

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC SYLVIO DE MATTOS CARVALHO

Curso Técnico em Mecatrônica M-Tec

ELVIS FERREIRA DE LIMA

KENIDY RUAN PEREIRA JARRO

SEBASTIÃO HEINZE LEMES MENDES

VICTTOR HENRIQUE GARDINI RAMOS

VINÍCIUS AUGUSTO FAVARIN

BRAÇO ROBÓTICO PROGRAMÁVEL PARA USO ACADÊMICO

Matão, SP

2025

SUMÁRIO

1	INTRODUÇÃO	3
2	OBJETIVOS (GERAL E ESPECÍFICOS)	4
	2.1 Objetivo Geral	4
	2.2 Objetivos Específicos	4
3	DESCRIÇÃO DO PROJETO.....	5
	3.1 Apresentação Geral do Projeto	5
	3.2 Materiais Utilizados e suas Especificações Técnicas	6
4	CRONOGRAMA DE ATIVIDADES	10
	4.1 Atividades e Metas 1º Semestre de 2025	10
	4.2 Atividades e Metas 2º Semestre de 2025	10
5	DESENVOLVIMENTO DO PROJETO	12
	5.1 Descrição do Funcionamento	12
	5.2 Montagem Física do Protótipo	13
	5.3 Desenvolvimento da programação do protótipo.....	17
6	CONSIDERAÇÕES FINAIS	19
	APÊNDICES.....	21

1 INTRODUÇÃO

Este documento apresenta o Projeto Técnico Garra Robótica Programável de Baixo Custo para Uso Acadêmico referente ao Trabalho de Conclusão de Curso de Mecatrônica elaborado pela equipe.

Esse projeto foi escolhido por proporcionar aos integrantes da nossa equipe uma oportunidade de desenvolver habilidades e competências relativas ao curso de Mecatrônica, contribuindo para nossa formação e preparando-nos para novas jornadas.

Além disso, a escolha desse projeto justificou-se pela necessidade de tornar o estudo da robótica mais acessível e aplicável em ambientes educacionais. Ao propor uma garra robótica programável de baixo custo, busca-se compreender o funcionamento dos servomotores e do Arduino, além de proporcionar aos alunos uma ferramenta prática para o aprendizado de automação, eletrônica e programação. Dessa forma, o projeto contribui para a formação técnica dos estudantes, estimulando o interesse pela área da robótica.

Isto posto, este documento encontra-se assim dividido:

No Capítulo 2, apresenta os objetivos geral e específicos do nosso projeto.

No Capítulo 3, é feita uma apresentação geral do projeto e a lista de materiais utilizadas para a montagem física de um protótipo que demonstre o funcionamento do nosso projeto.

O Capítulo 4 apresenta os cronogramas de atividades planejados e cumpridos no 1º e no 2º semestre de 2025 para a concretização do projeto.

O Capítulo 5 faz a descrição detalhada do projeto físico, apresentando o seu funcionamento, a montagem física do seu protótipo e o programa aplicativo desenvolvido.

Na sequência são apresentadas as considerações finais sobre o projeto, apresentando as dificuldades encontradas e contornadas, e o resultado final obtido.

Nas Referências Bibliográficas, são apresentadas toda a bibliografia utilizada para o desenvolvimento desse projeto.

E no Apêndice são apresentados os desenhos técnicos elaborados e o código fonte do aplicativo desenvolvido.

2 OBJETIVOS (GERAL E ESPECÍFICOS)

2.1 Objetivo Geral

O objetivo geral do projeto é:

- Desenvolver uma garra robótica programável de baixo custo, controlada por meio do microcontrolador Arduino, capaz de realizar movimentos de preensão de forma autônoma e semiautônoma, com foco em aplicações educacionais.

2.2 Objetivos Específicos

Os objetivos específicos do projeto são:

- Montar a estrutura da garra robótica, utilizando materiais acessíveis e de baixo custo;
- Implementar o sistema de controle por meio de um microcontrolador Arduino com servomotores;
- Programar o sistema da garra para controlar movimentos de subida, descida, rotação e preensão, possibilitando a execução automática de sequências de movimentação;
- Testar o funcionamento da garra simulando aplicações para a implementação nos ambientes educacionais.

3 DESCRIÇÃO DO PROJETO

3.1 Apresentação Geral do Projeto

O presente projeto consiste no desenvolvimento de uma Garra Robótica Programável de Baixo Custo para Uso Acadêmico, projetada com o objetivo de tornar o estudo de robótica mais acessível e prático para estudantes do ensino médio e cursos técnicos. A iniciativa busca integrar conceitos fundamentais de eletrônica, mecânica, programação e automação, oferecendo um protótipo funcional, econômico e fácil de montar.

A estrutura mecânica da garra foi impressa em 3D utilizando PLA, garantindo baixo custo, leveza e facilidade de montagem. Seu design é composto por uma base fixa, três segmentos articulados e uma pinça de preensão, todos conectados por parafusos M3 que permitem movimentos suaves entre os eixos. O modelo utilizado teve como referência o projeto MeArm, amplamente difundido em ambientes educacionais.

O sistema de controle foi desenvolvido com uma placa Arduino Uno, responsável por interpretar os sinais enviados por quatro potenciômetros e converter esses valores em pulsos PWM para controlar os servomotores. A eletrônica foi montada em uma protoboard de 830 furos, utilizando jumpers e um capacitor de 1000 μ F para estabilização da tensão. O código foi elaborado em linguagem C/C++, permitindo movimentos como subir, descer, girar, mover o braço e abrir/fechar a garra, tanto manualmente quanto de forma automatizada.

Durante os testes experimentais, o protótipo demonstrou bom desempenho, precisão satisfatória e resposta rápida aos comandos, sendo capaz de segurar objetos leves como tampas de garrafa e pequenos componentes. A utilização de impressão 3D reduziu significativamente os custos e possibilitou substituição fácil de peças.

O projeto, além de atingir seus objetivos iniciais, proporcionou uma experiência prática de grande valor educacional, permitindo aprofundamento em áreas como programação embarcada e montagem eletrônica. Assim, este trabalho contribui para a formação técnica dos integrantes e reforça o potencial da robótica de baixo custo como ferramenta pedagógica em escolas e laboratórios.

A Figura 1 apresenta uma foto do nosso protótipo montado.



Figura 1 – Foto do protótipo montado
 FONTE: Elaborado pelos autores (2025).

3.2 Materiais Utilizados e suas Especificações Técnicas

Os materiais utilizados na montagem do nosso protótipo estão apresentados na Tabela 1.

Tabela 1 – Lista de Materiais

Material	Quantidade	Preço Unitário	Preço Total
Estrutura da garra feita com material de impressora 3D (123,54 g - PLA)	1	R\$ 3,14	R\$ 3,14
Placa Arduino UNO R3 (ATmega328P)	1	R\$ 34,27	R\$ 34,27
Servomotores SG90	4	R\$ 11,65	R\$ 46,59
Potenciômetros 1kΩ	4	R\$ 3,90	R\$ 15,60
Capacitor 1000uf	1	R\$ 0,87	R\$ 0,87
Jumpers Macho-Macho	40	R\$ 0,34	R\$ 13,48
Protoboard 830 Furos	1	R\$ 19,00	R\$ 19,00
Fonte de alimentação 5V	1	R\$ 13,90	R\$ 13,90
Porca M3	15	R\$ 0,10	R\$ 1,50
Parafuso M3 20mm	4	R\$ 0,25	R\$ 1,00
Parafuso M3 12mm	11	R\$ 0,20	R\$ 2,20
Parafuso M3 10mm	7	R\$ 0,18	R\$ 1,26
Parafuso M3 8mm	6	R\$ 0,15	R\$ 0,90
Parafuso M3 6mm	9	R\$ 0,12	R\$ 1,08
Jogo Macho manual	1	R\$ 19,50	R\$ 19,50
Vira macho T 3mm a 8 mm	1	R\$ 27,90	R\$ 27,90
Broca Aço Rápido mm 2,5 IRWIN	1	R\$ 6,50	R\$ 6,50
Total			R\$ 209,69

FONTE: Elaborado pelos autores (2025).

Especificações Técnicas dos Materiais Utilizados no Protótipo:

1. Estrutura da garra (PLA – Impressão 3D)

- **Função:** Formar o corpo mecânico da garra robótica, incluindo base, articulações, segmentos e pinça responsável pela apreensão de objetos.
- **Especificações Técnicas:** Material PLA, densidade aproximada 1,24 g/cm³, resistência mecânica moderada, baixa deformação térmica e alto nível de detalhe em impressão FDM. Massa total utilizada: 123,54 g.
- **Justificativa:** O PLA foi escolhido por ser barato, fácil de imprimir e ideal para protótipos educacionais. Permite substituição rápida de peças, baixo custo e boa precisão dimensional para articulações pequenas.

2. Arduino Uno R3 (ATmega328P)

- **Função:** Processar os sinais recebidos dos potenciômetros e enviar comandos PWM para os servomotores.
- **Especificações Técnicas:** Microcontrolador ATmega328P de 8 bits, clock de 16 MHz, 6 entradas analógicas, 14 pinos digitais (6 PWM), interface USB e alimentação de 5V.
- **Justificativa:** Amplamente utilizada em projetos educacionais, possui excelente documentação, é fácil de programar e integra perfeitamente com servomotores e componentes simples.

3. Servomotores SG90

- **Função:** Responsáveis pelos movimentos da garra: base rotativa, dois segmentos articulados e abertura da pinça.
- **Especificações Técnicas:** Alimentação de 4,8–6 V, torque aproximado 1,8 kg/cm, ângulo de rotação de 0° a 180°, controle por sinal PWM.
- **Justificativa:** Servos leves, baratos e suficientes para aplicações de prototipagem. Ideais para projetos com cargas pequenas e excelente custo-benefício.

4. Potenciômetros 1 k Ω

- **Função:** Permitir o controle manual e independente dos quatro servomotores, ajustando o ângulo de cada articulação.
- **Especificações Técnicas:** Resistência de 1 k Ω , sinal analógico proporcional à rotação, compatível com entradas analógicas do Arduino.
- **Justificativa:** Componentes simples, baratos e ideais para controlar servos em baixa tensão. Não exigem circuitos adicionais.

5. Capacitor Eletrolítico 1000 μ F

- **Função:** Estabilizar a tensão da linha de alimentação dos servos, reduzindo oscilações e prevenindo quedas de energia durante movimentos simultâneos.
- **Especificações Técnicas:** Capacitância de 1000 μ F, tensão de trabalho de 16 V, polarizado, ideal para filtragem e amortecimento de picos de corrente.
- **Justificativa:** Fundamental para garantir funcionamento estável dos servomotores, evitando travamentos e resets no Arduino.

6. Protoboard 830 furos

- **Função:** Facilitar a montagem e interligação dos componentes eletrônicos sem necessidade de solda.
- **Especificações Técnicas:** 830 pontos de conexão, linhas positivas e negativas laterais, compatível com jumpers padrão 20–30 AWG.
- **Justificativa:** Ideal para prototipagem rápida, permite reorganizar o circuito facilmente e testar diferentes configurações.

7. Fonte de alimentação 5V

- **Função:** Fornecer energia externa dedicada aos servomotores.
- **Especificações Técnicas:** Saída de 5V, corrente entre 1A e 2A (dependendo do modelo), conexão via plug P4.
- **Justificativa:** Servos não podem ser alimentados diretamente pelo Arduino, portanto a fonte evita sobrecarga e garante potência suficiente.

8. Jumpers Macho-Macho

- **Função:** Conectar o Arduino, os potenciômetros, os servos e a protoboard.
- **Especificações Técnicas:** Cabos flexíveis de 20 cm a 30 cm, compatíveis com 3–5V, padrão macho-macho.
- **Justificativa:** Permitem montagem modular e organizada, essenciais para protótipos didáticos.

9. Parafusos e Porcas M3

- **Função:** Fixação das peças e criação dos pontos de articulação da garra.
- **Especificações Técnicas:** Rosca M3, diversos comprimentos (6 mm, 8 mm, 10 mm, 12 mm, 20 mm), aço zincado.
- **Justificativa:** Tamanho ideal para articulações pequenas, boa resistência e fácil padronização do projeto.

4 CRONOGRAMA DE ATIVIDADES

4.1 Atividades e Metas 1º Semestre de 2025

O semestre (Fev-Jun/25) inicia com Montagem da Equipe e Pesquisa. Em março, ocorre a Definição e Justificativa do Projeto. Em maio, há a Elaboração/Apresentação do PTCC para a Banca e a Aquisição de Materiais. A etapa final, em junho, é dedicada ao Início da Montagem física do Protótipo.

Tabela 2: Cronograma de atividades e metas no 1º Semestre de 2025

Atividade	Fevereiro		Março		Abril		Maio		Junho	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
Montagem da equipe	x									
Pesquisa para definição do projeto	x	x	x							
Justificativa da escolha do projeto			x							
Definição dos objetivos do projeto				x						
Elaboração do projeto				x	x	x				
Elaboração da apresentação do PTCC							x			
Apresentação do PTCC para a Banca								x		
Aquisição dos materiais para montagem do protótipo							x	x	x	
Início da montagem física do protótipo										x
Redação do Manual do Projeto Técnico				x	x	x	x	x	x	x

FONTE: Elaborado pelos autores (2025).

4.2 Atividades e Metas 2º Semestre de 2025

O 2º Semestre (Ago-Dez/25) foca na execução e finalização. Agosto concentra-se na Elaboração da Programação e Construção do Gabinete. O Término da

Montagem física ocorre em outubro. Em outubro e novembro, são realizados Testes Finais e Correções. A Redação do Manual é contínua (Ago-Out). Outubro e novembro inclui a Elaboração/Apresentação do DTCC para a Banca e Correções Finais do Manual e Projeto. E se encerra com a Entrega da Documentação em dezembro.

Tabela 3: Cronograma de atividades e metas no 2º Semestre de 2025

Atividade	Agosto		Setembro		Outubro		Novembro		Dezembro	
	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena	1ª Quinzena	2ª Quinzena
Elaboração da programação do protótipo	x	x	x	x						
Projeto e construção do gabinete do protótipo	x	x	x							
Término da montagem física do projeto				x	x	x				
Testes finais / correções						x	x			
Término da redação do Manual do Projeto Técnico	x	x	x	x	x	x				
Elaboração da apresentação do DTCC						x	x			
Apresentação do DTCC para a Banca								x		
Correções finais do Manual e do projeto								x	x	
Entrega da documentação									x	

FONTE: Elaborado pelos autores (2025).

5 DESENVOLVIMENTO DO PROJETO

5.1 Descrição do Funcionamento

A Figura 2 apresenta o esquema elétrico de toda a parte eletrônica do projeto.

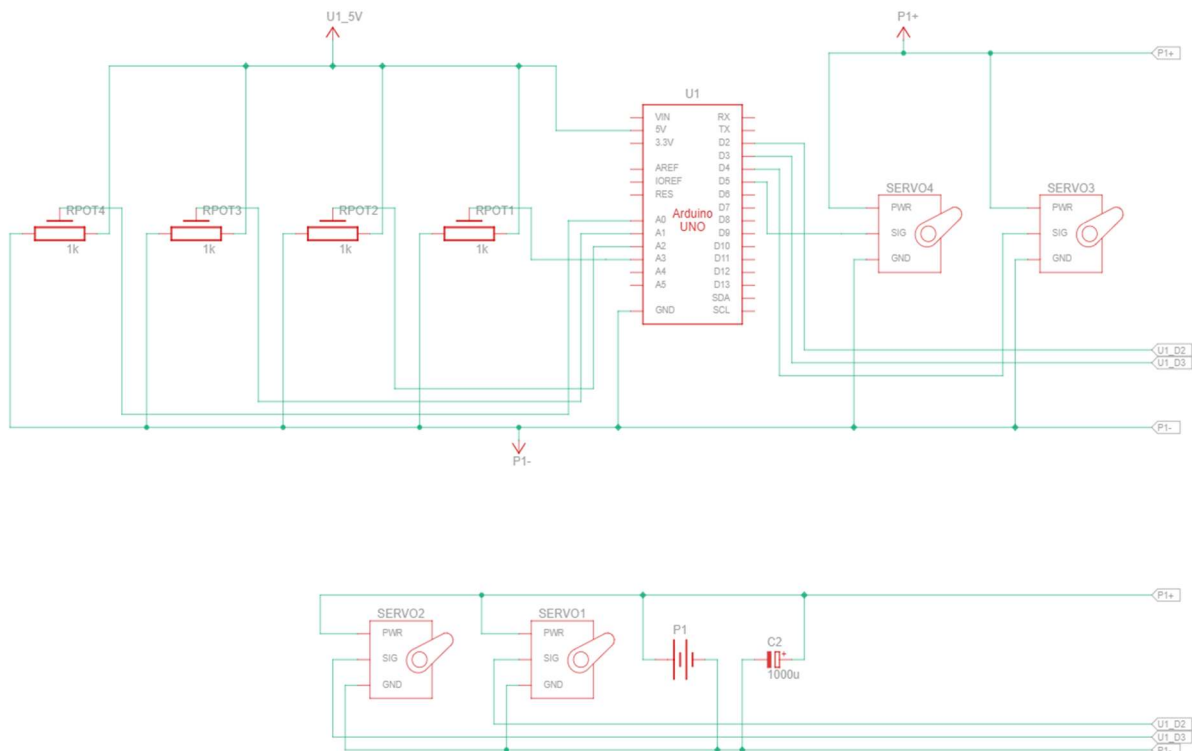


Figura 2 – Esquema elétrico do projeto

FONTE: MANUAL MAKER. Disponível em: <https://www.tinkercad.com/things/1KtkxZG6M7p> (2019).

O funcionamento da parte eletrônica é descrito a seguir.

O sistema eletrônico da garra robótica é responsável por interpretar os comandos do usuário e convertê-los em movimento mecânico nas articulações. O funcionamento inicia-se com a alimentação do circuito: a fonte externa de 5V fornece energia suficiente para os quatro servomotores, enquanto o Arduino Uno é alimentado via USB e pela própria fonte, garantindo estabilização do sistema. Um capacitor eletrolítico de 1000 μF é conectado em paralelo à linha de 5V para reduzir oscilações e impedir quedas de tensão provocadas pelos picos de corrente dos servos.

O controle do movimento é realizado por meio de quatro potenciômetros, cada um correspondente a um eixo da garra (base, braço 1, braço 2 e garra/pinça). Ao girar um potenciômetro, sua resistência interna varia, alterando a tensão de saída no pino

central. Essa tensão, que varia entre 0 e 5V, é lida pelas entradas analógicas do Arduino (A0 a A3). O microcontrolador converte esse sinal analógico em um valor digital de 0 a 1023 e o interpreta como o ângulo desejado pelo respectivo servomotor.

Com base nessa leitura, o Arduino gera sinais PWM (Pulse Width Modulation) nos pinos digitais 2, 3, 4 e 5. Cada sinal PWM possui um pulso entre 1 ms e 2 ms, correspondendo ao ângulo de 0° a 180° do servomotor. Esse sinal é enviado diretamente aos servos, que ajustam suas engrenagens internas para alcançar o ângulo determinado. Dessa forma, o movimento da garra ocorre em tempo real, seguindo exatamente a posição definida pelos potenciômetros.

O Arduino funciona como o centro do sistema, fazendo a leitura contínua dos valores analógicos, convertendo-os, aplicando uma proporcionalidade adequada e comandando cada servomotor de forma independente. Esse processo ocorre centenas de vezes por segundo, garantindo precisão e resposta imediata aos ajustes feitos pelo usuário. Como cada motor possui seu próprio potenciômetro, o operador pode controlar simultaneamente todos os eixos da garra, resultando em movimentos fluidos e intuitivos.

5.2 Montagem Física do Protótipo

As Figuras 3 a 6 apresentam a sequência da montagem física do nosso protótipo.



Figura 3 – Componentes utilizados na parte eletrônica
FONTE: Elaborado pelos autores (2025).

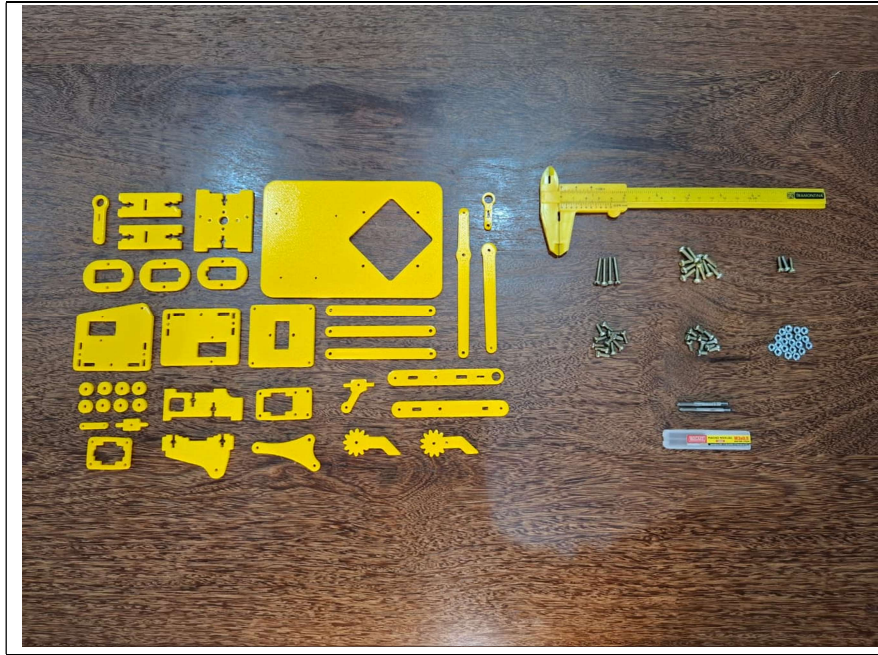


Figura 4 – Componentes utilizados na estrutura
FONTE: Elaborado pelos autores (2025).



Figura 5 – Término da montagem da estrutura
FONTE: Elaborado pelos autores (2025).

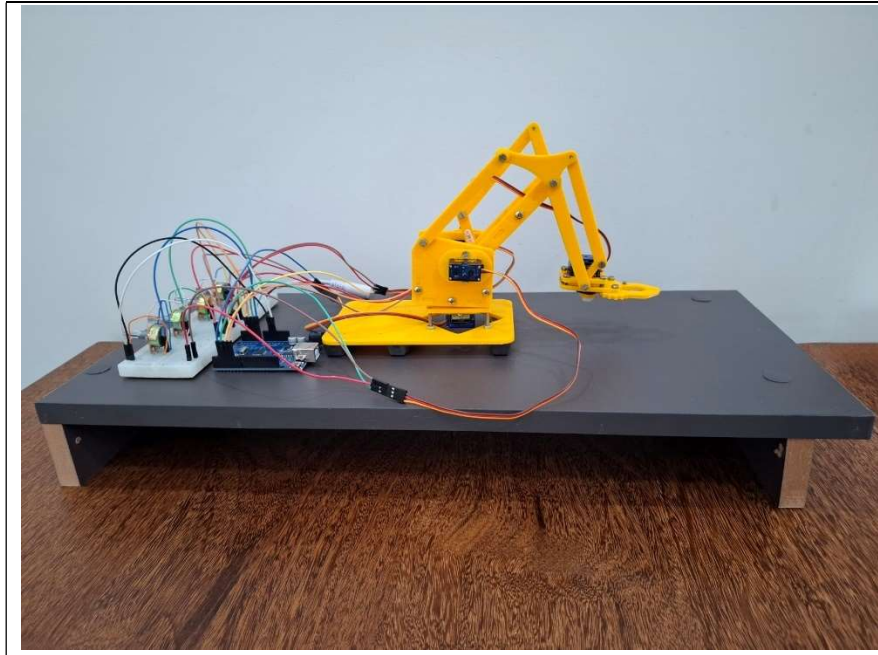


Figura 6 – Término da montagem do protótipo
 FONTE: Elaborado pelos autores (2025).

As Figuras 7 e 8 apresentam os testes finais efetuados em nosso projeto.

```

ProgramaGara
// inicial biblioteca do servomotor
#include <Servo.h>

// define pino dos servos
#define pinServo1 2
#define pinServo2 3
#define pinServo3 4
#define pinServo4 5

// define as portas dos potenciômetros
#define pot1 A0
#define pot2 A1
#define pot3 A2
#define pot4 A3

// movia os servos
Servo serv1,serv2,serv3,serv4;

// cria as variáveis dos ângulos de cada motor
int motor1,motor2,motor3,motor4;

unsigned long motor1diferenca = 1;
const unsigned long intervaloAtualizador = 1000;

void setup() {
  //inicia o monitor serial
  Serial.begin(9600);

  // atribui pino dos servos
  serv1.attach(pinServo1);
  serv2.attach(pinServo2);
  serv3.attach(pinServo3);
  serv4.attach(pinServo4);
}

void loop() {
  // lê os valores dos potenciômetros
  int val1 = analogRead(pot1);
  int val2 = analogRead(pot2);
  int val3 = analogRead(pot3);
  int val4 = analogRead(pot4);

  // converte o valor do potenciômetro para o ângulo do motor
  int ang1 = map(val1, 0, 1023, 0, 180);
  int ang2 = map(val2, 0, 1023, 0, 180);
  int ang3 = map(val3, 0, 1023, 0, 180);
  int ang4 = map(val4, 0, 1023, 0, 180);

  // atualiza o ângulo de cada motor
  motor1 = ang1;
  motor2 = ang2;
  motor3 = ang3;
  motor4 = ang4;

  // escreve no monitor serial
  Serial.print("Pos:");
  Serial.print(ang1);
  Serial.print(" Ângulo Motor:");
  Serial.print(ang2);
  Serial.print(" Ângulo Motor:");
  Serial.print(ang3);
  Serial.print(" Ângulo Motor:");
  Serial.print(ang4);
  Serial.println();

  delay(intervaloAtualizador);
}
  
```

COM7

```

Pos:167 Ângulo Motor:86
Pos:332 Ângulo Motor:44
Pos:1022 Ângulo Motor:100
.....
Pos:1548 Ângulo Motor:156
Pos:1167 Ângulo Motor:86
Pos:332 Ângulo Motor:44
Pos:1022 Ângulo Motor:100
.....
Pos:1548 Ângulo Motor:156
Pos:1167 Ângulo Motor:86
Pos:332 Ângulo Motor:44
Pos:1021 Ângulo Motor:100
.....
Pos:1548 Ângulo Motor:156
Pos:1167 Ângulo Motor:86
Pos:332 Ângulo Motor:44
Pos:1021 Ângulo Motor:100
.....
Pos:1548 Ângulo Motor:156
Pos:1167 Ângulo Motor:86
  
```

Figura 7 – Teste da programação
 FONTE: Elaborado pelos autores (2025).

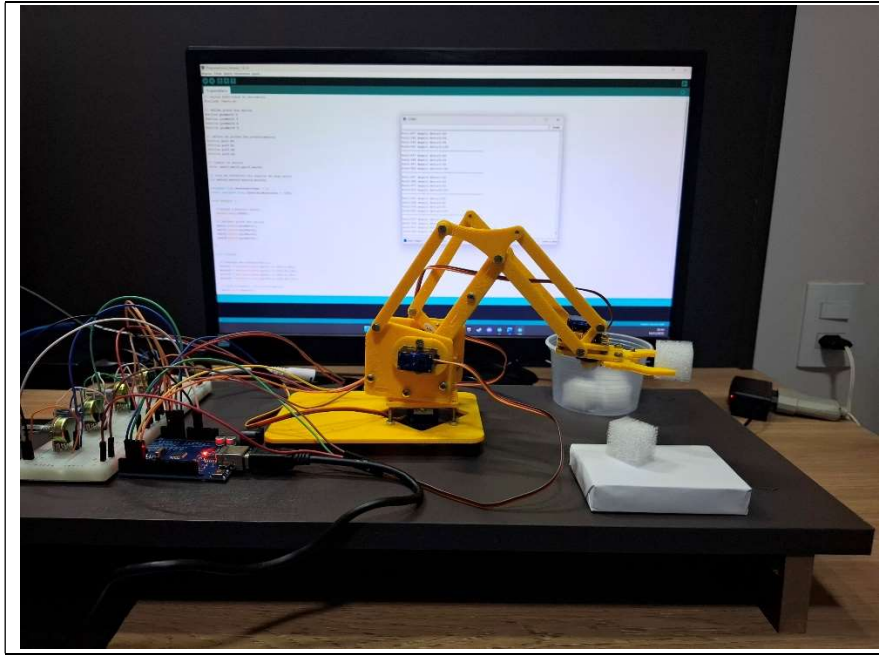


Figura 8 – Teste geral do protótipo
FONTE: Elaborado pelos autores (2025).

5.3 Desenvolvimento da programação do protótipo

A Figura 9 apresenta o fluxograma do programa.

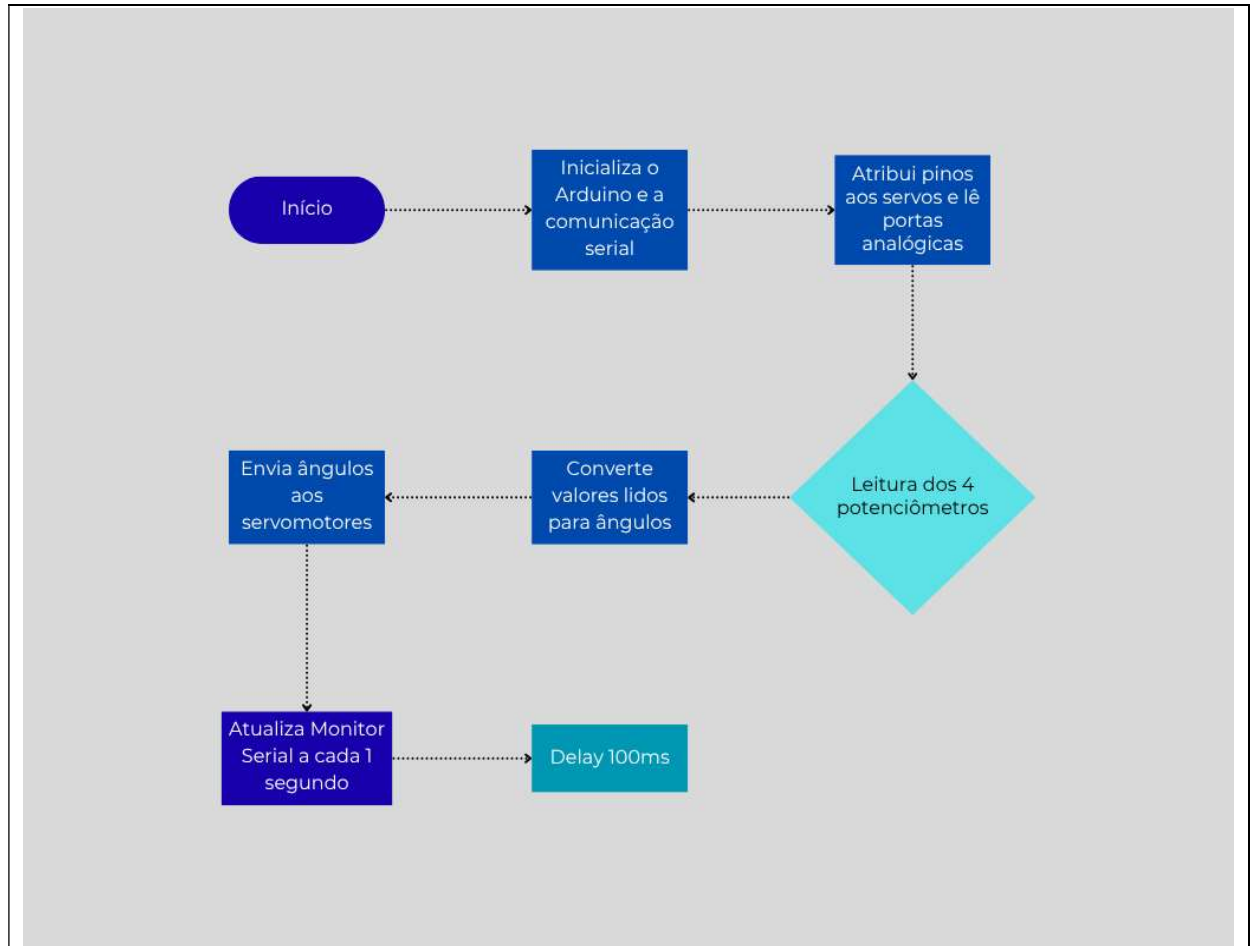


Figura 9 – Fluxograma da programação do protótipo
 FONTE: Elaborado pelos autores (2025).

Conforme apresentado no fluxograma, representa todas as etapas do funcionamento eletrônico e computacional da garra robótica. O sistema inicia com a energização do Arduino Uno e a inicialização da comunicação serial, que será utilizada para monitorar em tempo real os valores lidos pelos potenciômetros e os ângulos enviados aos servomotores.

Em seguida, o programa define os pinos digitais conectados aos quatro servos (2, 3, 4 e 5) e configura como entradas analógicas as portas A0, A1, A2 e A3, onde estão conectados os potenciômetros. Assim, cada potenciômetro passa a ser responsável por controlar um eixo da garra robótica.

Dentro do loop principal, o Arduino realiza continuamente a leitura dos quatro potenciômetros. Esses valores, variando de 0 a 1023, são convertidos por meio da

função `map()` para intervalos específicos de ângulos permitidos para cada motor. Essa etapa é importante para limitar o movimento dos servos, evitando travamentos mecânicos ou danos à estrutura impressa em 3D.

Após o mapeamento, os valores convertidos são enviados aos servos por meio da função `servo.write()`, que posiciona cada motor no ângulo correspondente. Esse processo ocorre em tempo real, fazendo com que a garra responda imediatamente aos ajustes feitos pelo usuário.

Paralelamente, o sistema verifica, a cada 1 segundo se deve atualizar o Monitor Serial. Caso o intervalo seja atingido, o programa envia informações sobre o valor lido de cada potenciômetro e o ângulo correspondente enviado ao servo. Isso permite um acompanhamento em tempo real do comportamento do sistema durante testes e validações.

Ao final do ciclo, o loop aguarda 100 milissegundos antes de reiniciar, garantindo estabilidade nas leituras e no movimento dos motores. Em seguida, o processo recomeça, repetindo continuamente todas as etapas do fluxograma enquanto o sistema estiver ligado.

A programação manual do braço está apresentada no Apêndice B.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento da garra robótica programável de baixo custo atingiu satisfatoriamente os objetivos propostos, demonstrando a viabilidade de construir um sistema funcional, acessível e aplicável ao contexto educacional. O projeto possibilitou a integração de conhecimentos de eletrônica, mecânica e programação, mostrando-se uma ferramenta eficiente para o aprendizado prático em robótica.

Durante o processo, foi possível compreender os princípios de funcionamento de sistemas automatizados, além de aprimorar habilidades em montagem de circuitos, controle de servomotores e desenvolvimento de códigos no microcontrolador Arduino. A garra apresentou bom desempenho nos testes realizados, executando movimentos coordenados de elevação, rotação e preensão de forma precisa e repetitiva.

Apesar dos resultados positivos, pequenas vibrações foram notadas durante movimentos simultâneos, que foram minimizadas com ajustes no código de controle e na calibração dos servomotores.

Por fim, o projeto mostrou-se uma alternativa viável para instituições de ensino que desejam introduzir a robótica de forma prática e econômica, estimulando o interesse dos estudantes por tecnologia, inovação e engenharia. Dessa forma, a garra robótica cumpre não apenas sua função técnica, mas também seu papel formativo dentro do ambiente educacional.

REFERÊNCIAS BIBLIOGRÁFICAS

KALATEC. *Braço robótico.* Blog Kalatec, 14 out. 2024. Disponível em: <https://blog.kalatec.com.br/braco-robotico/>. Acesso em: 3 mai. 2025.

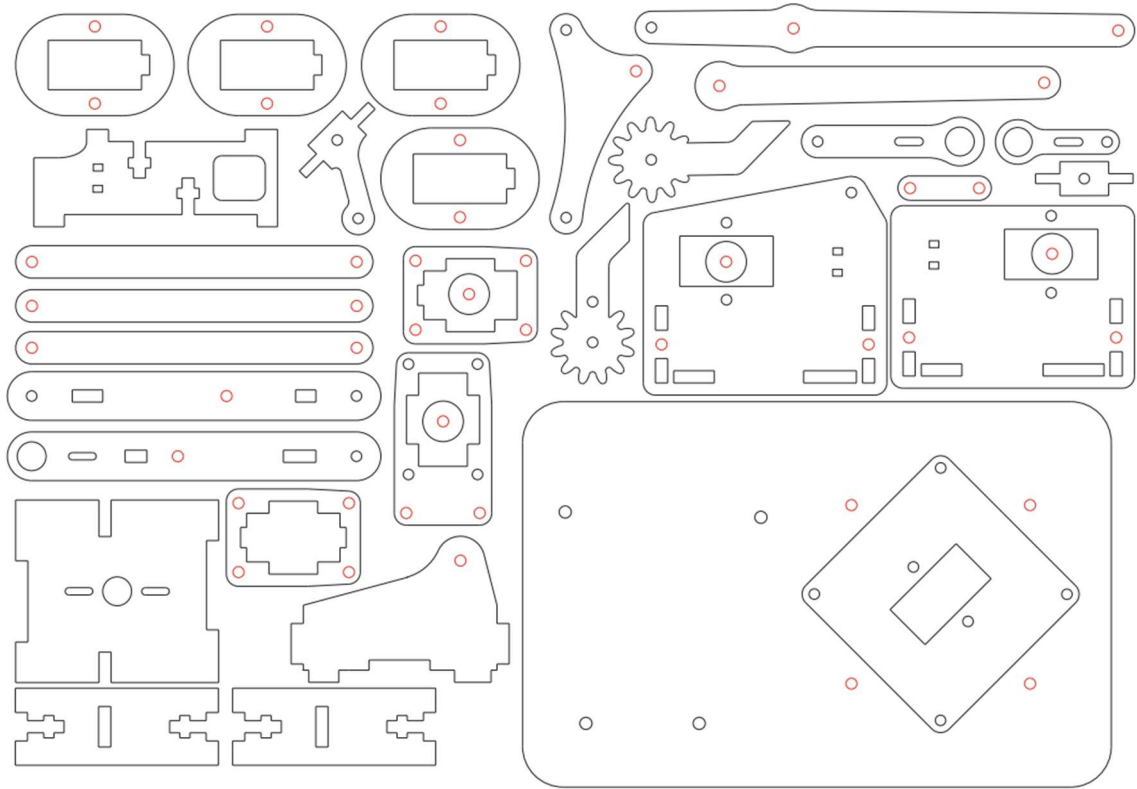
MANUAL DO MUNDO. *Nosso melhor robô de todos os tempos!* (vídeo) YouTube, 28 abr. 2020. Disponível em: <https://youtu.be/Ecw3kCo4AdQ?si=rM5xW8JOV4DdZIA>. Acesso em: 5 abr. 2025.

PHENOPTIX. *MeArm V0.3 – Pocket Sized Robot Arm.* Thingiverse, 14 abr. 2014. Disponível em: <https://www.thingiverse.com/thing:298820>. Acesso em: 7 abr. 2025.

TINKERCAD. *MANUAL MAKER – Braço robótico controlado por potenciômetros.* Tinkercad, 23 abr. 2019. Disponível em: <https://www.tinkercad.com/things/1KtkxZG6M7p>. Acesso em: 20 abr. 2025.

APÊNDICES

Apêndice A: Desenho técnico



Apêndice B: Programação Manual do Braço

```
// inclui biblioteca do servomotor
#include <Servo.h>

// define pinos dos servos
#define pinServ1 2
#define pinServ2 3
#define pinServ3 4
#define pinServ4 5

// define as portas dos potenciômetros
#define pot1 A0
#define pot2 A1
#define pot3 A2
#define pot4 A3

// nomeia os servos
Servo serv1,serv2,serv3,serv4;

// cria as variáveis dos ângulos de cada motor
int motor1,motor2,motor3,motor4;

unsigned long mostradorTimer = 1;
const unsigned long intervaloMostrador = 1000;

void setup() {

    //inicia o monitor serial
    Serial.begin(9600);

    // atribui pinos dos servos
    serv1.attach(pinServ1);
    serv2.attach(pinServ2);
```

```
serv3.attach(pinServ3);
serv4.attach(pinServ4);

}

void loop(){

// leitura dos potenciometros
motor1 = map(analogRead(pot1),0,1023,0,180);
motor2 = map(analogRead(pot2),0,1023,80,120);
motor3 = map(analogRead(pot3),0,1023,0,130);
motor4 = map(analogRead(pot4),0,1023,80,107);

// posicionamento dos potenciometros
serv1.write(motor1);
serv2.write(motor2);
serv3.write(motor3);
serv4.write(motor4);

if ((millis() - mostradorTimer) >= intervaloMostrador) {

// envio para o monitor serial do posicionamentos dos motores
Serial.println("*****");

Serial.print("Pot1:");
Serial.print(analogRead(pot1));
Serial.print(" Angulo Motor1:");
Serial.println(motor1);

Serial.print("Pot2:");
Serial.print(analogRead(pot2));
Serial.print(" Angulo Motor2:");
Serial.println(motor2);
```

```
Serial.print("Pot3:");
Serial.print(analogRead(pot3));
Serial.print(" Angulo Motor3:");
Serial.println(motor3);

Serial.print("Pot4:");
Serial.print(analogRead(pot4));
Serial.print(" Angulo Motor4:");
Serial.println(motor4);

mostradorTimer = millis();
}

// tempo de espera para recomeçar
delay(100);

}
```

Apêndice C: Programação Automática do Braço

```
// inclui biblioteca do servomotor
#include <Servo.h>

// define pinos dos servos
#define pinServ1 2
#define pinServ2 3
#define pinServ3 4
#define pinServ4 5

// nomeia os servos
Servo serv1,serv2,serv3,serv4;

// variavel para movimento gradual
```

```
int posicao;

unsigned long mostradorTimer = 1;
const unsigned long intervaloMostrador = 1000;

void setup() {

    //inicia o monitor serial
    Serial.begin(9600);

    // atribui pinos dos servos
    serv1.attach(pinServ1);
    serv2.attach(pinServ2);
    serv3.attach(pinServ3);
    serv4.attach(pinServ4);

}

void loop(){

    // posicoes iniciais dos servomotores
    serv1.write(93);
    serv2.write(80);
    serv3.write(39);
    serv4.write(88);

    delay(500);

    serv1.write(70);

    delay(500);

    for (posicao = 80; posicao <= 116; posicao += 1) {
        serv2.write(posicao);
```

```
    delay(10);  
  }  
  
  delay(500);  
  
  serv4.write(103);  
  
  delay(500);  
  
  serv2.write(90);  
  
  delay(500);  
  
  serv3.write(65);  
  
  delay(500);  
  
  serv1.write(122);  
  
  delay(600);  
  
  serv4.write(88);  
  
  //tempo de espera para recomeçar  
  delay(3000);  
  
}
```