



**CENTRO DE EDUCAÇÃO PROFISSIONAL HÉLIO AUGUSTO DE SOUZA**

**SALA AUTÔNOMA  
AUTOMAÇÃO RESIDENCIAL**

**Curso:** Técnico em Eletrônica

**Turno:** Noite

**Mód.:** III

**Componente Curricular:** Projeto

**Alunos:** Camilly Gurgel dos Santos

Nº02

Débora Cavalcante da Silva

Nº03

Julio Oscar Retamales

Nº08

Lucineide Maria de Brito

Nº 09

Wanderson Augusto da Cunhas Santos

Nº14

**Professor Orientador:** Antônio Carlos Cintra Pereira

**São José dos Campos - SP**

**2023**



Camilly Gurgel dos Santos  
Débora Cavalcante da Silva  
Julio Oscar Retamales  
Lucineide Maria de Brito  
Wanderson Augusto da Cunha Santos

Sala Autônoma

Projeto ou Monografia apresentado(a)  
como requisito parcial para a obtenção do  
título de técnico em Eletrônica pelo Centro  
de Formação Profissional Hélio Augusto de  
Souza em São José dos Campos

Professor Orientador: Antônio Carlos  
Cintra Pereira

**São José dos Campos - SP**

**2023**



## **AGRADECIMENTOS**

Dedico este trabalho a todos que nos ajudaram em nossa difícil caminhada, equilibrando estudo e trabalho, enfrentando noites mal dormidas e desafios constantes. Cada obstáculo superado foi parte essencial dessa trajetória.

Em especial dedicamos também ao nosso Orientador Antonio Carlos Cintra, que nos ajudou no desenvolvimento de nosso projeto.

Eu, Júlio, dedico especialmente à minha mãe, uma mulher batalhadora que me ensinou que momentos difíceis sempre existirão, mas é justamente isso que nos torna mais fortes para seguir em frente. Pois sem pedras no caminho, não há glória na chegada.

"Basta ser sincero e desejar profundo, você será capaz  
de sacudir o mundo".  
- Raul Seixas



## **Resumo**

Este projeto consiste no desenvolvimento de um sistema de automação para uma sala, utilizando sensores, motores e um módulo Bluetooth para controle remoto de portas, janelas e dispositivos eletrônicos. A automação permite que a porta se abra automaticamente ao detectar a presença de uma pessoa e que a janela seja controlada remotamente. Além disso, LEDs foram utilizados para representar eletrodomésticos, simulando um ambiente conectado.

A comunicação entre o usuário e o sistema ocorre via celular ou computador, garantindo praticidade e acessibilidade. Diferente de sistemas baseados em internet, o uso do Bluetooth possibilita a aplicação em locais onde a conexão de rede é limitada.

O projeto abrange desde a escolha dos componentes até a programação dos dispositivos, passando por testes práticos para validar seu funcionamento. Os resultados mostram que a automação pode trazer mais comodidade ao dia a dia e servir como base para futuras inovações na área.

Palavras-chave: Automação, Bluetooth, Sensores, Controle Remoto, Acessibilidade.



## Abstract

This project involves the development of an automation system for a room, using **sensors, motors, and a Bluetooth module** to remotely control doors, windows, and electronic devices. The automation allows the door to open automatically when detecting a person's presence, while the window can be controlled remotely. Additionally, LEDs were used to represent household appliances, simulating a connected environment.

Communication between the user and the system occurs via **mobile phone or computer**, ensuring practicality and accessibility. Unlike internet-based systems, the use of **Bluetooth** enables implementation in areas with limited network connectivity.

The project covers everything from component selection to device programming, including practical tests to validate its functionality. The results show that automation can bring more convenience to daily life and serve as a foundation for future innovations in the field.

**Keywords:** Automation, Bluetooth, Sensors, Remote Control, Accessibility



## Sumário

- 1. **Introdução** – 10
  - 1.1 Objetivo – 10
    - 1.1.1 Objetivos Específicos – 10
  - 1.2 Justificativa – 10
- 2. **Materiais** – 10
  - 2.1 Sensores – 11
    - 2.1.1 Sensor Ultrassônico – 11
    - 2.1.2 Processamento de Dados – 11
  - 2.2 Arduino – 12
  - 2.3 Módulo Bluetooth – 12
    - 2.3.1 Linguagem C++ – 13
  - 2.4 Controles – 13
    - 2.4.1 Motor Linear – 13
    - 2.4.2 CNC Shield – 13
- 3. **Métodos e Procedimentos** – 14
  - 3.1 Cronograma – 14
  - 3.2 Diagrama de Bloco – 14
  - 3.3 Divisão de Funções – 15
- 4. **Desenvolvimento** – 16
  - 4.1 Descrição de Montagem – 16
  - 4.2 Código – 17
- 5. **Resultados e Discussão** – 24
  - 5.1 Resultado Geral – 24
- 6. **Conclusão** – 24



## 1. Introdução

A automação residencial tem se tornado uma realidade cada vez mais presente no cotidiano das pessoas. Com o avanço da tecnologia e a popularização de sistemas inteligentes, torna-se possível controlar diversos dispositivos de uma casa de maneira remota, eficiente e personalizada. O conceito de casa inteligente envolve a utilização de sensores, atuadores e sistemas de comunicação que permitem a automação de tarefas diárias, trazendo comodidade, segurança e economia de energia.

Nos últimos anos, a automação deixou de ser um luxo restrito a grandes empresas e residências sofisticadas para se tornar acessível a um público mais amplo. Soluções como controle de iluminação, abertura e fechamento de portas e janelas, monitoramento de ambientes e acionamento remoto de eletrodomésticos estão cada vez mais presentes no mercado. Esse crescimento é impulsionado pela demanda por mais praticidade e pelo avanço de tecnologias como microcontroladores, sensores e comunicação sem fio.

O presente projeto propõe o desenvolvimento de uma sala automatizada, utilizando um Arduino Mega, motores de passo, sensores e um módulo Bluetooth. O sistema permite o controle remoto de portas, janelas e dispositivos simulados através de LEDs, proporcionando um ambiente interativo e funcional. Diferente de soluções que dependem de conexão com a internet, o uso do Bluetooth permite a implementação da automação em locais com infraestrutura limitada, tornando-a uma alternativa viável para diferentes cenários.

### 1.1. Objetivo

O principal objetivo deste projeto é demonstrar, de forma didática e acessível, o funcionamento de um sistema automatizado e suas possíveis aplicações no dia a dia. Buscamos apresentar a automação residencial como uma **solução prática e funcional**, destacando sua viabilidade em diferentes contextos e incentivando sua adoção em novos projetos.

#### 1.1.1. Objetivo específico

- Desenvolver um sistema de automação utilizando Arduino Mega, sensores e motores, explorando conceitos fundamentais de eletrônica e programação.
- Criar um sistema funcional e interativo, permitindo o controle remoto de portas, janelas e dispositivos através de um módulo Bluetooth.
- Fornecer um material introdutório para estudantes e entusiastas da área, abordando a implementação de automação com Arduino e C++.



## **1.2. Justificativa**

A crescente busca por soluções tecnológicas que simplifiquem tarefas diárias tem impulsionado o desenvolvimento da automação residencial. No entanto, muitas dessas tecnologias ainda são pouco exploradas por estudantes e pequenos desenvolvedores, seja por falta de conhecimento ou de acesso a materiais didáticos acessíveis.

Este projeto tem como propósito expandir a compreensão sobre automação, mostrando como sistemas simples podem ser implementados de forma prática e eficiente. Ao utilizar componentes acessíveis, como Arduino e módulos Bluetooth, buscamos tornar esse conhecimento mais democrático, incentivando novas aplicações e soluções voltadas para melhorar a qualidade de vida e a acessibilidade tecnológica.

tornando possível mais pessoas trabalharem nessa área, onde possam produzir equipamentos autônomos que facilitem a qualidade de vida de todos.





## 2. Materiais

Uma maquete demonstrativa em escala reduzida de uma sala onde se apresenta janela e porta automatizadas, que se movem de forma independente sem a necessidade de intervenção humana. Esses dispositivos são equipados com sistemas de percepção, tomada de decisão e controle, permitindo a interação com o indivíduo de forma autônoma. A fundamentação teórica da sala automatizada envolve diversos conceitos e tecnologias fundamentais , que são discutidas nos textos a seguir

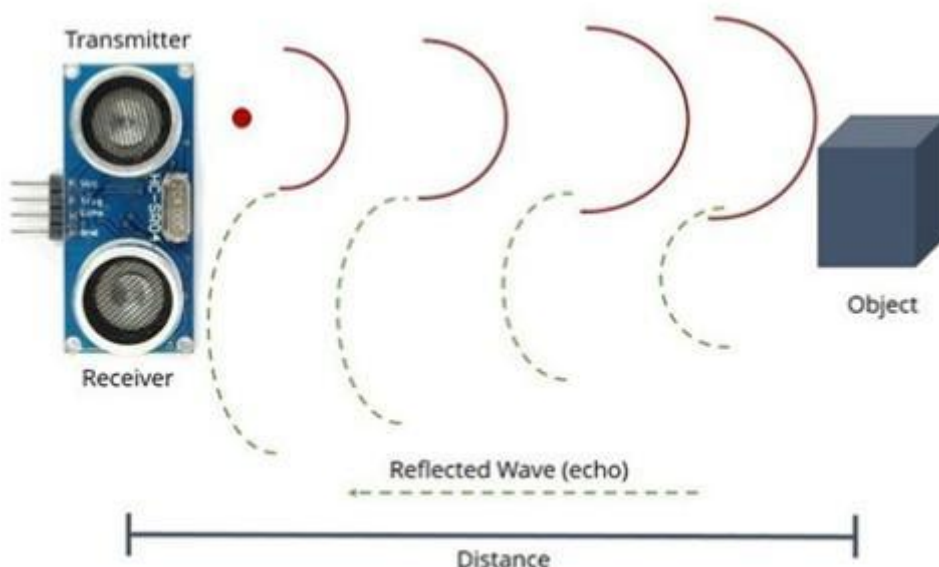
Escolhemos demonstrar apenas uma sala ao invés de um ambiente completo devido a facilidade demonstrativa do nosso protótipo.

### 2.1. Sensores

Os sensores são componentes essenciais para maquete autônoma, pois fornecem informações sobre o ambiente externo e a presença de um indivíduo.

#### 2.1.1. Sensor Ultrassônico

Utilizamos sensores ultrassônicos que funcionam medindo o tempo de propagação do eco, isto é, o intervalo de tempo entre o impulso sonoro emitido e o eco recebido de volta. Os sensores emitem pulsos de ultrassom ciclicamente. Quando um objeto reflete estes pulsos, o eco resultante é recebido e convertido em um sinal elétrico. Sendo o sensor principal para análise da área próximo a porta, compreendendo quando deve-se abrir a porta, mantê-la aberta ou fechá-la

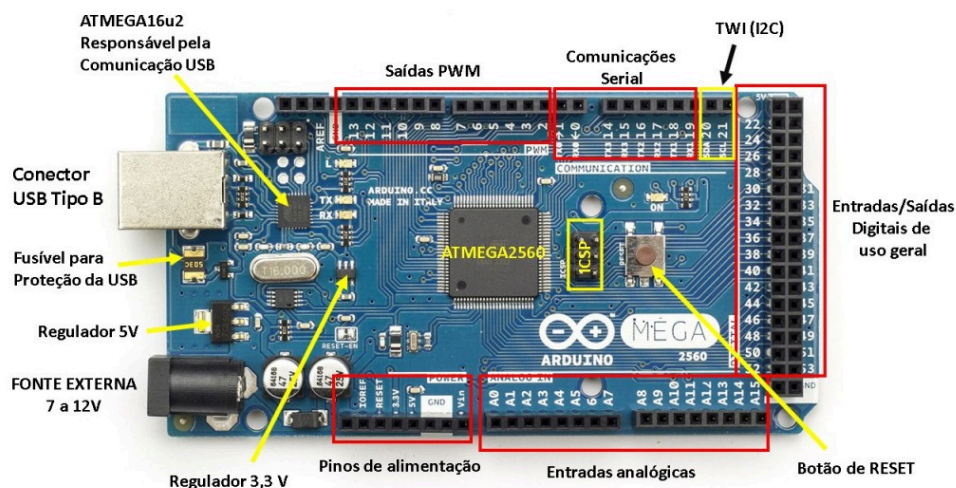


### 2.1.2. Processamento de dados

Após a coleta de dados pelos sensores, é necessário processar essas informações para que a porta automática possa tomar decisões. O processamento de dados envolve etapas como filtragem, fusão de sensores e extração de características relevantes. Os algoritmos de aprendizado de máquina e visão computacional são amplamente utilizados para processar os dados dos sensores e extrair informações úteis.

## 2.2. Arduino

Utilizamos o arduino para processar os dados pois é uma placa de prototipagem eletrônica de código aberto (open-source) e hardware livre. Sendo bem fácil de mexer e programar, podendo usá-la para armazenar todos os dados necessários para o procedimento de análise da porta e janela automatizadas.



### **2.3. Modulo Bluetooth hc-05**

O módulo Bluetooth HC-05 é um dos principais componentes do sistema, responsável por permitir a comunicação sem fio entre o Arduino Mega e um dispositivo externo, como um smartphone ou computador. Ele opera como uma ponte de comunicação serial, recebendo comandos enviados pelo usuário e transmitindo-os ao microcontrolador para a execução das ações correspondentes.

No projeto da sala automatizada, o módulo Bluetooth possibilita o controle remoto da porta, janela e LEDs através de um aplicativo ou terminal serial. Dessa forma, o usuário pode enviar comandos específicos para abrir ou fechar a janela, acionar a porta ou simular o funcionamento de dispositivos eletrônicos representados por LEDs.

A escolha do Bluetooth, em vez de uma conexão baseada na internet, foi feita para garantir que o sistema funcione mesmo em locais sem acesso à rede, tornando-o uma solução acessível e funcional para diferentes ambientes.



### **2.4. Tomada de Decisões**

Com base nas informações coletadas e processadas, a porta deve tomar decisões sobre seu trajeto. Algoritmos de planejamento de trajetória são empregados para determinar a rota mais adequada, considerando limitações de velocidade, regras de direção e outros fatores relevantes. Além disso, os sistemas de tomada de decisão também devem considerar aspectos de segurança e priorizar ações que minimizem riscos.



### **2.4.1. Linguagem C++**

Utilizamos C++ onde se torna bem mais útil para produção de um algoritmo pela flexibilidade e também por trabalhar em conjunto direto com o IDE do arduino, fazendo com que o robô faça a escolha de acordo com algoritmo programado em seu sistema.

## **2.5. Controles**

### **2.5.1. Motor Linear**

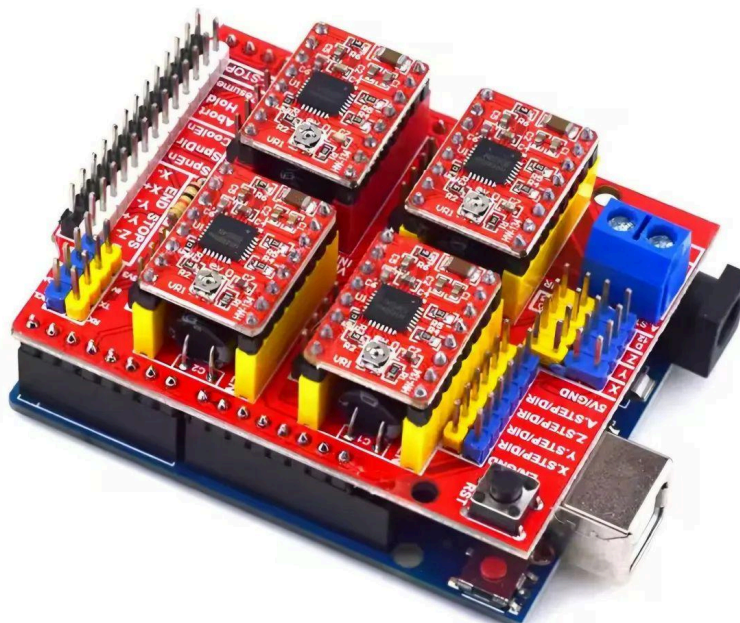
Foi feita a aplicação de um motor linear no projeto de automação de portas e janelas. O motor linear converte eficientemente energia elétrica de corrente contínua em movimento mecânico, sendo escolhido pela capacidade de controle preciso e reversibilidade. Amplamente utilizado em automação residencial, o motor linear oferece eficiência energética e resposta rápida, tornando-o ideal para sistemas que exigem movimentos suaves e bidirecionais.



Fonte: <https://www.amazon.com.br/lineares-acabamento-deslizante-equipamentos-deslizantes/dp/B097X521CQ>

### **2.5.2. CNC Shield**

Uma extensão do Arduino para o controle dos motores, ele controla melhor o circuito recebendo informações do arduino e passando com mais precisão para cada motor, sendo eficiente também por conseguir colocar outros equipamentos nele para trabalhar em conjunto de acordo com que foi programado e passado para ele do Arduino, exemplo como o uso do sensor ultrassônico e do servo motores.



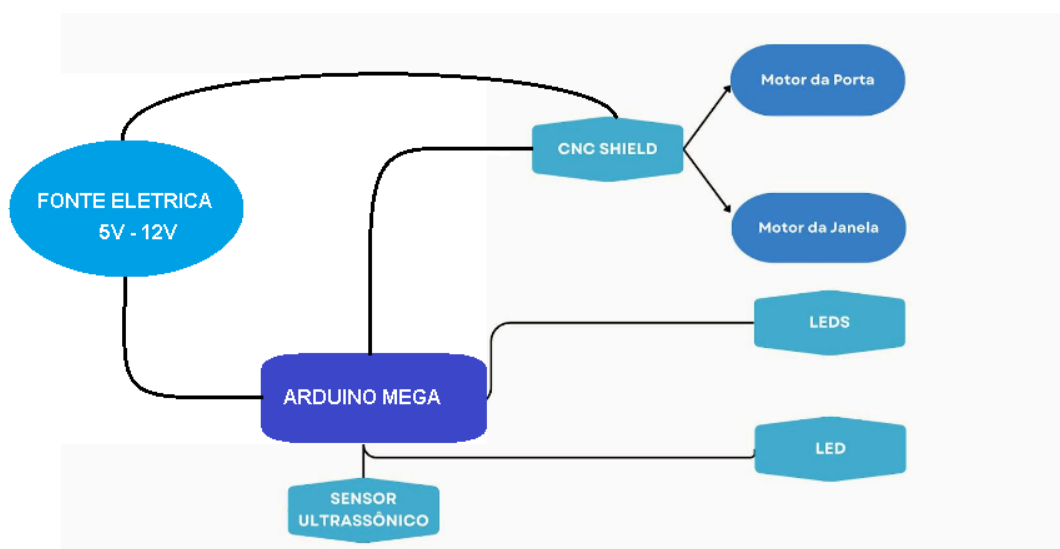
Fonte:arduinoecia.com.br

### 3. Métodos e Procedimentos

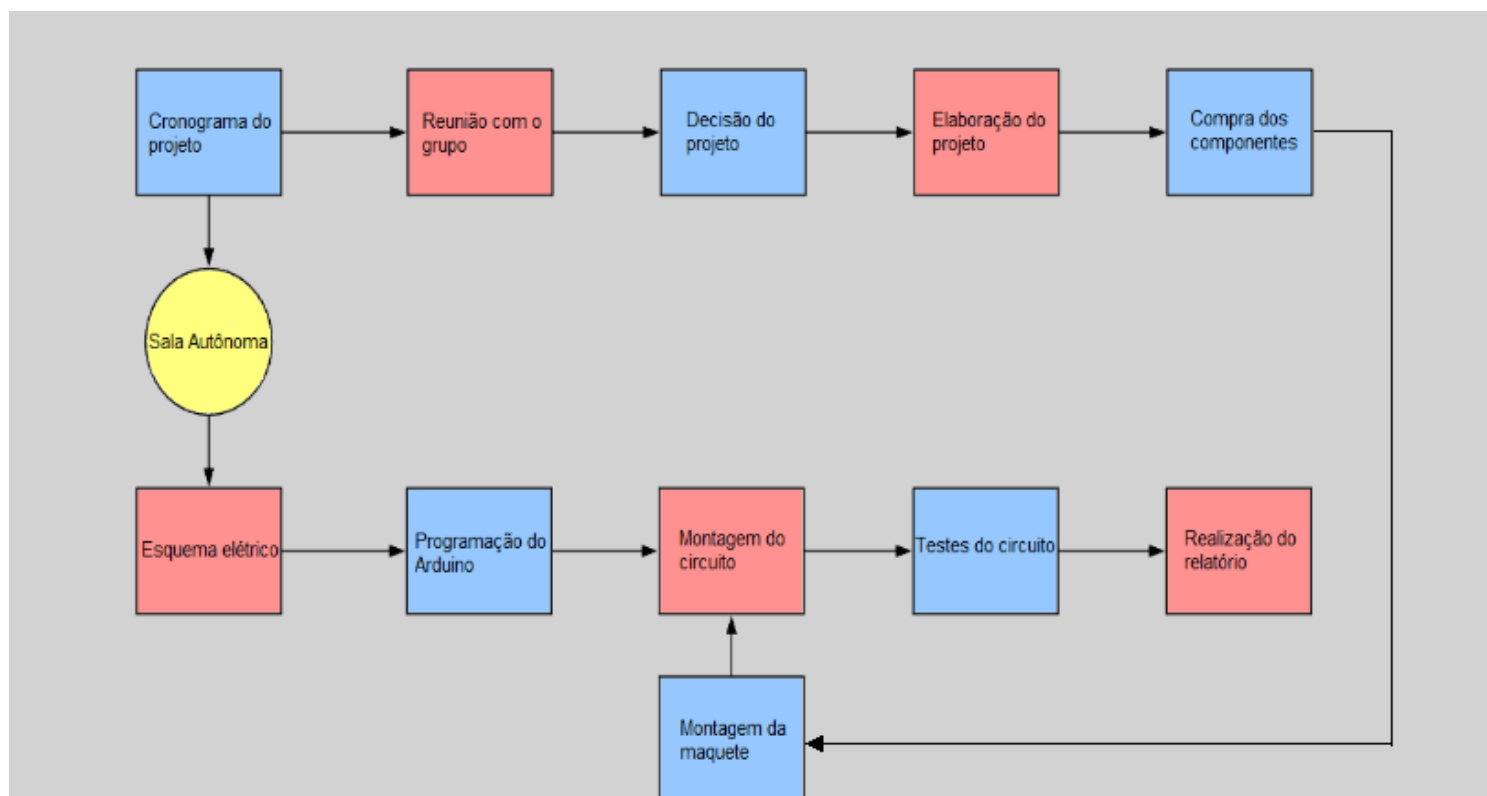
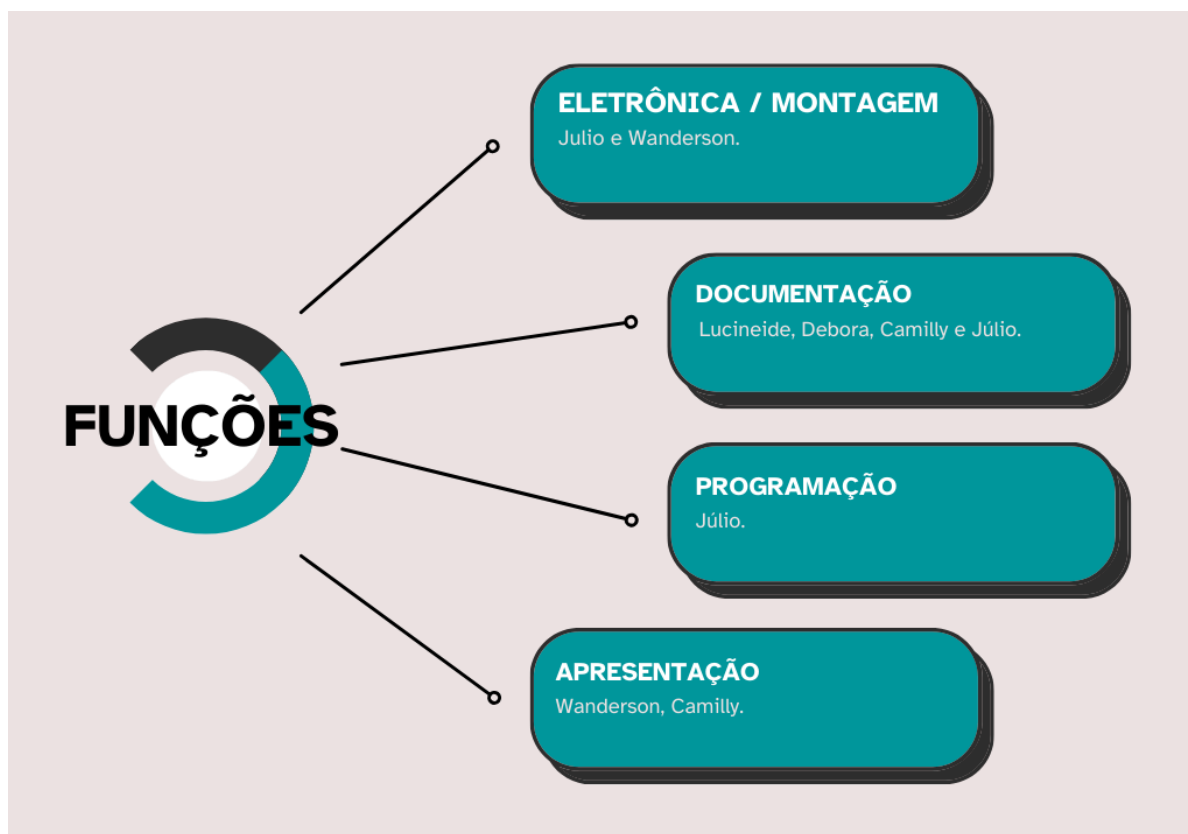
#### 3.1. Cronograma

Materias / Custos	Produção da programação	Conceito de Montagem	Processo de montagem	Testes/ Finalizações
4 Semanas	4 Semanas	2 Semanas	2 Semana	2 Semanas

#### 3.2. Diagramas de Bloco



### 3.3. Divisão de Funções





## **4. Desenvolvimento**

### **4.1. Descrição de Montagem**

Dando início ao desenvolvimento, iniciamos primeiramente com o desenvolvimento do Código de programação que viria ser utilizado, onde inserimos os comandos do monitor serial para cada componente presente na maquete, dando ênfase nos comandos da CNC Shield, e os executando em simuladores como Tinkercad.

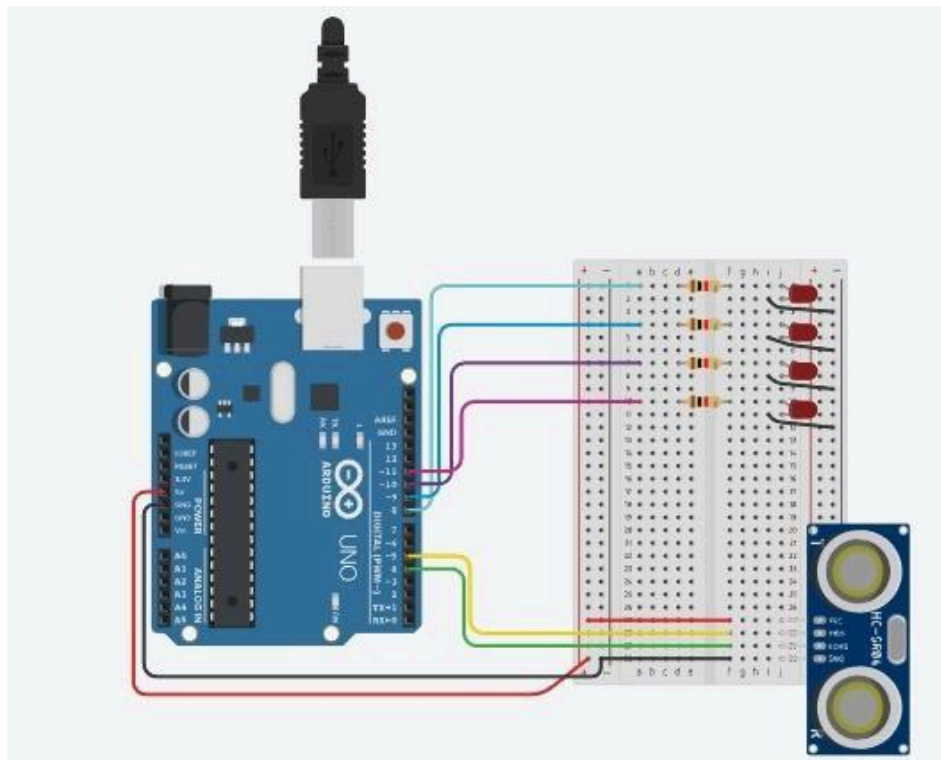
Para a ligação dos Motores Lineares a Shield foi necessário soldagem para os fios ( o processo de inserção dos motores a maquete e soldagem dos fios foram repetidas para a porta e para a janela).

Dando continuidade, inserimos cada motor linear em seus respectivos locais, onde em conjunto foram inseridas as peças que seriam utilizadas para simbolizar a porta e janela. A porta terá como adicional de comando o sensor ultrassônico que ao detectar a presença de um indivíduo dentro da delimitação determinada irá acionar automaticamente a abertura da porta.

Após esses procedimentos, nos atentamos à montagem e colocação de cada componente em cada local determinado, onde utilizamos o uso de parafusos para fixar os motores lineares responsáveis pelo deslocamento da porta e janela. Para a Maquete, realizamos a perfuração para a entrada do Sensor Ultrassônico ao motor linear correspondente a porta e as ligações necessárias para a Shield. Concluída essa fase, inserimos o arduino mega ao centro da maquete, e logo interligamos e posicionamos a Shield acima do arduino.

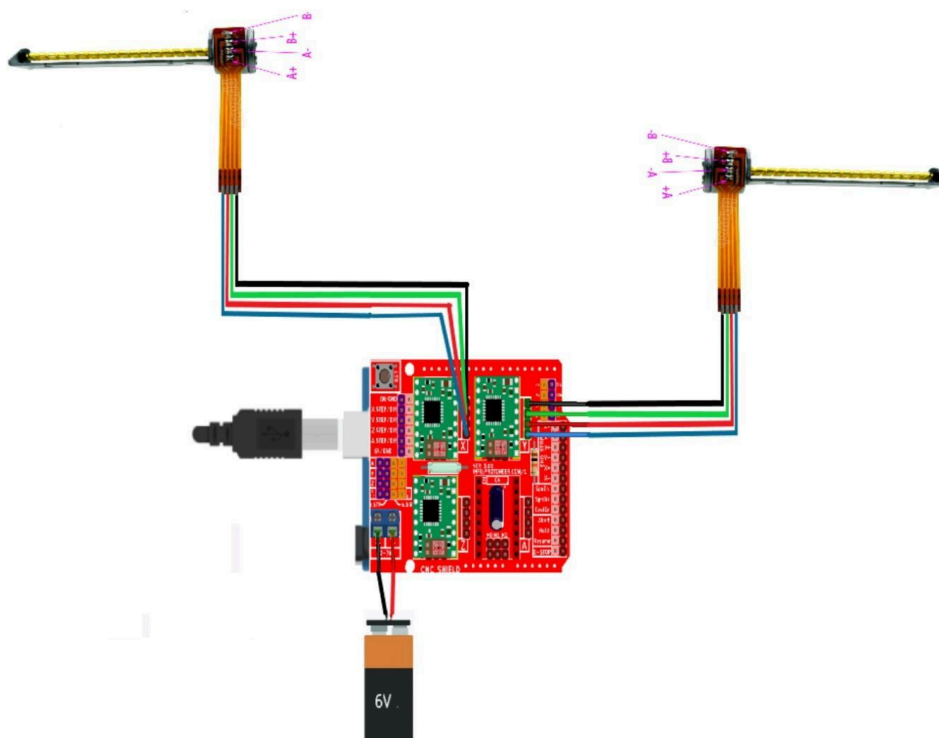
Como complemento, adicionamos a maquete e o código de programação comando para alguns LED's que vieram a ser adicionados na maquete para simbolizar o funcionamento correto do Sensor Ultrassônico, iluminação, cafeteira e outros equipamentos e sistemas que podem ser ativados nesse processo de automação.





Logo após, colocamos o cabo USB do arduino ao próprio e conectamos ao computador para transferência do código criado para o arduino. O código foi criado com programação C++.

Por fim, para conceder a energia que dará funcionamento aos motores fizemos a utilização de uma fonte de 12V junto a um regulador de tensão para que pudéssemos controlá-la a 6V, tensão essa necessária e mais do que suficiente para um bom funcionamento dos motores lineares.







Na representação acima foi utilizada de um pilhas de 6V, para simbolizar a fonte de 12V com o regulador de tensão.

## 4.2. Código

Para acessar o código acesse: [https://github.com/Xulio1337/TCC\\_Tec-Eletronico](https://github.com/Xulio1337/TCC_Tec-Eletronico)

```
1  const int enPin = 8;
2  const int stepXPin = 2; // X.STEP
3  const int dirXPin = 5;  // X.DIR
4  const int stepYPin = 3; // Y.STEP
5  const int dirYPin = 6;  // Y.DIR
6
7  int pulseWidthMicros = 100; // micro segundos
8  int millisBtwnSteps = 1000;
9
10 const int openDuration = 300;
11 const int pinLedA = 31; //lampada do quarto
12 const int pinLedB = 33; //cafeteira
13 const int pinLedC = 35; //radio
14 const int pinLedD = 39; //Luz do sensor
15 const int pinTrig = 24;
16 const int pinEcho = 26; //branco
17
18 const int Media_Interactions = 10;
19 const int Distance_Treshold = 10;
20 bool has_AutoGate = false;
21 String Serial_Data = "";
22 String Serial1_Data = ""; //Esse segundo Serial_Data serve para exibir
23 // a imagem no celular atraves do modulo Bluetooth
24
25 void setup()
26 {
27   Serial.begin(9600);
28   Serial.setTimeout(10);
29   Serial.flush();
30   Serial.println("Serial USB Inicializada.");
31
32   Serial1.begin(9600);
33   Serial1.setTimeout(10);
34   Serial1.flush();
35   Serial.println("Serial 1 Inicializada.");
36   Serial1.println("Serial 1 Inicializada.");
37   pinMode(enPin, OUTPUT);
38   digitalWrite(enPin, LOW);
39   pinMode(stepXPin, OUTPUT);
40   pinMode(dirXPin, OUTPUT);
41   pinMode(stepYPin, OUTPUT);
42   pinMode(dirYPin, OUTPUT);
43   pinMode(pinLedA, OUTPUT);
44   pinMode(pinLedB, OUTPUT);
45   pinMode(pinLedC, OUTPUT);
46   pinMode(pinLedD, OUTPUT);
47   pinMode(pinTrig, OUTPUT);
48   pinMode(pinEcho, INPUT);
49   Serial.println(F("CNC Shield iniciada"));
50 }
```



```
48 pinMode(pinEcho, INPUT);
49 Serial.println(F("CNC Shield iniciada"));
50 }
51 void loop()
52 {
53
54   Read_Serial();
55   Read_Serial1();
56   distancia();
57 }
58 void Read_Serial()
59 {
60   if (Serial.available() > 0)
61   {
62     int inChar = Serial.read();
63     if (inChar != '\n')
64     {
65       Serial_Data += (char)inChar;
66     }
67     else
68     {
69       Split_String();
70     }
71   }
72 }
73 void Read_Serial1()
74 {
75   if (Serial1.available() > 0)
76   {
77     int inChar = Serial1.read();
78     if (inChar != '\n')
79     {
80       Serial1_Data += (char)inChar;
81     }
82     else
83     {
84       Split_String_1();
85     }
86   }
87 }
88 void Split_String()
89 {
90   int CommaIndex = 0;
91   for (int i = 0; i < Serial_Data.length(); i++)
92   {
93     if (Serial_Data[i] == ',')
94     {
95       CommaIndex = i;
96     }
97   }
98   int State = (Serial_Data.substring(Serial_Data.length() - 2,
99   Serial_Data.length()).toInt();
100   int Port = (Serial_Data.substring(1, CommaIndex)).toInt();
101   Switch_Action(Serial_Data);
102   Serial_Data = "";
103 }
104 void Split_String_1()
105 {
106   Switch_Action(Serial1_Data);
107   //Serial.print("=====> Serial BT DATA ");
108   //Serial.println(Serial1_Data);
109   Serial1_Data = "";
110 }
111 void moveMotor(int stepPin, int dirPin, int duration)
112 {
113   unsigned long startTime = millis();
114   while (millis() - startTime < duration)
115   {
116     digitalWrite(dirPin, HIGH);
117     digitalWrite(stepPin, HIGH);
118     delayMicroseconds(pulseWidthMicros);
119     digitalWrite(stepPin, LOW);
120     delayMicroseconds(millisBtwnSteps);
121   }
122 }
```



```
125     }
126 void reverseMoveMotor(int stepPin, int dirPin, int duration)
127 {
128
129     unsigned long startTime = millis();
130     while (millis() - startTime < duration)
131     {
132
133         digitalWrite(dirPin, LOW); // Reverse
134         digitalWrite(stepPin, HIGH);
135         delayMicroseconds(pulseWidthMicros);
136         digitalWrite(stepPin, LOW);
137         delayMicroseconds(millisBtwnSteps);
138     }
139 }
140 void Switch_Action(String command)
141 {
142     char Value[20];
143     command.toCharArray(Value, 20);
144     Serial.println(Value[0]);
145     switch (Value[0])
146     {
147     case '1':
148         Serial.println(F("Opening Motor X"));
149         Serial1.println(F("Opening Motor X"));
150         moveMotor(stepXPin, dirXPin, openDuration);
151         Serial.println(F("Motor X Stopped"));
152         Serial1.println(F("Motor X Stopped"));
153         break;
154     case '2':
155         Serial.println(F("Closing Motor X"));
156         Serial1.println(F("Closing Motor X"));
157         reverseMoveMotor(stepXPin, dirXPin, openDuration);
158         Serial.println(F("Motor X Stopped"));
159         Serial1.println(F("Motor X Stopped"));
160         break;
161     case '4':
162         Serial.println(F("Opening Motor Y"));
163         Serial1.println(F("Opening Motor Y"));
164         moveMotor(stepYPin, dirYPin, openDuration);
165         Serial.println(F("Motor Y Stopped"));
166         Serial1.println(F("Motor Y Stopped"));
167         break;
168     case '5':
169         Serial.println(F("Closing Motor Y"));
170         Serial1.println(F("Closing Motor Y"));
171         reverseMoveMotor(stepYPin, dirYPin, openDuration);
172         Serial.println(F("Motor Y Stopped"));
173         Serial1.println(F("Motor Y Stopped"));
174         break;
175     case '7':
176         Serial.println(F("Simultaneously Opening Motors X and Y"));
177         Serial1.println(F("Simultaneously Opening Motors X and Y"));
178         moveMotor(stepXPin, dirXPin, openDuration);
179         moveMotor(stepYPin, dirYPin, openDuration);
180         Serial.println(F("Motors X and Y Stopped"));
181         Serial1.println(F("Motors X and Y Stopped"));
182         break;
183     case '8':
184         Serial.println(F("Simultaneously Closing Motors X and Y"));
185         Serial1.println(F("Simultaneously Closing Motors X and Y"));
186         reverseMoveMotor(stepXPin, dirXPin, openDuration);
187         reverseMoveMotor(stepYPin, dirYPin, openDuration);
188         Serial.println(F("Motors X and Y Stopped"));
189         Serial1.println(F("Motors X and Y Stopped"));
190         break;
191     case '3':
192         Serial.println(F("Custom Command: Opening Motor X and Closing Motor Y"));
193         Serial1.println(F("Custom Command: Opening Motor X and Closing Motor Y"));
194         moveMotor(stepXPin, dirXPin, openDuration);
195         reverseMoveMotor(stepYPin, dirYPin, openDuration);
196     }
```



```
200 break;
201 case '6':
202   Serial.println(F("Custom Command: Opening Motor Y and Closing Motor X"));
203   Serial.println(F("Custom Command: Opening Motor Y and Closing Motor X"));
204
205   moveMotor(stepYPin, dirYPin, openDuration);
206   reverseMoveMotor(stepXPin, dirXPin, openDuration);
207   Serial.println(F("Motors X and Y Stopped"));
208   Serial.println(F("Motors X and Y Stopped"));
209   break;
210 case 'a':
211   digitalWrite(pinLedA, !digitalRead(pinLedA)); // Inverte o estado do LED
212   break;
213 case 'b':
214   digitalWrite(pinLedB, !digitalRead(pinLedB));
215   break;
216 case 'c':
217   digitalWrite(pinLedC, !digitalRead(pinLedC));
218   break;
219 case 'd':
220   Serial.println(F("Opening Motor X"));
221   Serial.println(F("Opening Motor X"));
222   moveMotor(stepXPin, dirXPin, openDuration);
223   Serial.println(F("Motor X Stopped"));
224   Serial.println(F("Motor X Stopped"));
225   Serial.println(F("Opening Motor Y"));
226   Serial.println(F("Opening Motor Y"));
227   moveMotor(stepYPin, dirYPin, openDuration);
228   Serial.println(F("Motor Y Stopped"));
229   Serial.println(F("Motor Y Stopped"));
230   delay(100);
231   digitalWrite(pinLedA, LOW);
232   digitalWrite(pinLedC, HIGH);
233
234   delay(100);
235   digitalWrite(pinLedB, HIGH);
236   delay(5000);
237
238   break;
239 case 'n':
240   Serial.println(F("Closing Motor X"));
241   Serial.println(F("Closing Motor X"));
242   reverseMoveMotor(stepXPin, dirXPin, openDuration);
243   Serial.println(F("Motor X Stopped"));
244   Serial.println(F("Motor X Stopped"));
245   Serial.println(F("Closing Motor Y"));
246   Serial.println(F("Closing Motor Y"));
247   reverseMoveMotor(stepYPin, dirYPin, openDuration);
248   Serial.println(F("Motor Y Stopped"));
249   Serial.println(F("Motor Y Stopped"));
250   digitalWrite(pinLedA, LOW);
251   digitalWrite(pinLedB, LOW);
252   digitalWrite(pinLedC, LOW);
253   break;
254 case 't':
255   Serial.println(F("Opening Motor X"));
256   Serial.println(F("Opening Motor X"));
257   moveMotor(stepXPin, dirXPin, openDuration);
258   Serial.println(F("Motor X Stopped"));
259   Serial.println(F("Motor X Stopped"));
260   Serial.println(F("Closing Motor Y"));
261   Serial.println(F("Closing Motor Y"));
262   reverseMoveMotor(stepYPin, dirYPin, openDuration);
263   Serial.println(F("Motor Y Stopped"));
264   Serial.println(F("Motor Y Stopped"));
265   digitalWrite(pinLedA, HIGH);
266   delay(100);
267   digitalWrite(pinLedB, LOW);
268   delay(100);
269   digitalWrite(pinLedC, HIGH);
270   ...
271
272   break;
273 }
274 }
275 }
276 }
277 }
278 void distancia() {
279   digitalWrite(pinTrig, LOW);
280   delayMicroseconds(2);
281   digitalWrite(pinTrig, HIGH);
282   delayMicroseconds(10);
283   digitalWrite(pinTrig, LOW);
284
285   long duration = pulseIn(pinEcho, HIGH);
286   int distance = duration * 0.034 / 2;
287   //Aqui existia um Serial.Print para a distancia, retirado por motivos convencionais
288
289   if (distance <= 5) {
290
291     digitalWrite(pinLedD, HIGH); // Liga o LED
292     reverseMoveMotor(stepXPin, dirXPin, openDuration);
293     Serial.println(F("Opening Motor X"));
294     Serial.println(F("Opening Motor X"));
295     // Aguarda 3 segundos
296     delay(1000);
297     Serial.println(F("aguardando X"));
298     Serial.println(F("aguardando X"));
299     delay(3000);
300     //Inverte o motor após 3 segundos
301     moveMotor(stepXPin, dirXPin, openDuration);
302     digitalWrite(pinLedD, LOW);
303     Serial.println(F("Closing Motor X"));
304     Serial.println(F("Closing Motor X"));
305     // Mantém-se disponível para detecção contínua
306   }
307   // Delay opcional para evitar leituras muito frequentes
308   delay(1000);
309 }
```



## **5. Resultados e Discussão**

### **5.1. Resultados Gerais**

A maquete executa as funções que lhe foram delegadas, como utilizar-se de um sensor ou de comando através do celular para acionar os motores e assim abrir e fechar a porta e janela.

## **6. Conclusão**

O desenvolvimento deste projeto de sala automatizada com controle via Bluetooth demonstrou, na prática, como a automação pode tornar ambientes mais acessíveis, eficientes e inteligentes. Utilizando um Arduino Mega, motores de passo, sensores ultrassônicos e um módulo Bluetooth, conseguimos criar um sistema funcional que permite o controle remoto de portas, janelas e dispositivos simulados através de LEDs.

A implementação do Bluetooth como meio de comunicação foi uma escolha estratégica, permitindo que o sistema funcione independentemente de conexão com a internet, o que o torna uma solução viável para locais remotos ou com infraestrutura limitada. Além disso, a utilização de sensores ultrassônicos para detectar a presença do usuário e acionar a abertura da porta reforça a ideia de um sistema que minimiza a necessidade de intervenção manual, tornando o ambiente mais automatizado e intuitivo.

Durante o desenvolvimento, enfrentamos desafios relacionados à integração dos componentes, calibração dos sensores e estruturação do código. No entanto, essas dificuldades foram superadas com testes, ajustes e otimização do sistema, resultando em um projeto funcional e coerente com os objetivos iniciais.

Por fim, este trabalho não apenas contribui para a compreensão da automação residencial, mas também serve como base para futuras melhorias e expansões. A aplicação de novas tecnologias, como conexão Wi-Fi, inteligência artificial e maior integração com dispositivos IoT, pode ampliar ainda mais as possibilidades desse sistema, tornando-o ainda mais eficiente e aplicável a diferentes cenários do dia a dia.



## Referências

<https://web.tecnico.ulisboa.pt/~ist186955/wordpress/2016/12/14/162/#:~:text=Um%20motor%20DC%20%C3%A9%20um,for%C3%A7as%20produzidas%20por%20campos%20magn%C3%A9ticos>.

<http://www.um.pro.br/arduino/index.php?c=ArduinoUno>

<https://srituhobby.com/how-to-use-a-cnc-shield-step-by-step-guide-with-grbl>

[/ https://smallhammer.cc/product/sd1](https://smallhammer.cc/product/sd1)

<https://www.usinainfo.com.br/sensor-ultrassonico-502>