

UNIVERSIDADE FEDERAL DO PARANÁ

ROGER METZ MEIRA

ALGORITMOS PARA AUTOMAÇÃO DE MEDIÇÕES DE SINAIS DE RF
PARA IOT COM ANALISADOR DE ESPECTRO

CURITIBA

2025

UNIVERSIDADE FEDERAL DO PARANÁ

ROGER METZ MEIRA

Trabalho de Conclusão de Curso

**ALGORITMOS PARA AUTOMAÇÃO DE MEDIÇÕES DE SINAIS DE RF PARA IOT
COM ANALISADOR DE ESPECTRO**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica, De-
partamento de Engenharia Elétrica, Setor de Tec-
nologia, Universidade Federal do Paraná, como re-
quisito parcial à obtenção do título de Bacharel em
Engenharia Elétrica

Orientador(a): Prof. Dr. André Augusto Mariano

Coorientador(a): Dr. Arthur Liraneto

CURITIBA

2025

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão às pessoas e instituições que foram fundamentais para a realização deste Trabalho de Conclusão de Curso e para a minha jornada profissional.

À minha namorada, pelo apoio incondicional e pela compreensão durante todo o processo do TCC. Sua motivação constante e a capacidade de me guiar de volta ao caminho certo foram pilares essenciais para que eu pudesse superar os desafios e concluir este trabalho.

Ao meu orientador, Professor Doutor André Augusto Mariano, pela inestimável orientação e por ter sido a inspiração que me incentivou a seguir a área de eletrônica e, eventualmente, o campo da engenharia de testes. Sua mentoria foi de grande valia para que eu me descobrisse como profissional e definisse minhas ambições de carreira.

Ao meu coorientador, Doutor Arthur Liraneto, por todo o suporte técnico e prático oferecido durante o desenvolvimento deste TCC. Seus ensinamentos foram valiosos para corrigir algumas convicções prévias e para me mostrar a realidade e a dinâmica da engenharia de testes na prática, com um olhar pragmático e inspirador.

Ao laboratório LiraLab, por ter me proporcionado toda a infraestrutura necessária para a validação dos resultados deste projeto. Além disso, agradeço imensamente pela oportunidade de estágio, que enriqueceu meu portfólio e consolidou meu aprendizado em um ambiente profissional de excelência.

A todos que, direta ou indiretamente, contribuíram para esta conquista, o meu muito obrigado.

RESUMO

Este trabalho visa desenvolver algoritmos para automação de medições de sinais de radiofrequência (RF) para aplicações em Internet das Coisas (IoT), utilizando um analisador de espectro e sua Interface de Programação de Aplicações (API) em Python. Utilizar tal API traz o benefício de não precisar ter dependência de softwares que possuem licença de uso, possibilitando resultados similares ou com diferenças menores que 5%. A automação reduz a intervenção manual, melhorando a reprodutibilidade e a precisão das medições, essenciais para avaliar parâmetros como potência de sinal e largura de banda ocupada. A estrutura modular do código, orientada a objetos, permite uma operação flexível e robusta, sendo composta por blocos que realizam desde a configuração do analisador até o processamento dos dados com algoritmos de média móvel, potência integrada e banda ocupada.

PALAVRAS-CHAVE

automação de medições; analisador de espectro; Internet das Coisas; radio-frequência; Python.

ABSTRACT

This work aims to develop algorithms for the automation of radio frequency (RF) signal measurements for Internet of Things (IoT) applications, utilizing a spectrum analyzer and its Application Programming Interface (API) in Python. Using such an API offers the benefit of not requiring dependency on licensed software, enabling similar results with differences less than 5%. Automation reduces manual intervention, improving the reproducibility and accuracy of measurements, which are essential for evaluating parameters such as signal power and occupied bandwidth. The modular, object-oriented structure of the code allows for flexible and robust operation, comprising blocks that perform everything from analyzer configuration to data processing with algorithms for moving average, integrated power, and occupied bandwidth.

KEYWORDS

measurement automation; spectrum analyzer; Internet of Things; radio frequency; Python.

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivo geral	10
1.2	Objetivos específicos	10
2	REVISÃO BIBLIOGRÁFICA	12
2.1	ENGENHARIA DE TESTES E AUTOMAÇÃO	12
2.2	Internet das Coisas (IoT) e a Necessidade de Medições de RF	12
2.3	Regulamentação e Conformidade (ANATEL)	13
2.3.1	ANATEL Ato 14448 e Ato 237	14
2.4	Garantia da Qualidade Metrológica (INMETRO)	14
2.5	INSTRUMENTAÇÃO	14
2.5.1	Metrologia	15
2.6	Fundamentos de Processamento Digital de Sinais (DSP)	15
2.7	COMPLEXIDADE COMPUTACIONAL	16
2.7.1	TRANSFORMADAS	17
2.7.2	Transformada discreta de fourier	17
2.7.3	Transformada rápida de fourier	18
2.7.4	Transformada Chirp Z	18
2.8	AMBIENTE DE DESENVOLVIMENTO E OTIMIZAÇÃO	19
2.8.1	Paralelismo Concorrente: <i>threads</i> em Python para Medições de RF	19
2.8.2	Python	20
2.8.3	Programação orientada ao objeto	21
3	METODOLOGIA DE DESENVOLVIMENTO	22
3.1	Blocos funcionais	22
3.2	Funções matemáticas	24
3.3	PDF GENERATOR	27

4	RECURSOS NECESSÁRIOS	29
4.1	RSA306B USB Spectrum Analyzer	30
4.2	Signalhound VSG25A	31
4.3	ESP32	33
5	RESULTADOS	35
5.1	Validação dos resultados	39
6	CONCLUSÃO	46

1 INTRODUÇÃO

A engenharia de testes tem desempenhado um papel fundamental na evolução da indústria eletrônica, desde os primeiros procedimentos realizados manualmente com instrumentos simples até os sistemas modernos, altamente automatizados. Inicialmente, os testes eram conduzidos de forma artesanal, com foco na verificação funcional básica de componentes individuais[1]. Com o tempo, a crescente complexidade dos dispositivos eletrônicos e a demanda por maior confiabilidade impulsionaram o desenvolvimento de métodos de teste mais precisos, rápidos e padronizados, integrando cada vez mais recursos computacionais aos processos de medição[1].

No entanto, o rápido avanço da Internet das Coisas (IoT) tem aumentado significativamente o volume de dispositivos conectados por redes sem fio, os quais operam em diferentes frequências de radiofrequência (RF) para diferentes aplicações, assim como sistemas de comunicação, sistemas de segurança, sistemas de radar, sistemas de monitoramento meteorológicos e médicos[2]. Com essa proliferação, há uma demanda crescente do mercado por métodos eficientes e precisos que permitam a reprodutibilidade de ensaios em RF, fundamentais para garantir a qualidade e o desempenho dessas comunicações. As medições manuais, além de demandarem tempo, são suscetíveis a erros humanos. Estes ocorrem quando a aquisição de parâmetros é feita através da observação visual do comportamento do sinal, e assim, fica a critério do operador decidir qual critério de estabilização do sinal ele irá escolher. Isso acaba introduzindo variáveis que podem dificultar a reprodutibilidade dos ensaios, o que reforça a necessidade de automação nesse processo[3].

Nesse cenário, a padronização das medições torna-se indispensável para garantir a conformidade técnica e a interoperabilidade entre dispositivos de radiofrequência. No Brasil, a Agência Nacional de Telecomunicações (Anatel) é o principal órgão responsável por regulamentar o uso do espectro eletromagnético, estabelecendo limites de emissões, requisitos técnicos e procedimentos de certificação de equipamentos. Já o Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro) atua de forma

complementar, sendo responsável por assegurar a qualidade e a rastreabilidade das medições, por meio da calibração de instrumentos e da acreditação de laboratórios de ensaio. Essa separação de funções garante que os dispositivos sejam avaliados tanto em relação à sua conformidade regulatória quanto à exatidão dos dados obtidos durante os testes.

No âmbito internacional, normas emitidas por instituições como a Comissão Eletrotécnica Internacional (IEC), o Instituto Europeu de Normas de Telecomunicações (ETSI) e a *Federal Communications Commission* (FCC), dos Estados Unidos, fornecem diretrizes técnicas amplamente reconhecidas. No Brasil, embora existam regulamentações específicas elaboradas pela Anatel, muitas dessas diretrizes são baseadas ou inspiradas nos padrões estabelecidos pela FCC e pelo ETSI, garantindo alinhamento técnico com as principais exigências internacionais. Essas normas desempenham um papel fundamental na harmonização dos procedimentos de ensaio, assegurando a compatibilidade eletromagnética, a segurança e a interoperabilidade entre dispositivos que operam em diferentes regiões e mercados.

Visando o mercado de testes para homologação de equipamentos de telecomunicações, torna-se essencial o desenvolvimento de soluções automatizadas, pautadas pelas normas vigentes no país, que não apenas reduzam a intervenção humana, mas também promovam padronização, escalabilidade e confiabilidade nos testes[3]. Com esse desafio em mente, este trabalho propõe o desenvolvimento de algoritmos para automação de medições de sinais de RF, utilizando um analisador de espectro e sua API (*Application Programming Interface*), por meio da linguagem de programação Python. A proposta visa otimizar a coleta e o processamento de dados espectrais, assegurando maior precisão e reprodutibilidade nas medições, além de permitir a análise eficiente de grandes volumes de dados com intervenção manual mínima.

Ao automatizar tarefas repetitivas em ensaios de RF, este trabalho também viabiliza a adoção de técnicas mais avançadas de processamento de sinais, contribuindo diretamente para a eficiência e robustez de sistemas voltados à Internet das Coisas. Além disso, a solução proposta busca ser acessível e replicável, oferecendo suporte a engenheiros e técnicos envolvidos com metrologia e testes em sistemas de comunicação sem fio, atendendo tanto demandas acadêmicas quanto industriais.

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver uma solução de automação robusta para medições de sinais de radiofrequência (RF), focando no processamento de dados oriundos de um analisador de espectro controlado por software. Para isso, será implementado um software em linguagem Python que se comunicará com o equipamento para receber e adquirir dados, aplicar algoritmos de processamento de sinais e fornecer informações relevantes para a realização de ensaios em diversos tipos de sinais.

Além disso, o trabalho visa validar a solução em cenários práticos de Internet das Coisas (IoT), comparando a eficiência, precisão e reprodutibilidade das medições automatizadas em relação às realizadas manualmente. Espera-se, com isso, aumentar significativamente a produtividade no ambiente de laboratório e oferecer ferramentas que beneficiem tanto a pesquisa acadêmica quanto as aplicações industriais.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral proposto, este trabalho se propõe a cumprir os seguintes objetivos específicos:

- Estabelecer uma conexão eficiente e confiável com o analisador de espectro por meio de comandos via software, garantindo a comunicação para a aquisição de dados em uma cadência otimizada para o processamento subsequente.
- Implementar rotinas de aquisição de dados para testes e medições nos domínios do tempo e da frequência, considerando as características de sinais de radiofrequência.
- Desenvolver algoritmos para o tratamento e análise de sinais de radiofrequência (RF), incluindo, cálculo de potência e determinação de largura de banda ocupada.
- Validar os algoritmos implementados em cenários representativos de aplicações em Internet das Coisas (IoT), emulando condições operacionais típicas.

- Comparar a precisão dos resultados obtidos por meio das medições automatizadas com aqueles obtidos manualmente e com aquelas obtidas através do SignalVU, software proprietário do fabricante do analisador utilizado, quantificando as diferenças e avaliando a confiabilidade da automação.
- Avaliar e comparar o tempo de execução das medições automatizadas em relação às medições realizadas de forma manual, demonstrando a eficiência do processo automatizado.

2 REVISÃO BIBLIOGRÁFICA

2.1 ENGENHARIA DE TESTES E AUTOMAÇÃO

A engenharia de testes desempenha um papel essencial no desenvolvimento de sistemas de telecomunicações e Internet das Coisas (IoT), assegurando a conformidade técnica, a confiabilidade e a eficiência operacional de dispositivos e redes[3]. Essa área abrange desde a validação de hardware até a avaliação de software, com o uso de metodologias padronizadas e ferramentas avançadas, como analisadores de espectro e geradores de sinal. No contexto de um laboratório ou uma indústria, a automação de medições de sinais de RF é um exemplo direto de aplicação da engenharia de testes, em que a confiabilidade é um fator crucial para o controle de qualidade dos resultados obtidos, garantindo maior reprodução, precisão e redução de erros humanos. Técnicas como a análise de espectro e transformadas de tempo discreto, incluindo FFT (*Fast Fourier Transform*) e CZT (*Chirp-Z Transform*)[4], são cruciais para identificar interferências, medir a potência integrada e otimizar o desempenho de sistemas IoT, seguindo padrões técnicos estabelecidos por regulamentações como os Atos nº 14448 e nº 237 da Anatel. Essa abordagem automatizada não apenas garante aderência às normas, mas também contribui para a inovação e eficiência no desenvolvimento de novas tecnologias de comunicação.

2.2 INTERNET DAS COISAS (IOT) E A NECESSIDADE DE MEDIÇÕES DE RF

A Internet das Coisas (IoT) representa um paradigma tecnológico onde bilhões de dispositivos físicos, equipados com sensores, software e outras tecnologias, conectam-se e trocam dados com outros dispositivos e sistemas pela internet [5]. Essa expansão sem precedentes abrange uma vasta gama de aplicações, desde sistemas de monitoramento ambiental e cidades inteligentes até dispositivos vestíveis e automação industrial, operando frequentemente em diversas bandas de radiofrequência (RF) para comunicação. A vasta utilização desses dispositivos e a dependência de sua conectividade sem fio tornam a qualidade e a confiabilidade das comunicações de RF um fator crítico para o sucesso e a segurança das soluções de IoT [2].

Nesse cenário, a capacidade de realizar medições precisas e reproduzíveis de sinais de RF é mais do que uma necessidade técnica; é um imperativo para garantir a conformidade com padrões, otimizar o desempenho e diagnosticar falhas. Historicamente, essas medições eram executadas de forma manual, um processo que, além de ser demorado e caro, introduzia variáveis significativas devido à intervenção humana, comprometendo a reprodutibilidade dos ensaios [3]. A ineficiência e a suscetibilidade a erros inerentes às medições manuais tornam-se insustentáveis diante do volume massivo de dispositivos IoT e da velocidade com que novas tecnologias são introduzidas.

A evolução da engenharia de testes, impulsionada pela complexidade crescente de dispositivos como os *System-on-a-Chip* (SoC) em comunicações sem fio, tem apontado a automação como a solução fundamental para superar esses desafios [1][3]. A automação das medições de RF para IoT não apenas acelera o processo de validação, mas também eleva a precisão, a consistência e a rastreabilidade dos resultados, permitindo uma análise mais aprofundada do comportamento dos dispositivos no espectro de radiofrequência. Isso é vital para identificar interferências, verificar a conformidade com normas regulatórias e garantir a coexistência harmoniosa de múltiplos dispositivos em ambientes de RF cada vez mais congestionados.

2.3 REGULAMENTAÇÃO E CONFORMIDADE (ANATEL)

A Agência Nacional de Telecomunicações (Anatel) é a entidade responsável pela regulamentação e fiscalização do setor de telecomunicações no Brasil, sendo fundamental para assegurar a eficiência do espectro, promover a competitividade e proteger os consumidores. Criada pela Lei Geral de Telecomunicações (Lei nº 9.472/1997), a agência também adota diretrizes internacionais da União Internacional de Telecomunicações (UIT) para padronizar práticas e garantir a interoperabilidade global[6]. Recentemente, a Anatel tem avançado em modelos de regulação responsiva, que buscam engajar os atores do mercado em práticas colaborativas e menos punitivas, incentivando a conformidade técnica e a inovação. Essa abordagem é especialmente relevante em setores como IoT, onde a evolução tecnológica exige normas claras e atualizadas para estimular investimentos e garantir a segurança e qualidade de dispositivos e redes.

2.3.1 ANATEL Ato 14448 e Ato 237

Um exemplo do papel da Anatel na normatização técnica são os Atos nº14448 e nº237. O Ato nº14448[7] estabelece limites técnicos de emissões eletromagnéticas para equipamentos, garantindo que operem dentro de níveis seguros e compatíveis com os padrões internacionais. Já o Ato nº 237[8] define os critérios e procedimentos para avaliação da conformidade, assegurando que dispositivos de telecomunicações sejam submetidos a testes rigorosos antes de sua homologação. Esses atos normativos são fundamentais para o desenvolvimento de sistemas de medição de sinais de RF, como o proposto neste trabalho, fornecendo diretrizes que norteiam o projeto e garantem a aderência às regulamentações técnicas do setor.

2.4 GARANTIA DA QUALIDADE METROLÓGICA (INMETRO)

O Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO)[9] desempenha um papel fundamental no cenário brasileiro, fornecendo o suporte metrológico necessário para garantir a confiabilidade das medições e a qualidade dos produtos e serviços. Como órgão executivo do governo federal, o INMETRO estabelece as bases para a metrologia científica e industrial, coordena a Rede Brasileira de Calibração (RBC)[10] e acredita organismos de avaliação da conformidade, contribuindo para o desenvolvimento tecnológico e a competitividade da indústria nacional. No contexto dos ensaios de equipamentos de telecomunicações, o INMETRO complementa a atuação da Anatel ao assegurar que as medições realizadas para verificar a conformidade com os requisitos técnicos sejam rastreáveis a padrões nacionais ou internacionais e realizadas de acordo com as boas práticas de laboratório.

2.5 INSTRUMENTAÇÃO

A instrumentação voltada para a análise e medição de sinais de RF desempenha um papel central no desenvolvimento e validação de sistemas de comunicação. Ferramentas como analisadores de espectro e analisadores vetoriais de sinais permitem identificar parâmetros críticos como potência e largura de banda, fundamentais para a avaliação de dispositivos IoT. De acordo com o manual da National Instruments[11]

sobre analisadores de sinais de RF, a precisão dessas medições é garantida por sistemas de calibração, filtros digitais de alta resolução e algoritmos de aquisição otimizados para análise em tempo real. Complementarmente, o documento "Agilent Spectrum Analysis Basics" destaca que o uso de técnicas como filtros de banda intermediária (IF) e transformadas de tempo discreto, como FFT, possibilita alta seletividade espectral e identificação precisa de sinais, mesmo em ambientes ruidosos. Esses aspectos são críticos para garantir que os dispositivos atendam aos padrões regulamentares e funcionem de forma confiável em redes de comunicação modernas.

2.5.1 Metrologia

A metrologia, ciência da medição, é essencial para garantir a exatidão e a confiabilidade dos resultados dos ensaios. O INMETRO, como guardião da metrologia no Brasil, estabelece os padrões de medição e as diretrizes para a calibração de instrumentos, assegurando que as medições realizadas em diferentes laboratórios sejam consistentes e comparáveis. A rastreabilidade metrológica, ou seja, a capacidade de relacionar as medições a padrões de referência reconhecidos, é um requisito fundamental para a aceitação dos resultados dos ensaios e para a tomada de decisões confiáveis. No contexto dos ensaios de equipamentos de telecomunicações, a metrologia desempenha um papel crucial na garantia da qualidade dos produtos e serviços, na proteção dos consumidores e na promoção da inovação.

2.6 FUNDAMENTOS DE PROCESSAMENTO DIGITAL DE SINAIS (DSP)

O processamento digital de sinais (PDS) é uma área essencial para a manipulação e análise de sinais no domínio digital, permitindo transformar dados brutos em informações úteis por meio de técnicas matemáticas. Conforme descrito por Oppenheim e Schafer em *Discrete-Time Signal Processing*, o DSP utiliza operações como amostragem, filtragem, transformadas de Fourier e convolução para analisar características de sinais, como amplitude, frequência e fase. Um exemplo de uma configuração para a realização de processamento de sinais é mostrada na Figura 1.

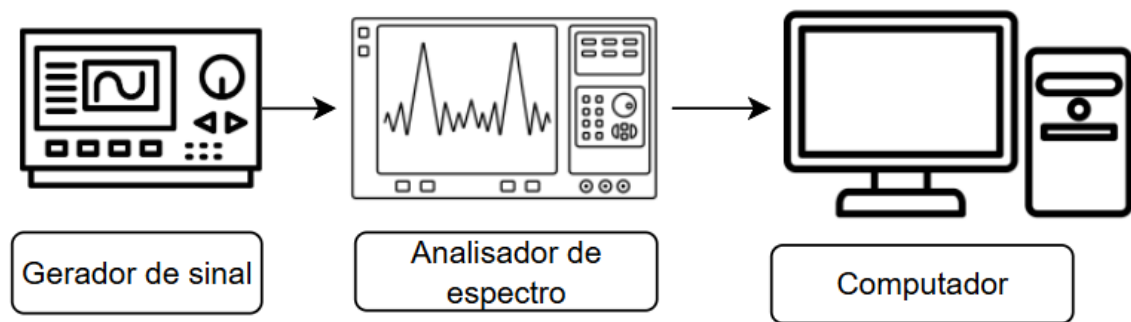


Figura 1: Diagrama genérico de uma aplicação de DSP

Uma aplicação central do PDS está na análise espectral, onde transformadas como a DFT e a FFT, geralmente realizadas no hardware do analisador de espectro, permitem identificar componentes espectrais em sinais complexos amostrados em uma determinada janela de tempo, viabilizando medições precisas e eficientes. No contexto de sistemas de comunicação e IoT, o PDS é especialmente relevante para melhorar a eficiência de algoritmos de medição automatizada, reduzindo o impacto de ruídos e otimizando o uso de recursos computacionais. Essas técnicas são fundamentais para assegurar que dispositivos atendam a padrões de desempenho e qualidade estabelecidos em ambientes de alta densidade espectral.

2.7 COMPLEXIDADE COMPUTACIONAL

De acordo com o livro *Computational Complexity: A Modern Approach* [12], para avaliar a eficiência computacional de um algoritmo, é comum analisar o número de operações básicas realizadas em função do tamanho da entrada. Essa entrada é geralmente denotada por N , representando o tamanho do problema, como o número de elementos em uma lista, o número de pontos em um sinal digital ou a dimensão de uma matriz. No entanto, essa análise pode exigir um nível elevado de detalhamento técnico. Para simplificar essa avaliação, utiliza-se a notação Big-O, que descreve o comportamento assintótico do algoritmo, focando apenas no termo de maior ordem de complexidade e desconsiderando constantes e termos de menor impacto. Por exemplo, considerando a função $2N^3 + 6N^2 + N + 6$, o termo dominante é n^3 , o que implica

que a complexidade assintótica do algoritmo é $O(N^3)$.

2.7.1 TRANSFORMADAS

A Transformada Discreta de Fourier (DFT) é uma ferramenta matemática fundamental para a análise espectral de sinais, permitindo converter um sinal no domínio do tempo em sua representação no domínio da frequência.

A DFT, entretanto, apresenta uma complexidade computacional de $O(N^2)$, o que a torna pouco eficiente para aplicações com grandes conjuntos de dados. Para superar essa limitação, a Transformada Rápida de Fourier (FFT) foi desenvolvida, reduzindo a complexidade para $N \log_2 N$ por meio de algoritmos otimizados, como o Cooley-Tukey, que exploram simetrias intrínsecas na DFT.

A FFT é amplamente utilizada em sistemas de comunicação, instrumentação e processamento de sinais em tempo real devido à sua eficiência computacional.

Complementarmente, a Transformada *Chirp Z* (CZT) generaliza a DFT ao permitir a análise em pontos arbitrários no plano z , fornecendo maior flexibilidade na escolha do intervalo e da resolução espectral. Embora a CZT compartilhe a complexidade $N \log N$, sua implementação inclui operações adicionais, como convoluções e escalonamento, o que pode aumentar os requisitos de recursos. Cada uma dessas transformadas possui aplicações específicas, sendo a FFT a mais eficiente para transformadas completas, enquanto a CZT é ideal para análises focadas em faixas específicas de frequência, como em sistemas de IoT e comunicação sem fio.

2.7.2 Transformada discreta de fourier

Desenvolvendo a complexidade da DFT a partir de pontos discretos no tempo, temos que:

$$X(k) = \sum_{n=0}^N \exp\left(\frac{-j2\pi kn}{N}\right) \quad (1)$$

Como pode ser abstraído da equação, para cada ponto discreto $X(k)$, é necessário realizar n operações, dessa forma, a complexidade é uma função de ordem $n \cdot k$, como a equivalência no tempo e na frequência é garantida pelo par de transformada $f(t) \Leftrightarrow F(\omega)$, $n=k$, trazendo uma ordem de complexidade computacional $O(n \cdot k) = O(N^2)$.

2.7.3 Transformada rápida de fourier

Para a transformada rápida de fourier, é explorado as simetrias implícitas das funções seno e cosseno e periodicidades nas rotações do plano z , dessa forma, a complexidade computacional é reduzida para $O(N\log N)$.

2.7.4 Transformada Chirp Z

A Transformada Chirp Z (CZT) é uma generalização da Transformada Discreta de Fourier (DFT), que permite realizar a análise de sinais em pontos arbitrários no plano z , ao contrário da DFT, que está restrita ao círculo unitário. O algoritmo de Bluestein é frequentemente utilizado para calcular a CZT de forma eficiente, convertendo a convolução presente na fórmula da CZT em uma forma computacionalmente viável utilizando a DFT. A identidade $nk = \frac{-(k-n)^2}{2} + \frac{n^2}{2} + \frac{k^2}{2}$ permite separar a contribuição dos dados x_n e adicionar uma fase extra, essencial para o cálculo da CZT. Para a transformada Chirp Z, utilizando o algoritmo de Bluestein, partimos do principio da DFT.

$$X(k) = \sum_{n=0}^N x_n \exp\left(\frac{-j2\pi kn}{N}\right) \quad (2)$$

Alterando o termo nk pela identidade $nk = \frac{-(k-n)^2}{2} + \frac{n^2}{2} + \frac{k^2}{2}$ é obtido:

$$X_k = \exp\left(-\frac{j\pi}{N}k^2\right) \sum_{n=0}^{N-1} (x_n \exp\left(-\frac{j\pi}{N}n^2\right)) \cdot \exp\left(\frac{j\pi}{N}(k-n)^2\right), k = 0, \dots, N-1 \quad (3)$$

Tal somatório pode ser descrito precisamente como uma convolução de 2 sequências a_n e b_n definida por:

$$a_n = x_n \exp\left(-\frac{j\pi}{N}n^2\right) \quad (4)$$

$$b_n = \exp\left(\frac{j\pi}{N}n^2\right) \quad (5)$$

Obtendo uma saída da convolução multiplicada por N fatores de fase b_k^* :

$$X_k = b_k^* \left(\sum_{n=0}^{N-1} a_n b_{k-n} \right), k = 0, \dots, N-1 \quad (6)$$

A principal diferença entre a CZT e a DFT está na flexibilidade de escolher pontos no plano z , permitindo uma análise mais direcionada a faixas de frequência específicas.

A razão pela qual a CZT gera uma espiral no plano z está relacionada ao comportamento dos pontos z_k , que são dados por:

$$z_k = A \cdot W^k, \quad k = 0, 1, \dots, M-1 \quad (7)$$

onde A é o ponto inicial no plano z e $W = e^{-j\frac{2\pi}{M}}$ define o passo angular. Quando $|W| \neq 1$, os pontos z_k não ficam distribuídos ao longo de um círculo, mas seguem uma trajetória espiral. Se $|W| > 1$, os pontos se afastam da origem, formando uma espiral para fora; se $|W| < 1$, eles se aproximam da origem, criando uma espiral para dentro. Esse comportamento é o que confere à CZT sua flexibilidade superior em relação à DFT, permitindo a análise de sinais em regiões específicas do espectro, em vez de uma análise global sobre todo o intervalo de frequência.

2.8 AMBIENTE DE DESENVOLVIMENTO E OTIMIZAÇÃO

A criação de um software robusto, eficiente e de fácil manutenção para automação de medições demanda uma cuidadosa consideração tanto do ambiente de desenvolvimento quanto da estrutura interna do código. A qualidade do design do software e a aplicação de técnicas de programação eficazes são essenciais para garantir não apenas o desempenho computacional, mas também a clareza, a testabilidade e a capacidade de evolução do sistema [13]. Esta seção explorará os pilares que sustentam a construção de softwares de alta qualidade, detalhando a importância do paralelismo computacional para otimização de performance, a versatilidade da linguagem Python como ferramenta principal e os benefícios da Programação Orientada a Objetos para uma arquitetura de código organizada e escalável.

2.8.1 Paralelismo Concorrente: *threads* em Python para Medições de RF

O paralelismo computacional é uma técnica essencial para otimizar o desempenho e a responsividade de softwares, especialmente em tarefas que envolvem grandes volumes de dados ou interações com hardware. Ele permite que múltiplas operações sejam executadas de forma simultânea ou concorrente. No desenvolvimento de software, isso é geralmente alcançado por meio de *threads* ou processos, que são unidades de execução com características distintas.

Threads são as menores unidades de execução dentro de um processo. Elas compartilham o mesmo espaço de memória, o que facilita a comunicação e o acesso a dados compartilhados. Essa característica é particularmente útil para o paralelismo de dados, onde tarefas semelhantes são aplicadas a diferentes partes de um conjunto de dados. Em Python, apesar do *Global Interpreter Lock* (GIL) restringir a execução simultânea de bytecode Python em múltiplos núcleos para operações intensivas em I/O (*Input/Output*). Nesses casos, o GIL é liberado enquanto a *thread* aguarda a conclusão de operações externas, como a comunicação com um analisador de espectro ou a leitura de dados de hardware, permitindo que outras *threads* do programa continuem a executar, na prática, operações de I/O são executadas em outro núcleo de processamento, efetivamente realizando o processo em paralelo. Tal técnica permite garantir a responsividade da aplicação e otimizar o fluxo de trabalho em sistemas de medição em tempo hábil [14]. Contudo, o compartilhamento de memória exige um controle de sincronização para evitar problemas de concorrência e travamento do algoritmo.

Por outro lado, processos são instâncias de programas que operam de forma independente, cada um possuindo seu próprio espaço de memória isolado. Essa característica os torna ideais para o paralelismo de tarefas, onde diferentes funções do programa podem ser executadas em núcleos separados, oferecendo maior isolamento e reduzindo a complexidade de sincronização de dados. A comunicação entre processos (IPC - *Inter-Process Communication*) é geralmente mais lenta e exige mecanismos específicos, como pipes ou memória compartilhada, para a troca de dados. O uso de múltiplos processos é vantajoso quando as tarefas são independentes e não demandam comunicação constante em alta velocidade. Em arquiteturas que visam o máximo desempenho, como as que utilizam FPGAs ou GPUs, a combinação estratégica de paralelismo de *threads* e processos pode maximizar a eficiência na execução de algoritmos complexos, como a Transformada Rápida de Fourier (FFT) ou a Chirp Z-transform (CZT), aumentando a escalabilidade e o desempenho geral [15].

2.8.2 Python

Python é uma linguagem de programação de alto nível, interpretada, e de propósito geral, amplamente utilizada por sua simplicidade, legibilidade e flexibilidade. Criada

por Guido van Rossum e lançada em 1991, Python adota uma sintaxe que permite escrever código de maneira concisa e fácil de entender, o que a torna uma excelente opção tanto para iniciantes quanto para desenvolvedores experientes. A linguagem segue o princípio de "menos é mais" e enfatiza a clareza, o que facilita a manutenção do código[15]. Python suporta múltiplos paradigmas de programação, como orientação a objetos, funcional e imperativa, o que oferece grande versatilidade para diferentes tipos de projetos, desde análise de dados até desenvolvimento web e automação. Além disso, a vasta biblioteca padrão, o ecossistema de pacotes e a própria comunidade envolvida tornam Python uma das linguagens mais populares no desenvolvimento de software, com forte presença em áreas como inteligência artificial, ciência de dados e desenvolvimento de aplicativos web.

2.8.3 Programação orientada ao objeto

A programação orientada a objetos (POO) em Python é um paradigma que organiza o código em torno de objetos, que são instâncias de classes, cada um contendo dados (atributos) e funcionalidades (métodos). Em Python, as classes são definidas com a palavra-chave `class`, permitindo a criação de novas entidades e a definição de seus comportamentos. Esse modelo facilita a reutilização de código e o encapsulamento de dados e funções, promovendo a modularização e uma manutenção de software mais eficiente [16]. Conceitos como herança, polimorfismo e encapsulamento são implementados de forma simples em Python, conferindo grande flexibilidade. A aplicação de POO é crucial para o desenvolvimento de sistemas complexos que buscam confiabilidade, escalabilidade e manutenibilidade, princípios essenciais para sistemas intensivos em dados[15]. Ao modularizar o software de automação em classes e objetos, é possível gerenciar a complexidade das interações com o analisador de espectro e o processamento de sinais de forma mais eficaz, facilitando futuras extensões e atualizações do sistema, alinhado aos princípios de um código limpo e bem estruturado[13]. O uso de Python para POO é amplamente adotado em várias áreas, incluindo desenvolvimento web, análise de dados e automação, devido à sua sintaxe intuitiva e à vasta quantidade de bibliotecas que suportam esse paradigma.

3 METODOLOGIA DE DESENVOLVIMENTO

Este trabalho visou desenvolver algoritmos diversos para automatização de medições utilizando um analisador de espectro. Para criar uma base sólida que permita a modularização e integrações futuras, foi utilizado um código estruturado em Python com programação orientada ao objeto. Dessa forma, para a elaboração do código fonte foi utilizada uma programação orientada ao objeto, a fim de permitir uma separação em blocos robusta, subdividindo o código principal em módulos que juntos permitem a utilização completa do sistema, além de utilizar a API oficial da Tektronix, permitindo a conexão simplificada com o *hardware*. Além de utilizar os artifícios da POO para subdividir o código em classes e métodos, a utilização de processamento paralelo em *threads* foi explorado, a fim de aumentar a eficiência do código, diminuir o peso de processamento em núcleos isolados do processador. Além de otimizar os processamentos, realizar essa manobra traz consequências visíveis para o usuário, se tratando de um processamento em tempo real de sinais.

3.1 BLOCOS FUNCIONAIS

Para promover a melhor legibilidade e organização do código, o programa foi separado em classes distintas, conforme ilustrado no diagrama de blocos da Figura 2. Essas classes são inicializadas em uma sequência pré-determinada, o que aprimora significativamente a escalabilidade, a legibilidade e a funcionalidade geral do software.

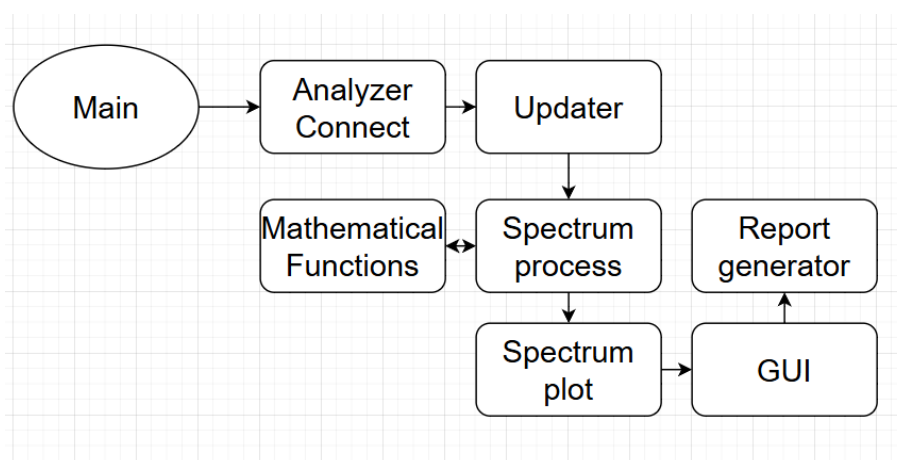


Figura 2: Diagrama de blocos do software desenvolvido

Embora essa abordagem modular possa apresentar desafios adicionais nos estágios iniciais de desenvolvimento, ela se mostra extremamente benéfica em fases mais avançadas. A separação em blocos facilita a implementação de novas funções e o gerenciamento de múltiplas partes do código simultaneamente, contribuindo para um desenvolvimento mais eficiente e uma manutenção simplificada.

- **main:** O coração do programa, responsável por inicializar todas as rotinas necessárias, além de inicializar as classes diretamente atreladas ao código principal.
- **updater:** Esse bloco é responsável por realizar a atualização dos gráficos transmitidos na tela, sendo definidos no bloco os parâmetros de taxa de atualização, assim como chamar as funções responsáveis por realizar a aquisição do sinal no domínio do tempo e na frequência, e das funções que realizam o processamento do sinal, utilizando a alocação de *threads* para maximizar o desempenho do *plot* em tempo real.
- **spectrumPlot:** A função *spectrumPlot* é responsável por definir as configurações artísticas no gráfico, como largura do pincel, cor do traço, limites dos eixos nos gráficos. Além de ser o recipiente para os sinais obtidos na função *updater*.
- **controlButton:** A função *controlButton* é responsável por armazenar os métodos de ativação de botões utilizados na interface interativa do usuário.
- **analyzerConnect:** A função *analyzerConnect* por realizar a conexão com o analisador de espectro. Essa função faz as primeiras configurações essenciais do analisador de espectro, além de realizar a captura do sinal, alocando espaço na memória a fim de obter um vetor que condiz com o sinal capturado no domínio do tempo ou na frequência.
- **spectrumProcess:** Responsável por conter todos os processamentos matemáticos disponíveis no programa, como as funções de *max hold*, média móvel, banda ocupada e potência integrada, assim como as eventuais novas funções presentes no código.

3.2 FUNÇÕES MATEMÁTICAS

Para possibilitar a manipulação dos vetores capturados pelo analisador de espectro, foi necessário realizar algumas manipulações matemáticas para que os resultados pudessem ser lidos de forma clara e visual.

- **max_hold:** Para realizar um *max hold*, que consiste em reter o maior valor dos vetores recebidos, é necessário alocar um vetor dedicado para os valores de amplitudes máximas para cada ponto de frequência e realizar uma comparação entre o vetor com o máximo e o vetor com o dado adquirido em tempo real, assim como exemplificado na Figura 3.

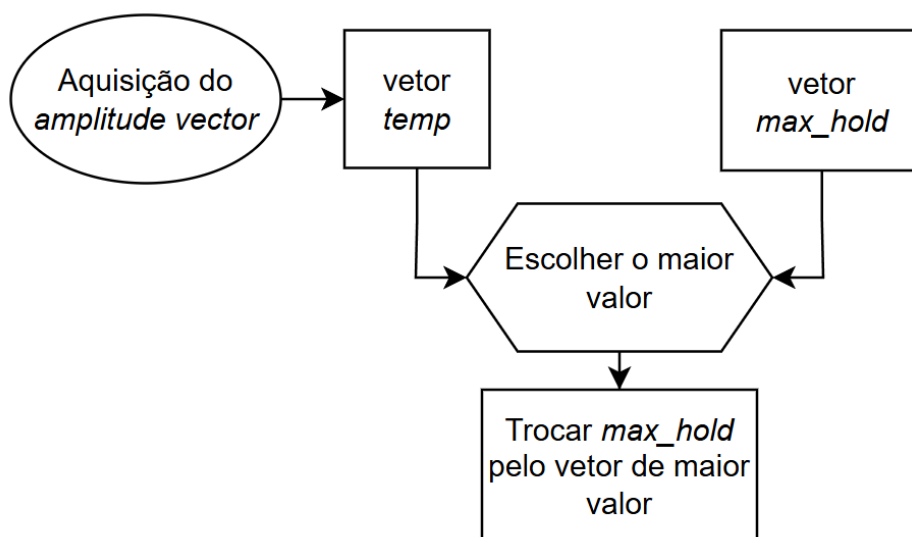


Figura 3: Diagrama funcional para max hold

- **moving_average:** Para criar uma média móvel, a fim de exibir no visor uma curva que represente o sinal recebido sem as variações causadas por ruídos e modulação do sinal, foi alocado um vetor bidimensional, com dimensões iguais ao número de pontos capturados pelo analisador de espectro e à quantidade de pontos para realizar a média, seguindo a seguinte equação.

Considerando N =número de pontos para a média; L =Número de pontos adquirido; $C = c_{LN}$ = Matriz contendo os vetores para média; $B = b_L$ =vetor contendo as

médias

$$b_i = \frac{\sum_{j=0}^{N-1} c_{ij}}{N} \quad (8)$$

Dessa forma, para aplicar o algoritmo, foi utilizado a seguinte lógica explicitada em Figura 4:

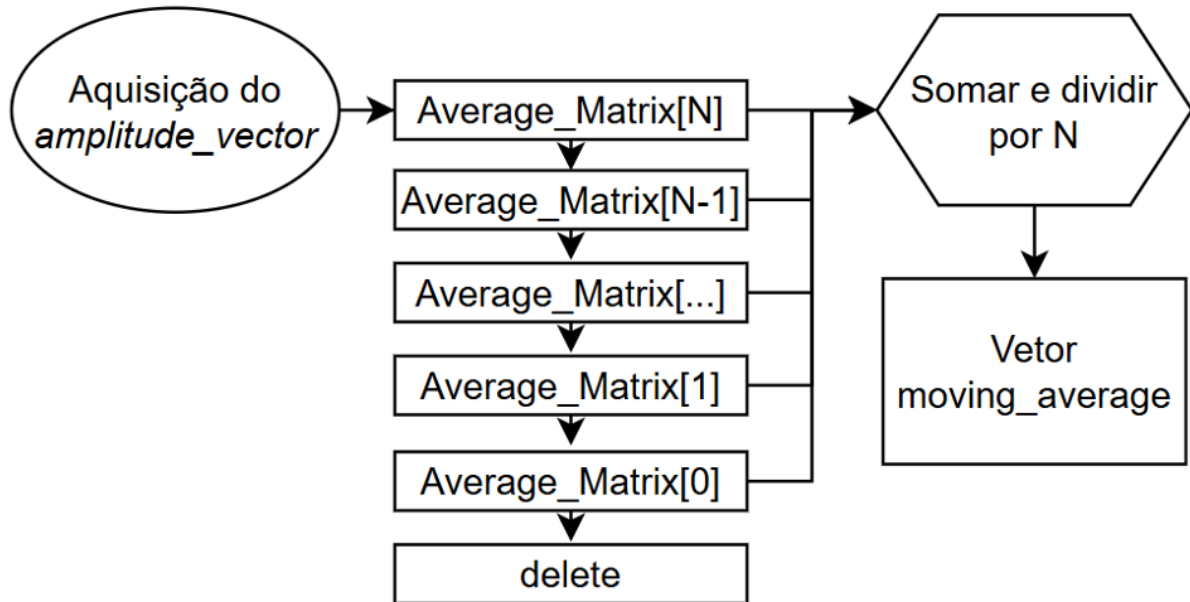


Figura 4: Diagrama funcional para média móvel

- **potência_integrada:** A potência integrada é uma das métricas importantes para a medição de sistemas de comunicação, pois ela mostra a potência efetivamente transmitida em um canal de comunicação dentro de uma determinada largura de banda. Com isso, é possível analisar o desempenho e a eficiência do sistema como um todo. Para realizar o cálculo de potência integrada, foi necessário ajustar os parâmetros de comprimento de vetor com resolução da largura de banda (RBW) e com a largura da janela de captura, o *span*. Assim, chegou-se à seguinte fórmula:

$$P_{ch} = \left(\sum_{n=0}^{L-1} 10^{\frac{a_n}{10}} \right) \cdot \frac{span \cdot \sqrt{2}}{TraceLenght \cdot RBW} \quad (9)$$

A partir dessa equação, temos o seguinte diagrama funcional em Figura 5:

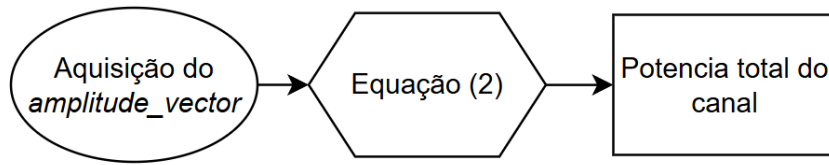


Figura 5: Diagrama funcional para potência integrada

- **banda_ocupada:** O conceito de banda ocupada é muito importante para sinais de telecomunicações e radiofrequência, visto que representa a porção de banda que um sinal transmitido ocupa no espectro. Essa medida é importante pelo fato das faixas de frequência serem disponibilizadas através de concessões de forma seletiva, com algumas faixas destinadas para uso livre. Para o cálculo da banda ocupada (OBW), foi aplicado o critério de 99% da potência do sinal, uma exigência estabelecida pelos Atos da Anatel para garantir a conformidade regulatória. Utilizando esse critério, a soma parcial de potência foi realizada para cada *bin* presente no vetor de aquisição, dentro da janela de observação. Dessa forma, as bordas laterais que delimitam a banda ocupada foram definidas nos pontos em que a potência acumulada atingiu 0,5% (para a borda esquerda) e 99,5% (para a borda direita) da potência total do sinal.

Para borda lateral esquerda:

$$\sum_{n=0}^L a^n \cdot 0.005 = \sum_{i=0}^x a^i \quad (10)$$

Para a borda lateral direita:

$$\sum_{n=0}^L a^n \cdot 0.995 = \sum_{i=0}^y a^i \quad (11)$$

Em que x representa o bin da borda lateral esquerda e y representa a borda lateral direita. Passando as equações para diagrama funcional, temos o diagrama em Figura 6:

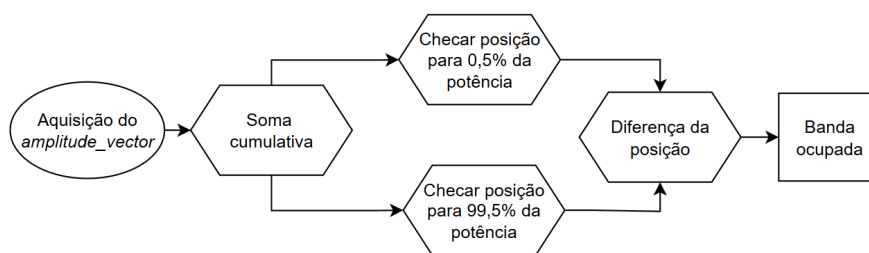


Figura 6: Diagrama funcional para banda ocupada

3.3 PDF GENERATOR

O Gerador de Relatórios (PDF Generator) é um bloco funcional crucial dentro do sistema de automação de medições de RF desenvolvido neste projeto, responsável por consolidar e formalizar os resultados dos testes de conformidade. Sua principal função é orquestrar a execução de uma sequência pré-definida de medições e, subsequentemente, compilar os dados coletados e as análises realizadas em um formato de relatório padronizado e acessível, escrito em latex e com saída direta para PDF. Este bloco garante a reprodutibilidade dos ensaios e a rastreabilidade dos resultados, elementos fundamentais para a metrologia e a certificação de dispositivos.

O fluxo de operação do Gerador de Relatórios é iniciado a partir de um arquivo de configuração, geralmente no formato CSV (Comma Separated Values), que atua como um script de teste. Este arquivo é cuidadosamente pré-montado com base nas normas e requisitos específicos, como os estabelecidos pela Agência Nacional de Telecomunicações (ANATEL) para a homologação de equipamentos de radiocomunicação. Cada linha do arquivo CSV pode especificar um conjunto de parâmetros de medição (e.g., frequência central, span, RBW, tempo de aquisição) e os tipos de análise a serem realizados (e.g., potência de pico, largura de banda ocupada, potência integrada, máscara de emissão).

Ao ser acionado, o Gerador de Relatórios interage com os demais blocos do sistema para:

- Interpretar o Script de Teste: Lê e processa as instruções contidas no arquivo CSV, convertendo-as em comandos específicos para o analisador de espectro e para os algoritmos de processamento de sinais.

- Orquestrar Medições: Envia os comandos apropriados para o módulo de controle do analisador de espectro (via API), configurando o equipamento para cada teste especificado no script.
- Coletar e Processar Dados: Recebe os dados brutos do analisador e os encaminha para os algoritmos de processamento de sinais (como os de cálculo de largura de banda ocupada, potência de pico e média móvel) para extração das métricas relevantes. A aplicação de técnicas de paralelismo computacional neste estágio é crucial para garantir a eficiência na coleta e análise de grandes volumes de dados.
- Consolidar Resultados: Agrega todos os resultados numéricos, gráficos espectrais e outras informações pertinentes de cada teste realizado.
- Gerar Relatório PDF: Utiliza bibliotecas de geração de PDF para formatar os resultados de maneira clara e padronizada. O relatório inclui todas as medidas relevantes (banda ocupada, potência de pico, potência integrada, informações de configuração do teste, etc.).

Para melhor exemplificar esse fluxo de geração de relatório, o diagrama de blocos da Figura 7 apresenta as etapas principais do processo.

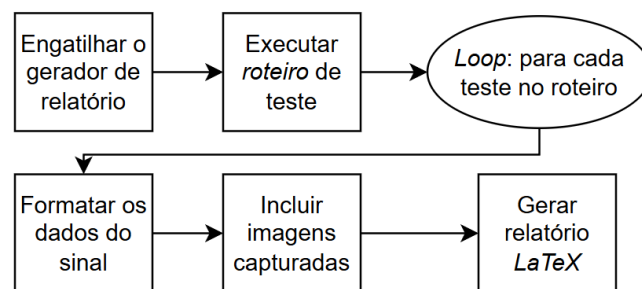


Figura 7: Diagrama funcional para o PDF Generator

A autonomia e a capacidade de documentação formal conferidas pelo Gerador de Relatórios elevam o sistema de automação a uma ferramenta completa para laboratórios de teste, pesquisa e desenvolvimento de produtos IoT, minimizando a intervenção manual e garantindo a integridade e a reprodutibilidade dos ensaios.

4 RECURSOS NECESSÁRIOS

A fim de realizar o projeto, foi necessário realizar a disposição dos equipamentos em um ambiente de laboratório, garantindo que os dados obtidos sejam confiáveis; assim, o *setup* apresentado na Figura 8 foi utilizado

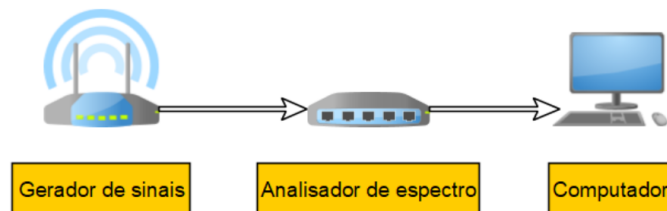


Figura 8: Diagrama de blocos do setup utilizado

Como o foco do trabalho é o analisador de espectro, um gerador de sinais (ou RF) confiável foi utilizado a fim de validar as aquisições que são realizadas.

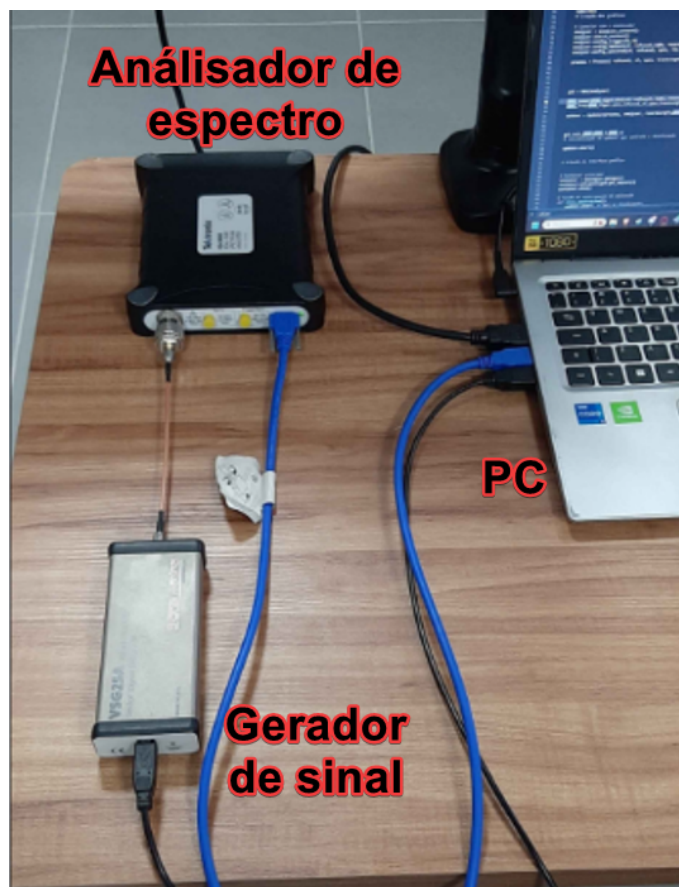


Figura 9: Setup utilizando o gerador VSG25A

Com o *setup* montado de acordo com a Figura 9 para garantir a confiabilidade do sinal gerado com o lido, foi utilizado uma conexão via cabo coaxial blindado, evitando interferências externas e garantindo um caminho quase ideal de comunicação entre o gerador e o analisador, ambos se conectando ao computador utilizando uma conexão USB 3.0.

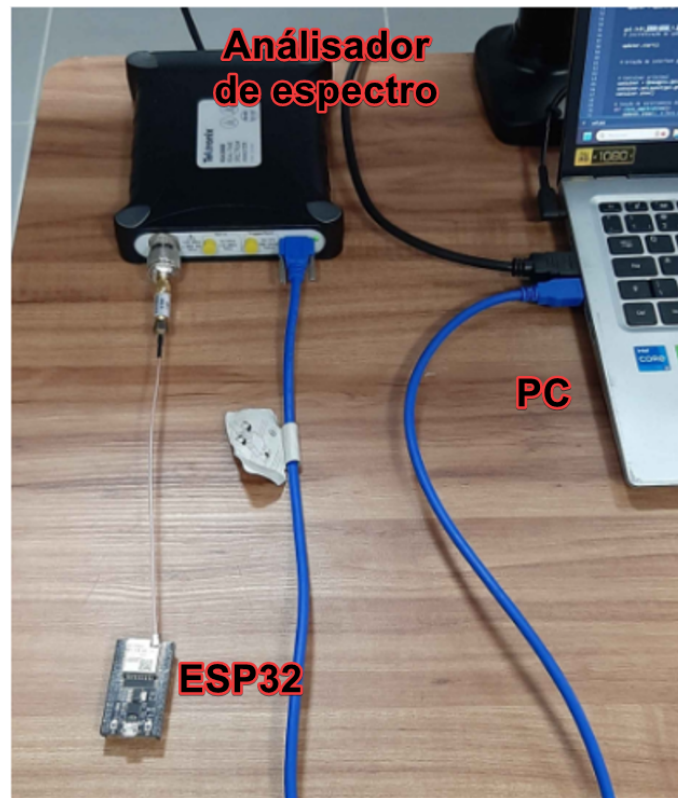


Figura 10: Setup utilizando o ESP32

Seguindo a mesma linha, o ESP32 foi utilizado para realizar emulações de sinais Wi-Fi com modulações que são amplamente utilizados em dispositivos IoT e o setup de acordo com a Figura 10 foi utilizado.

4.1 RSA306B USB SPECTRUM ANALYZER

A escolha dos equipamentos adequados é fundamental para a precisão e a confiabilidade das medições em radiofrequência. Para o desenvolvimento deste trabalho, o Analisador de Espectro USB RSA306B da Tektronix, ilustrado na Figura 11, foi selecionado utilizado para a elaboração desse projeto.



Figura 11: Analisador de espectro RSA306B, da Tektronix

O RSA306B é um analisador de espectro da Tektronix, compacto e com um custo significativamente mais baixo, variando geralmente entre R\$ 19.000 e R\$ 25.000 no Brasil. Este valor contrasta drasticamente com analisadores de bancada convencionais de capacidade similar (especialmente aqueles com funcionalidades de tempo real), que podem facilmente custar de R\$ 60.000 a R\$ 100.000 ou mais. Ele possui uma faixa de aquisição na frequência de 9 kHz a 6.2 GHz, uma faixa de aquisição de potência de +20 dBm a -160 dBm, uma taxa de captura de 112 Ms/s em frequência intermediária e 56 Ms/s em banda base, e realiza a transformada Chirp Z em sua arquitetura interna de FPGA [17]. Com essas características, é possível utilizar a API fornecida pela própria Tektronix para realizar medições, visto que os dados brutos são transferidos pela USB 3.0, permitindo o processamento digital de sinais em software com total liberdade entre RBW, *trace length* e *span*, considerando as implicações já discutidas sobre a CZT.

4.2 SIGNALHOUND VSG25A

A fim de gerar um sinal conhecido para analisar com o analisador de espectro, é necessário utilizar alguma fonte confiável de sinal, dessa forma, o gerador de sinal VSG25A, da Signal Hound, apresentado na Figura 12, mostra-se eficaz nesse quesito.



Figura 12: Gerador de sinais VSG25A

Este gerador também conta com um software próprio, o Signal Hound Vector Signal Generator, apresentado na Figura 13, que será utilizado para controlar o gerador de sinal para a realização dos experimentos.

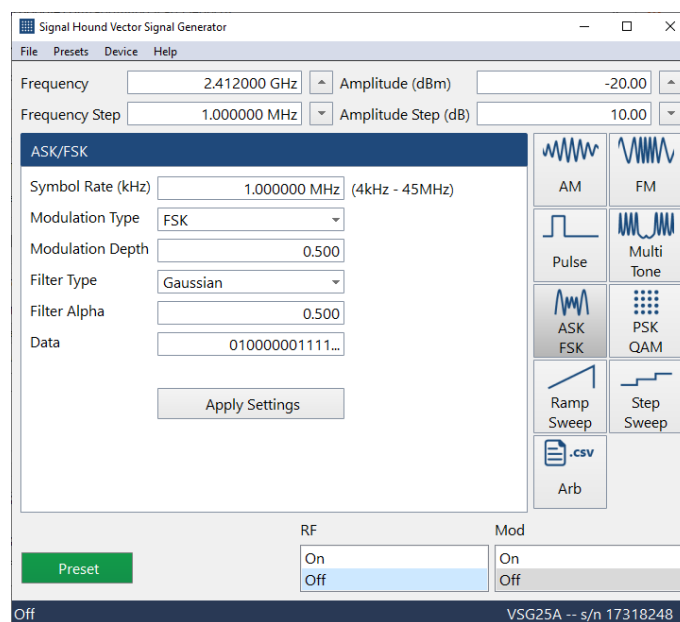


Figura 13: Software do gerador de sinal

Com o software apresentado na Figura 13, é possível gerar tons senoidais e sinais modulados, tanto em modulações simples, como AM e FM, quanto em modulações mais complexas, como QAM e FSK, ou até mesmo alguma modulação arbitrária determinada pelo usuário. Esse equipamento permite gerar sinais em uma faixa de frequência de 100 MHz à 2,5 GHz e modular o sinal com uma faixa de 54 kHz à 180 MHz.

4.3 ESP32

O ESP32 , apresentado na Figura 14 é um dos microcontroladores mais versáteis na área da eletrônica, inclusive para geração de sinais. Dessa forma, ele também pode ser utilizado como um gerador de sinal válido para realizar os testes com o analisador de espectro.



Figura 14: ESP32, utilizado como emulador Wi-Fi

A fim de emular transmissões de pacotes OFDM utilizando os protocolos atuais de Wi-Fi e Bluetooth, que são necessários para testes de homologação para a anatel, foi utilizado o software "RF Test Tool", disponibilizado pela ESPRESSIF:

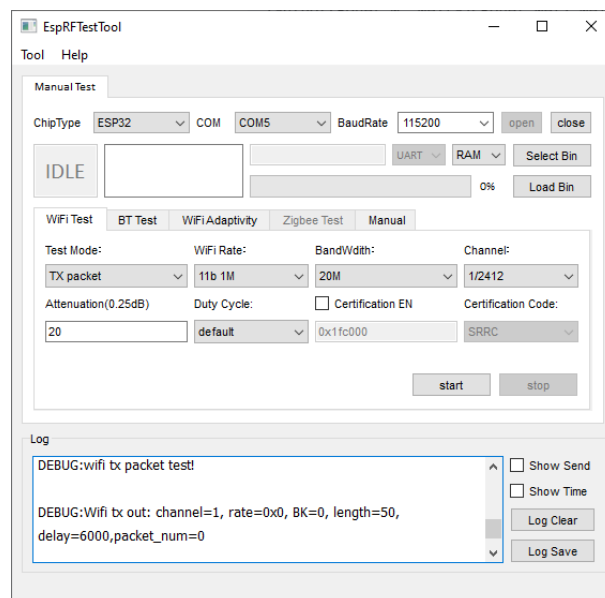


Figura 15: EspRFTTestTool, software utilizado para gerar os sinais

Com a utilização desse software, é possível fazer com que o ESP32 gere o mesmo tipo de sinal estabelecido para comunicações Bluetooth e Wi-Fi. Dessa forma, medições que utilizam transmissões pulsadas que necessitam da atuação do *trigger*, presente no analisador de espectro, podem ser realizadas e utilizadas para validação do software desenvolvido.

5 RESULTADOS

Este projeto alcançou resultados significativos que demonstram avanços substanciais na automação e análise de medições de sinais de radiofrequência (RF) para aplicações em Internet das Coisas (IoT). O analisador de espectro utilizado como base para a realização deste trabalho foi o Tektronix RSA306B, selecionado pela confiabilidade do equipamento, pelo alto desempenho e disponibilidade de sua API (Application Programming Interface) pública em [GitHub.com](https://github.com), o que possibilitou uma abordagem de desenvolvimento aberto e independente de softwares proprietários.

Como um software que faz a integração de processamento de sinal com a aquisição em tempo real de dados pré-tratados pelo equipamento, foi necessário realizar o desenvolvimento de uma interface gráfica que permitisse ao usuário ter controle sobre os parâmetros necessários para uma medição satisfatória, dentre tais parâmetros, cabe salientar a frequência central, a largura de banda de aquisição, o nível de referência para o conversor analógico-digital (ADC) do equipamento, a largura de banda do filtro de resolução (RBW), além de outros parâmetros necessários, assim como apresentado na Figura 16:

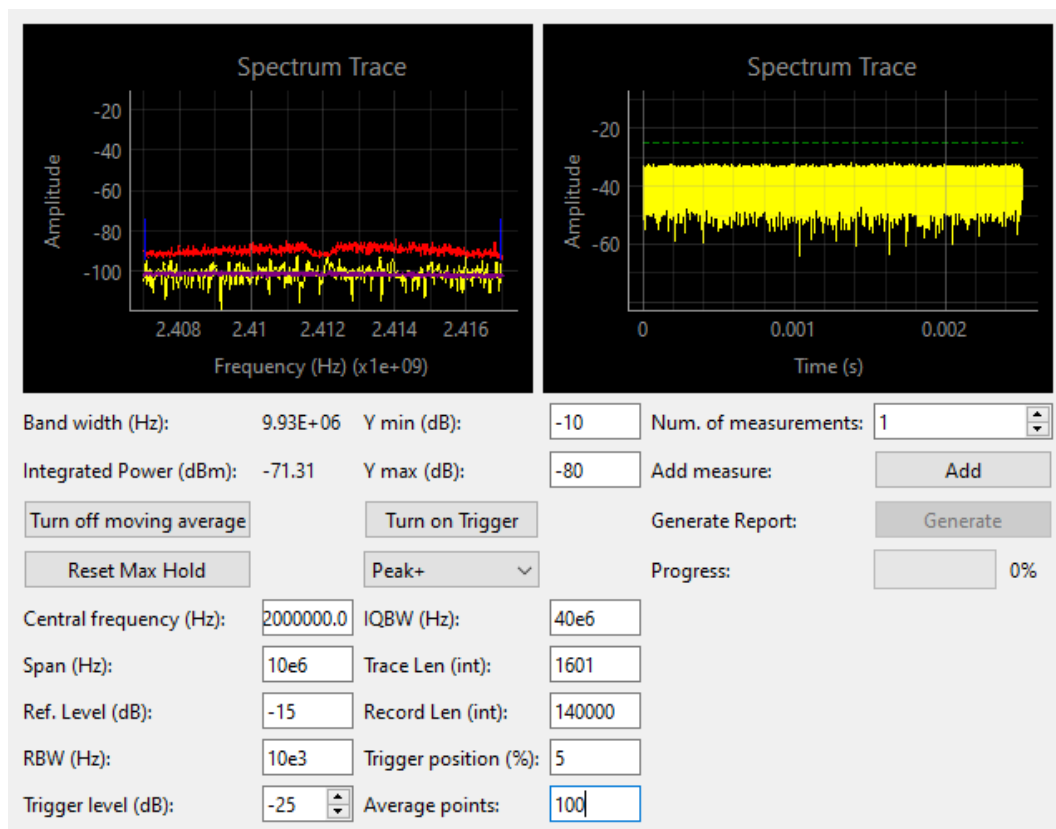


Figura 16: Interface gráfica do projeto realizado

Com essa interface, é possível não somente alterar parâmetros de aquisição do equipamento, mas também é possível alterar a resolução visual dos gráficos, graças ao equipamento utilizar a CZT para computar as transformadas do domínio do tempo para o domínio da frequência. Também é possível fazer o ajuste do trigger do equipamento em tempo real, possibilitando o usuário ter um feedback imediato ao realizar medições de sinais pulsados, assim como protocolos modernos de Wi-Fi e Bluetooth.

Para exemplificar o funcionamento do sinal com modulações conhecidas, foi realizado duas aquisições de sinais distintos. A primeira aquisição foi realizada com uma transmissão de um sinal com modulação Wi-Fi 11b com 1MHz de banda, apresentado na Figura 17. Já a segunda aquisição foi realizada com um sinal *Bluetooth Low Energy*, apresentado na Figura 18.

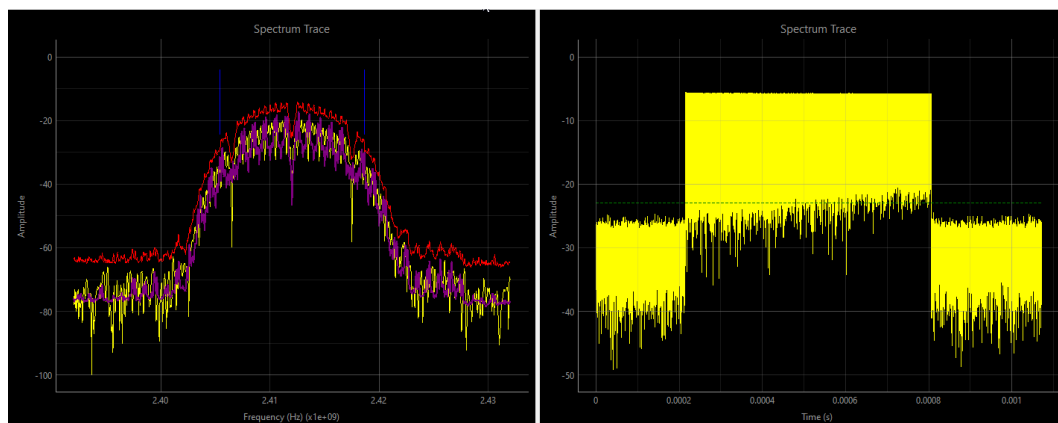


Figura 17: Exemplo de aquisição de um sinal Wi-Fi 11b 1M

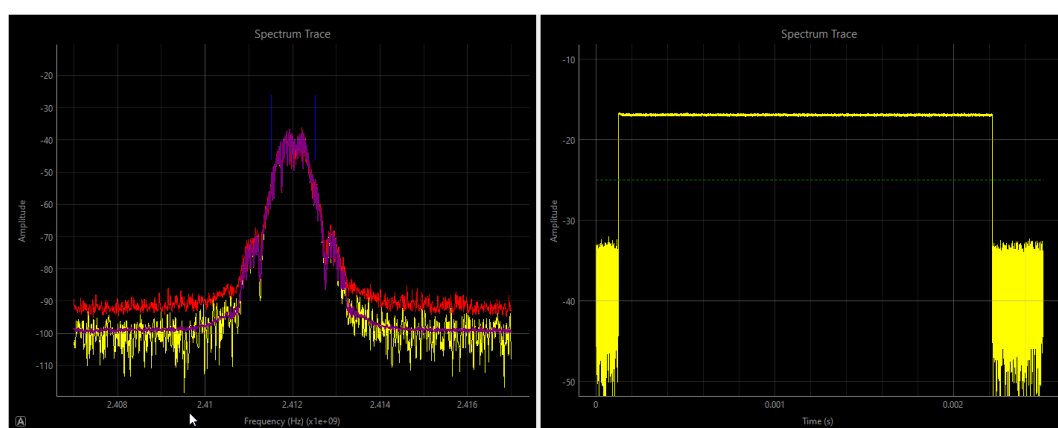


Figura 18: Exemplo de aquisição de um sinal Bluetooth Low Energy (BLE)

Com a exemplificação de capturas de sinais nas Figuras 17 e 18, é notável a apresentação de traços com cores distintas que representam aspectos importantes para a interpretação visual dos dados, conforme detalhado a seguir:

- Sinal em Tempo Real (Amarelo): Representa o espectro bruto do sinal capturado diretamente do analisador de espectro, sem qualquer processamento ou suavização adicional. Dada a natureza dinâmica e estocástica dos sinais de radiofrequência em ambientes reais, que são influenciados por fatores como ruído, interferência e multitrajetória, este traço exhibe alta volatilidade. Ele reflete as variações instantâneas do sinal no tempo, servindo como uma representação fiel, porém ruidosa, do que é recebido pelo instrumento.

- **Máxima Potência (Vermelho):** Representa o traço de *Max Hold* (retenção de máxima potência), que registra o valor de potência mais elevado para cada ponto de frequência ao longo de um período de observação contínua. Esta funcionalidade é particularmente útil para a captura e análise de sinais transientes, intermitentes ou pulsados, que poderiam ser perdidos em uma varredura de tempo real. Ao preservar os picos de potência mais altos, o Max Hold permite identificar o envelope máximo das emissões de RF, sendo essencial para verificar picos de interferência ou a máxima potência de transmissão de um dispositivo.
- **Limites de Banda Ocupada (Azul):** Delimita visualmente a largura da banda ocupada (**OBW** - *Occupied Bandwidth*) do sinal. Esta medida é crucial para a conformidade regulatória, pois indica a porção do espectro de frequência que um sinal ocupa de forma significativa. O cálculo é realizado com base em um critério de potência pré-definido pelo usuário (comumente 99% da potência total do sinal), em alinhamento com padrões de teste da indústria e normas como as da ANATEL para homologação de equipamentos de radiocomunicação. A visualização desses limites permite uma rápida avaliação da eficiência espectral e da aderência do sinal aos requisitos de canalização.
- **Média Móvel (Roxo):** Apresenta a média móvel do sinal, com o tamanho do vetor definido pelo usuário. No domínio da frequência, a principal vantagem da média móvel é a redução significativa da variância do ruído de fundo (chão de ruído). Ao suavizar as flutuações aleatórias, ela realça os componentes de sinal verdadeiros, tornando o espectro observado visualmente mais limpo e facilitando a identificação de picos de sinal, a análise de suas formas e a delimitação de suas larguras de banda, mesmo em ambientes com baixo nível de sinal ou alta interferência. Com esse processo, torna-se possível observar com maior clareza como a modulação do sinal e as condições do canal de rádio estão afetando a potência média, facilitando a identificação de tendências e a avaliação da estabilidade do sistema de comunicação ao longo do tempo.

5.1 VALIDAÇÃO DOS RESULTADOS

Os testes foram concentrados na banda de 2,4 GHz, uma faixa de frequência crítica para diversas aplicações de Internet das Coisas (IoT), incluindo Wi-Fi (IEEE 802.11b/g/n) e Bluetooth Low Energy (BLE).

Para a geração de sinais de validação, foram utilizados sinais com características conhecidas (e.g., um tom contínuo ou um sinal modulado simples) foram gerados pelo Signalhound VSG25A para fornecer uma mesma base de referência para ambos os testes.

Em primeira instância, foi realizado a validação da largura de banda com critério de 99% da potência e a potência integrada do sinal na banda de observação. Como ponto de partida, foi definido uma largura de banda de 5MHz, um RBW de 100kHz e foi mantido o nível de referência em 0dBm. Para o gerador de sinal, foi arbitrado 2 tipos de sinais conhecidos, primeiro um tom senoidal centrado em 2,412GHz e, segundo, um sinal modulado em 256QAM, com taxa de símbolo de 1MHz. Com os testes, foi obtidos os seguintes resultados:

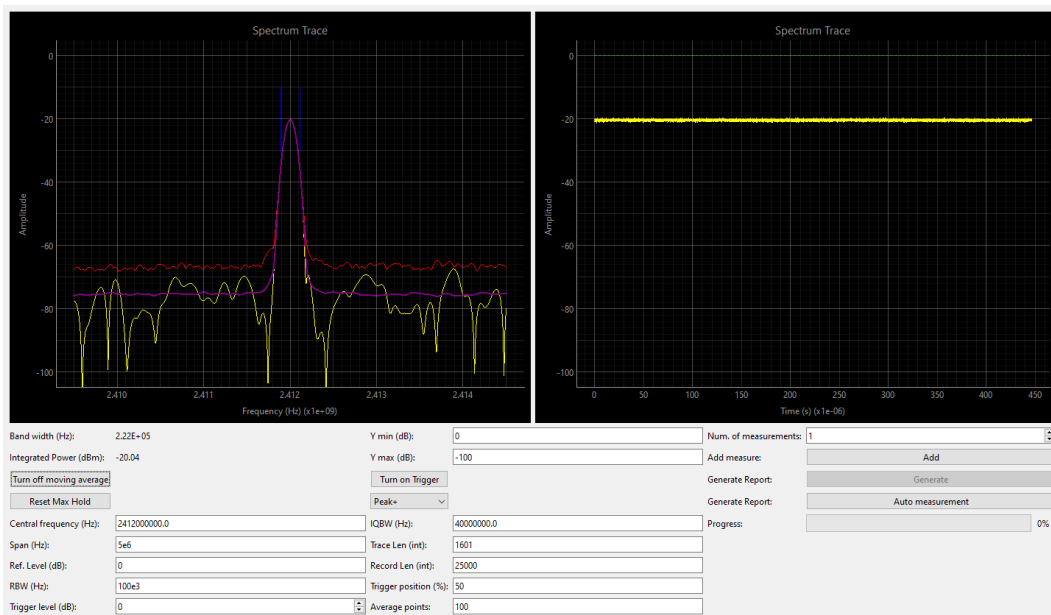


Figura 19: Tom senoidal capturado pelo software desenvolvido

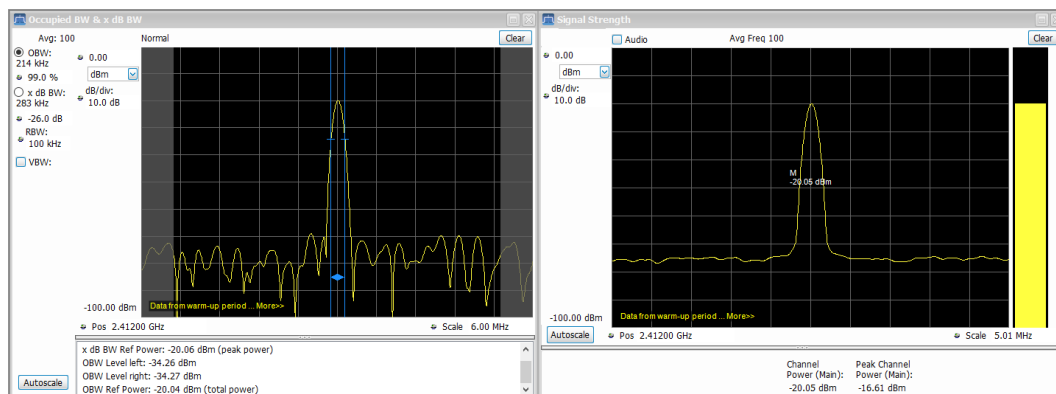


Figura 20: Tom senoidal capturado pelo SignalVu

Tabela 1: Tom senoidal: Software Desenvolvido vs. Tektronix SignalVu-PC

Métrica	Software	SignalVu	Dif. Abs.	Dif. %
Potência Integrada	-20.04 dBm	-20.05dBm	0.01dBm	0.23%
OBW, 99%	222 KHz	214KHz	8 KHz	3.74%

Como pode ser visto na Tabela 1, a diferença entre a captura da potência de sinal de um tom senoidal obteve um ótimo desempenho, com um erro percentual de 0,23% ao se comparar com o software proprietário. Além dessa métrica, o método de cálculo de banda ocupada também se mostrou promissor para um tom senoidal após passar por um RBW acima de sua largura de banda verdadeira, obtendo uma diferença percentual de 3,74%, permitindo a viabilidade do projeto até o momento.

Passando para um sinal com uma largura de banda um pouco maior, é feito o comparativo com um sinal 256QAM com 1MHz de taxa de símbolo, com sequência de símbolos aleatórios gerada por um bloco PN9. Com isso, foram obtidos os seguintes resultados:

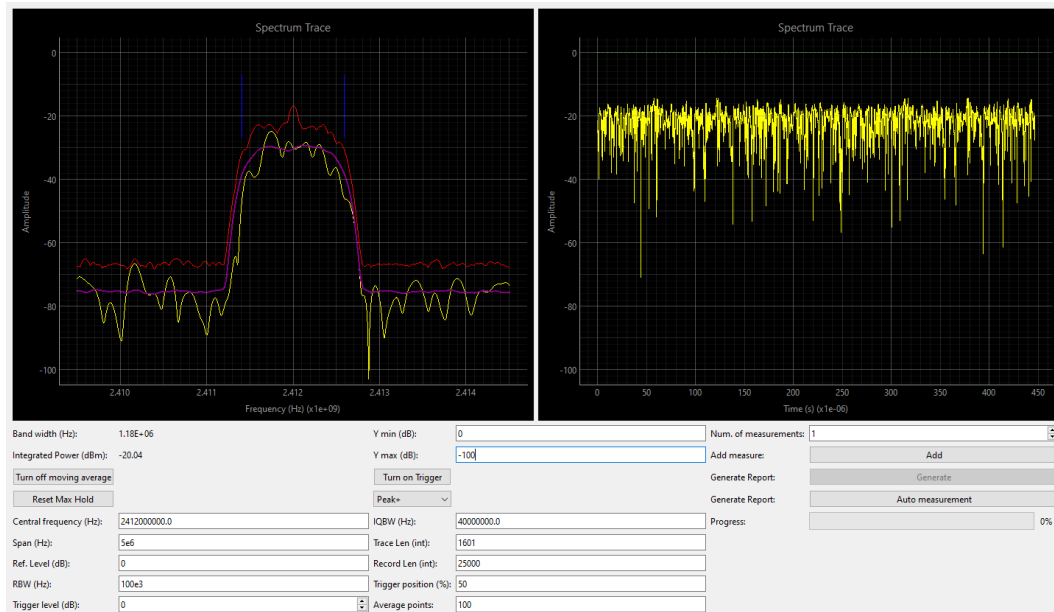


Figura 21: 256QAM capturado pelo software desenvolvido

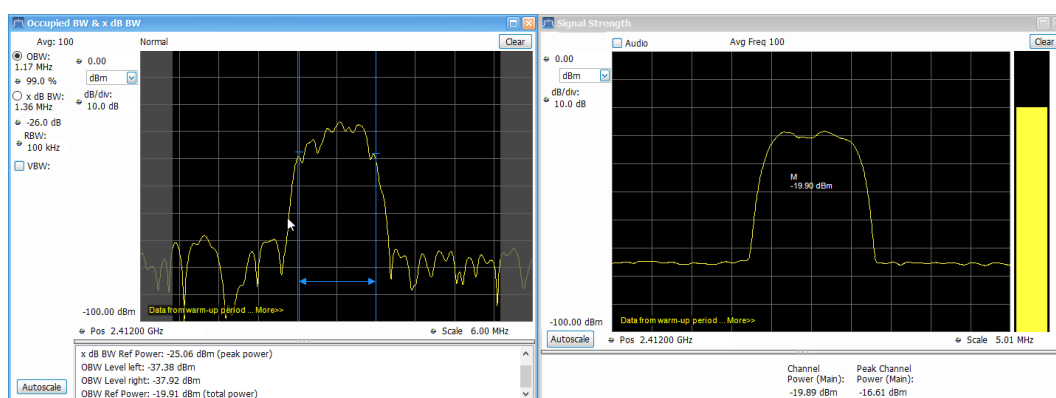


Figura 22: 256QAM capturado pelo SignalVu

Tabela 2: 256-QAM: Software Desenvolvido vs. Tektronix SignalVu-PC

Métrica	Software	SignalVu	Dif. Abs.	Dif. %
Potência integrada	-20.04 dBm	-19.90dBm	0.14dBm	3.17%
OBW, 99%	1.17MHz	1.18MHz	0.1MHz	0.85%

Como observado na tabela 2, mesmo com um sinal com uma largura de banda maior, o software desenvolvido obteve um resultado satisfatório, mantendo o erro percentual abaixo de 5%, demonstrando a consistência dos resultados obtidos através do

processamento dos sinais adquiridos.

Seguindo para a validação da automação das medições, foram definidas duas rotinas de testes, cada uma com 10 medições: uma para o *SignalVu-PC* e outra para o programa desenvolvido. O objetivo foi comparar o tempo gasto para realizar as medições. As rotinas consistiram em:

- Ligar o programa, medindo o tempo que o software leva para inicializar a partir do momento em que o executável é acionado.
- Inserir os valores para definir os parâmetros de aquisição do sinal, como frequência central, largura de banda e *RBW*.
- Realizar a captura da imagem da onda adquirida.
- Anotar os parâmetros relevantes, como largura de banda ocupada e potência do sinal.
- Transferir os dados adquiridos para um arquivo *Word* para elaboração de relatório.

Tabela 3: Tempos de Medição com *SignalVu-PC*

Medição	Anotar os valores	Inserção em relatório
1	24s	32s
2	28s	38s
3	25s	36s
4	31s	34s
5	22s	37s
6	23s	35s
7	25s	33s
8	28s	34s
9	21s	31s
10	23s	31s
Desvio padrão	3,13s	2,56s
Total	250s	349s

Tabela 4: Tempos de Medição com o programa desenvolvido

Medição	Aquisição de sinal e inserção em relatório via software
1	2s
2	3s
3	1,5s
4	1,9s
5	1,5s
6	2,3s
7	2,4s
8	2s
9	1,6s
10	1,7s
Desvio padrão	0,47s
Total	19,90s

Como o software desenvolvido já realiza a inserção de figuras e dados relevantes diretamente no relatório, o tempo total entre as medições realizadas pela solução desenvolvida e pelo *SignalVu-PC* possui uma grande diferença, conforme observado nas Tabelas 3 e 4.

Tabela 5: Comparativo de tempo gasto em medição

	Software	SignalVu	Dif. Abs.	Dif.%
Inicialização	5s	45s	40s	900%
Medição	19,9s	250s	230,1s	1.256%
Inserção em relatório	–	349s	–	–
Total	24,9s	644s	619,1s	2.586%

A partir da Tabela 5, foi possível validar que o software desenvolvido possibilita uma economia de tempo significativa em ambientes profissionais, resultando em uma diferença de aproximadamente 10 minutos (ou o valor exato em segundos, ex: 619,1s) para realizar a aquisição de 10 pontos de medição. Esta análise considerou apenas

a captura e anotação/transferência inicial de dados. Ao considerar que o software desenvolvido realiza a inserção automática dos dados em um modelo pré-definido em *LaTeX*, eliminando a necessidade de formatação manual do relatório final, a diferença no tempo total gasto em um cenário de uso real aumenta significativamente.

A partir disso, os resultados obtidos a partir da comparação do software desenvolvido com o ambiente proprietário Tektronix *SignalVu-PC* demonstram a robustez e a acurácia da solução proposta. A consistência observada nas medições de potência integrada e largura de banda ocupada (OBW) para diferentes tipos de sinais (tom senoidal e 256QAM) na banda de 2,4 GHz valida a eficácia dos algoritmos implementados. Apesar das pequenas variações percentuais, que são inerentes às diferenças de implementação algorítmica e à precisão de cada sistema, os desvios encontrados estão dentro de uma margem aceitável para aplicações práticas em IoT. Com isso, conclui-se que o software de medição e automação de medições de RF desenvolvido oferece uma alternativa viável e confiável às soluções comerciais, permitindo a realização de testes de conformidade e caracterização de sinais com uma precisão aceitável e necessária para o desenvolvimento e homologação de dispositivos, ao mesmo tempo em que proporciona a flexibilidade e a autonomia de uma ferramenta de código aberto.

6 CONCLUSÃO

Este trabalho de conclusão de curso teve como objetivo geral o desenvolvimento de uma solução de automação robusta para medições de sinais de radiofrequência (RF) em um analisador de espectro controlado por software, focando na coleta, tratamento e análise de dados para aplicações em Internet das Coisas (IoT). Ao longo do projeto, os objetivos específicos propostos foram integralmente alcançados, culminando em uma ferramenta funcional e validada.

A conexão eficiente e confiável com o analisador de espectro foi plenamente estabelecida, utilizando a API oficial do Tektronix RSA306B para gerenciar a comunicação e a aquisição de dados em uma cadência otimizada para o processamento subsequente. Isso permitiu a interação direta com o hardware para configurações de parâmetros e coleta de dados brutos, eliminando a dependência de softwares proprietários adicionais e garantindo uma comunicação robusta e programável. Adicionalmente, as rotinas de aquisição de dados nos domínios do tempo e da frequência foram implementadas com sucesso. Os módulos desenvolvidos possibilitam a captura contínua do espectro de RF e dos dados I/Q no domínio do tempo, aproveitando a capacidade de CZT interna do analisador e a manipulação desses dados brutos no software para visualização e análise em ambos os domínios, atendendo às características de sinais de radiofrequência para IoT.

Um pilar central do projeto foi o desenvolvimento de algoritmos para o tratamento e análise de sinais de radiofrequência. Funções de Max Hold, Média Móvel, Potência Integrada e Banda Ocupada foram implementadas com êxito, transformando os dados brutos do analisador em métricas relevantes, essenciais para a avaliação de desempenho de sistemas de comunicação. A validação desses algoritmos em cenários representativos de aplicações em IoT foi efetivada através de testes com um gerador de sinal (VSG25A) e um emulador Wi-Fi/Bluetooth (ESP32). Essa abordagem permitiu simular condições operacionais típicas de dispositivos IoT na banda de 2,4 GHz, confirmando a capacidade do sistema em lidar com uma variedade de sinais relevantes para esse ambiente.

A acurácia da solução proposta foi demonstrada pela comparação da precisão dos

resultados obtidos com as medições automatizadas em relação a medições manuais e ao software proprietário SignalVu-PC. As diferenças percentuais para Potência Integrada (0,23% para tom senoidal, 3,17% para QAM256) e Banda Ocupada (3,74% para tom senoidal, 0,85% para QAM256) foram consistentes e mantiveram-se dentro de uma margem aceitável, validando a confiabilidade dos algoritmos implementados. Complementarmente, a avaliação e comparação do tempo de execução das medições automatizadas em relação às manuais confirmou a eficiência do processo. A automação e o paralelismo por *threads* demonstraram ser capazes de reduzir drasticamente o tempo necessário para processar grandes volumes de dados (de dias para horas), sendo no mínimo 25,86 vezes mais rápidas. Isso otimiza significativamente a produtividade do laboratório e supera as limitações dos métodos manuais, com ganhos práticos que tendem a ser ainda maiores.

Em suma, este trabalho resultou no desenvolvimento de uma ferramenta de software eficaz e acessível para a automação de medições de RF. Ao combinar a flexibilidade do Python e os princípios da Programação Orientada a Objetos com a capacidade de um analisador de espectro via API, o projeto oferece uma alternativa viável e precisa às soluções comerciais. A capacidade de realizar medições confiáveis, reproduzíveis e em larga escala, com validação metrológica e transparência no processamento, é um diferencial significativo para a engenharia de testes, pesquisa acadêmica e homologação de equipamentos no crescente ecossistema da IoT. Os resultados consolidam a importância da automação para a eficiência e a qualidade na análise de sinais de RF.

TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se a expansão da biblioteca de algoritmos de processamento de sinais, incluindo outras manipulações de sinais. A integração com outros tipos de instrumentos de RF, aumentando o grau de automação para não somente o dispositivo de aquisição de sinal, mas também o dispositivo que gera o sinal e o dispositivo sob teste. A exploração do paralelismo computacional utilizando uma linguagem mais robustas para esse efeito, como C/C++. Além disso, a investigação da viabilidade técnica de uma jiga de testes compacta baseada em FPGA, pode abrir

portas para soluções de teste ainda mais eficientes e de baixo custo.

Referências

- [1] Computer History Museum, “Semiconductor Test Equipment Development: Oral History Panel,” Transcrição de Painel de História Oral, 2009, gravado em 14 de agosto de 2009, Mountain View, Califórnia. Referência CHM: X5501.2010.
- [2] D. M. Pozar, *Microwave Engineering*. John Wiley & Sons, 2011.
- [3] K. B. Schaub and J. Kelly, *Production Testing of RF and System-on-a-Chip Devices for Wireless Communications*. Artech House, n.d.
- [4] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Prentice Hall, 2010.
- [5] IoT Business News, “State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally,” *IoT Business News*, Sep. 2024. [Online]. Available: <https://iotbusinessnews.com/2024/09/04/26399-state-of-iot-2024-number-of-connected-iot-devices-growing-13-to-18-8-billion-global/>
- [6] Brasil, “Lei nº 9.472, de 16 de julho de 1997: Dispõe sobre a organização dos serviços de telecomunicações, a criação e funcionamento de um órgão regulador e outros aspectos institucionais, nos termos da Emenda Constitucional nº 8, de 1995.” http://www.planalto.gov.br/ccivil_03/leis/L9472.htm, 1997.
- [7] Agência Nacional de Telecomunicações (Anatel), “Ato nº 14448,” Diário Oficial da União.
- [8] —, “Ato nº 237,” Diário Oficial da União.
- [9] Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO), “Página Inicial do INMETRO,” <https://www.gov.br/inmetro/pt-br>.
- [10] —, “Rede Brasileira de Calibração (RBC),” <https://www.gov.br/inmetro/pt-br/assuntos/avaliacao-da-conformidade/acreditacao/redemetrologica>.

- [11] *Understanding RF Signal Analyzers*, National Instruments, Austin, TX, USA, 2020.
- [12] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [13] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008.
- [14] M. J. Quinn, *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Education, 2003.
- [15] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017.
- [16] A. J. Riel, *Object-Oriented Design Heuristics*. Addison-Wesley Professional, 1996.
- [17] Tektronix. (2024) RSA306B USB Real-Time Spectrum Analyzer: Datasheet. [Online]. Available: <https://www.tek.com/en/datasheet/spectrum-analyzer-0>