

Universidade Federal do Triângulo Mineiro

Felipe Nascimento Cerqueira Rodrigues

Sistema com aplicativo híbrido para a automação do estágio XY do porta-amostras do  
equipamento de plasma atmosférico

Uberaba – MG

2021

Felipe Nascimento Cerqueira Rodrigues

Sistema com aplicativo híbrido para a automação do estágio XY do porta-amostras do equipamento de plasma atmosférico

Dissertação apresentada ao Programa de Mestrado Profissional em Inovação Tecnológica da Universidade Federal do Triângulo Mineiro como requisito para obtenção do título de mestre.

Orientador: Prof. Dr. Rogério Valentim Gelamo.

UBERABA – MG

2021

**Catálogo na fonte: Biblioteca da Universidade Federal do  
Triângulo Mineiro**

R613s Rodrigues, Felipe Nascimento Cerqueira  
Sistema com aplicativo híbrido para a automação do estágio XY do  
porta-amostras do equipamento de plasma atmosférico / Felipe Nascimento  
Cerqueira Rodrigues. -- 2021.  
84 f. : il., tab.

Dissertação (Mestrado Profissional em Inovação Tecnológica) -- Uni-  
versidade Federal do Triângulo Mineiro, Uberaba, MG, 2021  
Orientador: Prof. Dr. Rogério Valentim Gelamo

1. Aplicativos móveis. 2. Ionic (Programa). 3. Arduino (Programa).  
4. Descarga de barreira dielétrica. I. Gelamo, Rogério Valentim. II. Uni-  
versidade Federal do Triângulo Mineiro. III. Título.

CDU 004.42

**FELIPE NASCIMENTO CERQUEIRA RODRIGUES**

**SISTEMA COM APLICATIVO HÍBRIDO PARA A AUTOMAÇÃO DO ESTÁGIO XY DO PORTA-AMOSTRAS DO EQUIPAMENTO DE PLASMA ATMOSFÉRICO**

Dissertação apresentada ao Programa de Pós-Graduação Profissional em Inovação Tecnológica da Universidade Federal do Triângulo Mineiro como requisito parcial para obtenção do título de mestre.

Uberaba, 08 de fevereiro de 2021

**Banca Examinadora:**

Prof. Dr. Rogério Valentim Gelamo – Orientador  
Universidade Federal do Triângulo Mineiro

Prof. Dr. Vinicius Abrão da Silva Marques  
Universidade Federal do Triângulo Mineiro

Prof. Dr. Gustavo Finholdt  
Instituto Federal do Triângulo Mineiro



Documento assinado eletronicamente por **ROGERIO VALENTIM GELAMO**, Professor do Magistério Superior, em 08/02/2021, às 15:19, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#) e no art. 14 da [Resolução nº 34, de 28 de dezembro de 2017](#).

Documento assinado eletronicamente por **VINICIUS ABRAO DA SILVA MARQUES**, Professor do Magistério Superior, em 08/02/2021, às 16:10, conforme horário oficial de Brasília, com fundamento no art.

12/02/2021

SEI/UFTM - 0468240 - Folha de Aprovação



6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#) e no art. 14 da [Resolução nº 34, de 28 de dezembro de 2017](#).



Documento assinado eletronicamente por **Gustavo Finholdt**, Usuário **Externo**, em 12/02/2021, às 10:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#) e no art. 14 da [Resolução nº 34, de 28 de dezembro de 2017](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufmt.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufmt.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0468240** e o código CRC **1C0D94B4**.

## **AGRADECIMENTOS**

Agradeço a Deus pela proteção, benção e por ter me dado essa oportunidade de realizar este trabalho.

Aos meus pais, Adriana e Marcelo; ao meu irmão, Gabriel; e a minha esposa, Carolina, por todo apoio, paciência, carinho e amor.

Ao professor doutor Rogério Valetim Gelamo que aceitou o convite de ser o orientador deste trabalho e por auxiliar no desenvolvimento do mesmo.

Ao discente Lúcio Rogério Junior e ao professor doutor Rogério Valetim Gelamo que desenvolveram o porta-amostras do equipamento de plasma atmosférico e me permitiram o aprimoramento do equipamento.

À Universidade Federal do Triângulo Mineiro, ao Programa de Mestrado Profissional em Inovação Tecnológica, ao departamento de Engenharia Mecânica, ao Laboratório de Filmes Finos e Processos a Plasma e à todos docentes e técnicos que contribuíram para minha formação.

“Não existe limites para aqueles que ousam cruzar a linha do horizonte”.

Carolina Miziara

## RESUMO

A automação substitui o trabalho humano, controlando os processos operacionais por meio de dispositivos mecânicos ou eletrônicos, garantindo-os maior segurança e precisão. Propõe-se a automação do estágio XY do porta-amostras do equipamento de plasma atmosférico do Laboratório de Filmes Finos e Processos a Plasma da Universidade Federal do Triângulo Mineiro, visto que, a manipulação deste equipamento pode causar choque elétrico ao usuário. Pode-se dividir este trabalho em duas etapas: o desenvolvimento do *hardware* e o do *software*. O *hardware* é composto por suportes e acopladores para a fixação e transmissão de movimento dos motores de passo para a mesa do equipamento. Também é composto pelas ligações eletrônicas com o Arduino®, os motores de passo, *drivers* de acionamento, *buzzer* e o módulo *Bluetooth*. Para o *software*, desenvolveu-se dois algoritmos, o primeiro é o programa elaborado no Arduino® para que o microcontrolador execute os movimentos desejados; e o segundo é o aplicativo desenvolvido em Ionic® que envia informações do formulário para o Arduino® através da comunicação *Bluetooth*. O desenvolvimento proposto resulta no seguinte processo: preenchimento do formulário com os deslocamentos e velocidades desejadas pelo usuário no aplicativo. Em seguida, o aplicativo envia essas informações para o Arduino® através da comunicação *Bluetooth* e, logo após o aviso sonoro de início de operação, o microcontrolador executa o movimento com o acionamento dos motores de passo e finaliza com o aviso sonoro. O investimento para a construção desse sistema foi de R\$191,34 e ele é considerado inovador, pois apresenta uma proposta inédita para a solução de acidentes envolvendo choque elétrico no porta-amostras do equipamento de plasma atmosférico. Além disso, ambos os softwares se classificam no registro de *software* do Instituto Nacional da Propriedade Industrial.

Palavras-chave: Aplicativo *mobile*, Ionic®, Arduino®, Descarga de barreira dielétrica (DBD).

## **ABSTRACT**

*Automation replaces human work, controlling operational processes through mechanical or electronic devices, ensuring greater safety and accuracy. It is proposed to automate the XY stage of the sample holder of the atmospheric plasma equipment of the Laboratory of Thin Films and Plasma Processes at the Federal University of Triângulo Mineiro, since the handling of this equipment can cause electric shock to the user. This work can be divided into two stages: the development of hardware and software. The hardware consists of supports and couplers for fixing and transmitting motion from the stepper motors to an equipment table. It also consists of electronic connections with Arduino©, stepper motors, drivers motor, buzzer and the Bluetooth module. For the software, two algorithms were developed, the first is the program developed in Arduino© for the microcontroller to execute the desired movements; and the second is the application developed in Ionic© that sends form information to Arduino© via Bluetooth communication. The proposed development results in the following process: filling in the form with the displacements and speeds desired by the user in the application. Then, the application sends this information to the Arduino© via Bluetooth communication and, immediately after the audible warning of the start of operation, the microcontroller executes the movement with the activation of the stepper motors and ends with the audible warning. The investment for the construction of this system was R\$ 191.34 and it is considered innovative because it presents an unprecedented proposal for the solution of accidents involving electric shock in the sample holder of the atmospheric plasma equipment. In addition, each software is classified in the software registry of the National Institute of Industrial Property.*

*Key words: Mobile application, Ionic©, Arduino©, Dielectric barrier discharge (DBD).*

## LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem de uma placa comercial da Plataforma do Arduino® Uno para demonstração da pinagem e componentes eletrônicos. ....	19
Figura 2 – Imagem de um motor de passo Nema 23 aberto para demonstração de seus constituintes internos, tais como rotor e estator. ....	21
Figura 3 – Imagem ilustrativa representando as possíveis ligações do motor de passo unipolar e bipolar. ....	22
Figura 4 - Imagem ilustrativa representando os possíveis acionamentos dos motores unipolar e bipolar. ....	22
Figura 5 - Imagem ilustrativa representando o circuito de ponte H para demonstração de suas ligações elétricas. Na figura abaixo $V_{in}$ representa uma fonte elétrica de corrente contínua, S1 a S4 representam as chaves de contato e M representa um motor de corrente contínua. ....	23
Figura 6 – Imagem do <i>driver</i> 2PH64011A para demonstração de suas ligações eletrônicas. ....	24
Figura 7 – Imagem do Módulo <i>Bluetooth</i> HC-05 para demonstração de suas ligações eletrônicas. ....	25
Figura 8 – Imagem ilustrativa do circuito divisor de tensão para demonstração de suas ligações elétricas. ....	25
Figura 9 – Imagem representativa dos <i>templates</i> de aplicativos disponíveis no Ionic®. ....	27
Figura 10 – Imagem da vista frontal (a) e posterior (b) do equipamento de plasma atmosférico do LFFPP da UFTM. ....	28
Figura 11 – Imagem representativa dos eixos X e Y do sistema de injeção de plasma e o porta-amostras do equipamento de plasma atmosférico para demonstração do posicionamento dos motores de passo. ....	30
Figura 12 – Imagem representativa do deslocamento apenas no eixo X no porta-amostras do equipamento de plasma atmosférico. ....	31
Figura 13 – Imagem representativa do deslocamento apenas no eixo Y no porta-amostras do equipamento de plasma atmosférico. ....	31

Figura 14 – Imagem representativa do deslocamento em ambos os eixos sem passo no porta-amostras equipamento de plasma atmosférico. ....	32
Figura 15 – Imagem representativa do deslocamento em ambos os eixos com passo no eixo X no porta-amostras equipamento de plasma atmosférico. ....	32
Figura 16 – Imagem representativa do deslocamento em ambos os eixos com passo no eixo Y no porta-amostras equipamento de plasma atmosférico. ....	33
Figura 17 – Imagem ilustrativa das ligações eletrônicas feitas no Arduino©, motores de passo, <i>drivers</i> de acionamento, módulo <i>Bluetooth</i> , resistores e <i>buzzer</i> . ....	34
Figura 18 – Imagem dos arquivos e pastas do projeto no VS Code para demonstração do tamanho e tipos de arquivos. ....	35
Figura 19 – Imagem da simulação do aplicativo em tela cheia do navegador para demonstração do seu <i>layout</i> . ....	36
Figura 20 – Imagem da simulação do aplicativo em tela no formato de celular para demonstração do seu <i>layout</i> . ....	37
Figura 21 – Imagem da calibração dos motores de passo para demonstração dos procedimentos metodológicos. ....	38
Figura 22 – Imagem ilustrativa do acoplador do eixo X. ....	39
Figura 23 – Imagem ilustrativa do acoplador do eixo Y. ....	39
Figura 24 – Imagem ilustrativa do suporte do motor do eixo X. ....	40
Figura 25 – Imagem ilustrativa do suporte do motor do eixo Y. ....	40
Figura 26 – Imagem do processo de impressão 3D do suporte do motor de passo do eixo Y, ilustrando o uso da manufatura aditiva no processo de fabricação do sistema XY do porta-amostras. ....	41
Figura 27 – Imagem do suporte do motor de passo do eixo Y fabricado por processo aditivo em impressão 3D. ....	41
Figura 28 – Imagem do porta-amostras do equipamento de plasma atmosférico ilustrando o posicionamento dos motores de passo e seus respectivos suportes e acopladores no equipamento. ....	42

Figura 29 – Imagem da montagem do sistema de controle e automação com Arduino®, motores de passo, <i>drivers</i> de acionamento, módulo <i>Bluetooth</i> e <i>buzzer</i> ilustrando a composição do projeto mecânico e eletrônico.....	43
Figura 30 – Imagem das ligações eletrônicas do sistema de controle e automação com Arduino®, motores de passo, <i>drivers</i> de acionamento, módulo <i>Bluetooth</i> e <i>buzzer</i> ilustrando o detalhamento das ligações eletrônicas.....	43
Figura 31 – Imagem da montagem final do sistema de controle e automação com Arduino®, motores de passo, <i>drivers</i> de acionamento, módulo <i>Bluetooth</i> e <i>buzzer</i> ilustrando para demonstração de seu <i>layout</i> .....	44
Figura 32 – Imagem dos arquivos da pasta <i>src</i> para demonstração dos tipos de arquivos do aplicativo. ....	45
Figura 33 – Imagem da página <i>home</i> do aplicativo desenvolvido neste estudo.....	47
Figura 34 – Imagem do menu lateral do aplicativo desenvolvido neste estudo (a) e <i>select</i> do formulário para informar qual é o eixo de deslocamento (b). ....	48
Figura 35 - Imagem da mensagem para campos do formulário vazios (a) e para velocidade igual a zero e deslocamento diferente de zero para o motor X (b).....	49
Figura 36 - Imagem da mensagem para deslocamento igual a zero e velocidade maior que zero para o motor X (a), para velocidade igual a zero e deslocamento diferente de zero para o motor Y (b) e para deslocamento igual a zero e velocidade maior que zero para o motor Y (c). ....	50
Figura 37 – Imagem da mensagem para os limites do deslocamento do motor X (a), para os limites da velocidade do motor X (b) e para os limites do deslocamento do motor Y (c). ....	50
Figura 38 – Imagem da mensagem para os limites da velocidade do motor Y (a), para o deslocamento do eixo do motor X (b) e para o deslocamento do eixo do motor Y (c).....	51
Figura 39 – Imagem da mensagem para verificar se o <i>Bluetooth</i> do celular está ligado (a) e erro ao conectar com o módulo <i>Bluetooth</i> (b).....	51
Figura 40 – Imagem do <i>loading</i> para o usuário aguardar a finalização do processo. ....	52
Figura 41 – Imagem sobre as informações do aplicativo (a) e sobre as informações de contato do desenvolvedor (b). ....	53

## LISTA DE TABELAS

Tabela 1 – Deslocamento do motor X para 1 volta do motor medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21. ....	54
Tabela 2 – Deslocamento do motor Y para 1 volta do motor medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21. ....	55
Tabela 3 – Tempo para o motor de passo completar uma volta a 100 passos por minuto medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21. ....	55
Tabela 4 - Relação de preços dos itens adquiridos e utilizados no projeto aqui proposto. ....	57

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>16</b>
<b>2 OBJETIVOS .....</b>	<b>18</b>
2.1 OBJETIVO GERAL .....	18
2.2 OBJETIVOS ESPECÍFICOS .....	18
<b>3 REVISÃO BIBLIOGRÁFICA .....</b>	<b>19</b>
3.1 ARDUINO© .....	19
3.2 DISPOSITIVOS ELETRÔNICOS .....	20
<b>3.2.1 Motor de passo.....</b>	<b>20</b>
<b>3.2.2 Driver de acionamento .....</b>	<b>23</b>
<b>3.2.3 Módulo <i>Bluetooth</i> .....</b>	<b>24</b>
3.3 IONIC© .....	26
<b>4 PROCEDIMENTOS METODOLÓGICOS .....</b>	<b>28</b>
4.1 FUNCIONAMENTO E ANÁLISE DO EQUIPAMENTO .....	28
4.2 ARDUINO© E LIGAÇÕES ELETRÔNICAS .....	33
4.3 DESENVOLVIMENTO DO APLICATIVO .....	35
4.4 CALIBRAÇÃO DOS MOTORES .....	37
<b>5 RESULTADOS E DISCUSSÕES .....</b>	<b>39</b>
5.1 ACOPLADORES E SUPORTES .....	39
5.2 ARDUINO© E LIGAÇÕES ELETRÔNICAS .....	42
5.3 APLICATIVO .....	44
5.4 CALIBRAÇÃO DOS MOTORES .....	54
5.5 INVESTIMENTO DO PROJETO.....	56
<b>6 CONCLUSÕES.....</b>	<b>58</b>
<b>REFERÊNCIAS .....</b>	<b>60</b>

<b>ANEXO A – PROJETO DO ACOPLADOR DO EIXO X .....</b>	<b>66</b>
<b>ANEXO B – PROJETO DO ACOPLADOR DO EIXO Y .....</b>	<b>67</b>
<b>ANEXO C – PROJETO DO SUPORTE DO EIXO X.....</b>	<b>68</b>
<b>ANEXO D – PROJETO DO SUPORTE DO EIXO Y.....</b>	<b>69</b>
<b>ANEXO E – ALGORITMO DO ARDUINO© <i>BLUETOOTH</i>.....</b>	<b>70</b>

## 1 INTRODUÇÃO

Os filmes finos são camadas de materiais da ordem nanométrica até micrométrica e são utilizados com o objetivo de modificar superfícies de outros materiais de modo a incrementar determinadas propriedades tais como mecânica, ótica, magnética, elétrica, térmica e química (OHRING, 1991; SESHAN, 2002). O processo de fabricação de filmes finos teve grande avanço a partir da década de 1980 (SESHAN, 2002). O PVD (*Physical Vapor Deposition* – Deposição física de vapor) e CVD (*Chemical Vapor Deposition* – Deposição química de vapor) são alguns dos principais métodos pesquisados para a síntese de filmes finos (OHRING, 1991; SILVA, 2019; RODRIGUES, 2016; BUSO, 2016; CONTIJO, 2018). As principais aplicações dos filmes finos são na indústria eletrônica com a fabricação de supercapacitores para o armazenamento e conversão de energia, química com o tratamento microbiológico de efluentes, mecânica com a manufatura de ferramentas de corte para usinagem de materiais e alimentícia com a produção de embalagens ativas para aumentar a durabilidade dos alimentos (NAHIME, 2007; ZARPELON, 2013; MACHUNO et al.; CARVALHO, 2016; PENNA, 2018).

Diversas pesquisas têm sido realizadas com o propósito de modificar superfícies de materiais através de técnicas relacionadas ao uso de descargas de plasma (ALVES et al., 2005; ZARPELON, 2013). O plasma é um gás ionizado superaquecido e para sua produção em laboratório é necessário aplicar alta tensão contínua ou alternada. A formação do plasma pode suceder em duas formas: o gás em condição de pressão reduzida e à pressão atmosférica (ROGÉRIO, 2019). Para baixas pressões, destaca-se a técnica sputtering ou pulverização catódica que consiste na deposição de filmes finos na ordem nanométrica em um sistema de vácuo (MACHUNO et al., 2016). Por outro lado, o processo de plasma a pressão atmosférica é gerado pela descarga da barreira dielétrica de um gás inerte (Hélio, argônio e entre outros) à temperatura ambiente (REIS, 2017; DAMAS, 2017; ROGÉRIO, 2019).

Atualmente, o Laboratório de Filmes Finos e Processos a Plasma (LFFPP) da Universidade Federal do Triângulo Mineiro (UFTM) possui um equipamento de plasma atmosférico para a produção de filmes finos, desenvolvido na dissertação de mestrado do discente Lúcio Rogério para o tratamento de superfícies de materiais (ROGÉRIO, 2019). Entretanto, o posicionamento do porta-amostras do equipamento funciona manualmente, de forma que, a manipulação da mesa deste equipamento pode causar choque elétrico ao usuário.

Assim, a fim de proporcionar uma maior segurança operacional e exatidão do experimento, propõe-se o desenvolvimento de um sistema com aplicativo híbrido para controlar e automatizar o estágio XY do porta-amostras do equipamento de plasma atmosférico utilizando tecnologias como Arduino®, Ionic®, motores de passo e *Bluetooth*.

## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é o desenvolvimento de um sistema com aplicativo híbrido utilizando as tecnologias e *frameworks* como Arduino©, Ionic©, motores de passo e *Bluetooth* para controlar e automatizar o estágio XY do porta-amostras do equipamento de plasma atmosférico, obtendo maior segurança operacional.

### 2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos do trabalho são:

- Entender o funcionamento dos motores de passo e seu acionamento utilizando Arduino©;
- Realizar a comunicação entre o Javascript e o Arduino© para controlar os motores via *Bluetooth*;
- Pesquisar quais são as tendências de *layout* de aplicativos para uma melhor experiência de usuário (UX - *User Experience*);
- Quantificar os custos envolvidos do projeto;
- Fazer a montagem do sistema;
- Realizar testes e efetuar possíveis correções e ajustes.

### 3 REVISÃO BIBLIOGRÁFICA

#### 3.1 ARDUINO©

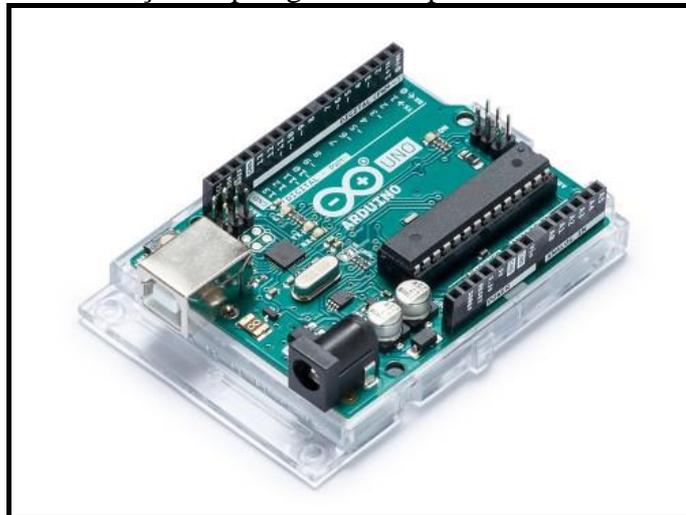
Arduino© é uma plataforma eletrônica *open-source*, ou seja, de código aberto, baseado em *hardware* e *software* de simples aplicação (ARDUINO©, 2019). Utilizando a plataforma do Arduino© é possível acessar dados de *inputs* (entradas do sistema) como sensores e botões, e atuar em *outputs* (saídas do sistema) como o acionamento de motores e *leds*.

A primeira placa foi criada em 2005 por um grupo de pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis (ARDUINO©, 2019). O principal objetivo dos pesquisadores era desenvolver uma placa microcontroladora de baixo custo (até 30 dólares), *open-source* de *hardware* e *software* de simples aplicação. O projeto Arduino© foi baseado na dissertação do Hernando Barragan sobre a placa Wiring que na época era orientando do professor Massimo Banzi.

O Arduino© utiliza uma linguagem de programação própria baseada em C / C++ e o desenvolvimento de programas é feito através da sua IDE (*Integrated Development Environment* - Ambiente Integral de Desenvolvimento), em outras palavras é um *software* que auxilia no desenvolvimento da aplicação (ARDUINO©, 2019).

Existem vários tipos de plataformas do Arduino© tais como Arduino© Uno, Leonardo, Micro, Nano, Esplora, Mega e dentre outros. A Figura 1 mostra a plataforma do Arduino© Uno que é a principal plataforma do microcontrolador.

Figura 1 – Imagem de uma placa comercial da Plataforma do Arduino© Uno para demonstração da pinagem e componentes eletrônicos.



Fonte: Arduino©, 2019.

Como demonstrado na Figura 1, a placa do Arduino® é composta por um microprocessador Atmega8U2, um oscilador, um regulador linear de 5 Volts, entrada USB (*Universal Serial Bus* – Porta Serial Universal), pinos digitais e analógicos para funcionamento das entradas e saídas do sistema (MCROBERTS, 2011).

Existem placas de circuitos prontas para a utilização em projetos mais complexos, essas placas são conhecidas como *shields* ou módulos, o emprego delas é imprescindível para diminuir o número de conexões e manter uma melhor organização do *hardware* (MCROBERTS, 2011). Ethernet Shields, módulo Relay e módulo para motor são alguns exemplos de *shields* para Arduino®.

Segundo Arduino® (2019), as vantagens de sua utilização são:

- Ambiente de desenvolvimento limpo e simples;
- Plataforma relativamente barata, com aproximadamente 50 reais é possível adquirir uma plataforma de Arduino® com outros módulos;
- *Cross-platform*: A IDE do Arduino® é desenvolvida para os sistemas operacionais Linux, Mac e Windows;
- *Open source* e extensão de *software*: Programadores podem desenvolver bibliotecas para atender suas demandas;
- *Open source* e extensão de *hardware*: Projetistas de circuitos podem criar sua própria versão do módulo da placa para possíveis melhoras ou adequação de projetos;
- Comunidade madura: Milhões de pessoas no mundo compartilham seus projetos em redes sociais para programadores.

Além disso, vários autores recomendam o uso da placa do Arduino®, principalmente pela sua vantagem comercial em relação a outras placas controladoras e pela sua versatilidade para vários projetos (D'AUSILIO, SOUZA et al.; 2011).

## 3.2 DISPOSITIVOS ELETRÔNICOS

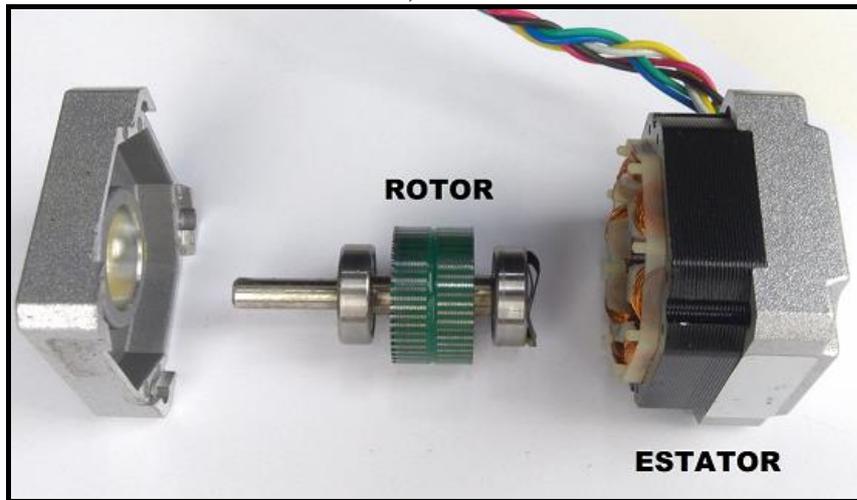
### 3.2.1 Motor de passo

O motor de passo é um dispositivo eletromecânico que converte os sinais elétricos em movimentos discretos mecânicos (THOMSEN, 2013). Com esses motores de corrente contínua (CC) é possível controlar sua velocidade, sentido de giro e os giros do eixo ou passos com muita

precisão, desta forma, eles são comumente empregados em impressoras, máquinas CNC (Comando Numérico Computadorizado), robôs e máquinas que exigem maior precisão (MCROBERTS, 2011; THOMSEN, 2013).

O motor é composto por um rotor de ferro, com forma de engrenagem, anexado aos eixos dentro do motor através de eletromagnetos dentados (MCROBERTS, 2011). A Figura 2 mostra um motor de passo aberto.

Figura 2 – Imagem de um motor de passo Nema 23 aberto para demonstração de seus constituintes internos, tais como rotor e estator.



Fonte: Robocore, 2018.

O funcionamento do motor de passo é realizado pelo acionamento das bobinas que atraem o rotor e alinha com o eixo determinado pelos solenoides, gerando uma pequena variação angular chamada de passo (BRITES; SANTOS, 2008). Normalmente os motores de passo possuem uma resolução de  $1,8^\circ$  por pulso, ou seja, 200 pulsos representam uma volta (MCROBERTS, 2011). A quantidade de pulsos gerados pelas bobinas corresponde a quantidade passos do motor.

Os motores de passo possuem três tipos de estruturas: relutância variável, ímã permanente e híbrido (BRITES; SANTOS, 2008):

- Motores com relutância variável possuem um rotor de ferro com múltiplos dentes e um estator com bobinas que, quando acionadas, geram polos magnéticos, desta forma os dentes do rotor são atraídos para os polos do estator;
- Os motores de ímã permanente são formados por um rotor com ímãs permanentes e não possuem dentes. Os polos magnetizados do rotor geram a intensidade de fluxo

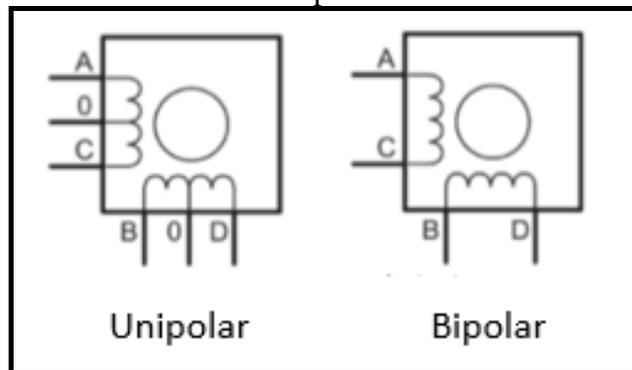
magnético, assim, os motores de ímã permanente exibem um maior torque comparado aos motores de relutância variáveis, entretanto possuem menor resolução por passo;

- Já os motores híbridos combinam as melhores características dos motores de relutância variável e ímã permanente. O rotor é multidentado e possui um ímã permanente ao redor do eixo. Eles são os motores mais comuns do mercado e possuem uma resolução entre 3,6° a 0,9°.

Para energizar as bobinas dos motores de passo é necessário fazer as ligações elétricas. Essas ligações podem ser unipolares ou bipolares (BRITES; SANTOS, 2008; MCROBERTS, 2011).

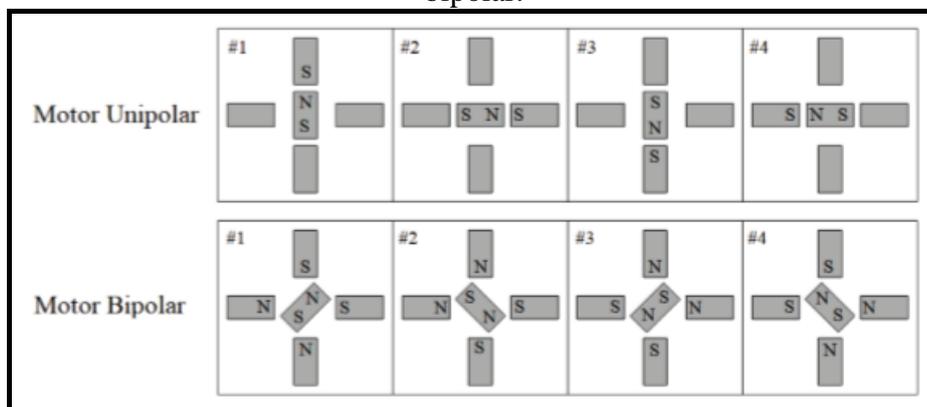
Os motores unipolares possuem cinco ou seis fios que são direcionados para as quatro bobinas, sendo que um ou dois dos fios são de derivação central (MCROBERTS, 2011). Já os motores bipolares possuem quatro fios sem derivação central, ou seja, suas bobinas são acionadas em conjunto. A Figura 3 representa ligações do motor de passo unipolar e bipolar. Já a Figura 4 mostra o acionamento de cada tipo de motor.

Figura 3 – Imagem ilustrativa representando as possíveis ligações do motor de passo unipolar e bipolar.



Fonte: Adaptado de McRoberts, 2011.

Figura 4 - Imagem ilustrativa representando os possíveis acionamentos dos motores unipolar e bipolar.



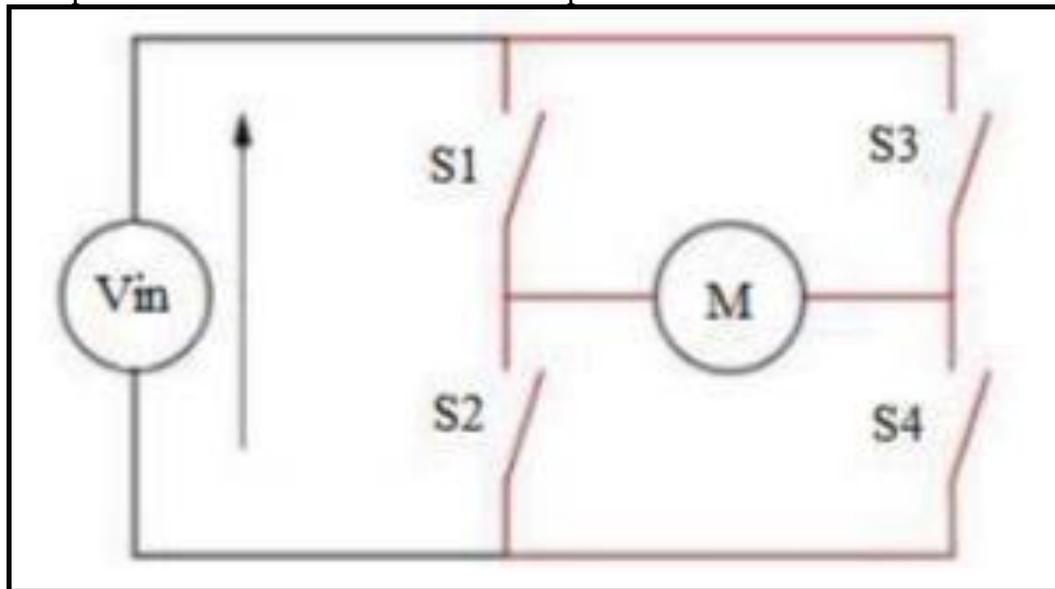
Fonte: Fiebig, 2018.

Os motores de passo têm algumas vantagens como alta exatidão em seu posicionamento, lógica digital, precisão no torque e excelente resposta a aceleração e desaceleração. Entretanto, possuem desvantagens como o baixo desempenho em altas velocidades e a ocorrência de ressonância por controle inadequado (BRITES; SANTOS, 2008):

### 3.2.2 *Driver de acionamento*

O *driver* de acionamento ou ponte H é um circuito eletrônico que fornece um sinal com potência e corrente elétrica suficiente para acionar um motor em determinada velocidade e sentido de rotação. O nome ponte H é devido a montagem dos componentes eletrônicos. O circuito é composto por 4 transistores que são acionados de forma alternada (BRITES; SANTOS, 2008). A Figura 5 representa um circuito da ponte H.

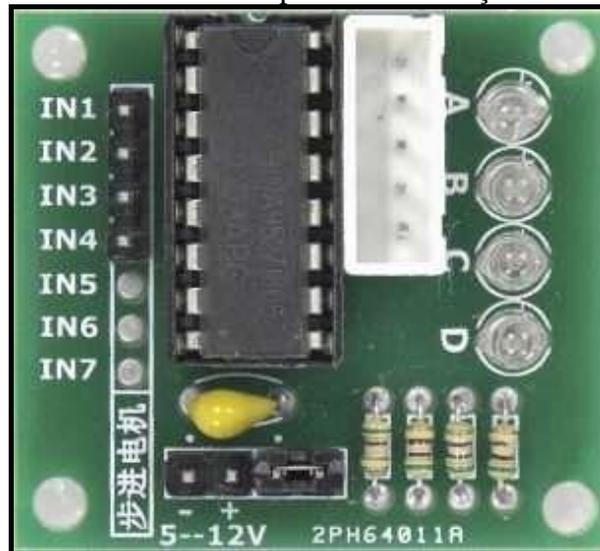
Figura 5 - Imagem ilustrativa representando o circuito de ponte H para demonstração de suas ligações elétricas. Na figura abaixo  $V_{in}$  representa uma fonte elétrica de corrente contínua, S1 a S4 representam as chaves de contato e M representa um motor de corrente contínua.



Fonte: Fiebig, 2018.

O *driver* 2PH64011A é bastante utilizado no controle do motor de passo 28BYJ-48. Ele é composto por um *chip* controlador ULN2003, um pino GND (*Ground* - Terra), um pino 5V-12V, quatro entradas digitais (IN1 a IN4) e cinco saídas para alimentação do motor. O *datasheet* do *driver* 2PH64011A está disponível em <https://datasheet4u.com/datasheet-parts/ULN2003APG-datasheet.php?id=925702>. A Figura 6 representa o *driver* 2PH64011A.

Figura 6 – Imagem do *driver* 2PH64011A para demonstração de suas ligações eletrônicas.



Fonte: Murta, 2018.

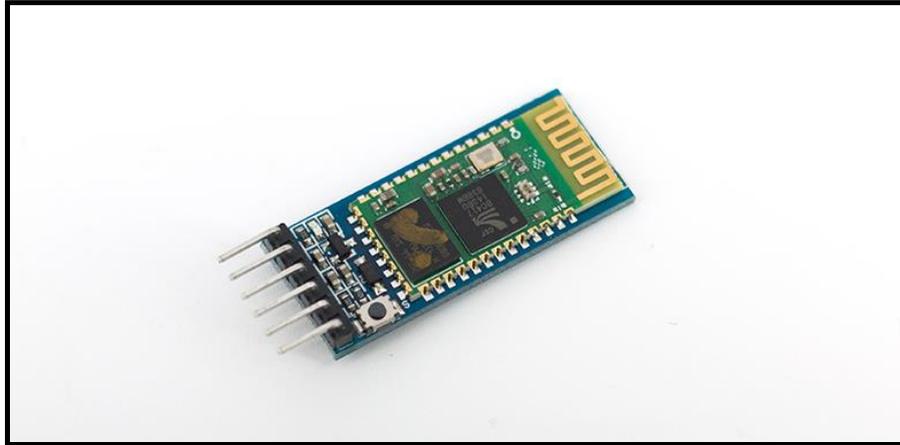
### 3.2.3 Módulo *Bluetooth*

*Bluetooth* é uma forma simples de comunicação sem fio entre dispositivos a uma curta distância (HUANG, RUDOLPH, 2005). Essa tecnologia foi desenvolvida em 1994 pela empresa de telecomunicações Ericsson e atualmente é bastante utilizada para conectar e compartilhar arquivos entre dispositivos móveis como celulares e *tablets* (GRÉGIO, 2009; BONATTO; CANTO, 2007).

O módulo *Bluetooth* permite que o Arduino© conecte com outro dispositivo e realize a transferência de informações entre eles (OLIVEIRA, 2019). Os principais módulos *Bluetooth* para conectar com Arduino© são o HC-05 e HC-06, a diferença entre eles é que o HC-05 funciona como mestre e escravo, em outras palavras o dispositivo envia e recebe informações; e o HC-06 funciona somente como escravo, ou seja, apenas recebe informações (VIDAL, 2017).

A pinagem do módulo *Bluetooth* HC-05 são as seguintes: *State*, RX, TX, GND, 5V e EN. Para o funcionamento do módulo é necessário apenas os pinos de alimentação (5V e GND) e os pinos de comunicação serial (RX – TX). O pino EN modifica as configurações do dispositivo como nome e senha. Já o terminal *State* está conectado com o *led* do dispositivo e indica os status de conexão através da velocidade entre cada piscada. A Figura 7 representa o módulo *Bluetooth* HC-05.

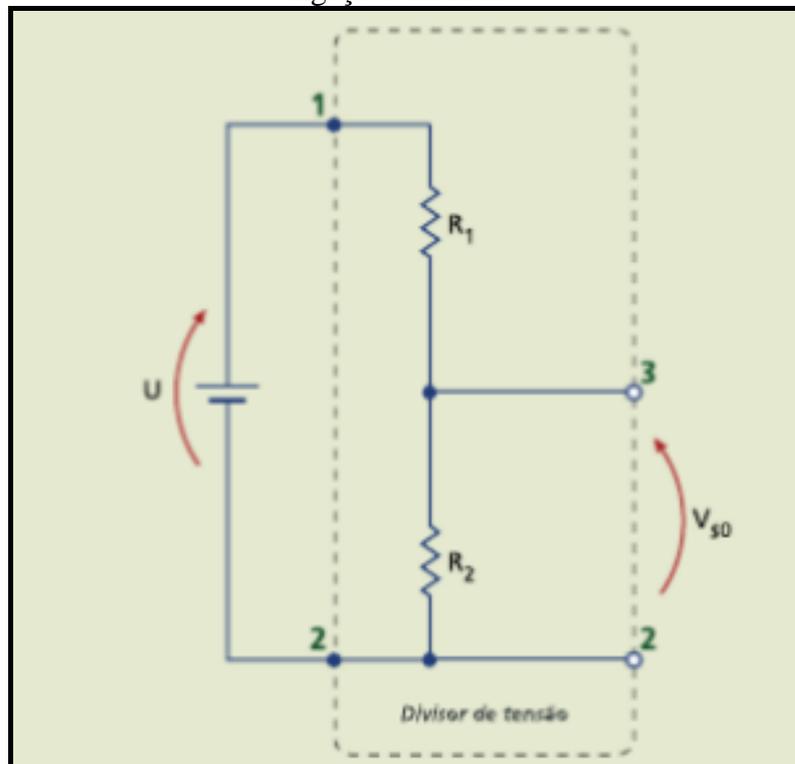
Figura 7 – Imagem do Módulo *Bluetooth* HC-05 para demonstração de suas ligações eletrônicas.



Fonte: Thomsen, 2015.

Apesar do dispositivo ser alimentado com 5 Volts, os pinos de comunicação serial funcionam com, 3,3 Volts (THOMSEN, 2015). Desta forma, é necessário montar um circuito divisor de tensão para não danificar o módulo. O divisor de tensão é um circuito eletrônico que fornece em sua saída um valor de tensão menor que da entrada do circuito (AFONSO, FILONI, 2011). A Figura 8 representa um circuito divisor de tensão.

Figura 8 – Imagem ilustrativa do circuito divisor de tensão para demonstração de suas ligações elétricas.



Fonte: Afonso, Filoni, 2015.

O funcionamento do circuito divisor de tensão é da seguinte forma: A tensão de entrada ( $U$ ) é aplicada nos terminais 1 e 2 com dois resistores em série ( $R_1$  e  $R_2$ ), deste modo a tensão de saída ( $V_{S0}$ ) é obtida entre os terminais 3 e 2, conforme demonstrado na Figura 8 (AFONSO, FILONI, 2011). A tensão de saída do circuito divisor de tensão é representada pela Equação (1).

$$V_{S0} [V] = \frac{U [V] * R_2[\Omega]}{R_1[\Omega] + R_2[\Omega]} \quad (1)$$

### 3.3 IONIC©

O *framework* Ionic© é um *kit* de ferramentas de interface do usuário *open source* para a criação de aplicativos *mobile* (móveis) e de *desktop* de alto desempenho e de alta qualidade, usando tecnologias da Web como HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript (IONIC©, 2020). O HTML é responsável por criar conteúdo da página, o CSS é responsável por estilizar os elementos da página e o JavaScript é responsável por executar funções e controle (GUANABARA, 2019). Desta forma, o *framework* é uma combinação de várias tecnologias que tornam o desenvolvimento *mobile* mais rápido e fácil (GRIFFITH, 2017).

O *framework* foi lançado em novembro de 2013 e se popularizou rapidamente (GRIFFITH, 2017). Atualmente, o Ionic© é um dos *frameworks* mais utilizado no mundo para o desenvolvimento *mobile* junto com o React Native e o Flutter (LIMA, 2019).

A grande vantagem em desenvolver aplicativos em Ionic© é a utilização do mesmo código para criar dois aplicativos, um para Android e outro para IOS.

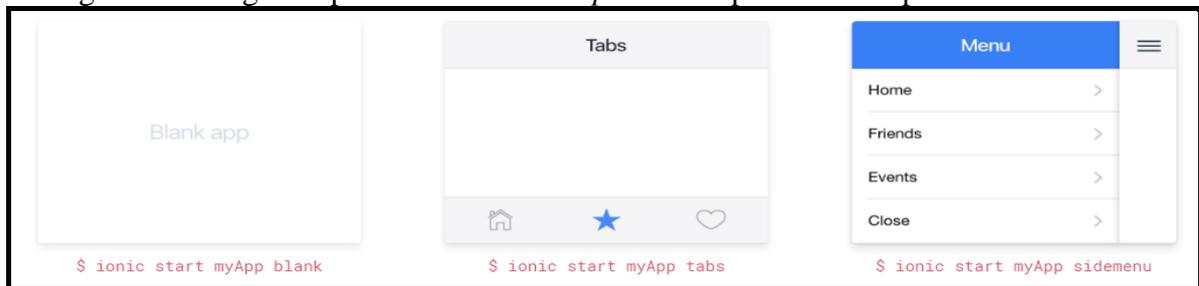
Segundo Ionic© (2020), as vantagens de utilizar o Ionic© são:

- *Cross-platform*: Desenvolvimento de aplicativos em várias plataformas como iOS, Android e Desktop utilizando o mesmo código;
- *Web Standards-based*: O *framework* é baseado sobre tecnologias web padronizadas como HTML, CSS e JavaScript;
- *Design*: Projeto simples, limpo e funcional;
- *Simplicidade*: A curva de aprendizagem é rápida;
- *Comunidade madura*: Milhões de pessoas no mundo compartilham seus projetos em redes sociais para programadores.

Para instalar o Ionic© *framework*, é necessário instalar o Node.JS® (JavaScript) e o npm (*Node package manager*). O Node JS® é um *framework* que executa o JavaScript de forma assíncrona orientada a eventos (NODEJS, 2020). O npm é o gerenciador de pacotes do Node JS ® e ele registra e compartilha *softwares* de código aberto que são desenvolvidos por programadores do mundo todo, desta forma outros programadores podem usar esse código em seus *softwares* (NPM, 2020).

O Ionic disponibiliza para o programador alguns *templates* para aplicativos, conforme Figura 9.

Figura 9 – Imagem representativa dos *templates* de aplicativos disponíveis no Ionic©.



Fonte: Ionic©, 2020.

Desta forma o programador pode optar por um aplicativo em branco (*blank*), separador inferior (*tabs*) ou menu lateral (*sidemenu*).

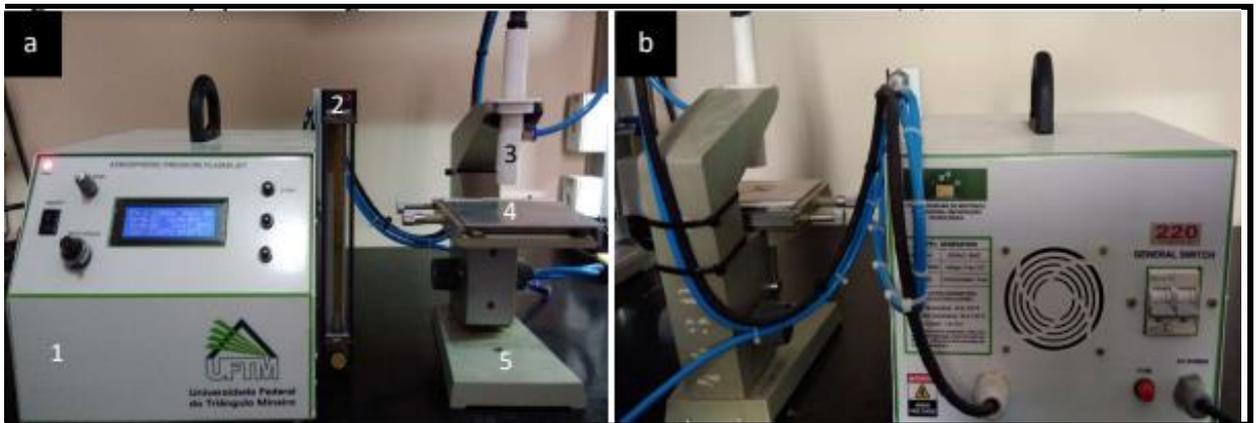
Além disso, o Ionic© possui vários *plugins* que realizam a integração com sensores e dispositivos dos celulares. Câmera, geolocalização, sensor biométrico e *Bluetooth* são alguns exemplos.

## 4 PROCEDIMENTOS METODOLÓGICOS

### 4.1 FUNCIONAMENTO E ANÁLISE DO EQUIPAMENTO

O equipamento de plasma atmosférico (APPJ – *Atmospheric Pressure Plasma Jet*) é empregado na modificação de superfícies e no tratamento de materiais através do princípio de barreira dielétrica (DBD). Desta forma, ocorre micro descargas aleatórias com duração de 10 a 100 nanosegundos sobre a superfícies de diversos materiais (Policloreto de vinila – PVC, cobre e outros) alterando as características superficiais (ROGÉRIO, 2019). A Figura 10 representa o equipamento do laboratório.

Figura 10 – Imagem da vista frontal (a) e posterior (b) do equipamento de plasma atmosférico do LFFPP da UFTM.



Fonte: Rogério, 2019.

Observa-se pela Figura 10 que o equipamento é composto pelos seguintes itens:

- Módulo eletroeletrônico de controle e potência (1): Responsável por controlar e fornecer os parâmetros da geração do plasma (tensão, corrente elétrica, largura do pulso elétrico, frequência do pulso, reset do equipamento e outros) através do *software* desenvolvido em C e o microcontrolador PIC 16F877A;
- Regulador de fluxo (2): Responsável por controlar o fluxo do gás inerte (Hélio, argônio e entre outros) através do regulador de vazão de precisão;
- Bico injetor de plasma (3): Composto por um tubo dielétrico com eletrodo de cobre responsável por transmitir o fluxo de gás inerte para o porta-amostras e gerar a descarga de plasma;
- Porta-amostras (4): Composta por uma placa de alumínio responsável receber as amostras de diversos materiais;

- Base suporte de alumínio (5): Responsável por sustentar a estrutura do bico injetor de plasma e o porta-amostras.

O funcionamento do equipamento de plasma atmosférico consiste em ligar o módulo de controle e potência, realizar o setup do equipamento (temperatura da bobina de ignição e temperatura do componente eletrônico Mosfet e tensão) e selecionar a frequência para a produção da descarga de plasma. Em seguida, o usuário deve controlar a vazão do fluxo de gás inerte pelo regulador de fluxo e a altura do porta-amostras pela base do equipamento até a geração do plasma. Caso haja alguma falha no equipamento, o microcontrolador desliga automaticamente o circuito de potência. Esse equipamento foi desenvolvido na dissertação de mestrado do discente Lúcio Rogério no Programa de Mestrado Profissional em Inovação Tecnológica da UFTM (ROGÉRIO, 2019).

O equipamento de plasma atmosférico possui uma mesa contendo dois eixos para fazer o controle da posição XY das amostras a serem tratadas, sendo chamada de porta-amostras, desta forma, são necessários dois motores de passo com seus respectivos *driver* de acionamento para realizar o controle e automação do processo. O eixo X é responsável pela movimentação longitudinal da mesa, já o eixo Y é responsável pela movimentação transversal da mesa. Os eixos X e Y são nomeados de acordo com a Figura 11. Essa mesma mesa é responsável por prender o jato de plasma posicionando-o acima do porta-amostras.

Figura 11 – Imagem representativa dos eixos X e Y do sistema de injeção de plasma e o porta-amostras do equipamento de plasma atmosférico para demonstração do posicionamento dos motores de passo.



Fonte: Autor, 2019.

Para instalar os dois motores de passo no equipamento, é necessário confeccionar dois acopladores e dois suportes. Os suportes têm a função de fixar os motores na estrutura do equipamento, já os acopladores transmitem a rotação do eixo do motor de passo para o eixo dos *charriots*<sup>1</sup> do sistema de plasma.

O deslocamento do porta-amostras do equipamento de plasma deve ser da seguinte forma:

- Apenas no eixo X;
- Apenas no eixo Y;
- Em ambos os eixos sem passo;

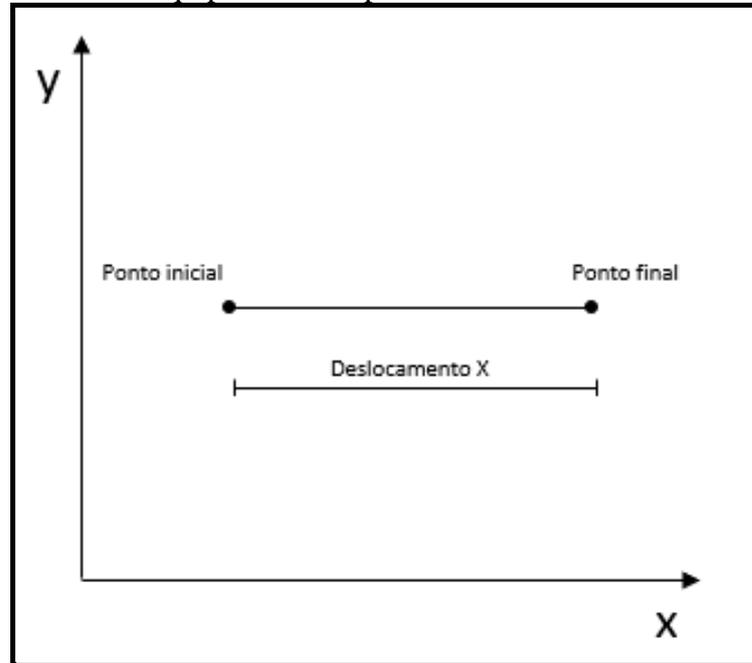
---

<sup>1</sup> *Charriots* é uma peça responsável pela movimentação da mesa do equipamento no plano horizontal.

- Em ambos os eixos com passo no eixo X;
- Em ambos os eixos com passo no eixo Y.

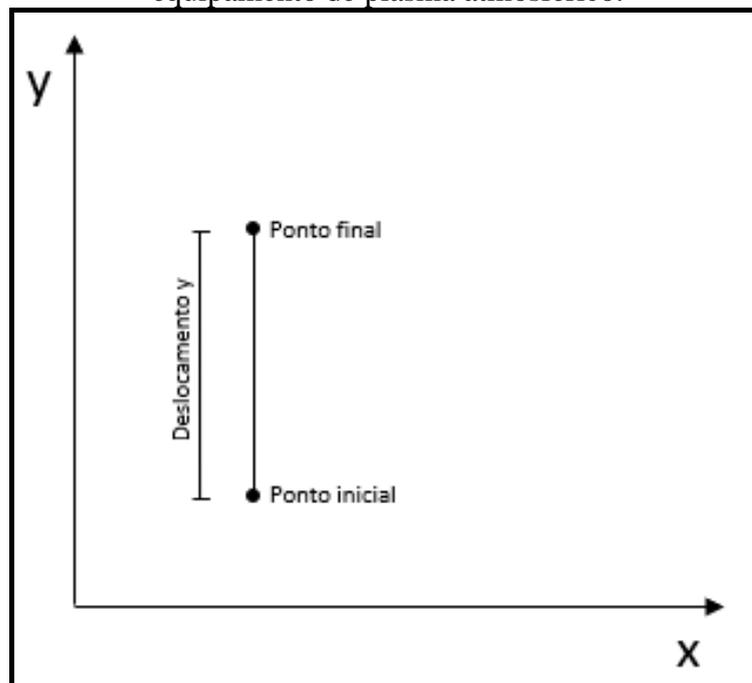
A Figura 12 até a Figura 16 demonstram cada possível movimento.

Figura 12 – Imagem representativa do deslocamento apenas no eixo X no porta-amostras do equipamento de plasma atmosférico.



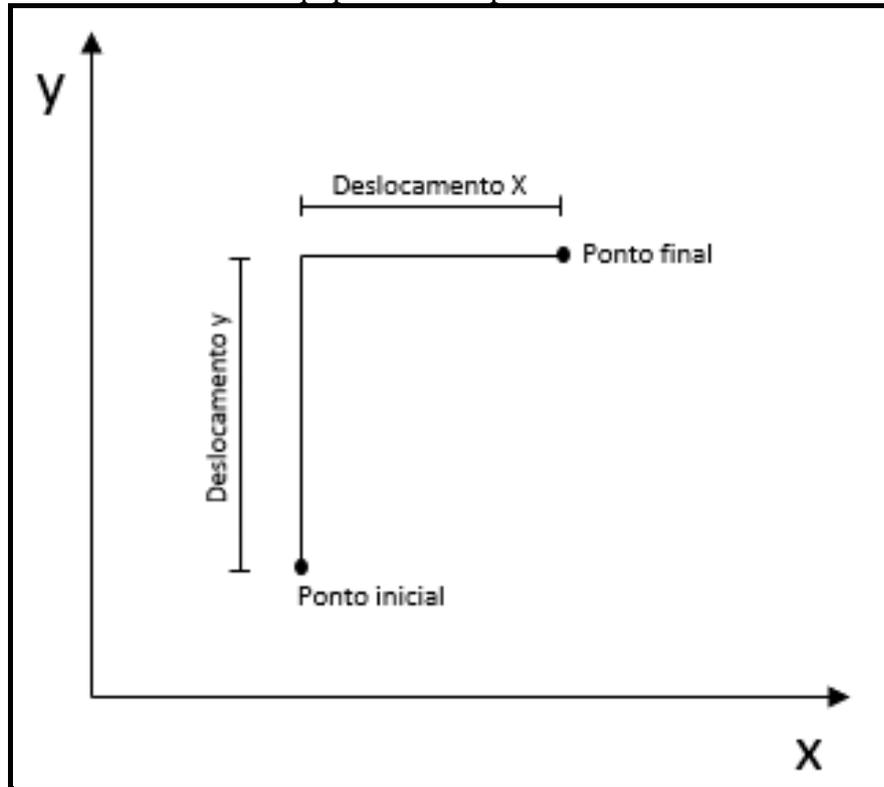
Fonte: Autor, 2020.

Figura 13 – Imagem representativa do deslocamento apenas no eixo Y no porta-amostras do equipamento de plasma atmosférico.



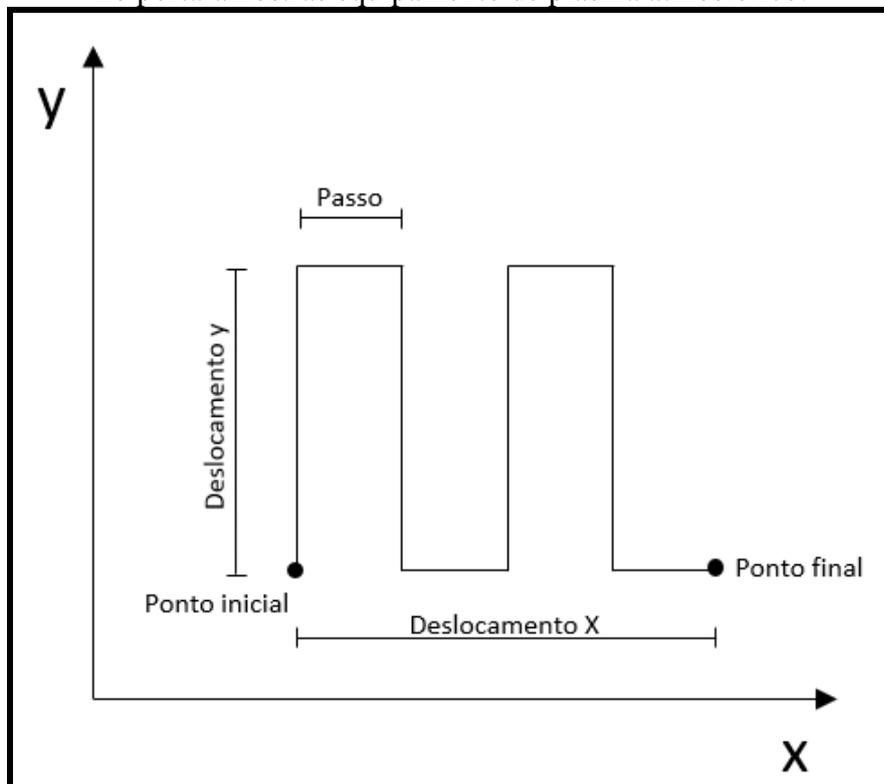
Fonte: Autor, 2020.

Figura 14 – Imagem representativa do deslocamento em ambos os eixos sem passo no porta-amostras equipamento de plasma atmosférico.



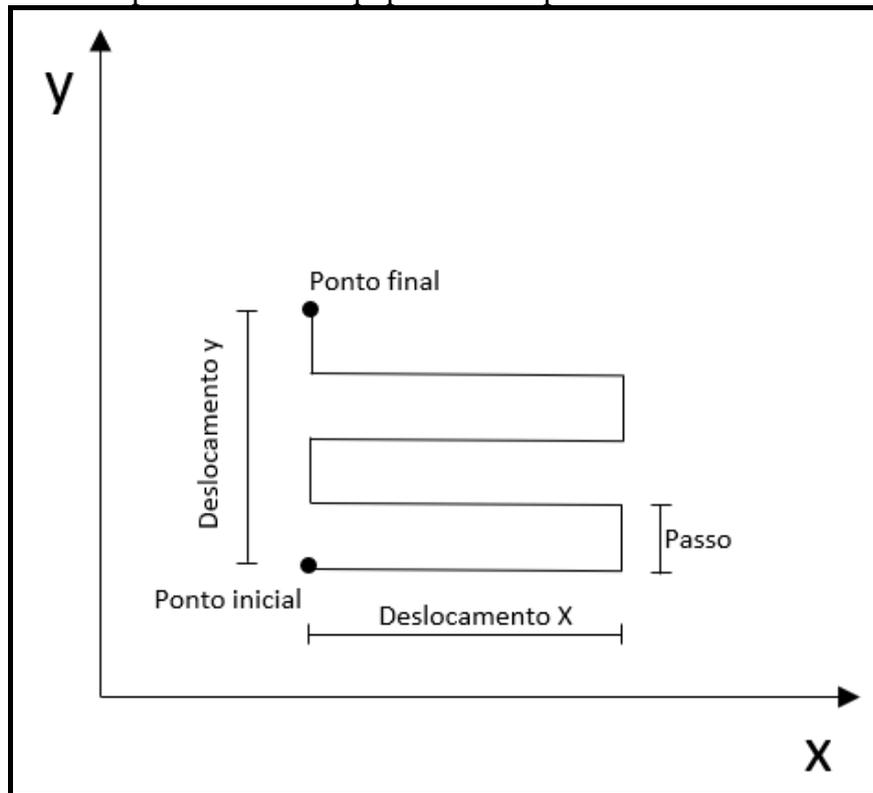
Fonte: Autor, 2020.

Figura 15 – Imagem representativa do deslocamento em ambos os eixos com passo no eixo X no porta-amostras equipamento de plasma atmosférico.



Fonte: Autor, 2020.

Figura 16 – Imagem representativa do deslocamento em ambos os eixos com passo no eixo Y no porta-amostras equipamento de plasma atmosférico.



Fonte: Autor, 2020.

#### 4.2 ARDUINO® E LIGAÇÕES ELETRÔNICAS

O motor de passo e o *driver* de acionamento utilizados no desenvolvimento deste trabalho são o 28BYJ-48 e 2PH64011A, respectivamente. Esse motor é alimentado com 5 Volts, é unipolar com 4 bobinas e 5 fios, possui uma redução de 1:64, ângulo de passo de  $5,625^\circ/64$  e um torque de  $34,3\text{mN}\cdot\text{m}$ . O *datasheet* do motor de passo 28BYJ-48 está disponível em [https://img.filipeflop.com/files/download/Datasheet\\_28BYJ-48.pdf](https://img.filipeflop.com/files/download/Datasheet_28BYJ-48.pdf). O *driver*, por sua vez, possui alimentação de 5 Volts a 12 Volts, contém o controlador ULN2003 e corrente de controle máxima de 500mA/canal (KIATRONICS; TOSHIBA, 2019). O *datasheet* do módulo *Bluetooth* HC-05 está disponível em [https://components101.com/sites/default/files/component\\_datasheet/HC-05%20Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf). Deste modo, o *driver* de acionamento utiliza o pino 12 Volts de uma fonte externa, GND e 4 pinos digitais para o controle das bobinas. No caso do sistema de plasma, o equipamento deve possuir dois motores para controlar e automatizar o movimento de sua mesa.

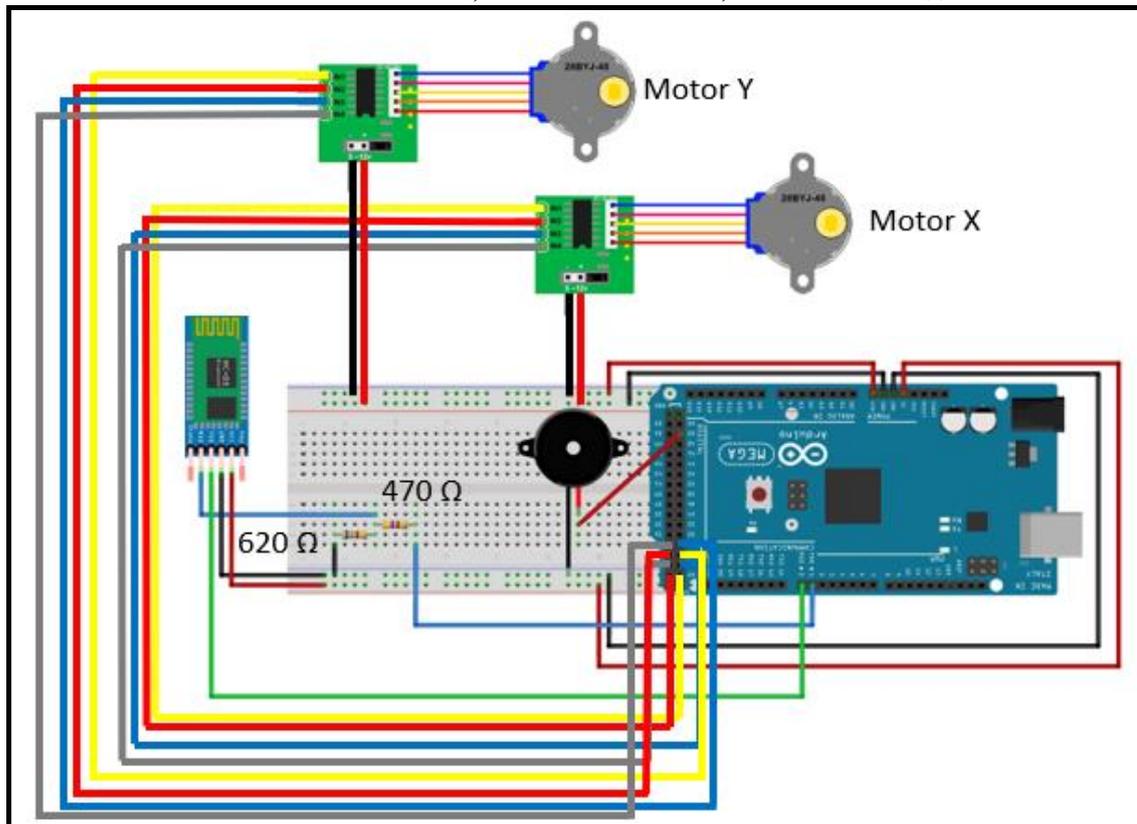
Para a comunicação *Bluetooth* utilizou-se o módulo *Bluetooth* HC-05 no modo *slave*, com isso o aplicativo envia os dados para o dispositivo e este por sua vez transmite a informação

para o Arduino® via comunicação serial. Com isso, o módulo *Bluetooth* utiliza os pinos 5 Volts, GND, TX e RX com divisor de tensão para a comunicação serial.

Com o intuito de alertar o usuário sempre que o sistema inicie e finalize a operação, foi conectado um *buzzer* de 5 Volts para emitir um sinal sonoro. O *buzzer* utiliza um pino digital e GND.

Em posse de todas essas informações, montou-se um esquema das ligações eletrônicas dos motores de passo, *drivers* de acionamento, módulo *Bluetooth* e o *buzzer* com a placa Arduino® Mega. O Arduino® foi ligado a uma fonte externa de 12 Volts. O motor X foi ligado aos pinos 22 a 25, 12 Volts e GND. O motor Y foi ligado aos pinos 28 a 31, 12 Volts e GND. O módulo Bluetooth foi ligado aos TX, RX e pinos 5 Volts, GND, ressalta se que o TX do Arduino® (RX do módulo *Bluetooth*) foi feito um divisor de tensão com resistores de  $620\ \Omega$  e  $470\ \Omega$  para atender a tensão de 3,3 Volts do módulo. O *buzzer* foi ligado ao pino 50 e GND. A Figura 17 mostra esse esquema das ligações detalhado.

Figura 17 – Imagem ilustrativa das ligações eletrônicas feitas no Arduino®, motores de passo, *drivers* de acionamento, módulo *Bluetooth*, resistores e *buzzer*.



Fonte: Autor, 2020.

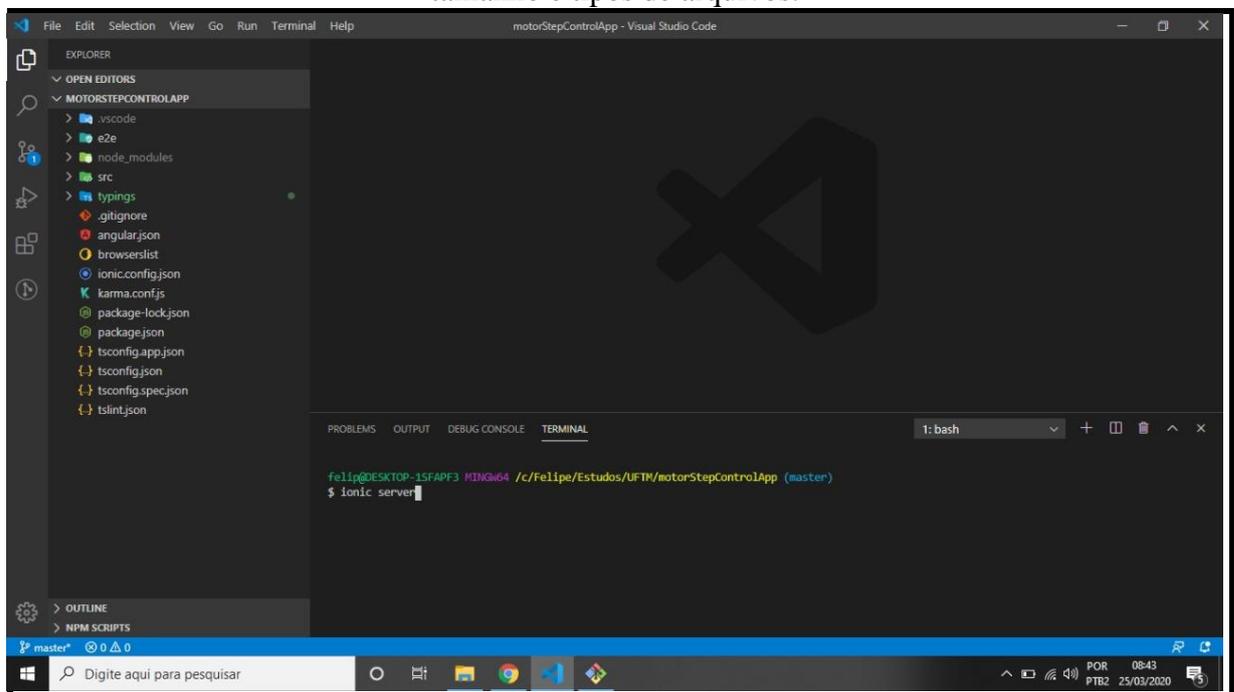
### 4.3 DESENVOLVIMENTO DO APLICATIVO

Após instalar e configurar o ambiente de desenvolvimento no computador, escolheu-se o *template sidemenu* da Figura 9, por apresentar melhor experiência de usuário para o aplicativo proposto baseado em aplicativos IoT (*Internet of Things*). Para gerar o aplicativo, é necessário abrir um terminal a partir de uma pasta que não possua sinais de acentuação gráfica ou espaços; e digitar o comando `ionic start stepperMotorControlApp sidemenu`. Com isso, o Ionic cria uma pasta chamada `stepperMotorControlApp` com todo o código do aplicativo e suas dependências.

Para poder alterar o código existente ou desenvolver novas funções, é necessário instalar um editor de código. Neste caso, utilizou-se o Visual Studio Code (VS Code) que é um editor de código livre e gratuito desenvolvido pela Microsoft© disponível para as plataformas Windows, MacOS e Linux (VISUAL STUDIO CODE, 2020).

A Figura 18 mostra os arquivos e pastas gerados no projeto do aplicativo visualizados a partir do VS Code.

Figura 18 – Imagem dos arquivos e pastas do projeto no VS Code para demonstração do tamanho e tipos de arquivos.



Fonte: Autor, 2020.

Ao observar a Figura 18, nota-se que o projeto do aplicativo possui inúmeros arquivos e pastas. São aproximadamente 70.000 arquivos e 7.000 pastas, totalizando 448 megabytes. Deste modo, recorreu-se a um sistema de gerenciamento de código e repositório em nuvem para

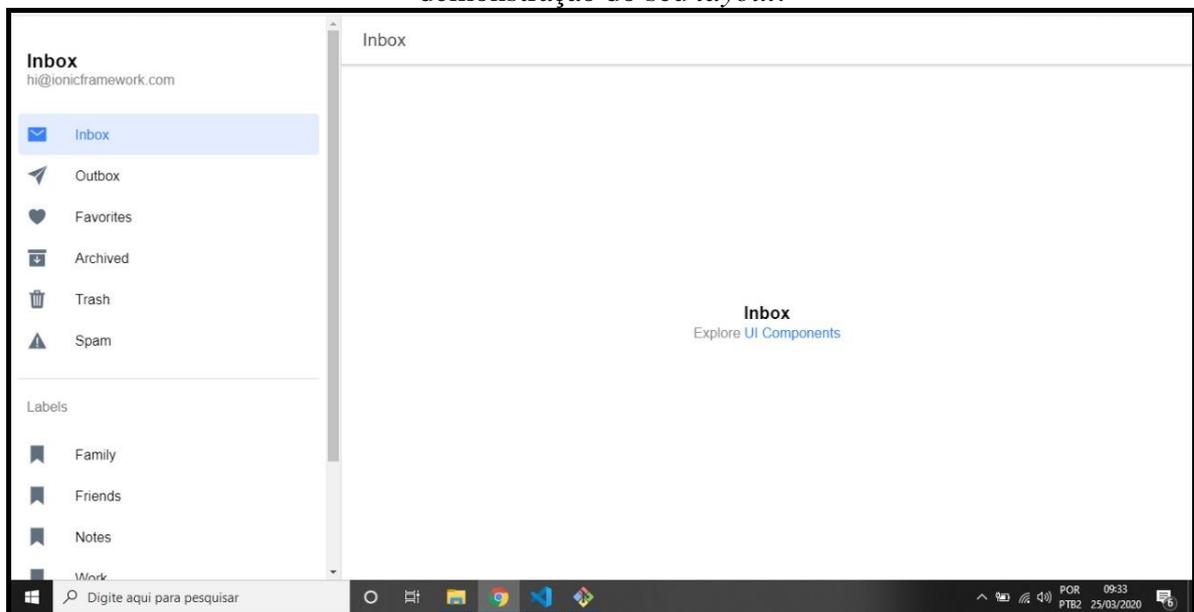
realizar o compartilhamento do código do aplicativo. Uma das ferramentas mais utilizadas é o git e o github.

O git é um sistema de controle de versão de arquivos gratuito e *open-source* (GIT, 2020), já o github é uma plataforma de desenvolvimento *open-source* que hospeda códigos, além disso ela também funciona como uma rede social de programadores (GITHUB, 2020).

Para compartilhar o código na plataforma GitHub é necessário criar uma conta e em seguida acessar e criar um novo repositório. Para vincular o repositório criado com o código, basta abrir o terminal na pasta do projeto e inserir os seguintes comandos: `git init`; `git commit -m "Nome do primeiro commit"`, `git remote add origin git@github.com:"Nome do usuário do GitHub"/"Nome do repositório".git` e `git push -u origin master`.

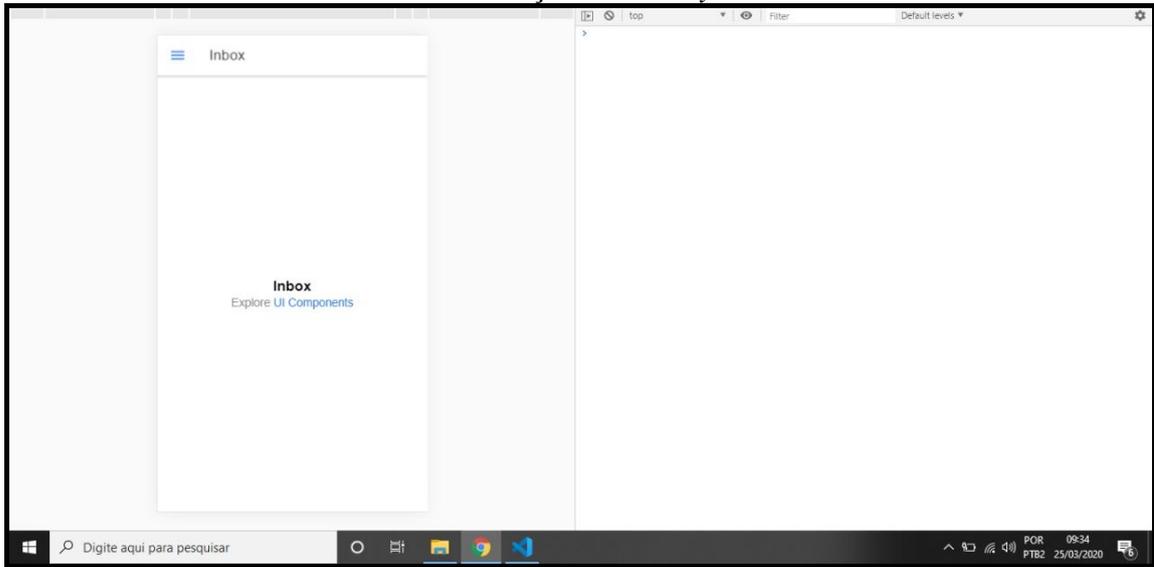
Além de visualizar, editar o código existente e criar novas funções, o VS Code possui vários *plugins* e extensões que auxiliam no desenvolvimento do aplicativo, desta forma é possível simular o aplicativo através de navegadores como o Google Chrome usando o computador. Para simular o aplicativo e suas novas funções, executou-se o comando `ionic server` no terminal do VS Code, conforme Figura 19 e Figura 20.

Figura 19 – Imagem da simulação do aplicativo em tela cheia do navegador para demonstração do seu *layout*.



Fonte: Autor, 2020.

Figura 20 – Imagem da simulação do aplicativo em tela no formato de celular para demonstração do seu *layout*.



Fonte: Autor, 2020.

Para realizar a comunicação entre o aplicativo em Ionic© e o Arduino© instalou-se o *plugin Bluetooth Serial* disponível em: <https://ionicframework.com/docs/native/bluetooth-serial> através dos comandos `ionic cordova plugin add cordova-plugin-bluetooth-serial` e `npm install @ionic-native/bluetooth-serial`.

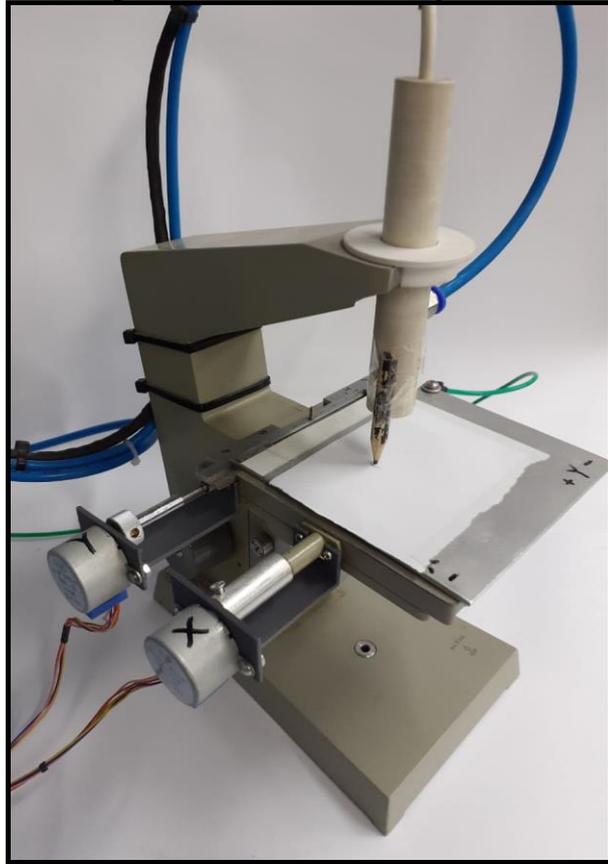
#### 4.4 CALIBRAÇÃO DOS MOTORES

Em decorrência das grandezas de rotação dos motores de passo e deslocamento linear de cada eixo da mesa do sistema de plasma atmosférico não estarem relacionadas, foi necessário realizar a calibração dos motores. Desta forma, a rotação do motor X e Y corresponderá a um valor de deslocamento de seus respectivos eixos no sistema de plasma. De forma semelhante, a velocidade angular de cada motor corresponderá a uma velocidade linear.

Sendo assim, executou-se a calibração de cada motor individualmente. O processo consiste em realizar a rotação de uma volta no motor, medir o deslocamento causado por essa volta e relacionar as rotações do motor com o deslocamento da mesa do equipamento.

Para tanto, fixou-se os motores de passo do eixo X e Y em seus respectivos suportes, um lápis no bico injetor de plasma e uma folha na mesa do equipamento, conforme demonstrado na Figura 21. Em seguida, realizou-se três deslocamentos para cada motor, obtendo-se uma média para realizar a calibração do motor.

Figura 21 – Imagem da calibração dos motores de passo para demonstração dos procedimentos metodológicos.



Fonte: Autor, 2020.

Para realizar a calibração da velocidade dos motores, fixou-se a velocidade angular em 100 passos por minuto durante uma rotação. Com o auxílio de um cronômetro, mediu-se a duração que o motor precisa para completar este processo. Realizou-se três testes a fim de obter uma média para calibrar a velocidade em cada eixo do porta-amostras do equipamento de plasma atmosférico.

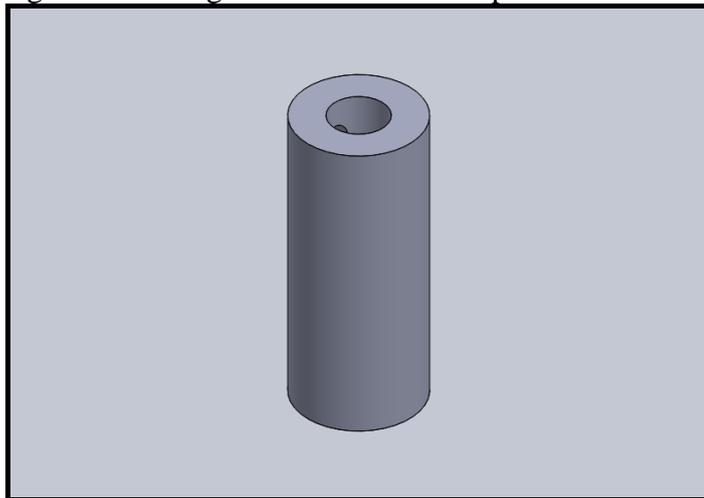
## 5 RESULTADOS E DISCUSSÕES

### 5.1 ACOPLADORES E SUPORTES

Ao analisar o porta-amostras do equipamento de plasma, observou-se que é necessário realizar uma adaptação nos *charriots* do equipamento para acoplar os eixos nos motores de passos e fabricar dois suportes para realizar a fixação dos motores.

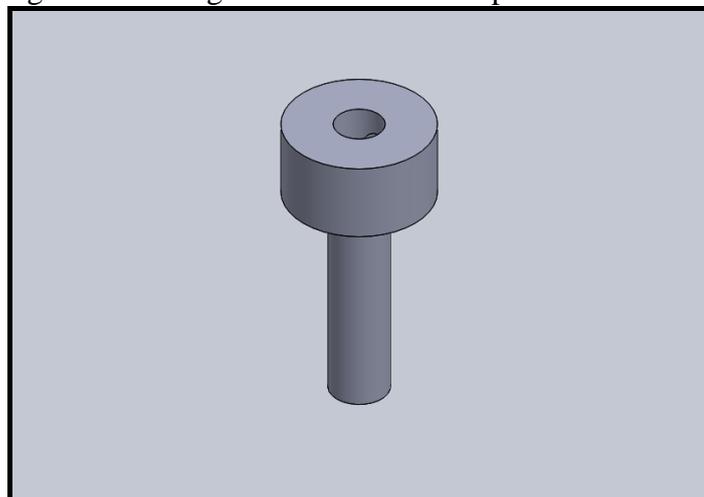
Desta forma, projetou-se dois acopladores e dois suportes, um para cada eixo, representados pelas Figura 22 até a Figura 25. Os projetos dos dois acopladores e dos dois suportes se encontram nos anexos A, B, C e D deste trabalho e disponível em: <https://drive.google.com/drive/folders/1k0A4O0B5OsKlQweZBcFM-2Ck-UGI8ZuG?usp=sharing>.

Figura 22 – Imagem ilustrativa do acoplador do eixo X.



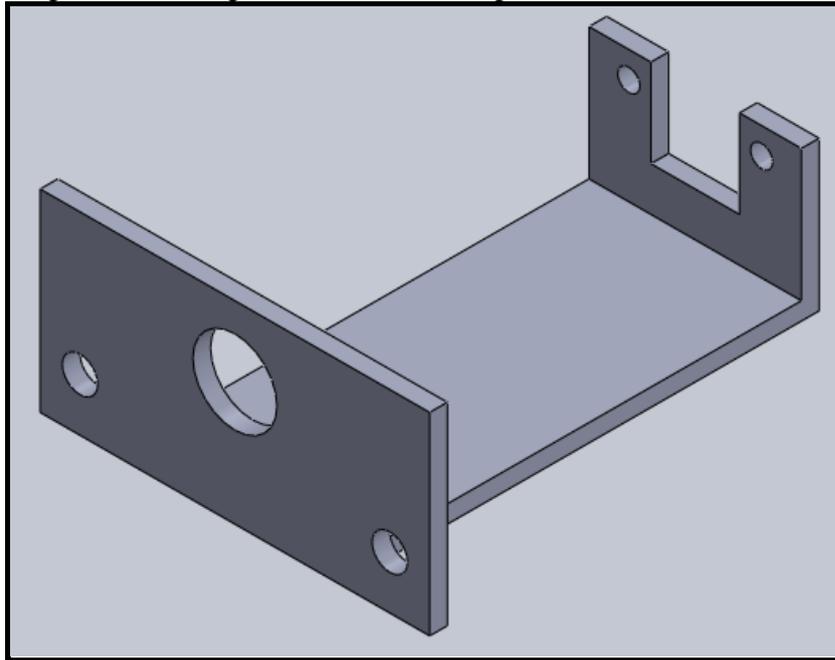
Fonte: Autor, 2019.

Figura 23 – Imagem ilustrativa do acoplador do eixo Y.



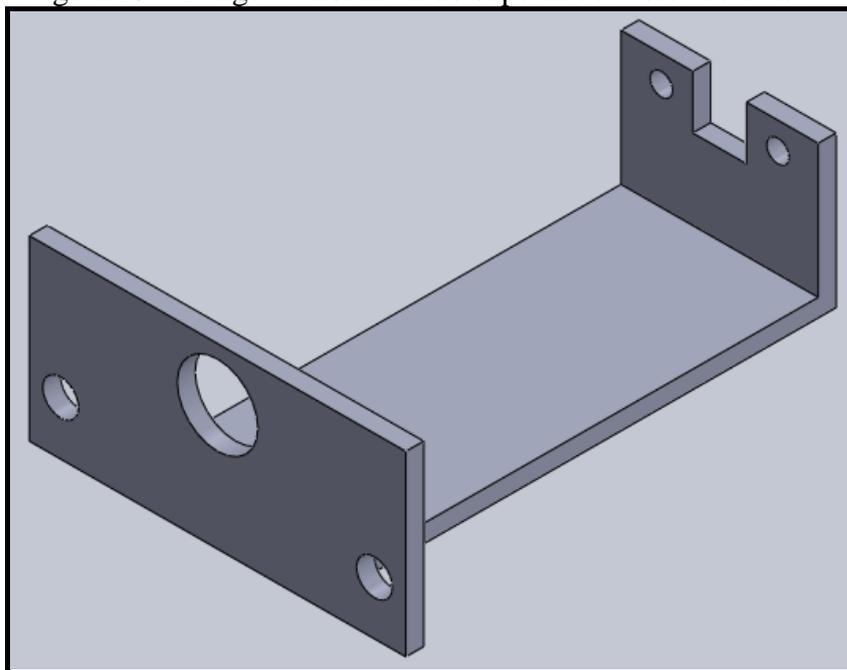
Fonte: Autor, 2019.

Figura 24 – Imagem ilustrativa do suporte do motor do eixo X.



Fonte: Autor, 2020.

Figura 25 – Imagem ilustrativa do suporte do motor do eixo Y.

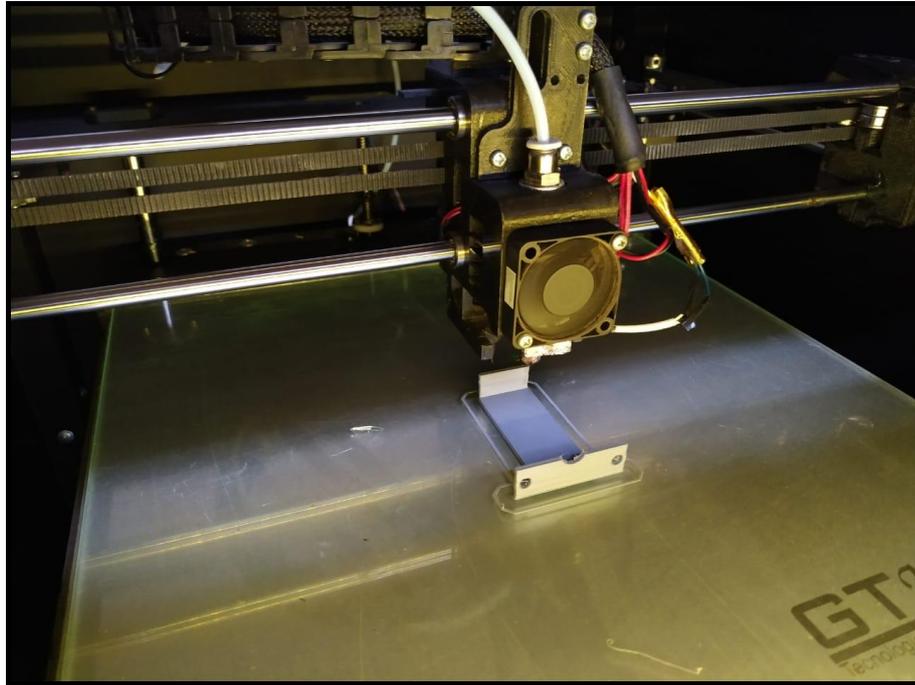


Fonte: Autor, 2020.

Para confeccionar os dois acopladores, empregou-se algumas operações de usinagem como a furação, lixamento e rosqueamento.

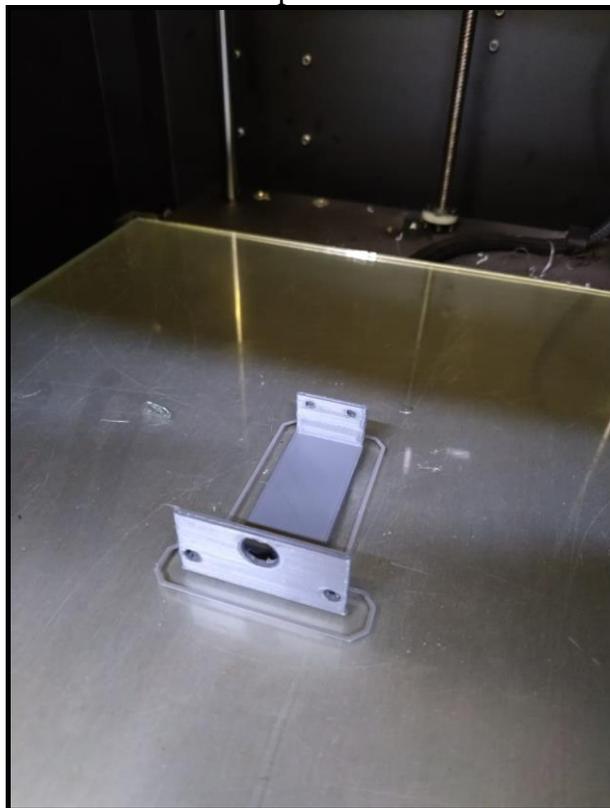
Por outro lado, para a fabricação dos dois suportes utilizou-se o processo aditivo de impressora 3D (Dimensões). A Figura 26 e a Figura 27 mostram o processo de impressão 3D do suporte do eixo Y.

Figura 26 – Imagem do processo de impressão 3D do suporte do motor de passo do eixo Y, ilustrando o uso da manufatura aditiva no processo de fabricação do sistema XY do porta-amostras.



Fonte: Autor, 2020.

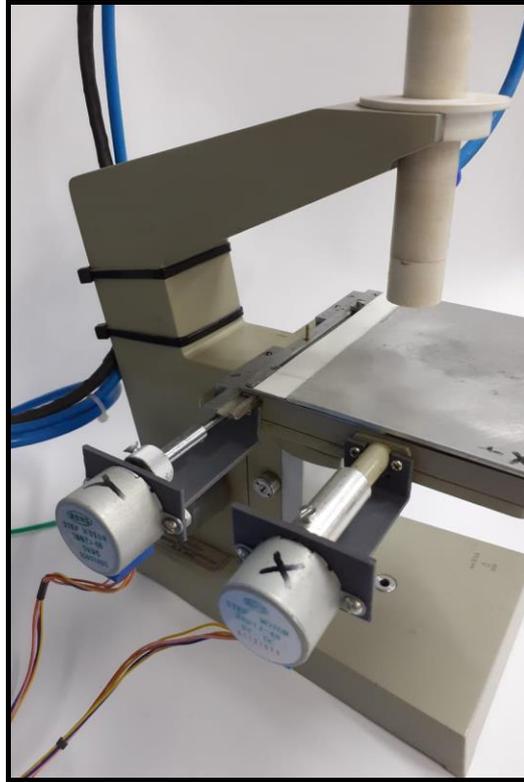
Figura 27 – Imagem do suporte do motor de passo do eixo Y fabricado por processo aditivo em impressão 3D.



Fonte: Autor, 2020.

Após a fabricação das quatro peças, elas foram montadas no sistema de plasma atmosférico conforme a Figura 28.

Figura 28 – Imagem do porta-amostras do equipamento de plasma atmosférico ilustrando o posicionamento dos motores de passo e seus respectivos suportes e acopladores no equipamento.

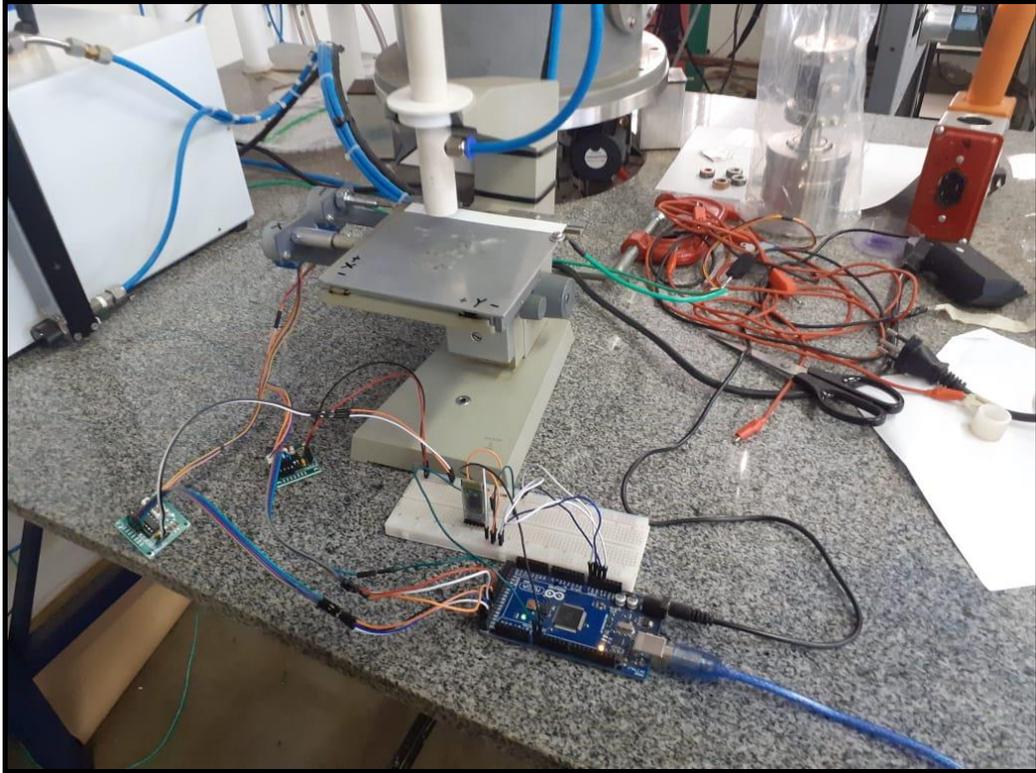


Fonte: Autor, 2020.

## 5.2 ARDUINO® E LIGAÇÕES ELETRÔNICAS

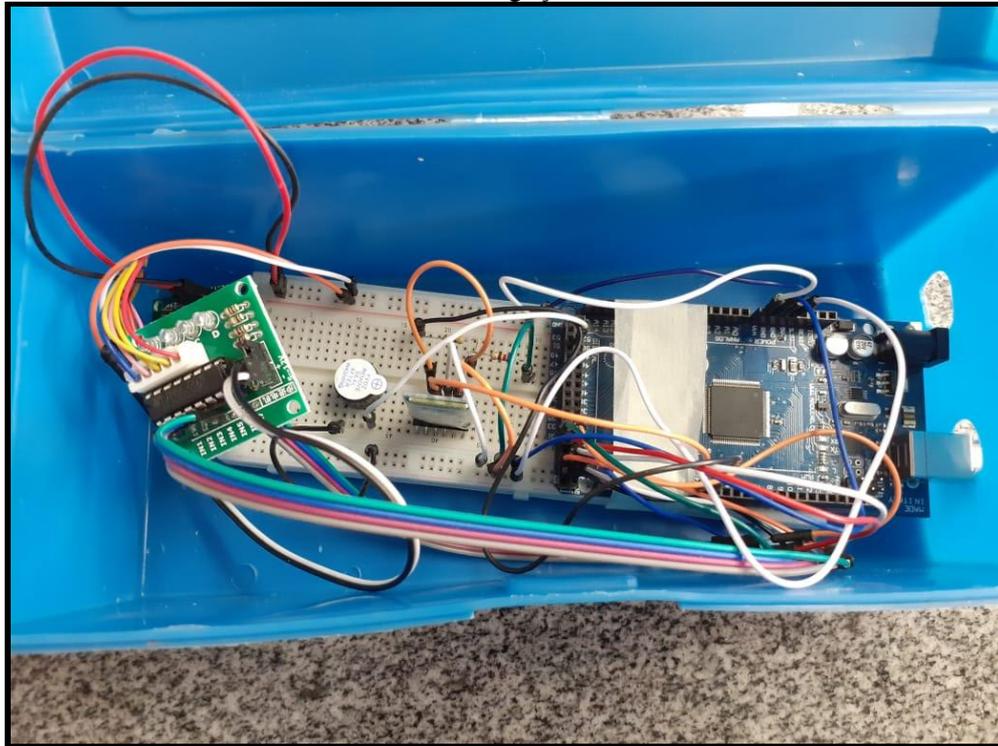
Primeiramente fez-se todas as ligações eletrônicas dos motores de passo, *drivers* de acionamento, módulo *Bluetooth*, *buzzer* de acordo com a Figura 17. Para proteger as ligações eletrônicas, colocou-se o circuito com o Arduino® dentro de uma caixa, isolando assim do meio externo. O processo de montagem está representado pela Figura 29 até a Figura 31

Figura 29 – Imagem da montagem do sistema de controle e automação com Arduino®, motores de passo, *drivers* de acionamento, módulo *Bluetooth* e *buzzer* ilustrando a composição do projeto mecânico e eletrônico.



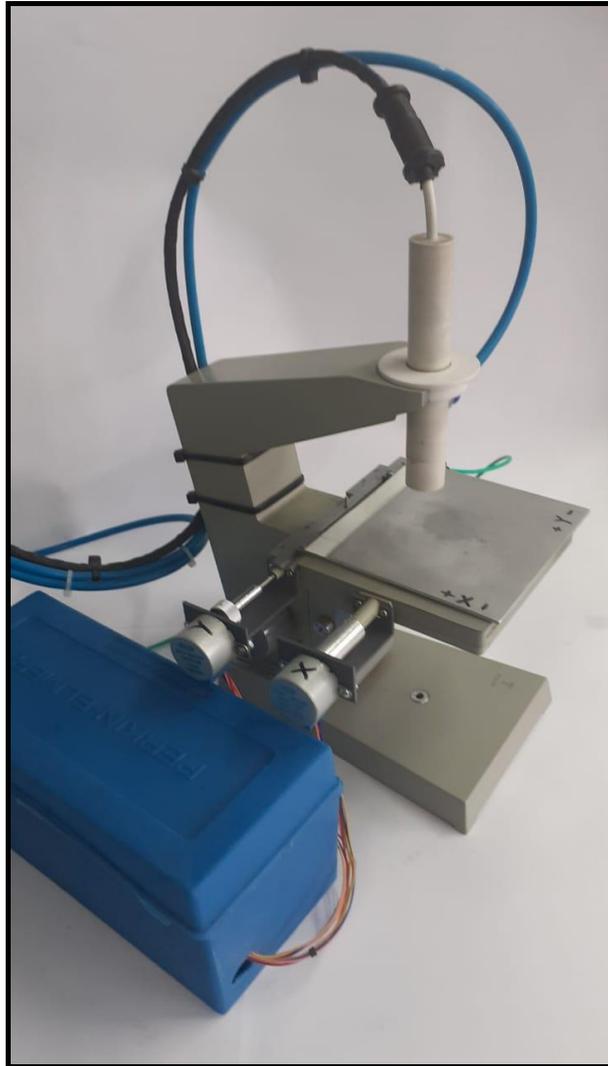
Fonte: Autor, 2020.

Figura 30 – Imagem das ligações eletrônicas do sistema de controle e automação com Arduino®, motores de passo, *drivers* de acionamento, módulo *Bluetooth* e *buzzer* ilustrando o detalhamento das ligações eletrônicas.



Fonte: Autor, 2020.

Figura 31 – Imagem da montagem final do sistema de controle e automação com Arduino®, motores de passo, *drivers* de acionamento, módulo *Bluetooth* e *buzzer* ilustrando para demonstração de seu *layout*.



Fonte: Autor, 2020.

Foi desenvolvido o algoritmo no Arduino® para receber todos os dados enviados pelo aplicativo; tratar esses dados; emitir um sinal sonoro antes de executar a operação com os motores; executar os movimentos desejados pelo usuário de acordo com a Figura 12 até a Figura 16; e finalizar com um sinal sonoro. O Algoritmo se encontra no anexo E deste trabalho e disponível em:

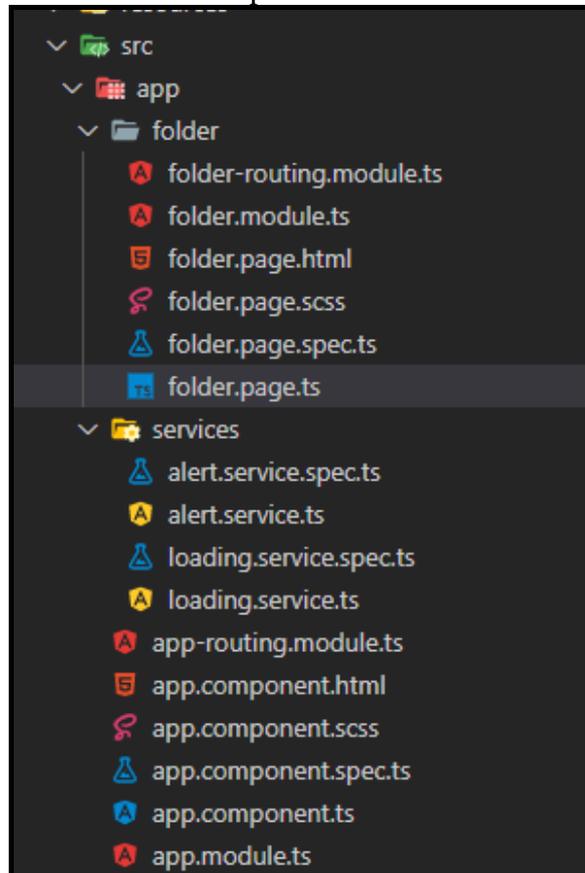
[https://github.com/felipenascimento147/Plasma\\_bluetooth/blob/main/bluetooth](https://github.com/felipenascimento147/Plasma_bluetooth/blob/main/bluetooth).

### 5.3 APLICATIVO

Criou-se arquivos e pastas com as novas funcionalidades do aplicativo dentro da pasta *src* do *template*. Entre os principais arquivos estão o HTML, CSS e JavaScript que são

responsáveis por criar conteúdo da página, estilizar e executar funções, respectivamente (GUANABARA, 2019). A Figura 32 mostra os arquivos e pastas dentro da pasta src.

Figura 32 – Imagem dos arquivos da pasta src para demonstração dos tipos de arquivos do aplicativo.



Fonte: Autor, 2020.

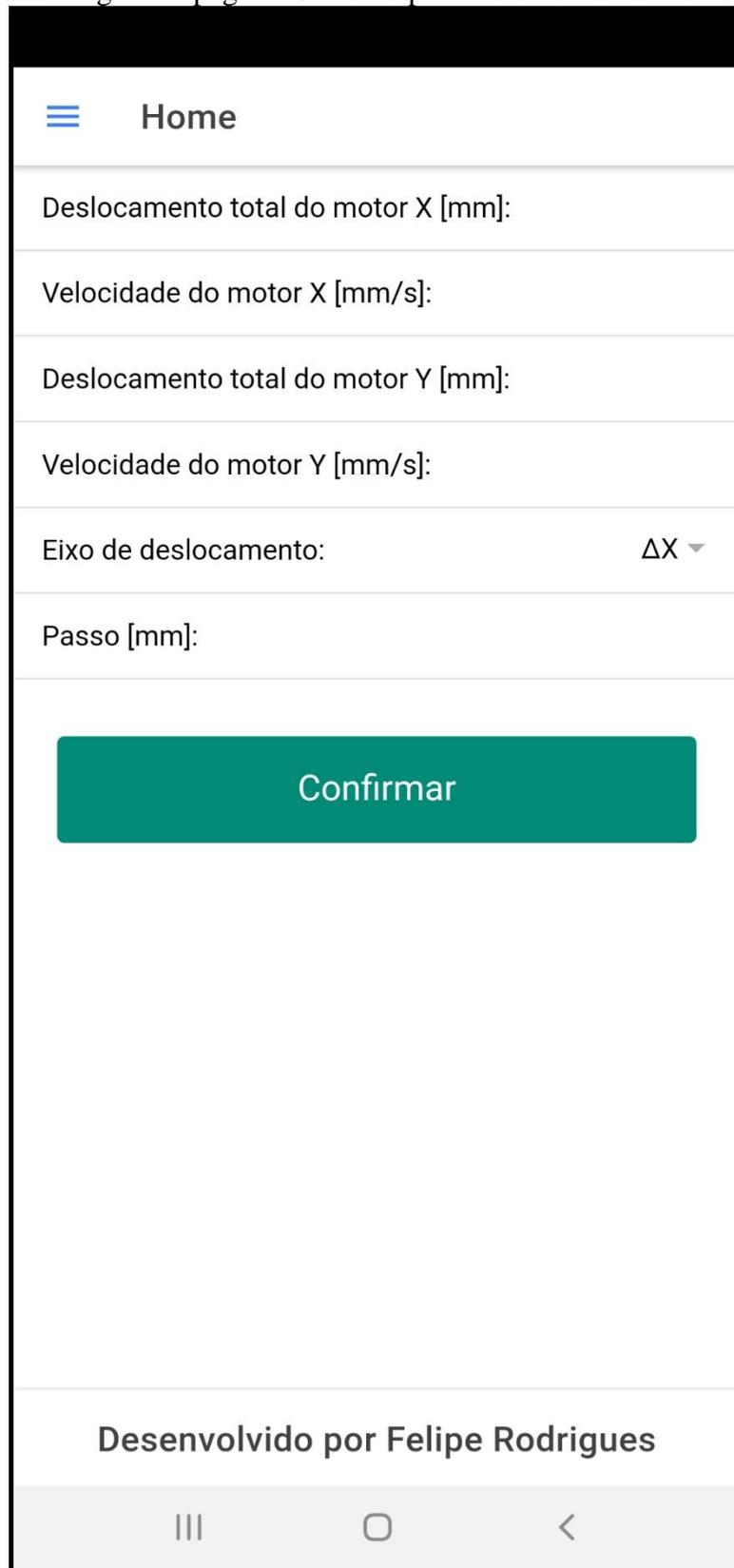
Como o principal objetivo do aplicativo é o envio de dados para o Arduino®, e o microcontrolador por sua vez executa comandos para os motores de passo realizarem o movimento varrendo uma região do espaço, o aplicativo possui um formulário HTML dentro do arquivo folder.page.html. Neste formulário, o usuário preenche as posições e velocidades de cada motor, a variação do passo e qual motor realizará essa variação de passo na página *home* e envia as informações para o Arduino® via comunicação *Bluetooth*.

Além disso, o JavaScript deverá validar os campos do formulário para as seguintes situações:

- Os campos não podem ser vazios;
- Os valores de velocidade não podem ser negativos;
- Caso o usuário digitar o deslocamento de um motor, a velocidade do motor respectivo deve ser positiva;

- Caso o usuário digitar a velocidade de um motor, o deslocamento do motor respectivo deve ser diferente de zero;
- O valor do deslocamento do motor X deve ser entre  $-30$  mm e  $30$  mm devido ao tamanho da mesa do equipamento de plasma atmosférico;
- A velocidade do motor X deve ser entre  $0,3$  mm/s e  $3$  mm/s, pois esse intervalo apresentou melhor eficiência operacional do motor;
- O valor do deslocamento do motor Y deve ser entre  $-30$  mm e  $30$  mm devido ao tamanho da mesa do equipamento de plasma atmosférico;
- A velocidade do motor Y deve ser entre  $0,1$  mm/s e  $0,7$  mm/s, pois esse intervalo apresentou melhor eficiência operacional do motor;
- Caso o usuário selecione o eixo de deslocamento para o motor X ( $\Delta X$ ), o passo deve ser maior que zero, menor ou igual ao deslocamento do motor X e divisível pelo deslocamento do motor X;
- Caso o usuário selecione o eixo de deslocamento para o motor Y ( $\Delta Y$ ), o passo deve ser maior que zero, menor ou igual ao deslocamento do motor Y e divisível pelo deslocamento do motor Y.

Essas validações estão dentro do arquivo `folder.page.ts`. A Figura 33 mostra a página *home* do aplicativo desenvolvido neste estudo e; a Figura 34 mostra o menu lateral e o *select* do formulário para informar qual é o eixo de deslocamento.

Figura 33 – Imagem da página *home* do aplicativo desenvolvido neste estudo.

☰ Home

Deslocamento total do motor X [mm]:

Velocidade do motor X [mm/s]:

Deslocamento total do motor Y [mm]:

Velocidade do motor Y [mm/s]:

Eixo de deslocamento:  $\Delta X$  ▾

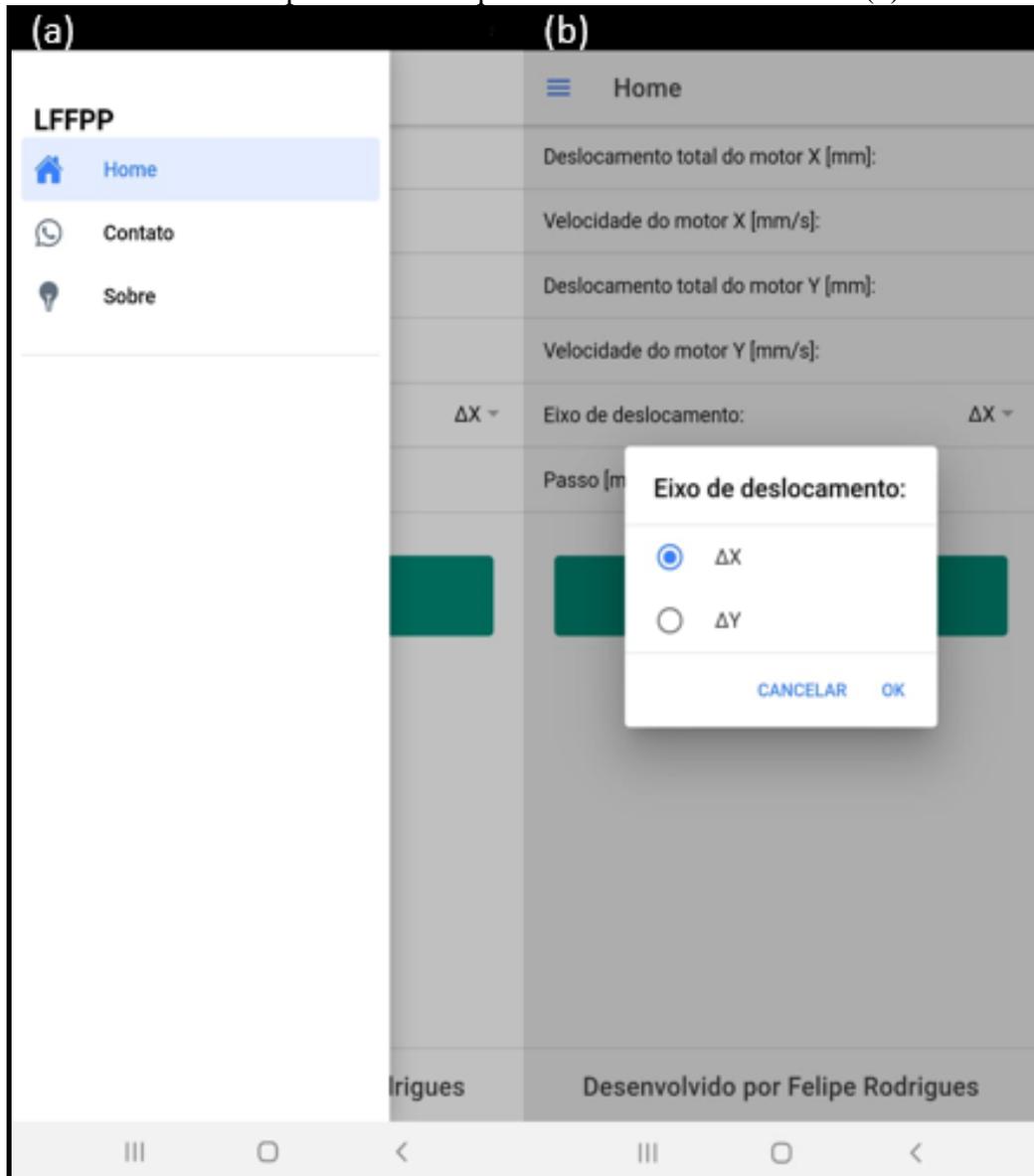
Passo [mm]:

Confirmar

Desenvolvido por Felipe Rodrigues

Fonte: Autor, 2020.

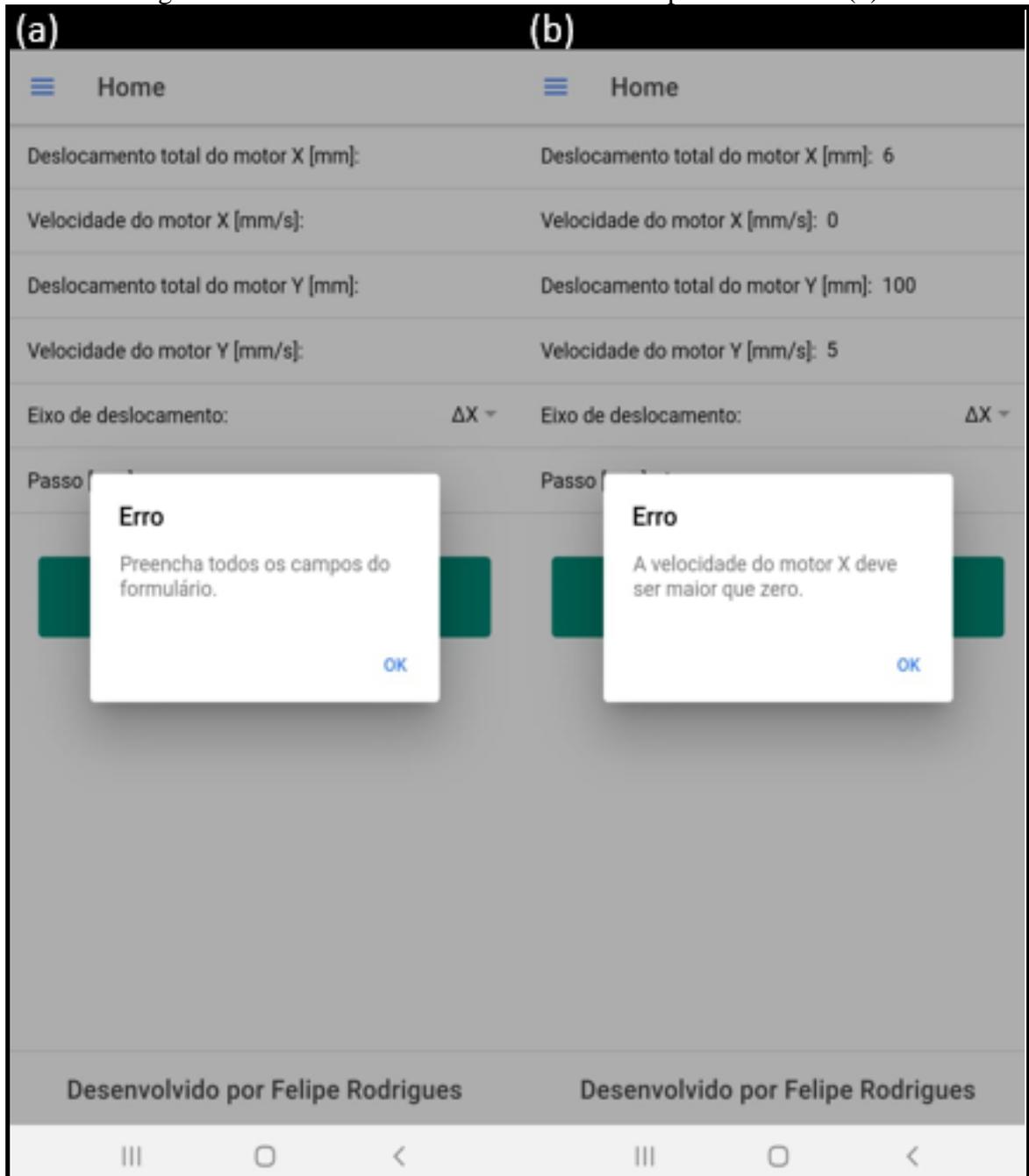
Figura 34 – Imagem do menu lateral do aplicativo desenvolvido neste estudo (a) e *select* do formulário para informar qual é o eixo de deslocamento (b).



Autor, 2020.

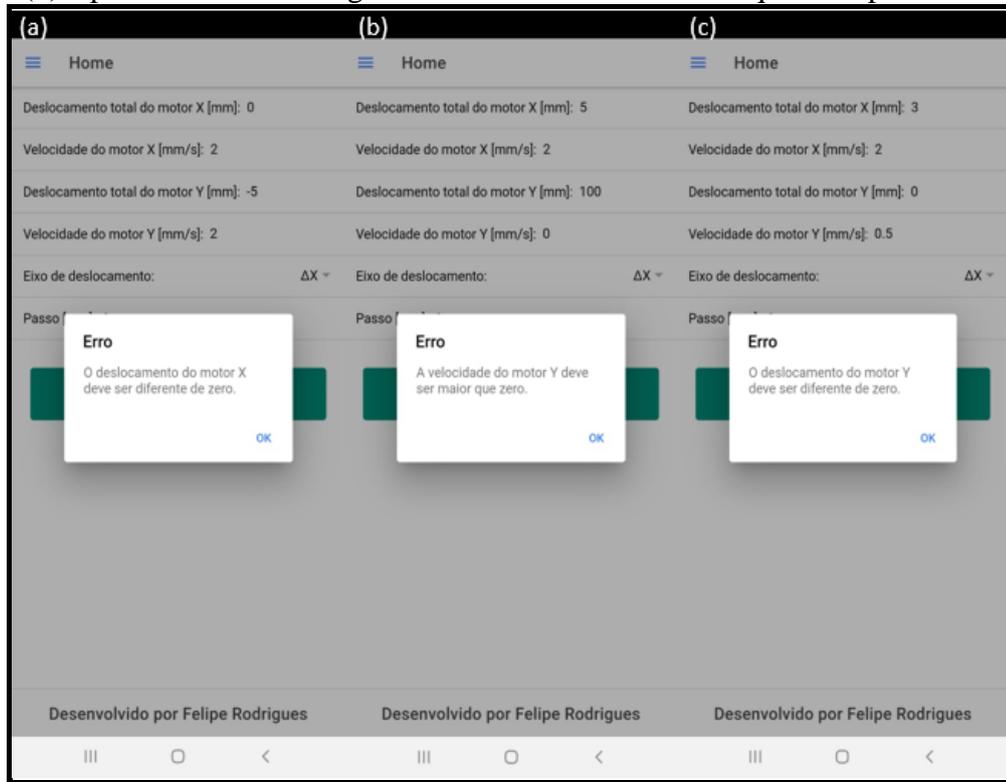
O *service alert* é um serviço responsável por mostrar mensagens de erro caso as validações nos campos do formulário na página *home* ou erros de conexão *Bluetooth* sejam verdadeiras. A Figura 35 até a Figura 38 mostram as mensagens geradas por preenchimento incorreto do formulário; e a Figura 39 e mostra as mensagens devido a erros de conexão *Bluetooth*.

Figura 35 - Imagem da mensagem para campos do formulário vazios (a) e para velocidade igual a zero e deslocamento diferente de zero para o motor X (b).



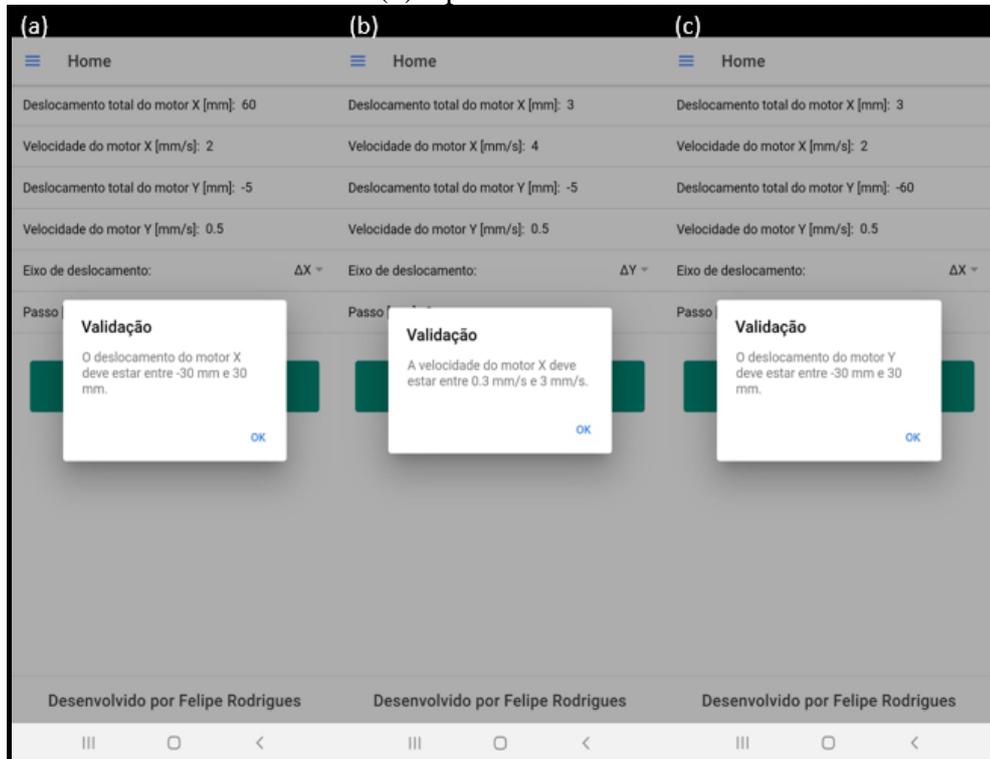
Fonte: Autor, 2020.

Figura 36 - Imagem da mensagem para deslocamento igual a zero e velocidade maior que zero para o motor X (a), para velocidade igual a zero e deslocamento diferente de zero para o motor Y (b) e para deslocamento igual a zero e velocidade maior que zero para o motor Y (c).



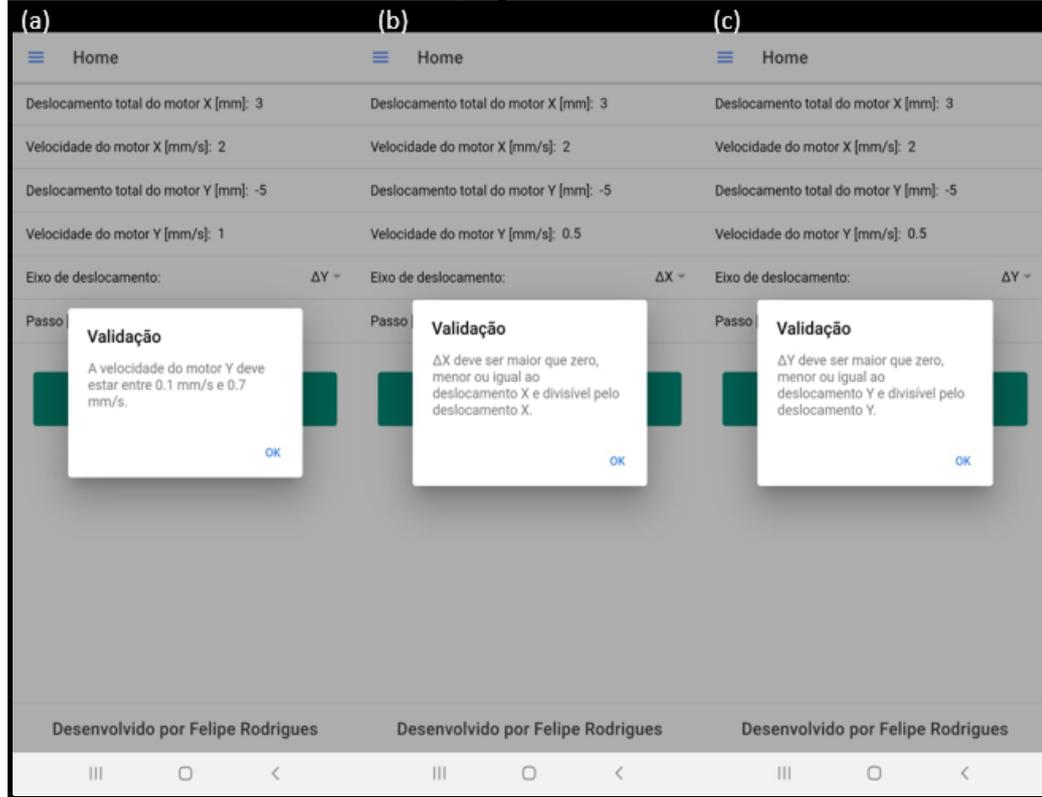
Fonte: Autor, 2020.

Figura 37 – Imagem da mensagem para os limites do deslocamento do motor X (a), para os limites da velocidade do motor X (b) e para os limites do deslocamento do motor Y (c).



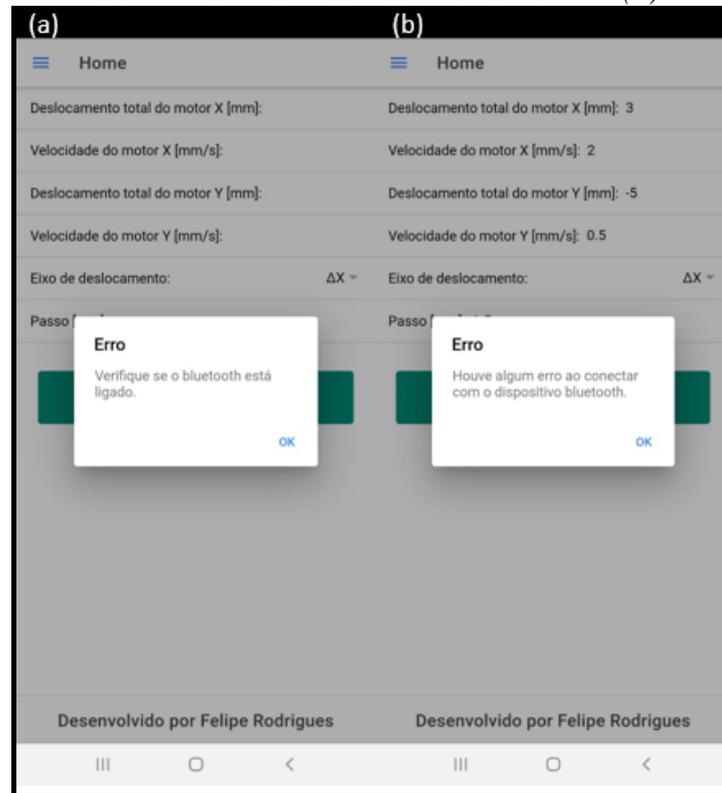
Fonte: Autor, 2020.

Figura 38 – Imagem da mensagem para os limites da velocidade do motor Y (a), para o deslocamento do eixo do motor X (b) e para o deslocamento do eixo do motor Y (c).



Fonte: Autor, 2020.

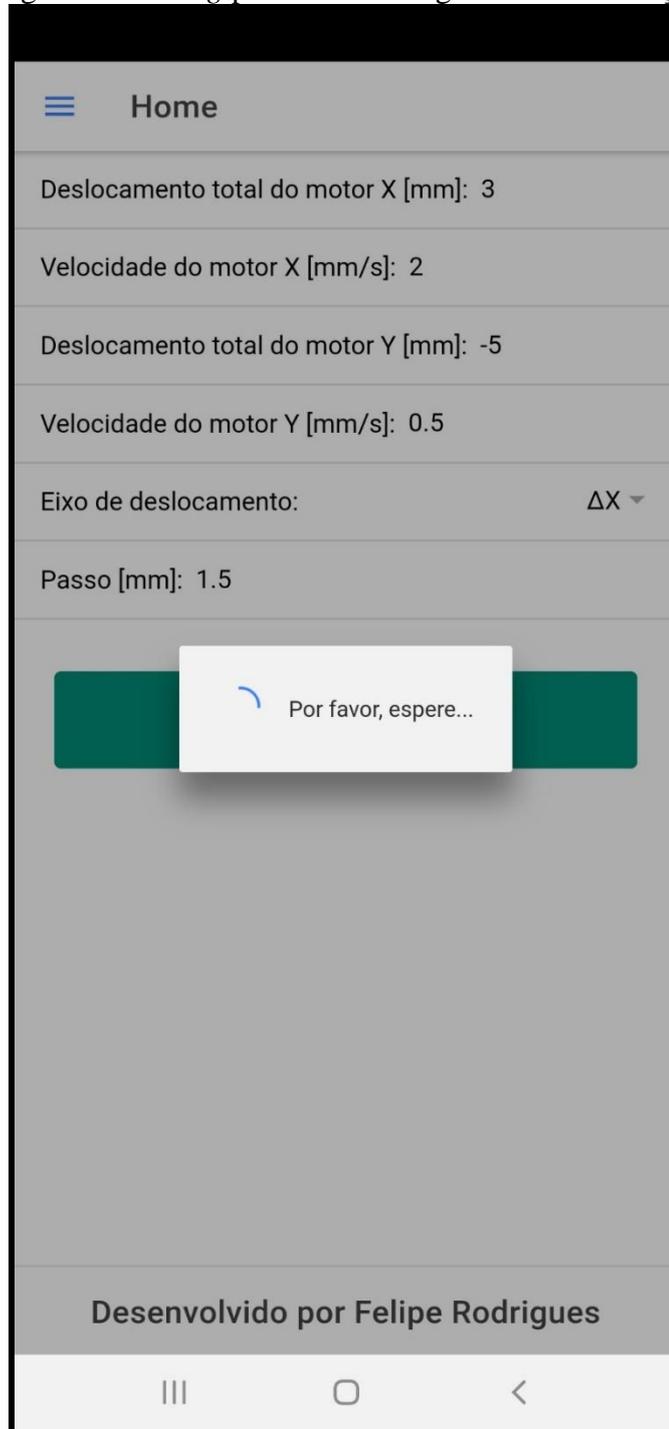
Figura 39 – Imagem da mensagem para verificar se o *Bluetooth* do celular está ligado (a) e erro ao conectar com o módulo *Bluetooth* (b).



Fonte: Autor, 2020.

O *service loading* é um serviço responsável por mostrar o *loading* para o usuário esperar o envio das informações do formulário. A Figura 40 representa esse processo.

Figura 40 – Imagem do *loading* para o usuário aguardar a finalização do processo.



Fonte: Autor, 2020.

Também se criou as páginas sobre e contato, assim os usuários do aplicativo tem acesso ao resumo do aplicativo e poderão entrar em contato com o desenvolvedor para tirar dúvidas através de *e-mail* e aplicativo WhatsApp conforme a Figura 41.

Figura 41 – Imagem sobre as informações do aplicativo (a) e sobre as informações de contato do desenvolvedor (b).



Fonte: Autor, 2020.

Por conta da quantidade de arquivos e pastas; e o tamanho de memória ocupado pelo código, ele foi compartilhado na plataforma do GitHub e está disponível em <https://github.com/felipenascimento147/stepperMotorControlApp>, desta maneira, os visitantes têm acesso ao código de forma *online* ou podem realizar o *download* dele.

Disponibilizou-se a versão APK (*Android Application Pack*) do aplicativo para Android com o nome de `app.apk` em: <https://drive.google.com/drive/folders/1k0A4O0B5OsKIQweZBcFM-2Ck-UGI8ZuG?usp=sharing>. Deste modo, os usuários podem baixar e instalar o aplicativo no *smartphone* com sistema operacional Android. Para colocar o aplicativo na loja Google Play (Android) de forma vitalícia, é necessário investir 25 dólares. Para colocar o aplicativo na loja App Store (iOS), é necessário investir 100 dólares anuais.

Também foi disponibilizado um vídeo mostrando o funcionamento do sistema de plasma atmosférico com o aplicativo em: <https://drive.google.com/drive/folders/1k0A4O0B5OsKlQweZBcFM-2Ck-UGI8ZuG?usp=sharing>.

#### 5.4 CALIBRAÇÃO DOS MOTORES

Após realizar a montagem da Figura 21 no porta-amostras do equipamento de plasma atmosférico, realizou-se três experimentos com uma rotação completa para a calibração do motor X. Cada procedimento gerou um deslocamento da mesa, desta forma foi medido o deslocamento com um paquímetro e em seguida calculou-se uma média aritmética dos valores para realizar a calibração do motor X. Os valores obtidos nos experimentos estão dispostos na Tabela 1.

Tabela 1 – Deslocamento do motor X para 1 volta do motor medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21.

Experimento	Deslocamento [mm]
Teste 1	14,63
Teste 2	15,02
Teste 3	15,21
Média	14,95

Fonte: Autor, 2020.

A calibração do deslocamento do motor X é relacionado de acordo com a Equação (2).

$$X \text{ [passos]} = \frac{\text{Valor que o usuário deseja [mm]} * 2048 \text{ [passos]}}{14,95 \text{ [mm]}} \quad (2)$$

Do mesmo modo, executou-se o processo para o motor do Y. Os valores obtidos nos experimentos estão dispostos na Tabela 2.

Tabela 2 – Deslocamento do motor Y para 1 volta do motor medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21.

Experimento	Deslocamento [mm]
Teste 1	3,65
Teste 2	3,40
Teste 3	3,60
Média	3,55

Fonte: Autor, 2020.

A calibração do deslocamento do motor Y é relacionado de acordo com a Equação (3).

$$Y [\text{passos}] = \frac{\text{Valor que o usuário deseja [mm]} * 2048 [\text{passos}]}{3,55 [\text{mm}]} \quad (3)$$

Com um dos motores de passo, mediu-se o tempo necessário para o motor completar uma volta a 100 passos por minuto. Esse experimento foi feito 3 vezes e em seguida foi realizado a média aritmética. Os valores obtidos neste procedimento estão dispostos na Tabela 3.

Tabela 3 – Tempo para o motor de passo completar uma volta a 100 passos por minuto medidos com o auxílio das marcações descritas na seção 4.4 CALIBRAÇÃO DOS MOTORES e Figura 21.

Experimento	Tempo [s]
Teste 1	19,21
Teste 2	19,29
Teste 3	19,17
Média	19,22

Fonte: Autor, 2020.

Com isso, a calibração da velocidade do motor X e Y é relacionado de acordo com a Equação (4) e Equação (5), respectivamente.

$$X [ppm] = \frac{\text{Valor que o usuário deseja [mm/s]} * 100 [ppm] * 19,22 [s]}{14,95 [mm]} \quad (4)$$

$$Y [ppm] = \frac{\text{Valor que o usuário deseja [mm/s]} * 2048[\text{passos}] * 19,22 [s]}{3,55 [mm]} \quad (5)$$

Sendo que a unidade ppm é passos por minuto.

## 5.5 INVESTIMENTO DO PROJETO

Devido a uma necessidade de segurança operacional do porta-amostras do equipamento de plasma, foi desenvolvido esse projeto para evitar acidentes com o choque elétrico. Desse modo, foi investido R\$191,34 em materiais e frete para a entrega dos mesmos. Os valores de cada item detalhado foram dispostos na Tabela 4.

Tabela 4 - Relação de preços dos itens adquiridos e utilizados no projeto aqui proposto.

Item	Quantidade	Preço	Preço final
Arduino© Mega 2560 r3	1	R\$37,71	R\$37,71
Conjunto motor de passo 28BYJ-48 e <i>driver</i> ULN2003	2	R\$ 8,20	R\$16,40
Frete	1	R\$57,83	R\$57,83
Impressão 3D do suporte do motor do eixo X	1	R\$8,00	R\$8,00
Impressão 3D do suporte do motor do eixo Y	1	R\$8,00	R\$8,00
Módulo <i>Bluetooth</i> HC-05	1	R\$44,50	R\$44,50
<i>Buzzer</i> ativo	1	R\$1,90	R\$1,90
Resistores e <i>jumpers</i>	-	R\$2,00	R\$2,00
Fonte 12 Volts	1	R\$15,00	R\$15,00
Valor total	-	-	R\$191,34

Fonte: Autor, 2020.

Ressalta-se que esse investimento de R\$191,34 foi referente apenas a adaptação do equipamento existente. O equipamento de plasma atmosférico foi avaliado em R\$800,00 (ROGÉRIO, 2019). Desse modo, a construção de um novo protótipo com a adaptação será de R\$991,34.

## 6 CONCLUSÕES

Neste trabalho desenvolveu-se um sistema para automatizar e controlar o deslocamento XY do porta-amostras do equipamento de plasma atmosférico do LFFPP da UFTM utilizando as tecnologias e *frameworks* Arduino®, Ionic®, *Bluetooth* e motores de passo. O desenvolvimento foi dividido em duas etapas: o *hardware* e o *software*.

O *hardware* do sistema é composto pela fabricação de suportes e acopladores para a fixação e transmissão de movimento dos motores de passo para a mesa do equipamento. Para a confecção dessas peças utilizou-se processos de fabricação aditiva e usinagem. Ademais, montou-se o circuito eletrônico com o Arduino®, dois motores de passo 28BYJ com seus *drivers* de acionamento, módulo *Bluetooth*, dois resistores e um *buzzer*.

Na etapa do *software* desenvolveu-se dois algoritmos para realizar o controle e automação do processo. O primeiro *software* é o programa elaborado no Arduino® para que o microcontrolador execute os movimentos desejados; já o segundo *software* é o aplicativo desenvolvido em Ionic® que envia informações para o Arduino® através da comunicação *Bluetooth*. Disponibilizou-se o aplicativo em nuvem na versão APK para Android, pois sua publicação nas lojas Google Play e App Store necessitaria de um investimento de 25 dólares e 100 dólares, respectivamente. Ressalta-se que para o aplicativo ser publicado nas lojas, o algoritmo passa por um processo de otimização, deste modo melhorando o uso dos processadores e memória dos *smartphones*.

O processo inicia-se com o preenchimento do formulário com posições e velocidades desejadas pelo usuário no aplicativo. Em seguida o aplicativo envia essas informações para o Arduino® através da comunicação *Bluetooth* e o microcontrolador por sua vez executa o movimento com o acionamento dos motores de passo. Vale ressaltar que antes de iniciar e ao finalizar o movimento dos motores, o *buzzer* é acionado para alertar o usuário que o sistema entrará em operação.

O sistema desenvolvido neste trabalho teve um investimento de R\$191,34, mediante ao benefício de segurança operacional e maior qualidade no processo, o projeto é extremamente viável.

O projeto é considerado inovador, pois apresenta uma proposta inédita para a solução de acidentes envolvendo choque elétrico no porta-amostras do equipamento de plasma

atmosférico. Ademais, por se tratar do desenvolvimento de *software*, o algoritmo no Arduino© e o aplicativo em Ionic© se classificam no registro de *software* do INPI (Instituto Nacional da Propriedade Industrial).

Em relação a trabalhos futuros, é recomendado a publicação do aplicativo para as lojas Google Play e App Store, além disso o pesquisador poderá criar uma aplicação *Back End* com banco de dados para realizar o cadastro de usuários e identificar os usuários que acionaram o sistema de plasma no decorrer do tempo. Outra sugestão é o pesquisador realizar testes com sistema de plasma para várias velocidades e variação de passo para verificar qual velocidade é ideal para tratamento de superfícies de materiais e filmes finos comumente estudados no Laboratório de Filmes Finos e Processos a Plasma da UFTM e na área de Ciências dos Materiais.

## REFERÊNCIAS

AFONSO, Antônio Pereira; FILONI, Ênio. **Eletrônica: Circuitos eletrônicos**. [S. l.: s. n.], 2011. Disponível em: <https://www.passeidireto.com/arquivo/19811249/eletronica-vol-1-circuitos-eletricos-livro-completo>. Acesso em: 08/11/ 2020.

ALVES, C.; GUERRA, C.L.B. Neto; MORAIS, G.H.S.; SILVA, C.F. da; HAJEK, V. *Nitriding of titanium disks and industrial dental implants using hollow cathode discharge*. **Surface and Coatings Technology**, [S. l.], p. 196-202, 15/03/2006. DOI <https://doi.org/10.1016/j.surfcoat.2004.10.009>. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0257897204009843>. Acesso em: 27/02/2021.

ARDUINO©. Arduino© *Glide*. 2019. Disponível em: <https://www.arduino.cc>. Acesso em: 25/08/2019.

BONATTO, Aurélio; CANTO, Diego Oliveira do. **BLUETOOTH TECHNOLOGY** (IEEE 802.15). **Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)**, [s. l.], 2007. Disponível em: <https://www.inf.pucrs.br/~cnunes/redes/Trabalho%20Bluetooth.pdf>. Acesso em: 08/10/2020.

BRITES, Felipe G.; SANTOS, Vinicius P. de A. **Motor de Passo**. Universidade Federal Fluminense, Grupo PET-Tele, Niterói, 2008.

BUSO, Rafael Rocha. **Desenvolvimento e aplicação de um equipamento gerador de jato de plasma frio à pressão atmosféricas para tratamento de materiais**. Orientador: Rogério Valentim Gelamo. 2019. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2015.

CARVALHO, Flávio Henrique Oliveira. **Síntese de nanoestruturas de carbono utilizando plasmas frios**. Orientador: Rogério Valentim Gelamo. 2013. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2013. Disponível em: <http://bdtd.uftm.edu.br/handle/tede/76>. Acesso em: 02/03/2020.

CARVALHO, Renata Gomes. **Deposição e caracterização de filmes finos de NbAlN por magnetron sputtering reativo**. Orientador: Eduardo Kirinus Tentardini. 2016. Dissertação (Programa de Pós-Graduação em Ciências e Engenharia de Materiais da Universidade Federal

de Sergipe) - Universidade Federal de Sergipe, [S. l.], 2016. Disponível em: [https://ri.ufs.br/bitstream/riufs/3526/1/RENATA\\_GOMES\\_CARVALHO.pdf](https://ri.ufs.br/bitstream/riufs/3526/1/RENATA_GOMES_CARVALHO.pdf). Acesso em: 20/12/2020.

CONTIJO, José Marcelo. **FABRICAÇÃO E CARACTERIZAÇÃO DE SUPERCAPACITORES FLEXÍVEIS COM ELETRODOS COMPOSTOS POR MULTICAMADAS DE GRAFENO E FILMES FINOS DE ÓXIDO METÁLICO**. Orientador: Rogério Valentim Gelamo. 2018. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2018.

D'AUSILIO, Alessandro. *Arduino: A low-cost multipurpose lab equipment*. **Behavior Research Methods**, [S. l.], 25 out. 2011. DOI <https://doi.org/10.3758/s13428-011-0163-z>. Disponível em: <https://link.springer.com/article/10.3758/s13428-011-0163-z>. Acesso em: 25/08/2019.

DAMAS, Antonio Méndez Montoro de. *Plasmas de Descarga de Barrera Dielétrica con Empaquetamiento Ferroeléctrico a Presión Atmosférica Aplicados a la Producción de Hidrógeno, Amoníaco y a la Descontaminación de Aire*. Orientador: Carlos Soria del Hoyo. 2017. Tese de Doutorado (Instituto de Ciencia de Materiales de Sevilla) - Universidad de Sevilla, [S. l.], 2017. Disponível em: [https://idus.us.es/bitstream/handle/11441/71159/Tesis\\_Antonio%20Me%20ndez\\_Dic17.pdf?sequence=1](https://idus.us.es/bitstream/handle/11441/71159/Tesis_Antonio%20Me%20ndez_Dic17.pdf?sequence=1). Acesso em: 20/12/2020.

FIEBIG, Rafael. **Protótipo de um CNC Router**. 2018. Trabalho de conclusão de curso (Engenharia Elétrica) - Universidade de Passo Fundo, Passo Fundo - RS, 2018.

GITHUB. [S. l.], 19/05/2020. Disponível em: <https://github.com/>. Acesso em: 19/05/2020.

GRÉGIO, André Ricardo Abed. *Tecnologia Bluetooth e Aspectos de Segurança*. **Instituto de Computação Unicamp**, [s. l.], 2009. Disponível em: <https://www.ic.unicamp.br/~ducatte/mo401/1s2009/T2/079779-t2.pdf>. Acesso em: 08/10/2020.

GRIFFITH, Chris. *Mobile App Development with Ionic*. *United States of America*: O'Reilly, 18/08/2017.

GUANABARA, Gustavo. *What is JavaScript capable of doing? - JavaScript Course #01*. [S. l.], 27/05/2019. Disponível em:

[https://www.youtube.com/watch?v=Ptbk2af68e8&list=PLHz\\_AreHm4dlsK3Nr9GVvXCbpQyHQ11o1&t=0s](https://www.youtube.com/watch?v=Ptbk2af68e8&list=PLHz_AreHm4dlsK3Nr9GVvXCbpQyHQ11o1&t=0s). Acesso em: 28/04/2020.

HUANG, Albert; RUDOLPH, Larry. *Bluetooth for Programmers*. [S. l.: s. n.], 2005. Disponível em: <http://people.csail.mit.edu/rudolph/Teaching/Articles/BTBook.pdf>. Acesso em: 08/10/2020.

IONIC©. Ionic© Docs. 2020. Disponível em: <https://ionicframework.com/docs>. Acesso em: 27/02/2020.

KIATRONICS. **28BYJ-48 – 5V Stepper Motor**. [S. l.], 2019. Disponível em: <https://blog.eletrogate.com/wp-content/uploads/2018/07/Motor28BYJ48Kiatronics.pdf>. Acesso em: 29/11/2019.

LIMA, Fernando Fortunato de. **Avaliação de Frameworks para o Desenvolvimento de Aplicações Híbridas**. Orientador: Aline Vieira de Mello. 2019. Trabalho de conclusão de curso (Engenharia de Software) - Universidade Federal do Pampa, [S. l.], 2019. Disponível em: <http://dspace.unipampa.edu.br/bitstream/riu/4224/1/Fernando%20Fortunato%20de%20Lima%20-%20202019.pdf>. Acesso em: 12/03/2021.

MACHUNO, Luís Gustavo Baptista; RANGEL, Elidiane Cipriano; LIMA, Anderson Barbosa; GELAMO, Rogério Valetim. Desenvolvimento e avaliação de uma fonte DC de alta tensão para utilização em sistema de deposição de filmes finos por pulverização catódica. **Revista Matéria**, [S. l.], p. 492-500, 13/06/2016. DOI 10.1590/S1517-707620160002.0046. Disponível em: [https://www.researchgate.net/publication/303807867\\_Desenvolvimento\\_e\\_avaliacao\\_de\\_uma\\_fonte\\_DC\\_de\\_alta\\_tensao\\_para\\_utilizacao\\_em\\_sistema\\_de\\_depositacao\\_de\\_filmes\\_finos\\_por\\_pulverizacao\\_catodica](https://www.researchgate.net/publication/303807867_Desenvolvimento_e_avaliacao_de_uma_fonte_DC_de_alta_tensao_para_utilizacao_em_sistema_de_depositacao_de_filmes_finos_por_pulverizacao_catodica). Acesso em: 20 dez. 2020.

MCROBERTS, Michael. **Arduino Básico**. São Paulo - SP: Novatec, 2011.

MURTA, Gustavo. **Guia completo do Motor de Passo 28BYJ-48 + Driver ULN2003**. [S. l.], 23/07/2018. Disponível em: <https://blog.eletrogate.com/guia-completo-do-motor-de-passo-28byj-48-driver-uln2003/>. Acesso em: 09/02/2021.

NAHIME, Bacus de Oliveira. **Síntese e caracterização estrutural de filmes finos Bi<sub>4</sub>Ti<sub>3</sub>O<sub>12</sub>**. Orientador: Eudes Borges de Araújo. 2007. Dissertação (Programa de Pós-Graduação em

Ciência dos Materiais) - Universidade Estadual Paulista, [S. l.], 2007. Disponível em: <http://livros01.livrosgratis.com.br/cp037321.pdf>. Acesso em: 20/12/2020.

NODEJS®. Node JS® *About*. 2020. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 06/04/2020.

NPM. Npm *About*. 2020. Disponível em: <https://docs.npmjs.com/about-npm/>. Acesso em: 06/04/2020.

OHRING, Milton. *The Materials science of thin films*. [S. l.: s. n.], 1991. Disponível em: [https://arshadnotes.files.wordpress.com/2018/02/the\\_materials\\_science\\_of\\_thin\\_films.pdf](https://arshadnotes.files.wordpress.com/2018/02/the_materials_science_of_thin_films.pdf). Acesso em: 17/11/2020.

OLIVEIRA, Euler. **Como usar com Arduino – Módulo Bluetooth HC-05 / HC-06**. [S. l.], 2019. Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-bluetooth-hc-05-hc-06/>. Acesso em: 09/10/2020.

PENNA, Bruno Grieco. **PRODUÇÃO DE FILMES FINOS DE GRAFITE E ÓXIDO DE ZINCO POR MAGNETRON SPUTTERING**. Orientador: Renata Antoun Simão. 2018. Trabalho de conclusão de curso (Bacharelado em Engenharia de Materiais) - Universidade Federal do Rio de Janeiro, [S. l.], 2018. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10024301.pdf>. Acesso em: 27/02/2021.

REIS, Diego Glauco Azarias dos. **DISPOSITIVO DE PLASMA ATMOSFÉRICO COM PRECURSOR E SUA APLICAÇÃO EM DEPOSIÇÃO POLIMÉRICA**. Orientador: Milton Eiji Kayama. 2017. Dissertação (Mestrado em Engenharia Mecânica na Área de Materiais) - Universidade Estadual Paulista, Guaratinguetá/SP, 2017. Disponível em: [https://repositorio.unesp.br/bitstream/handle/11449/150392/azarias\\_dga\\_me\\_guara.pdf?sequence=5&isAllowed=y](https://repositorio.unesp.br/bitstream/handle/11449/150392/azarias_dga_me_guara.pdf?sequence=5&isAllowed=y). Acesso em: 22/11/2020.

ROBOCORE TECNOLOGIA LTDA. **Controlando motores: Motor de passo**. [S. l.], 21 mar. 2018. Disponível em: <https://www.robocore.net/tutorials/97>. Acesso em: 26/08/2019.

RODRIGUES, Robson Borges. **Desenvolvimento e aplicação de um equipamento de espectroscopia de impedância para medição de capacitância**. Orientador: Rogério Valentim Gelamo. 2016. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2016.

ROGÉRIO, Lúcio Junior. **Projeto e construção de uma fonte de alta tensão DC para aplicação em pulverização catódica**. Orientador: Rogério Valentim Gelamo. 2016. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2019.

SESHAN, Krishna. *HANDBOOK OF THIN-FILM DEPOSITION PROCESSES AND TECHNIQUES: Principles, Methods, Equipment and Applications. Second Edition*. ed. [S. l.]: NOYES PUBLICATIONS, 2002. Disponível em: [https://is.muni.cz/el/1431/jaro2017/F4280/um/Handbook\\_of\\_Thin-Film\\_Deposition\\_ed-Seshan-Krishna.pdf](https://is.muni.cz/el/1431/jaro2017/F4280/um/Handbook_of_Thin-Film_Deposition_ed-Seshan-Krishna.pdf). Acesso em: 17 nov. 2020.

SILVA, Camila Cristina da. **ESTUDO DO EFEITO DA MODIFICAÇÃO DA FIBRA DE SISAL USANDO PLASMA FRIO REATIVO E SEU IMPACTO NA REMOÇÃO DO AZUL DE METILENO**. Orientador: Rogério Valentim Gelamo. 2019. Dissertação (Programa de Mestrado Profissional em Inovação Tecnológica) - Universidade Federal do Triângulo Mineiro, Uberaba/MG, 2019.

SOUZA, Anderson de R.; PAIXÃO, Alexsander C.; ZUÊDA, Diego D.; DIAS, Marco A.; DUARTE, Sergio; DE AMORIM, Helio S. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**, São Paulo - SP, Março - 2011. DOI <http://dx.doi.org/10.1590/S1806-11172011000100026>. Disponível em: [http://www.scielo.br/scielo.php?pid=S1806-11172011000100026&script=sci\\_arttext](http://www.scielo.br/scielo.php?pid=S1806-11172011000100026&script=sci_arttext). Acesso em: 25/08/2019.

THOMSEN, Adilson. **Controlando um Motor de Passo 5v com Arduino®**. [S. l.], 18/03/2013. Disponível em: <https://www.filipeflop.com/blog/controlando-um-motor-de-passo-5v-com-arduino/>. Acesso em: 28/04/2020.

THOMSEN, Adilson. **Tutorial Módulo Bluetooth com Arduino®**. [S. l.], 06/07/2015. Disponível em: <https://www.filipeflop.com/blog/tutorial-modulo-bluetooth-com-arduino/>. Acesso em: 09/10/2020.

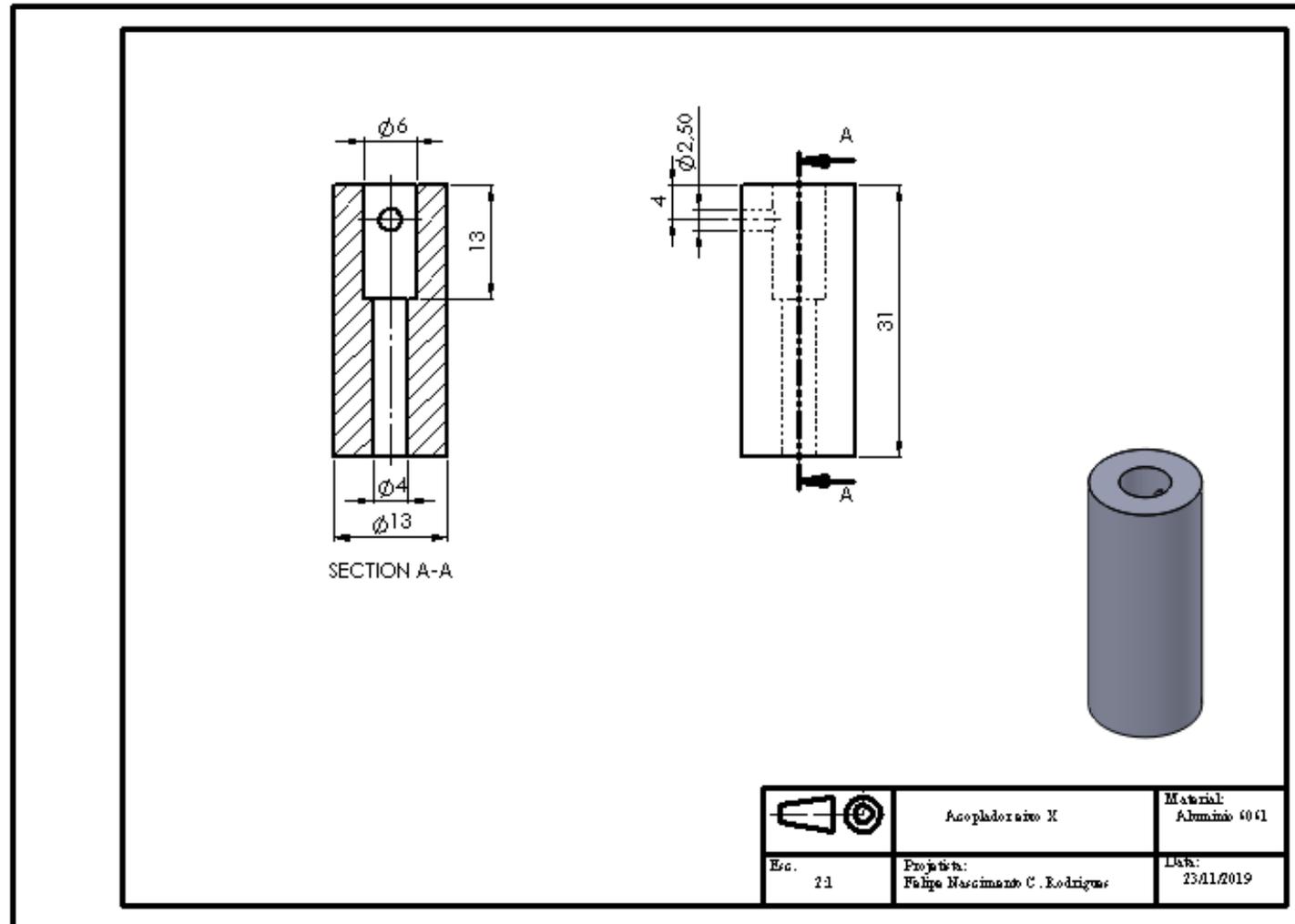
VIDAL, Vitor. **Módulos Bluetooth HC05 e HC06 para comunicação com dispositivos móveis com Arduino®**. [S. l.], 22/06/2017. Disponível em: <https://blog.eletrogate.com/modulos-bluetooth-hc05-e-hc06-para-comunicacao-com-dispositivos-moveis-com->

[arduino/?gclid=Cj0KCQjw8fr7BRDSARIsAK0Qqr56pf\\_JF4Urr2ySDFTQrZaz5k8pZ167S-AjWO-N3L5QY-yEY0g9BTIaAiVWEALw\\_wcB](https://www.arduino.cc/en/Reference/ArduinoProMini). Acesso em: 09/10/2020.

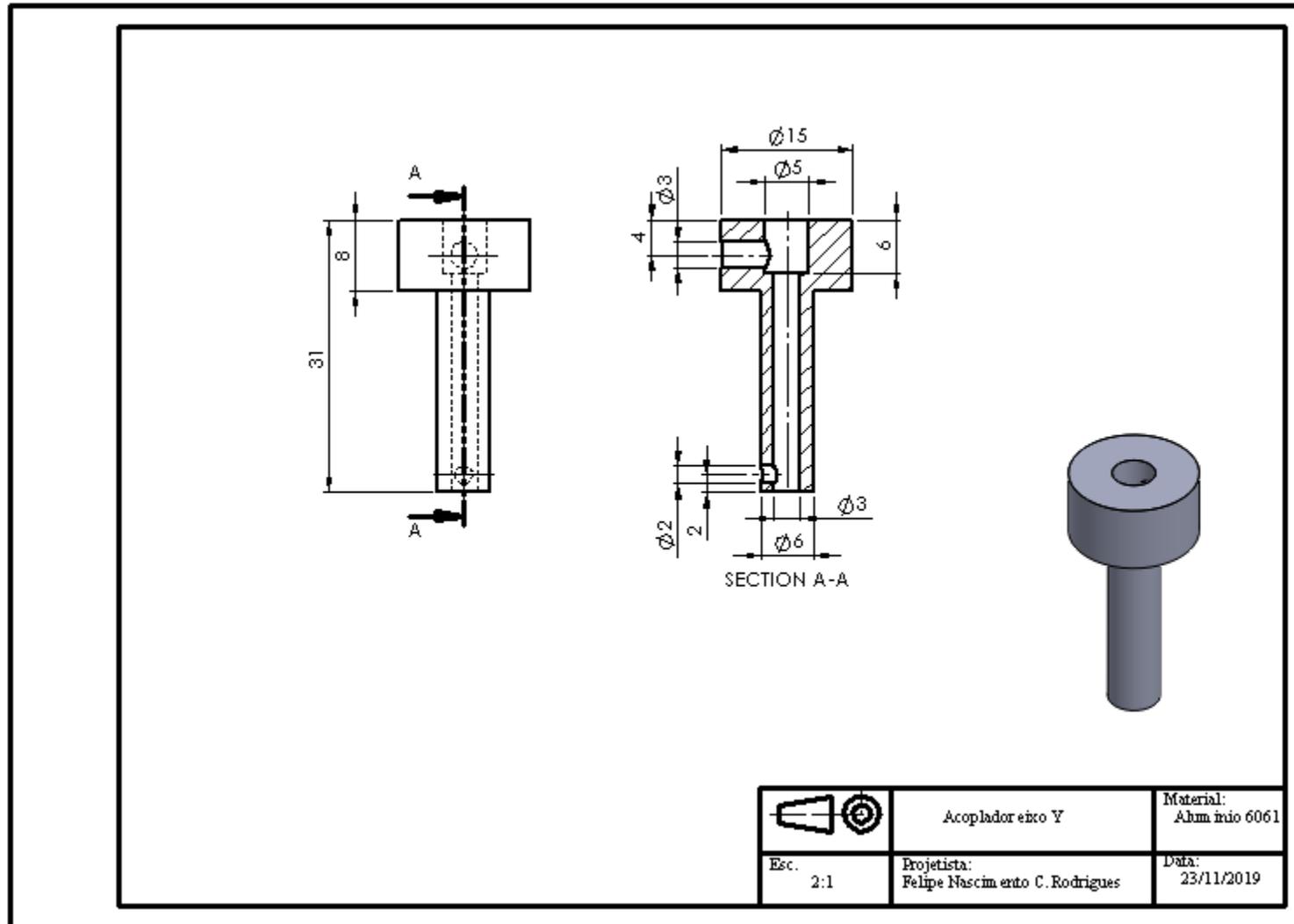
VISUAL STUDIO CODE. *Getting Started*. [S. l.], 12/05/2020. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 12/05/2020.

ZARPELON, Fabiana. **Preparação, Caracterização e Aplicação de Filmes Finos de PAH/PAA com Nanopartículas de Prata no Tratamento Microbiológico de Efluentes Industriais para Reuso**. Orientador: Marcelo Giovanela. 2013. Dissertação de mestrado (Programa de Pós-Graduação em Engenharia e Ciência dos Materiais) - Universidade Caxias do Sul, [S. l.], 2013. Disponível em: <https://repositorio.uces.br/xmlui/bitstream/handle/11338/722/Dissertacao%20Fabiana%20Zarpeleon.pdf?sequence=1&isAllowed=y>. Acesso em: 27/02/2021.

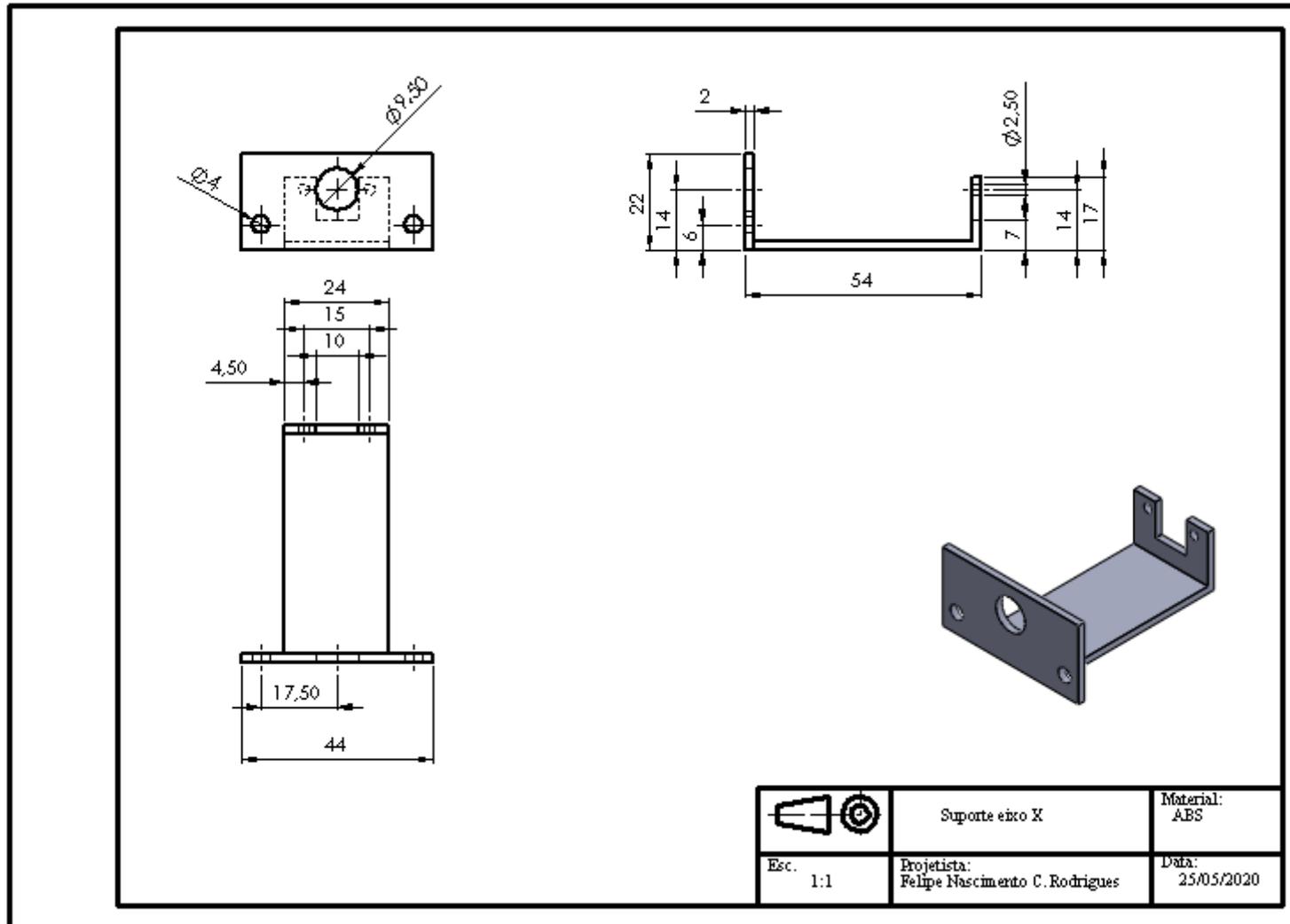
## ANEXO A – PROJETO DO ACOPLADOR DO EIXO X



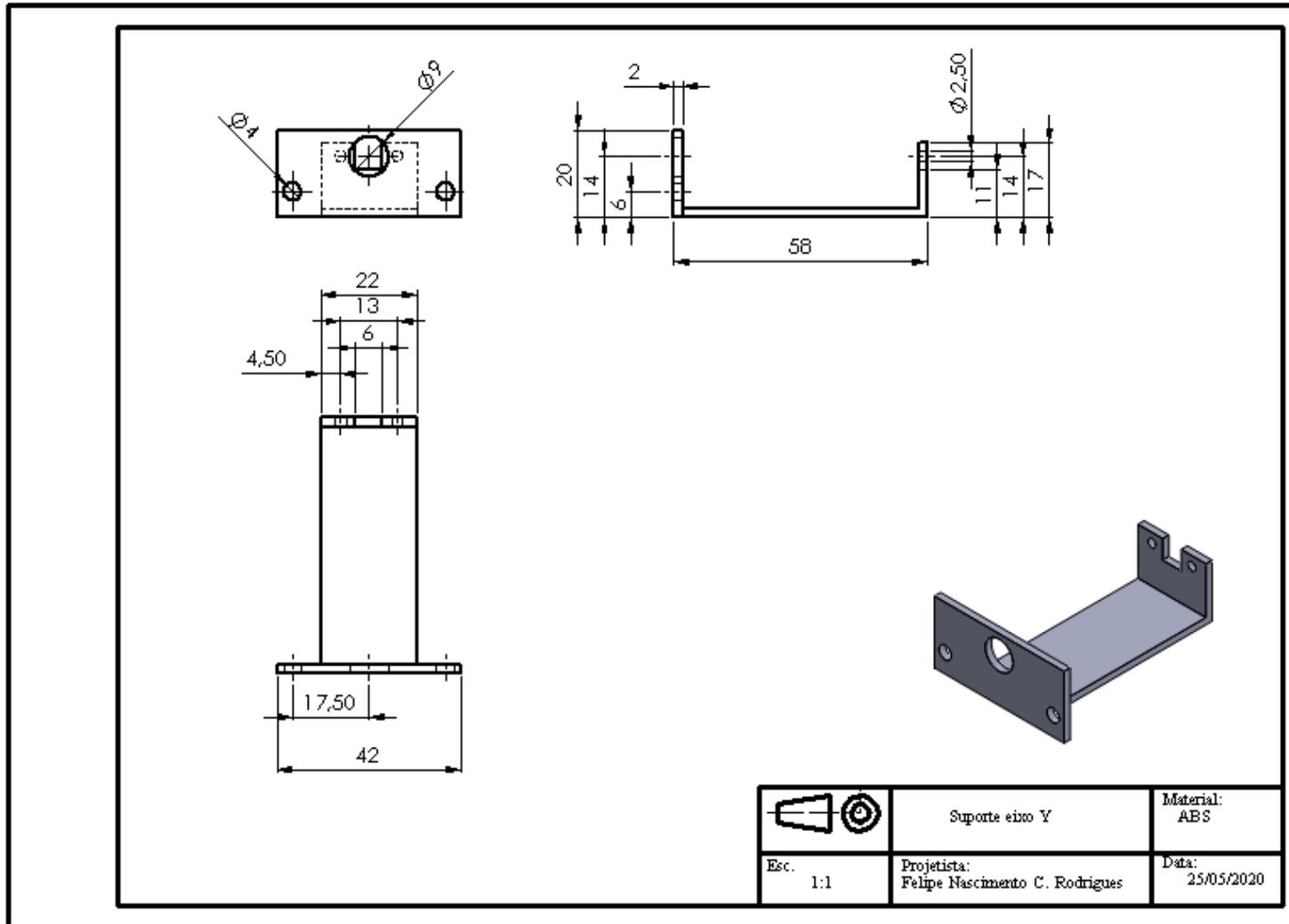
## ANEXO B – PROJETO DO ACOPLADOR DO EIXO Y



## ANEXO C – PROJETO DO SUPORTE DO EIXO X



## ANEXO D – PROJETO DO SUPORTE DO EIXO Y



**ANEXO E – ALGORITMO DO ARDUINO© *BLUETOOTH*.**

```
#include <Stepper.h>

int stepsPerRevolution = 64;

Stepper motorX(stepsPerRevolution, 22, 24, 23, 25);

Stepper motorY(stepsPerRevolution, 28, 30, 29, 31);

int buzzer = 50;

String readString = "";

float stepX = 0;

float speedX = 0;

float stepY = 0;

float speedY = 0;

String DeltaSelectString = "";

float delta = 0;

void setup() {

  Serial.begin(9600);

  pinMode(buzzer, OUTPUT);

  digitalWrite(buzzer, LOW);

}

void loop() {

  if (Serial.available() > 0) {
```

```
char data = Serial.read();

readString += data;

if (data == '/') {
    getData();
}
}
}

void getData() {

    int stepXStarPosition = readString.indexOf("stepX=") + sizeof("stepX=") - 1;
    int stepXEndPosition = readString.indexOf("&speedX=");
    String stepXString = readString.substring(stepXStarPosition, stepXEndPosition);
    stepX = stepXString.toFloat();

    int speedXStarPosition = readString.indexOf("speedX=") + sizeof("speedX=") - 1;
    int speedXEndPosition = readString.indexOf("&stepY=");
    String speedXString = readString.substring(speedXStarPosition, speedXEndPosition);
    speedX = speedXString.toFloat();

    int stepYStarPosition = readString.indexOf("stepY=") + sizeof("stepY=") - 1;
    int stepYEndPosition = readString.indexOf("&speedY=");
    String stepYString = readString.substring(stepYStarPosition, stepYEndPosition);
    stepY = stepYString.toFloat();

    int speedYStarPosition = readString.indexOf("speedY=") + sizeof("speedY=") - 1;
    int speedYEndPosition = readString.indexOf("&deltaSelect=");
```

```
String speedYString = readString.substring(speedYStarPosition, speedYEndPosition);
speedY = speedYString.toFloat();
```

```
int DeltaSelectStarPosition = readString.indexOf("deltaSelect=") + sizeof("deltaSelect=") - 1;
```

```
int DeltaSelectEndPosition = readString.indexOf("&delta=");
```

```
DeltaSelectString = readString.substring(DeltaSelectStarPosition, DeltaSelectEndPosition);
```

```
int deltaStarPosition = readString.indexOf("delta=") + sizeof("delta=") - 1;
```

```
int deltaEndPosition = readString.indexOf("/");
```

```
String deltaString = readString.substring(deltaStarPosition, deltaEndPosition);
```

```
delta = deltaString.toFloat();
```

```
turnOn();
```

```
}
```

```
void turnOn() {
```

```
digitalWrite(buzzer, HIGH);
```

```
delay(2000);
```

```
digitalWrite(buzzer, LOW);
```

```
if (DeltaSelectString == "deltaX") {
```

```
int j = abs(stepX / delta);
```

```
int signal = 1;
```

```
if (stepX < 0) {
```

```
delta = -delta;
```

```
}

stepX = round(stepX * 2048 / 14.95);
speedX = round(speedX * 100 / 14.95 * 19.22);
stepY = round(stepY * 2048 / 3.55);
speedY = round(speedY * 100 / 3.55 * 19.22);
delta = round(delta * 2048 / 14.95);

motorX.setSpeed(speedX);
motorY.setSpeed(speedY);

for (int i = 0; i < j; i++) {
    motorY.step(signal * stepY);
    motorX.step(delta);
    signal = signal * -1;
}
}

else {

    int j = abs(stepY / delta);
    int signal = 1;

    if (stepY < 0) {
        delta = -delta;
    }

    stepX = round(stepX * 2048 / 14.95);
```

```
speedX = round(speedX * 100 / 14.95 * 19.22);
stepY = round(stepY * 2048 / 3.55);
speedY = round(speedY * 100 / 3.55 * 19.22);
delta = round(delta * 2048 / 3.55);

motorX.setSpeed(speedX);
motorY.setSpeed(speedY);

for (int i = 0; i < j; i++) {
    motorX.step(signal * stepX);
    motorY.step(delta);
    signal = signal * -1;
}

readString = "";
stepX = 0;
speedX = 0;
stepY = 0;
speedY = 0;
DeltaSelectString = "";
delta = 0;
digitalWrite(buzzer, HIGH);
delay(2000);
digitalWrite(buzzer, LOW);
}
```