



# MANUAL



Guia para implementação dos recursos  
educacionais desenvolvidos na pesquisa:

**Modelagem Matemática da Voz, Trigonometria e  
Robótica: Atividades Interativas**

<b>1. Introdução .....</b>	<b>3</b>
<b>2. Descrição do Componentes .....</b>	<b>4</b>
2.1. Drawbot.....	4
2.2. Programa Octave .....	5
2.3. Aplicativo para celular .....	6
<b>3. Construção do DrawBot.....</b>	<b>7</b>
3.1. Visão Geral .....	7
3.2. Lista de materiais .....	9
3.3. Montagem do DrawBot .....	14
3.4. Programação do Arduino .....	37
<b>4. Instalação e Execução do Octave .....</b>	<b>43</b>
4.1. Configuração da Placa Bluetooth.....	43
4.2. Instalação do Octave .....	49
4.3. Criação dos arquivos de execução no Octave .....	52
4.4. Execução e teste dos arquivos no Octave .....	56
<b>5. Instalação do aplicativo de controle do DrawBot no celular .....</b>	<b>65</b>
5.1. Visão geral do aplicativo .....	65
5.2. Etapas para a instalação do aplicativo.....	66
<b>6. Descrição das atividades .....</b>	<b>69</b>
6.1. Proposta de atividade .....	69
 <b>ARQUIVO A - Código Octave SINNSTART.m.....</b>	<b>78</b>
<b>ARQUIVO B - Gabarito 1.....</b>	<b>81</b>
<b>ARQUIVO C - Gabarito 2.....</b>	<b>82</b>
<b>ARQUIVO D - Código para gravação da placa Arduino .....</b>	<b>83</b>
<b>ARQUIVO E - Código Octave SETUP.m .....</b>	<b>86</b>
<b>ARQUIVO F - Código Octave SERIAL.m .....</b>	<b>86</b>
<b>ARQUIVO G - Código Octave AJUSTECECURVA_CALCULO.m.....</b>	<b>87</b>
<b>ARQUIVO H - Código Octave AJUSTECECURVA_PLOT.m.....</b>	<b>89</b>
<b>ARQUIVO I - Código Octave DRAWBOT.m .....</b>	<b>91</b>
<b>ARQUIVO J - Código Octave GRAFICO3D.m.....</b>	<b>94</b>
<b>ARQUIVO K - Código Octave GRAFICOF4.m .....</b>	<b>94</b>

## 1. Introdução

O presente material foi desenvolvido como parte da dissertação de Mestrado Profissional em Ensino de Ciências e Matemática, da Universidade Federal de Uberlândia, com o título *MODELAGEM MATEMÁTICA DA VOZ, TRIGONOMETRIA E ROBÓTICA: ATIVIDADES INTERATIVAS* e trata-se de um manual e tutorial com objetivo de auxiliar os professores de matemática para que possam desenvolver atividades com a utilização de Tecnologias Digitais da Informação e Comunicação (TDIC) com tema modelagem da voz e robótica. Tem como finalidade integrar a realidade, modelagem matemática do som de uma vogal e um robô, à matemática com o intuito de dar sentido aos conceitos de trigonometria, em especial às funções seno e cosseno.

Para a realização da pesquisa descrita na dissertação foi desenvolvido um programa computacional no Octave e o DrawBot (robô que desenha) para implantação das atividades em sala de aula e responder à questão norteadora do trabalho: *É possível o desenvolvimento de atividades a partir da união entre modelagem matemática e robótica de forma a contextualizar e aplicar conceitos de funções trigonométricas em sala de aula com a participação ativa dos alunos?*. Em vista disso, foi feita uma atividade interativa onde os alunos participam de forma interativa e o som de suas vozes são utilizadas para demonstrar que o som de uma vogal pode ser expresso por uma soma de senos e os parâmetros dessas funções podem ser demonstrados a partir da comparação da voz masculina e feminina. A integração com a robótica ocorre no momento em que o sinal de voz é desenhado pelo DrawBot. Além disso, foi desenvolvida outra atividade interativa com o dispositivo robótico Drawbot, de forma a explorar diferentes possibilidades didáticas para a função trigonométrica com base no desenho de formas geométricas. Todos os materiais e programas computacionais utilizados nas atividades foram projetados e construídos a partir de softwares gratuitos e de componentes acessíveis de baixo custo de forma que sua replicação seja possível pelos professores.

Introdução<sup>1</sup>- [https://youtu.be/ybWXXs\\_gy6s](https://youtu.be/ybWXXs_gy6s)

 Portanto, o manual tem a finalidade de descrever os dois principais produtos: o modelo matemático do som de uma vogal no Octave e o funcionamento do DrawBot. Além disso será demonstrado como implementar os mesmos e também a proposta de aplicação em sala de aula.

A estrutura do presente material está dividida da seguinte maneira: descrição dos produtos, construção do DrawBot, Instalação e uso do Octave, instalação e uso do aplicativo no celular e proposta de atividade que utiliza e relaciona os itens anteriores.

A seguir, serão dadas as instruções para a implementação dos recursos desenvolvidos e

---

<sup>1</sup> Link para vídeo explicativo.

também sua aplicação.

## 2. Descrição dos componentes

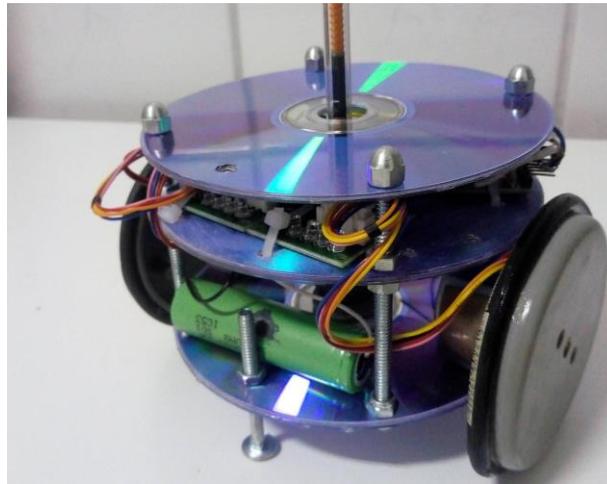
### 2.1. DrawBot

O DrawBot (Figura 1) é um carrinho robótico que é programado para desenhar e seu nome é a junção das palavras em inglês “Draw” e “Robot” que significam desenhar e robô respectivamente. O seu controle é feito por meio do Arduino<sup>2</sup> que é uma plataforma integrada para programação de diversos tipos de placas eletrônicas (Figura 2), ou seja, permite que por meio do computador e um cabo de dados USB, programar o mesmo para executar uma certa tarefa, como por exemplo, piscar um led ou controlar a velocidade de um pequeno motor<sup>3</sup>. Para programar as placas é utilizada a linguagem de programação C que possui boa curva de aprendizado e vasto material de apoio na internet. Além disso, possuem vários recursos para aplicação em projetos de automação residencial e industrial, internet das coisas entre outros. Na perspectiva da sala de aula, educadores estão percebendo o potencial da plataforma no ensino devido à possibilidade de abordar diversos conceitos de forma interdisciplinar e também pela fácil utilização e o seu baixo custo. Uma observação a ser feita é que o Drawbot utilizado na pesquisa em sala de aula é diferente do apresentado no manual. O Drawbot mostrado na figura 1 é uma versão que utiliza ferramentas simples para, como tesoura, alicates e ferro de solda. O utilizado em sala de aula seria necessário o uso de furadeira e também outros recursos que as pessoas talvez não tivessem disponibilidade. Portanto, o intuito é facilitar ao máximo a construção do mesmo.

---

<sup>2</sup> Arduino IDE. Disponível em: <https://www.arduino.cc/en/main/software>.

<sup>3</sup> Atividades com Arduino. Disponível em: [https://www.emc.ufg.br/~jwilson/pdf/Atividades\\_com\\_o\\_Arduino.pdf](https://www.emc.ufg.br/~jwilson/pdf/Atividades_com_o_Arduino.pdf).



*Figura 1: DrawBot.*

#### Ambiente de Programação - Arduino



#### Diversas placas que podem ser programadas



*Figura 2: Esquema do ambiente Arduino de programação de placas.*

## 2.2. Programa Octave

Octave<sup>4</sup> é uma linguagem computacional, desenvolvida para computação matemática. Possui uma interface em linha de comando para a solução de problemas numéricos, lineares e não-lineares, também é usada em experimentos numéricos. Possui compatibilidade com MATLAB<sup>5</sup> e tem um grande número de recursos e funções semelhantes que facilita na aprendizagem da linguagem de

<sup>4</sup> Octave. Disponível em: <https://www.gnu.org/software/octave/>

<sup>5</sup> Matlab. Disponível em: <https://www.mathworks.com/products/matlab.html>

programação do Octave devido a grande quantidade de conteúdo disponível na internet. Além disso o programa foi escolhido por ser gratuito e possuir diversas ferramentas para a implementação do projeto desenvolvido na pesquisa.

A programação que permite relacionar o sinal de voz, ou som da vogal, com a função trigonométrica seno é feita no Octave. Esse processo ocorre por meio da captura o som de uma vogal a partir da fala do aluno e retorna à função, obtida pelo método dos mínimos, que melhor se aproxima dos dados derivados da amostragem do som (Figura 3). Em seguida o Octave envia os dados, através de comunicação sem fio Bluetooth, ao DrawBot e este desenha a função obtida anteriormente.

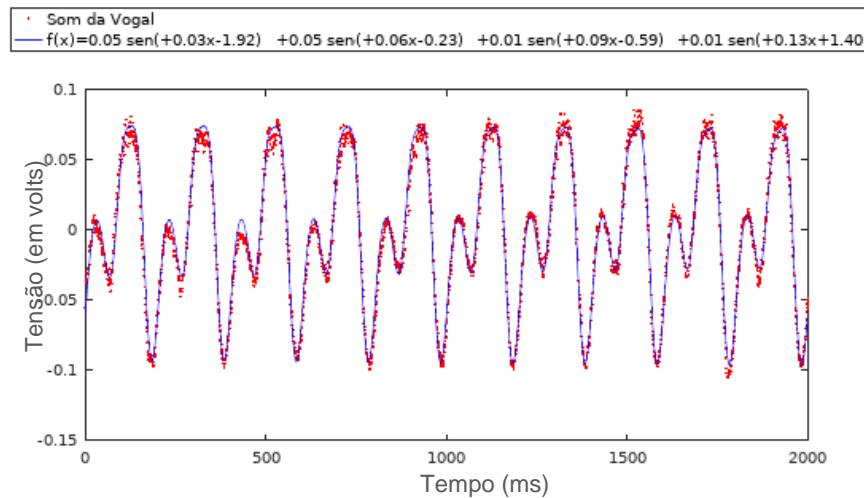
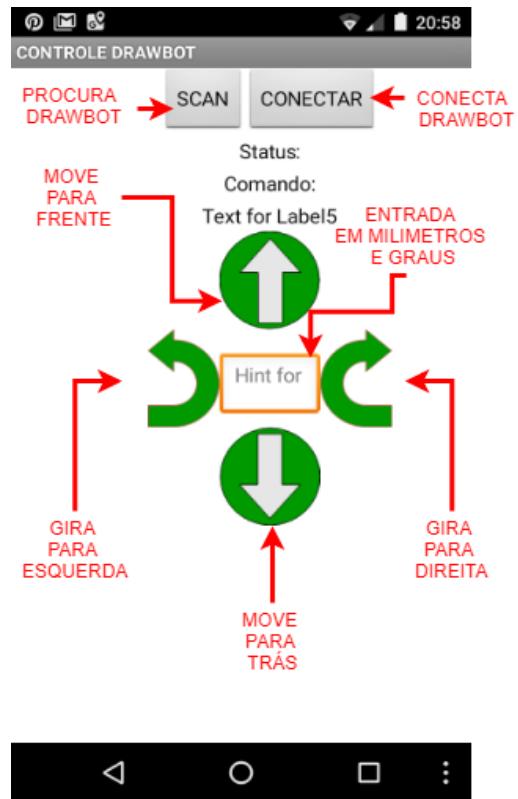


Figura 3: Ajuste de curva no Octave do som da vogal “i”.

### 2.3. Aplicativo de celular

O aplicativo (Figura 4) tem a função de controlar o DrawBot foi desenvolvido para realização de atividades que leva ao uso de funções trigonométricas. No final deste material serão disponibilizadas as atividades desenvolvidas em sala de aula com o uso do aplicativo e o DrawBot.



*Figura 4: Visão geral do aplicativo*

### 3. Construção do DrawBot

#### 3.1. Visão geral do DrawBot

O DrawBot é um robô de duas rodas programado para receber comandos enviados a partir do celular ou computador e que resulta em movimentos para frente e para trás com resolução de 1 milímetro e giro para esquerda e direita com resolução de 1 grau. Assim, se for enviado o comando ou sequência de caracteres “pf10” o carrinho move para frente 10 mm e já o comando “ge45” faz um giro no próprio eixo de 45 graus para a esquerda, ou seja, sentido anti-horário. Todo este processo acontece por meio de comunicação sem fio Bluetooth e o protocolo é o padrão Rs-232. Este permite que dois dispositivos se comuniquem, ou seja, troquem dados um com o outro de forma simples e confiável. Para o controle do DrawBot foi escolhido a placa Arduino Nano devido as suas dimensões reduzidas, recursos necessários exigidos no projeto e por ser umas das placas mais utilizadas na plataforma Arduino. Esta tem a função de controlar os motores do carrinho e receber e decodificar os dados enviados pelo computador e o celular.

Já a estrutura física do DrawBot foi projetada para ser construída com peças e materiais de fácil

disponibilidade e com o uso de ferramentas simples. Sua estrutura é composta de DVDs e as rodas são feitas com uma peça chamada roseta que pode ser encontrada em lojas de materiais elétricos e anel de vedação, para tubo de PVC, vendido em casa de materiais de construção. A parte eletroeletrônica é composta por 2 motores, 1 bateria, 1 carregador da bateria, 1 módulo Bluetooth, e o Arduino Nano (Figura 4, 5 e 6).

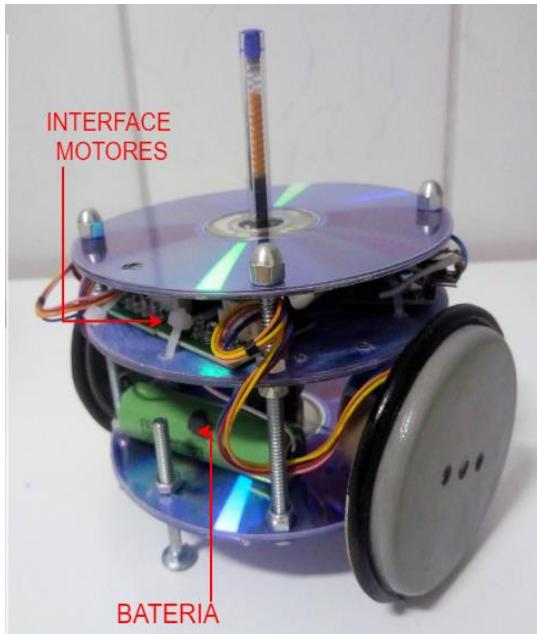


Figura 4: Frente DrawBot.

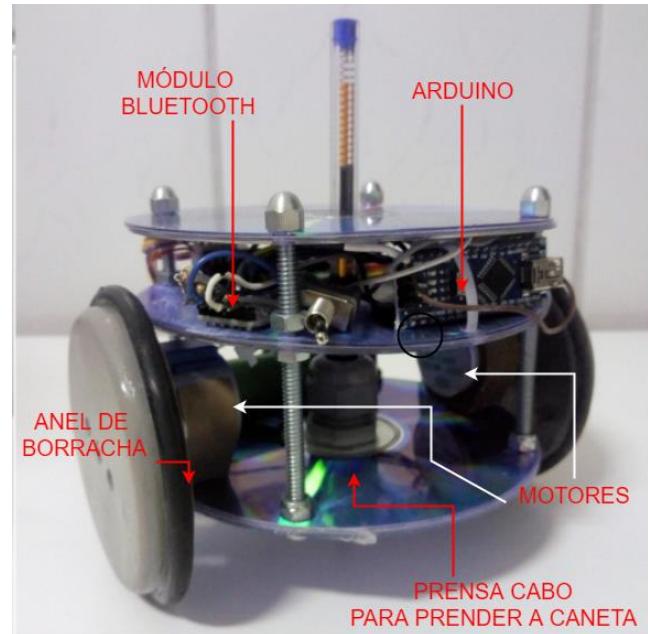


Figura 5: Parte de trás do DrawBot.

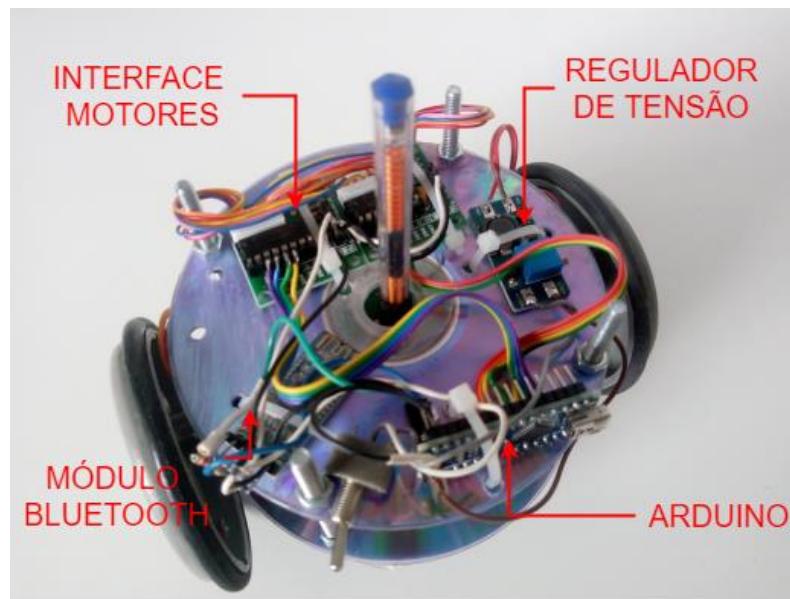
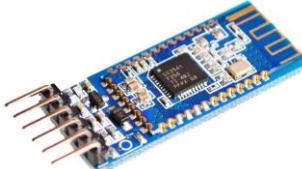


Figura 6: Parte interna.

Na próxima subseção, apresentamos as listas com os itens necessários para a montagem do DrawBot.

### 3.2. Lista de materiais

Antes do início da montagem do DrawBot é necessário ter todos os materiais e ferramentas em mãos. A lista de materiais é descrita em 4 tabelas onde nas colunas é fornecido o nome, função, imagem e quantidade de cada item para a construção do DrawBot. A Tabela 1 descreve os itens eletrônicos, a Tabela 2 os itens relativos a estrutura do chassi, a Tabela 3 os itens da caneta e por fim, a Tabela 4 as ferramentas.

- LISTA DE MATERIAIS PARA CONSTRUÇÃO DO DRAWBOT - ELETRÔNICA			
NOME	FUNÇÃO	IMAGEM	QUANTIDA DE
Arduino Nano	Controlar os motores e decodifica os dados enviados pelo computador e o celular.		1
Módulo Bluetooth AT-09	Fazer a comunicação sem fio entre o computador e a placa Arduino Nano.		2
Módulo Conversor USB 2.0 RS232 UART 5 Pinos CP2102	O módulo faz a comunicação via USB com o módulo Bluetooth AT-09.		1
Motor-de-passo 28byj-48 5v com driver uln2003	O motor é ligado ao Arduino Nano e tem a função de movimentar o DrawBot.		2

TP4056 - Micro USB - do Módulo para carregar Bateria de Lítio	Carregar a bateria de Lítio com segurança.		1
Regulador de tensão Mt3608	Elevar a tensão de 3,6 volts Para 5 volts		1
Bateria de lítio 18650	Fornecer energia ao DrawBot.		1
Carregador de Celular 5v	Carregar a bateria.		1
Fios	Interconectar os diversos componentes eletrônicos.		50

Tabela 1: Itens eletrônicos.

-LISTA DE MATERIAIS PARA CONSTRUÇÃO DO DRAWBOT - CHASSI			
NOME	FUNÇÃO	IMAGEM	QUANTIDA DE
Abraçadeira Para Tubo Soldável 25mm Marrom	Prender o motor ao chassi.		2
Anel de Borracha para cano de PVC 75mm	Dar aderência as rodas.		2
Parafuso 3,5x76,2mm	Unir os 3 níveis do chassis.		4
Prensa cabo 3/8 de polegada	Segurar a caneta.		1
Parafuso Francês 3,5x55mm	Apoiar a parte frontal do DrawBot.		1
DVDs	Estrutura do chassi.		6

Tabela 2: Itens do chassi.

-LISTA DE MATERIAIS PARA CONSTRUÇÃO DO DRAWBOT - CANETA			
NOME	FUNÇÃO	IMAGEM	QUANTIDADE
Fita Isolante	Montagem da caneta		1
Caneta Esferográfica	Desenhar		1
Mola de caneta retrátil	Pressionar a caneta sobre o papel.		1

Tabela 3: Itens da caneta.

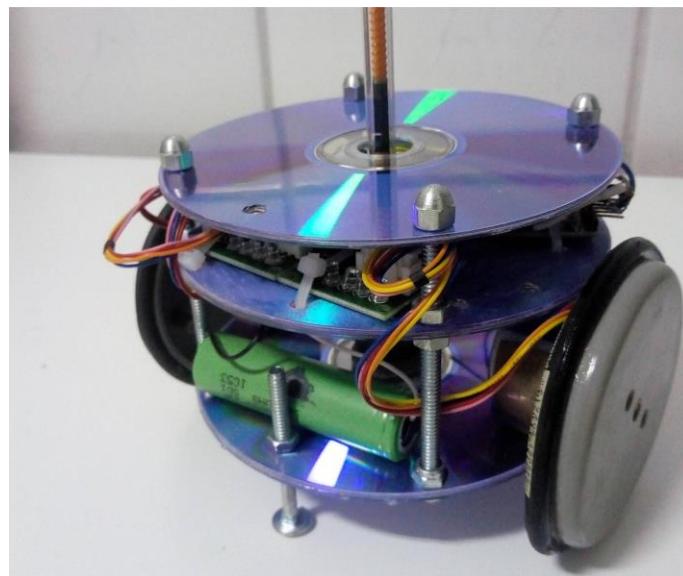
-LISTA DE MATERIAIS PARA CONSTRUÇÃO DO DRAWBOT - FERRAMENTAS E OUTROS			
NOME	FUNÇÃO	IMAGEM	QUANTIDAD E
Ferro de solda	Soldar componente e furar os DVDs		1
Chave de fenda	Parafusar.		1
Alicate	Segurar as porcas para aparafusar.		1
Estilete	Cortar rebarbas		1
Tesoura	Recortar os gabarito.		1
Gabarito	Localização dos furos nos DVDs.		3
Supercola	Colar os DVDs		1
Multímetro	Medições diversas		1

Tabela 4: Ferramentas e outros.

### 3.3. Montagem do DrawBot

Nesta etapa será descrito as etapas de montagem do DrawBot (Figura 7). Este tem como característica possuir somente duas rodas que permite ao mesmo girar em torno do próprio eixo. Além disso no centro do carrinho está presente uma caneta para desenhar enquanto o DrawBot se movimenta a partir do envio de comandos do computador ou celular.

Sua estrutura é composta por materiais simples encontrados facilmente no comércio e que tem como intuito a possibilidade de replicação por quem tiver interesse. Seu chassi é constituído de DVDs, parafusos, abraçadeira de tubo de PVC entre outros.



*Figura 7: DrawBot montado.*

## Montagem do primeiro nível



[Montagem dos DVDs](https://youtu.be/nno3yFbWS_8) - [https://youtu.be/nno3yFbWS\\_8](https://youtu.be/nno3yFbWS_8)

1. A estrutura do DrawBot é composta por 3 níveis (Figura 8), onde o primeiro é para o suporte das rodas, o segundo para a parte eletrônica e o terceiro para proteção. Para cada nível deverá ser colado 2 DVDs como mostrado na Figura 9 e Figura 10.

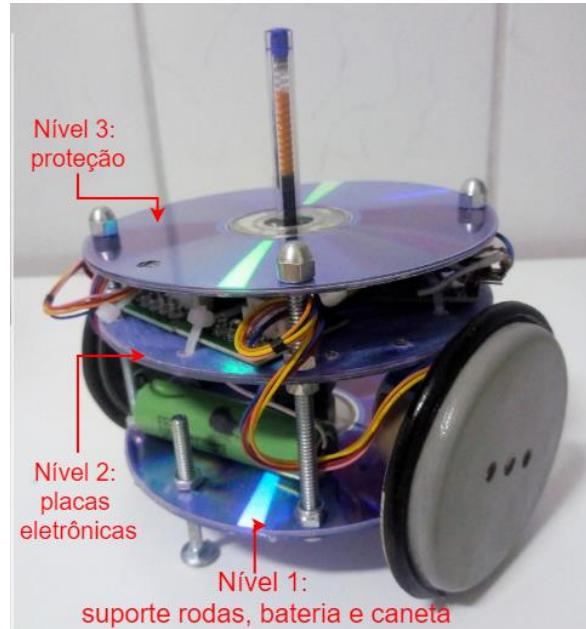


Figura 8: Níveis do DrawBot.



Figura 9: Super cola para união dos DVDs.



Figura 10: União dos DVDs.

2. A perfuração dos DVDs para colocação dos parafusos e montagem dos 3 níveis é feita com uso de gabaritos. Deve-se imprimir duas cópias do gabarito 1, em ARQUIVO B -Gabarito 1 e recortar como

indicado na Figura 11.

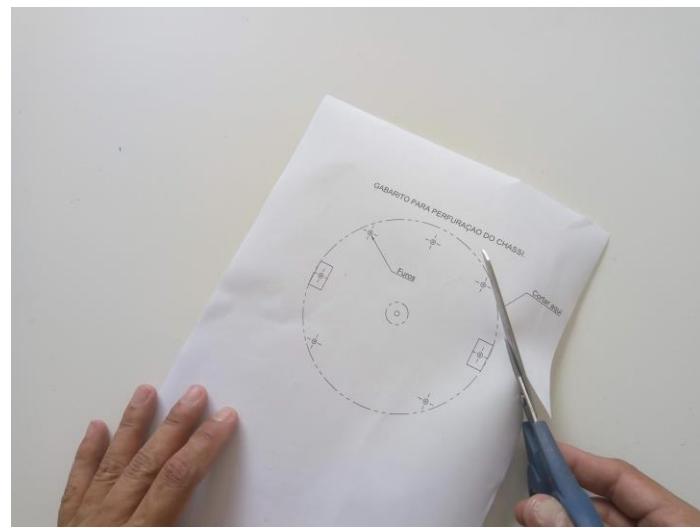


Figura 11: Recorte do gabarito 1.

3. Imprimir 1 cópia do gabarito 2, em ARQUIVO C – Gabarito 2 e recortar como indicado na Figura 12.

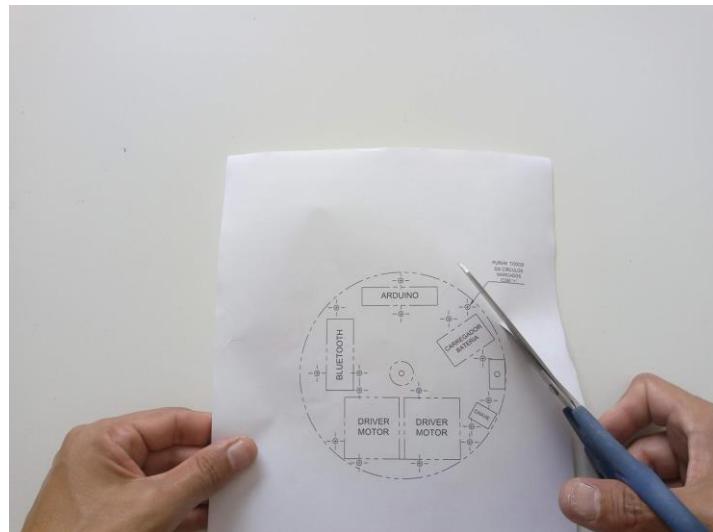


Figura 12: Recorte do gabarito 2.

4. Colar os gabaritos 1 e 2 nos DVDs como indicado na Figura13, Figura 14 e Figura 15.



Figura 13: Cola sendo aplicada no DVD.

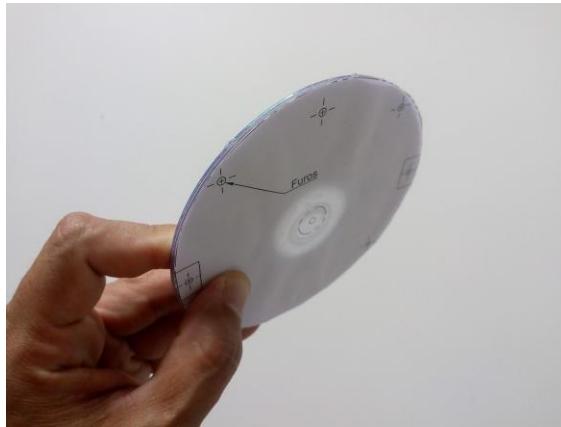


Figura 14: Gabaritos colados no DVD.



Figura 15: Gabaritos colados no DVD.

5. Para colocar os parafusos e demais componentes, furar com o ferro de solda os círculos com o símbolo “+” como mostrado nas figuras 16, 17 e 18.

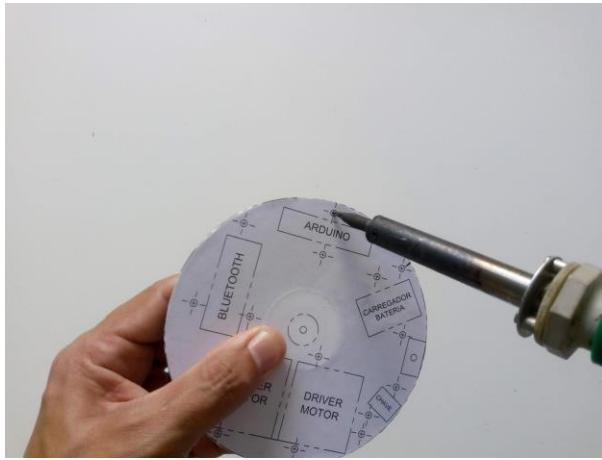


Figura 16: DVDs sendo perfurados com o ferro de solda.

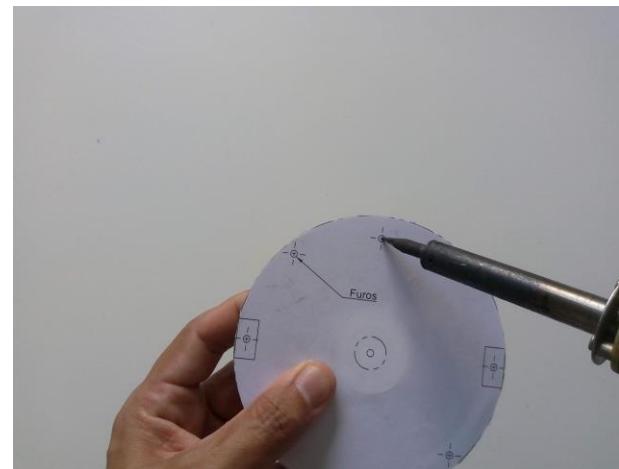


Figura 17: DVDs sendo perfurados com o ferro de solda.



Figura 18: Furos finalizados DVD.

6. Após a furação, tirar as rebarbas dos furos com o estilete (Figura 19).



*Figura 19: Furos finalizados DVD.*



[Montagem do primeiro nível](https://youtu.be/y0uKaMvW31k) - <https://youtu.be/y0uKaMvW31k>

7. Em seguida instalar o suporte dos motores (Figura 20, Figura 21 e Figura 22).



*Figura 20: Instalação do suporte dos motores.*



*Figura 21: Instalação do suporte do motor.*



*Figura 22: Suporte dos motores instalados.*

8. Com o ferro de solda alargar o buraco do centro para encaixe do suporte da caneta (Figura 23).

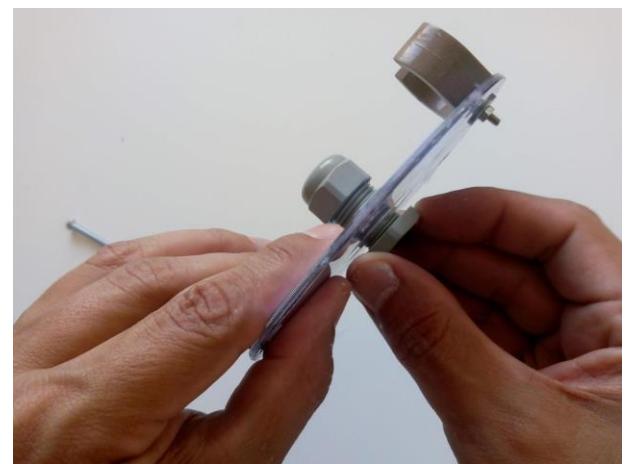


*Figura 23: Alargamento do buraco para o encaixe do suporte da caneta.*

9. Instalar o suporte da caneta ao chassis (Figura 24, Figura 25 e Figura 26).



*Figura 24: Encaixando o suporte da caneta.*

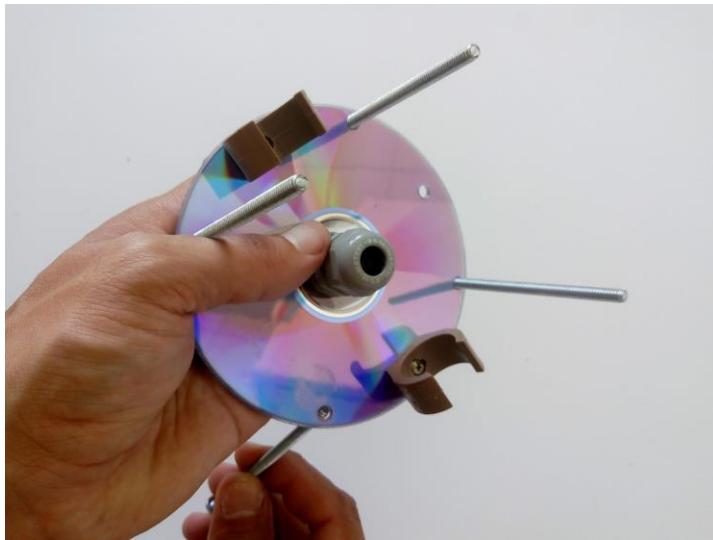


*Figura 25: Apertando a porca.*

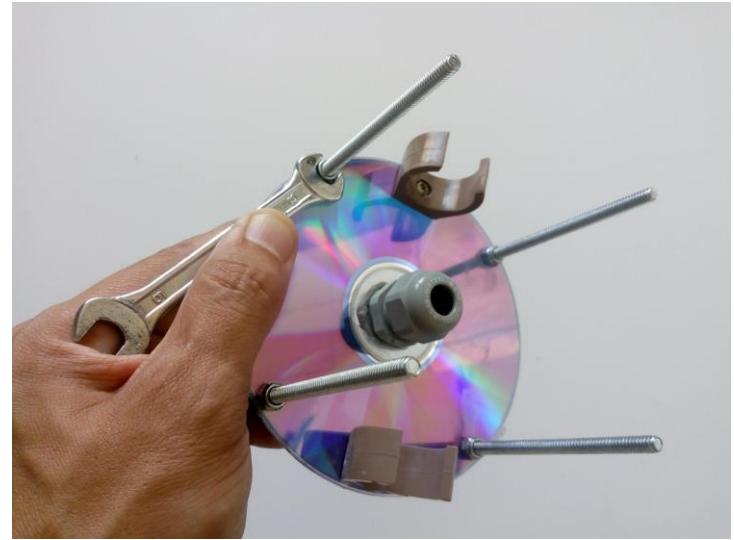


*Figura 26: Suporte instalado.*

10. Após a colocar o suporte da caneta, inserir os parafusos (Figura 27) e apertar as porcas (Figura 28). Com isso conclui-se o primeiro nível do chassi.



*Figura 27: Inserção dos parafusos no DVD.*



*Figura 28: Porcas colocadas e apertadas no DVD.*

## Montagem da parte eletrônica



[Interligação das placas eletrônicas](https://youtu.be/vikELCCTh-w) - <https://youtu.be/vikELCCTh-w>

Antes de iniciar a montagem do segundo nível do chassi, será montada a parte eletrônica e em seguida colocado no chassi.

Como visto anteriormente, controle do DrawBot é feito pelo Arduino e a comunicação sem fio é através do módulo Bluetooth. Já alimentação de energia é feita por uma bateria recarregável de 3,6 volts que faz todo o sistema funcionar. O esquema eletrônico a seguir demonstra como os módulos são interligados (Figura 29).

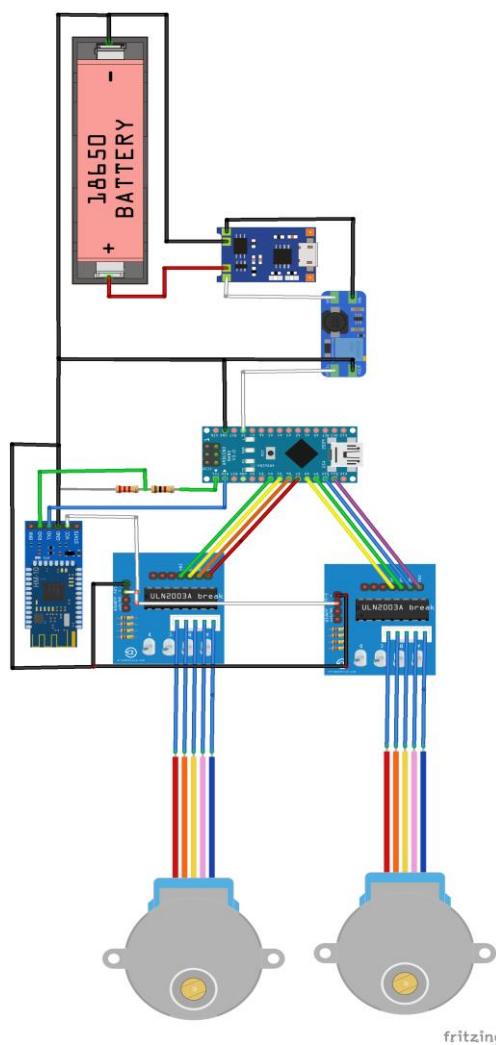


Figura 29: Esquema eletrônico.

A sequência de montagem segue a numeração crescente do número 1 ao 8 como demonstrado na Figura 30.

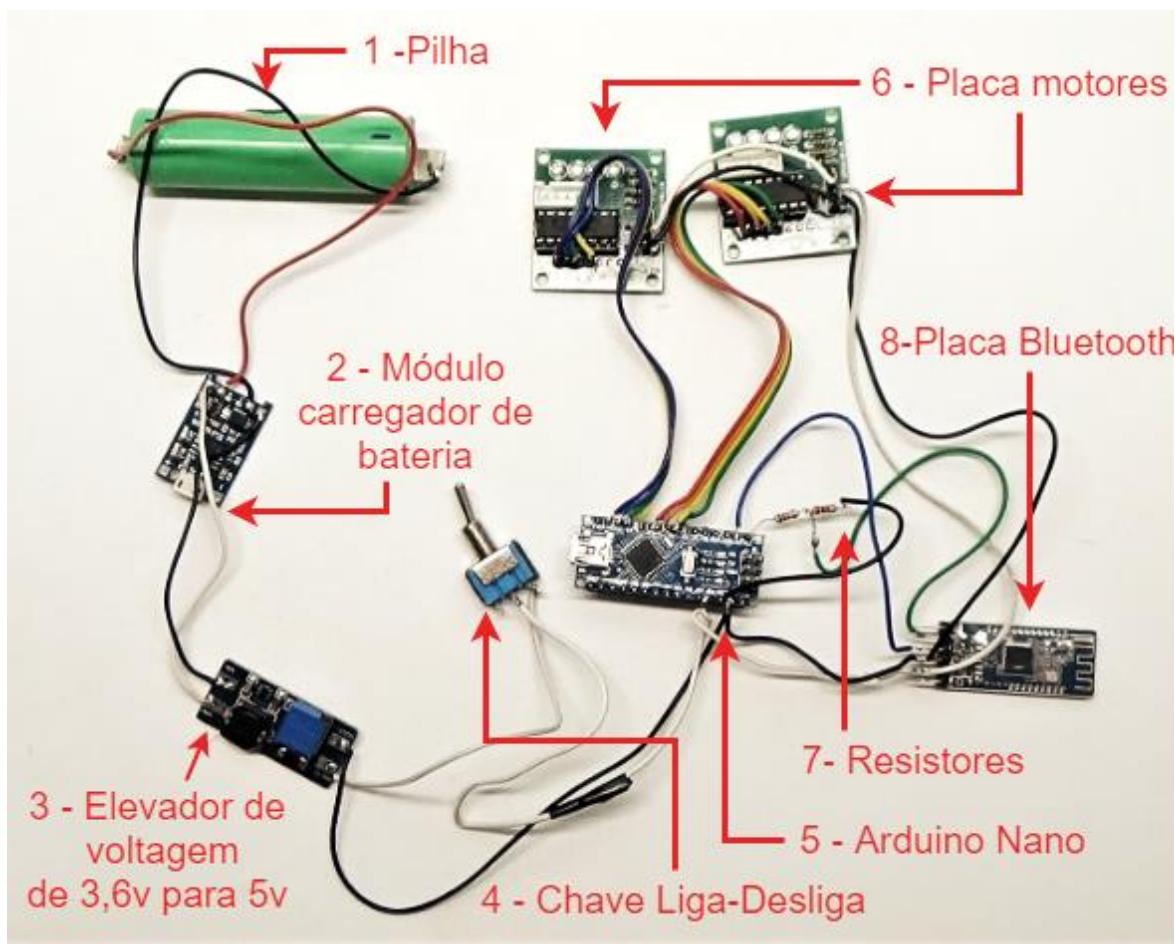


Figura 30: Visão geral do circuito interligado.

1. Soldar a bateria de lítio de 3,6 volts a placa de carregamento, itens 1 e 2 da Figura 30, por meio de dois fios de aproximadamente 15 cm como indicados na Figura 31.

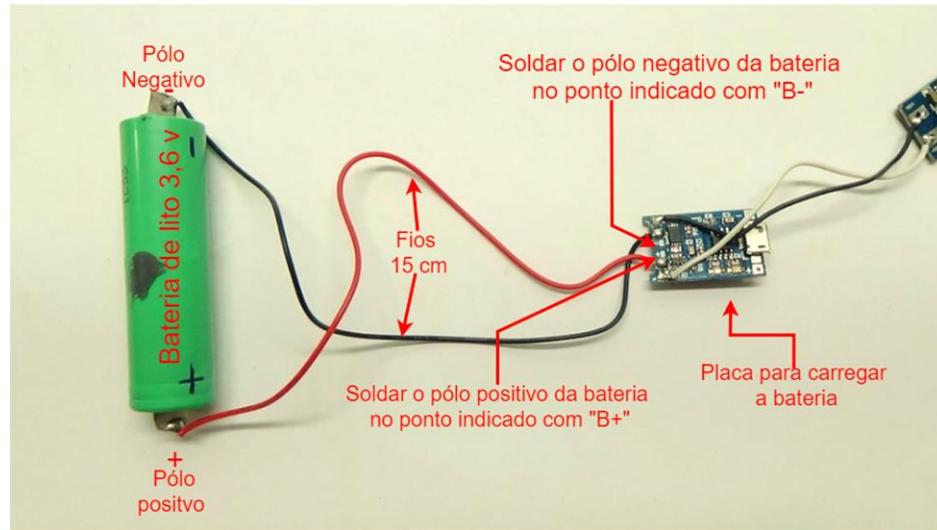


Figura 31: Ligação da bateria ao carregador.

2. Ligar o carregador da bateria ao regulador de tensão, itens 2 e 3 da Figura 30. O regulador de tensão tem a função de aumentar a voltagem de 3,6 volts para 5 volts, pois, tanto o Arduino e os motores funcionam em 5 volts (Figura 32).

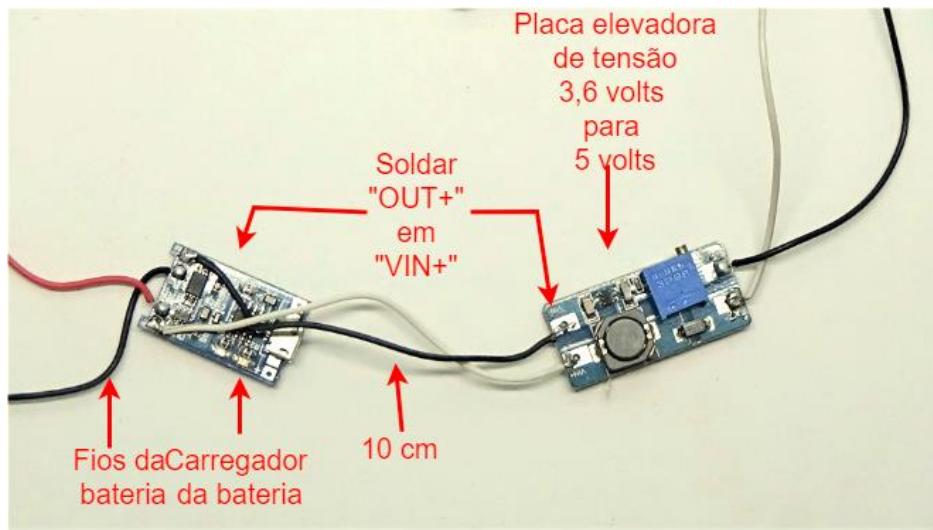


Figura 32: Ligação do carregador ao elevador de tensão.

3. Ligar o elevador de tensão à chave de liga-desliga, itens 3 e 4 da Figura 30. A chave tem a função de ligar e desligar o DrawBot (Figura 33).

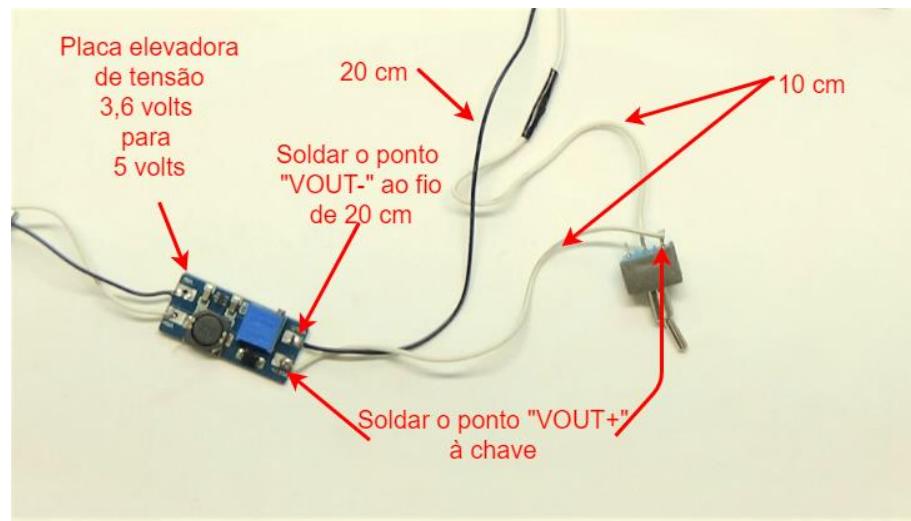


Figura 33: Ligação do regulador de tensão a chave de liga e desliga.

4. Ligar a chave de liga-desliga ao Arduino, itens 4 e 5 da Figura 30. O Arduino exulta todas as funções de controle DrawBot (Figura 34).

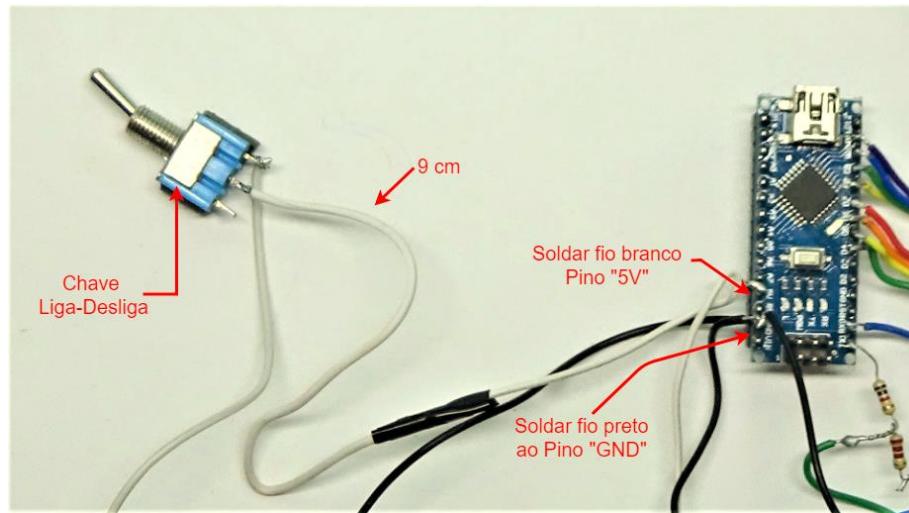


Figura 34: Ligação da chave ao Arduino.

5. Soldar os fios do Arduino para a placa do motor, item 5 da figura 30. Os fios levam os sinais de controle para os motores (Figura 35).

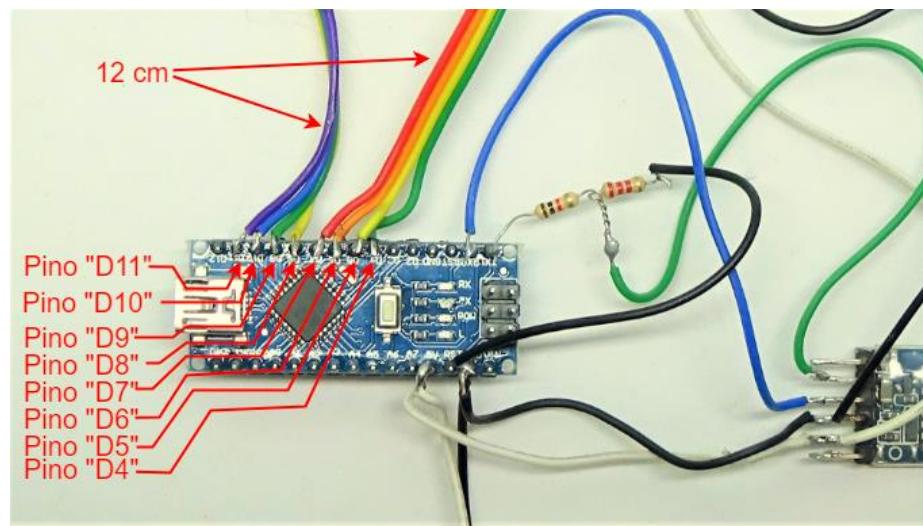


Figura 35: Ligação dos fios do Arduino para os motores.

6. Soldar os fios da placa do motor e de energia, item 6 da figura 30. Os fios branco e preto levam energia e os coloridos os sinais de controle aos motores (Figura 36).

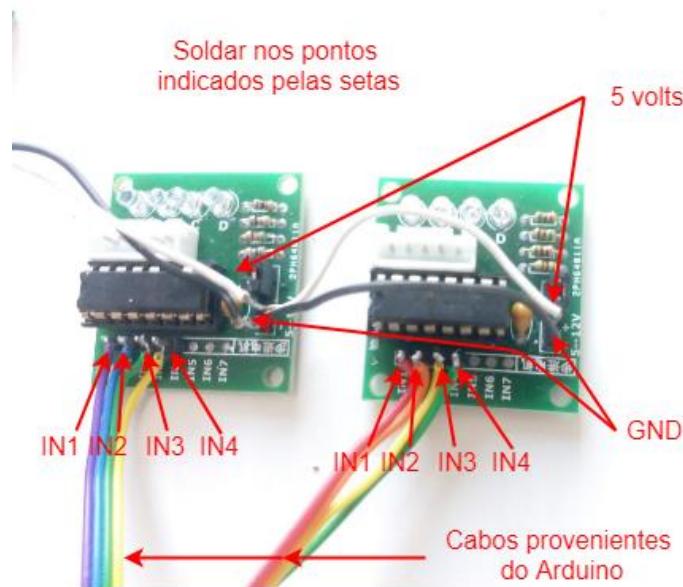


Figura 36: Ligação dos fios aos motores

7. Soldar os resistores ao pino “TX” do Arduino, item 7 da Figura 30. Os dois resistores tem a função de reduzir a voltagem do pino “TX” do Arduino (Figura 37).

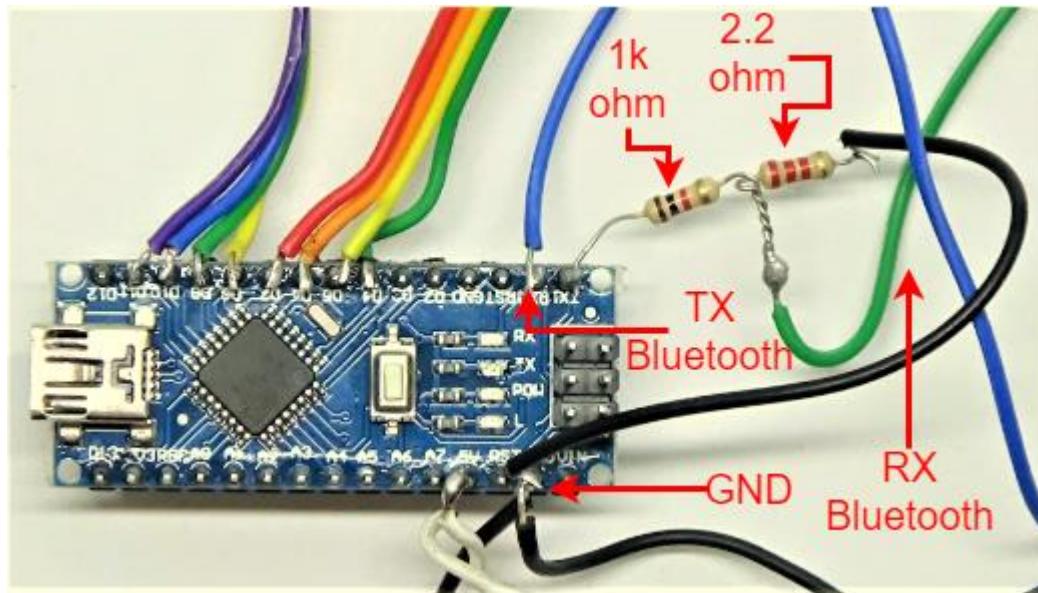


Figura 37: Ligação dos fios do módulo Bluetooth.

8. Soldar os fios do Arduino a placa Bluetooth, itens 5,7 e 8 da Figura 30. O módulo Bluetooth recebe os dados por meio de comunicação sem fio e os envia ao Arduino (Figura 38).

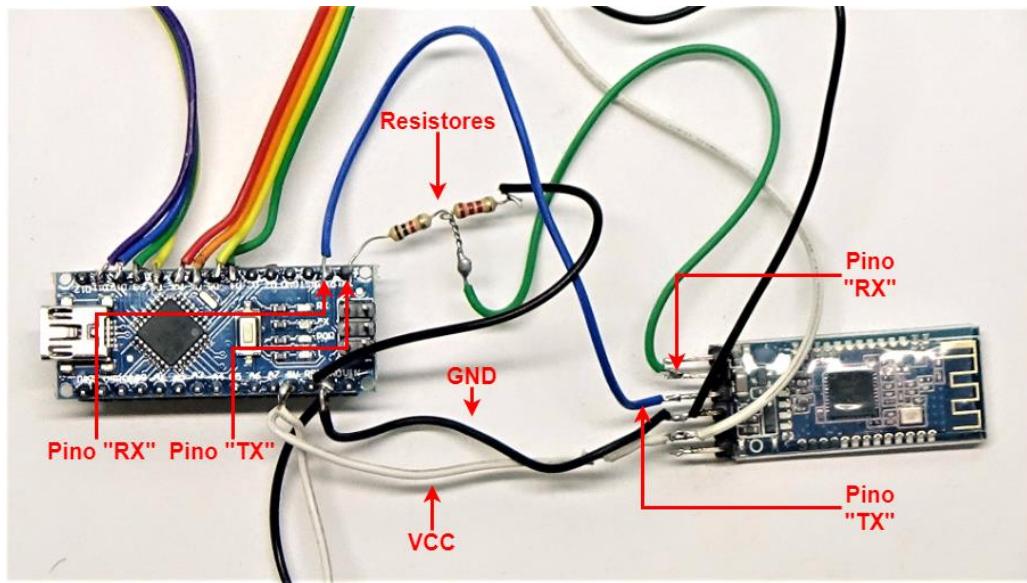
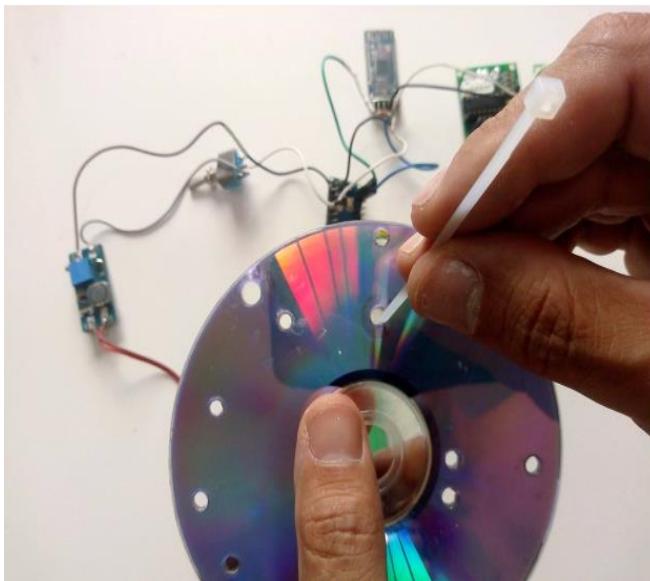


Figura 38: Ligação dos fios do módulo Bluetooth.

- **Montagem da parte eletrônica no chassi (DVD)**

 [Montagem das placas eletrônicas no chassi \(DVD\) -](https://youtu.be/j0Jt0NevWnk) <https://youtu.be/j0Jt0NevWnk>

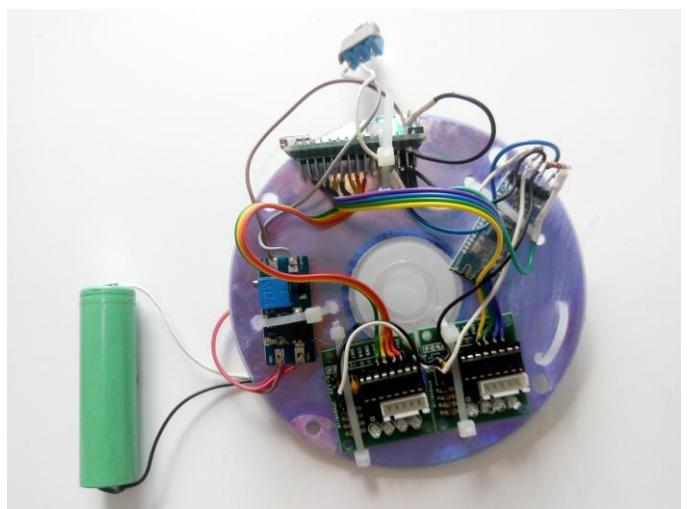
1. Montagem da parte eletrônica no chassi serão utilizadas abraçadeiras de nylon como mostrado nas imagens a seguir (Figuras 39, 40 e 41):



*Figura 39: Passagem das abraçadeiras através dos furos.*



*Figura 40: Abraçadeiras passadas.*



*Figura 41: Componentes presos pelas abraçadeiras.*

## ▪ Montagem da roda



Nesta etapa será demonstrado como montar a roda que vai ao motor.

1. Alargar o furo da roda com uma chave de 5 milímetros de diâmetro (Figura 41).

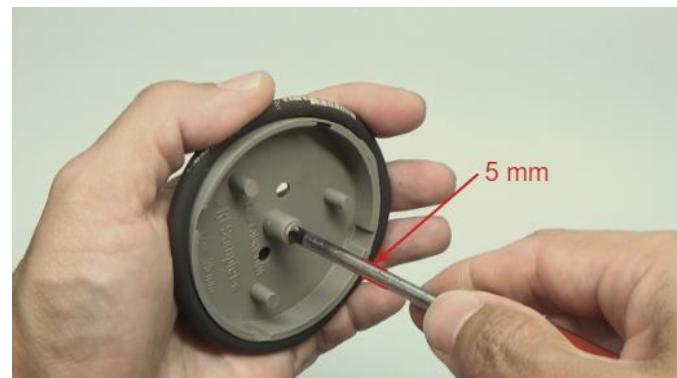


Figura 41: Anel de vedação e a roseta.

2. Colocar o anel de borracha na roda (Figura 42 e Figura 43) que é denominada roseta e em seguida colada com supercola (Figura 44).



Figura 42: Anel de vedação e a roseta.

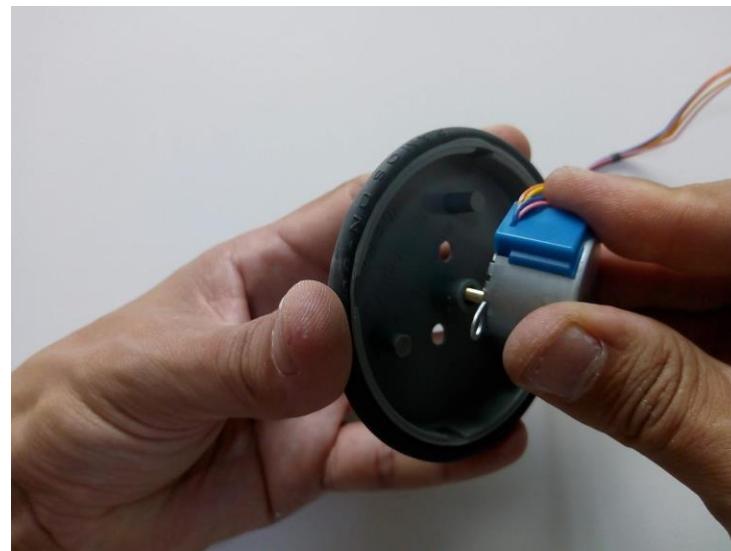


Figura 43: Colocação do anel de borracha na roseta.



*Figura 44: Passar cola na junção entre a roda e o anel.*

3. Agora é acoplado a roda ao motor de passo como demonstrado na Figura 45.



*Figura 45: Colocação da roda do motor.*

## ▪ Montagem do segundo e terceiro nível



[Montagem do segundo e terceiro nível](https://youtu.be/5WIG-NLgwD4) - <https://youtu.be/5WIG-NLgwD4>

1. Com o motor acoplado a roda, basta encaixar o motor no chassi como mostrado na Figuras 46 e 47.

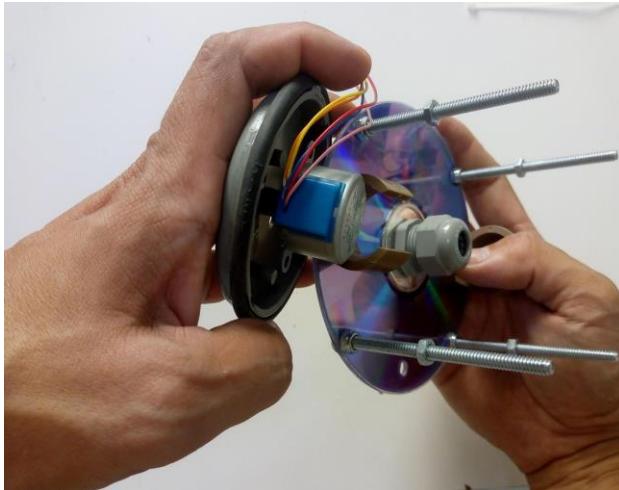


Figura 46: Colocação do motor no chassi.

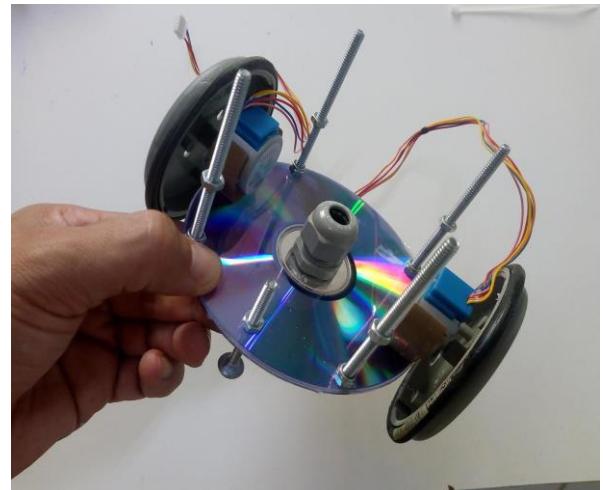


Figura 47: Motores colocados.

2. Após a montagem dos motores no nível 1 do chassi é colocado as placas eletrônicas montadas anteriormente (figura 48). Para isso primeiro é colocado quatro porcas nos parafusos (Figura 49). Em seguida encaixa-se a parte eletrônica montada anteriormente no primeiro chassi (Figura 50).

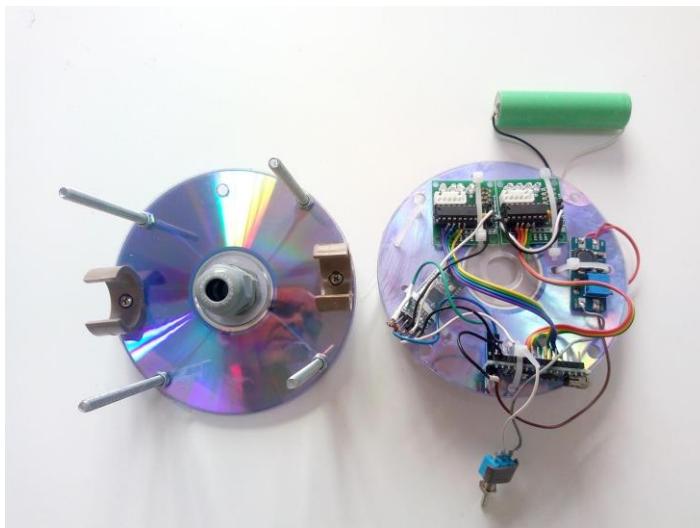


Figura 48: Suporte dos motores e placas eletrônicas montadas.



Figura 49: Colocação das porcas.

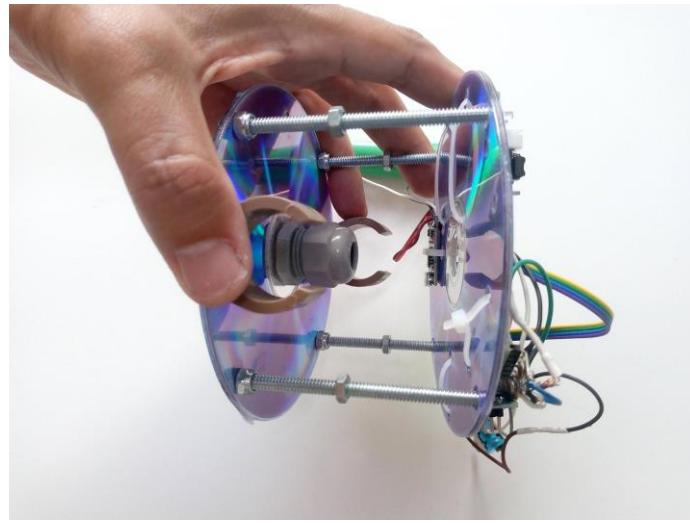


Figura 50: Encaixe da parte eletrônica com o suporte dos motores.

3. Após montar o nível 2 são conectados os motores à parte eletrônica (Figura 51 e 52) e colocadas as porcas para o fechamento do nível 2 (Figura 53).

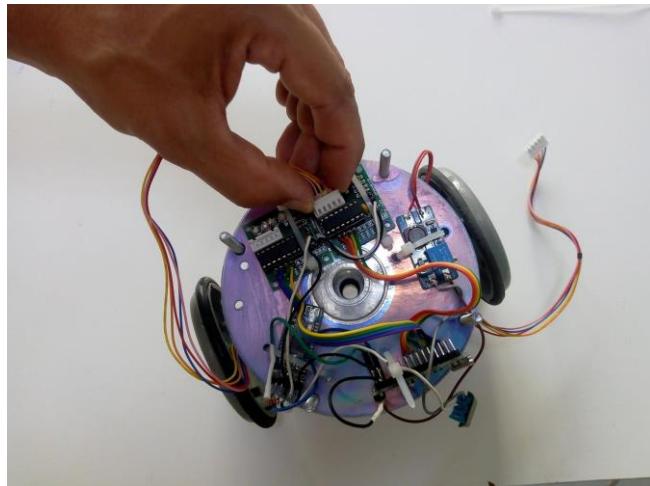


Figura 51: Encaixe do cabo do motor esquerdo na placa.

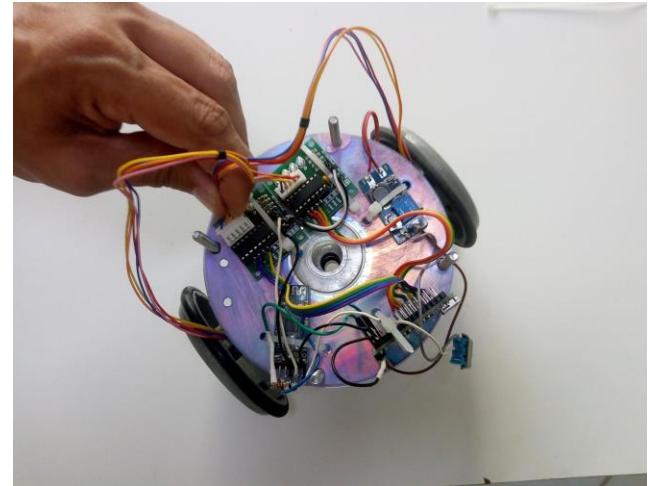
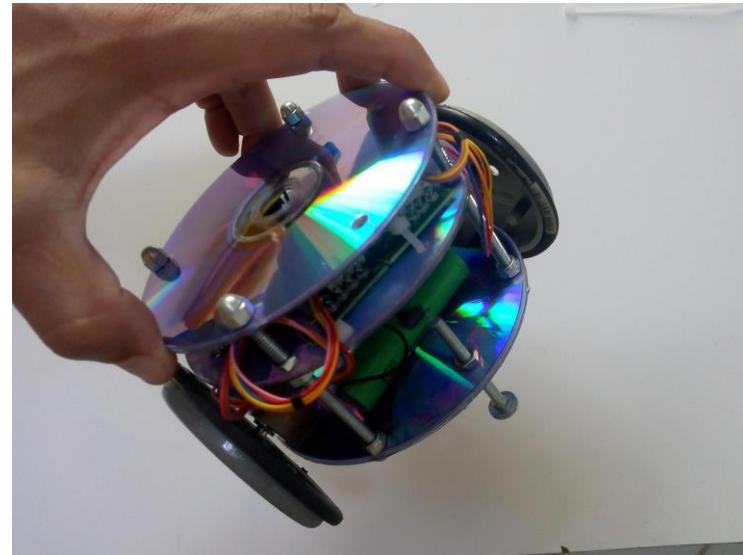


Figura 52: Encaixe do cabo do motor direito na placa.



*Figura 53: Motores conectados.*

4. Para finalizar é colocado o nível 3 para a proteção da parte eletrônica (Figura 54).



*Figura 54: Tampa de proteção colocada.*

## ▪ Montagem da caneta



1. Para montagem da caneta será necessária uma caneta esferográfica, uma mola de caneta retrátil e fita isolante (Figura 55). A mola tem a função de criar uma pressão suave sobre o papel para o desenho ocorra de forma constante e sem interferir no movimento do carrinho.



Figura 55: Materiais necessários para construção da caneta

2. A função da fita isolante é de melhorar o encaixe no suporte da caneta. Colocar fita isolante nos pontos indicados (Figura 56 e 57).

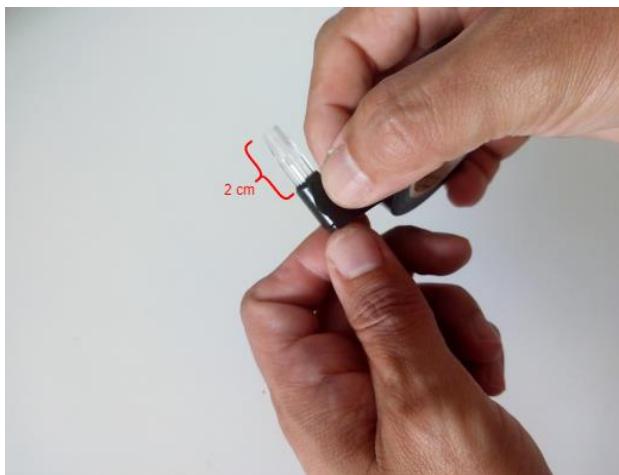


Figura 56: Fita isolante colocada a 2 cm da extremidade inferior.

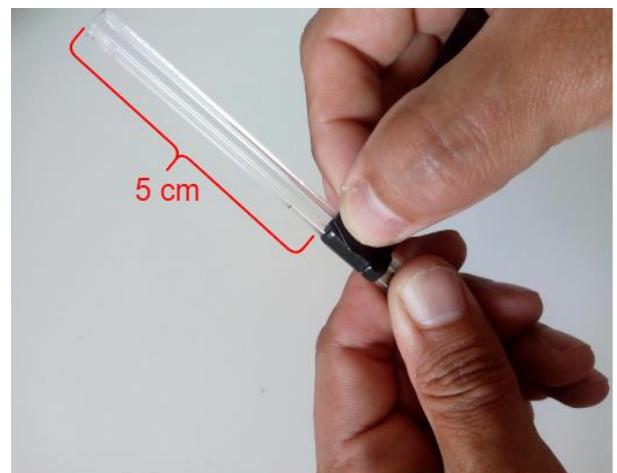


Figura 57: Fita isolante colocada a 5 cm da extremidade superior.

3. A função da fita isolante no refil da caneta é para que a mola possa exercer uma leve pressão na ponta da caneta. Colocar fita isolante nos pontos indicados (Figura 58).

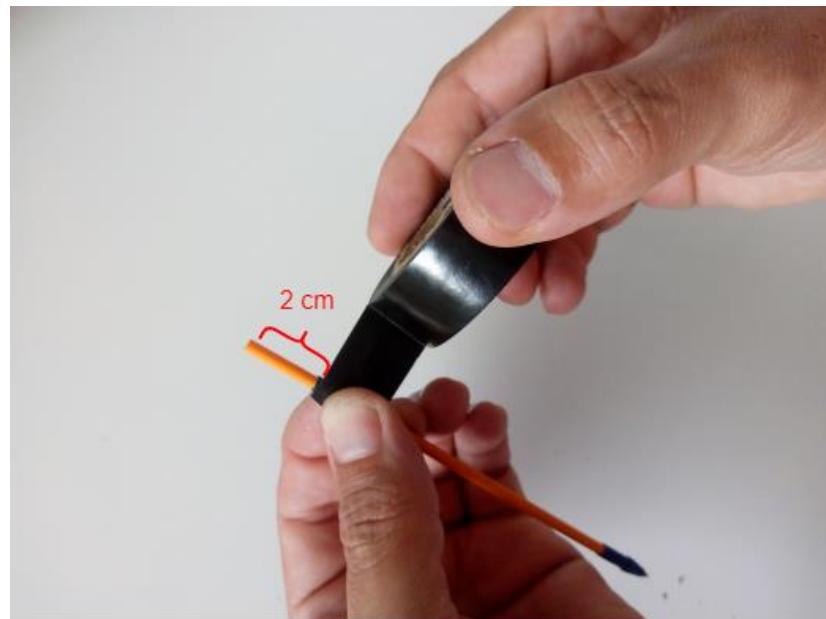


Figura 58: Fita isolante colocada a 2 cm da extremidade superior refil da caneta.

4. Para montar a caneta primeiro é colocado o refil no corpo de acrílico e em seguida colocada a mola (Figura 59). Por fim é colocada a tampa superior na caneta (Figura 60).



Figura 59: Colocar a mola na parte superior da caneta.

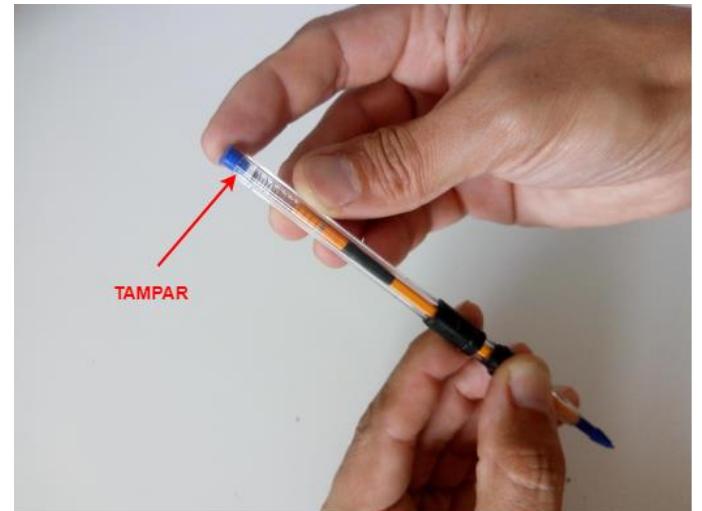


Figura 60: Tampar a parte superior da caneta.

A etapa de montagem do DrawBot está finalizada. A seguir será demonstrado como programar a placa Arduino.

### 3.4. Programação do Arduino



A seguir é descrito o procedimento de programação do Arduino Nano para que possa receber os comandos e realizar os movimentos.

1. Baixar e instalar a IDE de programação do Arduino em: <https://www.arduino.cc/en/guide/windows>
2. Abrir o software Arduino (Figura 61):

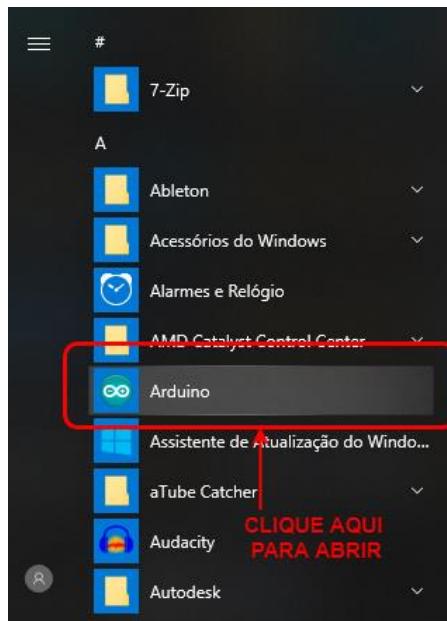


Figura 61: Atalho para o software de programação das placas Arduino

3. Conectar a placa Arduino Nano no DrawBot ao computador para fazer a programação da placa (Figura 62).

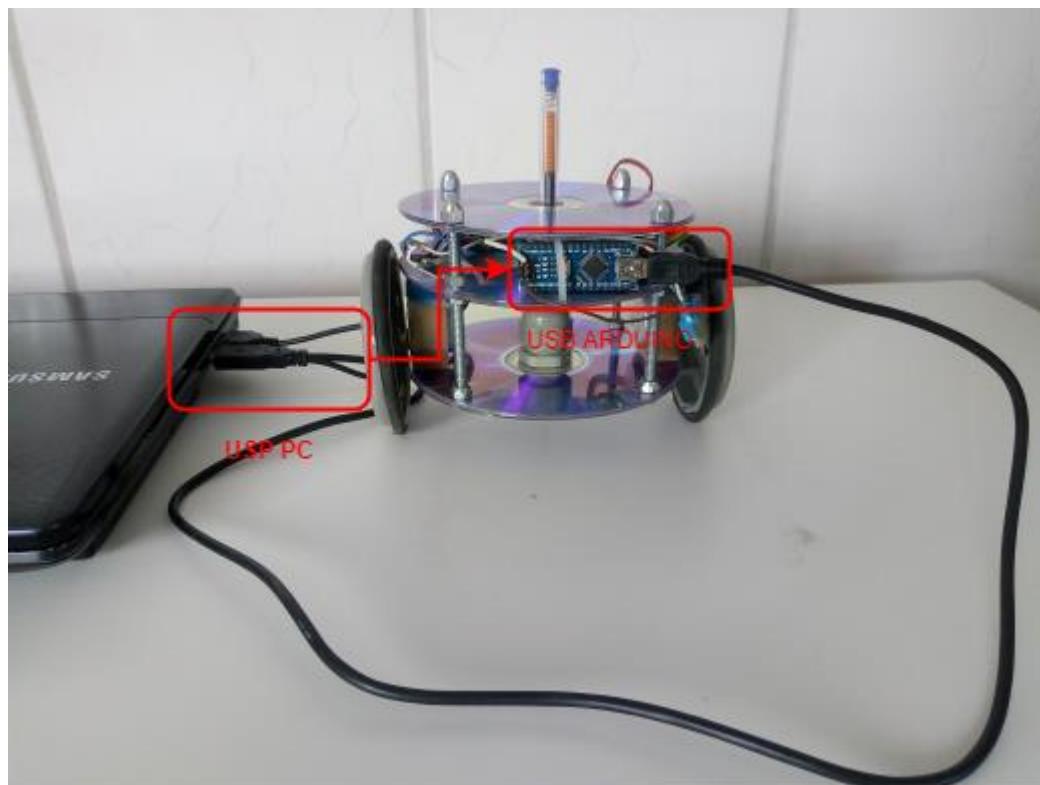


Figura 62: Conexão do computador ao Arduino Nano.

4. Selecionar a placa Arduino Nano. Como dito anteriormente a plataforma Arduino permite programar diversas placas, no nosso caso é a placa Arduino Nano (Figura 63).

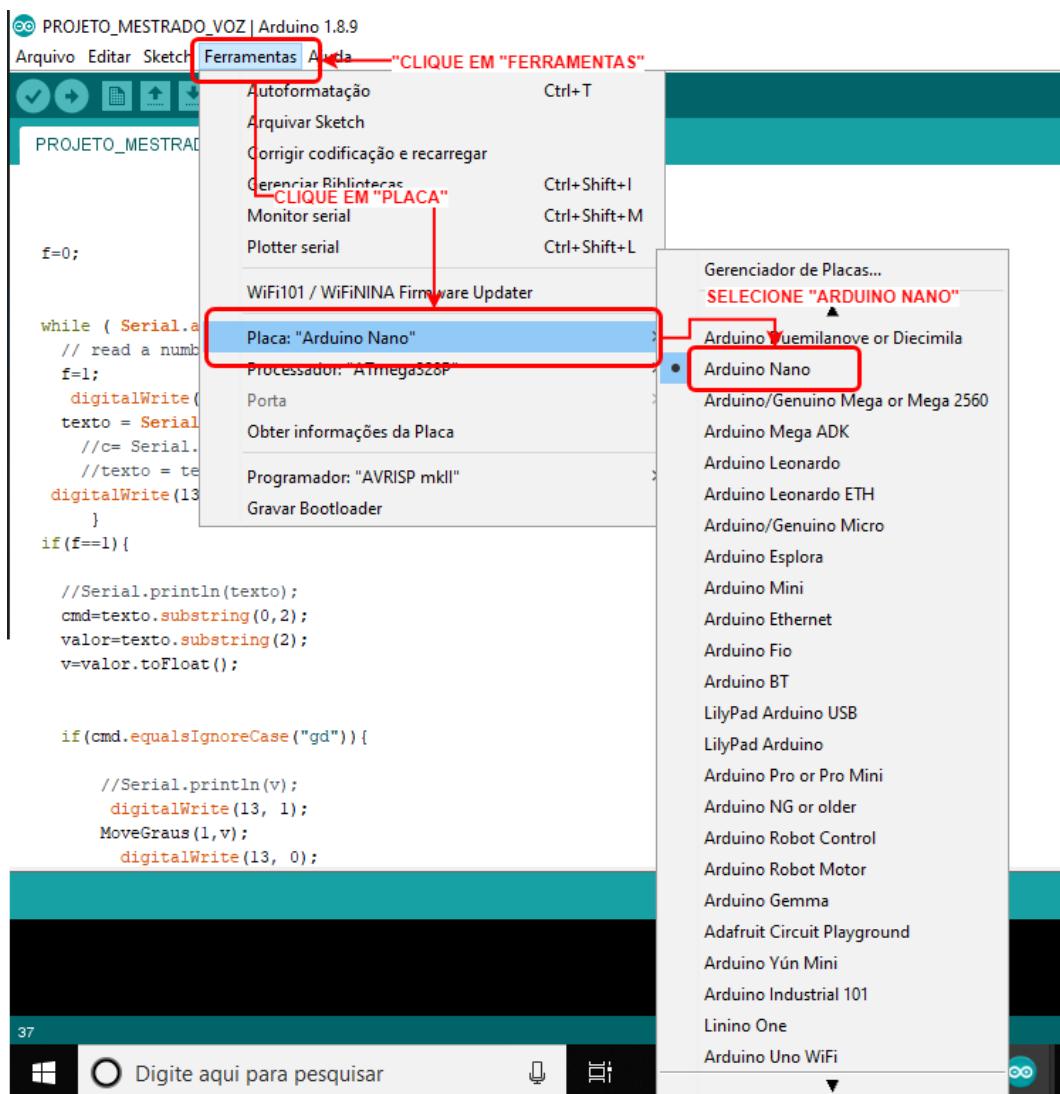


Figura 63: Etapas para selecionar a placa a ser programada.

5. Selecionar a porta de comunicação serial. Além de selecionar a placa é necessário selecionar a porta de comunicação que é feito como indicado na Figura 64.

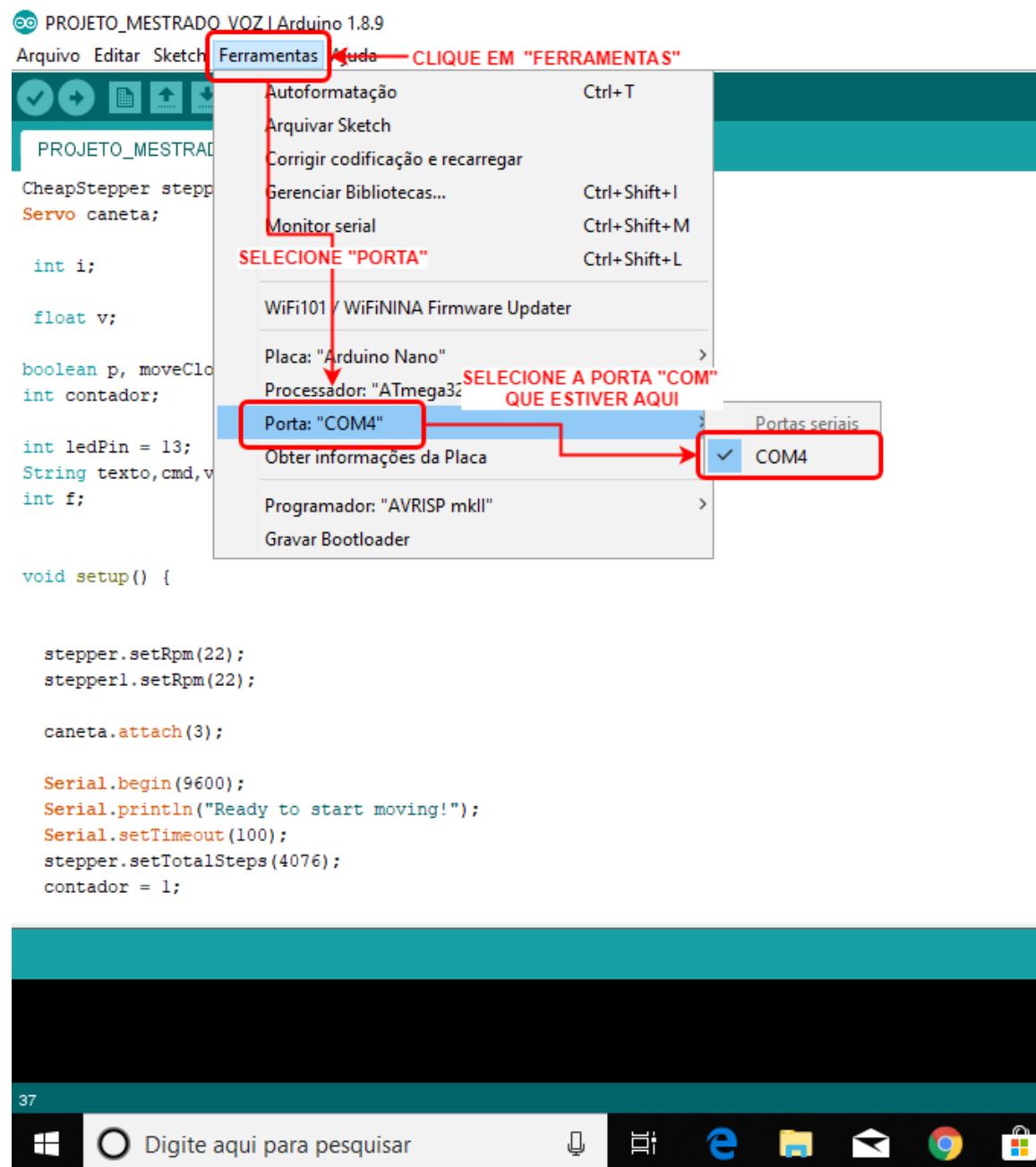


Figura 64: Selecionar a porta de comunicação.

6. Para gravar o código no Arduino, primeiro cria-se um novo arquivo e copia-se o código presente em **ARQUIVO D - Código para gravação da placa Arduino**, para o programa de gravação (Figura 65).

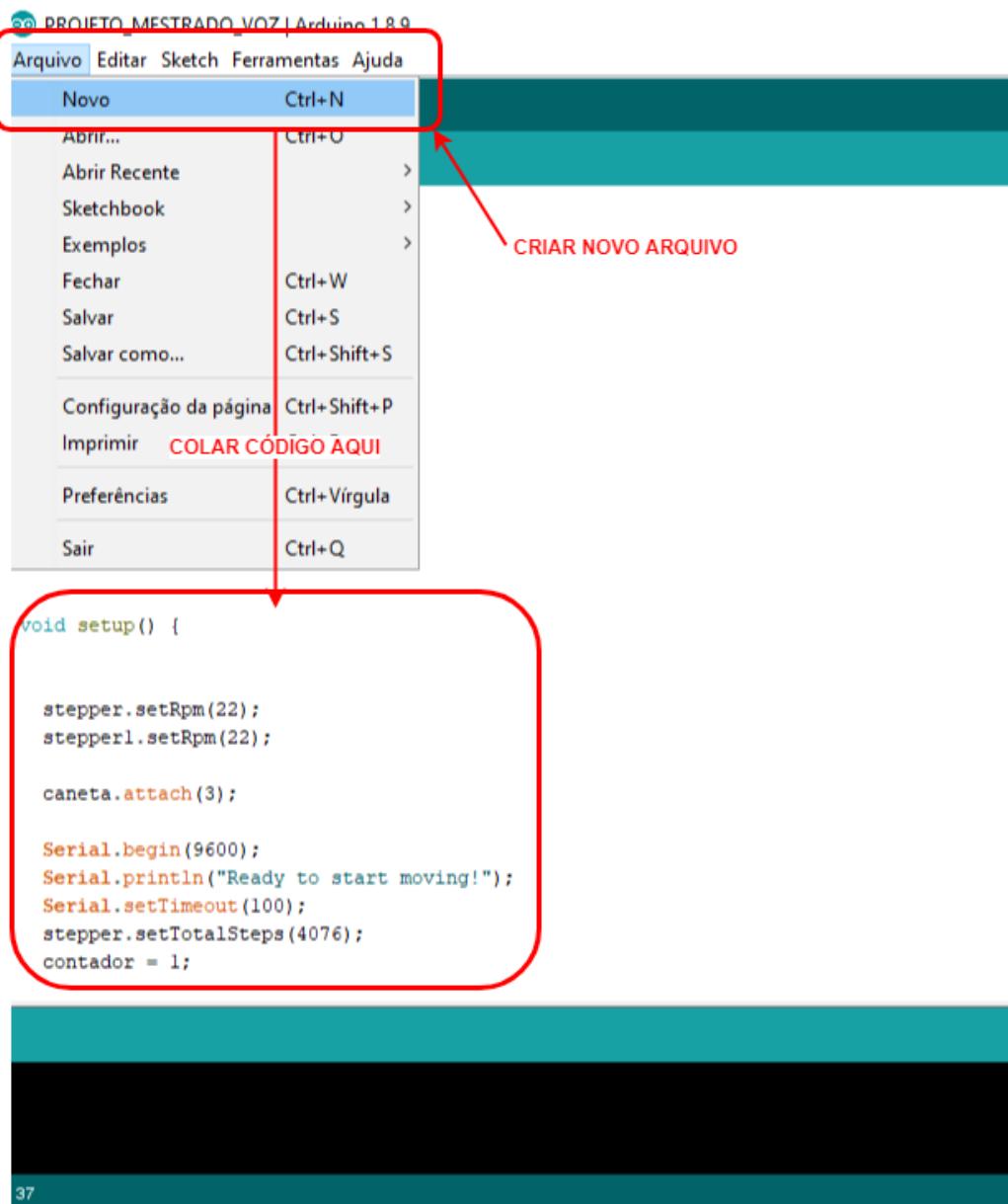


Figura 65: Criar novo arquivo e colar o código em ARQUIVO D.

7. Após criar o arquivo basta apertar o botão compilar e gravar o programa no Arduino Nano (Figura 66).



Figura 66: Botão para gravar o código no Arduino.

Não ocorrendo nenhum erro na gravação da placa Arduino o DrawBot executará um giro de 90 graus indicando que o mesmo foi programado corretamente.

## 4. Instalação e Execução do Octave

### 4.1. Configuração da placa Bluetooth.



[Configuração da placa Bluetooth](https://youtu.be/s3y8gXixg_g) - [https://youtu.be/s3y8gXixg\\_g](https://youtu.be/s3y8gXixg_g)

Para que o Octave possa se comunicar com o DrawBot é necessário configurar o módulo da placa Bluetooth antes da instalação do Octave.

1. Conectar a placa rs-232 (vermelha) a placa Bluetooth como indicado na Figura 67.

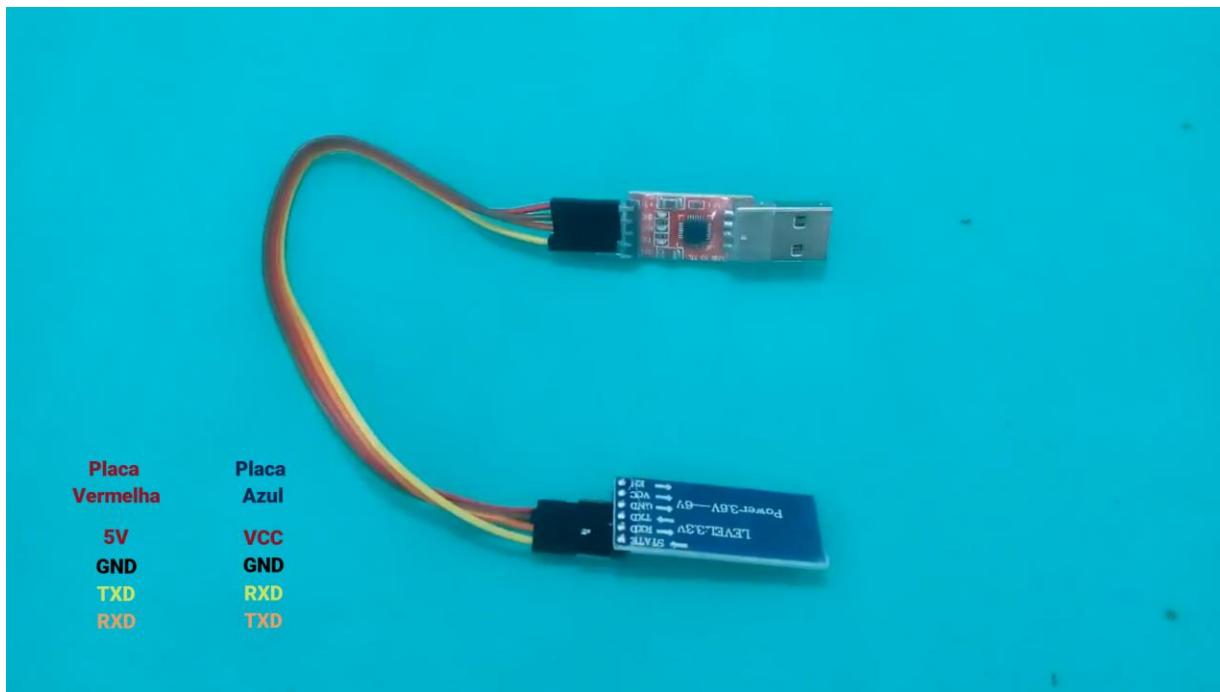


Figura 67: Conexão entre a placa rs-232 e o módulo Bluetooth.

2. Conectar a placa rs-232 (placa vermelha) à porta USB do computador como indicado na Figura 68.

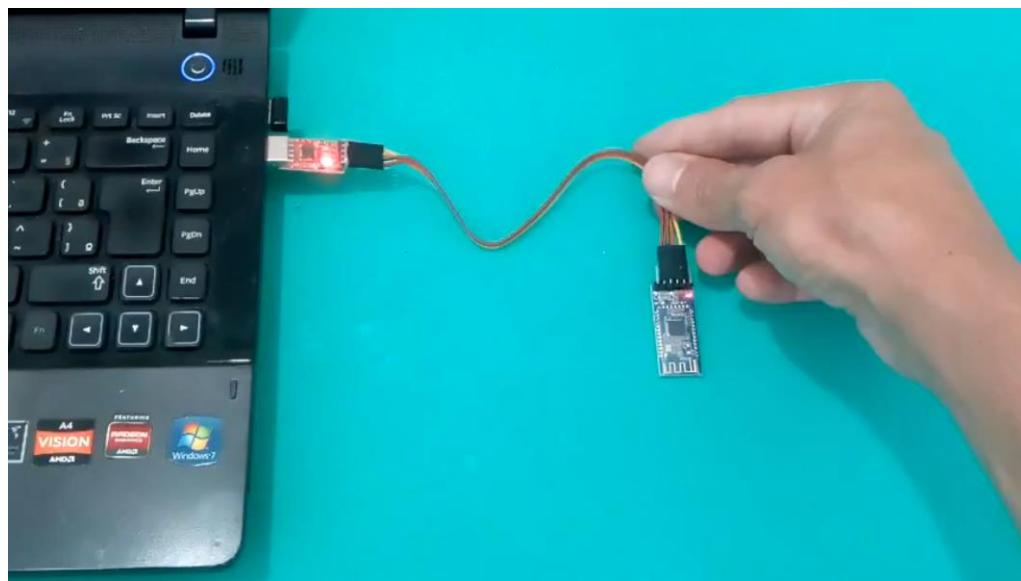


Figura 68: Conexão da placa rs-232 ao computador.

3. Abrir o programa Arduino IDE (Figura 69).



Figura 69: Ícone para abertura do programa ARDUINO IDE.

4. Selecionar porta serial (Figura 70).

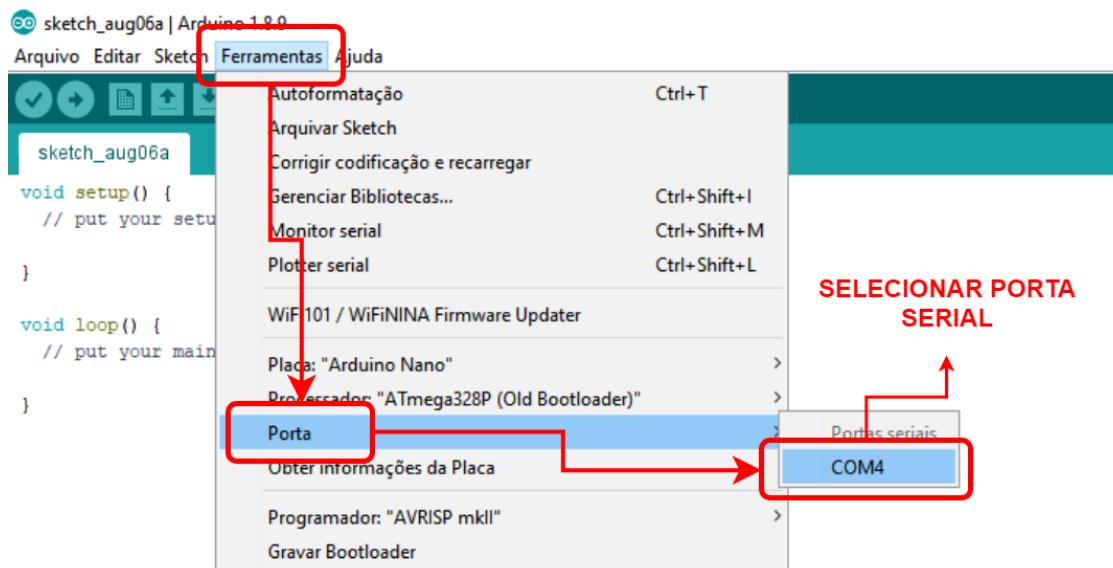


Figura 70: Porta serial.

5. Selecionar “Monitor serial” (Figura 71).

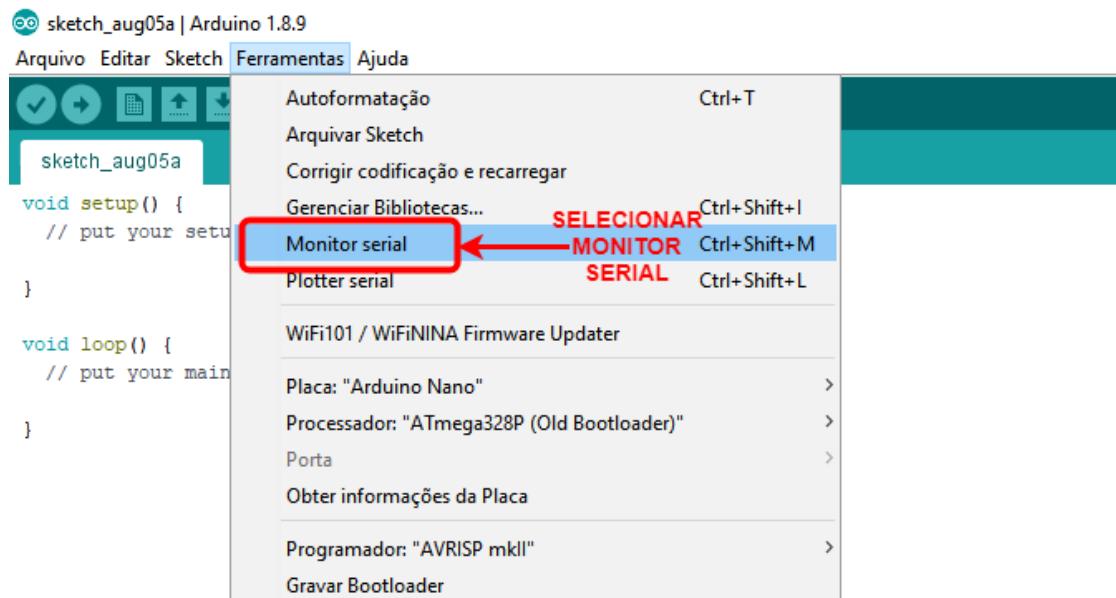


Figura 71: Funcionalidade de comunicação com a porta serial.

6. Digitar o comando “AT+HELP” (Figura 72) para lista comandos.

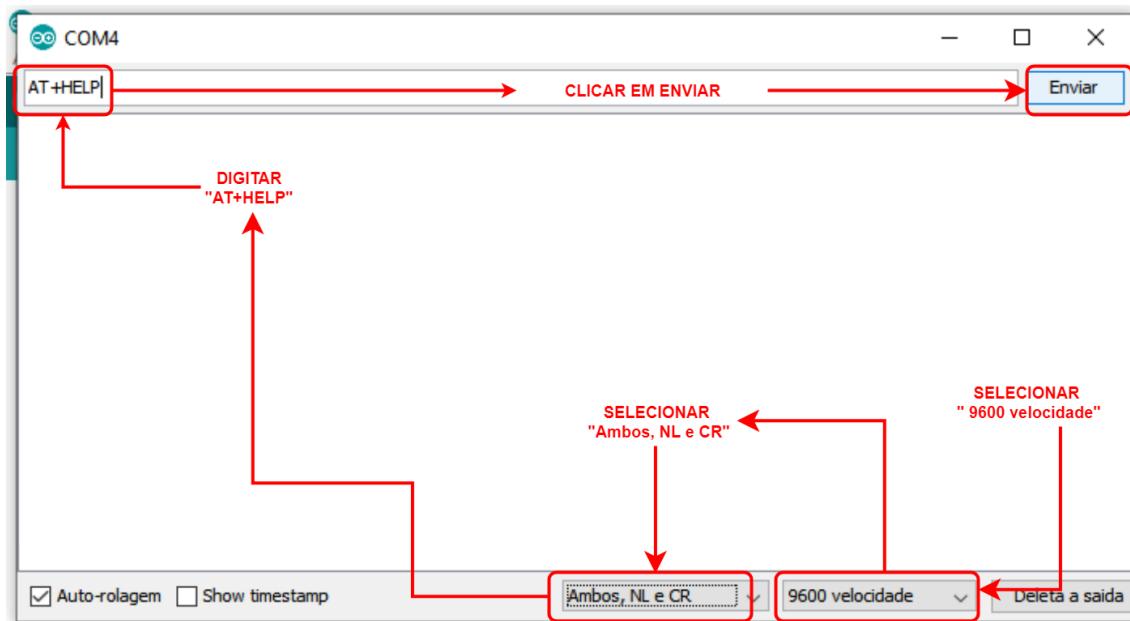


Figura 72: Funcionalidade de comunicação com a porta serial.

7. Tela com a lista de comandos (Figura 73).

Figura 73: Lista de comando do módulo Bluetooth.

8. Digitar o comando “AT+ROLE1” e clicar em enviar (Figura 74).

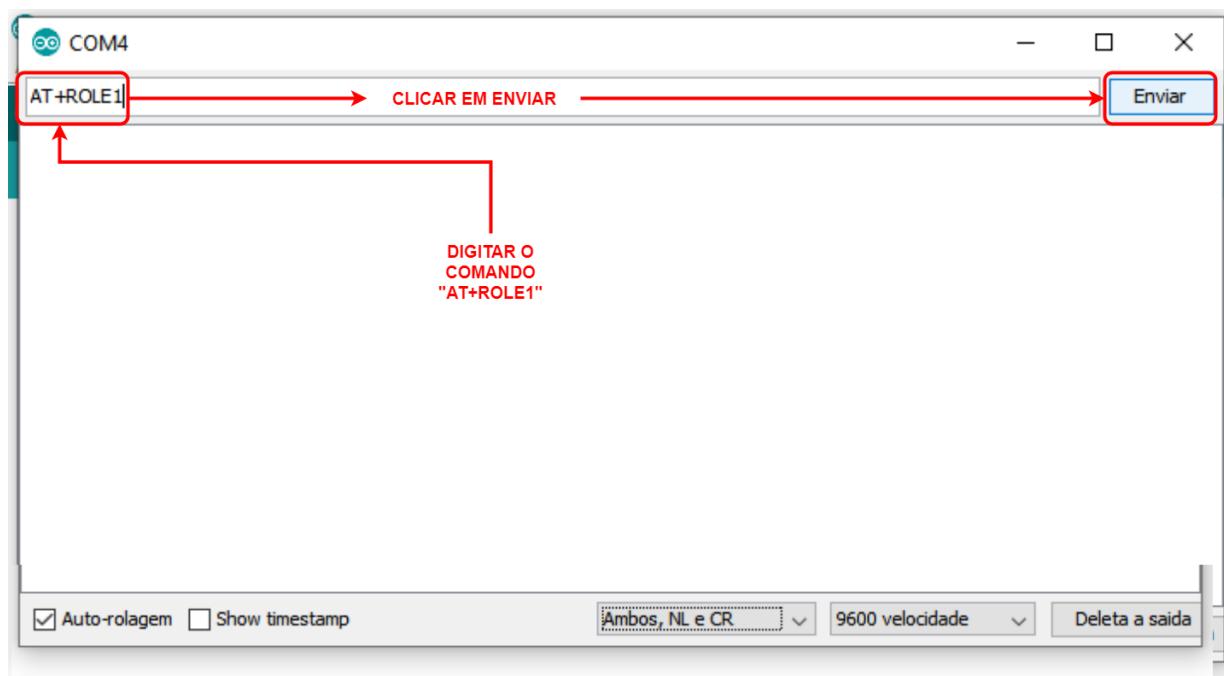


Figura 74: Comando “AT+HOLE1”.

9. Digitar o comando “AT+NAMEDRAWBOT” para dar nome ao dispositivo Bluetooth (Figura 75).

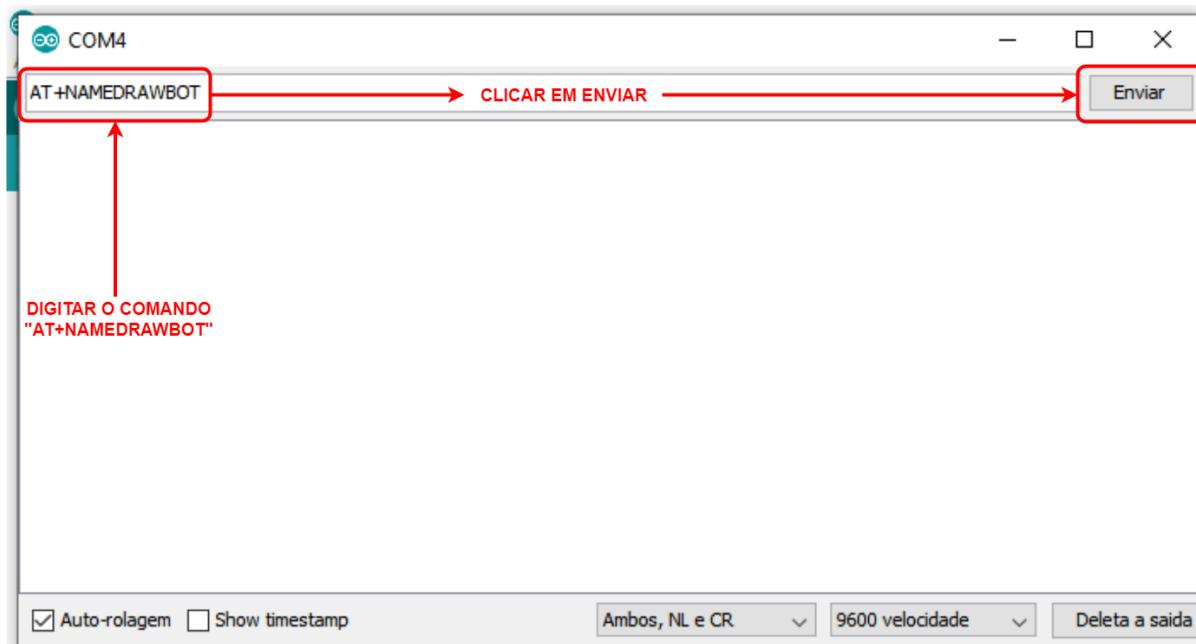


Figura 75: Comando para dar nome ao módulo Bluetooth.

10. Verificar se a configuração está correta (Figura 76, 77 e 78).

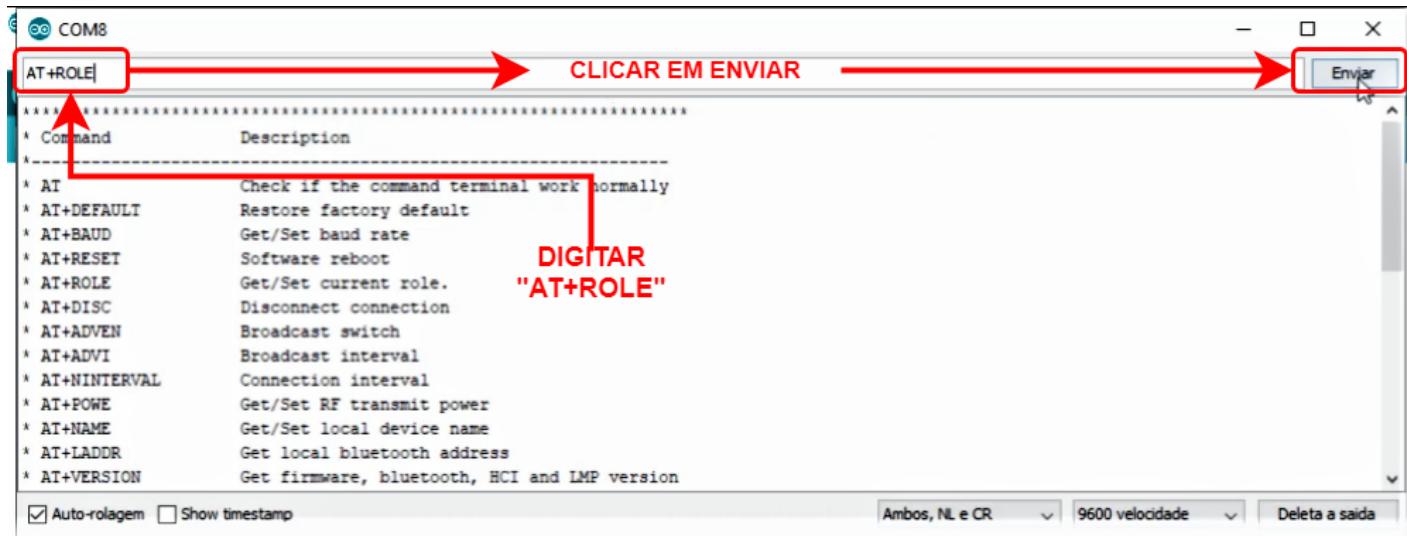


Figura 76: Comando "AT+ROLE".



Figura 77: Comando "AT+NAME".

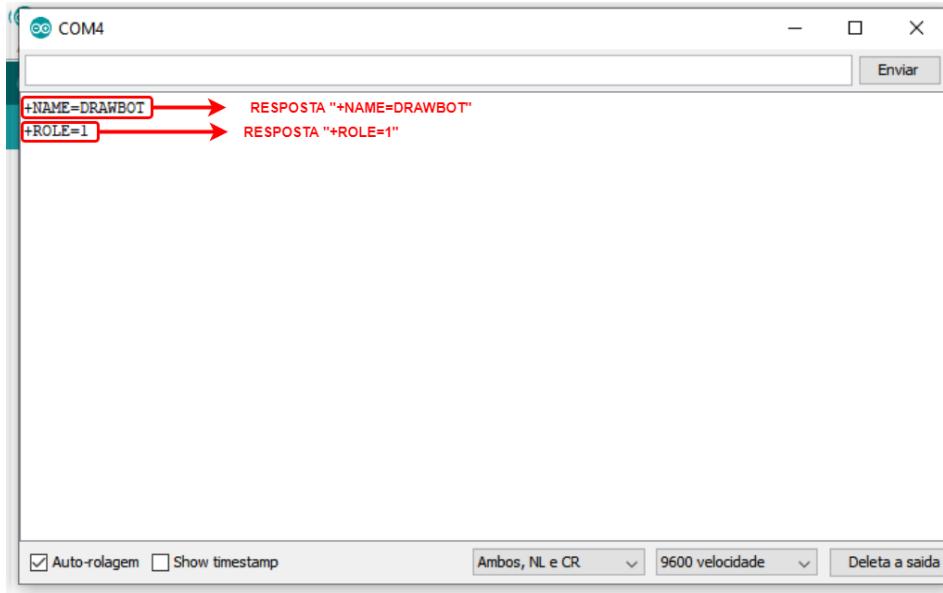


Figura 78: Resposta aos comandos “AT+HOLE e AT+NAME”.

## 4.2. Instalação do Octave.

1. Baixar o software Octave no computador em (Figura 79).

<https://www.gnu.org/software/octave/download.html>

Figura 79: Site para download do Octave.

2. Clicar no arquivo baixado (Figura 80).

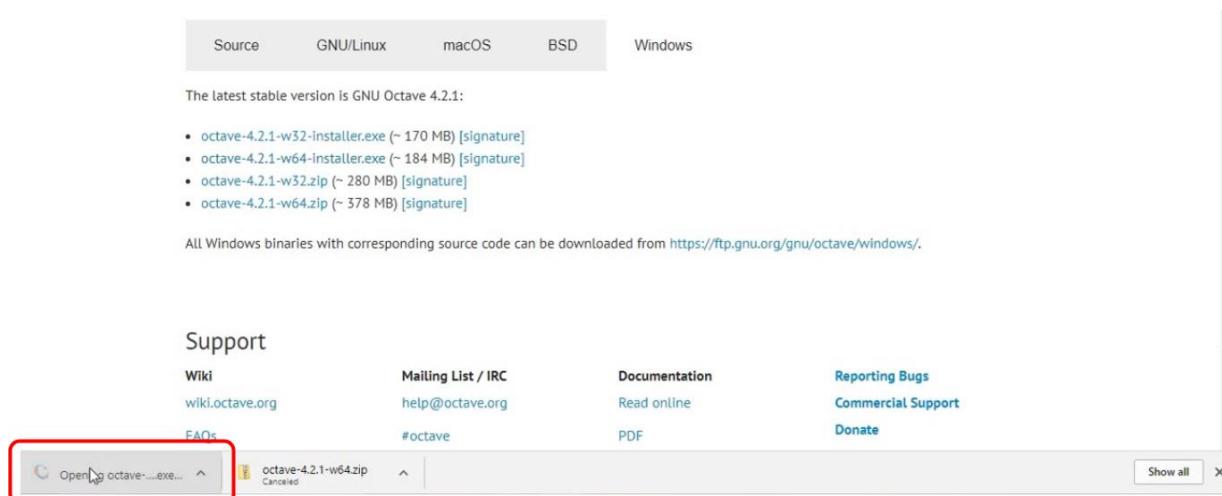


Figura 80: Executar o arquivo baixado.

3. Clicar em “next” (Figura 81).

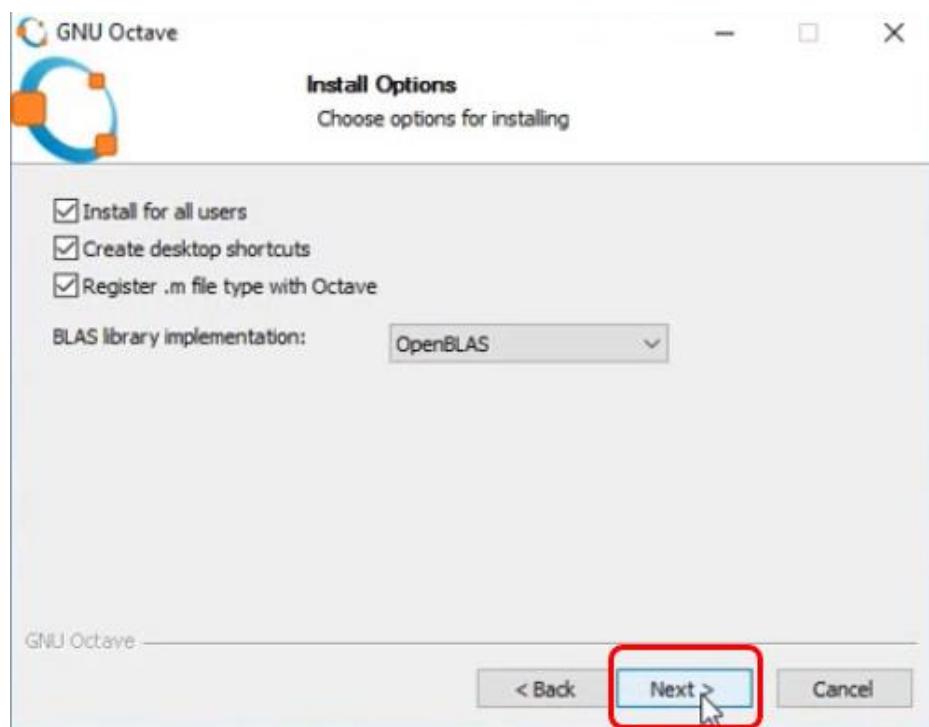


Figura 81: Clicar em "Next".

4. Esperar instalação (Figura 82).

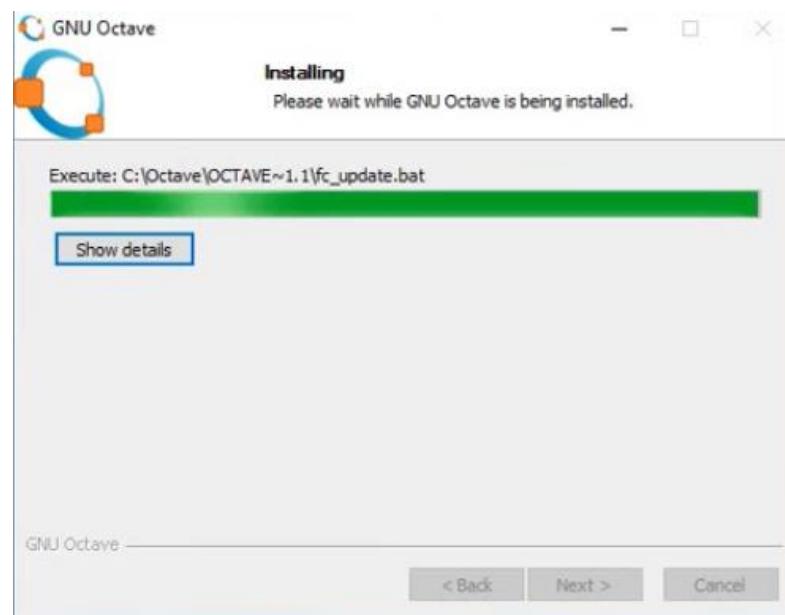


Figura 82: Octave sendo instalado.

5. Clicar em “Finish” (Figura 83).

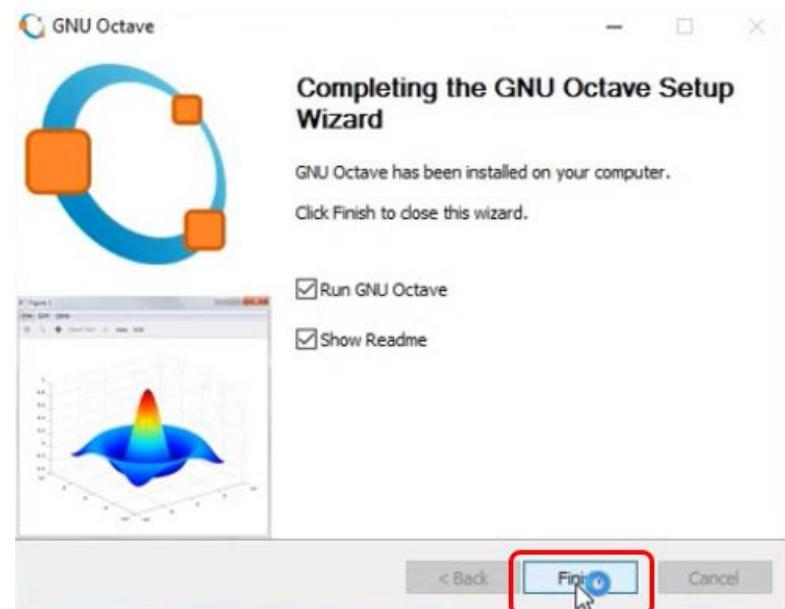


Figura 83: Tela de finalização da instalação.

#### 4.3. Criação dos arquivos de execução no Octave



[Criação dos arquivos de execução no Octave](https://youtu.be/ZNhSl_d_ZLHE) - [https://youtu.be/ZNhSl\\_d\\_ZLHE](https://youtu.be/ZNhSl_d_ZLHE)

Etapas para criação dos arquivos Octave que permitem a realização das atividades.

1. Criar uma pasta com o nome “ARQUIVOS OCTAVE” (Figura 84 e 85).

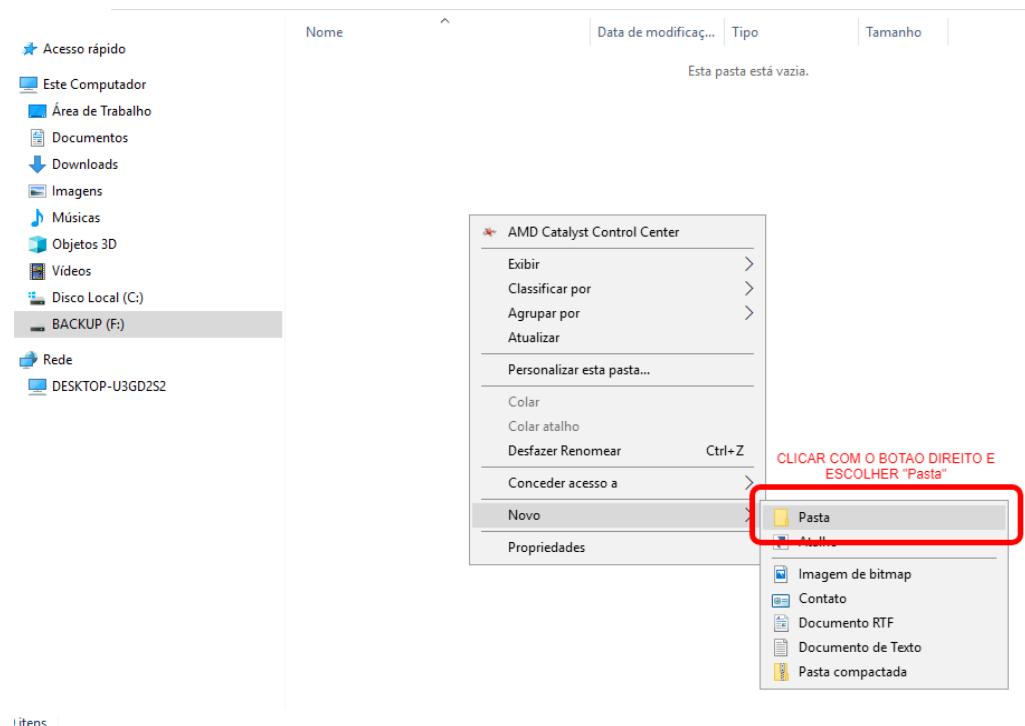
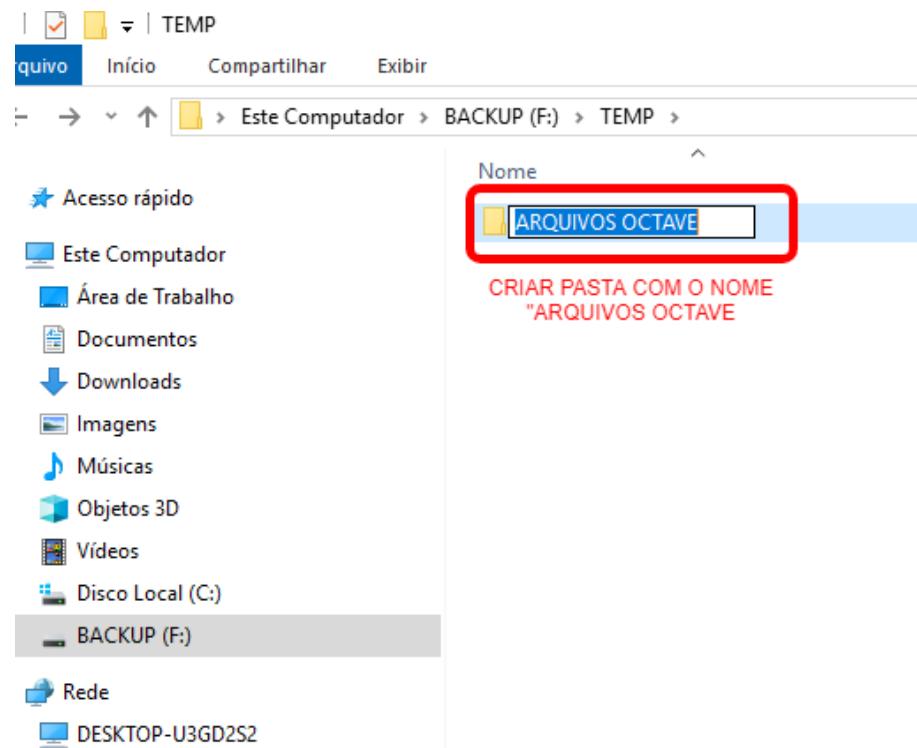
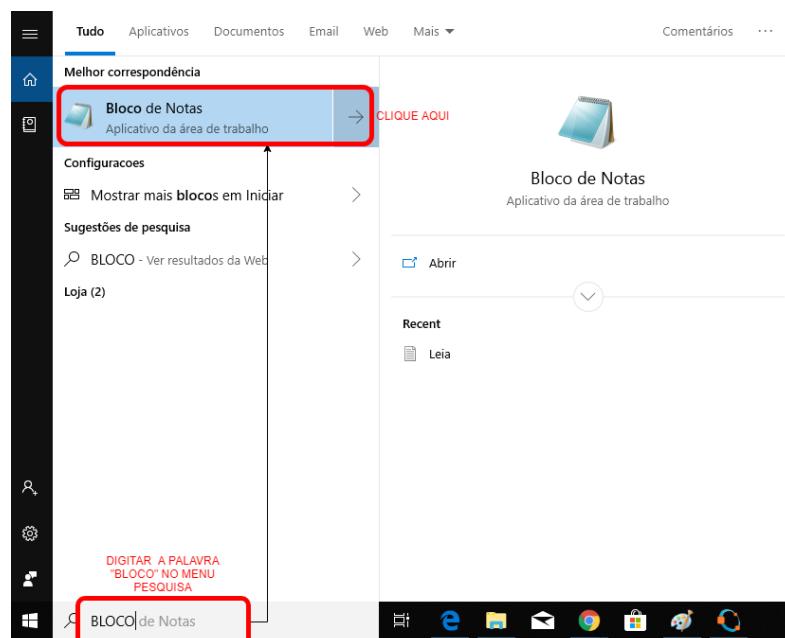


Figura 84: Processo criação da pasta para salvar os arquivos.



*Figura 85: Criação da pasta.*

2. Abrir o editor de texto “Bloco de notas” (Figura 86).



*Figura 86: Atalho para o bloco de notas.*

3. No ARQUIVO E – Código Octave, SETUP.m (Figura 87), copiar o texto delimitado por %INICIO E %FIM e colar no editor de texto Bloco de Notas (Figura 88) e salvar com o nome “SETUP.m” dentro da pasta “ARQUIVOS OCTAVE” (Figura 89).

#### ARQUIVO E - Código Octave SETUP.m

Este inicia as bibliotecas necessárias para o funcionamento do programa.

Para gerar o arquivo SETUP.m, copiar o código delimitado por "%INICIO" e "%FIM".

```
%INICIO
%SETUP.m
%BIBLIOTECAS E ARQUIVOS DE INICIALIZAÇÃO
graphics_toolkit('qt'); % Biblioteca gráfica
pkg load optim; % Biblioteca que possui a função do método dos mínimos quadrados
pkg load instrument-control; % Biblioteca para comunicação serial
pkg load signal; % Biblioteca para captura do som
SINNSTART;
AJUSTEDECURVA_CALCULO;
AJUSTEDECURVA_PLOT;
%FIM
```

Figura 87: Código a ser copiado.

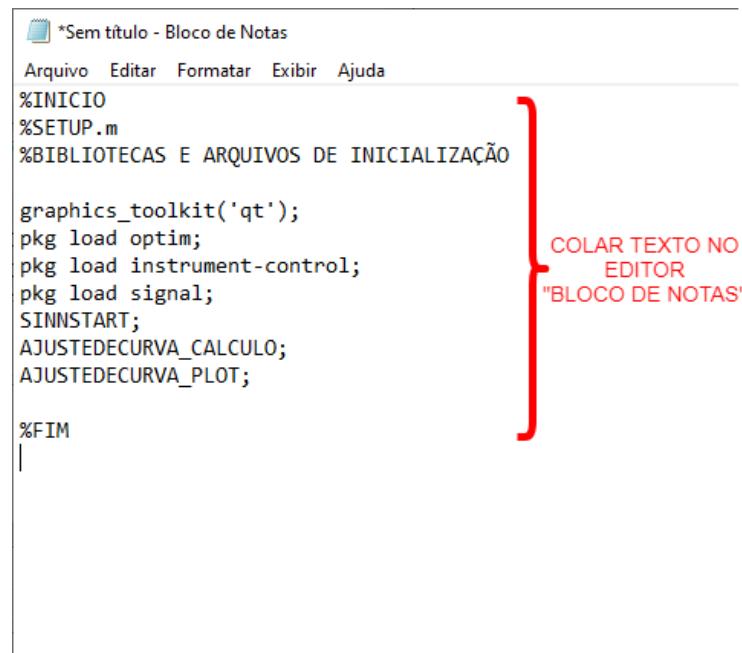


Figura 88: Código copiado para o editor de texto Bloco de Notas.

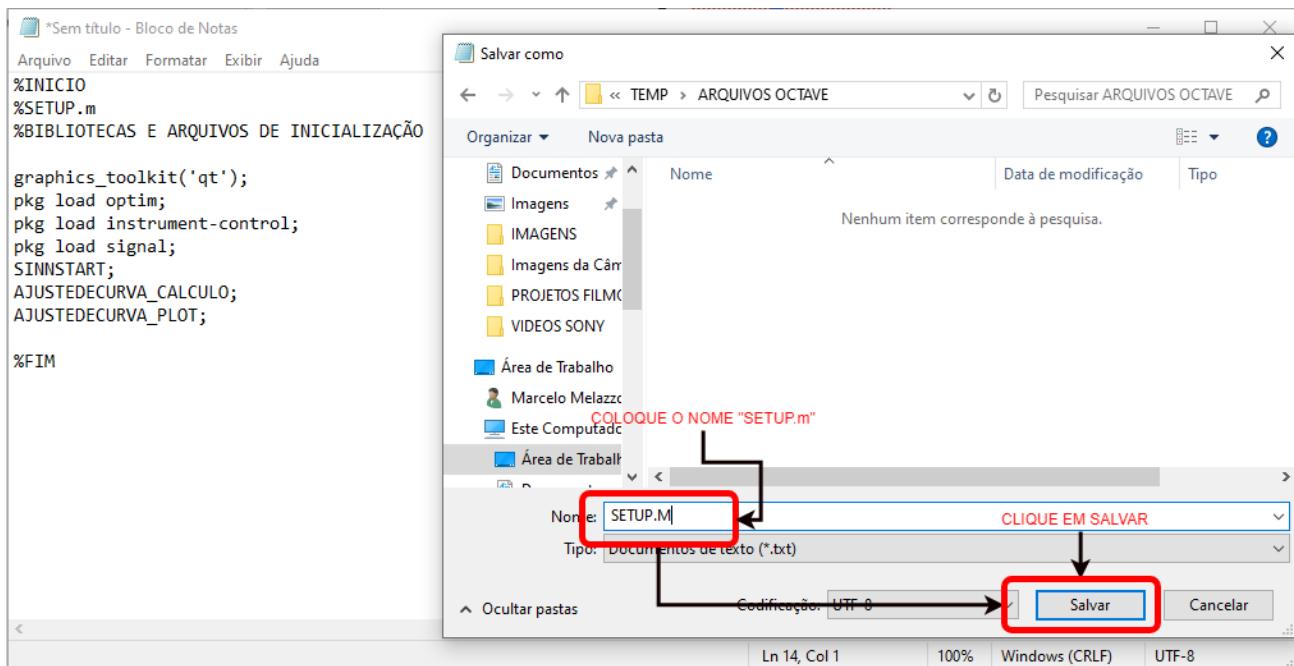


Figura 89: Salvar arquivo na pasta "ARQUIVOS OCATAVE"

4. Repetir o procedimento anterior para a criação dos arquivos localizados ao fim deste manual:

**ARQUIVO F – Código Octave SERIAL.m;**

**ARQUIVO G – Código Octave AJUSTEDECURVA\_CALCULO.m;**

**ARQUIVO H – Código Octave AJUSTEDECURVA\_PLOT.m;**

**ARQUIVO I – Código Octave DRAWBOT.m;**

**ARQUIVO J – Código Octave GRAFICO3D.m;**

**ARQUIVO K – Código Octave GRAFICO4.m;**

#### 4.4. Execução e teste dos arquivos no Octave

 [Execução e teste dos arquivos no Octave - https://youtu.be/3NR55jhEOvI](https://youtu.be/3NR55jhEOvI)

1. O Octave no computador se comunica com o DrawBot por meio do “MÓDULO USB” e o “MÓDULO BLUETOOTH” (Figura 90). Este se comunica com o módulo Bluetooth presente no DrawBot. Abaixo é demonstrado a conexão dos mesmos.



Figura 90: Conexão do Módulo USB ao Módulo Bluetooth

2. Para que o Octave possa se comunicar com o DrawBot é necessário descobrir a porta de comunicação do módulo USB. Abaixo é dado os passos para a identificação do número da porta (Figura 91 e 92).

Ajuste as configurações do computador Exibir por: Categoría ▾

 Sistema e Segurança Verificar o status do computador Salvar cópias de backup dos arquivos com Histórico de Arquivos Backup e Restauração (Windows 7)	 Contas de Usuário Alterar o tipo de conta
 Rede e Internet Exibir o status e as tarefas da rede	 Aparência e Personalização
 Hardware e Sons <b>Exibir impressoras e dispositivos</b> <span style="border: 1px solid red; padding: 2px;">CLIQUE AQUI</span> Adicionar dispositivo Ajustar as configurações de mobilidade comumente usadas	 Relógio e Região Alterar formatos de data, hora ou número
 Programas Desinstalar um programa	 Facilidade de Acesso Permitir que o Windows sugira configurações Otimizar exibição visual

Figura 91: Local onde se identifica o número da porta

### 3. Acessar no Octave a pasta “ARQUIVOS OCTAVE” (Figura 93).

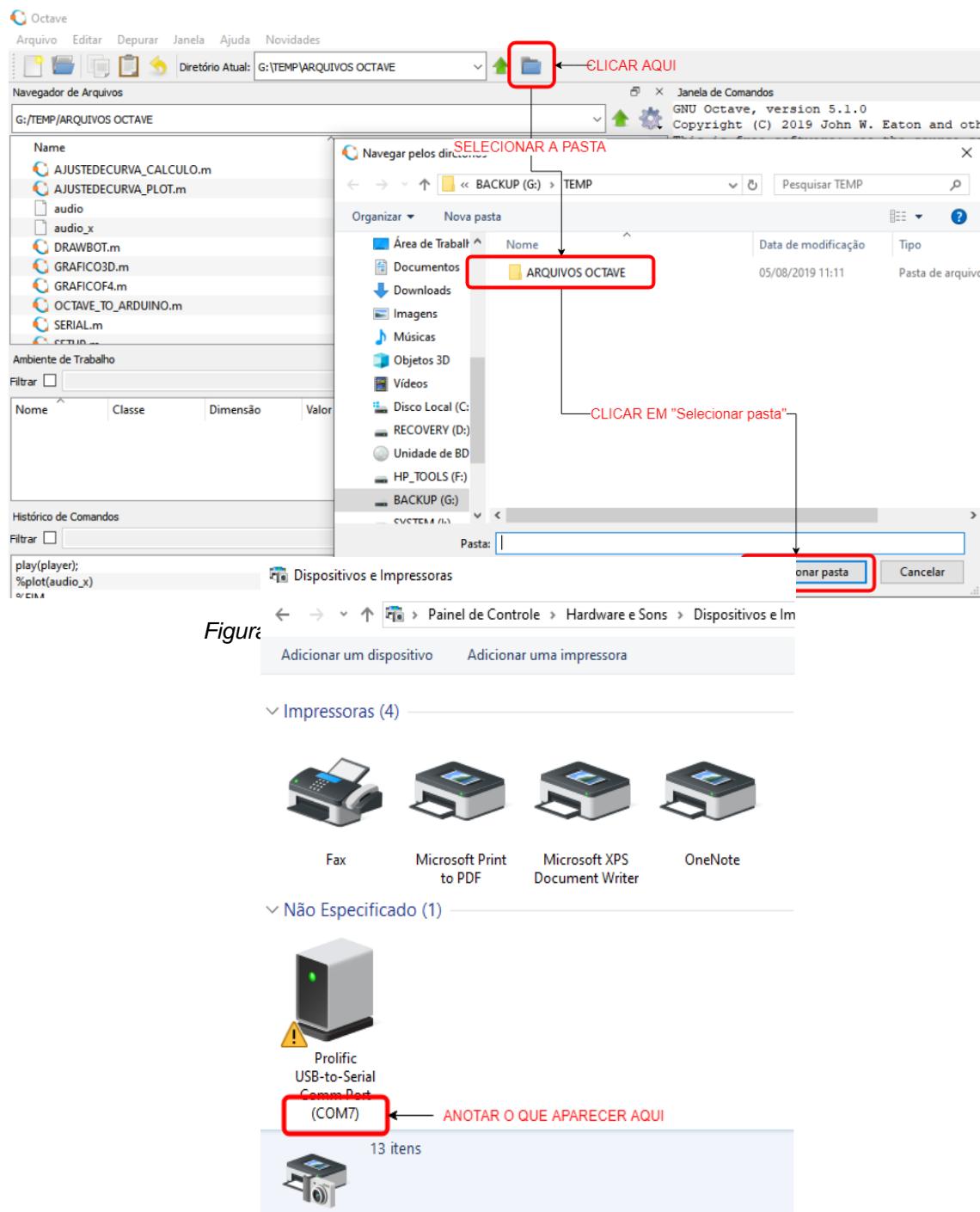


Figura 92: anotar o número que aparece nesse local

4. Clicar no arquivo “SETUP.m” e executar (Figura 94). Este arquivo é o responsável pelas configurações iniciais.

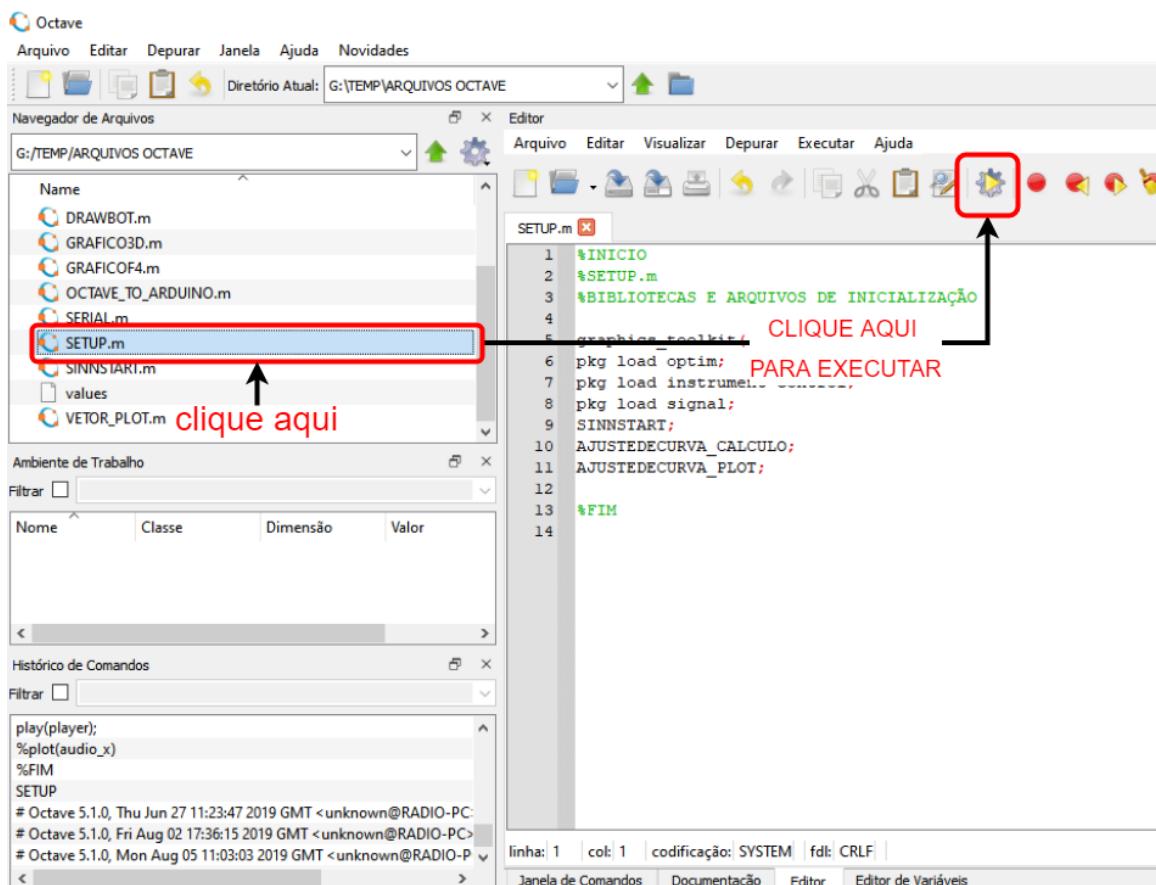
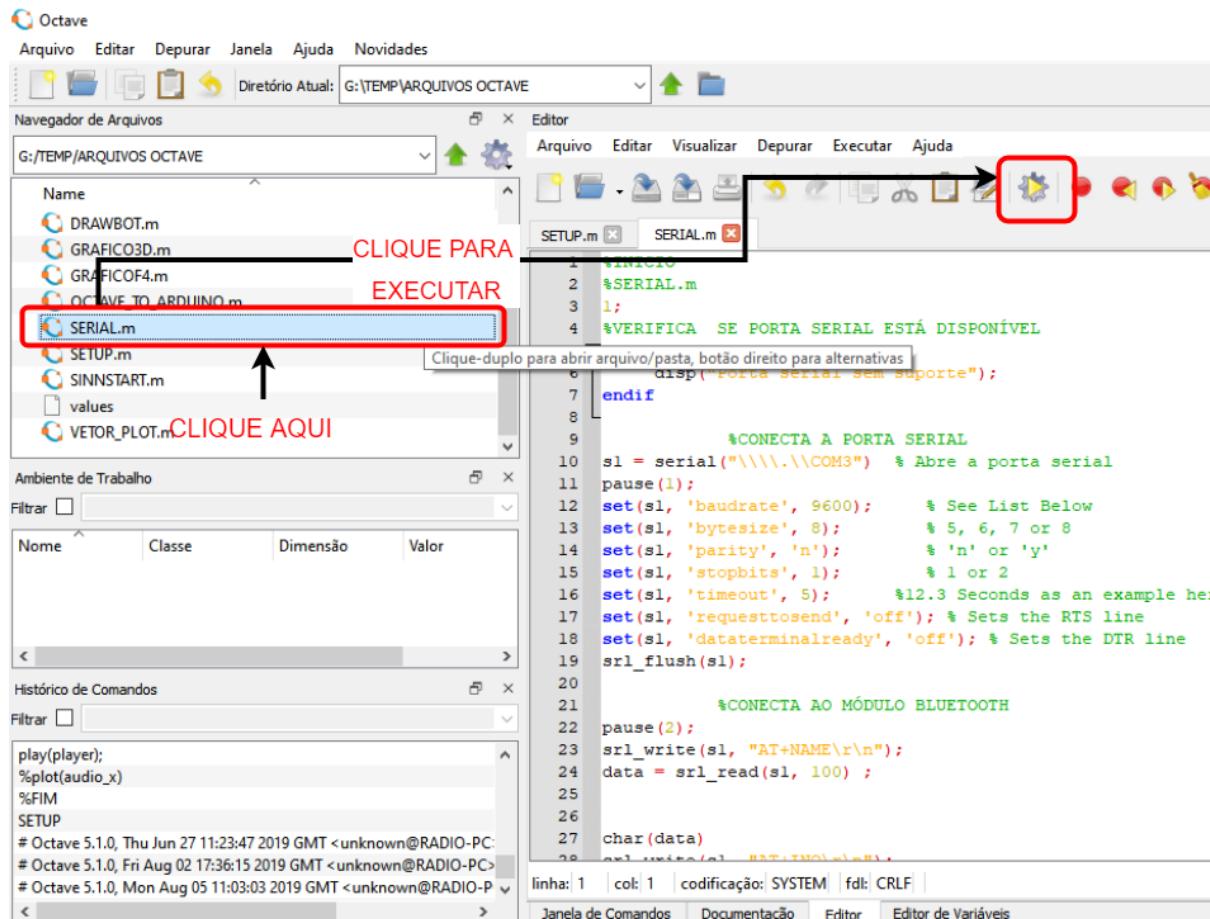


Figura 94: Procedimento para execução do arquivo setup.m.

5. Clicar no arquivo “**SERIAL.m**” e executar (Figura 95). Neste momento o DrawBot deve estar ligado para que o Octave possa se comunicar com o DrawBot.



*Figura 95: Execução do arquivo SERIAL.m.*

6. Clicar no arquivo “AJUSTEDECURVA\_CALCULO.m” e falar a vogal “i” e apertar a tecla “F5” e (Figura 96). Neste passo é feito o cálculo do ajuste de curvas relativos ao som da vogal “i”.

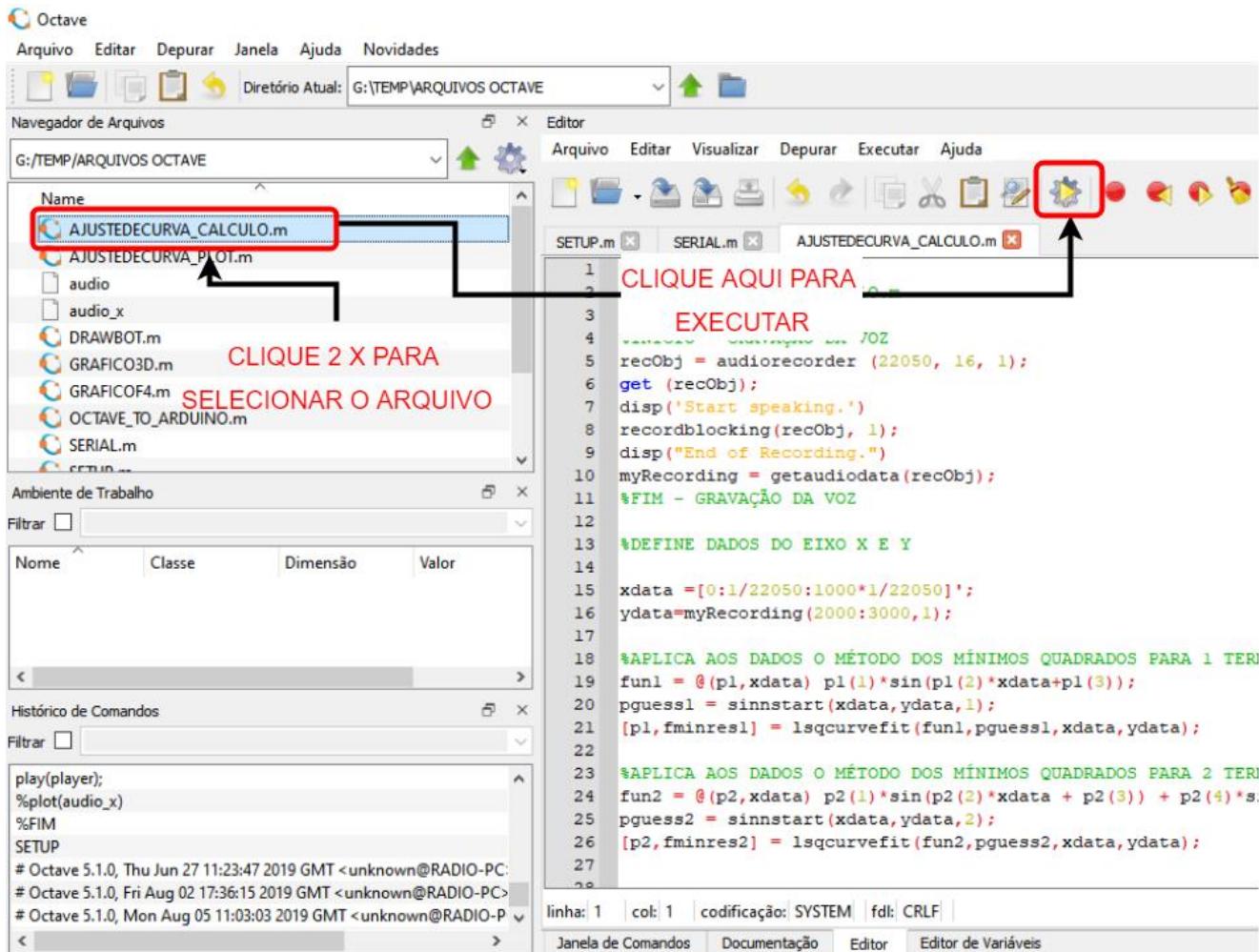


Figura 96: Execução do AJUSTEDECURVA\_CALCULO.m.

7. Clicar no arquivo “AJUSTEDECURVA\_PLOT.m” e apertar a tecla “F5” e (Figura 97). Aqui é construído os gráficos relativos ao som da vogal “i”.

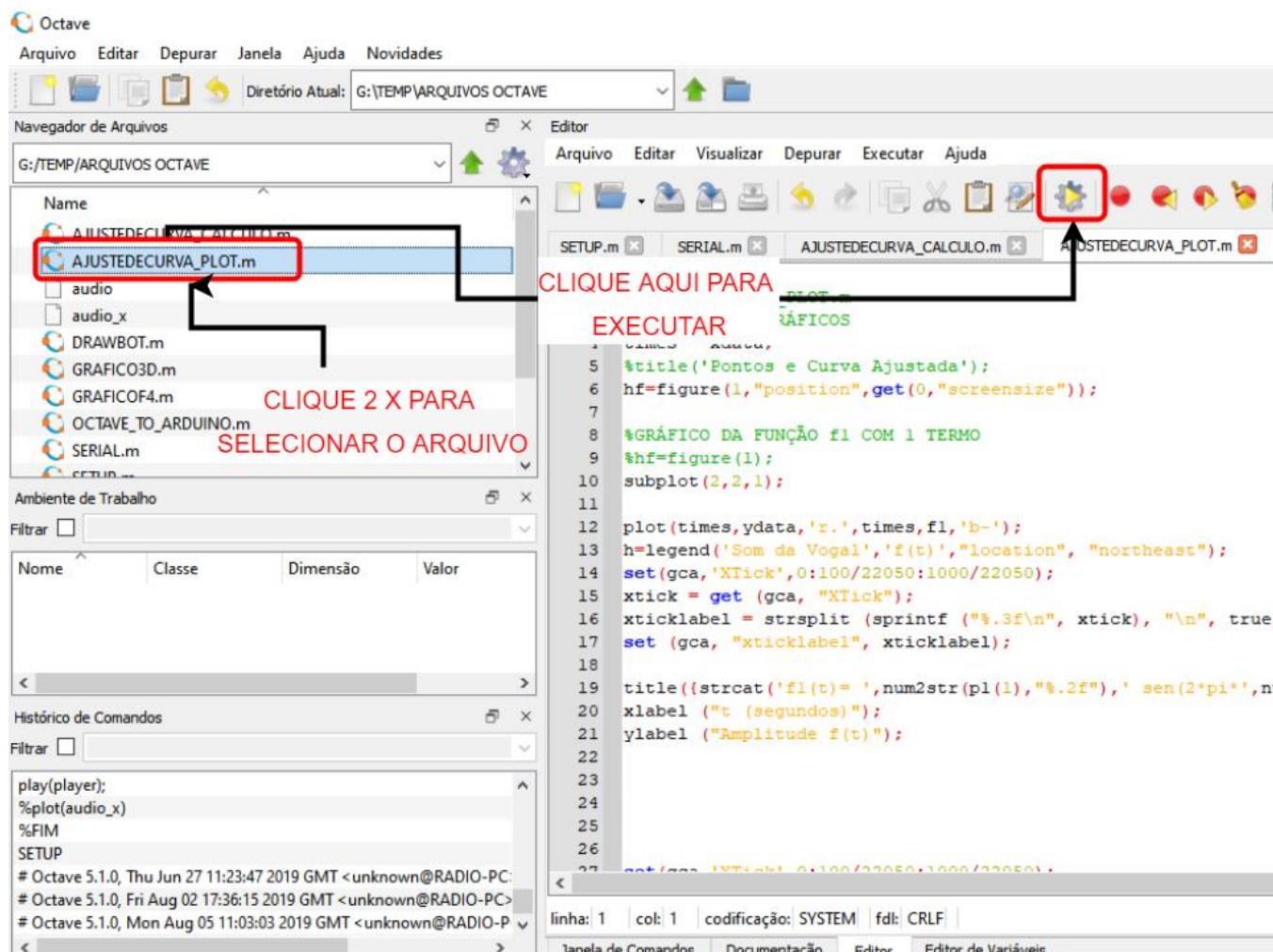


Figura 97: Execução do arquivo AJUSTEDECURVA PLOT.m.

8. Clicar no arquivo “DRAWBOT.m” e apertar a tecla “F5” e esperar o DrawBot fazer o gráfico (Figura 98). Nesse momento é dado o comando para o DrawBot desenhar o gráfico obtido pelo o algoritmo presente no arquivo “AJUSTEDECURVA\_CALCULO.m” que foi executado anteriormente.

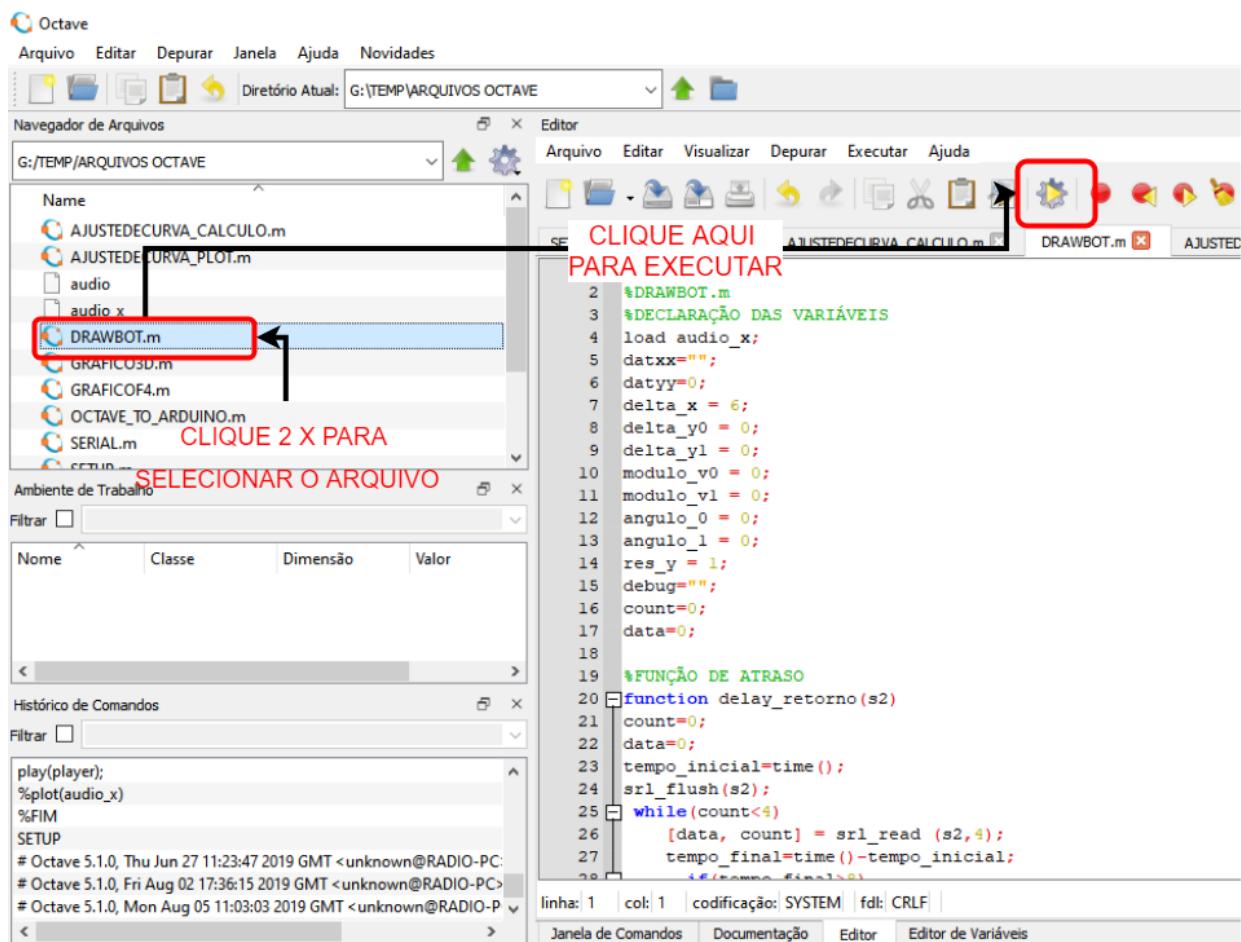


Figura 98: Execução do arquivo DRAWBOT.m.

9. Clicar no arquivo “GRAFICO3D.m” e o gráfico será gerado. Esta etapa permite visualizar em um gráfico em 3 dimensões a soma obtida pelo ajuste de curvas (Figura 99 e 100).

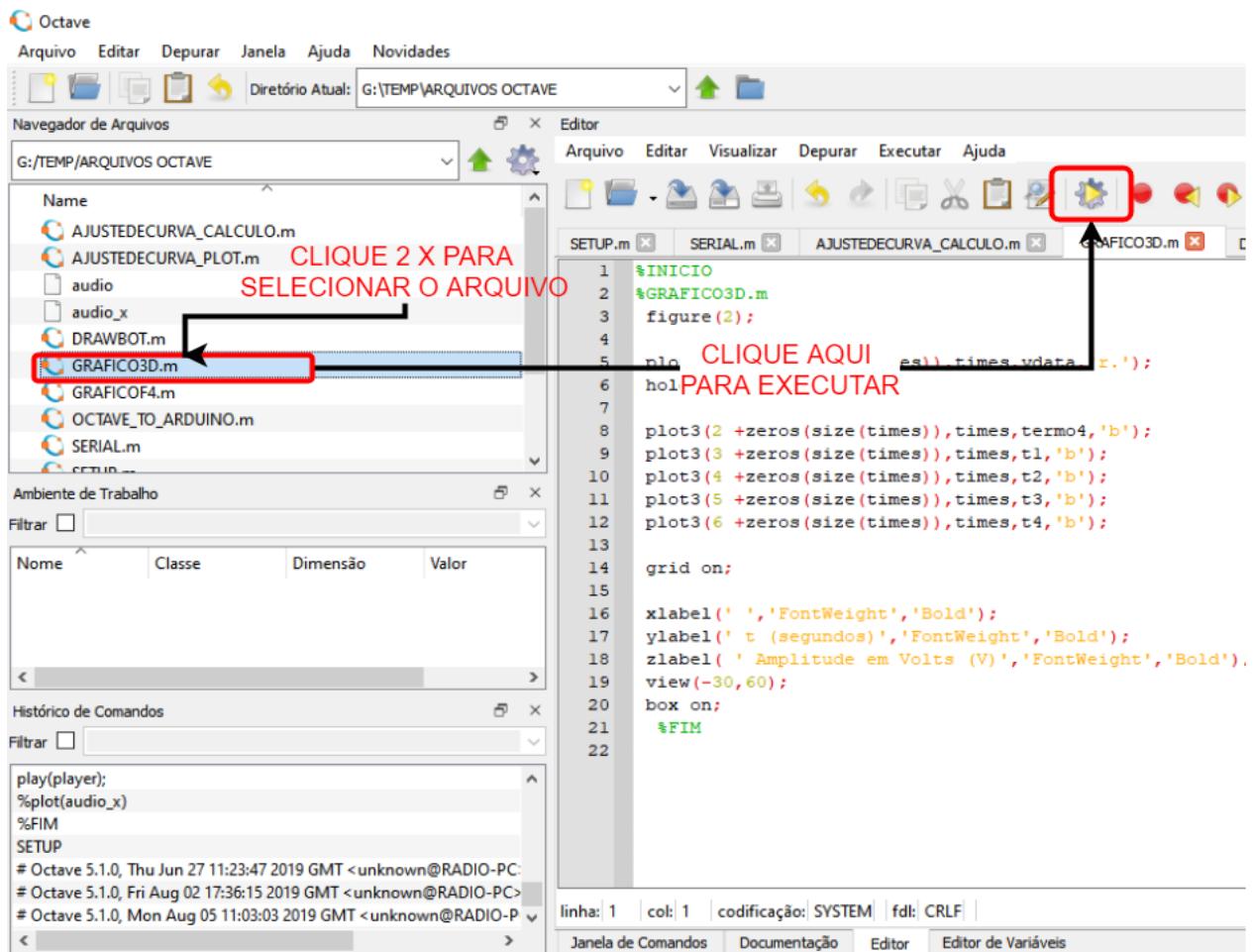


Figura 99: Execução do arquivo GRAFICO3D.m.

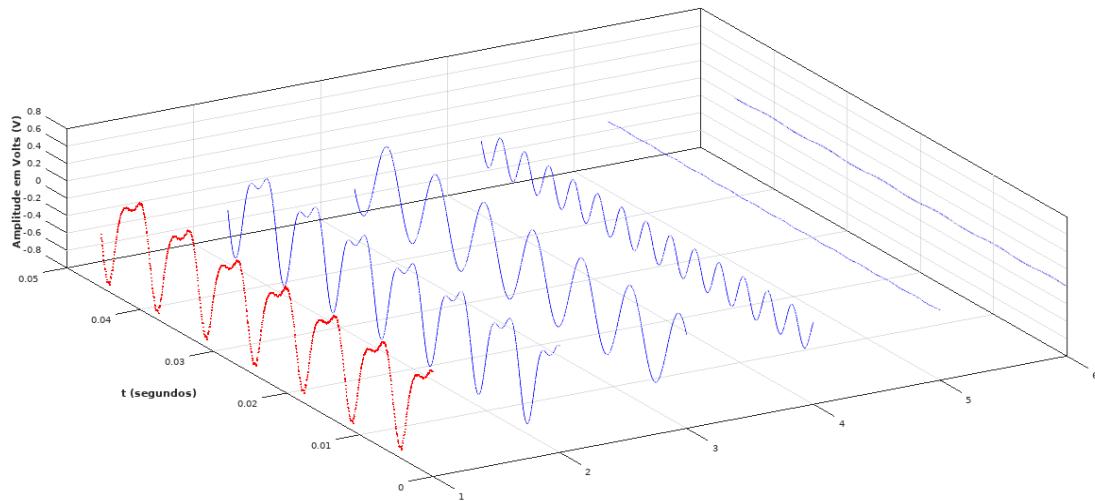


Figura 100: Geração do gráfico 3D.

10. Clicar no arquivo “GRAFICOF4.m” e o gráfico será gerado (Figura 101).

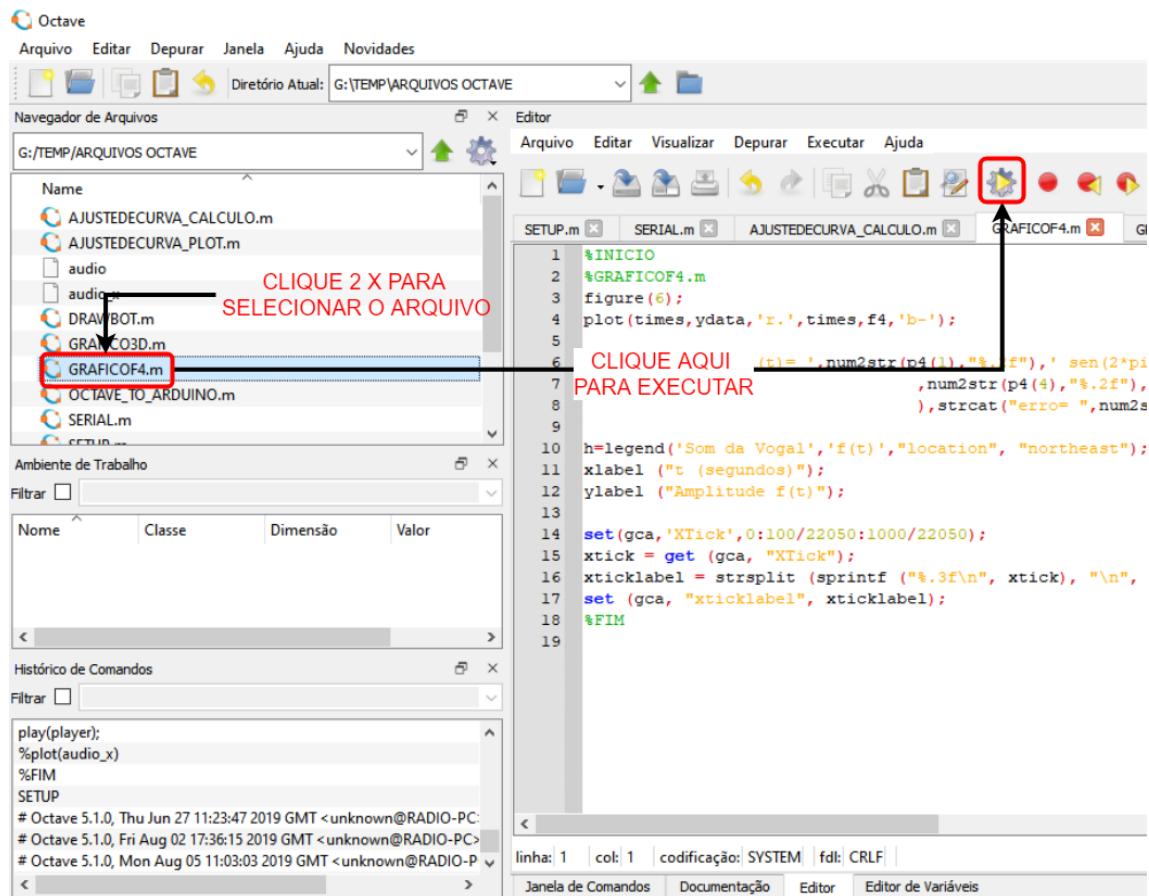


Figura 101: Execução do arquivo GRAFICOF4.m.

## 5. Instalação do aplicativo de controle do DrawBot no celular

### 5.1. Visão geral do aplicativo.

O aplicativo (Figura 102) tem a função de controlar o DrawBot foi desenvolvido para realização de atividades que leva ao uso de funções trigonométricas. Sua tela é composta de comandos simples e possui os botões de movimento para frente, para trás e giro à esquerda e direita. Além disso possui também uma caixa de entrada numérica para valores em milímetros para os movimentos lineares e em graus para o giro do Drawbot. No final deste material serão disponibilizadas as atividades desenvolvidas em sala de aula com o uso do aplicativo e o DrawBot.

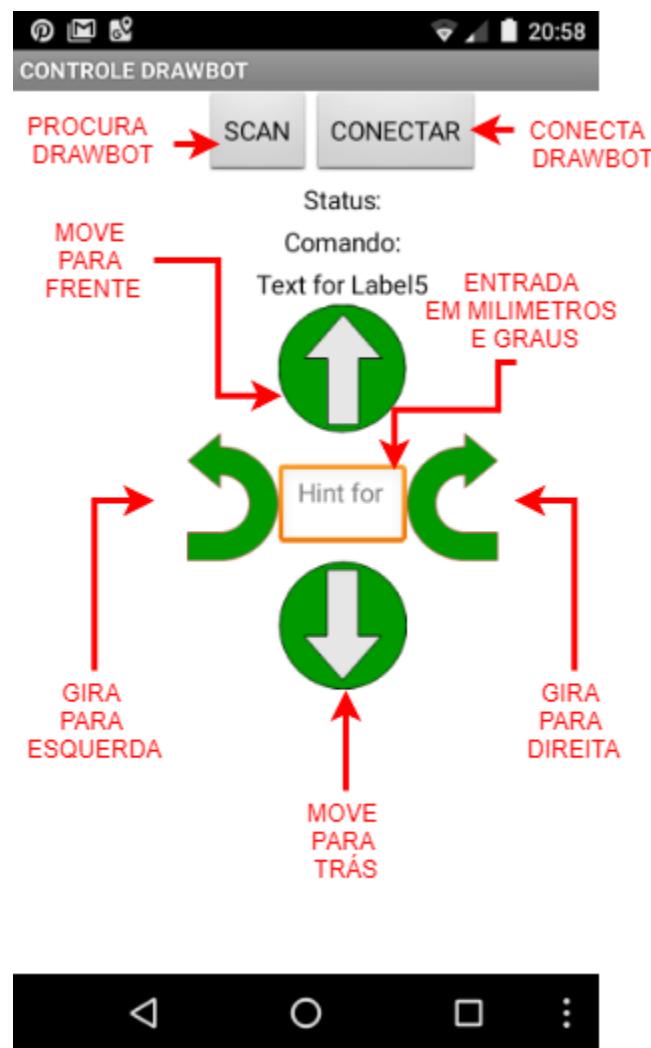


Figura 102: Visão geral do aplicativo.

## 5.2. Instalação e uso do aplicativo de controle do DrawBot no celular.



[Instalação e uso do aplicativo](https://youtu.be/S5bwerBPbTk) - <https://youtu.be/S5bwerBPbTk>

As etapas para a instalação devem ser realizadas na seguinte ordem.

1. Baixar o arquivo em:

<https://drive.google.com/open?id=1bqdilUkQSIZt4g-SGmepIYs4y7yW-INF>

2. Instalar o aplicativo no celular.

3. Ligar o DrawBot e apertar o botão scan (Figura 103) que tem a função de encontrar o DrawBot.

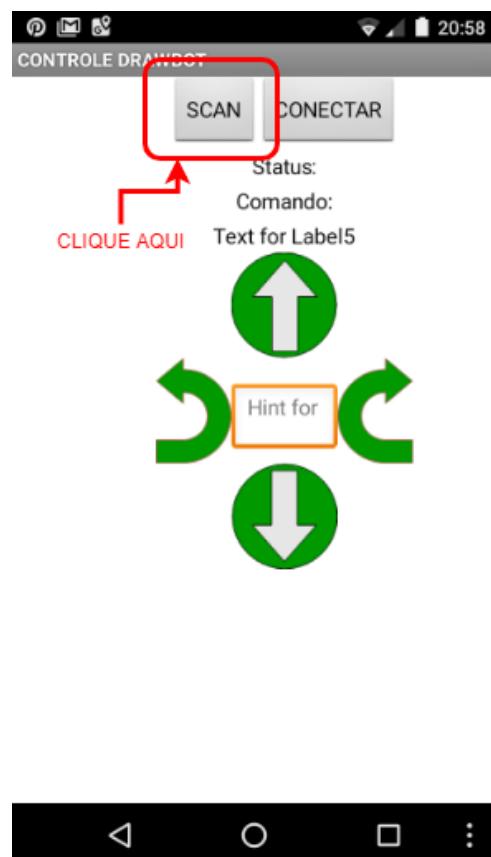


Figura 103: Botão SCAN.

4. Após apertar o botão SCAN (Figura 104) e será dada uma lista com diversos dispositivos e selecione o nome “DRAWBOT”.

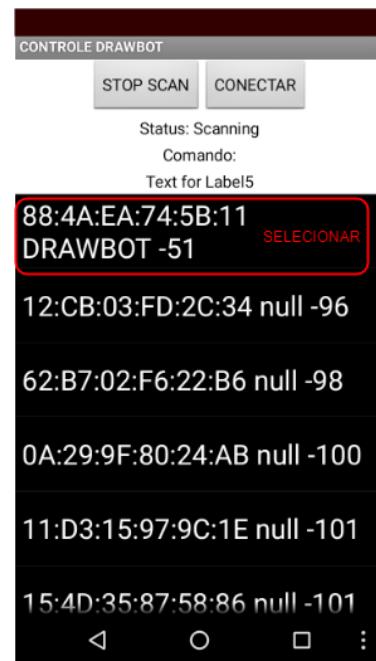


Figura 104: Lista de dispositivos

5. Apertar o botão conectar (Figura 105).

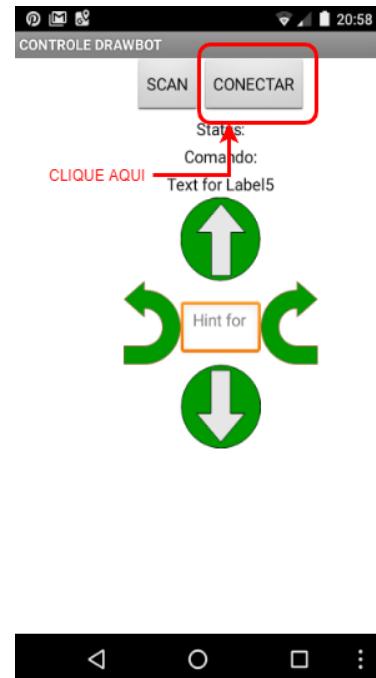


Figura 105: Apertar conectar.

6. Mover para frente o DrawBot em 50 milímetros (Figura 106).

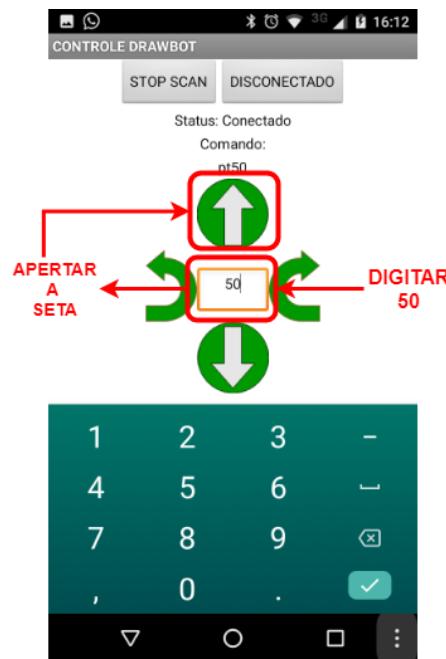


Figura 106: Procedimento para mover para frente o DrawBot.

7. Girar para esquerda o DrawBot em 90 graus (Figura 107).

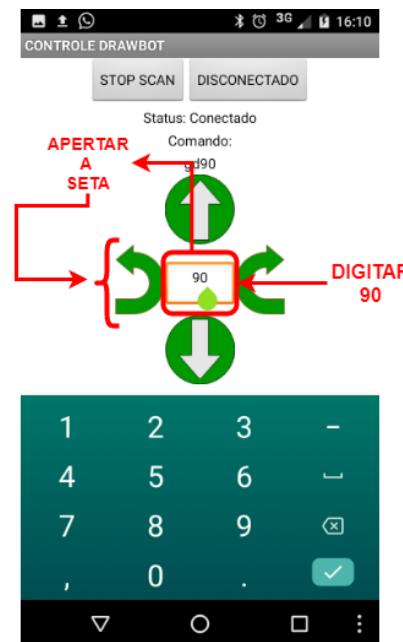


Figura 107: Procedimento para girar o DrawBot.

## 6. Descrição das atividades

A atividade estão definidas em dois momentos: a) modelagem de um som produzido pela fala de uma vogal, com o intuito de demonstrar como um fenômeno biológico pode ser matematizado em termos da soma da função seno e de caráter motivacional; b) interação como DrawBot por meio de aplicativo e tem com finalidade trabalhar conceitos de trigonometria.

### 6.1. Proposta de Atividade

#### 1. MOMENTO 1 - ATIVIDADE INTERATIVA COM A VOZ E ROBÓTICA: Parte 1.

A primeira etapa é ligar todo o equipamento e fazer os testes. Em seguida é escolhido um aluno do sexo masculino para gravar a voz no Octave. Para isso o estudante deve fazer o som de uma vogal de forma constante e nesse momento o professor aperta o botão executar o programa no Octave (Figura 108). Neste momento o gráfico é desenhado pelo DrawBot (Figura 109) de forma a despertar a curiosidade dos alunos. Em seguida serão gerados quatro gráficos e a soma de senos (Figura 110) que representa aproximadamente o som da vogal feito pelo aluno e O mesmo processo é repetido com a voz feminina e comparado os dados obtidos entre as duas vozes.

 [Vídeo Atividade interativa – parte 1](https://youtu.be/2NKadqs_o5c) [https://youtu.be/2NKadqs\\_o5c](https://youtu.be/2NKadqs_o5c)

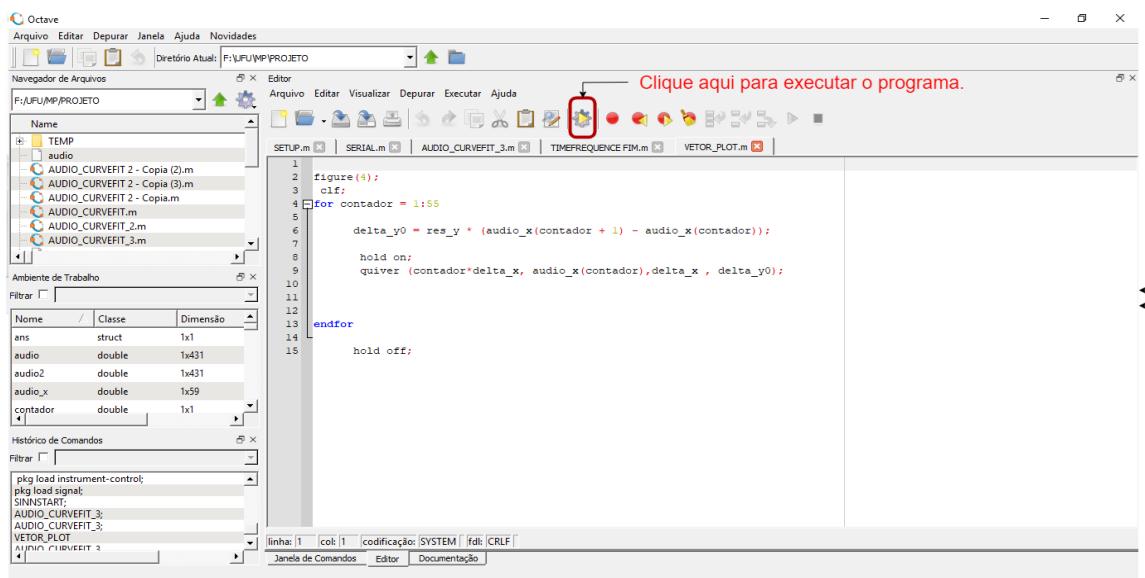


Figura 108: Botão para execução do programa.

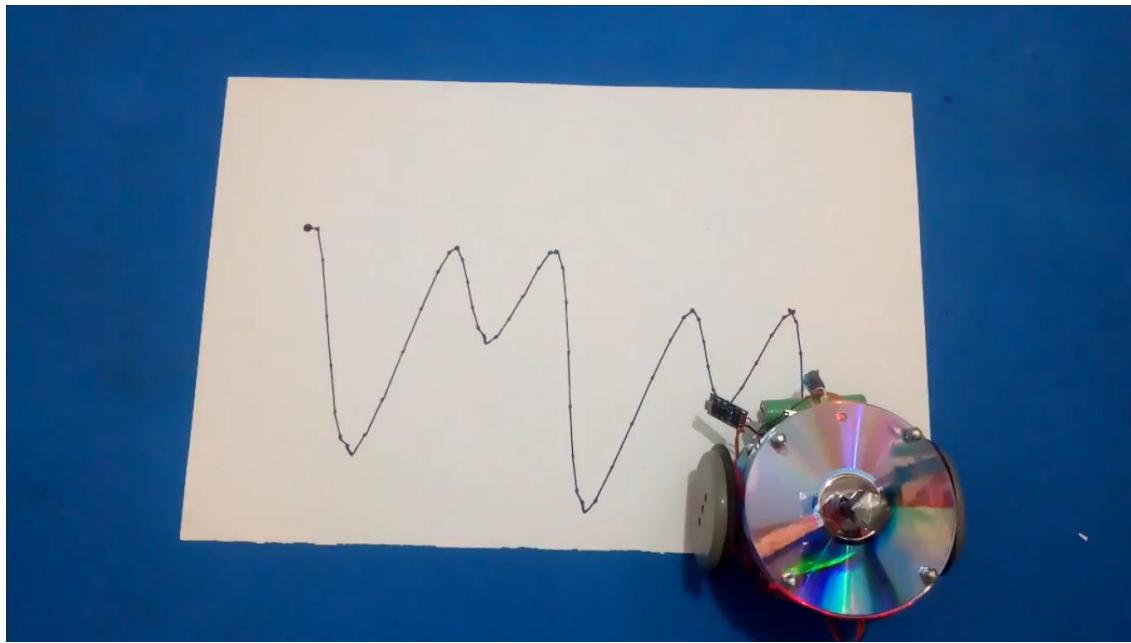


Figura 109: Desenho do gráfico feito pelo DrawBot.

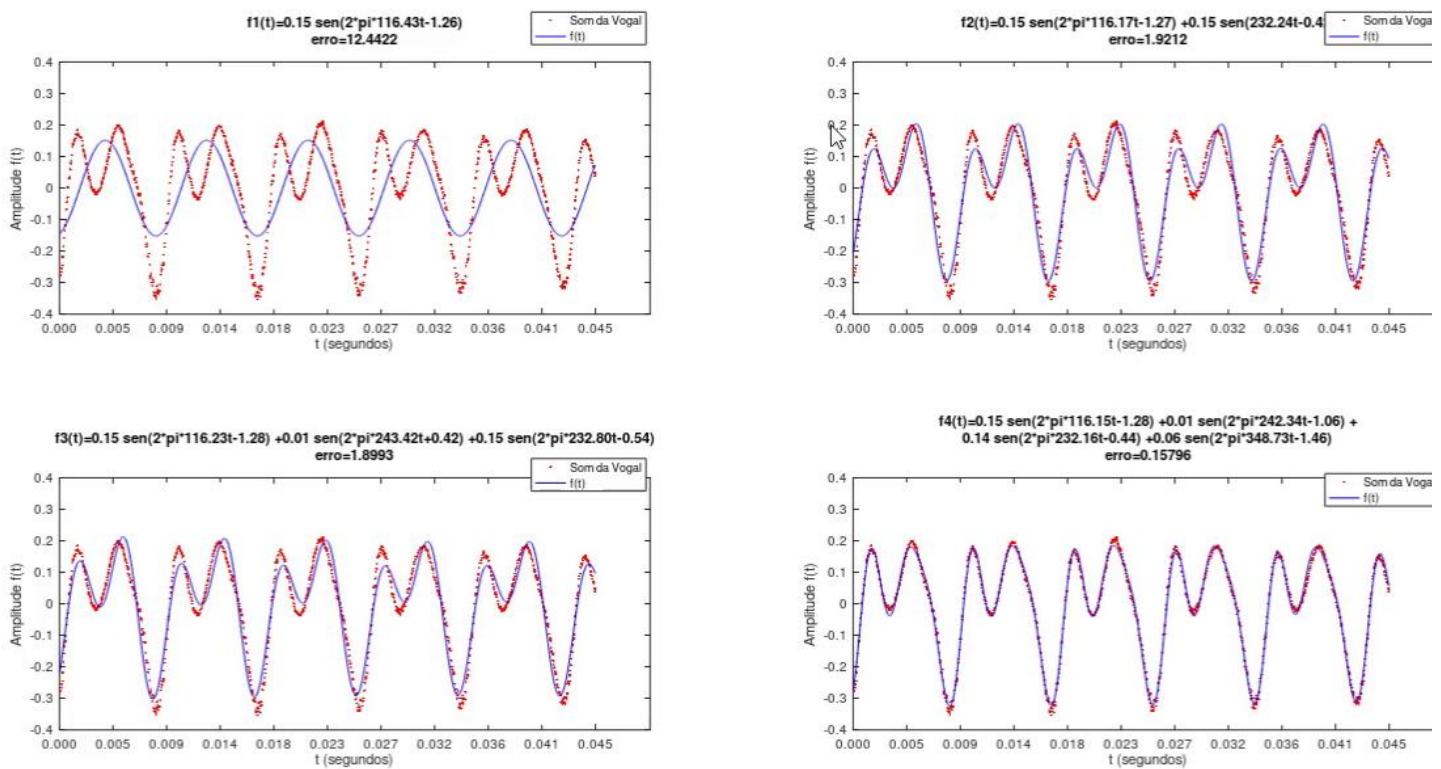


Figura 110: Gráficos e suas funções.

## **Sugestões de questões a serem abordadas na atividade:**

Questões:

Esse gráfico se parece com alguma função que você já estudou? Qual função?

Qual o ponto de máximo e mínimo da curva azul e vermelha? Qual a relação com a intensidade do som?

Quais as diferenças encontradas nas funções e nos gráficos da voz masculina e feminina em relação à amplitude? E ao período?

Qual a diferença entre o primeiro e último gráfico em relação ao número de termos da função  $f(t)$ ? A função  $f(t)$  se aproximou dos pontos vermelhos representados pela voz?

Compare as funções obtidas entre a voz masculina e feminina. Os valores dos coeficientes da voz masculina são maiores que os da voz feminina?

## MOMENTO 1 - ATIVIDADE INTERATIVA COM A VOZ E ROBÓTICA: Parte 2

Nesta atividade é efetuado é executado o arquivo “GRAFICO3D.m” (Figura 111). Em seguida será gerado um gráfico em 3 dimensões (Figura 112). Tem como objetivo demonstrar por meio de um gráfico 3d que o som de uma vogal pode ser expresso em uma soma de senos.



[Vídeo Atividade interativa parte 2](#)

The screenshot shows the Octave IDE interface. The left pane displays a file browser with files like 'audio\_x', 'DRAWBOT.m', 'GRAFICO 4 PLOT.m', 'GRAFICO3D.m' (highlighted with a red box), 'OCTAVE\_TO\_ARDUINO.m', 'OCTAVE\_TO\_ARDUINO\_TESTE.m', 'octave-workspace', and 'scratch.sb2'. The right pane shows the code editor with the 'GRAFICO3D.m' script. The script contains MATLAB code for generating a 3D plot of a waveform. Red boxes highlight the file name 'GRAFICO3D.m' in the browser and the file tab in the editor, and an arrow points to the F5 execute button in the toolbar above the editor. The code itself includes commands for creating a figure, plotting multiple sine waves, and labeling the axes.

```
figure(2);
plot3(1 +zeros(size(times)),times,ydata,'r.');
hold on;
plot3(2 +zeros(size(times)),times,termo4,'b');
plot3(3 +zeros(size(times)),times,t1,'b');
plot3(4 +zeros(size(times)),times,t2,'b');
plot3(5 +zeros(size(times)),times,t3,'b');
plot3(6 +zeros(size(times)),times,t4,'b');
grid on;
xlabel(' ', 'FontWeight', 'Bold');
ylabel(' t (segundos)', 'FontWeight', 'Bold');
zlabel(' Amplitude em Volts (V)', 'FontWeight', 'Bold');
view(-30,60);
box on;
```

Figura 111: Procedimento de geração do gráfico 3D.

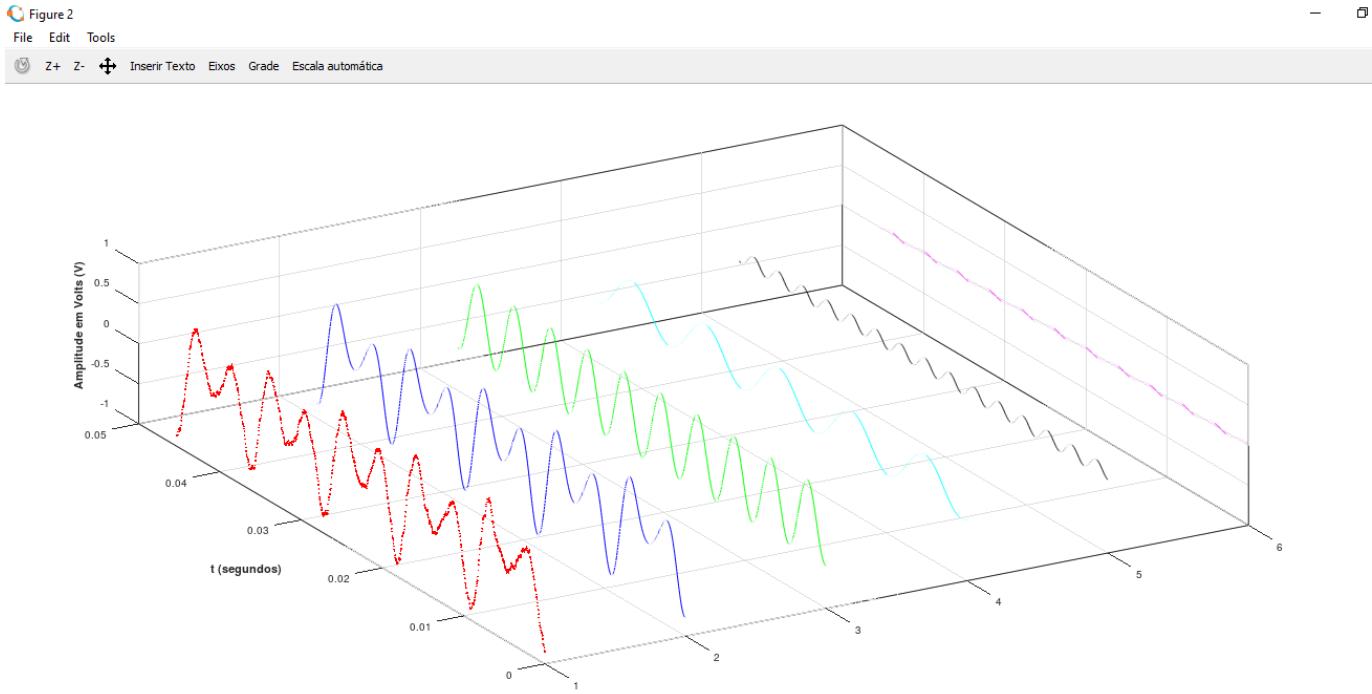
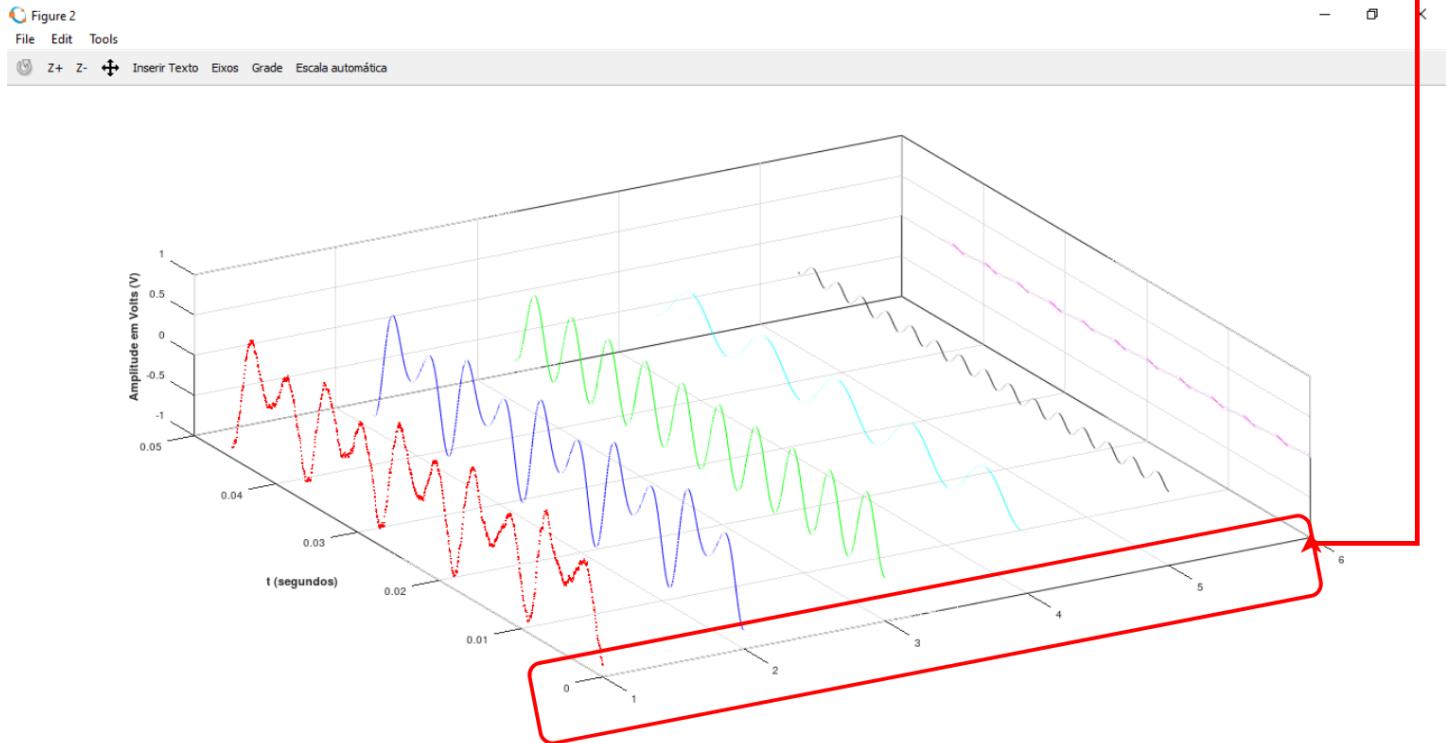
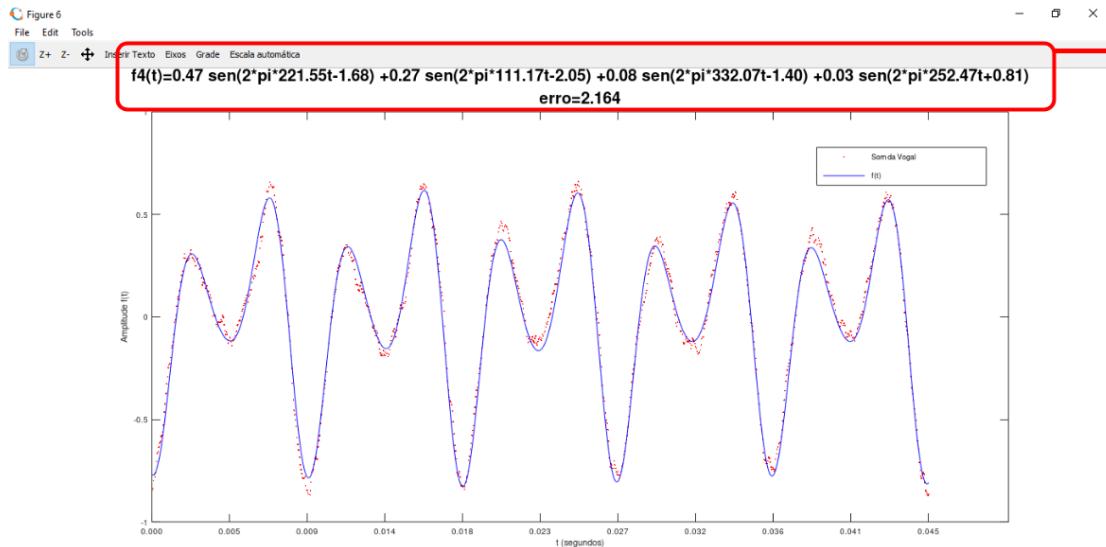


Figura 112: Geração do gráfico 3D.

## Sugestões de questões a serem abordadas na atividade.

Questões:

Relacione os dados e função  $f_4(t)$  com o gráfico em 3D. Escreva a função e os termos da mesma em frente aos números do gráfico em 3D?



## 2. MOMENTO 2 – INTERAÇÃO COM DRAWBOT.

Os alunos em grupos interagem com o DrawBot por meio do celular para conhecer os comandos de controle do mesmo como indicado no item 5 do sumário. Em seguida desenham a forma geométrica pedida.

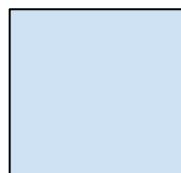


[Video interação com o Drawbot](https://youtu.be/X34Pt_7_6Lo) - [https://youtu.be/X34Pt\\_7\\_6Lo](https://youtu.be/X34Pt_7_6Lo)

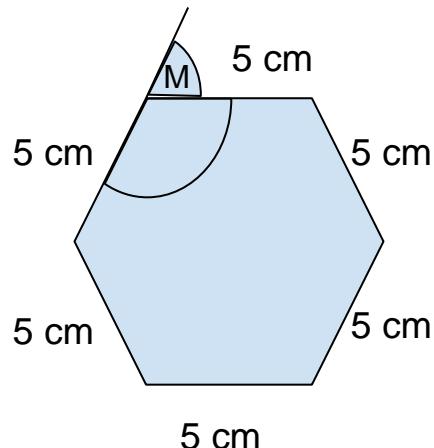
questões:

1. desenhe um quadrado de lado igual a 9 cm e descreva passo a passo os comandos utilizados.

9 cm



2. desenhe um polígono com 6 lados medindo 5 cm e descreva passo a passo os comandos utilizados.



Ângulo interno L:

$$L = \frac{(n - 2)180}{n}$$

Ângulo externo M:

$$M = \frac{360}{n}$$

n é número de lados

### 3. MOMENTO 3 – DRAWBOT E FUNÇÕES TRIGONOMÉTRICAS.

A atividade tem como objetivo aplicar as funções trigonométricas seno e cosseno com o uso do DrawBot a partir dos comandos apresentados no item 5 do sumário. O vídeo abaixo exemplifica a atividade:



[Vídeo Drawbot e funções trigonométricas](https://youtu.be/-TpaqEd0e5M) - <https://youtu.be/-TpaqEd0e5M>

TEMA 2 – *Drawbot e função trigonométrica*: atividade 4, duração 50 min.

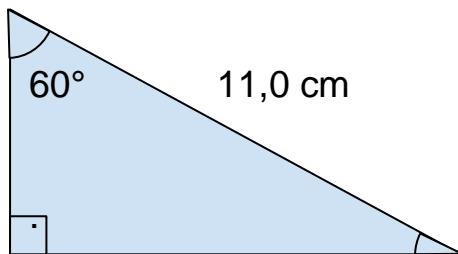
**Título:** formas geométricas e trigonometria.

**Objetivo:** aplicar as funções trigonométricas seno e cosseno com robótica.

**Recursos:** drawbot, smartphone com Android e cartolina.

**Questões:**

1. Desenhe o triângulo retângulo abaixo e descreva passo a passo os comandos utilizados.

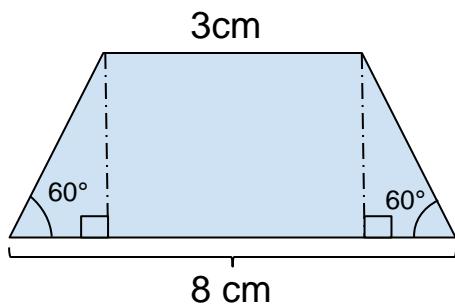


$$\text{Seno}(\alpha) = \frac{\text{medida do cateto oposto}}{\text{medida da hipotenusa}}$$

$$\text{Cosseno}(\alpha) = \frac{\text{medida do cateto adjacente}}{\text{medida da hipotenusa}}$$

$\alpha$  é o ângulo procurado

2. Desenhe o trapézio retângulo abaixo e descreva passo a passo os comandos utilizados.



$$\text{Seno}(\alpha) = \frac{\text{medida do cateto oposto}}{\text{medida da hipotenusa}}$$

$$\text{Cosseno}(\alpha) = \frac{\text{medida do cateto adjacente}}{\text{medida da hipotenusa}}$$

$\alpha$  é o ângulo procurado

#### 4. MOMENTO 4 – UNIÃO DO ANALÓGICO E DIGITAL.

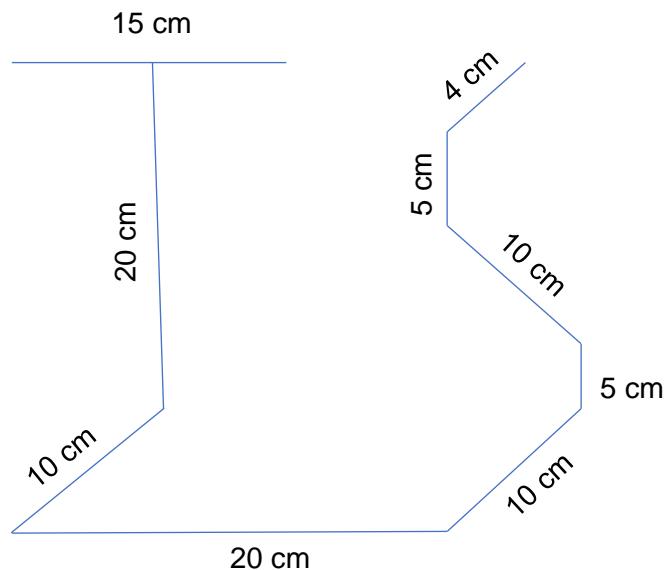
A atividade tem como objetivo a integração entre ferramentas analógicas e digitais. O desenho abaixo deve ser feito em uma cartolina sem as dimensões.



[Vídeo União do analógico e digital](https://youtu.be/u6m82Q2_e4k) - [https://youtu.be/u6m82Q2\\_e4k](https://youtu.be/u6m82Q2_e4k)

Questões:

1. Reproduza o desenho abaixo com o uso do Drawbot, régua e transferidor.



## ARQUIVO A - Código Octave SINNSTART.m

Para gerar o arquivo SINNSTART.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO
1;
function [x,y] = getxygrid(x,y)
% Determining the range of x values.
lengthx = length(x);
% Checking number of data points to be > 2
if lengthx < 2
    iTwoDataPointsRequiredError();
endif
% Sorting data points to be in order of increasing x.
diffx = diff(x);
if any(diffx < 0)
    [x,idx] = sort(x);
    y = y(idx);
    diffx = diff(x);
endif
% To avoid dividing by zero, we will get rid of repeated x entries.
tol = eps^0.7;
idx = [(diffx < tol); false];
idx2 = [false ; idx(1:end-1)];
x(idx) = (x(idx) + x(idx2)) / 2;
x(idx2) = [];
y(idx) = (y(idx) + y(idx2)) / 2;
y(idx2) = [];
lengthx = length(x);
% Data size too small, cannot find fit
if lengthx > 2
    % Checking to see whether the set of data points [x, y] are equally spaced
    % If idx has contains non-zero elements, then data points are scattered
    % Applying interpolation on the data points
    % if (sum(idx) > 0.0001)
    if all(abs(diffx-diffx(1)) < tol*max(diffx))
```

```

% [newx, newy] = interpolate1(x, y);
newx = linspace(min(x), max(x), numel(x));
newy = interp1(x, y, newx);
x = newx(:);
y = newy(:);

endif
endif

endfunction % getxygrid

function start = sinnstart(x,y,n)
[x,y] = getxygrid(x,y);

% Data size too small, cannot find starting values
if length(x) < 2
    start = rand(3*n,1);
    return;
endif

% Loop for sum of sines functions
start = zeros(3*n,1);
oldpeaks = [];
lengthx = length(x);
freqs = zeros(n,1);
res = y; % residuals from fit so far
for j=1:n
    % Apply fast fourier transform to the current residuals
    fy = fft(res); % don't subtract mean, no constant term
    fy(oldpeaks) = 0; % omit frequencies already used
    % Get starting value for frequency using fft peak
    [~,maxloc] = max(fy(1:floor(lengthx/2)));
    oldpeaks(end+1) = maxloc; %#ok<AGROW>
    w = 2*pi*(max(0.5,maxloc-1))/(x(end)-x(1));
    freqs(j) = w;
    % Compute Fourier terms using all frequencies we have so far
    X = zeros(lengthx,2*j);
    for k=1:j
        X(:,2*k-1) = sin(freqs(k)*x);
        X(:,2*k) = cos(freqs(k)*x);
    endfor
endfor

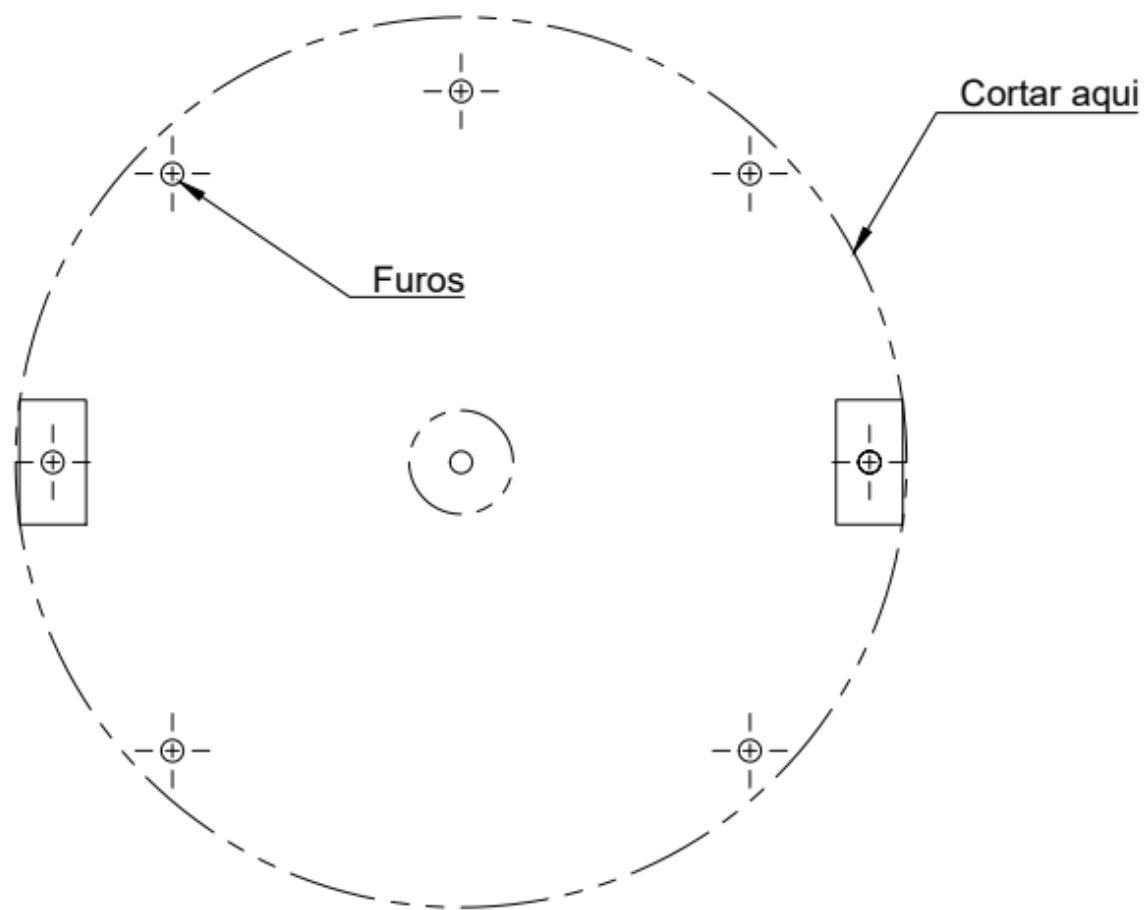
```

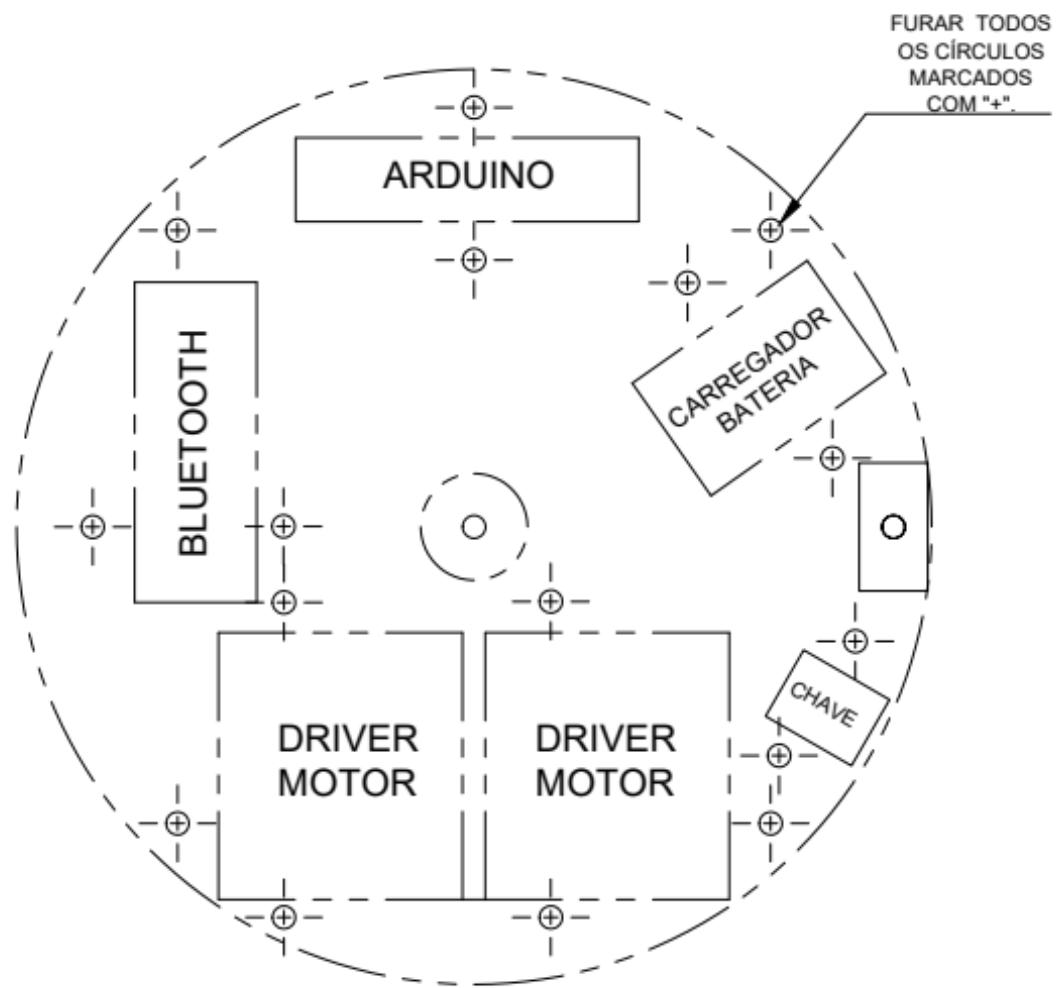
```

% Fit these terms to get the non-frequency starting values
ab = X \ y(:);
if j<n
    res = y - X*ab; % remove these components to get next frequency
endif
endfor
% All frequencies found, now compute starting values from all frequencies
% and the corresponding coefficients
for k=1:n
    start(3*k-2) = sqrt(ab(2*k-1)^2 + ab(2*k)^2);
    start(3*k-1) = freqs(k);
    start(3*k) = atan2(ab(2*k),ab(2*k-1));
endfor
endfunction % sinnstart
%FIM

```

GABARITO PARA PERFURAÇÃO DO CHASSI.





## ARQUIVO D - Código para gravação da placa Arduino

```
#include <CheapStepper.h>
#include <Servo.h>
#include <Math.h>

// CONFIGURAÇÃO DO MOTOR DE PASSO
CheapStepper stepper (8, 9, 10, 11);
CheapStepper stepper1 (4, 5, 6, 7);

boolean p, moveClockwise = true;
int i;
int contador;
int ledPin = 13;
int f;
String texto,cmd,valor;
float v;

void setup() {

    stepper.setRpm(22);
    stepper1.setRpm(22);
    stepper.setTotalSteps(4076);
    Serial.begin(9600);
    Serial.println("Pronto");
    Serial.setTimeout(100);
    contador = 1;
    p = 1;
    digitalWrite(13, 0);
}

void loop() {

f=0;
while ( Serial.available() > 0 ) {
    // LÊ PORTA SERIAL
    f=1;
    digitalWrite(13, 1);
    texto = Serial.readString();

    digitalWrite(13, 0);
}
}
```

```

if(f==1){
    cmd=texto.substring(0,2);
    valor=texto.substring(2);
    v=valor.toFloat();

    if(cmd.equalsIgnoreCase("gd")){
        digitalWrite(13, 1);
        gira_graus(1,v);
        digitalWrite(13, 0);
    }
    else if(cmd.equalsIgnoreCase("ge")){
        digitalWrite(13, 1);
        gira_graus (0,v);
        digitalWrite(13, 0);
    }
    else if(cmd.equalsIgnoreCase("pf")){
        digitalWrite(13, 1);
        Move_milimetros(1, v);
        digitalWrite(13, 0);
    }
    else if(cmd.equalsIgnoreCase("pt")){
        digitalWrite(13, 1);
        move_milimetros(0, v);
        digitalWrite(13, 0);
    }
    else{
        Serial.println("erro");
    }
    texto="";
    cmd="";
    valor="";
}
}

void gira_graus(bool clockwise, float deg) {
    float temp;
    temp=round(deg);

```

```

int nSteps = round(deg *16.9); // 16,9 = (14,35*424)/360
                                // 16,9 passos para cada grau
for (int n = 0; n < nSteps; n++) {
    stepper.step(!clockwise);
    stepper1.step(clockwise);
}
Serial.println(temp);
temp=0;

digitalWrite(4, 0);
digitalWrite(5, 0);
digitalWrite(6, 0);
digitalWrite(7, 0);
digitalWrite(8, 0);
digitalWrite(9, 0);
digitalWrite(10, 0);
digitalWrite(11, 0);
}

void move_milimetros(bool clockwise, float mil) {

    int nSteps = round(mil*14.35); // 14,35=4076/284
                                    // 14,35 passos para milímetro
    for (int n = 0; n < nSteps; n++) {
        stepper.step(clockwise);
        stepper1.step(clockwise);
    }
    digitalWrite(4, 0);
    digitalWrite(5, 0);
    digitalWrite(6, 0);
    digitalWrite(7, 0);
    digitalWrite(8, 0);
    digitalWrite(9, 0);
    digitalWrite(10, 0);
    digitalWrite(11, 0);
    Serial.println(mil);
}

```

## ARQUIVO E - Código Octave SETUP.m

Este inicia as bibliotecas necessárias para o funcionamento do programa.

Para gerar o arquivo SETUP.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO
%SETUP.m
%BIBLIOTECAS E ARQUIVOS DE INICIALIZAÇÃO
graphics_toolkit('qt'); % Biblioteca gráfica
pkg load optim; % Biblioteca que possui a função do método dos mínimos quadrados
pkg load instrument-control; % Biblioteca para comunicação serial
pkg load signal; % Biblioteca para captura do som
SINNSTART;
AJUSTEDECURVA_CALCULO;
AJUSTEDECURVA_PLOT;
%FIM
```

## ARQUIVO F - Código Octave SERIAL.m

Este algoritmo conecta o Octave ao Arduino via porta serial.

Para gerar o arquivo SERIAL.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO
%SERIAL.m
1;
%VERIFICA SE PORTA SERIAL ESTÁ DISPONÍVEL
if (exist("serial") != 3)
    disp("Porta serial sem suporte");
endif

%CONECTA A PORTA SERIAL
s1 = serial("\.\.\COM3") % Abre a porta serial
pause(1);
set(s1, 'baudrate', 9600); % See List Below
set(s1, 'bytesize', 8); % 5, 6, 7 or 8
set(s1, 'parity', 'n'); % 'n' or 'y'
set(s1, 'stopbits', 1); % 1 or 2
set(s1, 'timeout', 5); %12.3 Seconds as an example here
set(s1, 'requesttosend', 'off'); % Sets the RTS line
set(s1, 'dataterminalready', 'off'); % Sets the DTR line
srl_flush(s1);

%CONECTA AO MÓDULO BLUETOOTH
pause(2);
srl_write(s1, "AT+NAME\r\n");
```

```

data = srl_read(s1, 100) ;
char(data)
srl_write(s1, "AT+INQ\r\n");
data = srl_read(s1, 50) ;

char(data)
pause(3);
srl_write(s1, "AT+CONN1\r\n");
data = srl_read(s1, 100) ;
char(data)
srl_write(s1, "gd90");
%FIM

```

## **ARQUIVO G – Código Octave AJUSTECECURVA\_CALCULO.m**

Este algoritmo faz o ajuste de curvas.

Para gerar o arquivo AJUSTEDECURVA\_CALCULO.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```

%INICIO
%AJUSTECECURVA_CALCULO.m
1;
%INICIO - GRAVAÇÃO DA VOZ
recObj = audiorecorder (22050, 16, 1);
get (recObj);
disp('Start speaking.')
recordblocking(recObj, 1);
disp("End of Recording.")
myRecording = getaudiodata(recObj);
%FIM - GRAVAÇÃO DA VOZ

%DEFINE DADOS DO EIXO X E Y

xdata =[0:1/22050:1000*1/22050]';
ydata=myRecording(2000:3000,1);

%APLICA AOS DADOS O MÉTODO DOS MÍNIMOS QUADRADOS PARA 1 TERMO DE
SENO
fun1 = @(p1,xdata) p1(1)*sin(p1(2)*xdata+p1(3));
pguess1 = sinnstart(xdata,ydata,1);
[p1,fminres1] = lsqcurvefit(fun1,pguess1,xdata,ydata);

%APLICA AOS DADOS O MÉTODO DOS MÍNIMOS QUADRADOS PARA 2 TERMOS SENO
fun2 = @(p2,xdata) p2(1)*sin(p2(2)*xdata + p2(3)) + p2(4)*sin(p2(5)*xdata + p2(6));

```

```
pguess2 = sinnstart(xdata,ydata,2);
[p2,fminres2] = lsqcurvefit(fun2,pguess2,xdata,ydata);
```

```
%APLICA AOS DADOS O MÉTODO DOS MÍNIMOS QUADRADOS PARA 3 TERMOS SENO
fun3 = @(p3,xdata) p3(1)*sin(p3(2)*xdata + p3(3)) + p3(4)*sin(p3(5)*xdata + p3(6)) +
p3(7)*sin(p3(8)*xdata + p3(9));
pguess3 = sinnstart(xdata,ydata,3);
[p3,fminres3] = lsqcurvefit(fun3,pguess3,xdata,ydata);
```

```
%APLICA AOS DADOS O MÉTODO DOS MÍNIMOS QUADRADOS PARA 4 TERMOS SENO
fun4 = @(p4,xdata) p4(1)*sin(p4(2)*xdata + p4(3)) + p4(4)*sin(p4(5)*xdata + p4(6)) +
p4(7)*sin(p4(8)*xdata + p4(9)) + p4(10)*sin(p4(11)*xdata + p4(12));
pguess4 = sinnstart(xdata,ydata,4);
[p4,fminres4] = lsqcurvefit(fun4,pguess4,xdata,ydata);
```

**%FUNÇÕES GERADAS A PARTIR DOS COEFICIENTES ENCONTRADOS PELO METODO DOS %MINIMOS QUADRADOS**

```
f1= p1(1)*sin((p1(2))*xdata + p1(3));
f2= p2(1)*sin(p2(2)*xdata + p2(3))+p2(4)*sin(p2(5)*xdata + p2(6));
f3= p3(1)*sin(p3(2)*xdata + p3(3)) + p3(4)*sin(p3(5)*xdata + p3(6)) + p3(7)*sin(p3(8)*xdata +
p3(9));
f4= p4(1)*sin(p4(2)*xdata + p4(3)) + p4(4)*sin(p4(5)*xdata + p4(6)) + p4(7)*sin(p4(8)*xdata +
p4(9)) + p4(10)*sin(p4(11)*xdata + p4(12));

t1=p4(1)*sin(p4(2)*xdata + p4(3));
t2=p4(4)*sin(p4(5)*xdata + p4(6));
t3=p4(7)*sin(p4(8)*xdata + p4(9));
t4=p4(10)*sin(p4(11)*xdata + p4(12));
```

```
%GERAÇÃO DOS DADOS PARA SEREM ENVIADOS AO DRAWBOT
values=0:1/22050:430/22050;
audio = fun4(p4,values);
```

**%AJUSTE DE GANHO**

```
audio2=audio*(80/max(audio));
```

```
save audio audio;
save values values;
load 'audio';
load 'values';
[p,q]=rat(3000/22050,0.0001);
```

**%AJUSTE DE ESCALA**

```
audio_x = resample(audio2,p,q); %biblioteca pkg load signalsave
%audio_x(1,1)=audio_x(1,3);
```

```
%audio_x(1,2)=audio_x(1,3);  
%audio_x(1,3)=audio_x(1,3);
```

#### %SALVA DADOS

```
save audio_x audio_x;  
load 'audio_x';  
  
player = audioplayer (audio, 22050, 16);  
play(player);  
  
%plot(audio_x)  
%FIM
```

### ARQUIVO H – Código Octave AJUSTEDECURVA\_PLOT.m

Este algoritmo gera os gráficos pelo método de ajuste de curvas.

Para gerar o arquivo AJUSTEDECURVA\_PLOT.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO  
%AJUSTEDECURVA_PLOT.m  
%GERAÇÃO DOS GRÁFICOS  
times = xdata;  
%title('Pontos e Curva Ajustada');  
hf=figure(1,"position",get(0,"screensize"));  
  
%GRÁFICO DA FUNÇÃO f1 COM 1 TERMO  
%hf=figure(1);  
subplot(2,2,1);  
  
plot(times,ydata,'r.',times,f1,'b-');  
h=legend('Som da Vogal','f(t)', "location", "northeast");  
set(gca,'XTick',0:100/22050:1000/22050);  
xtick = get (gca, "XTick");  
xticklabel = strsplit (sprintf ("%3f\n", xtick), "\n", true);  
set (gca, "xticklabel", xticklabel);  
  
title({strcat('f1(t)= ',num2str(p1(1),"%.2f"),'  
sen(2*pi*',num2str(p1(2)/(2*pi),"%.2f"),'t,num2str(p1(3),"%.2f"),')'),strcat("erro=  
",num2str(fminres1))))};  
xlabel ("t (segundos)");  
ylabel ("Amplitude f(t)");
```

```

set(gca,'XTick',0:100/22050:1000/22050);
xtick = get (gca, "XTick");
xticklabel = strsplit (sprintf ("%3f\n", xtick), "\n", true);
set (gca, "xticklabel", xticklabel);

```

### %GRÁFICO DA FUNÇÃO f2 COM 2 TERMOS

```

subplot(2,2,2);
plot(times,ydata,'r.',times,f2,'b-');
h=legend('Som da Vogal','f(t)',"location", "northeast");
title({strcat('f2(t)= ',num2str(p2(1),"%.2f"),' 
sen(2*pi*',num2str(p2(2)/(2*pi),"%.2f"),'t',num2str(p2(3),"%+.2f"),') +' ,num2str(p2(4),"%.2f"),
sen(' ,num2str(p2(5)/(2*pi),"%.2f"),'t',num2str(p2(6),"%+.2f"),')'),strcat("erro=
",num2str(fminres2)))});

```

```

xlabel ("t (segundos)");
ylabel ("Amplitude f(t)");

```

```

set(gca,'XTick',0:100/22050:1000/22050);
xtick = get (gca, "XTick");
xticklabel = strsplit (sprintf ("%3f\n", xtick), "\n", true);
set (gca, "xticklabel", xticklabel);

```

### %GRÁFICO DA FUNÇÃO f3 COM 3 TERMOS

```

subplot(2,2,3);
plot(times,ydata,'r.',times,f3,'b-');
title({strcat('f3(t)= ',num2str(p3(1),"%.2f"),' 
sen(2*pi*',num2str(p3(2)/(2*pi),"%.2f"),'t',num2str(p3(3),"%+.2f"),') +' ...
,num2str(p3(4),"%.2f"),'
sen(2*pi*',num2str(p3(5)/(2*pi),"%.2f"),'t',num2str(p3(6),"%+.2f"),') +' ...
,num2str(p3(7),"%.2f"),'
sen(2*pi*',num2str(p3(8)/(2*pi),"%.2f"),'t',num2str(p3(9),"%+.2f"),')'...
),strcat("erro= ",num2str(fminres3)))});

```

```

h=legend('Som da Vogal','f(t)',"location", "northeast");
xlabel ("t (segundos)");
ylabel ("Amplitude f(t)");

```

```

set(gca,'XTick',0:100/22050:1000/22050);
xtick = get (gca, "XTick");
xticklabel = strsplit (sprintf ("%3f\n", xtick), "\n", true);
set (gca, "xticklabel", xticklabel);

```

```

%GRÁFICO DA FUNÇÃO f4 COM 4 TERMO
subplot(2,2,4);
plot(times,ydata,'r.',times,f4,'b-');

title({strcat('f4(t)= ',num2str(p4(1),"%.2f"),'
sen(2*pi*',num2str(p4(2)/(2*pi),"%.2f"),'t',num2str(p4(3),"%.2f"),') +'...
,num2str(p4(4),"%.2f"),'
sen(2*pi*',num2str(p4(5)/(2*pi),"%.2f"),'t',num2str(p4(6),"%.2f"),') ...
+'),strcat(num2str(p4(7),"%.2f"),'
sen(2*pi*',num2str(p4(8)/(2*pi),"%.2f"),'t',num2str(p4(9),"%.2f"),') +'...
,num2str(p4(10),"%.2f"),'
sen(2*pi*',num2str(p4(11)/(2*pi),"%.2f"),'t',num2str(p4(12),"%.2f"),')...
),strcat("erro= ",num2str(fminres4)))});

h=legend('Som da Vogal','f(t)',"location", "northeast");
xlabel ("t (segundos)");
ylabel ("Amplitude f(t)");

set(gca,'XTick',0:100/22050:1000/22050);
xtick = get (gca, "XTick");
xticklabel = strsplit (sprintf ("%3f\n", xtick), "\n", true);
set (gca, "xticklabel", xticklabel);
%FIM

```

## ARQUIVO I - Código Octave DRAWBOT.m

Para gerar o arquivo DRAWBOT.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```

%INICIO
%DRAWBOT.m
%DECLARAÇÃO DAS VARIÁVEIS
load audio_x;
datxx="";
datyy=0;
delta_x = 6;
delta_y0 = 0;
delta_y1 = 0;
modulo_v0 = 0;
modulo_v1 = 0;
angulo_0 = 0;
angulo_1 = 0;
res_y = 1;
debug="";
count=0;
data=0;
%FUNÇÃO DE ATRASO

```

```

function delay_retorno(s2)
    count=0;
    data=0;
    tempo_inicial=time();
    srl_flush(s2);
    while(count<4)
        [data, count] = srl_read (s2,4);
        tempo_final=time()-tempo_inicial;
        if(tempo_final>8)
            break;
        endif
    endwhile

    disp(char(data))
    count=0;
    data=0;
endfunction

```

#### %AQUI DEFINE O DESENHO DE 55 PONTOS

```

for contador = 1:55
    delta_y0 = res_y * (audio_x(contador + 1) - audio_x(contador));
    delta_y1 = res_y * (audio_x(contador + 2) - audio_x(contador + 1));
    modulo_v0 = sqrt(delta_x * delta_x + delta_y0 * delta_y0);
    modulo_v1 = sqrt(delta_x * delta_x + delta_y1 * delta_y1);
    angulo_0 = asin(delta_y0 / modulo_v0)*57.29577;
    angulo_1 = asin(delta_y1 / modulo_v1)*57.29577;

    teta= angulo_0 - angulo_1;

```

```

if (teta > 0)
```

```

    strcat( "ge",num2str(teta,"%05.2f"))
    srl_write(s1, strcat( "gd",num2str(teta,"%05.2f")));
    delay_retorno(s1);
    srl_write(s1, strcat( "pt",num2str(modulo_v1,"%05.2f")));
    delay_retorno(s1);

```

```

endif
```

```

if (teta < 0)
```

```
strcat( "gd",num2str(-teta,"%05.2f"))
srl_write(s1,strcat( "ge",num2str(-teta,"%05.2f")));
delay_retorno(s1);
srl_write(s1,strcat( "pt",num2str(modulo_v1,"%05.2f")));
delay_retorno(s1);

endif

if (teta == 0)

strcat( "pt",num2str(modulo_v1,"%05.2f"))
srl_write(s1,strcat( "pt",num2str(modulo_v1,"%05.2f")));
delay_retorno(s1);

endif

%pause(1);

endfor
%FIM
```

## ARQUIVO J - Código Octave GRAFICO3D.m

Para gerar o arquivo GRAFICO3D.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO
%GRAFICO3D.m
figure(2);

plot3(1 +zeros(size(times)),times,ydata,'r.');
hold on;
plot3(2 +zeros(size(times)),times,f4,'b');
plot3(3 +zeros(size(times)),times,t1,'g');
plot3(4 +zeros(size(times)),times,t2,'c');
plot3(5 +zeros(size(times)),times,t3,'k');
plot3(6 +zeros(size(times)),times,t4,'m');

grid on;

xlabel(' ','FontWeight','Bold');
ylabel(' t (segundos)','FontWeight','Bold');
zlabel(' Amplitude em Volts (V)','FontWeight','Bold');
view(-30,60);
box on;
%FIM
```

## ARQUIVO K - Código Octave GRAFICOF4.m

Para gerar o arquivo GRAFICOF4.m, copiar o código delimitado por “%INICIO” e “%FIM”.

```
%INICIO
%GRAFICOF4.m
figure(6);
plot(times,ydata,'r.',times,termo4,'b-');

title({strcat('f4(t)= ',num2str(p4(1),"%.2f"),'
sen(2*pi',num2str(p4(2)/(2*pi),"%.2f"),'t',num2str(p4(3),"%.2f"),') +'...
,num2str(p4(4),"%.2f"),'
sen(2*pi',num2str(p4(5)/(2*pi),"%.2f"),'t',num2str(p4(6),"%.2f"),') +' ,num2str(p4(7),"%.2f"),'
sen(2*pi',num2str(p4(8)/(2*pi),"%.2f"),'t',num2str(p4(9),"%.2f"),') +' ,num2str(p4(10),"%.2f"),'
sen(2*pi',num2str(p4(11)/(2*pi),"%.2f"),'t',num2str(p4(12),"%.2f"),')...
),strcat("erro= ",num2str(fminres4))}, "fontsize", 20);

h=legend('Som da Vogal','f(t)',"location", "northeast");
xlabel ("t (segundos)");
```

```
ylabel ("Amplitude f(t)");

set(gca,'XTick',0:100/22050:1000/22050);
xtick = get (gca, "XTick");
xticklabel = strsplit (sprintf ("% .3f\n", xtick), "\n", true);
set (gca, "xticklabel", xticklabel);
%FIM
```