



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
DEPARTAMENTO: CENTRO DE ENGENHARIA ELÉTRICA E
INFORMÁTICA

ESPECTÔMETRO

DISCENTES:

- Guilherme Araújo Ferreira
- Maria Isabel de Barros Azevedo
- Pedro Daniel Chaves Monteiro de Almeida Seixas
- Raylson Gabriel Moraes da Silva

CAMPINA GRANDE – PARAÍBA

2025

SUMÁRIO

1 - Introdução	1
2 - Motivação	1
3 - Objetivos	1
4 - Metodologia	1
5 - Problemas Encontrados e Aprendizados	4
6 - Resultados	4
7 - Possíveis melhorias	7
8 - Conclusão	8
9 - Contribuição dos integrantes	8
10 - Referências	9

1 - Introdução

Este relatório apresenta o desenvolvimento de um projeto realizado na disciplina de Introdução à Engenharia Elétrica, no semestre 2025.1.

O objetivo principal é criar um aplicativo capaz de receber sinais de um arduino e processá-los, permitindo a análise de seus componentes por meio do uso de uma transformada de Fourier.

Por mais que o foco do projeto seja no aplicativo em si, foram também utilizados circuitos amplificadores e filtros para poder realizar o teste da sua eficácia, além de diversas fontes de sinais para serem analisados, provando sua versatilidade.

2 - Motivação

A principal motivação do projeto é o estudo de sinais analógicos, pois possibilita compreender de forma mais profunda as suas composições e fontes, por meio da análise de apenas certas faixas de frequência.

Além disso, a prática com aquisição, amplificação e processamento de sinais fortalece o entendimento de eletrônica e telecomunicações, ensinando conceitos muito importantes para o resto do curso.

3 - Objetivos

De forma geral, o objetivo do projeto é desenvolver um sistema capaz de captar, amplificar, filtrar e analisar sinais analógicos, aplicando a Transformada de Fourier para identificação das frequências componentes. E, de forma mais específica, temos os seguintes objetivos:

1. Captar sinais analógicos de diferentes origens.
2. Implementar amplificação utilizando amplificadores operacionais.
3. Aplicar filtragem com circuitos RC.
4. Converter os sinais analógicos para digitais com Arduino.
5. Criar software em Java para análise espectral do sinal.

4 - Metodologia

1. Utilização de diversas fontes como: piezoelétrico para captação de ondas mecânicas provenientes de celulares e instrumentos musicais; plug fêmea P10 para captação de ondas eletromagnéticas provenientes de uma guitarra; antena caseira feita com 70 voltas de fio de cobre em torno de um aro de 60cm de diâmetro para captação de ondas eletromagnéticas atmosféricas e ruídos da rede elétrica urbana; e plug P3 para captação de ondas elétricas provenientes de um celular. Essas fontes tinham o objetivo de testar o funcionamento do aplicativo, com destaque ao piezoelétrico, que permitia

com que os sinais também fossem ouvidos (pelo fato de serem ondas mecânicas, permitindo a vibração do ar), assim facilitando a percepção de falhas na captação e amplificação.

2. Idealização do circuito com o uso do programa de simulação Falstad (modelo abaixo), para depois fazer a implementação na protoboard. Foi utilizado um OpAmp modelo LM324 (representado pelo triângulo no centro) para realizar a amplificação de até 250 vezes por meio de um circuito não-inversor. No modelo final, foi utilizado um potenciômetro no lugar do resistor de 250k Ω , para permitir um controle melhor da amplificação.

3. Utilização de um filtro RC com um capacitor de 22nF e um resistor de 10k Ω (mostrado logo antes da saída analógica), caracterizando um frequência de corte de aproximadamente 700Hz, que se mostrou muito eficaz na redução de ruído do circuito.

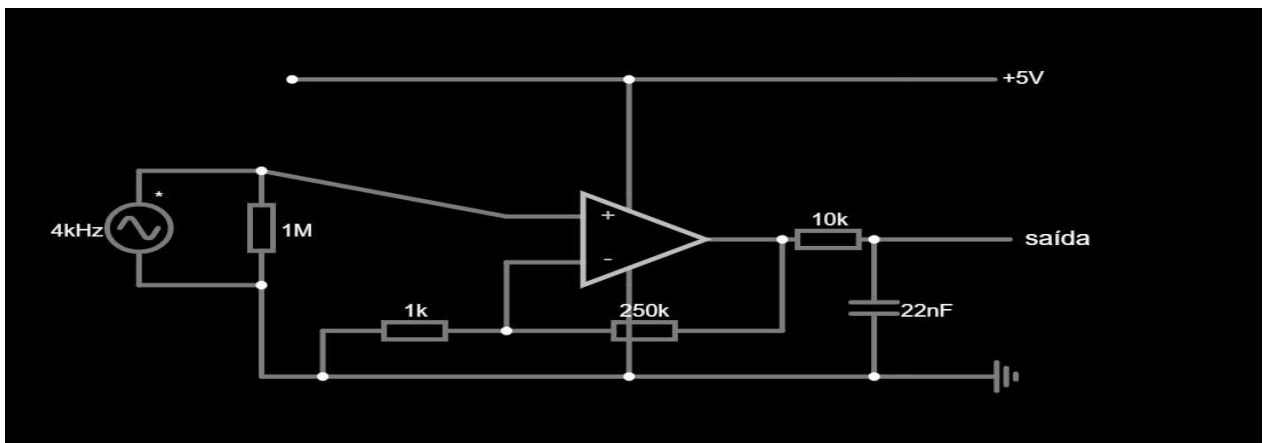


Imagem do circuito utilizado quando a fonte era um piezoelétrico

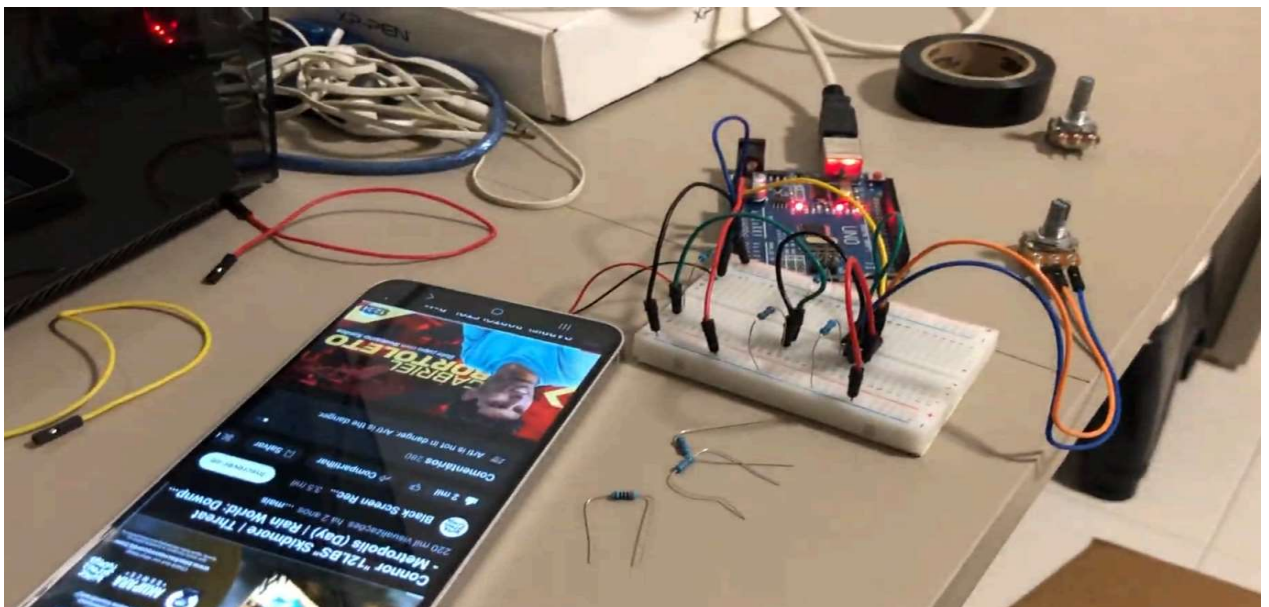


Foto de um dos protótipos, usando um celular e um piezoelétrico (abaixo do celular) para captar as vibrações de uma música nos alto falantes do aparelho.

4. A saída analógica do circuito tem destino a porta A0 de um arduino uno, que possui um código para transformar os dados analógicos em digitais, para então enviar por meio de um cabo usb esses dados para um computador. Apesar do arduino uno não ser adequado para esse tipo de aplicação, por conta de sua baixa capacidade de processamento, ele foi suficiente para a realização dos testes e seu desempenho foi satisfatório.

```
espectometro.ino
1  void setup() {
2      Serial.begin(115200);
3  }
4
5  void loop() {
6      int v = analogRead(A0);          // 0-1023
7      uint8_t out = v >> 2;           // shrink to 0-255
8      Serial.write(out);              // send one byte
9  }
10
```

Código utilizado pelo arduino. Ocorre a leitura da porta A0, transformação do dado lido em 1 byte, e envio desse byte para porta serial USB. Processo de ocorre milhares de vezes por segundo, capaz de distinguir sinais de até 4kHz

5. Por fim, a utilização de um aplicativo desenvolvido na linguagem Java, com a utilização da IDE processing. Por meio da utilização da ferramenta “serial” da IDE, o aplicativo capta os dados da porta USB de forma rápida e eficaz, sendo um dos principais motivos para a utilização dessa linguagem, uma vez que o Python (linguagem que tentamos utilizar no começo), é muito lento, e não consegue acompanhar a velocidade que os dados chegavam. Enquanto ocorre o recebimento dos dados em tempo real, ocorre a sua plotagem em um gráfico, mostrando o sinal em uma linha verde em tempo real. Além disso, por meio da biblioteca “minim”, é possível realizar a transformada de Fourier, onde um outro gráfico representa as frequências primitivas do sinal, por meio de barras azuis, permitindo a sua análise. E para concluir, adicionamos uma interface simples com a biblioteca “controlP5”, permitindo com que o usuário escolha a porta USB para ser utilizada, aplique uma amplificação digital (diminui a resolução do sinal, mas permite uma visualização melhor em sinais que a amplificação analógica não é suficiente), mude a velocidade de plotagem do gráfico em verde, mude as janela de frequências da transformada de Fourier e habilite a compensação logarítmica na transformada, para facilitar a visualização. Assim, a versão final possui 146 linhas de código, capaz de receber e analisar sinais de até 4kHz.

Link para download do código (pode ser aberto com o bloco de notas) - >

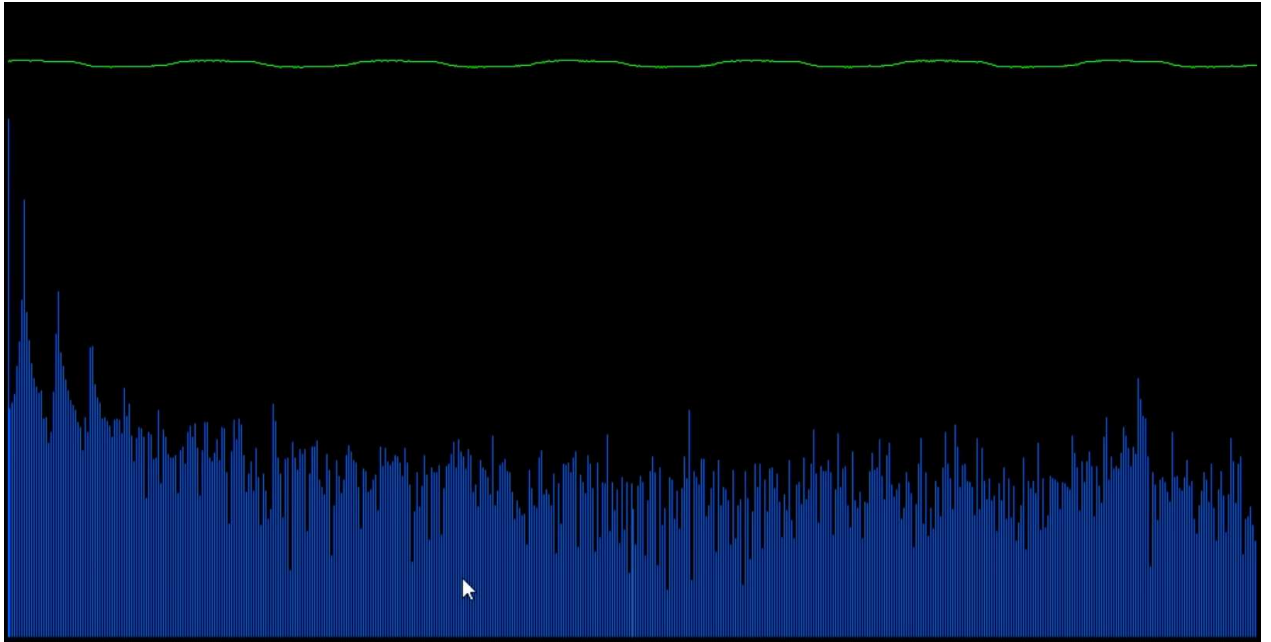
5 - Problemas Encontrados e Aprendizados

- Inicialmente, tentou-se utilizar Python para a plotagem dos sinais, mas o tempo de plotagem mostrou-se elevado, impossibilitando a sua utilização em tempo real. Isso motivou a busca por linguagens mais rápidas e eficazes, apesar de mais complexas. Foi considerada a migração para C, entretanto, o uso de bibliotecas de renderização apresentou alto grau de dificuldade. Por fim, a migração para Java, utilizando a IDE Processing, mostrou-se uma solução eficiente e equilibrada para a plotagem em tempo real dos dados recebidos do Arduino, sem a complexidade excessiva de C.
- Por conta do processamento baixo do arduino uno, foi necessário limitar o intervalo de frequências captadas pelo aplicativo para até 4kHz, uma vez que frequências mais altas não conseguem ser plotadas em uma resolução suficiente para transformada de Fourier mesmo com uma redução do número de variáveis e mudanças na estrutura de algoritmos para otimizar o processamento.
- Persistiram problemas relacionados à ruído no próprio arduino, que não puderam ser resolvidos (não era possível realizar uma filtragem, pois o problema ocorria entre a porta A0 do arduino e a saída USB). Porém, esses problemas não afetaram de forma substancial o resultado dos testes uma vez que o objetivo principal do projeto é separar o sinal em suas componentes, possibilitando isolar a frequência do ruído da frequência sendo observada.

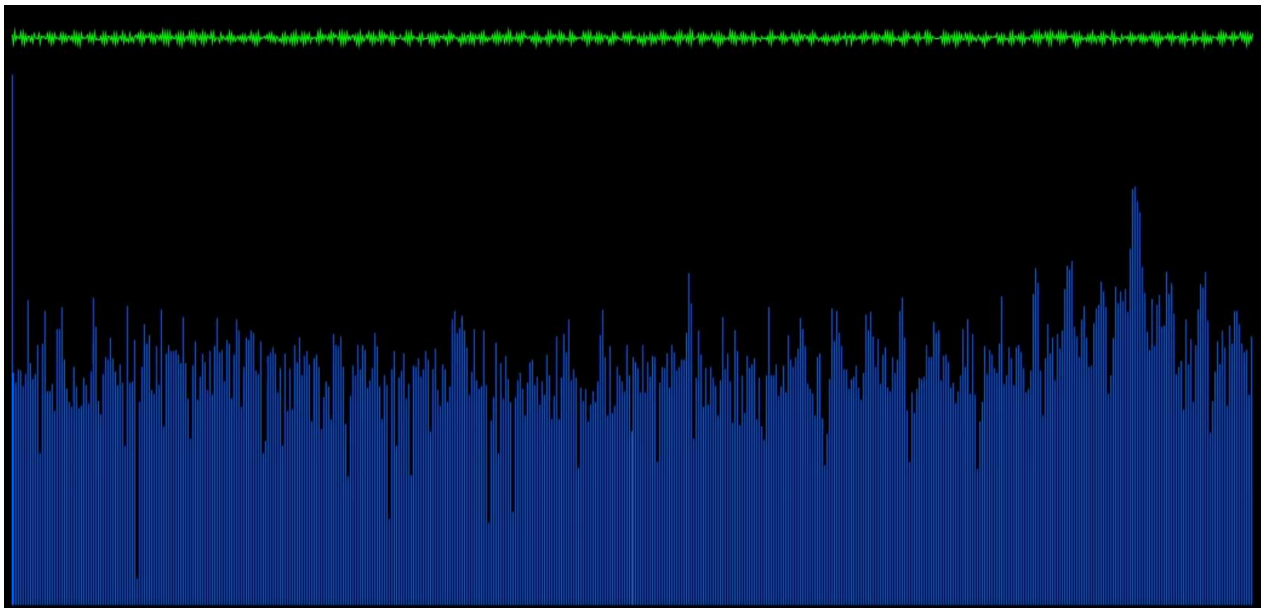
6 - Resultados

Apesar de alguns problemas de ruído com o arduino (o próprio arduino gerava o ruído, logo não era possível filtrá-lo) o resultado do projeto foi extremamente satisfatório. Com o aplicativo, é possível visualizar as diversas faixas de frequência de qualquer sinal, sendo possível distinguir, por exemplo, diferentes instrumentos musicais de uma música, harmônicos presentes em notas musicais, ruído presente na rede elétrica, entre outros.

Abaixo estão algumas imagens e links de vídeos de testes realizados ao longo do desenvolvimento do projeto, demonstrando a sua funcionalidade e eficácia:



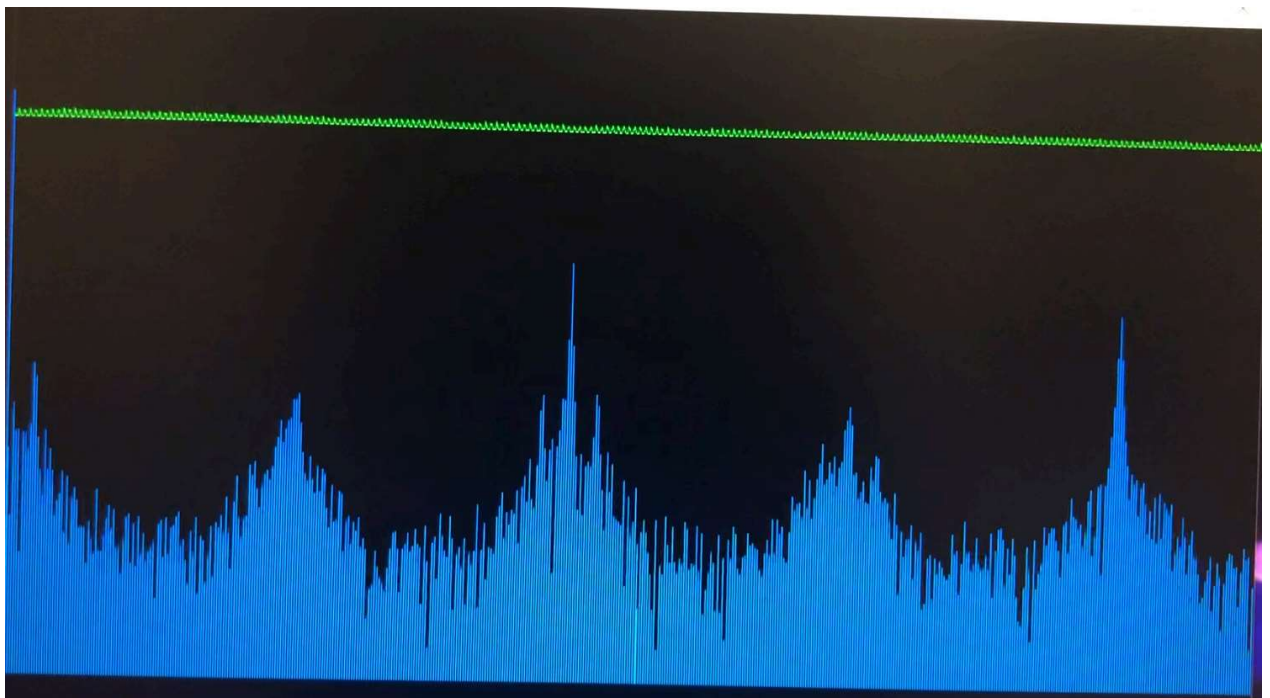
Sinal recebido por uma antena. Note a presença do ruído em 60Hz com alguns harmônicos, causado pela aproximação da antena a um carregador de celular ligado na tomada (nas primeiras etapas do protótipo ainda não havia a indicação da frequência abaixo da transformada de Fourier, porém é possível estimar as frequências sabendo que o intervalo é de 4kHz).



Outro sinal recebido pela antena, dessa vez sem a utilização do filtro RC. Note a presença de um ruído na faixa de 3,5kHz (também presente na imagem anterior, porém com menos intensidade). Não identificamos a causa exata desse sinal, mas acreditamos que seja a faixa de alguma rádio AM da região.



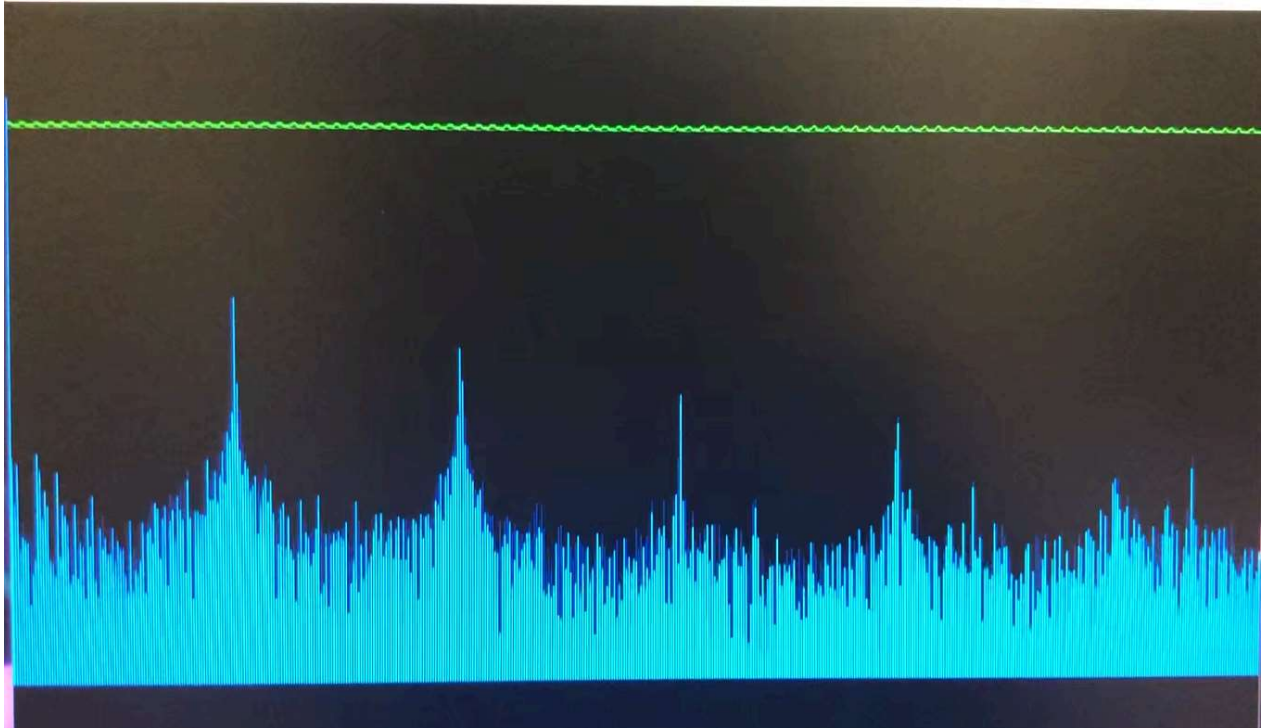
Teste utilizando um gerador de frequências em um celular e o piezoelétrico. Note a limitação no nosso circuito de captar a tensão negativa do sinal, problema que não conseguimos resolver, porém não impossibilita a utilização do aplicativo, e ainda entrega um resultado satisfatório na transformada de Fourier.



O mesmo teste, porém utilizando uma frequência mais alta (aproximadamente 2kHz). Note a presença dos harmônicos. Note também a queda na resolução do sinal, em razão do processamento limitado do arduino em frequências maiores. O teste utilizando o gerador de frequências foi um dos mais úteis, pois

permitiu com que nós analisássemos o funcionamento do aplicativo com precisão absoluta, uma vez que tínhamos total controle do sinal. Link para visualização completa do teste ->

<https://drive.google.com/file/d/1d9c55RJKwLd8MAVDGnO-i0J02J2THISE/view?usp=sharing>



Sinal de uma nota de um piano, captado com o piezoelétrico (note novamente a presença dos harmônicos). Link para o vídeo completo desse teste ->

<https://drive.google.com/file/d/1CXM4FNse3WsJtQnzJjbznPTixnF8Z9pd/view?usp=sharing>

Por fim, segue o link de uma pasta no google drive, com 10 vídeos de testes realizados ao longo do desenvolvimento do projeto, incluindo o espectro de algumas músicas e dos testes citados nesse relatório (recomendamos fazer o download dos vídeos para ver na maior qualidade):

<https://drive.google.com/drive/folders/1P9q1whGIFoISBqQ7-TyG3fVSQoT2alsO?usp=sharing>

7 - Possíveis Melhorias

- Ajuste da amplificação utilizando voltagem intermediária, permitindo o arduino captar tensões negativas, melhorando a precisão na transformada de Fourier.
- Migração definitiva para C, possibilitando maior taxa de amostragem (sample rate).
- Implementação de técnicas de filtragem mais avançadas, incluindo filtragem por indução.

- Substituição do arduino uno por um microcontrolador mais robusto, como o ESP32 ou arduino giga, que oferece maior taxa de amostragem e maior capacidade de processamento.

8 - Conclusão

Captar e analisar sinais se provou uma tarefa extremamente difícil, porém muito interessante e produtiva, com diversas lições aprendidas, que certamente deixaram marcas positivas no nosso conhecimento sobre eletrônica e telecomunicações, ensinando conceitos valiosos para o futuro como amplificação, filtragem, captação e plotagem de sinais e dados.

9 - Contribuição dos Integrantes

O desenvolvimento deste projeto contou com a participação ativa de todos os integrantes do grupo, cada um contribuindo em áreas específicas de acordo com seus conhecimentos e habilidades. Essa divisão de tarefas foi fundamental para a conclusão do trabalho, uma vez que permitiu a integração de diferentes etapas do processo, desde a concepção do circuito até a interface do software utilizada para análise. A seguir, são descritas as principais responsabilidades de cada integrante:

- Raylson Gabriel Moraes da Silva: Atuou no desenvolvimento da interface do usuário (UI) do software. Foi responsável por criar uma interface mais intuitiva e interativa, possibilitando uma melhor experiência na visualização dos dados coletados e na análise espectral. Sua contribuição garantiu que os resultados pudessem ser acessados e compreendidos de forma clara.
- Guilherme Araújo Ferreira: Responsável pela idealização e implementação do circuito eletrônico. Trabalhou diretamente na captação e amplificação dos sinais, assegurando que os dados fossem transmitidos corretamente para posterior processamento.
- Pedro Daniel Chaves Monteiro de Almeida Seixas: Ficou encarregado do desenvolvimento do back-end em Java, realizando a integração entre o Arduino e o software de análise. Foi responsável pelo processamento dos dados e pela estruturação lógica do sistema, possibilitando a aplicação da Transformada de Fourier (FFT).
- Maria Isabel de Barros Azevedo: Atuou na construção prática do circuito, realizando a montagem e a prototipação. Também foi responsável pela implementação e testes com antenas e outros sinais, como piezoelétricos, essenciais para a validação do sistema.

10 - Referências

MALVINO; **Eletrônica Volume 2**; Editora McGraw-Hill, Ltda; 1987.

WIKIPEDIA CONTRIBUTORS; **Amplificador operacional**; Disponível em:
<https://pt.wikipedia.org/w/index.php?title=Amplificador_operacional&oldid=66937299>.

PROCESSING; **Reference and Documentation**; Disponível em:
<<https://processing.org/reference/>>. Acesso em: 25 set. 2025.

SMITH, Steven; **The Scientist and Engineer's Guide to Digital Signal Processing**; 1997.