

INSTITUTO FEDERAL DA PARAÍBA
CAMPUS JOÃO PESSOA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VINÍCIUS JOSÉ WANDERLEY VIEIRA

SIMULAÇÃO DE UM SISTEMA DE COMUNICAÇÃO OFDM
UTILIZANDO GNU RADIO

João Pessoa-PB
2026

VINÍCIUS JOSÉ WANDERLEY VIEIRA

**SIMULAÇÃO DE UM SISTEMA DE COMUNICAÇÃO OFDM
UTILIZANDO GNU RADIO**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso Superior de Bacharelado em
Engenharia Elétrica, no Instituto Federal da Paraíba
– IFPB, como requisito parcial à obtenção do grau de
Bacharel em Engenharia Elétrica.

Orientadora: Prof. Dra. Suzete Élide Nóbrega Cor-
reia

**João Pessoa-PB
2026**

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha do IFPB, *Campus* João Pessoa

V658s Vieira, Vinícius José Wanderley.

Simulação de um sistema de comunicação OFDM
utilizando GNU radio / Vinícius José Wanderley Vieira. - 2026.
50 f. : il.

TCC (Graduação – Bacharelado em Engenharia Elétrica) –
Instituto Federal de Educação da Paraíba / Coordenação do
Curso Superior de Bacharelado em Engenharia Elétrica, 2026.
Orientação: Prof^a. Dra. Suzete Élide Nóbrega Correia.

1.OFDM. 2. GNU Radio. 3. Rádio definido por software. 4.
Padrão IEEE 802.IIa. 5. Taxa de erro de Bit. I. Título.

CDU 621.396.67:004.41(043)

Bibliotecária responsável: Lucrecia Camilo de Lima – CRB 15/132

VINÍCIUS JOSÉ WANDERLEY VIEIRA

**SIMULAÇÃO DE UM SISTEMA DE COMUNICAÇÃO OFDM
UTILIZANDO GNU RADIO**

A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso “SIMULAÇÃO DE UM SISTEMA DE COMUNICAÇÃO OFDM UTILIZANDO GNU RADIO” elaborado por “VINÍCIUS JOSÉ WANDERLEY VIEIRA” como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba.

João Pessoa-PB, 05/02/2026

Comissão Examinadora

Documento assinado digitalmente



SUZETE ELIDA NOBREGA CORREIA

Data: 19/02/2026 14:19:46-0300

Verifique em <https://validar.iti.gov.br>

**Prof. Dra. Suzete Élide Nóbrega
Correia - IFPB
(Orientadora)**

Documento assinado digitalmente



ADAILDO GOMES D ASSUNCAO JUNIOR

Data: 19/02/2026 14:12:48-0300

Verifique em <https://validar.iti.gov.br>

**Prof. Dr. Adaildo Gomes D. Assunção
Junior - IFPB**

Documento assinado digitalmente



LUIS ROMEU NUNES

Data: 19/02/2026 12:52:22-0300

Verifique em <https://validar.iti.gov.br>

Prof. Dr. Luis Romeu Nunes - IFPB

Dedico este trabalho aos meus pais, pois é graças aos seus esforços que hoje posso concluir o meu curso.

Agradecimentos

- Agradeço primeiramente à minha família, especialmente aos meus pais Aryana Wanderley e Francilberto Vieira, e às minhas tias por todo apoio ao longo deste curso. O suporte financeiro e emocional de vocês fizeram toda a diferença nessa trajetória. Não teria chegado a onde estou sem esse apoio.
- Agradeço ao meu amigo de longa data Pedro Henrique, pelo companheirismo de casa e de vida durante esta etapa. Obrigado por dividir os problemas e as alegrias do cotidiano. Sua presença e parceria foram fundamentais para eu manter o equilíbrio necessário para chegar até aqui.
- Aos meus colegas de jornada, Isabely Samara e Adelson Ataíde, que tornaram os anos de graduação mais leves. Obrigado pelas discussões em grupo, pelas madrugadas de estudo e por compartilharem comigo não apenas o conhecimento técnico, mas uma amizade que certamente levarei para a vida.
- Agradeço ao Programa de Educação Tutorial (PET). Pelo ambiente de aprendizado constante e pelo incentivo à excelência acadêmica, que foram fundamentais para a realização deste trabalho
- Por último, agradeço à minha orientadora Suzete Elida, por todo apoio, incentivo e atenção depositados em mim durante minha trajetória acadêmica, que começou na monitoria, amadureceu durante os anos de PET e se consolidou nesta orientação de TCC. Sua dedicação e competência são constantes fontes de inspiração para todos que têm o privilégio de trabalhar ao seu lado.

SIMULAÇÃO DE UM SISTEMA DE COMUNICAÇÃO OFDM UTILIZANDO GNU RADIO

Resumo

Este trabalho consiste na implementação e avaliação de desempenho em uma simulação de um sistema comunicação digital utilizando a técnica OFDM (*Orthogonal Frequency-Division Multiplexing*), junto a uma plataforma de Rádio Definido por Software através da ferramenta *Open Source Gnu Radio Companion*, seguindo os parâmetros do padrão IEEE 802.11a. Foi desenvolvido um fluxograma do transceptor completo, integrando as etapas de modulação (BPSK e 16QAM), sincronização de tempo e frequência, estimação e equalização de canal, além de mecanismos de embaralhamento e checagem de erro. Utilizando esse conjunto de ferramentas foi possível implementar e testar o sistema em um cenário real de transmissão de imagens sob a influência de ruído AWGN. Os resultados foram obtidos por meio de gráficos (tempo e frequência), diagramas de constelações e da Taxa de Erro de Bit (BER - *Bit Error Rate*). A abordagem via *software* é viável e eficaz para o estudo prático de sistemas de telecomunicações, evidenciando as vantagens da técnica OFDM e auxiliando na implementação de diversos trabalhos futuros na área.

Palavras-chave: OFDM. Gnu Radio. Rádio Definido por Software. Padrão IEEE 802.11a. Taxa de Erro de Bit.

IMPLEMENTATION OF AN OFDM SYSTEM USING THE OPEN SOURCE GNU RADIO TOOL

Abstract

This work consists of the implementation and performance evaluation of a digital communication system simulation using the OFDM (Orthogonal Frequency-Division Multiplexing) technique, together with a Software Defined Radio platform using the Open Source Gnu Radio Companion tool, following the parameters of the IEEE 802.11a standard. A flowchart of the complete transceiver was developed, integrating the stages of modulation (BPSK and 16QAM), time and frequency synchronization, channel estimation and equalization, as well as scrambling and error checking mechanisms. Using this set of tools, it was possible to implement and test the system in a real image transmission scenario under the influence of AWGN noise. The results were obtained through graphs (time and frequency), constellation diagrams, and the Bit Error Rate (BER). The software approach is feasible and effective for the practical study of telecommunications systems, highlighting the advantages of the OFDM technique and assisting in the implementation of several future works in the area.

Keywords: OFDM. Gnu Radio. Software Defined Radio. IEEE 802.11a Standard. Bit Error Rate.

Lista de ilustrações

Figura 1 – Diagrama de Blocos de um Sistema de Comunicação Digital	17
Figura 2 – Multiplexação por Divisão na Frequência (FDM)	19
Figura 3 – Comparação da eficiência espectral entre FDM e OFDM	19
Figura 4 – Espectro de frequência OFDM	20
Figura 5 – Sistema de comunicação multiportadora	20
Figura 6 – Símbolo OFDM - com adição do prefixo Cíclico	22
Figura 7 – Scrambler/Descrambler	24
Figura 8 – Transceptor SDR	25
Figura 9 – Interface GNU Radio Companion	26
Figura 10 – Blocos de variáveis	29
Figura 11 – Fluxograma Transmissor - Fonte e Integração de dados	29
Figura 12 – Fluxograma Transmissor - Modulação	30
Figura 13 – Fluxograma Transmissor - OFDM	31
Figura 14 – Fluxograma Receptor - AWGN e Sincronização	32
Figura 15 – Bloco <i>Header/Payload Demux</i>	33
Figura 16 – Diagrama Demodulação do <i>Header</i>	34
Figura 17 – Diagrama Demodulação do <i>Payload</i>	35
Figura 18 – Processo final	36
Figura 19 – Blocos de visualização	37
Figura 20 – Imagem original a ser transmitida	38
Figura 21 – Sinal Modulado OFDM	39
Figura 22 – Sinal original/recebido no tempo (ruído 30%)	40
Figura 23 – Sinal original/recebido na frequência (ruído 30%)	40
Figura 24 – Diagrama constelação Header antes da equalização	41
Figura 25 – Diagrama constelação Payload antes da equalização	42
Figura 26 – Diagrama constelação Header depois equalização sem ruído	43
Figura 27 – Diagrama constelação Payload depois equalização sem ruído	43
Figura 28 – Barra de BER	44
Figura 29 – Curva da BER em função do Ruído	45
Figura 30 – Validação entre imagens	46
Figura 31 – Diagrama completo do sistema de comunicação digital OFDM	52

Lista de tabelas

Tabela 1 – Parâmetros IEEE 802.11a de um sistema OFDM	28
Tabela 2 – Relação entre Ruído e Taxa de Erro de Bit (BER)	44

Lista de abreviaturas e siglas

AWGN	Additive White Gaussian Noise (Ruído Branco Gaussiano Aditivo)
BER	Bit Error Rate (Taxa de Erro de Bit)
BPSK	Binary Phase Shift Keying (Chaveamento de Fase Binário)
CP	Cyclic Prefix (Prefixo Cíclico)
CRC	Cyclic Redundancy Check (Verificação de Redundância Cíclica)
FDM	Frequency Division Multiplexing (Multiplexação por Divisão de Frequência)
FEC	Forward Error Correction (Código de Correção de Erro)
FFT	Fast Fourier Transform (Transformada Rápida de Fourier)
ICI	InterCarrier Interference (Interferência Interportadora)
IDFT	Inverser Discrete Fourier Transform (Transformada Discreta de Fourier)
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform (Transformada Rápida de Fourier Inversa)
ISI	InterSymbol Interference (Interferência Intersimbólica)
MIMO	Multiple-Input and Multiple-Output (Múltiplas-Entradas e Múltiplas-Saídas)
OFDM	Orthogonal Frequency-Division Multiplexing (Multiplexação Ortogonal por Divisão de Frequência)
QAM	Quadrature Amplitude Modulation (Modulação de Amplitude em Quadra- tura)
RF	Radio Frequency (Frequência de Rádio)
SDR	Software Defined Radio (Rádio Definido por Software)
SNR	Signal-to-Noise Ratio (Relação Sinal-Ruído)

Lista de símbolos

N	Número de subportadoras (Tamanho da FFT)
T_s	Período total do símbolo OFDM
T_g	Intervalo de guarda (Prefixo Cíclico)
Δf	Espaçamento entre as subportadoras
$x(t)$	Sinal no domínio do tempo contínuo
$x[n]$	Sinal no domínio do tempo discreto (amostrado)
X_k	Símbolo de dados na k-ésima subportadora
$g(X)$	Polinômio gerador do CRC/Scrambler
ν	Número de amostras do prefixo cíclico

Sumário

1 –INTRODUÇÃO	15
1.1 Objetivo Geral	16
1.2 Objetivos Específicos	16
2 –FUNDAMENTAÇÃO TEÓRICA	17
2.1 Introdução aos Sistemas de Comunicação Digital	17
2.2 OFDM (Orthogonal Frequency-Division Multiplexing)	18
2.2.1 Sistema Comunicação Digital OFDM	20
2.3 Integração de dados	23
2.3.1 CRC - Código de Checagem por Redundância Cíclica	23
2.3.2 Embaralhador	23
2.4 SDR - Software Defined Radio	24
2.5 GNU Radio	26
2.5.1 GNU Radio Companion	26
3 –METODOLOGIA	28
3.1 Implementação do Sistema OFDM	28
3.1.1 Variáveis	28
3.1.2 Fluxograma de transmissão	29
3.1.3 Fluxograma de recepção	32
3.1.3.1 Demodulação Header	34
3.1.3.2 Demodulação Payload	35
3.1.4 Blocos de visualização	36
4 –RESULTADOS	38
4.1 Análise espectral tempo e frequência	38
4.2 Análise das constelações	41
4.3 BER	44
4.4 Validação da imagem	45
5 –CONCLUSÃO	47
5.1 Trabalhos Futuros	47
Referências	49

Anexos	51
ANEXO A –Diagrama completo do sistema	52

1 INTRODUÇÃO

Ao longo dos anos, os sistemas de telecomunicações passaram por transformações significativas para atender às novas demandas de serviços. Essa evolução não apenas melhorou a qualidade das comunicações, mas também impulsionou o surgimento de novas aplicações. A transição de redes da quarta geração (4G) para a quinta geração (5G) e com perspectivas futuras para a sexta geração (6G), não representa apenas o aumento da taxa de transmissão de dados, mas também uma nova forma de arquitetura de conectividade, impulsionada pela demanda massiva dos usuários. Segundo [Jiang et al. \(2021\)](#), para suportar esse avanço, se faz necessário uma melhoria contínua em técnicas para eficiência na rede, nomeadamente a eficiência nos custos, eficiência energética, eficiência de espectro e eficiência operacional do sistema.

No centro dessa evolução consolidou-se a técnica Multiplexação Ortogonal por Divisão de Frequência (OFDM). De acordo com o estudo de [Hwang et al. \(2009\)](#), na técnica OFDM todo o canal é dividido em vários subcanais de banda estreita, que são transmitidos em paralelo para manter a transmissão de alta taxa de dados e, ao mesmo tempo, para aumentar a duração de símbolo para combater a ISI (*Intersymbolic Interference*). Embora seus conceitos matemáticos sejam de décadas passadas, sua capacidade de lidar com desvanecimento seletivo em frequência e combater a ISI, tornou essa técnica uma escolha dominante para os padrões de banda larga atualmente. A união da técnica OFDM e uma antena MIMO (*Multi-Input Multi-Output*) ativa de pequena escala governou o mundo das redes de acesso móvel durante toda a era do 4G, e ainda mostra continuamente o seu domínio nos progressos recentes do 5G ([JIANG et al., 2021](#)), sendo hoje amplamente empregada em tecnologias como Wi-Fi, LTE e radiodifusão digital.

O crescente avanço dos sistemas e dessas técnicas modernas apresentam limitações impostas sobre o *hardware* do rádio tradicional, geralmente dedicado a uma única função. Como solução para este gargalo, surgiu o conceito de Rádio Definido por Software (SDR - *Software Defined Radio*). Proposto por [Mitola \(1995\)](#), o SDR busca substituir dispositivos baseados em rádio de características físicas por software, rodando em computadores ou dispositivos embarcados. Tal substituição é realizada por meio de uma combinação de técnicas que inclui antenas multibanda e conversão de RF; conversão de banda larga analógico para digital (A/D) e digital para analógico (D/A) (conversão A/D/A); e a implementação de funções de processamento, banda base e fluxo de bits em processadores programáveis de uso geral.

Resumidamente, os processamentos de sinais como modulação, filtragem e codificação em vez de serem feitos em circuitos analógicos fixos são transferidos para um

software para serem processados digitalmente. Essa abordagem reduziu custos, acelerou o ciclo de desenvolvimento e democratizou o acesso à prototipagem de sistemas complexos (ULVERSOY, 2020).

Devido a flexibilidade de implementação por *software* imposta aos sistemas de comunicação, surgiram ferramentas de ambiente integrado de desenvolvimento e simulação como o GNU Radio Companion. A plataforma através de sua interface gráfica e de sua ampla biblioteca de blocos de processamento digital de sinais, permite a construção, teste e validação de transceptores completos de forma rápida e acessível, tornando-se uma ferramenta indispensável para pesquisa e ensino em telecomunicações (MOURA; STEWART, 2023; GNU Radio, 2025);

1.1 Objetivo Geral

Este trabalho consiste na implementação e avaliação de um sistema de comunicação digital OFDM, incluindo o transmissor e receptor, de acordo com os parâmetros do padrão IEEE 802.11a. O sistema foi desenvolvido, utilizando a interface gráfica do GNU Radio Companion.

1.2 Objetivos Específicos

- Estudar os fundamentos teóricos da modulação OFDM, com ênfase na ortogonalidade entre subportadoras e no uso do prefixo cíclico ;
- Desenvolver a cadeia de transmissão (Tx) e recepção (Rx) no ambiente de simulação GNU Radio Companion, implementado por diagrama de blocos de funções específicas;
- Implementar algoritmos de modulação digital, sincronização de tempo e frequência e equalização de canal para corrigir as distorções introduzidas pelo meio;
- Avaliar a robustez do sistema através da análise da Taxa de Erro de Bit (BER);
- Validar o funcionamento prático do transceptor através da transmissão e recuperação de arquivos de imagem, comparando a qualidade visual com as métricas obtidas.

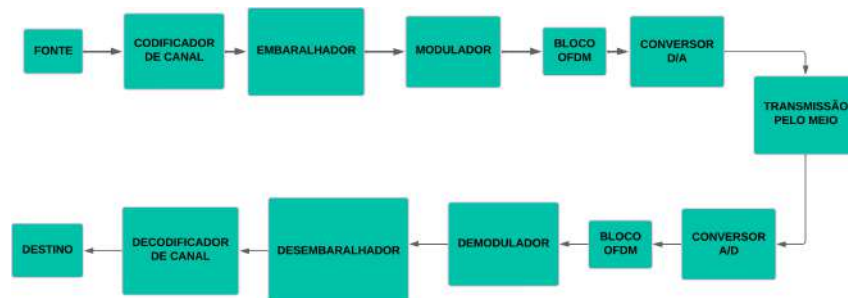
2 FUNDAMENTAÇÃO TEÓRICA

2.1 Introdução aos Sistemas de Comunicação Digital

Um sistema de comunicação digital tem como objetivo transmitir informação de forma confiável entre um transmissor e um receptor, utilizando sinais digitais. Esses sistemas estão presentes em nosso cotidiano em tecnologias como Wi-Fi, celulares, TV digital e transmissão de dados pela internet.

Resumidamente, o processo envolve a conversão de uma informação como voz e vídeo em uma sequência de bits, que são então processados, modulados e enviados por um meio de transmissão podendo ser via cabo, fibra óptica ou ar. Para ilustrar de forma simplificada, um sistema digital típico é composto por blocos funcionais sequenciais, conforme mostrado no diagrama de blocos geral da Figura 1:

Figura 1 – Diagrama de Blocos de um Sistema de Comunicação Digital



Fonte: : (Próprio Autor, 2026).

O fluxo básico de um sistema de comunicação digital é dividido entre transmissor e receptor. No transmissor, a fonte gera a informação original, que passa por um codificador de canal que adiciona redundância aos dados e permite possíveis correções de erros. Em seguida, o embaralhador reorganiza a sequência de bits para distribuir possíveis erros em rajada. O modulador converte os bits em símbolos adequados para transmissão, os quais são processados pelo bloco OFDM, onde ocorre a divisão em subportadoras ortogonais, transformação para o domínio do tempo e inserção do prefixo cíclico. Por fim, o conversor D/A transforma o sinal digital em analógico para envio pelo meio físico.

No receptor, o processo é invertido, o conversor A/D amostra o sinal recebido; o bloco OFDM remove o prefixo cíclico e aplica a FFT para retornar ao domínio da frequência; o demodulador recupera os bits a partir dos símbolos; o desembaralhador restaura a ordem original dos bits; o decodificador de canal corrige eventuais erros utilizando a

redundância inserida no transmissor; e, finalmente, a informação é entregue ao destino. Esse tipo de sistema utilizando OFDM é amplamente utilizado em comunicações sem fio e com fio de alta velocidade. Sendo assim, cada um desses blocos será detalhado e analisado ao longo deste capítulo, com ênfase nos princípios de funcionamento, implementação e relevância para sistemas modernos como o OFDM.

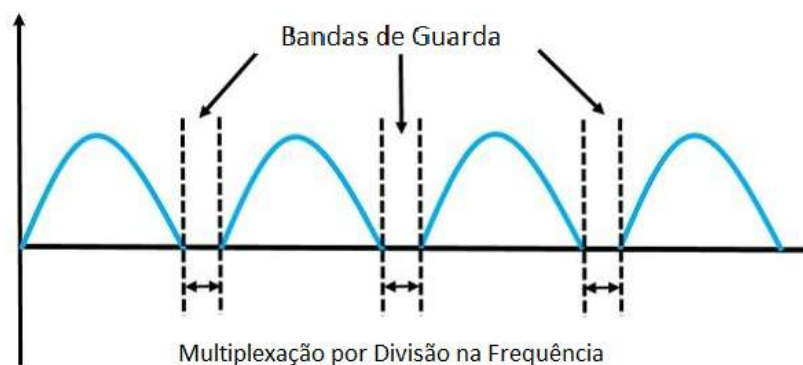
2.2 OFDM (Orthogonal Frequency-Division Multiplexing)

No cenário de sistemas de comunicação digital, a distorção linear do canal representa uma barreira significativa para a integridade do sinal. Essa distorção é degradante aos sistemas de comunicações, pois introduz o que é conhecido como interferência intersimbólica (ISI — *InterSymbol Interference*). Esta interferência manifesta-se quando o canal apresenta resposta impulsiva de duração superior ao período de símbolo, fazendo com que a contribuição de um pulso se estenda além do intervalo temporal que lhe foi atribuído (LATHI; DING, 2012). Consequentemente, os símbolos próximos interferem uns com os outros, aumentando a taxa de erro no receptor.

Em transmissões de portadora única, a ISI é um dos fatores que causa limitações aos canais de comunicação, pois a dispersão temporal é geralmente muito maior que o inverso da taxa de símbolos e, portanto, a ISI resulta das características de resposta de frequência não ideais para esse tipo de canal (PROAKIS; SALEHI, 2008). A solução convencional recai sobre equalizadores adaptativos para compensar essa dispersão no canal. Contudo, como observado em Proakis e Salehi (2008), a complexidade computacional do receptor cresce linearmente com o número de coeficientes exigidos, implicando aumento de custo de hardware e consumo de potência.

Como alternativa a essa complexidade de combate a ISI, surgiu o conceito de FDM (*Frequency Division Multiplexing*). Um método de transmissão de vários sinais simultaneamente, em faixas de frequências que não se sobrepõem (LATHI; DING, 2012). Nessa técnica, cada sinal é modulado por uma frequência portadora diferente. Essas portadoras, referidas como subportadoras, são adequadamente separadas para evitar sobreposição (interferência) dos espectros dos vários sinais modulados (LATHI; DING, 2012). Porém, na Figura 2 é mostrado que isso requer bandas de guarda, o que resulta em uma largura de banda ampla, recurso esse limitado em um sistema de comunicação, surgindo a necessidade de transmitir uma maior quantidade de informação com uma menor largura de banda.

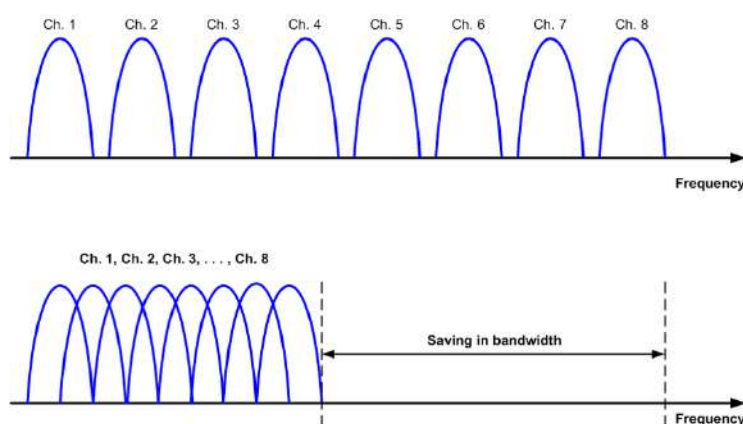
Figura 2 – Multiplexação por Divisão na Frequência (FDM)



Fonte: Imagem adaptada de [Tech Differences](#) (2018).

Diante da limitação e da ineficiência das bandas de guarda do FDM, o OFDM consolidou-se como uma solução predominante. A técnica de transmissão surgiu em 1968, desenvolvida por Chang e Gibby, e foi patenteada, posteriormente, nos Estados Unidos ([WEINSTEIN, 2009](#)). O OFDM é um tipo especial de modulação multiportadora em que as subportadoras dos subcanais correspondentes são mutuamente ortogonais ([PROAKIS; SALEHI, 2008](#)). A Figura 3 ilustra a eficiência dessa técnica comparada à técnica FDM, demonstrando a economia no espectro obtida.

Figura 3 – Comparação da eficiência espectral entre FDM e OFDM

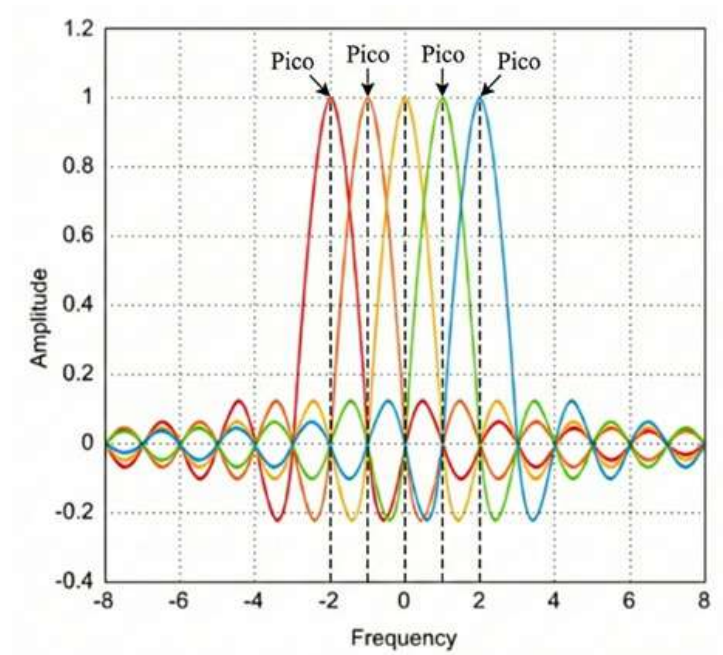


Fonte: [Mahmood e Newagy](#) (2015).

Entretanto, como o espectro de uma subportadora tem a forma da função sinc, Figura 4, possuindo lóbulos laterais, acaba ocasionando uma alta sensibilidade para deslocamento em frequência e ruído de fase no sistema, sendo uma das principais desvantagens da técnica. Segundo [Armstrong](#) (2009), se essa ortogonalidade entre as subportadoras é

perdida, o resultado é uma interferência entre elas. Em contrapartida, o uso de algoritmos robustos de sincronização de tempo e frequência são implementados no sistema para correção desses erros entre as subportadoras.

Figura 4 – Espectro de frequência OFDM

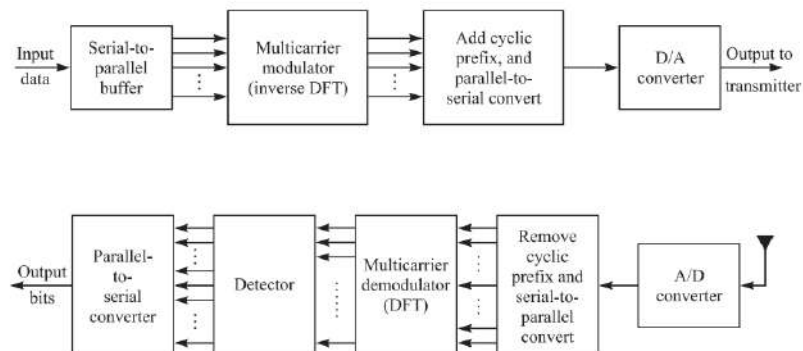


Fonte: Imagem adaptada [YateBTS \(2026\)](#).

2.2.1 Sistema Comunicação Digital OFDM

O seguinte esquema de blocos é apresentado na Figura 5 para demonstrar o processo de implementação de um sistema completo de comunicação OFDM.

Figura 5 – Sistema de comunicação multiportadora



Fonte: [Proakis e Salehi \(2008\)](#).

Inicialmente, antes de seguir para o buffer S/P, o fluxo de dados é modulado digitalmente em modulações específicas. Os tamanhos das constelações mais usadas em conjunto com OFDM variam de 4QAM a 64QAM, embora a PSK seja compatível com OFDM (ARMSTRONG, 2009). Neste trabalho foi proposto o uso de dois tipos de modulação: a BPSK (*Binary Phase Shift Keying*) e a 16QAM (*16-Quadrature Amplitude Modulation*). O motivo disso se dá pelo fato do fluxo de bits ser feito em duas partes: cabeçalho (*Header*) e carga útil (*Payload*). O cabeçalho contém informações de controle, modulado em BPSK, enquanto a carga útil em 16QAM, priorizando a taxa de transmissão de dados.

Após o mapeamento, o fluxo é agrupado novamente (N) para ser feita a conversão (S/P). Esse bloco forma um vetor complexo de tamanho N (que representa o número de subportadoras), denotado por:

$$\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]^T, \quad (1)$$

em que cada elemento deste vetor representa o símbolo de dados a ser transmitido na k -ésima subportadora (ARMSTRONG, 2009).

Seguindo o diagrama, o vetor de símbolos paralelos no domínio da frequência é processado por um bloco de IFFT (*Inverse Fast Fourier Transform*), responsável por converter o vetor de símbolos da frequência para o tempo. A operação da IFFT é matematicamente definida pela expressão para cada amostra de saída $x(t)$ em banda base no tempo contínuo, dada por (PROAKIS; SALEHI, 2008):

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi k \Delta f t}, \quad 0 \leq t < T_s, \quad (2)$$

onde $\Delta f = 1/T_s$ é o espaçamento entre as subportadoras ideal para garantir ortogonalidade.

Em implementações em sistemas digitais, deve ser feita a discretização do sinal antes de prosseguir no sistema. Ao amostrar o sinal com período $T_a = T_s/N$, obtém-se a expressão para a n -ésima amostra temporal $x[n]$ (BAHAI et al., 2004):

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi \frac{kn}{N}}, \quad \text{para } 0 \leq n \leq N-1, \quad (3)$$

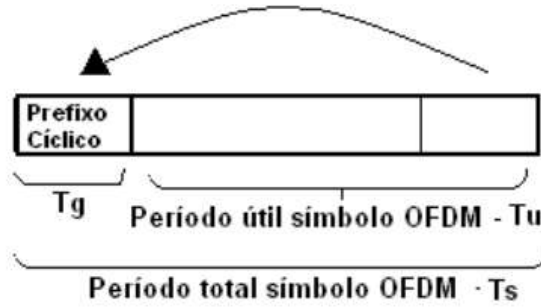
a Equação 3 é a exata definição da Transformada Discreta de Fourier Inversa (IDFT). Assim, a modulação do bloco é realizada de maneira eficaz utilizando o algoritmo IFFT (Transformada Rápida de Fourier Inversa).

Consequentemente, a saída do bloco IFFT se dá por outro vetor complexo, contudo no domínio do tempo, $x = [x_0, x_1, x_2, \dots, x_{N-1}]^T$, subsequente a isso o sinal é submetido

a inserção do prefixo cíclico. Segundo [Proakis e Salehi \(2008\)](#), o prefixo cíclico consiste em adicionar uma cópia das últimas amostras ν fornecidas pelo bloco IFFT ao início do mesmo bloco, a Figura 6 ilustra essa inserção. Como o vetor no domínio do tempo é dado por $x = [x_0, x_1, x_2, \dots, x_{N-1}]^T$, ao se adicionar as cópias tem-se a equação:

$$\mathbf{x}_{cp} = [x_{N-\nu}, \dots, x_{N-1}, x_0, x_1, \dots, x_{N-1}]^T. \quad (4)$$

Figura 6 – Símbolo OFDM - com adição do prefixo Cíclico



Fonte: [Arndt \(2012\)](#).

Essa ação realiza duas funções essenciais, elimina tanto a ISI quanto a interferência interportadora (ICI) do sinal recebido e é a chave para a equalização simples em OFDM ([ARMSTRONG, 2009](#)). O comprimento do prefixo cíclico pode ser 1/32, 1/16, 1/8 ou 1/4 do comprimento da FFT ([LATHI; DING, 2012](#)). Após essa inserção, o sinal é condicionado a um buffer P/S para ser convertido de digital para analógico e ser transmitido.

Para a recepção do sinal seguindo o diagrama, tem-se a operação inversa do transmissor para extração da informação.

Antes de iniciar a cadeia de processos do OFDM é feito uma conversão analógica para digital para prosseguir com os demais processos. A próxima operação no receptor é o descarte do prefixo cíclico. As primeiras amostras de cada símbolo recebido são removidas, preservando-se apenas as N amostras subsequentes ([PROAKIS; SALEHI, 2008](#)).

O vetor de amostras no tempo, $\mathbf{y} = [y_0, y_1, \dots, y_{N-1}]^T$, é submetido à Transformada Discreta de Fourier, com o intuito de trazer o sinal para o domínio da frequência:

$$Y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y_n e^{-j2\pi \frac{kn}{N}}, \quad 0 \leq k \leq N-1, \quad (5)$$

esta operação matemática é inversa à realizada no transmissor Eq 3.

Devido à ortogonalidade preservada com o prefixo, a convolução linear do canal no tempo se torna uma multiplicação na frequência. Para recuperar os símbolos, aplica-se

um equalizador. Essa simplicidade na equalização é a principal vantagem matemática do OFDM ([PROAKIS; SALEHI, 2008](#)).

Por fim, o fluxo é então convertido de paralelo para série (P/S) e os símbolos complexos estimados são submetidos ao fatiador, que realiza o demapeamento das constelações (BPSK e 16-QAM) para recuperar os bits originais.

2.3 Integração de dados

Em conjunto as técnicas de proteção contra interferências já presentes no OFDM, existem outros procedimentos adicionais que são utilizados nos sistemas de comunicação digital para oferecer mais qualidade a transmissão e recepção, como códigos de checagem de erro, embaralhadores, códigos corretores entre outros. Entretanto, nesse trabalho só houve a implementação de dois desses procedimentos, sendo eles o código de checagem de erro e o embaralhador.

2.3.1 CRC - Código de Checagem por Redundância Cíclica

Os códigos de checagem de erro mais empregados atualmente são os de redundância cíclica para detecção de erros em um sistema. Segundo [Lathi e Ding \(2012\)](#), para verificar a integridade do bloco de dados de carga (pacote), cada pacote de dados é codificado por códigos CRC de comprimento $n \leq 2^m - 1$. Os códigos CRC mais comuns têm $m = 12, 16$ ou 32. O polinômio gerador dado por:

$$g(X) = X^m + g_{m-1}X^{m-1} + \dots + g_1X + 1, \quad (6)$$

a utilização de padrão depende da aplicação, por exemplo o padrão IEEE 802.11b é verificado pela sequência de 16 bits, já o OFDM por padrão utiliza uma redundância de 32 bits definido no padrão IEEE 802.3:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1, \quad (7)$$

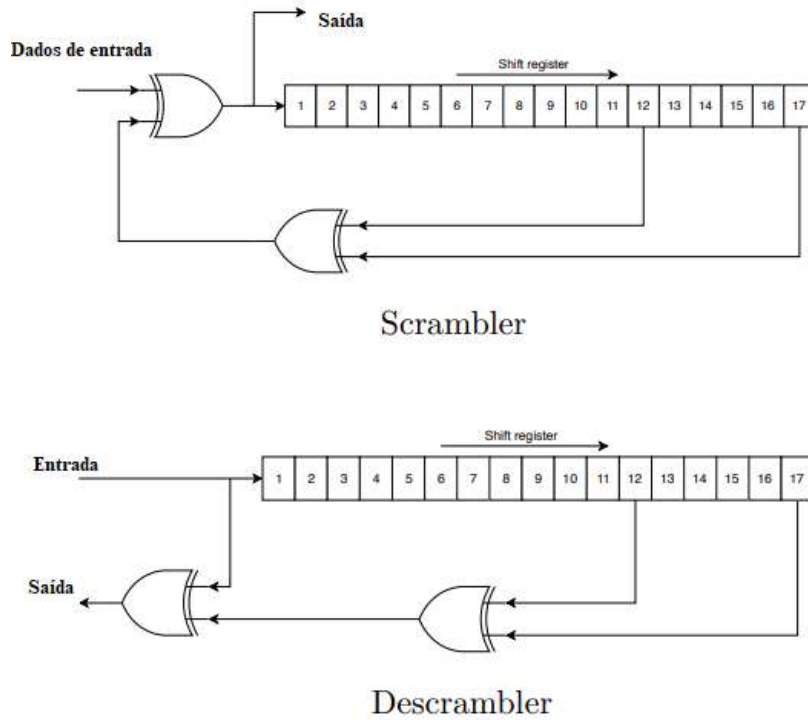
essa técnica garante que, ao final do processo de recepção, qualquer incoerência seja detectada, para assim ser corrigida por um possível código corretor de erro ou descartar o pacote corrompido.

2.3.2 Embaralhador

Em sistemas de comunicações digitais, enviar uma sequência de dados longos repetitivos, como vários '1' e '0' é arriscado, pois caso algum desses bits seja perdido durante

um processo de transmissão/recepção, há uma possibilidade de que a informação seja perdida. Entretanto o padrão OFDM utiliza *Scramblers* (Embaralhadores) para evitar essas longas sequências de dados. Segundo Bahai et al. (2004), o embaralhador ajuda a alterar o padrão de bits com uma sequência pseudoaleatória de bits, e este mesmo embaralhador é usado para desembaralhar os dados na recepção, demonstrado na Figura 7:

Figura 7 – Scrambler/Descrambler



Fonte: Imagem adaptada de [StackOverflow \(2022\)](#).

Neste trabalho, adotou-se o polinômio gerador primitivo de sétima ordem, conforme especificado pelo padrão IEEE 802.11a. Tal polinômio é definido pela Equação 8:

$$S(x) = x^7 + x^4 + 1, \quad (8)$$

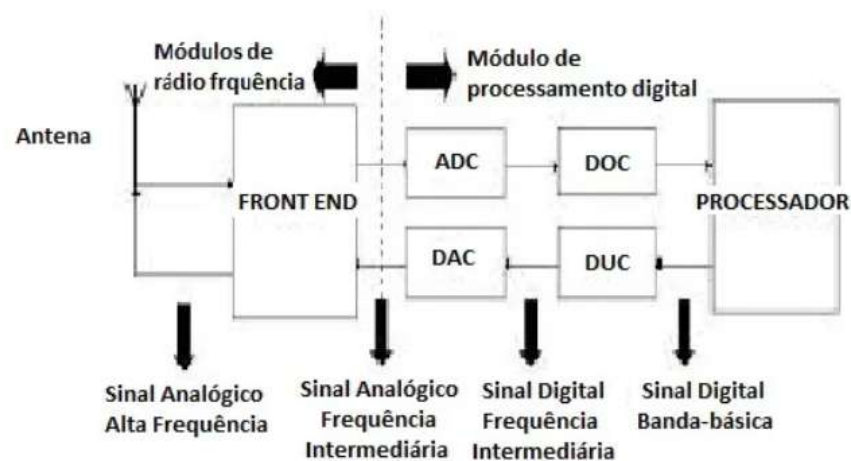
durante a transmissão, o estado inicial do embaralhador será definido como um estado pseudoaleatório diferente de zero. Os 7 bits menos significativos são definidos como todos zeros antes de embaralhar, para permitir a estimativa do estado inicial do embaralhador no receptor (BAHAI et al., 2004).

2.4 SDR - Software Defined Radio

O avanço tecnológico nos sistemas de comunicações possibilitou que componentes essenciais de um rádio, tais como filtros, moduladores, demoduladores, amplificadores e

outros, puramente baseados em hardware, pudessem ser definidos em software dentro do sistema. De acordo com [Selva et al. \(2012\)](#) o conceito de SDR (*Software Defined Radio*) permite substituir a tradicional implementação em hardware dos dispositivos de comunicação por uma implementação mais flexível, usando dispositivos programáveis controlados por software. O SDR é composto por duas partes dominantes, o *front end* de rádio que a ele são atribuídos o recebimento e a transmissão das frequências de rádio e o *back end* de rádio que a ele fica atribuído o processamento do sinal de rádio, isso é demonstrado na Figura 8 ([SOARES, 2016](#)):

Figura 8 – Transceptor SDR



Fonte: [Soares \(2016\)](#).

Além dessa reconfigurabilidade, existem diversos outros tipos de vantagens no uso dessas tecnologias. [Selva et al. \(2012\)](#) também destacam que essas soluções são mais robustas a temperaturas e envelhecimentos de componentes. Além disso, possuem uma facilidade de desenvolvimento como ferramentas para simulação e correção de erros. Exigem menos complexidade do hardware já que se limita a uma implementação de um front-end de RF (*Radio Frequency*). Isso resulta em uma alta integração, dada a eliminação, através do uso de um processador, de muitos dos elementos ativos e passivos do rádio, antes responsáveis pelo processamento e modulação do sinal. Todos esses aspectos têm influência no custo final reduzido do produto.

Portanto os projetos baseados em SDR são mais fáceis de se implementar, possuem custo reduzido e caso necessário oferecem técnicas para correções erros e simulações. Outro ponto interessante é caso seja necessário questões de manutenção e operação dos equipamentos, devido a arquitetura, os serviços podem ser feitos sem que ele precise ser desligado ou sem que a infra-estrutura deva ser alterada.

2.5 GNU Radio

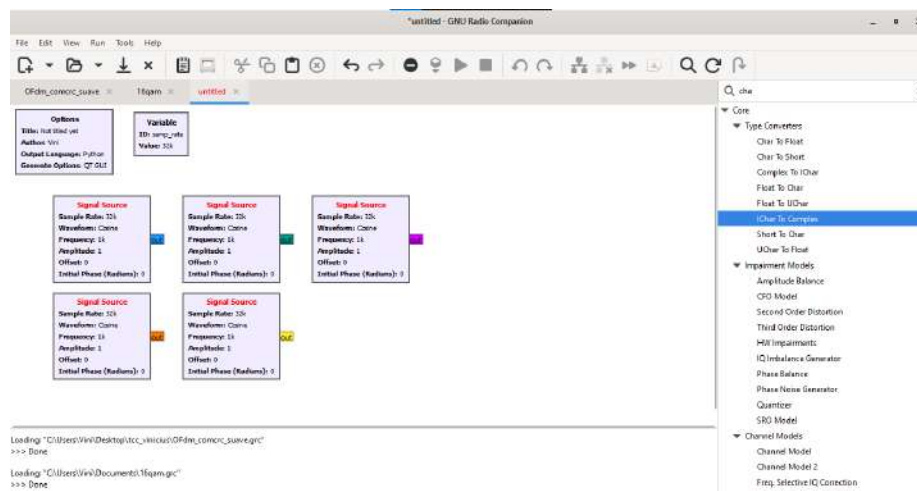
O GNU Radio é um kit de ferramentas de desenvolvimento de software livre e de código aberto que fornece blocos de processamento de sinais para implementar rádios definidos por software. Ele pode ser usado com hardware de RF externo de baixo custo e prontamente disponível para utilizar rádios definidos por software ou ser implementado em ambiente de simulação (GNU Radio, 2025).

O aplicativo tem versões para Linux, Mac e Windows e utiliza principalmente linguagem de programação Python, enquanto o processamento de sinal crítico é implementado em C++ usando extensões de ponto flutuante do processador, por exemplo para operações de I/O, FFT/IFFT, modulação/demodulação.

2.5.1 GNU Radio Companion

No intuito de facilitar a interatividade do GNU Radio com o usuário, foi disponibilizado outro aplicativo chamado GNU Radio Companion. O Software é uma interface gráfica de usuário que funciona a base de blocos, semelhante ao Simulink. Nele é permitido criar e executar aplicativos de processamento de sinais somente arrastando e soltando os blocos (GNU Radio, 2025). Entretanto, caso o usuário queira adicionar novas funcionalidades, é necessário que o mesmo escreva o próprio código, escolhendo uma das duas linguagens disponibilizadas. A Figura 9 mostra a interface programa:

Figura 9 – Interface GNU Radio Companion



Fonte: (Próprio Autor, 2026).

Os blocos tem diferentes tipos de dados, dentre eles: complexo (8 Bytes - Azul escuro), float (4 Bytes - Laranja), inteiro (4 Bytes - Azul claro), short (2 Bytes - Amarelo) e char (Rosa). Na parte à direita da interface tem-se a lista de blocos do app, na parte

inferior o terminal de saída, para verificação de erros ou de estágio completo e na parte superior as ferramentas disponíveis.

Na página oficial do Gnu Radio, são disponibilizados diversos manuais sobre os blocos e práticas existentes, para auxiliar o usuário no manuseio da ferramenta, muitas dessas práticas foram usadas de base como auxílio para montagem do sistema OFDM proposto neste trabalho.

3 METODOLOGIA

Neste capítulo será abordado os procedimentos adotados para realização do enlace ponto a ponto do sistema de comunicação digital baseado em OFDM. O desenvolvimento foi realizado integralmente no ambiente de simulação do GNU Radio Companion versão (3.10.9.2) em um sistema operacional Windows, mas podendo ser executado da mesma forma em um sistema Linux. A escolha desta ferramenta justifica-se pela sua natureza de código aberto e pela vasta biblioteca de blocos de processamento de sinal voltados para telecomunicações.

3.1 Implementação do Sistema OFDM

3.1.1 Variáveis

Para garantir que a simulação represente um cenário ideal de um sistema de comunicação OFDM, a escolha de variáveis como número de portadoras, tamanho de FFT, entre outras, foram baseados nos padrões IEEE 802.11a (Wifi), para uma banda de frequência de 10 MHz. A Tabela 1 mostra os valores adotados na simulação:

Tabela 1 – Parâmetros IEEE 802.11a de um sistema OFDM

Parâmetro	Configuração	Descrição
Tamanho da FFT	64	Total de subportadoras
Prefixo Cíclico (CP)	16 amostras	Intervalo de guarda (1/4 da FFT)
Sincronização	Sync Word 1 e 2	Preâmbulo (Schmidl & Cox)
Aleatorizador (<i>Scrambler</i>)	$S(x) = x^7 + x^4 + 1$	Polinômio gerador (127 bits)
Deteção de Erros	CRC-32	Verificação de integridade
Modulação (Constelação)	BPSK / 16QAM	Mapeamento dos símbolos
Subportadoras de Dados	48	Portadoras ativas
Subportadoras Piloto	4	Posições: -21, -7, 7, 21

Fonte: (Próprio Autor, 2026).

A Figura 10 mostra os blocos de variáveis criados no GNU Radio Companion, usando c base as informações extraídas do padrão IEEE:

O cabeçalho segue o caminho para o bloco *Protocol Formatter*, responsável por ler as *tags* gerar os bits de controle da camada física. Nele estão contidas as informações cruciais para o receptor, desde o esquema de modulação ao comprimento do pacote que está sendo enviado.

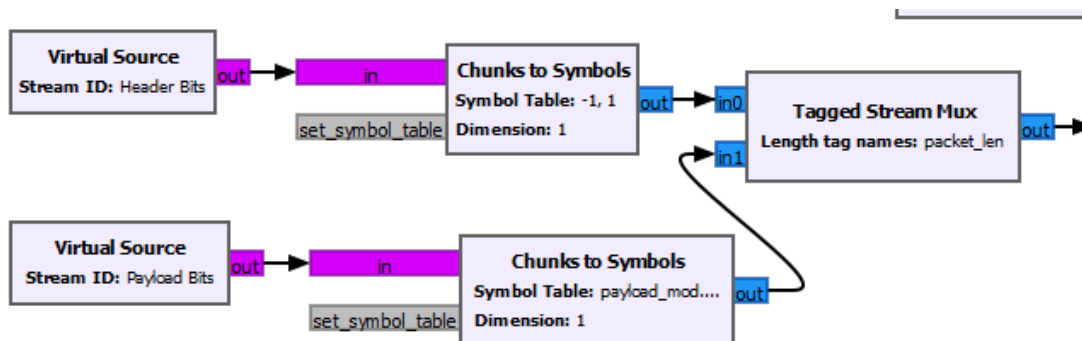
Paralelamente a ele, o payload (dados) segue para o bloco *Additive Scrambler*. Este bloco é responsável por fazer a aleatorização dos bits de dados no fluxo, isso é feito a partir de uma operação lógica XOR entre os bits de entrada e uma sequência pseudoaleatória gerada pela bloco. O scrambler foi configurado com base nos padrões adotados, usando uma máscara 0x61 e comprimento de 7 bits.

Por fim, ambos os fluxos paralelos passam pelos blocos *Repack Bits*. Com a finalidade de reagrupar os bits de entrada (originalmente 8 bits/byte) para a quantidade de bits exigida pelo esquema de modulação consecutivamente. O cabeçalho é preparado para a modulação BPSK(1 bit/byte), enquanto o payload para a modulação 16QAM (4 bit/byte).

Dentro desse fluxo existem blocos denominados *virtual Sink*, esses blocos servem como um ponto de conexão virtual com outro bloco chamado de *Virtual Source*. A conexão é estabelecida através da atribuição de um identificador único (*Stream ID*) comum a ambos os blocos, isso é útil para organizar um fluxograma complexo, evitando longa conexões durante o processo. As saídas entrada, entrada1 e entrada2, são usadas em blocos de visualização, que vão ser vistos futuramente, já as saídas *Header Bits* e *Payload Bits* são as que vão dar continuidade ao fluxo do sistema.

A seguir a Figura 12 demonstra a continuação do fluxo do transmissor, onde os fluxos de bits, anteriormente separados e formatados, são recebidos através dos blocos (*Virtual Sources*) de mesma ID dos blocos *Virtual Sink* vistos anteriormente e encaminhados para a etapa de mapeamento.

Figura 12 – Fluxograma Transmissor - Modulação



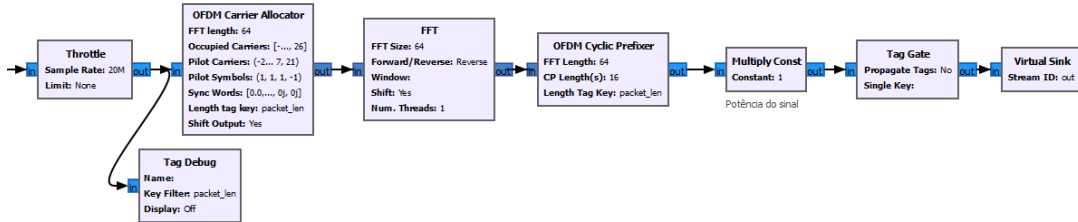
Fonte: (Próprio Autor, 2026).

No mapeamento, os bits são convertidos em símbolos complexos pelos blocos *Chunks to Symbols*. O cabeçalho como dito anteriormente foi mapeado em uma constelação BPSK, com uma tabela de símbolos definida por $[-1, 1]$, já o payload foi mapeado em uma constelação 16QAM possuindo 16 símbolos definidos por padrão.

Após o mapeamento os dois fluxos entram no bloco *Tagged Stream Mux*. A função que esse bloco tem é unir pacotes inteiros das diferentes entradas e os colocar um atrás do outro em uma única saída, baseando-se nas etiquetas de comprimento (*stream tags*) anexadas aos pacotes anteriormente, ou seja, ele gera um único fluxo onde os símbolos do header são seguidos pelos símbolos do payload. Sendo fundamental para estabelecer a estrutura física do pacote OFDM, garantindo que as informações de controle precedam os dados antes da etapa de alocação de portadora.

O sinal de saída segue então para o processo de OFDM, visto na Figura 13, mas antes de prosseguir para a alocação de portadores, o sinal é submetido ao bloco *Throttle* que serve como um limitador de taxa de amostras, controlando a velocidade com que os dados são processados, configurado para uma taxa de amostragem de 20 MHz.

Figura 13 – Fluxograma Transmissor - OFDM



Fonte: (Próprio Autor, 2026).

Com o sinal controlado, seguiu-se para o bloco *OFDM Carrier Allocator*. Nesta etapa o fluxo de símbolos complexos é convertido em vetores paralelos compatíveis com a Transformada Rápida de Fourier. Sua função principal é mapear o símbolos de entrada em subportadoras ortogonais, especificamente 64 seguindo o padrão adotado. Destas 64 portadoras, 48 são alocadas para transmissão de dados e 4 (posições ± 7 e ± 21) são reservadas para a inserção dos símbolos pilotos. Adicionalmente, o bloco insere o preâmbulo no início de cada quadro e anula as subportadoras restantes nas bordas e na componente contínua (DC).

Uma vez estruturado o vetor no domínio da frequência, o sinal é submetido ao bloco FFT, configurado para operar no modo reverso (IFFT - Inverse Fast Fourier Transform). Esta etapa realiza a operação matemática de modulação propriamente dita, convertendo os vetores de subportadoras ortogonais em símbolos OFDM no domínio do tempo.

Após a conversão para o domínio do tempo, o fluxo de amostras atravessa o bloco *OFDM Cyclic Prefix*. Este estágio insere um intervalo de guarda entre os símbolos consecutivos. Seguindo o padrão IEEE, o bloco foi configurado para utilizar as últimas 16 cópias de amostras do símbolo e replicando no início do mesmo, resultando em um prefixo cíclico com duração de 1/4 do tempo de símbolo.

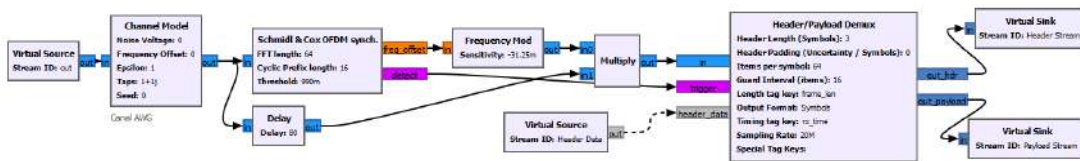
Finalizada a modulação do sinal em OFDM, o mesmo segue para o bloco *Multiply Const*, esse bloco atua como um controlador de ganho do transmissor, permitindo o ajuste e amplitude do sinal. Embora esteja configurado com valor unitário (1), sua presença na arquitetura permite emular diferentes potências de transmissão para testes de relação sinal-ruído (SNR).

Em sequência, o sinal é enchaminhado para o bloco *Tag Gate*. A sua função é suprimir as tags do fluxo, impedindo que as mesmas se propaguem para o modelo de canal. Por fim, após esse condicionamento o sinal segue para o bloco de entrada virtual com ID (*out*), este é ponto que conecta a saída do transmissor OFDM diretamente à entrada do *Channel Model* no receptor, concluindo o fluxo de transmissão OFDM.

3.1.3 Fluxograma de recepção

A Figura 14, mostra o início do processo de recepção do sinal, diferente da transmissão, o sinal de recepção apresenta-se com uma degradação sofrida por um ruído (AWGN), no intuito de simular um canal real de transmissão que sofre perdas ao longo do caminho, isso é possível graças ao bloco *Channel Model*.

Figura 14 – Fluxograma Receptor - AWGN e Sincronização



Fonte: (Próprio Autor, 2026).

Devido a essa degradação exposta, o sinal sofre uma dessincronização o que leva à perda da ortogonalidade entre as portadoras, resultando em uma interferência entre elas. Para resolver esse problema, o sinal é submetido a um bloco de sincronização em frequência e tempo, chamado de *Schmidl & Cox OFDM Synch*. Esta técnica foi proposta por Schmidl e Cox (1997), ela permite tanto a identificação do ponto inicial dos símbolos no domínio do tempo discreto como a correção do erro de frequência de portadora, por

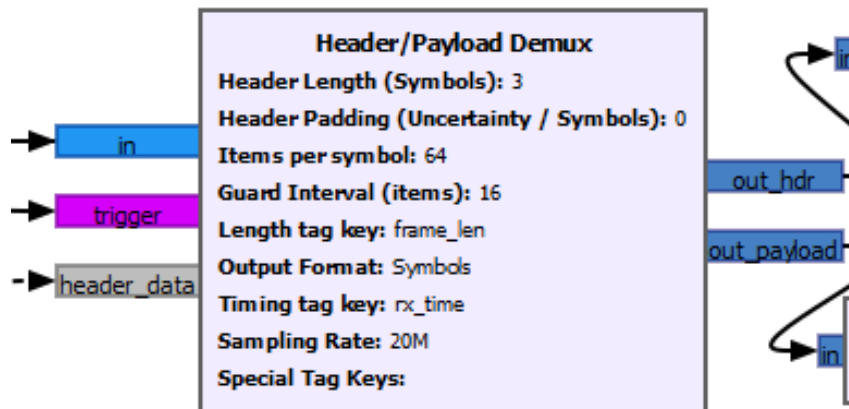
um algoritmo de correlação. O bloco é configurado com FFT e prefixo cíclico usado no canal e o *threshold* usando configuração padrão 0,9.

Para executar a correção, o sistema emprega um laço de realimentação com os blocos *Frequency Mod*, *Multiply* e *Delay*. O *Frequency Mod*, tem como intuito converter o erro escalar da frequência em um sinal complexo exponencial, com uma sensibilidade de fase ajustada para $-2/N_{\text{FFT}}$.

O bloco *Multiply*, faz a multiplicação desse sinal estimado em frequência com um bloco *Delay*, configurado com 80 amostras. Este atraso é intencional e matematicamente calculado como a soma do tamanho da FFT (64) com o Prefixo Cíclico (16), compensando a latência de processamento do algoritmo de detecção e garantindo que, quando o sinal de correção de frequência chegar ao multiplicador, ele esteja perfeitamente alinhado temporalmente com o símbolo OFDM correspondente no fluxo de dados.

Feito esta correção, o fluxo de dados segue para o bloco *Header/Payload Demux*. A função desse componente é separar o fluxo de entrada em dois caminhos de processamento distintos, ou seja, separar os dados em Header e Payload novamente, para isso este bloco tem seu funcionamento com base na interação de três entradas distintas *in*, *trigger* e *header_data*, visto na Figura 15.

Figura 15 – Bloco *Header/Payload Demux*



Fonte: (Próprio Autor, 2026).

A porta identificado como *in*, irá receber o sinal contínuo complexo já corrigido e sincronizado em frequência. Porém, o sinal ainda apresenta um encadeamento dos símbolos, misturando tanto as informações de controle como e dados. Para processar esse

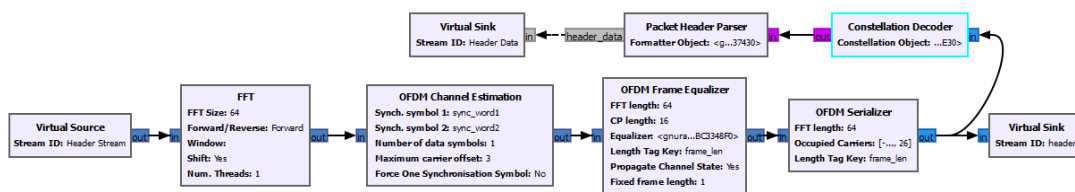
fluxo, existe uma dependência da porta *trigger*, que está conectada diretamente a saída do algoritmo de Schmidl & Cox, monitorando o recebimento do sinal de *trigger*, que indica o instante de início do quadro. Ao detectar esse instante, o demultiplexador é acionado, capturando os primeiros símbolos do fluxo e direcionando-os para a saída do Header.

Por fim, o controle final ocorre pela porta de realimentação *header data*, sua função é receber os dados referentes ao tamanho do pacote, informação essa que vai ser vista posteriormente a medida que for feita demodulação do cabeçalho no sistema. É através do recebimento desta mensagem que o bloco determina a quantidade exata de símbolos subsequentes que correspondem à carga útil, permitindo o roteamento preciso para a saída de dados *payload* até o final do pacote. Desta forma, garante-se que o Header e o Payload, sejam separados e encaminhados corretamente para suas respectivas cadeias de demodulação. Além disso, este bloco tem a função crucial de retirar o prefixo cíclico, configurado pelo parâmetro *Guard interval*, ele automaticamente identifica e descarta as amostras pertencentes ao intervalo de guarda, deixando as amostras preparadas pra os processos subsquente de demodulação.

3.1.3.1 Demodulação Header

A Figura 16 representa a cadeia de demodulação do cabeçalho separado anteriormente pelo bloco *Header/Payload Demux*, o resultado dessa cadeia serve de realimentação para entrada *header data* do mesmo.

Figura 16 – Diagrama Demodulação do *Header*



Fonte: (Próprio Autor, 2026).

A primeira etapa do processo de demodulação do sinal OFDM, consiste em fazer a transformada inversa do tempo para a frequência dos símbolos recebidos, isso é feito através do mesmo bloco utilizado no transmissor *FFT*, configurado para o parâmetro *Forward* com 64 pontos.

Após a transformada, os vetores de frequência passam pelo bloco *OFDM Channel Estimation*, este bloco é responsável por calcular a resposta impulsiva dentro do canal. O funcionamento do bloco se baseia no uso de palavras de sincronização, posicionadas

no início do preâmbulo, comparando o sinal degradado com a referência sem distorção armazenada. Com base nessa comparação, o estimador determina a distorção de fase e amplitude inserida no sistema.

Essas informações são passadas para o bloco subsequente *OFDM Frame Equalizer*, que tem como função mitigar essas distorções em fase e amplitude, recuperando a integridade do sinal. Em seguida, ele realiza a equalização em uma ou duas dimensões em um quadro OFDM marcado.

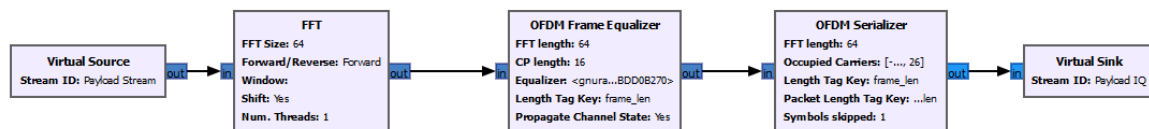
Com as subportadoras equalizadas, as mesmas seguem para o bloco *OFDM Serializer*, com trabalho de transformar os vetores paralelos de dados em um fluxo contínuo serial de símbolos complexos. Este fluxo Serial é então submetido ao bloco *Constellation Decoder*, configurado como objeto de constelação BPSK, realizando a decisão bruta e decodificando os pontos de uma constelação de um espaço complexo para bits (descompactados) com base no mapa do objeto.

Seguindo, o processo final da cadeia é o bloco *Packet Header Parser*. Seu funcionamento é o inverso do bloco *Protocol Formatter*, nele o valor numérico correspondente ao campo de comprimento (*packet length*) é extraído e o convertido em uma mensagem de controle assíncrona, utilizada para realimentação do bloco *Header/Payload Demux* pelo bloco *Virtual Sink*, fechando o laço de controle do receptor e informando ao sistema quantos símbolos de carga útil devem ser processados.

3.1.3.2 Demodulação Payload

Na demodulação do Payload, o processo inicial segue o mesmo padrão da cadeia anterior. Porém, nesse processo não é utilizado o bloco *OFDM Channel Estimation*, já que esse fluxo não possui palavras de sincronização. Seguindo, o sinal é submetido aos mesmos processos, transformada do tempo para frequência, equalização do canal e serialização do fluxo. Mostrado na Figura 17:

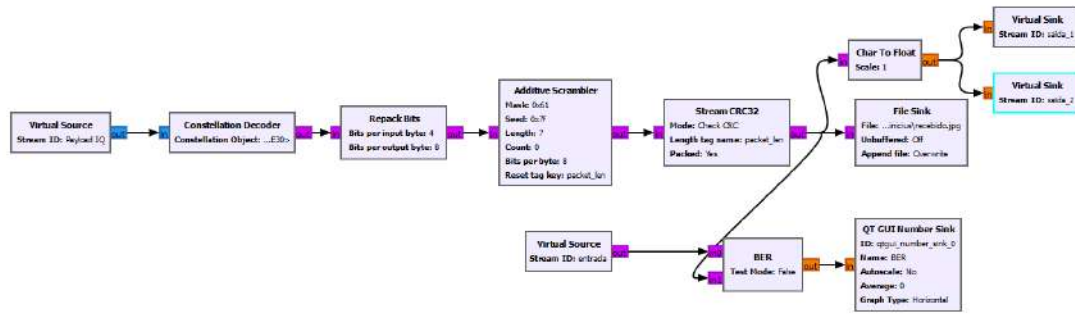
Figura 17 – Diagrama Demodulação do *Payload*



Fonte: (Próprio Autor, 2026).

Após essas etapas, o fluxo é enviado para uma saída virtual para completar os últimos estágios da OFDM, que consiste em decodificar a constelação, desembaralhar e retirar verificar o código de checagem, demonstrado na Figura 18.

Figura 18 – Processo final



Fonte: (Próprio Autor, 2026).

O fluxo passa pelo *Constellation Decoder*, configurado como objeto de constelação 16QAM, realizando a decisão bruta e decodificando os pontos de uma constelação de um espaço complexo para bits. Faz-se necessário transformar a saída de 4 bits do 16QAM para 8 bits, já que essa é a estrutura geral do arquivo original, isso é feito utilizando o bloco *Repack Bits*, já visto no transmissor.

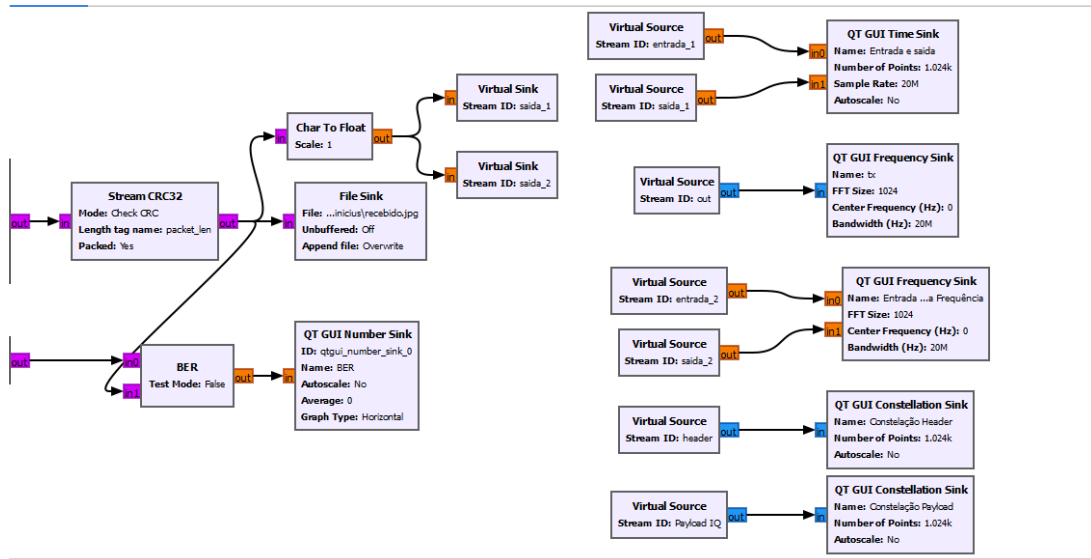
Após essa conversão, a sequência de bits foi desembaralhada utilizando o bloco *Additive Scrambler*, na função desembaralhar e usando os mesmos parâmetros do transmissor. A sequência desembaralhada é submetida ao *Stream CRC32* operando no modo *Check CRC*, recalculando a soma de verificação do pacote recebido e comparando com o código CRC anexado ao quadro. A saída desse bloco segue para o bloco *File Sink*, sua função é armazenar o arquivo de saída no local determinado, finalizando toda a cadeia do sistema OFDM completamente.

Os demais blocos de *Virtual Sink*, *Virtual Source*, *BER* e *QT GUI Number Sink* são para visualização e serão explicados na próxima seção.

3.1.4 Blocos de visualização

Na visualização dos resultados, o próprio GNU Radio Companion oferece uma gama de blocos de visualizações, seja em frequência, tempo, constelação e até mesmo verificar a BER do sistema. Durante todo o sistema foram postos blocos de saídas virtuais para capturar pontos na transmissão e recepção que se desejava visualizar, como mostra a Figura 19:

Figura 19 – Blocos de visualização



Fonte: (Próprio Autor, 2026).

Os blocos de *QT GUI Time sink* serve para ver o sinal no domínio do tempo em saídas especificadas. O *QT GUI Frequency sink*, serve para ver no domínio da frequência, já o *QT GUI Constellation sink* para visualizar a constalação modulada (BPSK ou 16QAM). Nesse sistema também foi utilizado o bloco *BER* junto com o *QT GUI Number sink*, para adquirir a taxa de erro de bit durante todo o processo de envio de informação.

4 RESULTADOS

Objetivando analisar o desempenho da recepção do sinal modulado em OFDM implementado via software pelo GNU Radio Companion, foram realizados testes simulando um cenário real de transmissão com ruído aditivo (AWGN). A verificação foi separada em quatro parâmetros de perspectiva, sendo eles análise espectral no tempo e na frequência do sinal, constelações utilizadas, taxa de erro de bit (BER) e validação da imagem recebida e armazenada.

4.1 Análise espectral tempo e frequência

Utilizando os blocos *QT GUI Time Sink* e *QT GUI Frequency Sink* mostrados na seção 3.1.4, foi possível visualizar o comportamento do sinal OFDM. Como sinal de entrada original, foi escolhida a imagem “Lena”, como mostra a Figura 20, muito usada como referência em processamento de imagem e comunicações digitais. A imagem tem resolução de 512×512 pixels, com cada pixel mostrado em níveis de cinza de 8 bits, resultando em 256 tons possíveis. Essa representação em matriz de pixels será transformada em um fluxo binário serial, que será então modulado e enviado via OFDM.

Figura 20 – Imagem original a ser transmitida

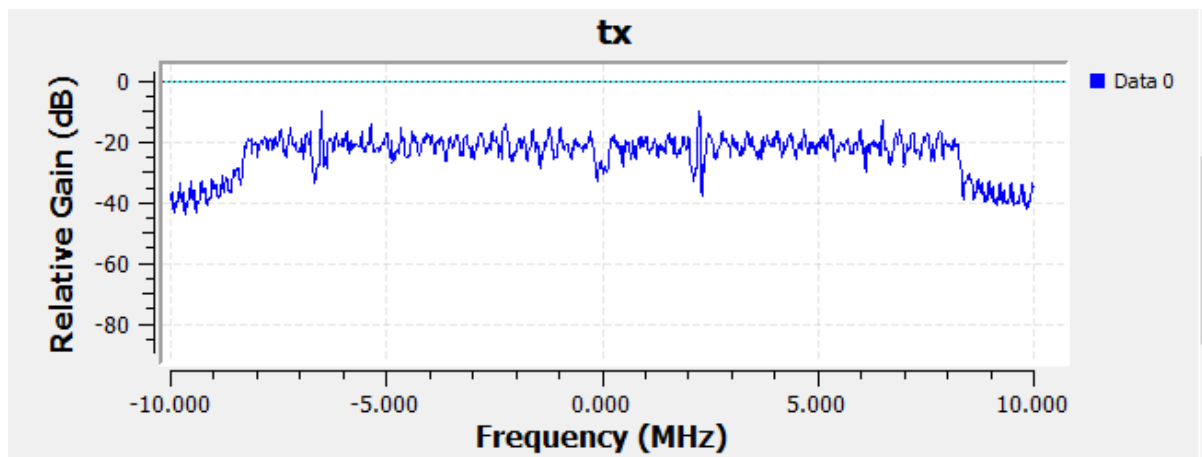


Fonte: Imagem padrão do banco de testes em processamento de imagem.

A partir disso o sinal é submetido as etapas de modulação discutidas anteriormente, mostrando um padrão no espectro de uma transmissão OFDM. Isso é demonstrado na

Figura 21. Observa-se a característica retangular típica do espectro do sinal modulado, ocupando uma largura de banda de 10 MHz, isso está em conformidade com o teorema de Nyquist, que estabelece que a taxa de amostragem mínima necessária para representar um sinal sem ambiguidade deve ser igual ou superior ao dobro da sua frequência máxima. No caso em questão, como visto anteriormente adotou-se uma taxa de amostragem de 20 MHz, garantindo assim a conformidade com o teorema e a fidelidade na representação do sinal. Outra característica é decaimento da potência nas bordas da banda onde justamente estão as 12 subportadoras nulas, com uma queda entre 15 a 20 db da potência do sinal, confirmando a eficiência espectral do técnica.

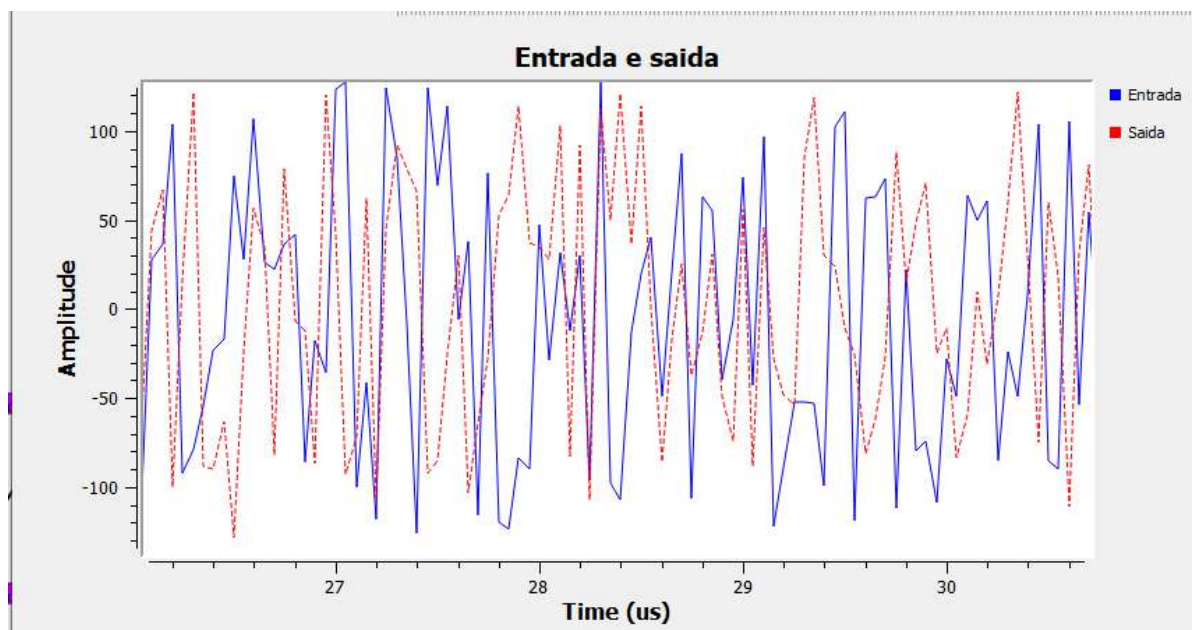
Figura 21 – Sinal Modulado OFDM



Fonte: (Próprio Autor, 2026).

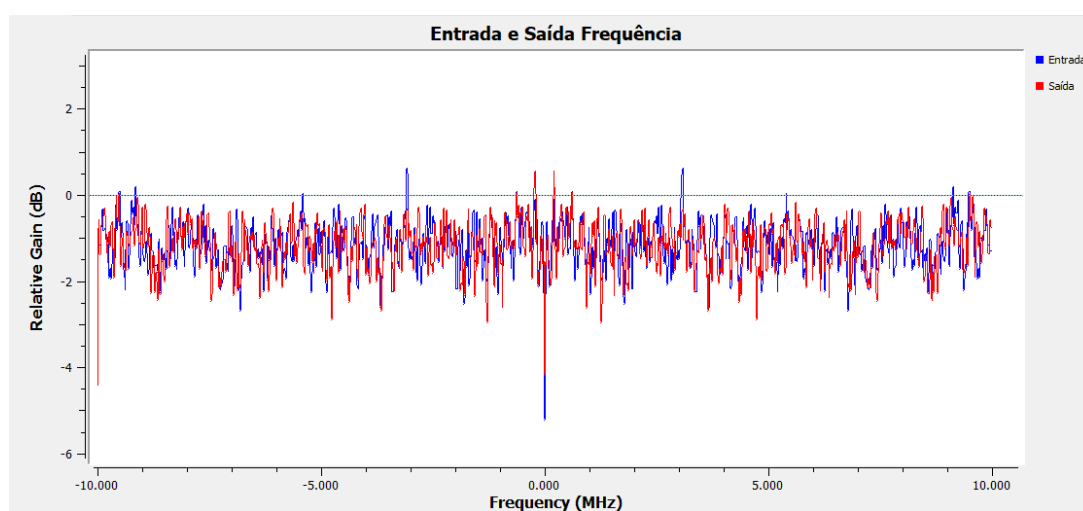
Nos mesmos blocos de tempo e frequência foram capturados também o sinais recebidos após a demodulação do sinal e etapas de equalização, comparando-se com a entrada original. As Figuras 22 e 23 mostram os sinais transmitidos e recebidos em tempo e frequência, sobrepostos a um ruído com amplitude de 0,3, o que representa 30% da potência em relação à atenuação unitária do sinal.

Figura 22 – Sinal original/recebido no tempo (ruído 30%)



Fonte: (Próprio Autor, 2026).

Figura 23 – Sinal original/recebido na frequência (ruído 30%)



Fonte: (Próprio Autor, 2026).

Observando ambas as figuras é evidente a degradação do sinal imposta pelo canal de comunicação simulado. No domínio do tempo, a diferença entre as formas de onda do sinal entrada e saída, apresenta tanto um atraso no tempo quanto alteração na amplitude do sinal recebido. Em conjunto, o sinal no espectro da frequência mantém um pouco mais de integridade com pequenas flutuações na potência do sinal e alguns atrasos. Apesar

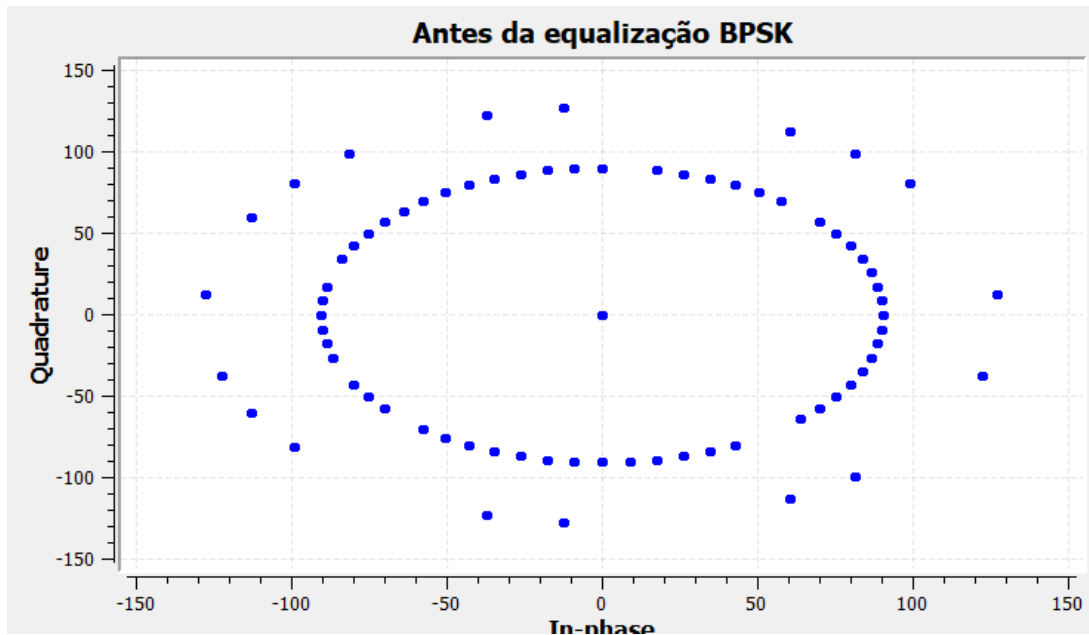
dessas diferenças, o conteúdo recebido manteve sua integridade com até 40% do sinal comprometido por ruído.

4.2 Análise das constelações

A verificação dos diagramas das constelações permitiu avaliar o comportamento dos símbolos complexos e a eficácia dos algoritmos de sincronização e equalização do canal. A análise foi segmentada em dois pontos, o sinal bruto recebido pós transformada de Fourier e o sinal processado após equalização e serialização, utilizando o bloco *QT GUI Constellation Sink*.

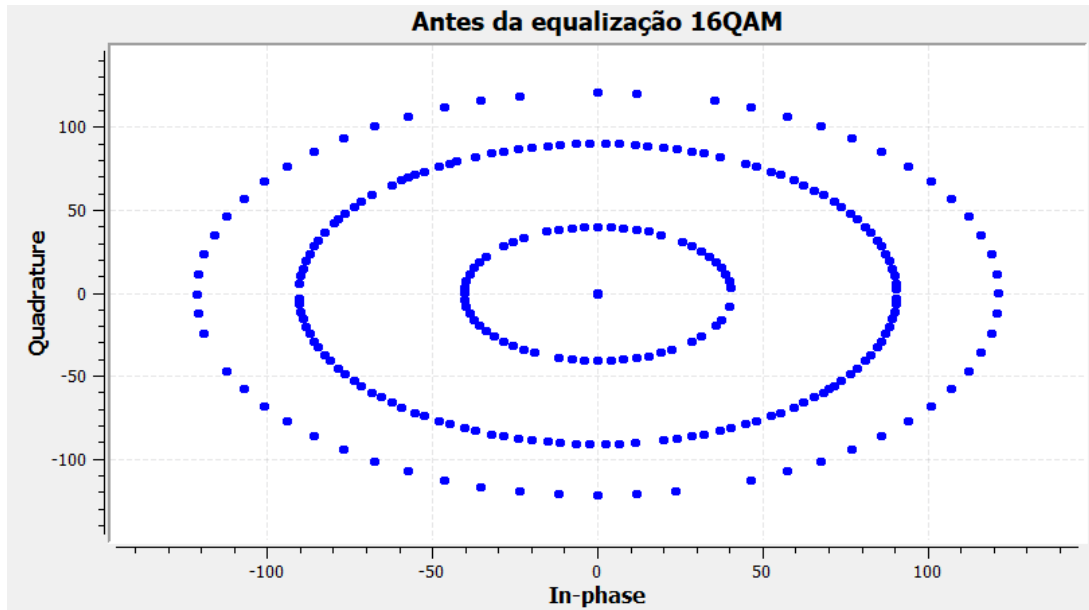
As Figuras 24 e 25 demonstram, respectivamente, as constelações do Header e do Payload, após a etapa demodulação OFDM, sem a correção de canal.

Figura 24 – Diagrama constelação Header antes da equalização



Fonte: (Próprio Autor, 2026).

Figura 25 – Diagrama constelação Payload antes da equalização

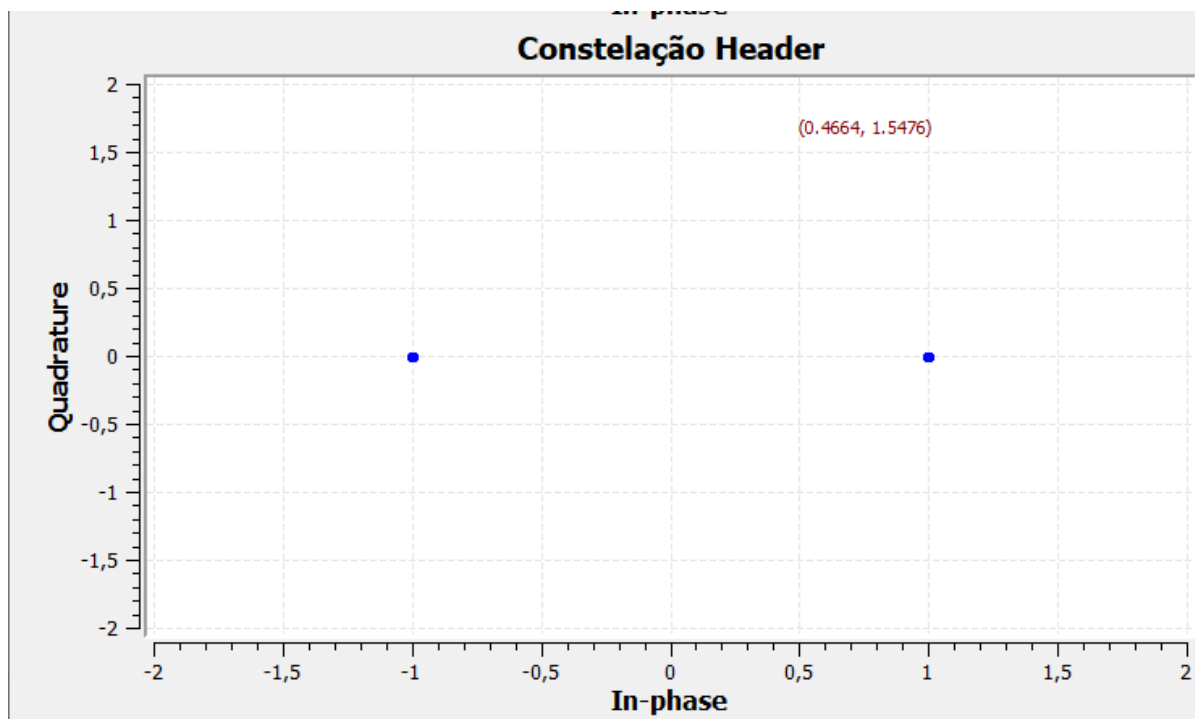


Fonte: (Próprio Autor, 2026).

Os símbolos não aparecem em posições fixas, mas sim dispersos em anéis, isso ocorre devido à rotação de fase introduzida pelo canal e pelo desalinhamento temporal. Como as subportadoras sofrem rotações de fase distintas, o resultado na constelação é o espalhamento circular dos pontos. Nota-se também uma concentração de pontos na origem (0,0), correspondente às subportadoras nulas.

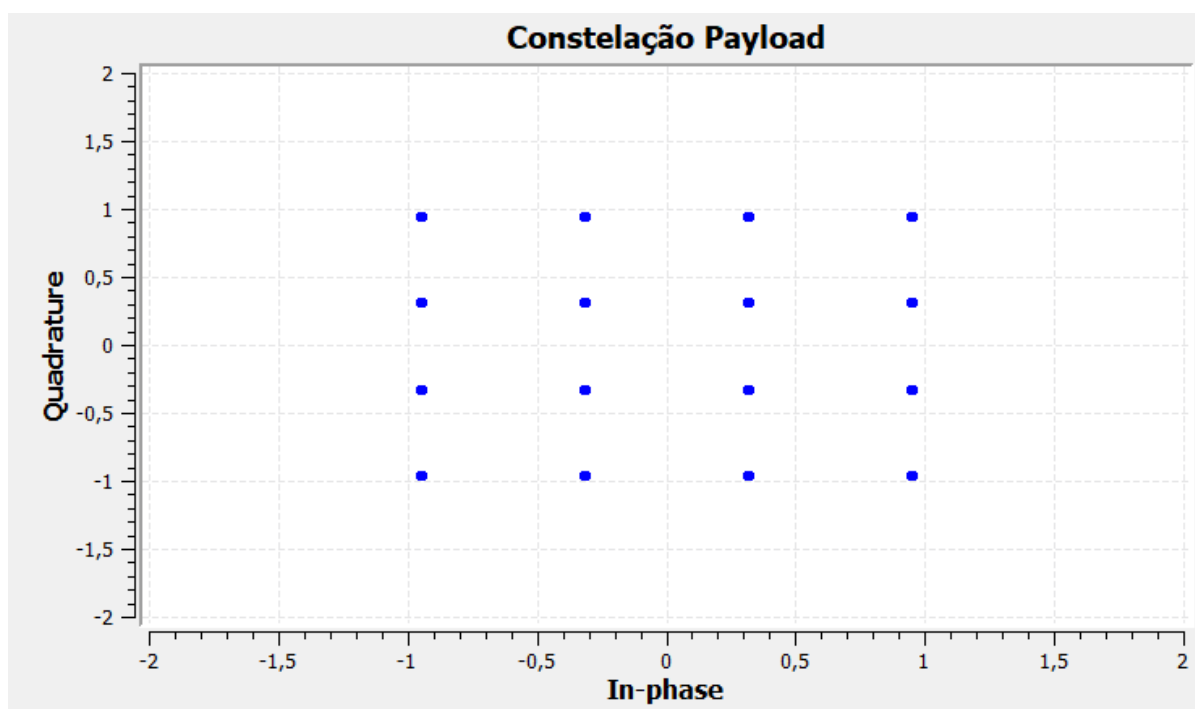
Após as etapas de equalização e serialização do sinal, os símbolos são realinhados as suas posições de modulação original, isso é observado nas Figuras 26 e 27:

Figura 26 – Diagrama constelação Header depois equalização sem ruído



Fonte: (Próprio Autor, 2026).

Figura 27 – Diagrama constelação Payload depois equalização sem ruído



Fonte: (Próprio Autor, 2026).

Na modulação do Header, observam-se dois agrupamentos distintos (fase 1 e -1), enquanto no gráfico da modulação do Payload inferior é possível distinguir claramente os 16 símbolos da modulação, validando as etapas anteriores.

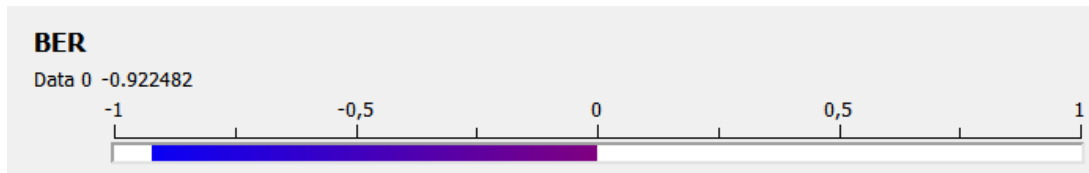
4.3 BER

Com o intuito de avaliar a eficiência do sistema contra condições de interferência por ruído, foi realizado uma verificação de desempenho baseado na métrica da Taxa de Erro de Bit (BER). O experimento consistiu em submeter o sinal transmitido a diferentes níveis de ruído, variando a potência do ruído no bloco *Channel Model*. A variação do parâmetro de ruído foi ajustada em uma faixa de 0,1 a 0,55 de potência.

A medição da métrica do erro foi feita utilizando o bloco *BER* disponível no software, nele a taxa de erro é calculada comparando o sinal original do arquivo com o sinal na saída. A saída do bloco gera um resultado em escala logarítmica de base 10, com um valor de magnitude $-x$ correspondendo a uma probabilidade de erro dada por 10^{-x} .

A visualização dessa métrica é feita pelo bloco *QT GUI Number sink*. Esse bloco apresenta uma barra vertical com valores escolhidos de -1 a 1. Porém, só ocorreu alteração barra quando o ruído atingiu uma potência de 0,49, demonstrado na Figura abaixo.

Figura 28 – Barra de BER



Fonte: (Próprio Autor, 2026).

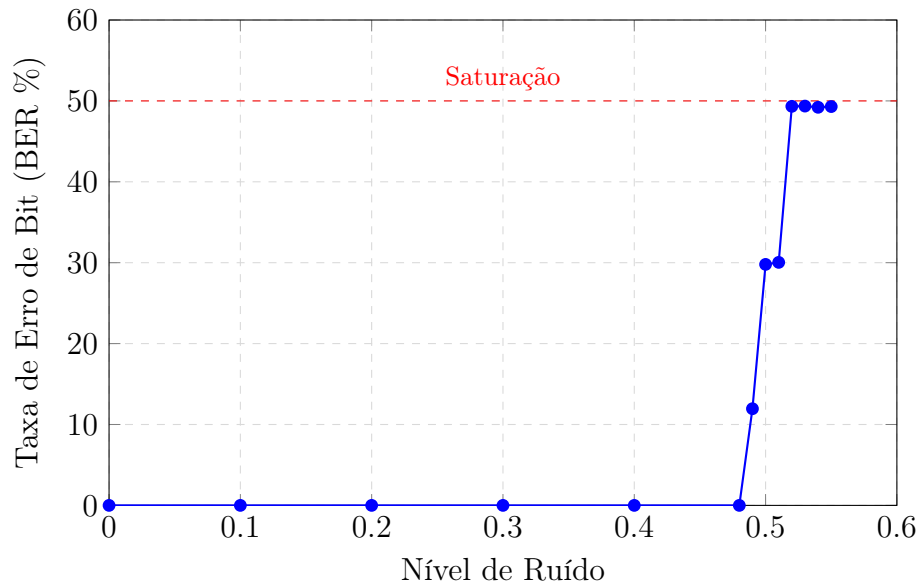
A partir dessa barra foi montado uma Tabela 2 e um gráfico com valores de BER obtidos a medida que o ruído aumentava:

Tabela 2 – Relação entre Ruído e Taxa de Erro de Bit (BER)

Nível de Ruído	BER (10^{-x})	Erro Percentual (%)
0,49	-0.9225	11,95%
0,50	-0.5259	29,79%
0,51	-0.5222	30,04%
0,52	-0.3068	49,33%
0,53	-0.3066	49,36%
0,54	-0.3079	49,21%
0,55	-0.3071	49,30%

Fonte: (Próprio Autor, 2026).

Figura 29 – Curva da BER em função do Ruído



Fonte: (Próprio Autor, 2026).

Os resultados apresentados tanto na tabela como no gráfico, evidenciam uma rápida degradação do sistema a medida que o nível do ruído é incrementado. Pode-se observar que em 0,49 de ruído, o sistema ainda consegue operar e receber a informação com cerca de 11% de erro. Porém, a medida que esse nível aumenta em cerca de 0,01 na amplitude, o erro aumenta em 20%. Com o aumento no nível da amplitude, o sistema atinge a saturação e chega próximo dos 50% de erro, indicando que metade da informação contida no receptor está sendo aleatória.

4.4 Validação da imagem

Para demonstrar visualmente os efeitos do ruído sobre o canal de comunicação, em cada cenário de degradação um arquivo de imagem foi salvo. Com o objetivo de verificar de forma direta e intuitiva a qualidade de transmissão da imagem em cada cenário de taxa de erro apresentados anteriormente. A Figura 30 mostra esse comparativo entre a imagem original e as recebidas para os níveis propostos de ruído:

Figura 30 – Validação entre imagens



Fonte: (Próprio Autor, 2026).

A inspeção visual das imagens confirma a avaliação feita na seção anterior, onde a faixa de ruído entre 0,1 a 0,4 de potência, o sistema não apresenta erros em bit na métrica, entregando na saída imagens visualmente semelhantes à original.

Da mesma forma, a partir do nível 0,49, a imagem apresenta perda de informação em algumas partes, mas ainda segue reconhecível. Porém a medida que o ruído aumenta para 0,50 a 0,51 mais erros são adicionados como cores distorcidas e perda de informação. Conforme previsto pela curva de BER, a saturação desse erro chega aos cenários de 0,52 a 0,55 resultando em uma degradação destrutiva de informação. Nestes casos, a informação visual é praticamente suprimida pelo ruído, evidenciando incapacidade do receptor de interpretar os símbolos corretamente.

5 CONCLUSÃO

Neste trabalho cumpriu-se o objetivo de implementar e expor o desempenho de um sistema de comunicação digital baseado na técnica de modulação por multiplexação ortogonal por divisão de frequência (OFDM), utilizando a plataforma de rádio definido por software GNU Radio Companion, com uma arquitetura baseada em parâmetros do padrão IEEE 802.11a adaptada a uma faixa de frequência de 10 MHz, demonstrando viabilidade em sua construção complexa através dos blocos de processamento digital em software.

Pela metodologia utilizada foi possível construir todo diagrama de transmissão e recepção, integrando etapas como modulação dos símbolos, estimação e equalização do canal, além de técnicas como sincronização de tempo e frequência, aleatorização de dados e verificações de integridade.

Os resultados obtidos através dos gráficos e da medição da métrica de erro de bit (BER) mostraram uma robustez do sistema em cenários com ruído moderado. Observou-se uma estabilidade no sistema até um limiar de 0,49 de potência de ruído, ou seja, 49% do sinal comprometido por ruído, com uma BER de 11%. Porém, como dito no capítulo anterior, ocorreu um aumento bruto da BER à medida que o nível de ruído passa de 0,52, onde o sistema atinge um ponto de saturação, atingindo uma taxa de 50% de erro, caracterizando uma perda de metade do conteúdo original.

Em conjunto a isso, a validação das imagens comprovaram os dados estatísticos obtidos na BER. A comparação entre as imagens recebidas e a original, confirmou que mesmo sob taxas de erro de 11% a 29%, a informação ainda era visivelmente semelhante a original, degradando-se quando o canal atinge o estado de aleatorização.

Portanto, o projeto desenvolvido serviu para demonstrar a viabilidade e eficácia para o estudo prático dos conceitos de um sistema de comunicação OFDM, demonstrando tanto as vantagens na mitigação de interferências e quanto nas limitações impostas sobre um canal real na ausência de códigos corretores de erro.

5.1 Trabalhos Futuros

Considerando os resultados obtidos e as limitações identificadas neste estudo, sugerem-se as seguintes propostas para a continuidade e aprimoramento da pesquisa:

- **Implementação de Código de Correção de Erro (FEC):** Adicionar codificação para correção de erros (Convolutacional, Reed-Solomon, entre outros) para melhorar

a robustez do sistema e diminuir a BER em cenários de baixo SNR, evitando a degradação abrupta observada nos resultados.

- **Testes em Hardware Real:** Portar o fluxograma desenvolvido para equipamentos de SDR reais, com intuito de avaliar o comportamento e eficiência do sistema sob efeitos físicos reais.

Referências

- ARMSTRONG, J. Ofdm for optical communications. **Journal of Lightwave Technology**, v. 27, n. 3, p. 189–204, 2009.
- ARNDT, D. M. **Análise Comparativa entre os Sistemas OFDM e FBMC na Transmissão de TV Digital**. Dissertação (Dissertação (Mestrado em Engenharia Elétrica)) — Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2012.
- BAHAI, A. R. S.; SALTZBERG, B. R.; ERGEN, M. **Multi-Carrier Digital Communications: Theory and Applications of OFDM**. 2. ed. New York: Springer, 2004.
- GNU Radio. **GNU Radio - The Free & Open Software Radio Ecosystem**. 2025. Acesso em: 15 jan. 2026. Disponível em: <<https://www.gnuradio.org/>>.
- HWANG, T.; YANG, C.; WU, G.; LI, S.; LI, G. Y. Ofdm and its wireless applications: A survey. **IEEE Transactions on Vehicular Technology**, v. 58, n. 4, p. 1673–1694, May 2009.
- IEEE. **IEEE Standard 802.11b-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band**. [S.l.], 1999. Supplement to IEEE Std 802.11-1999.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Std 802.11a-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band**. New York, 1999. Amendment 1 to IEEE Std 802.11.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Std 802.3-2018: IEEE Standard for Ethernet**. New York, 2018. Section 3: Media Access Control (MAC) Frame Structure.
- JIANG, W.; HAN, B.; HABIBI, M. A.; SCHOTTEN, H. D. The road towards 6g: A comprehensive survey. **IEEE Open Journal of the Communications Society**, v. 2, p. 334–366, 2021.
- LATHI, B. P.; DING, Z. **Modern Digital and Analog Communication Systems**. 4. ed. New York: Oxford University Press, 2012.
- MAHMOOD, S. H.; NEWAGY, F. **Effect of switching loads on the power line communication(PLC) modems inside direct local area network (dlan)**. 2015. Disponível em: ResearchGate. Disponível em: <https://www.researchgate.net/publication/284450450_Effect_of_switching_loads_on_the_power_line_communicationPLC_modems_inside_direct_local_area_network_dlan>. Acesso em: 20 jan. 2026.
- MITOLA, J. The software radio architecture. **IEEE Communications Magazine**, v. 33, n. 5, p. 26–38, May 1995.

- MOURA, J. M. F.; STEWART, R. W. Gnu radio: An open-source toolkit for software-defined radio. **IEEE Proceedings**, v. 111, n. 3, p. 355–368, 2023.
- PROAKIS, J. G.; SALEHI, M. **Digital Communications**. 5. ed. New York: McGraw-Hill, 2008.
- SCHMIDL, T. M.; COX, D. C. Robust frequency and timing synchronization for ofdm. **IEEE Transactions on Communications**, v. 45, n. 12, p. 1613–1621, Dec. 1997.
- SELVA, A. F. B.; REIS, A. L. G.; LENZI, K. G.; MELONI, L. G. P.; BARBIN, S. E. Introduction to the software-defined radio approach. **IEEE Latin America Transactions**, v. 10, n. 1, p. 1156–1161, Jan. 2012.
- SOARES, A. Rádio definido por software: Pesquisas e verificação de testes em uma plataforma livre. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, v. 8, p. 228–244, Set. 2016. Disponível em: <<https://www.nucleodoconhecimento.com.br/tecnologia/radio-definido>>. Acesso em: 20 jan. 2026.
- STACKOVERFLOW. **G3RUH Multiplicative Scrambler**. 2022. Disponível em: <<https://stackoverflow.com/questions/71847704/g3ruh-multiplicative-scrambler>>. Acesso em: 6 de fevereiro de 2026.
- Tech Differences. **Difference Between FDM and OFDM**. 2018. Acesso em: 10 jan. 2026. Disponível em: <<https://techdifferences.com/difference-between-fdm-and-ofdm.html>>.
- ULVERSOY, T. Software defined radio: Challenges and opportunities. **IEEE Communications Surveys & Tutorials**, v. 12, n. 4, p. 531–550, 2020.
- WEINSTEIN, S. B. The history of orthogonal frequency-division multiplexing. **IEEE Communications Magazine**, v. 47, n. 11, p. 26–35, November 2009.
- YATEBTS. **OFDM: The Science Behind LTE**. 2026. Disponível em: <<https://yatebts.com/ofdm-the-science-behind-lte/>>. Acesso em: 6 de fevereiro de 2026.

Anexos

Fonte: (Elaborado pelo autor, 2026).

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus João Pessoa - Código INEP: 25096850
	Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, João Pessoa (PB)
	CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Restrito

Trabalho de conclusão de curso - versão final

Assunto:	Trabalho de conclusão de curso - versão final
Assinado por:	Vinicius Wanderley
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:
■ Vinicius Jose Wanderley Vieira , DISCENTE (20202610030) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA, em 19/02/2026 14:49:27.

Este documento foi armazenado no SUAP em 19/02/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1770562
Código de Autenticação: 5a500d44e2

