



**AFINADOR DE VIOLÃO ATIVO AUTOMÁTICO MOTORIZADO**

**Santo André – SP**

**2024**

## **Etec JULIO DE MESQUITA**

Relatório técnico apresentado ao Curso Técnico em Eletrônica da Etec Júlio de Mesquita, orientado pelo professor Robson Fractucello como requisito parcial para obtenção do título de Técnico em Eletrônica.

**Anderson da Silva  
Bruna da Silva Ferreira  
Elias Gabriel Dos Reis Paschoal  
Gabriel Silva Fernandes Daré  
Iago Oliveira Pataro**

**Santo André – SP  
2024**

## **AGRADECIMENTOS**

O desenvolvimento deste trabalho de conclusão do curso Técnico em Eletrônica contou com a ajuda de diversas pessoas, aos Professores Orientadores: Robson Fractucello, Caio Roberto Dos Santos e Egmar Accetto, que nos ajudou alcançar e finalizar esse projeto

E aos demais professores que durante 1 Ano e 6 meses de Curso, que nos acompanharam nesse trajeto, com seus conhecimentos e ensinamentos, também fizeram parte da conclusão desse projeto

A Deus Por guiar-nos nesses caminhos a quais iremos trilhar.

## **RESUMO**

Existem várias formas de afinar um violão desde de aplicativos de celular que auxiliam o cliente e o próprio afinador de violão, porém no nosso trabalho tivemos a ideia de automatizar essa tarefa para esse mercado, afinador com entrada de cabo USB-C , escolha dessa entrada segue pelo fato de que a entrada está se tornando universal em dispositivos pequenos que necessitam carregar sua bateria, o projeto vai conter os seguintes componentes: Arduino , motor DC com caixa de redução, encaixe cabo P10 ,Amplificadores, LCD I2C , Bateria.

Palavras – Chaves: violão, Cabo USB-C, instrumento

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	6
1.2 Objetivos .....	7
1.3 Justificativas .....	8
<b>2 Desenvolvimento</b> .....	9
2.1 metodologia .....	10
2.2 materiais importante do protótipo.....	11
Arduino uno r3 .....	11
2.2.2 Motor DC com caixa de Redução .....	12
2.2.3 Plug j10.....	13
<b>3. Funcionamento do circuito</b> .....	14
3.1 VBIAS .....	14
3.2 Capacitores de Desacoplamento .....	15
<b>4. Programação</b> .....	16
Código .....	16
Código .....	17
Código .....	18
Código .....	19
Código .....	20
Código .....	21
Código .....	22
<b>5. Análise de Custo</b> .....	23
<b>6. Testes</b> .....	24
Fotos .....	24
Foto.....	25
Foto.....	26
<b>7. CONCLUSÃO</b> .....	26
<b>8. REFERÊNCIAS</b> .....	27
<b>9. APÊNDICE</b> .....	28

## **1. INTRODUÇÃO**

O Afinador de violão Ativo Motorizado Podemos dizer que é um avanço perceptível com a tecnologia para instrumentos musicais, esse tipo de Afinador conta com sensores processamento digitais é atuadores motorizados.

Este Dispositivo pode ser integrado diretamente no violão ou conectado como acessório externo, dando uma maior visualização para os músicos de como estão a afinação da corda.

## 1.2 Objetivos

O Afinador de violão tem como objetivo principal ajustar a tensão das cordas do instrumento à frequência correta, garantindo que cada corda emita a nota musical correspondente. Isso é fundamental para que o violão soe afinado e agradável ao tocar, seja em acordes, melodias ou solos.

Existem diversos tipos de afinadores automático disponíveis no mercado, cada um com suas características e vantagens:

- Precisão e facilidade.
- Melhoria na qualidade do som
- Maior versatilidade
- Dispositivo Portátil

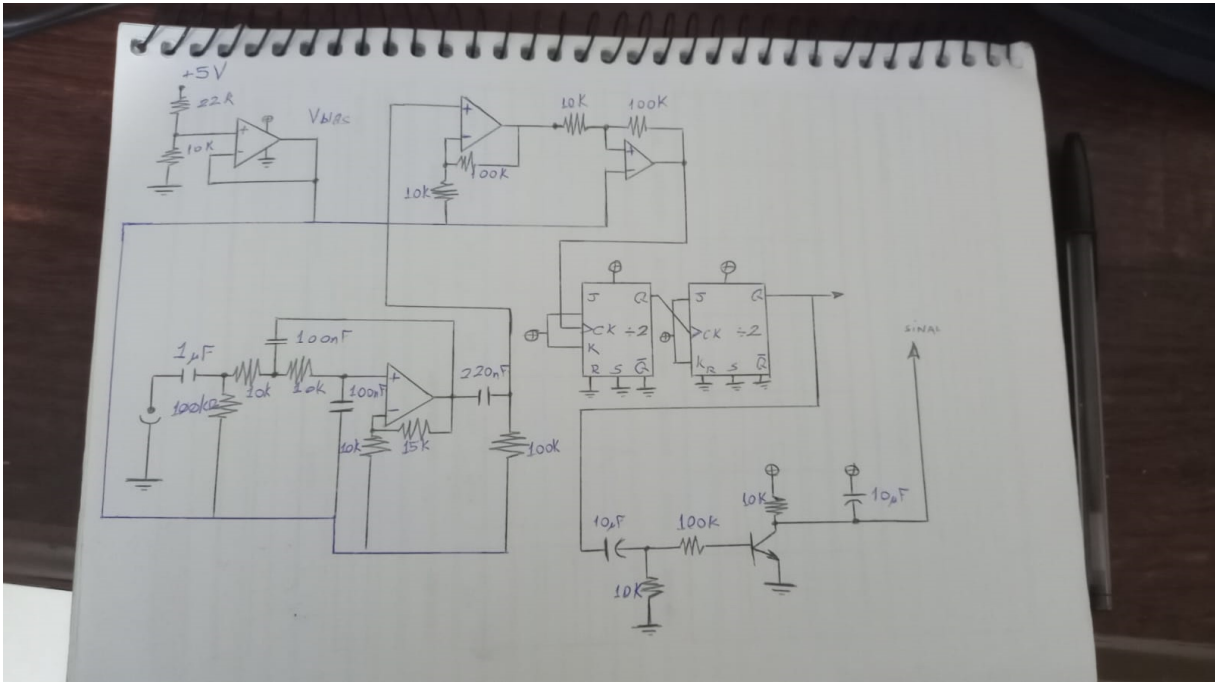
### **1.3 Justificativas**

Escolhemos em fazer o projeto Afinador de violão Ativo Motorizado porque nos garante uma melhor precisão em afinar as cordas do violão, bom para leigos e excelente para quem já tem noção em notas musicais e admiram um bom som.

Este Produto é Prático em ser usado, e não precisa de esforços assim minimizando o tempo, garantindo agilidade e eliminando a precisão de afinação manual demorada, este afinador de violão motorizado garante uma afinação com mais durabilidade, pois se faz necessário essa duração das cordas afinadas, para os músicos que tocam direto ou muitas vezes por horas. Nosso projeto não se limita em afinar somente violão, mais também a outros instrumentos de cordas como: Guitarra, Baixo e Cavaquinho.



## 2 DESENVOLVIMENTO



A referência para comença a part do testes e programação

## **2.1. METODOLOGIA.**

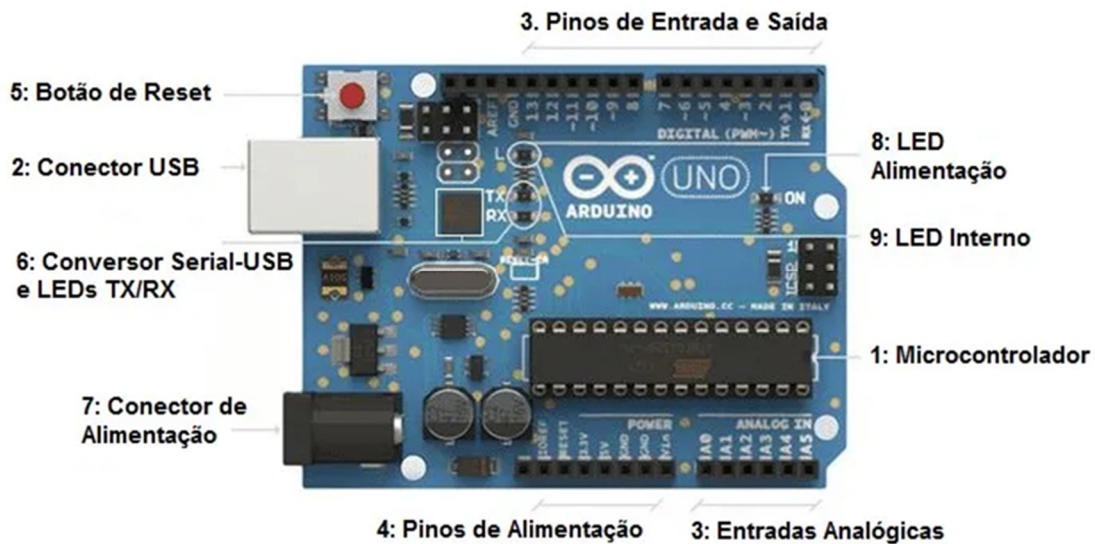
Inicialmente o grupo havia decidido por um projeto muito complexo de uma Máquina de Sorvete (Americana), precisava de um motor e sensores entre de sensor de Pressão e temperatura para fazer a massa do sorvete fica na temperatura ideal, feita a partir do uso de um motor específico para o projeto, e o gás específico para geladeira e freezer. Após muita discussão o grupo notou que havia pouco tempo e dinheiro e nenhuma empresa disponibilizou o esquema elétrico para executar um projeto assim complexo e decidimos mudar para um tema mais simples, que poderia ser feito no tempo restante. Escolhido um afinador de violão ativo motorizado escolhida a nova ideia, foram feitas pesquisas em internet para saber quais seriam os materiais ideais para o projeto.

## 2.2. Materiais importantes do protótipo.

### 2.2.1 Arduino Uno R3.

Arduino Uno é uma plataforma da família arduino das mais conhecidas.

Segundo o site vida de silício o arduino uno é composto pelos seguintes blocos:



Características do arduino uno:

- **Microcontrolador: ATmega328**
- **Tensão de Operação: 5V**
- **Tensão de Entrada: 7-12V**
- **Portas Digitais: 14 (6 podem ser usadas como PWM)**
- **Portas Analógicas: 6**
- **Corrente Pinos I/O: 40mA**
- **Corrente Pinos 3,3V: 50mA**
- **Memória Flash: 32KB (0,5KB usado no bootload)**
- **SRAM: 2KB**
- **EEPROM: 1KB**
- **Velocidade do Clock: 16MHz (2)**

Arduino é uma plataforma que possibilita o desenvolvimento de projetos eletrônicos. Em outras palavras, é uma plataforma de prototipagem eletrônica. O Arduino é constituído de **hardware** e **software**, tornando assim possível a realização de diversos projetos tecnológicos.

### 2.2.2 Motor DC com caixa d Redução



- **Peso unitário: 188g**
- **Corrente: 4.4A Potência :3.7 w**
- **Tensão Nominal**
- **Torqu: 8.5Kgf.cm**
- **Velocidade (sem carga): 13RPM**
- **Velocidade (carga máxima): 11.7RPM**

Os motores de corrente contínua (CC) ou motores (DC – **Direct Current**), como também são chamados, são dispositivos que operam aproveitando as forças de atração e repulsão geradas por eletroímãs e ímãs permanentes.

### 2.2.3 Plug j10

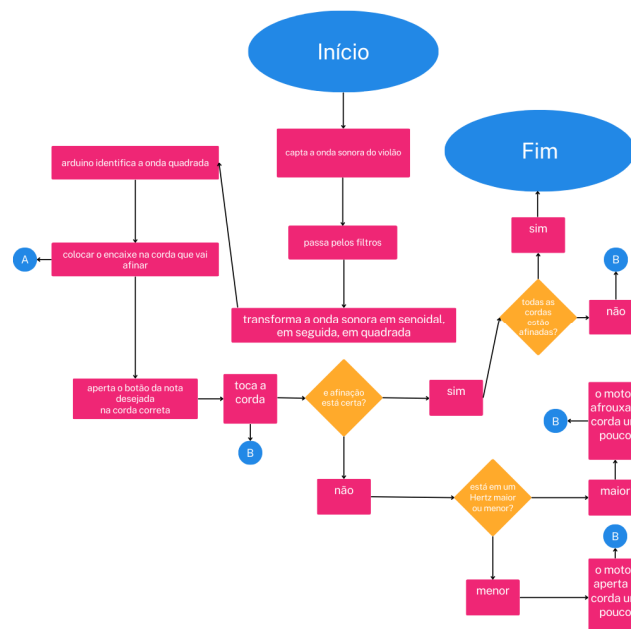


- **Tipo de conector**
- **P10**
- **Gênero do conector**
- **Fêmea**
- **Material de revestimento**
- **Metal**
- **Dispositivos compatíveis**
- **placa, caixa**

Os Conectores P10 podem ser tanto mono, (transmitindo um canal de áudio) quanto estéreo, (transmitindo dois canais de áudio). A diferença em relação aos outros conectores é que seu diâmetro é de 6,35mm e não possuem a função microfone

### 3. Funcionamento do circuito

O circuito que será responsável por lidar com as ondas geradas pelo Violão, todos o circuito foi criado em mente que a alimentação será única e fornecida por uma bateria de 5V de celular/Tablet. Seu funcionamento é mostrado no fluxograma abaixo:



Dos autores

A seguir partes do funcionamento do projeto:

- **VBIAS**

Essa parte do circuito é responsável por descolar o offset de Onda, o VBIAS vai ser interligado em uma grande parte do circuito, a primeira onda indica como ela seria sem o VBIAS com o offset normal negativo e positivo

Já a segunda parte mostra como a onda se comporta com o VBIAS sendo exibida como se fosse inteiramente positiva.

- **Filtro Ativo Passa Baixa de Segunda Ordem**

Tentativa de Eliminar os harmônicos deixando assim passar apenas a frequência fundamental (Primeiro Harmico ) , os resistores de 15K e 10K são responsáveis por amplificar a amplitude em 2,5x

- **Capacitor de Desacoplamento**

Ambos são responsáveis por reforçar o offset criado no VBIAS ambos estão ligados no mesmo (linha azul).

- **Amplificador Inversor**

É responsável por amplificar onda que foi gerada pelo violão e transformada em senoidal pelo 2° FILTRO ATIVO DE PASSA BAIXA.

- **Flip-flop com JK em modo Toggle**

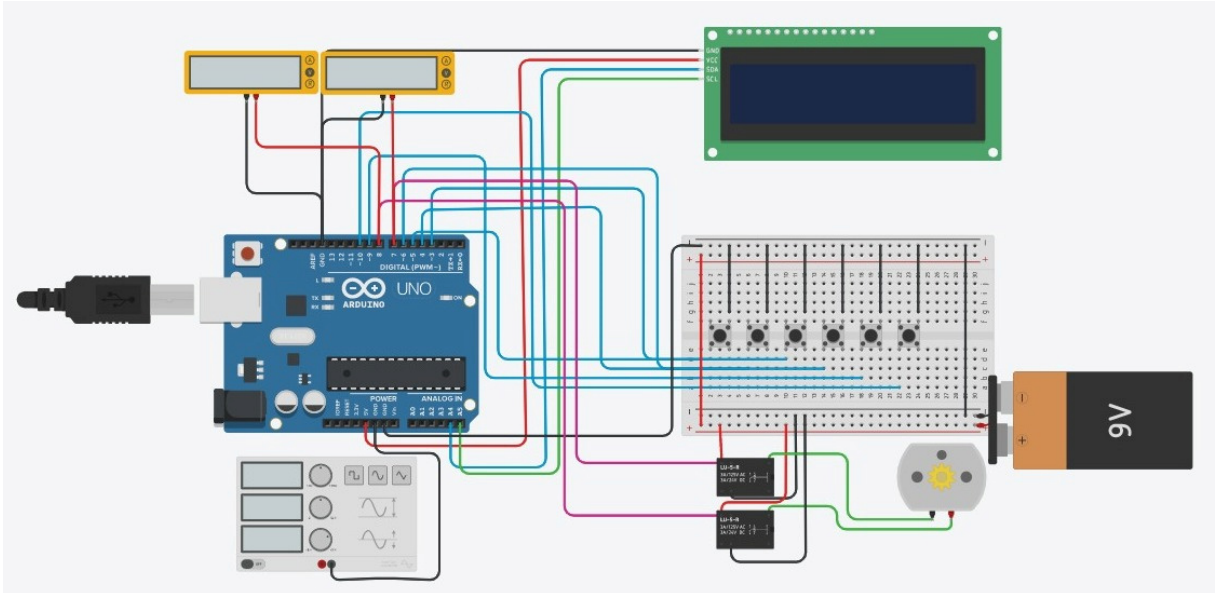
Ele serve como um divisor de frequência que tem  $\frac{1}{2}$  em cada um deles que dividindo por 4 deixando a onda com menos intervalos e assim facilitando o processo do Arduino ler a onda.

- **Identificador de Sinal Barrado**

Primeiro capacitor de 10nf (Diferenciador) pega a variação da onda e cria um sinal que é detectado pelo segundo capacitor que filtra a onda como uma espécie de retificador de onda.

Sinal barrado quando houver onda será 0 e quando não houver será 1, a origem desse ajuste vem na intenção de regular o motor para em caso de erro não continue funcionando mesmo sem onda.

## 4 Programação



### Código:

```
#include <LiquidCrystal_I2C.h>
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x20,16, 2); // Endereço do LCD I2C pode variar, ajuste conforme necessário
```

```
const int relayPin1 = 7; // Pino digital 7 do Arduino
```

```
const int relayPin2 = 8; // Pino digital 8 do Arduino
```

```
//const int pwmPin = 6; // Pino PWM para controle de velocidade do motor (alterado para evitar conflito)
```

```
const int buttonE = 3; // Botão para E MENOR (82 Hz) - G2 (sol na oitava 2)
```

```
const int buttonA = 4; // Botão para A (110 Hz) - A2 (lá na oitava 2)
```



```

const int buttonD = 5; // Botão para D (146 Hz) - D3 (ré na oitava 3)
const int buttonG = 6; // Botão para G (196 Hz) - G3 (sol na oitava 3)
const int buttonB = 9; // Botão para B (247 Hz) - B3 (si na oitava 3)
const int buttonEM = 10; // Botão para E MAIOR (330 Hz) - E4 (mi na oitava 4)

const int frequencyInputPin = 2; // Pino digital 2 do Arduino para medir a frequência (INT0)

volatile unsigned long pulseCount = 0;
unsigned long frequency = 0;
unsigned long desiredFrequency = 82; // Frequência inicial de E MENOR (82 Hz)
unsigned long previousMillis = 0;

int motorSpeed = 255; // Velocidade inicial do motor (0-255)

void setup() {
  lcd.init(); // Inicializa o LCD
  lcd.backlight(); // Liga o backlight do LCD
  lcd.setCursor(0, 0); // Define a posição inicial do cursor no LCD
  lcd.print("Frequencia:"); // Imprime o texto inicial no LCD

  pinMode(relayPin1, OUTPUT);
  pinMode(relayPin2, OUTPUT);
  //pinMode(pwmPin, OUTPUT); // Configura o pino PWM como saída
  pinMode(buttonE, INPUT_PULLUP);
  pinMode(buttonA, INPUT_PULLUP);
  pinMode(buttonD, INPUT_PULLUP);
  pinMode(buttonG, INPUT_PULLUP);
  pinMode(buttonB, INPUT_PULLUP);
  pinMode(buttonEM, INPUT_PULLUP);

```

```

digitalWrite(relayPin1, LOW);
digitalWrite(relayPin2, LOW);

attachInterrupt(digitalPinToInterrupt(frequencyInputPin), countPulse, RISING); // Alterado
para RISING

Serial.begin(9600);
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= 1000) {
    previousMillis = currentMillis;
    noInterrupts();
    frequency = pulseCount;
    pulseCount = 0;
    interrupts();

    Serial.print("Frequency: ");
    Serial.println(frequency);

    updateLCD(); // Atualiza o LCD com a informação da frequência

    if (frequency > desiredFrequency) {
      moveMotorLeft();
    } else if (frequency < desiredFrequency) {
      moveMotorRight();
    } else {

```

```

        stopMotor();
    }
}

// Chamada para funções de debounce
debounceButtons();
}

void setFrequencyE() {
    desiredFrequency = 82;
    Serial.println("Frequência desejada ajustada para E MENOR (G2)");
}

void setFrequencyA() {
    desiredFrequency = 110;
    Serial.println("Frequência desejada ajustada para A (A2)");
}

void setFrequencyD() {
    desiredFrequency = 146;
    Serial.println("Frequência desejada ajustada para D (D3)");
}

void setFrequencyG() {
    desiredFrequency = 196;
    Serial.println("Frequência desejada ajustada para G (G3)");
}

void setFrequencyB() {
    desiredFrequency = 247;

```

```

    Serial.println("Frequência desejada ajustada para B (B3)");
}

void setFrequencyEM() {
    desiredFrequency = 330;
    Serial.println("Frequência desejada ajustada para E MAIOR (E4)");
}

void moveMotorLeft() {
    digitalWrite(relayPin1, HIGH);
    digitalWrite(relayPin2, LOW);
    Serial.println("Motor movendo para a esquerda");
}

void moveMotorRight() {
    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, HIGH);
    Serial.println("Motor movendo para a direita");
}

void stopMotor() {
    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, LOW);
    Serial.println("Motor parado");
}

void countPulse() {
    pulseCount++;
}

```

```

void updateLCD() {
    lcd.setCursor(0, 1); // Posição para escrever no LCD (segunda linha)

    if (frequency == desiredFrequency) {
        lcd.print("Ideal");
    } else if (frequency > desiredFrequency) {
        lcd.print("Maior ");
    } else {
        lcd.print("Menor ");
    }
    lcd.print(" "); // Limpa qualquer texto residual no LCD
}

```

```

void debounceButtons() {
    static unsigned long lastDebounceTime = 0;
    const unsigned long debounceDelay = 50; // 50 ms debounce time

    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (digitalRead(buttonE) == LOW) {
            setFrequencyE();
        }
        if (digitalRead(buttonA) == LOW) {
            setFrequencyA();
        }
        if (digitalRead(buttonD) == LOW) {
            setFrequencyD();
        }
        if (digitalRead(buttonG) == LOW) {
            setFrequencyG();
        }
    }
}

```

```
}  
if (digitalRead(buttonB) == LOW) {  
    setFrequencyB();  
}  
if (digitalRead(buttonEM) == LOW) {  
    setFrequencyEM();  
}  
lastDebounceTime = millis(); // Atualiza o tempo de debounce  
}  
}
```

## 5. Análise de Custo

Item	Quantidade	Preço Unitário	Total :
Arduino	1	51.75	51.75
Motor DC com Caixa de redução	1	14	14
ProtoBoard	3	57.36	172.08
Cabo p10	1	59.9	59.9
Enrolador de corda	1	20	20
Outros (Resistores, Capacitores,etc)		150	150
TOTAL			467.73

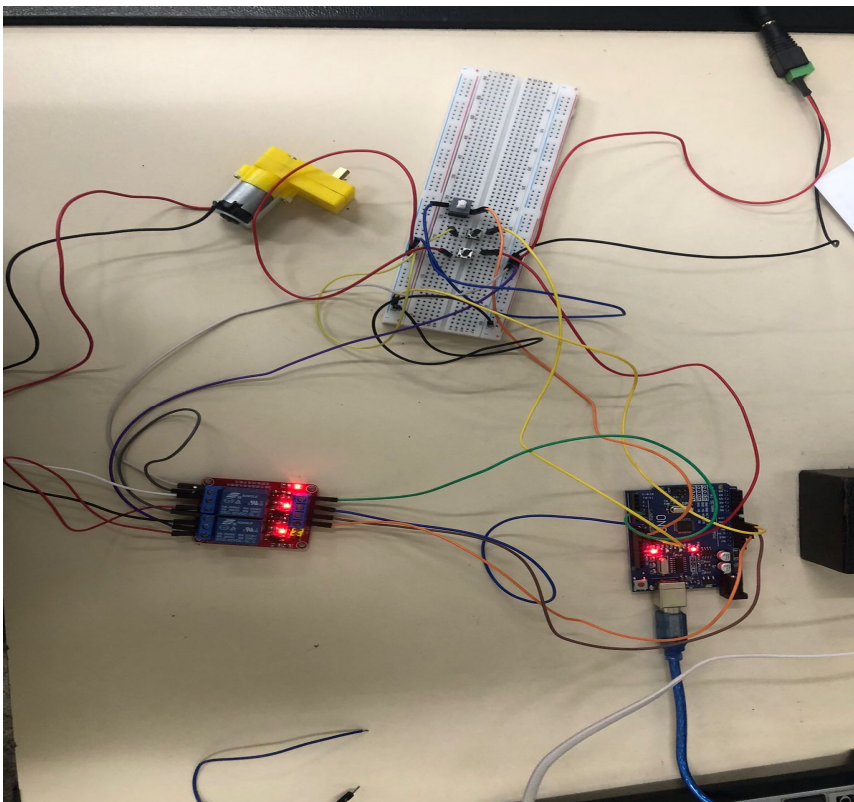
## 6. Testes

No decorrer no projeto ocorreram diversos testes com sensores, amplificadores e placa de fenolite. O que ajudou a melhorar e desenvolver o projeto.

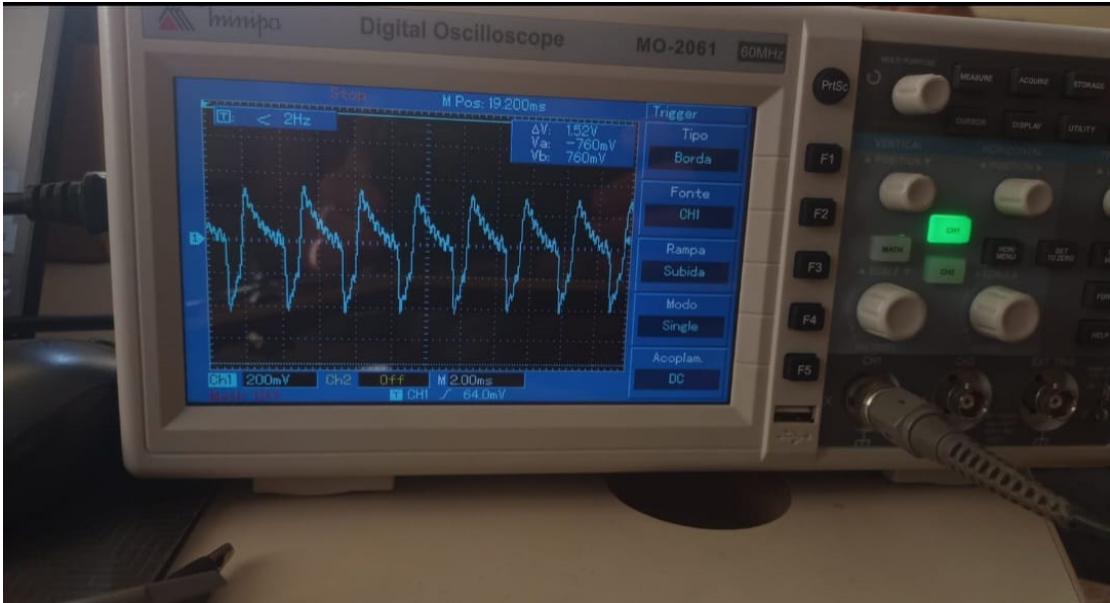
O primeiro teste feito foi com sensores ultrassônicos. Programamos os sensores um para controle de frequência e outro para amplitude.

O segundo teste foi com uma placa de fenolite. Primeiramente usamos um bombril para tirar a superfície de cobre da placa e deixa lá lisa e homogênea, logo após soldamos um jumper macho para conectar no Arduino. A placa está sendo usada para gerar a capacitância do corpo, ela poderia ser usada como um sensor quando aproximar a mão a capacitância varia.

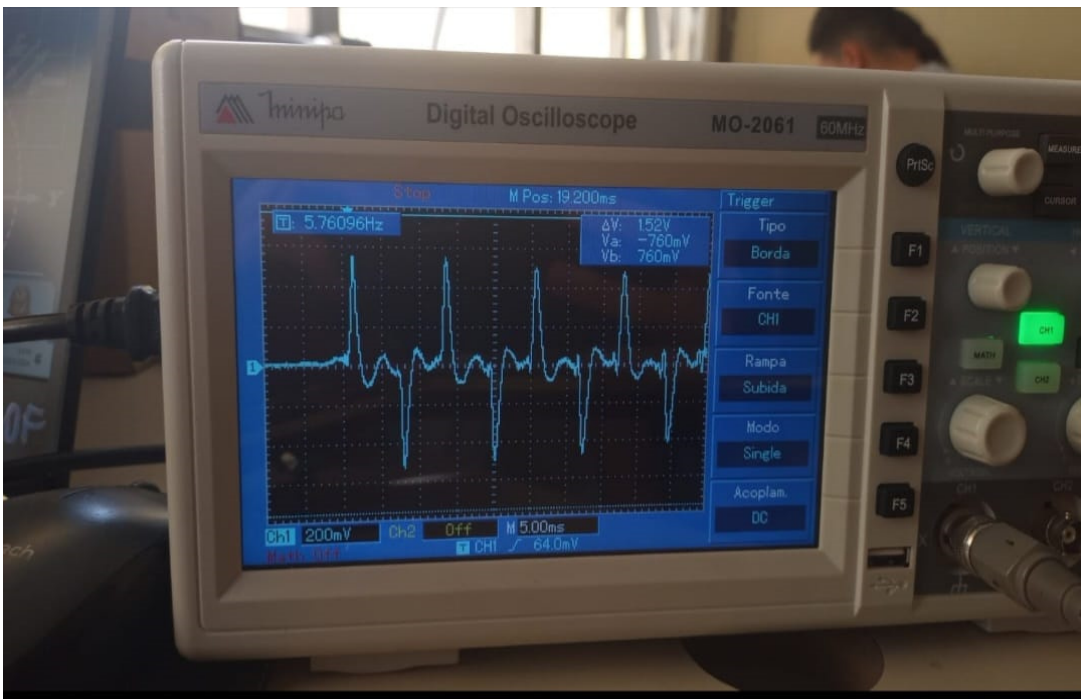
O terceiro teste foi feito com amplificador operacional, o objetivo era deixar o som do theremin mais alto e legível.



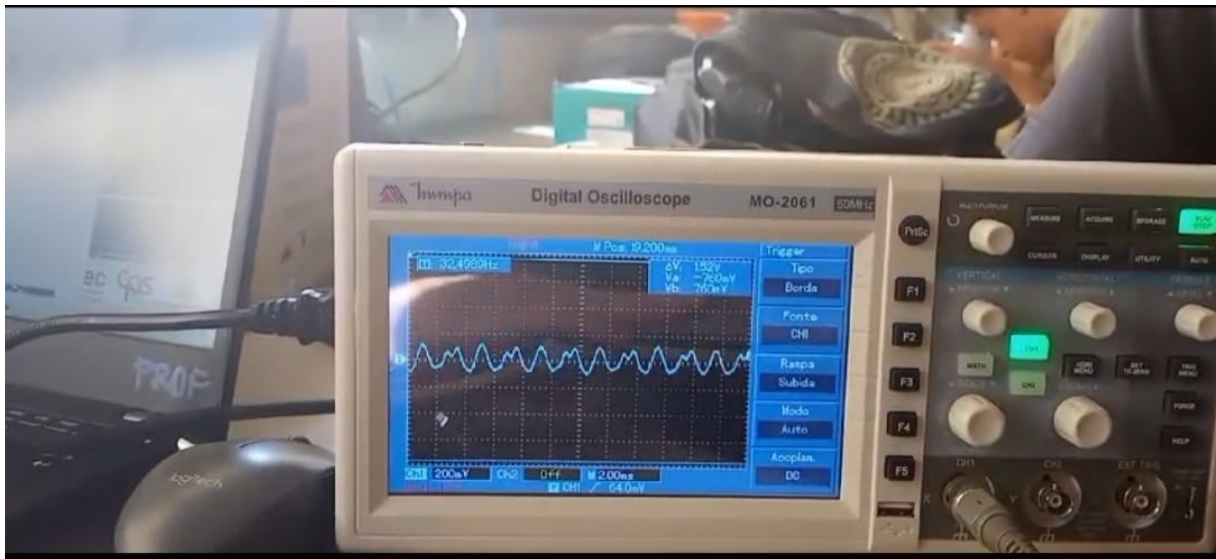




Forma d'onda Mi (aguda)



Forma d'onda Mi (Grav)



**Comportamento da onda em todas as cordas**

## 7 CONCLUSÃO

O filtro ativo sim e a programação, mas, o motor e o rele não. Porque, de princípio, nós íamos usar a ponte H porém de última hora optamos por trocar pelo rele, pelo fato de que a ponte h seria necessário ter que produzir mais um circuito. Não atendeu todas as nossas expectativas, assim, ficando pior do que imaginávamos, mas de certa forma, produzimos e aprendemos demais, principalmente com componentes e teorias que nos foram apresentados no semestre anterior e tivemos muita dificuldade de entender .O aprendizado e montagem da parte analógica do projeto, o funcionamento, e a troca da ponta quase, pelo fato de não ter funcionado a mecânica do motor, não conquistamos uma essa parte do projeto e H pelos relés, que mesmo não funcionando, foi um grande avanço para o trabalho Que faltamos na parte de planejamento e noção da nossa ideia, porque tínhamos tempo suficiente para tal projeto, entretendo, em nossas suposições a parte mecânica não teria tanta influência, acabou que no final, foi o que destinou o motor a não funcionar corretamente, implementação de, na parte mecânica e na programação, uma adaptação para diferentes tipos de afinação e instrumentos de corda.

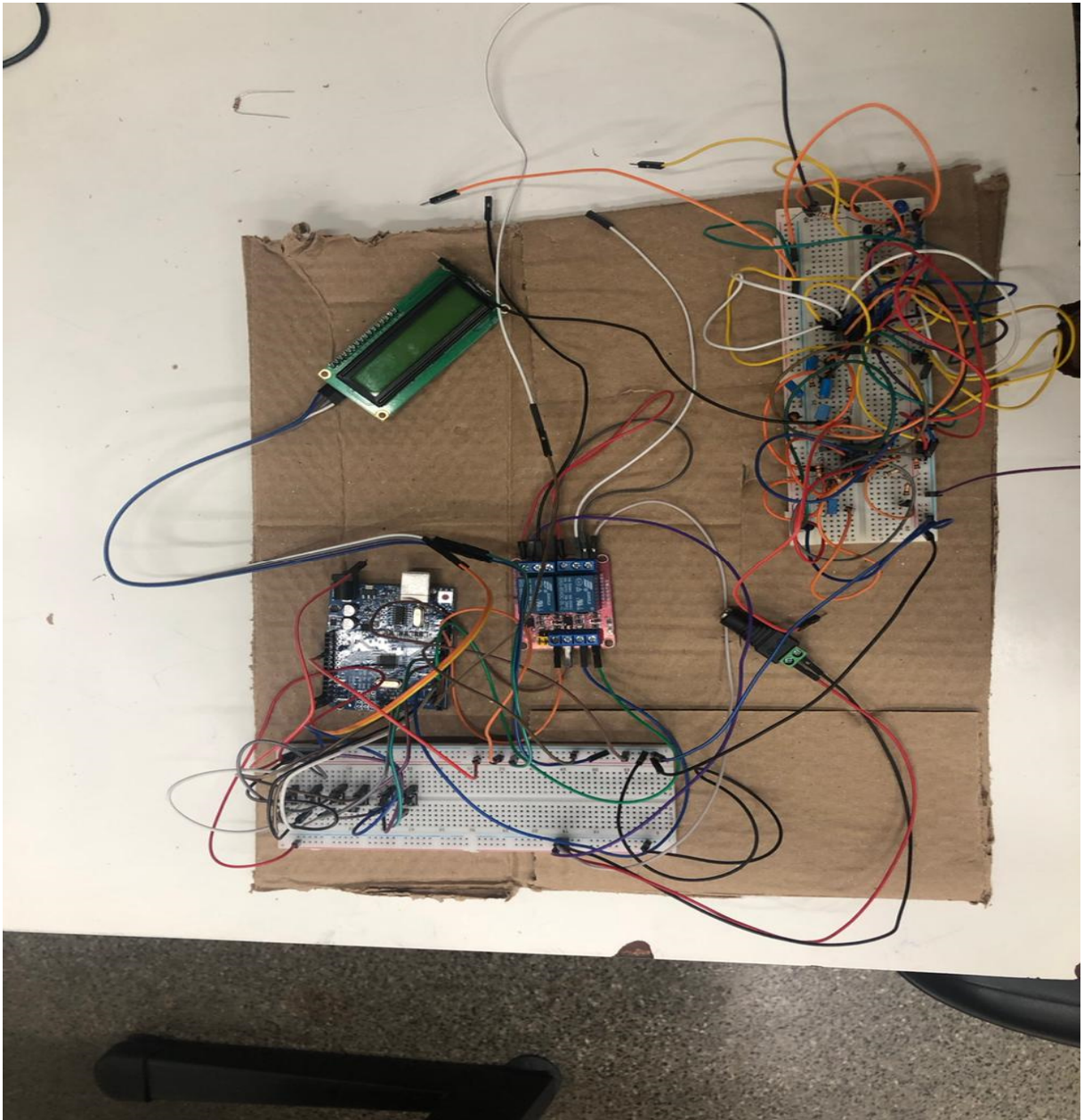
## REFERÊNCIAS

<https://www.amazon.com.br/Elementos-eletr%C3%B4nica-digital-Francisco-Gabriel/dp/8536530383>

. <https://youtube.com/shorts/nxMJGhYYaql?si=5JjJo6pO77X5oT-y>

[https://www.google.com/shopping/product/3443842923361526423?q=++livro+de+Eletr%C3%B4nica+analogica&sca\\_esv=ae0e588ff55da115&sca\\_](https://www.google.com/shopping/product/3443842923361526423?q=++livro+de+Eletr%C3%B4nica+analogica&sca_esv=ae0e588ff55da115&sca_)

## APÊNDICE



**Protótipo Final**

