da taxa de amostragem para 480000 (ou 48e3) no campo **Value**, ilustrado na [Figura 50](#). Pode-se notar que ao alterar a taxa de amostragem deste bloco, também é alterado nos blocos **Signal Source**, **QT GUI Frequency Sink** e **QT GUI Time Sink**, pois o campo **Sample Rate** destes blocos possuem a variável **samp_rate** como parâmetro, como pode ser observado na [Figura 51](#) ao se clicar duas vezes com o botão esquerdo em cima do bloco **Signal Source**.

Figura 50 – Configurações avançadas do bloco **Variable**

Fonte: O autor (2023).

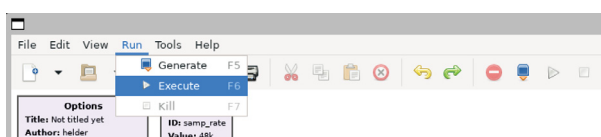
Figura 51 – Configurações avançadas do bloco **Signal Source**



Fonte: O autor (2023).

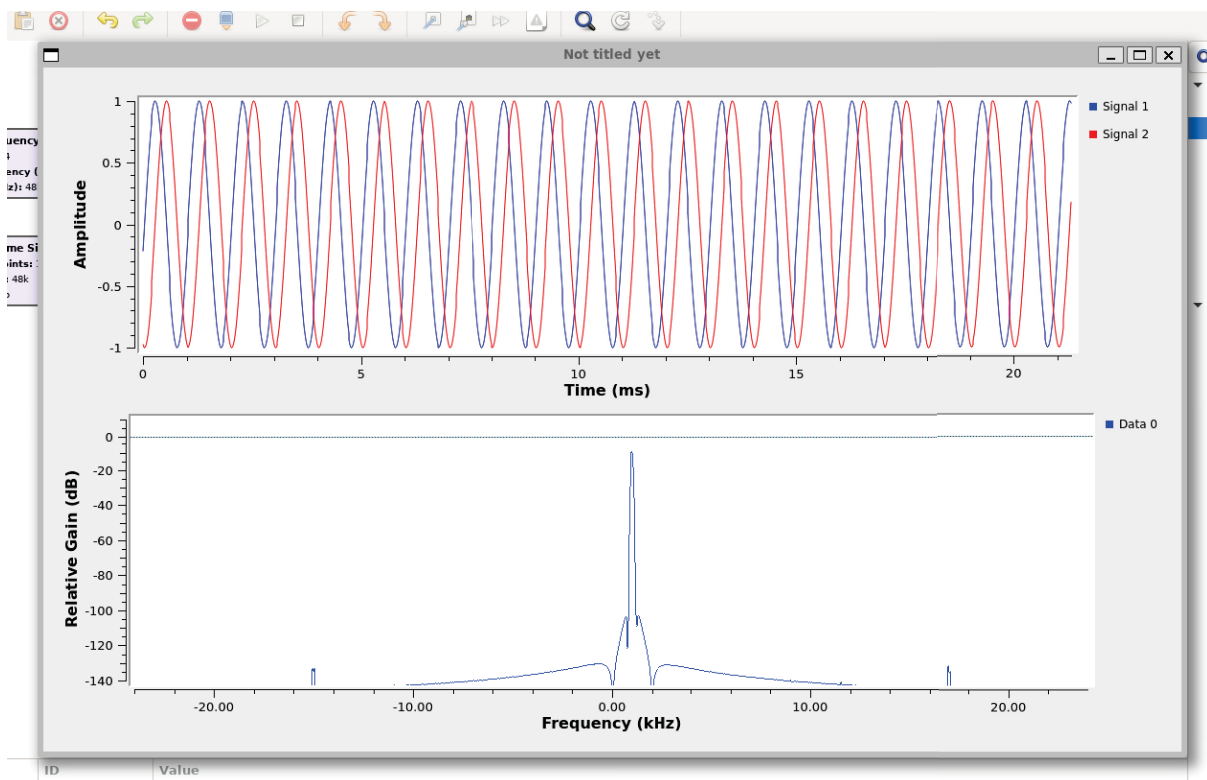
Para iniciar a simulação, clica-se no botão que possui uma seta direita, ou então pelo menu **Run > Execute** conforme é mostrado na Figura 52. Após a simulação iniciar, será gerada uma nova janela contendo os gráficos da forma de onda no tempo e na frequência, em tempo real, ilustrado na Figura 53.

Figura 52 – Configurações avançadas do bloco **Variable**



Fonte: O autor (2023).

Figura 53 – Sinal senoidal representado no tempo e na frequência

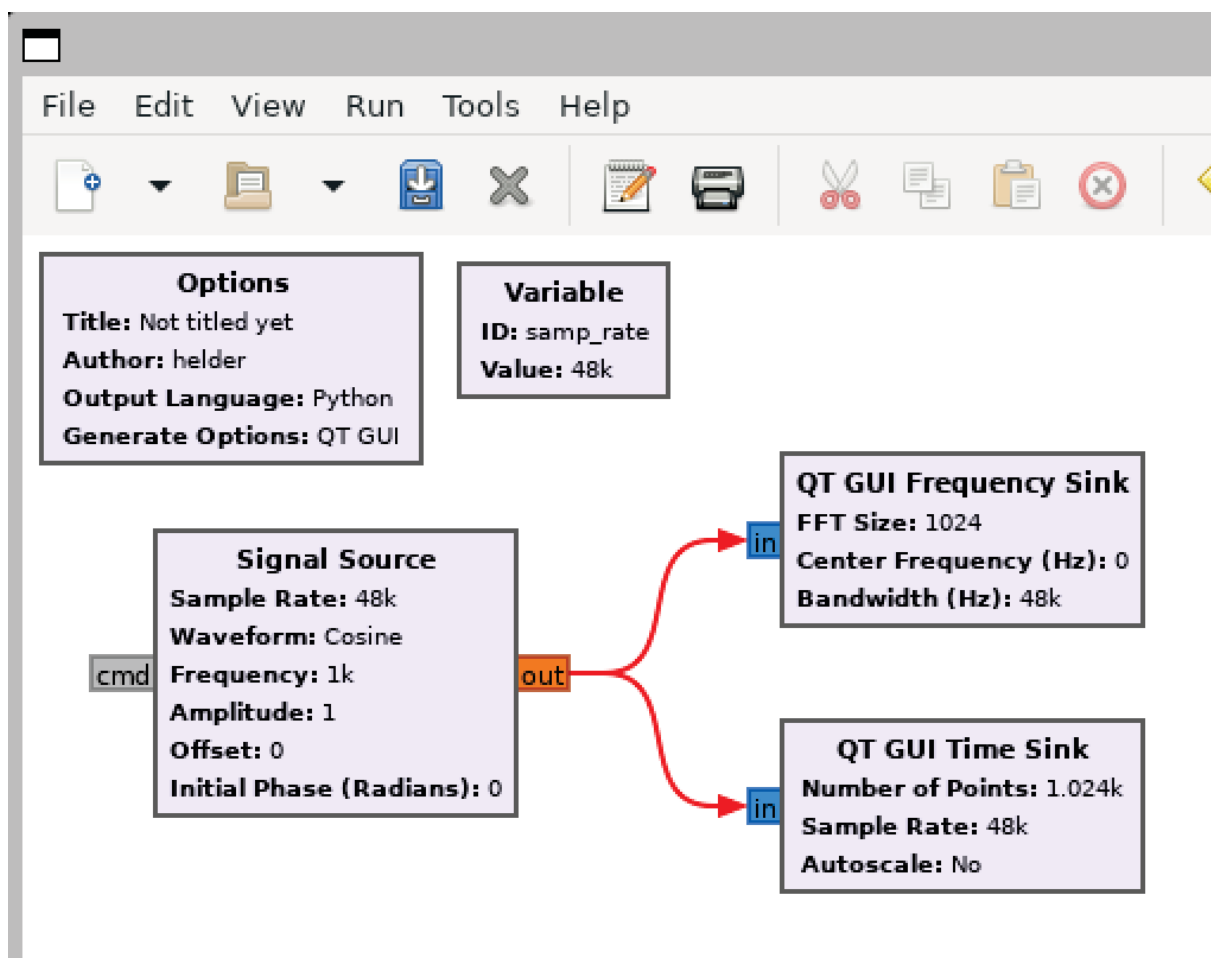


Fonte: O autor (2023).

A forma de onda gerada na [Figura 53](#) contém duas componentes, uma imaginária e uma real, pois o formato de saída e entrada dos blocos por padrão é do tipo **complex**, além do gráfico da onda em função da frequência apresentar apenas um lóbulo. Portanto, será feita a modificação do formato **complex** para **float** a seguir, de modo a gerar apenas a componente real da onda cossenoidal.

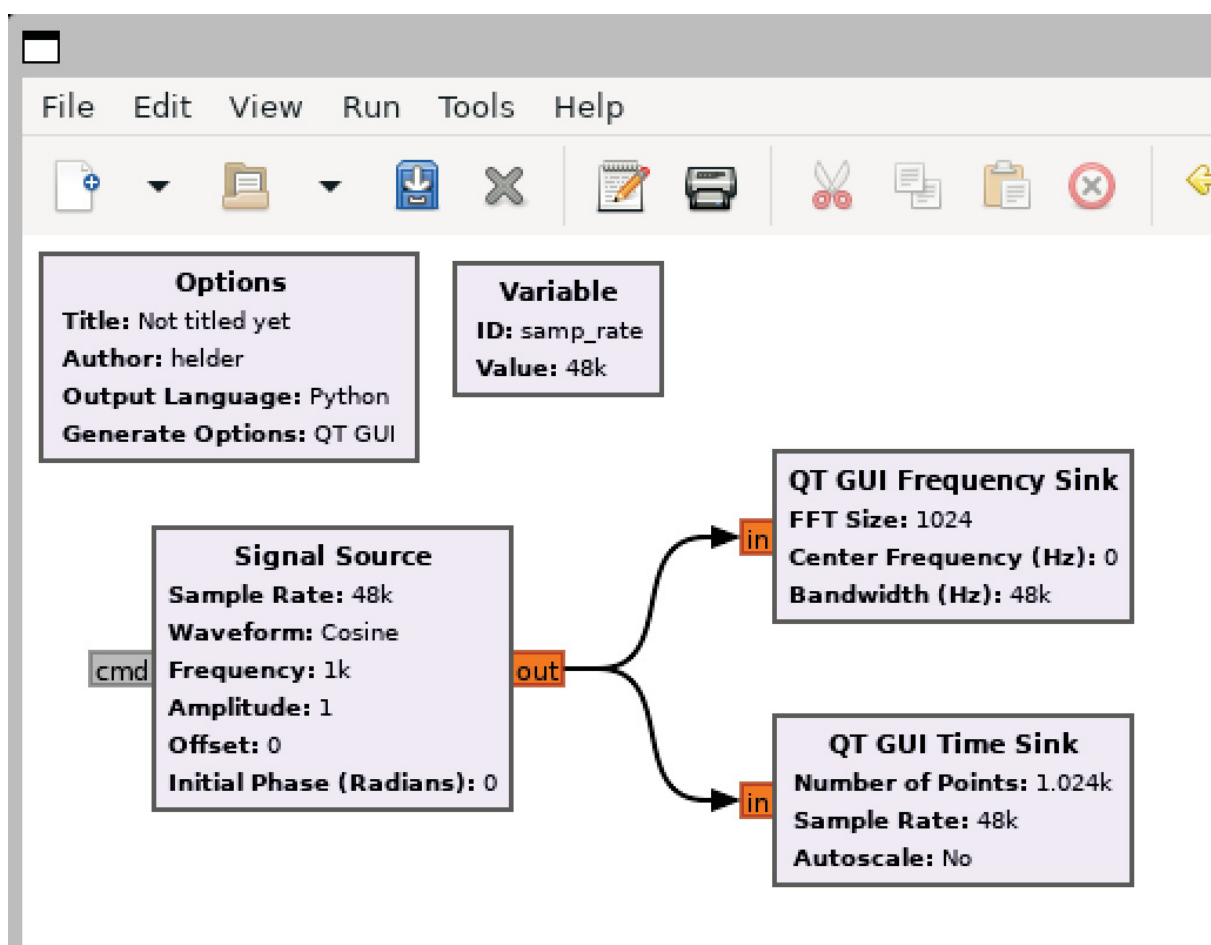
Para encerrar a simulação, clica-se no quadrado ao lado da seta utilizada para iniciá-la, ou então em **Run > Kill** do menu apresentado na [Figura 52](#). Após o encerramento da simulação, abre-se novamente o menu do bloco **Signal Source** representado na [Figura 51](#) e muda-se o campo **Output Type** de **complex** para **float**. Ao mudar o tipo da variável de entrada/saída dos blocos, a cor destes também mudam, passando de azul (**complex**) para laranja (**float**), conforme mostra a [Figura 54](#). Porém, agora as ligações do bloco **Signal Source** para os blocos de plotagem dos gráficos estão em vermelho, indicando um problema no fluxograma. Isto ocorre devido aos blocos conectados terem tipos de variáveis de entrada/saída diferentes entre si, portanto é necessário que todos os blocos estejam configurados para gerar/receber variáveis do tipo **float**. Assim, muda-se as configurações dos demais blocos para que fiquem com o mesmo tipo de variável (**float**), ilustrado na [Figura 55](#).

Figura 54 – Fluxograma com blocos possuindo tipos de variáveis diferentes em conflito (float e complex)



Fonte: O autor (2023).

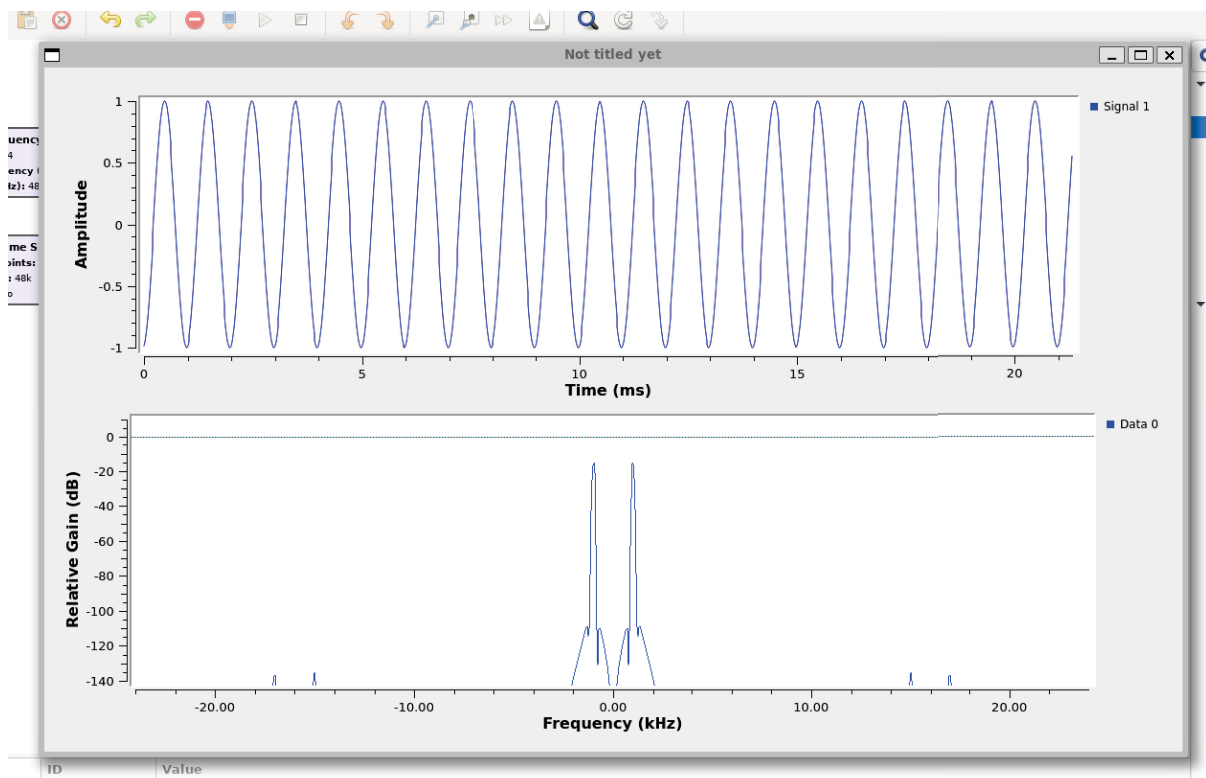
Figura 55 – Fluxograma com blocos possuindo variáveis do mesmo tipo (**float**)



Fonte: O autor (2023).

Ao rodar a simulação novamente, o gráfico agora possui apenas a componente real da onda cossenoidal, além de na frequência possuir dois lóbulos simétricos entre si em relação à origem (frequência zero), como é mostrado na [Figura 56](#).

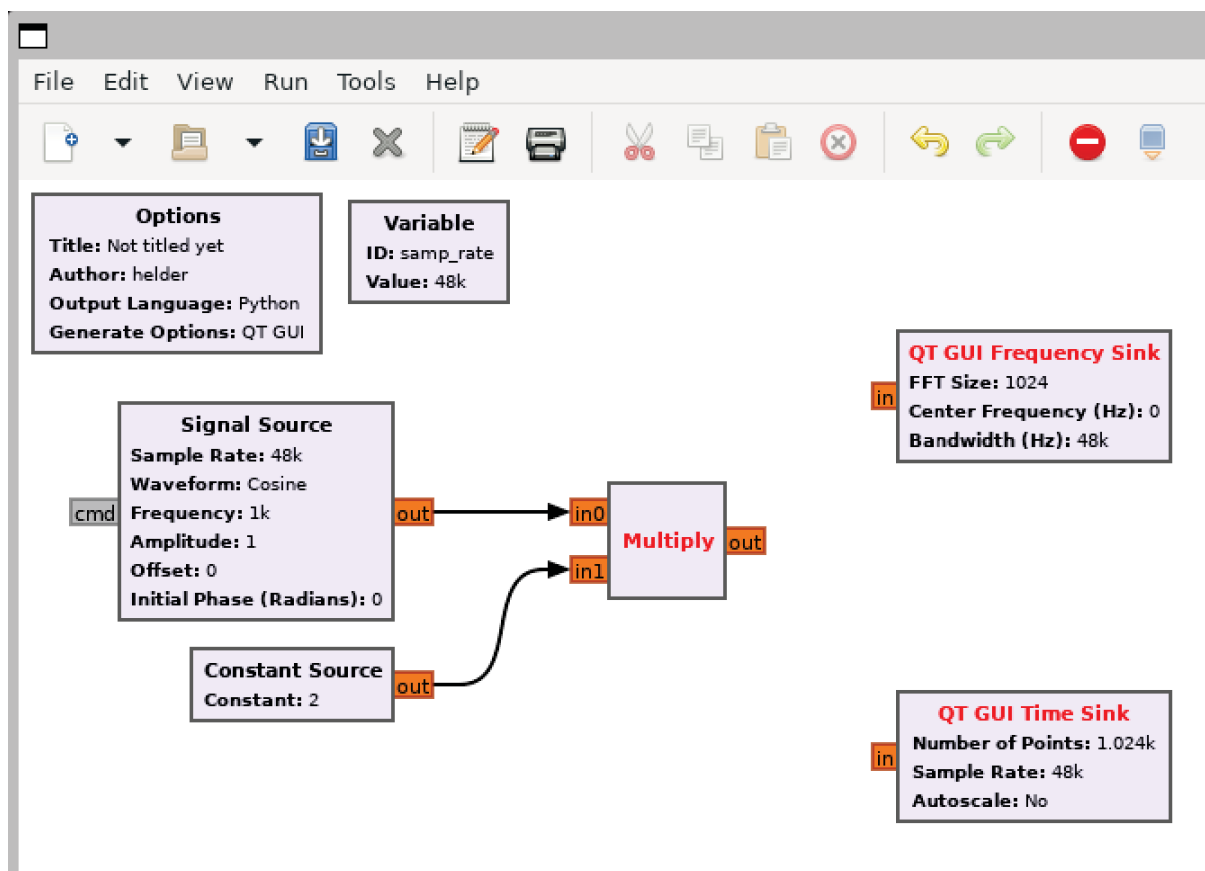
Figura 56 – Sinal senoidal representado no tempo e na frequência, com o sinal de entrada do tipo **float**



Fonte: O autor (2023).

Para manipular a forma de onda gerada pelo bloco **Signal Source**, pode-se inserir blocos intermediários entre o gerador de sinal e o bloco que gera os gráficos. Como exemplo, adicionam-se os blocos **Multiply** e **Constant Source**, conectando as suas entradas e saídas conforme a [Figura 57](#), em que o parâmetro **Constant** do bloco **Constant Source** vale **2** e os tipo dos dados são **float**.

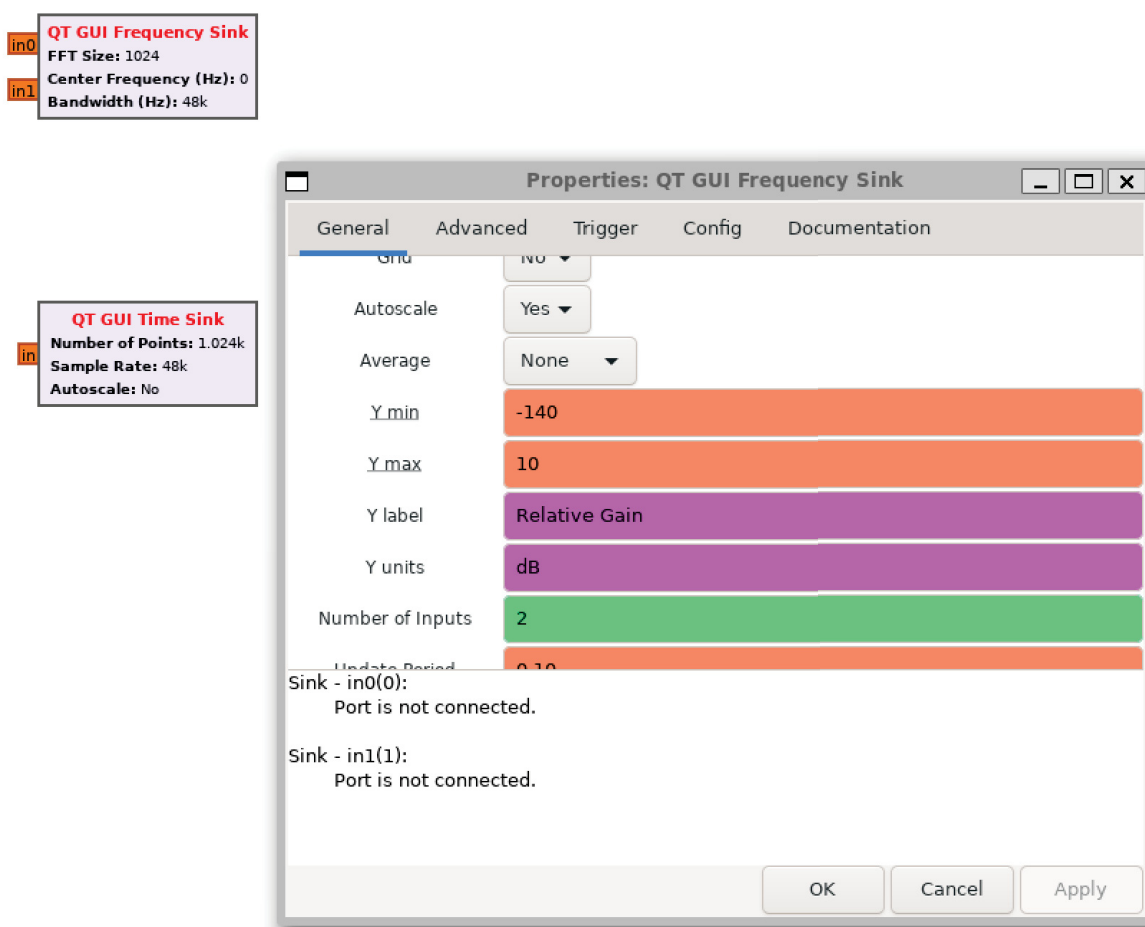
Figura 57 – Fluxograma com bloco **Multiply**



Fonte: O autor (2023).

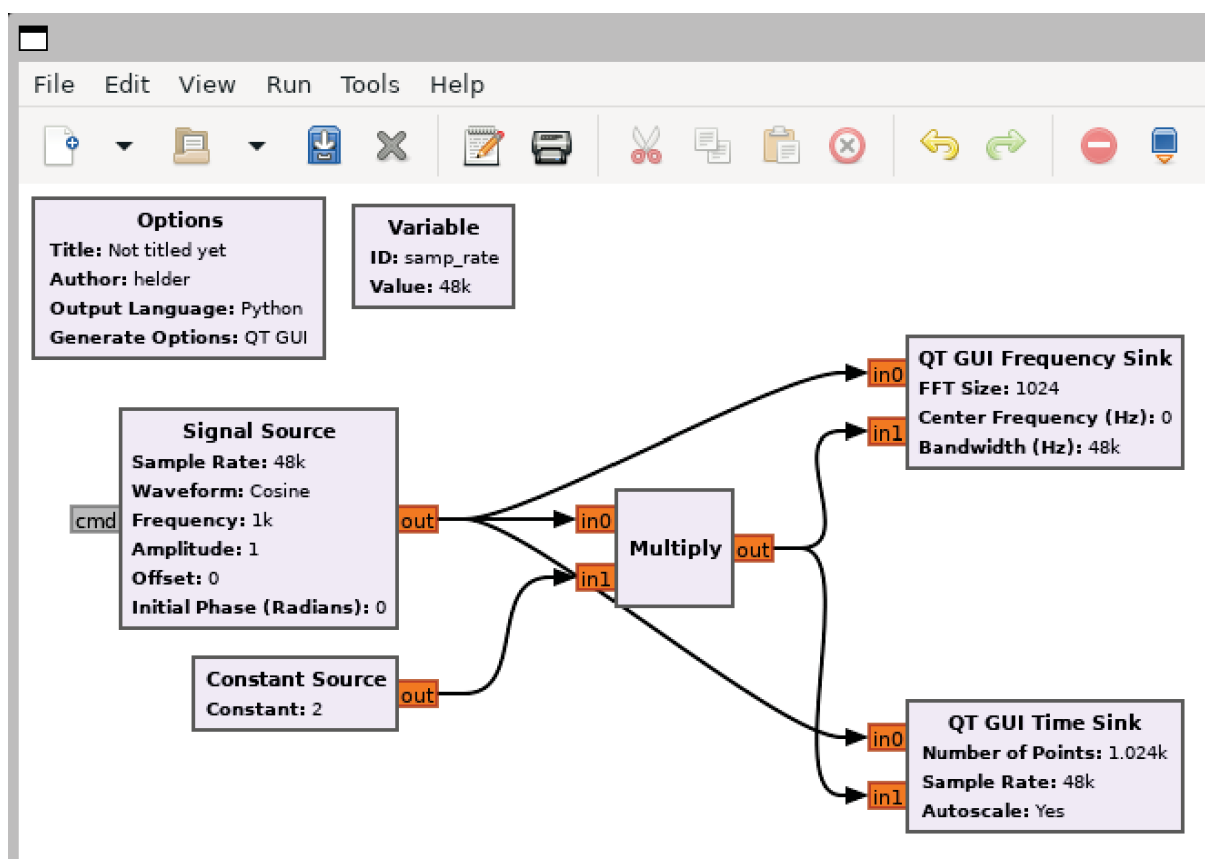
Para visualizar em um mesmo gráfico as duas formas de onda, abrem-se as configurações avançadas do **QT GUI Time Sink** e do **QT GUI Frequency Sink**, selecionando **Yes** no campo **Autoscale** e inserindo **2** no campo **Number of Inputs**, ilustrado na Figura 58, com o fluxograma resultante mostrado na Figura 59.

Figura 58 – Janela de configurações do bloco QT Gui Frequency Sink



Fonte: O autor (2023).

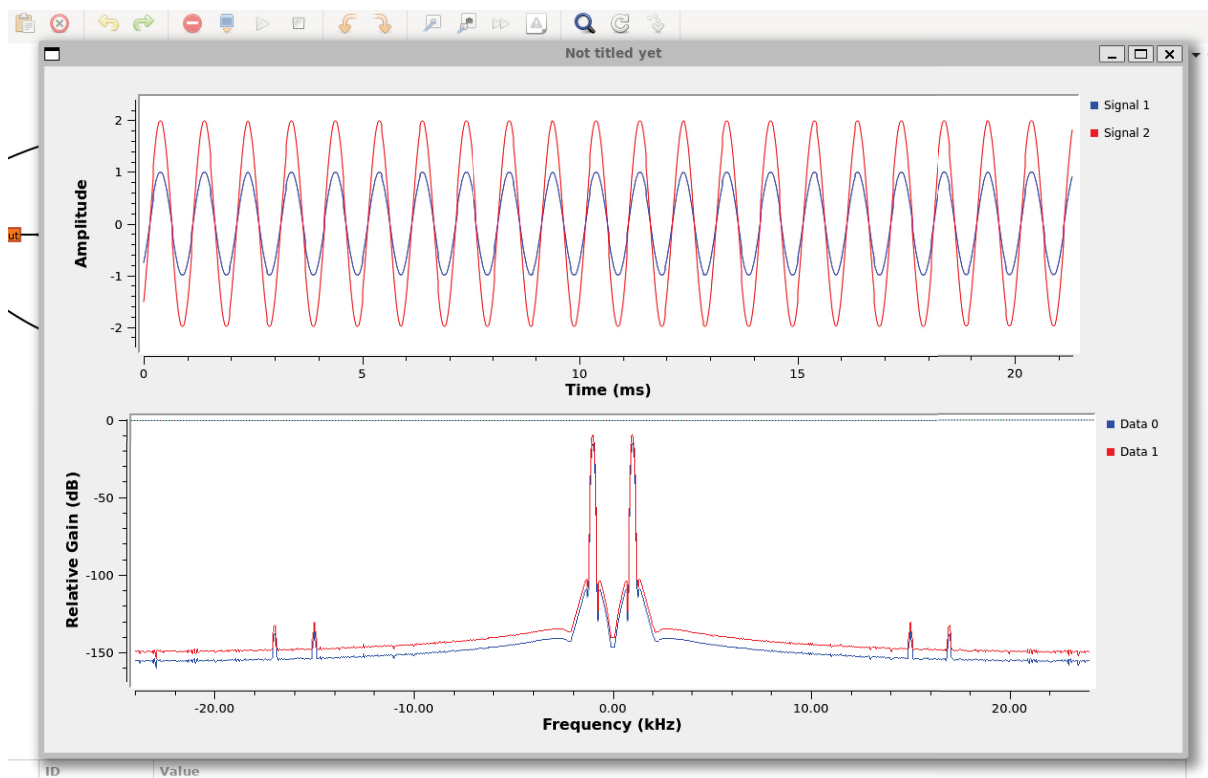
Figura 59 – Fluxograma completo



Fonte: O autor (2023).

Ao executar a simulação, geram-se gráficos que representam o sinal cossenoidal diretamente da fonte em azul e após passar pelo bloco multiplicador em vermelho, ambos em uma mesma janela para facilitar a visualização e comparação das duas formas de onda, em que uma possui o dobro da amplitude da outra (pois passa pelo bloco multiplicador com valor **2**), representado na [Figura 60](#).

Figura 60 – Sinal senoidal representado no tempo e na frequência, com um dos sinais multiplicado por 2



Fonte: O autor (2023).

Com este tutorial, cobre-se o básico para a utilização do GNU Radio para realizar simulações e processamento digital de sinais, podendo ser expandido as aplicações por meio dos blocos disponíveis no programa, por blocos customizados disponíveis na internet ou por blocos programados pelo próprio usuário.

APÊNDICE B – CONFIGURAÇÃO DO SERVIDOR

Neste apêndice são mostrados os passos para a instalação dos softwares e a realização das conexões dos equipamentos no servidor localizado na Universidade Federal do Paraná. É feita a instalação do software GNU Radio no sistema Linux CentOS 7 por meio de uma distribuição Python e também instalado um bloco do GNU Radio que permite o acesso ao dispositivo de rádio definido por software RTL-SDR. Por fim, é descrito os passos para conectar os equipamentos [USRP](#) e o RTL-SDR no servidor, além das configurações necessárias para operá-los pelo sistema operacional instalado.

B.1 Instalação do GNU Radio

No servidor, com o sistema operacional Linux CentOS 7 instalado, para se ter acesso ao hardware de rádio e realizar simulações gráficas é necessário ter privilégios de administrador e realizar a instalação do software GNU Radio. Porém, a distribuição CentOS 7 é bastante antiga, possuindo 8 anos de vida a partir da data de escrita deste trabalho, e cujo fim de vida está próximo (2024). Por esta razão, não é possível instalar diretamente o software GNU Radio atualizado através dos repositórios oficiais do CentOS 7. Sendo assim, é necessário instalar manualmente mediante construções a partir do código fonte ou por meio da ferramenta Anaconda, uma distribuição Python voltado para computação científica. Escolheu-se então a ferramenta Anaconda por permitir que seja possível instalar o GNU Radio em outras distribuições Linux, como por exemplo o Ubuntu, assim permitindo que o sistema proposto seja implantado em outros sistemas operacionais visto que a data de vencimento do CentOS 7 está próxima do fim.

Com a ferramenta Anaconda instalada, cria-se um *conda environment* chamado *gnuradio*, que é um ambiente onde pacotes do Anaconda são instalados de forma isolada do sistema principal. Assim, é possível instalar o GNU Radio na versão 3.10 através do repositório *conda-forge*, dentro do ambiente *gnuradio* criado.

Os seguintes comandos bash mostram como criar o ambiente *gnuradio* e instalar o GNU Radio pelo Anaconda:

```
// Cria um diretorio para acesso publico
$ sudo mkdir /home/gnuradio_envs
// Define permissoes para que todos os usuarios
// possam acessar a pasta
```

```
$ sudo chmod 755 /home/gnuradio_envs
// Cria ambiente chamado gnuradio
$ conda create -p /home/gnuradio_envs/gnuradio
// Ativa (entra) no ambiente criado
$ conda activate /home/gnuradio_envs/gnuradio/
// Define o repositório para a obtenção de pacotes como
// conda-forge e restringe para pacotes estáveis
// gerando menor possibilidade de conflitos
$ conda config --env --add channels conda-forge
$ conda config --env --set channel_priority strict
// Instala o GNU Radio
$ conda install gnuradio
```

Além disto, alguns pacotes adicionais do Anaconda são necessários para o controle do RTL-SDR, pois o bloco *Soapy RTLSDR Source* não funcionou no CentOS 7, sendo então feita a instalação do *gnuradio-osmosdr* pelo repositório *conda-forge*.

O seguinte comando bash realiza a instalação do bloco *RTL-SDR Source* presente no pacote *gnuradio-osmosdr* (o comando precisa ser executado dentro do ambiente *gnuradio* criado anteriormente):

```
// Instala o bloco RTL-SDR Source no GNU Radio
$ conda install -c conda-forge gnuradio-osmosdr
```

Para a configuração do computador (ou notebook) cliente, seguem-se os mesmos passos descritos anteriormente. A distribuição Linux pode ser alterada para outras versões tais como o popular Ubuntu, Linux Mint, etc. Os passos para a instalação do GNU Radio através do Anaconda independem do sistema operacional utilizado. Por fim, pode ser que o bloco *Soapy RTLSDR Source* que vem junto com o GNU Radio funcione em outras distribuições Linux, portanto dispensando a instalação do pacote *gnuradio-osmosdr* pelo *conda-forge*.

O comando para executar o GNU Radio em modo gráfico e realizar os experimentos está mostrado a seguir:

```
// Inicia o GNU Radio em modo grafico
$ gnuradio-companion
```

Por fim, para se obter acesso remoto ao servidor que está executando o GNU Radio, é necessário utilizar o protocolo [SSH](#) com o seguinte comando:

```
// Inicia uma conexão remota com o usuário <user> no
// endereço <address> na porta <port>
// e habilita a execução de programas com
```

```
// interface grafica atraves da opcao "-X"  
$ ssh -X <user>@<address> -p <port>
```

B.2 Conexão dos equipamentos de rádio

Para conectar no servidor e tornar acessível o **USRP**, é preciso plugá-lo via porta Ethernet. Após a conexão, com o comando *ifconfig* atribui-se o endereço IP do **USRP** para que este seja acessível pelo sistema.

```
// Atribui o endereço do USRP <ip_usrp>  
// na porta Ethernet de nome <interface>  
$ sudo ifconfig <interface> <ip_usrp> \  
> netmask 255.255.255.0
```

No caso do **USRP** utilizado neste trabalho, *<ip_usrp>* tem como valor o endereço *192.168.10.1* e a interface *<interface>* utilizada é *enp4s0u1*.

Para se configurar o RTL-SDR, pluga-se via porta **USB 2.0**. No caso do sistema operacional CentOS 7, é necessário a instalação do pacote *rtl-sdr* que inclui os drivers necessários para seu funcionamento. Em outros sistemas operacionais Linux, o modo de instalar os drivers do RTL-SDR pode variar, sendo indicada a consulta na internet de como instalar na distribuição pretendida.

A instalação do *rtl-sdr* no CentOS 7 é feita através do seguinte comando:

```
// Instala as dependencias do rtl-sdr  
$ sudo yum install rtl-sdr
```

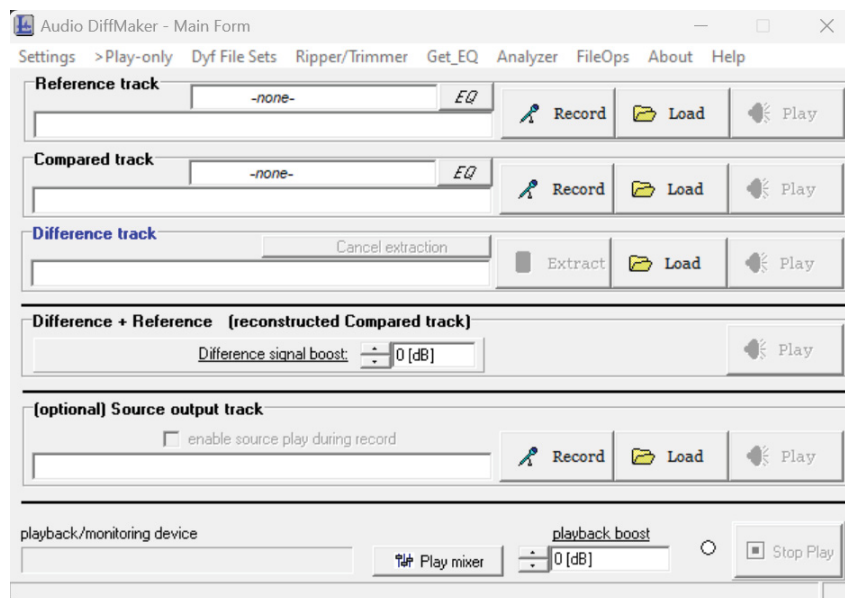
Com a instalação do GNU Radio descrita no [Apêndice B.1](#) e com os equipamentos conectados, o sistema está pronto para a realização de experimentos.

APÊNDICE C – TUTORIAL DE UTILIZAÇÃO DO DIFFMAKER

Neste apêndice será apresentado um passo-a-passo de como utilizar o software de análise de áudio DiffMaker (gratuito, disponível apenas para Windows). O programa possui um conjunto de ferramentas para determinar a diferença absoluta de uma gravação, omitindo diferenças devido à diferença de nível, sincronização de tempo ou respostas de frequência linear simples e é baseado na técnica de diferenciação digital (WASLO, 2008).

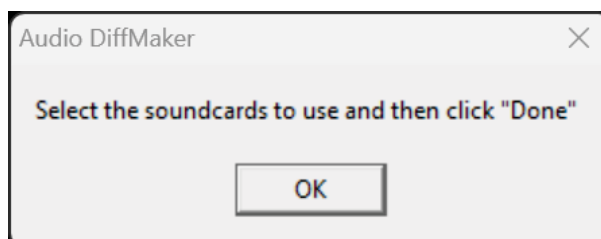
O tutorial consiste em apresentar como comparar dois arquivos de áudio supostamente iguais, mas em que um deles foi inserido um ruído branco no qual é detectado pelo DiffMaker e gerado um arquivo de áudio à parte. Primeiro, inicializa-se o programa DiffMaker no qual irá aparecer a janela principal do programa mostrada na Figura 61 e uma janela de popup solicitando confirmação para selecionar uma placa de som, conforme ilustra a Figura 62.

Figura 61 – Janela inicial do programa



Fonte: O autor (2023).

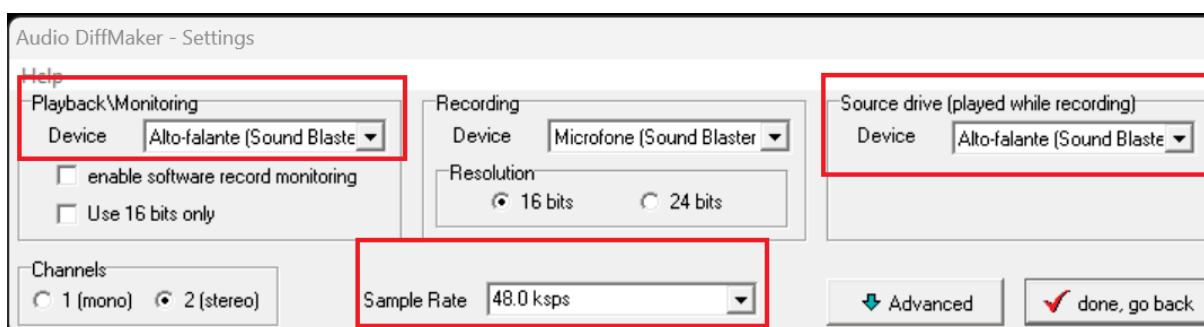
Figura 62 – Janela de popup solicitando ao usuário para selecionar as placas de som



Fonte: O autor (2023).

Ao se clicar em **OK** na janela pedindo para selecionar a placa de áudio, uma nova é aberta com as configurações gerais do programa. Seleciona-se o dispositivo de áudio disponível no computador e a taxa de amostragem desejada, conforme mostra a [Figura 63](#), e clica-se em **done, go back**.

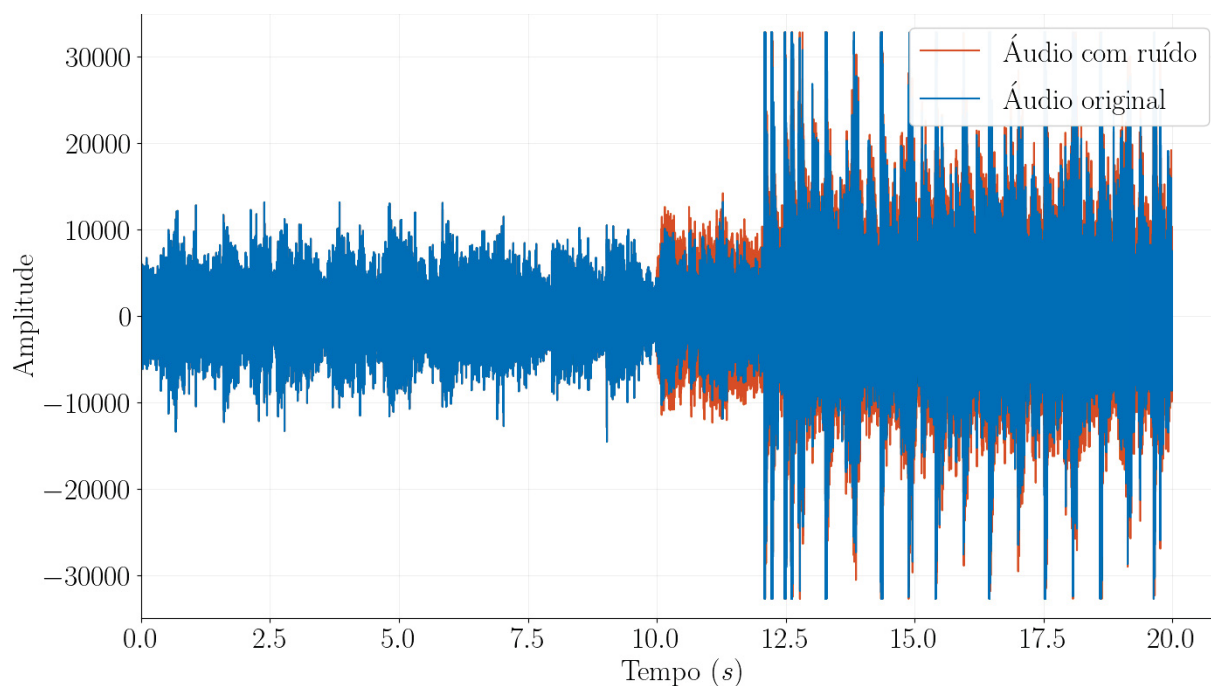
Figura 63 – Janela de configurações do programa



Fonte: O autor (2023).

O programa irá voltar para interface da [Figura 61](#), onde serão selecionados os arquivos de áudio para comparação. Foi utilizado um arquivo de áudio chamado *Deep State* ([JAPAN, 2020](#)), em que foi transformado em arquivo *WAV* de um canal de áudio (mono) e foi cortado a duração para 20 segundos. Para realizar uma comparação simples com o *DiffMaker*, criou-se uma cópia do arquivo de áudio com um ruído branco injetado nos 10 segundos finais. Ambos áudio original e com ruído adicionadl estão ilustrados na [Figura 64](#).

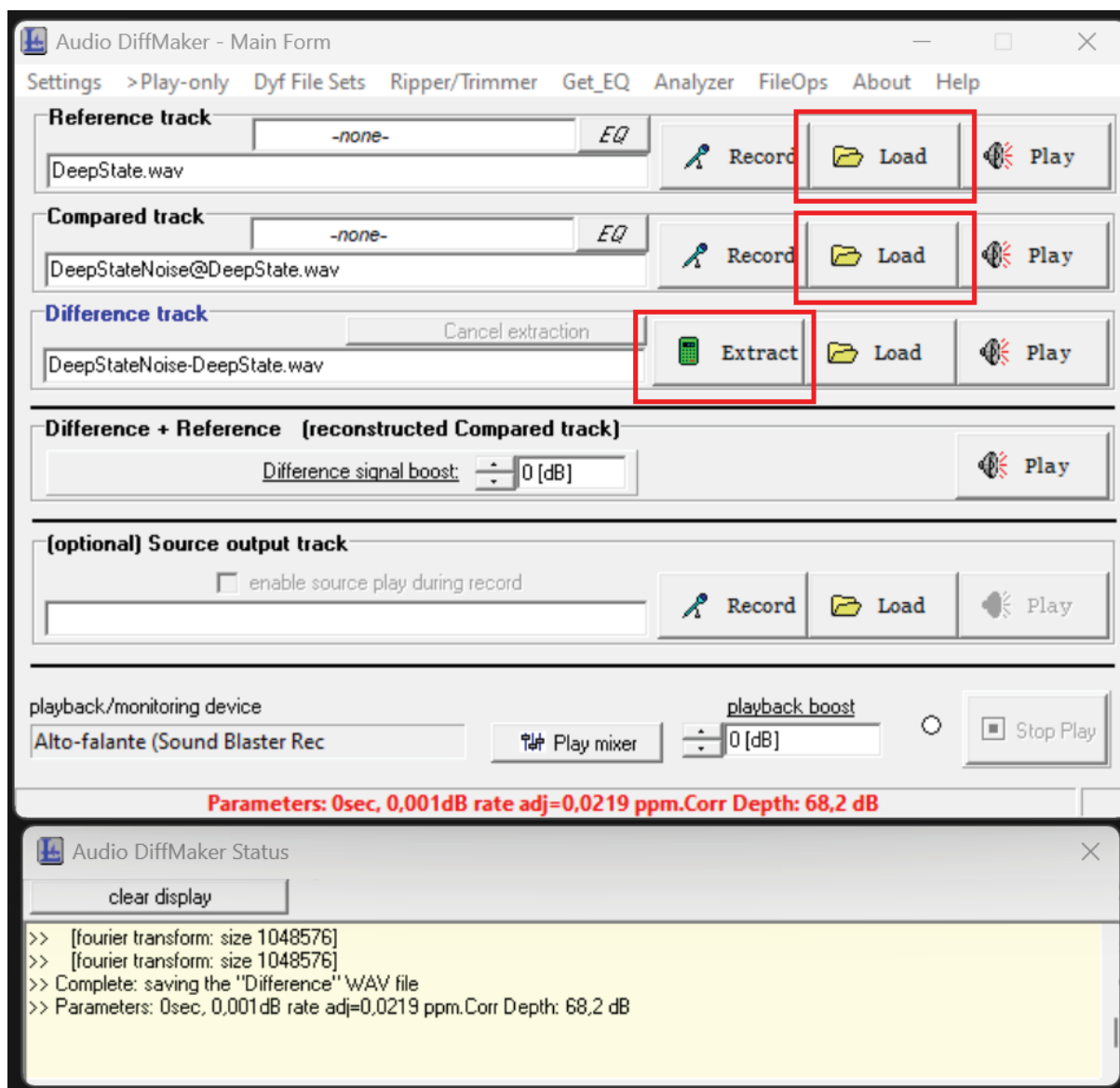
Figura 64 – Áudio original e com ruído adicionado



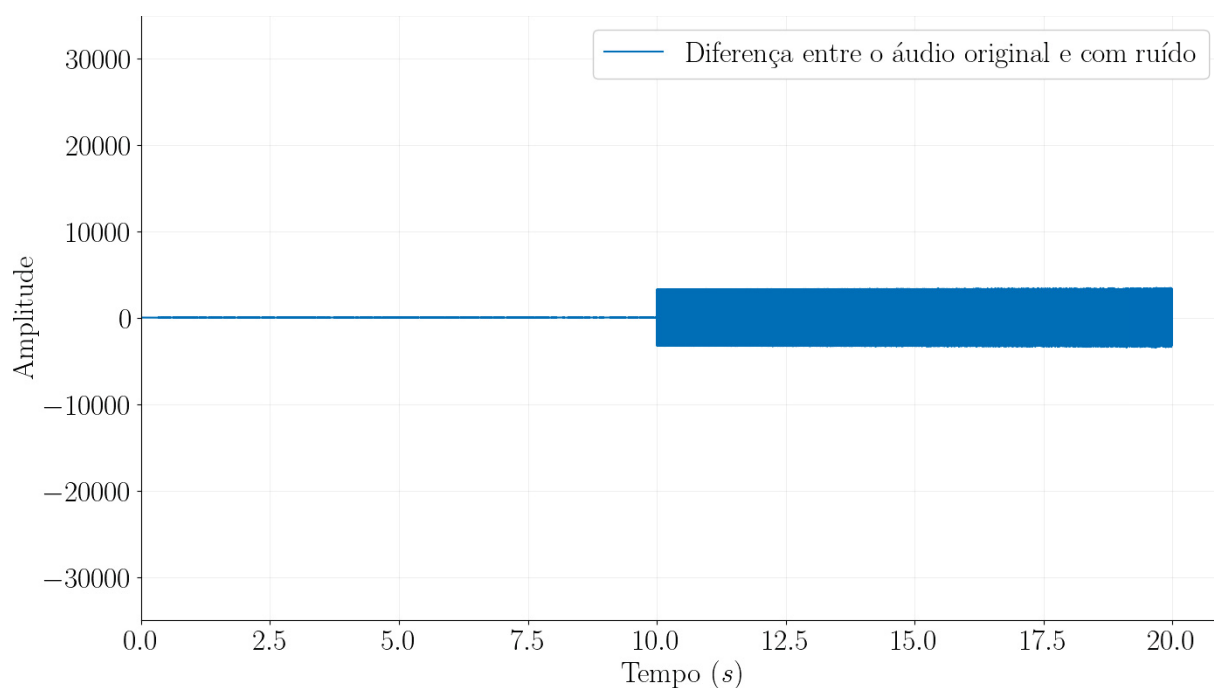
Fonte: O autor (2023).

Conforme ilustra a [Figura 65](#) com retângulos vermelhos, seleciona-se o áudio de referência através do botão **Load** localizado em *Reference track*, o áudio com ruído no botão **Load** localizado em *Compared track* e finalmente clica-se em **Extract** localizado em *Difference track*. O programa irá executar e fazer a comparação dos arquivos de áudio, com uma pequena janela para informar o progresso abaixo da janela principal. Serão gerados dois arquivos adicionais, chamados *DeepStateNoise-DeepState* e *DeepStateNoise@DeepState*, cujo arquivo de interesse é o primeiro que contém o ruído extraído, mostrado na [Figura 66](#), em que o sinal de ruído é detectado e está presente nos 10 segundos finais do áudio, conforme esperado.

Figura 65 – Janela principal do programa



Fonte: O autor (2023).

Figura 66 – Ruído extraído pelo *DiffMaker*

Fonte: O autor (2023).