**Escola SENAI “A. Jacob Lafer”**

DANIEL BARBOSA

JULIA MANDELLI

JULIA RIBEIRO

LETICIA RODRIGUES

**BRAÇO ROBÓTICO**

Santo André

2020

**Escola SENAI “A. Jacob Lafer”**

DANIEL BARBOSA

JULIA MANDELLI

JULIA RIBEIRO

LETICIA RODRIGUES

**BRAÇO ROBÓTICO**

**Natureza e objetivo:**

Projeto de conclusão de curso da área de

Eletroeletrônica da escola SENAI “A Jacob Lafer”.

**Área de concentração**:

Técnico em Eletroeletrônica – 3TA

**Nome dos orientadores:**

Celso Milan

Tales Trocolletto.

Santo André

2020

**SUMÁRIO**

1. INTRODUÇAO.....................................................................................................................9

2. Histórico do ORIGEM..................................................................................................10

[3. Viabilidade Técnica 11](#_Toc279606083)

[4. Viabilidade Econômica..............................................................................................12](#_Toc279606084)

[5. descrição do projeto](#_Toc279606087) 13

5.1 Funcionamento do projeto .................................................................................13

5.1.1 [Diagrama em blocos.................................................................................................](#_Toc279606089)15

5.1.2 [Circuitcipo principal](#_Toc279606090) 16

[5.2 Fotos..............................................................................................................................1](#_Toc279606102)7

5.2.1 protótipo [.................................................................................................................](#_Toc279606103)17

5.2.2 [circuito](#_Toc279606104) 17

[6. Programação](#_Toc279606107) 18

6.3.1 [Fluxograma](#_Toc279606103) 35

[7. Orçamento](#_Toc279606085) 39

[8. Cronograma de Execução](#_Toc279606086) 39

9. [Conclusão](#_Toc279606108) 40

10. [REFERÊNCIAS](#_Toc279606108) 41

11. [ANEXOS (DATASHEET)................................................................................................42.](#_Toc279606109)

1. **INTRODUÇÃO**

Este projeto trata-se de um braço robótico que trabalha com movimentos da “mão” similares ao do ser humano, porém utilizando-se garras. Ele é controlado pelo controlador Atmega 2560 do sistema Arduino, que é responsável por todo o controle do braço. A programação é realizada em linguagem C++ e compilada para a linguagem do microcontrolador, que é o utilizado. A escolha deste microcontrolador deve-se ao fato de possuir entradas A/D (analógicas e digitais) e memória suficiente para o projeto.

Optamos por esse projeto por ser muito utilizado em diversas funções industriais, principalmente, na indústria automobilística até o ambiente educacional (como o SENAI). O equipamento é capaz de realizar movimentos simples, porém extremamente importantes como: mover objetos, escrever, manipular pequenas peças com precisão e fazer impressões em 3D. Quando programado para realizar determinadas tarefas com precisão, tornando as atividades mais fáceis e visando otimizar o tempo e obter melhor custo/benefícios dos seus resultados e onde será usado (como em empresas).

1. **HISTÓRICO DE ORIGEM**

O primeiro robô industrial surgiu na década de 50, pelas mãos do engenheiro norte-americano [George Devol](http://roboticagrupo4.blogspot.com.br/2009/05/historia-da-robotica.html). Ele projetou o “*[Unimate](https://www.robotics.org/joseph-engelberger/unimate.cfm" \t "_blank)”*, um dispositivo de braço robótico que automatizava tarefas em uma fábrica da General Motors em Nova Jersey.

A *Unimation*, empresa fundada por Devol, foi a primeira fabricante de robôs. O robô foi inicialmente visto como forma de entretenimento, mas logo começou a se desenvolver como uma ferramenta fundamental na produção industrial.

Os robôs industriais foram inicialmente criados para executar tarefas comuns. Hoje em dia, eles são multifuncionais. Podem ser reprogramados para exercer diversas funções nos mais diferentes segmentos da indústria. Algumas aplicações típicas dos robôs industriais são de: fundição, pintura, soldagem, montagem, movimentação de cargas, inspeção de produtos, reconhecimento de imagem, e realização de teste.

1. **VIABILIDADE TÉCNICA**

Esse projeto é viável tecnicamente, pois o grupo tem as capacidades técnicas para fazê-lo e conhecimentos em micro controladores PIC, montagem de painel, eletrônica analógica digital e programação. Com essas capacidades conseguiremos montar o nosso projeto. O controle será feito por um sistema microcontrolador por sua velocidade, número de pinos processamento dos sinais analógicos.

A interface de entrada é composta pelo sensor biométrico, que nos dá a informação em protocolo serial. A interface de saída é feita com contatores ou relés ou TRIAC ou SCR.

É viável tecnicamente, pois temos no mercado nacional todos os componentes necessários para fazer, tais como o micro controlador, servos motores, sensores, e tem o seu preço acessível.

1. **VIABILIDADE ECONÔMICA**

Tendo o custo do braço robótico alto, em comparação com a mão-de-obra humana. Mesmo possuindo a ideia de que irá gerar o desemprego (pela falta de mão-de-obra humana) porém quanto maior o consumo maior a produção e a empregabilidade. Possuindo como único gasto sua manutenção.

É o tipo de tecnologia que apresenta um rápido retorno sobre o investimento, além de rápida adaptação à rotina industrial colaborativa. Por isso, se trata de um projeto que traz uma opção de automação importante e viável independentemente do porte da indústria. Tornando-se viável economicamente.

O Braço Robótico possui o valor de R$ 263,60.

A mão de obra se daria em torno de: R$ 2.304,00

|  |
| --- |
| R$ 96,00 (preço p/ dia) x 8 (dias uteis) = R$ 768,00 |
| R$ 768,00 x 3 (meses trabalhados) = R$ 2.304,00 |

Pois,

1. **DESCRIÇÃO DO PROJETO**

O projeto trata-se de uma garra que executa uma sequência o qual é programada, a sua programação é feita manualmente mexendo os seus eixos e salvando cada estado. Depois de toda programação concluída o Braço Robótico atua automaticamente através da sequência feita na programação.

Servo = 4kg / cm

Braço = 4kg / 10 cm = 400 g

O Braço Robótico aguenta 400 gramas.

**5.1 Funcionamento do Projeto**

O projeto terá quatro servos motores, cada servo terá um potenciômetro. O potenciômetro será como um encoder rotativo, sendo utilizado para identificar o ângulo do servo motor para ser configurado cada estado da sequência da garra robótica.

O software será a parte que irá ler os potenciômetros, gravar, executar e mostrar informações no monitor serial, para cada estado, e controlará o ângulo dos servos motores.

Para cada estado terá uma variável armazenando do ângulo de cada servo motor.

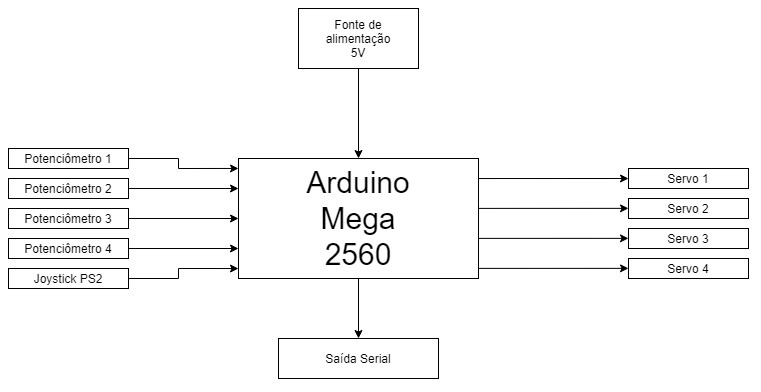
Com os potenciômetros conseguimos fazer a nossa sequência. Cada potenciômetro tem a sua função dentro do programa, o P1 (mover a base do braço robótico), P2 (mover articulação do braço), P3 (mover o braço para cima e para baixo) e por fim P4 (garra aberta e fechada).

Para começar a gravar pressione “X” no menu inicial, a cada movimento que for feito com cada potenciômetro sempre apertar o “X” no controle, para salvar o estado feito. O ângulo dos servos motores acompanha o ângulo dos potenciômetros. Quanto mais estados colocarmos, mais precisa a sequência do Braço Robótico irá ficar fazendo assim o objeto ser carregado com segurança durante o percurso.

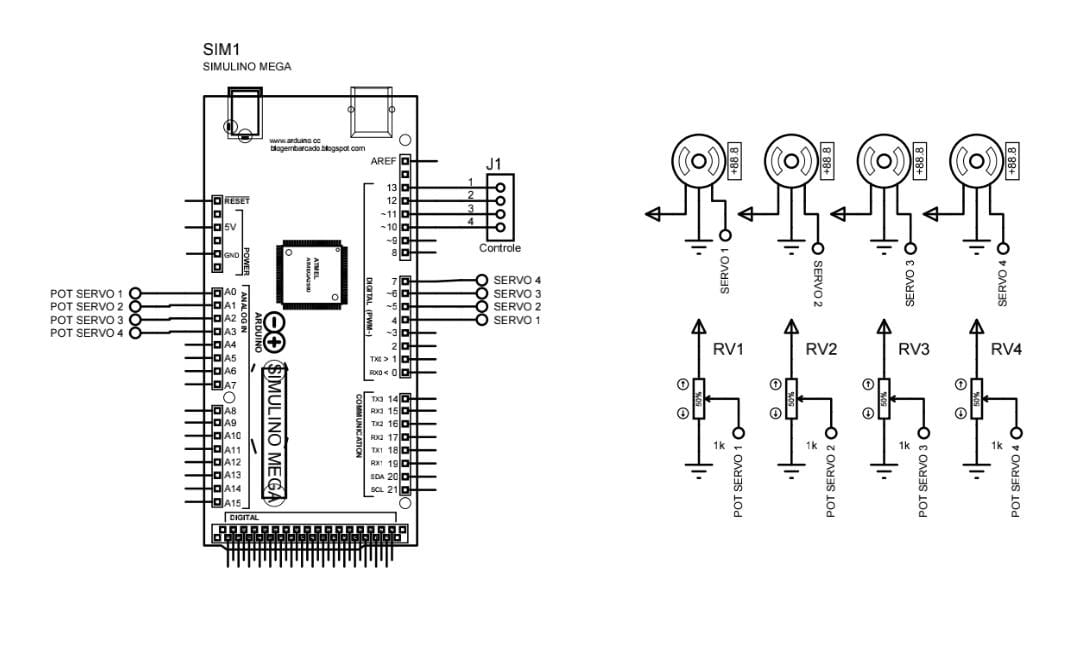
Depois de todos os movimentos salvos apertamos o “Δ” (triângulo), e depois “O” ou “X” para indicar se a sequência irá ser, vai e volta, ou vai e recomeça. Após isso voltará ao menu inicial. Para executar a sequência aperta “O” para definir o ângulo dos servos como o primeiro estado da sequência, e para prosseguir com a sequência aperta “X”, para parar aperta “Δ” e a sequência para e vai para o menu inicial.

Para apagar a sequência aperta “Δ” no menu inicial, e aperta “X”, então podemos configurar de novo a sequência.

* + 1. **Diagrama de Blocos**

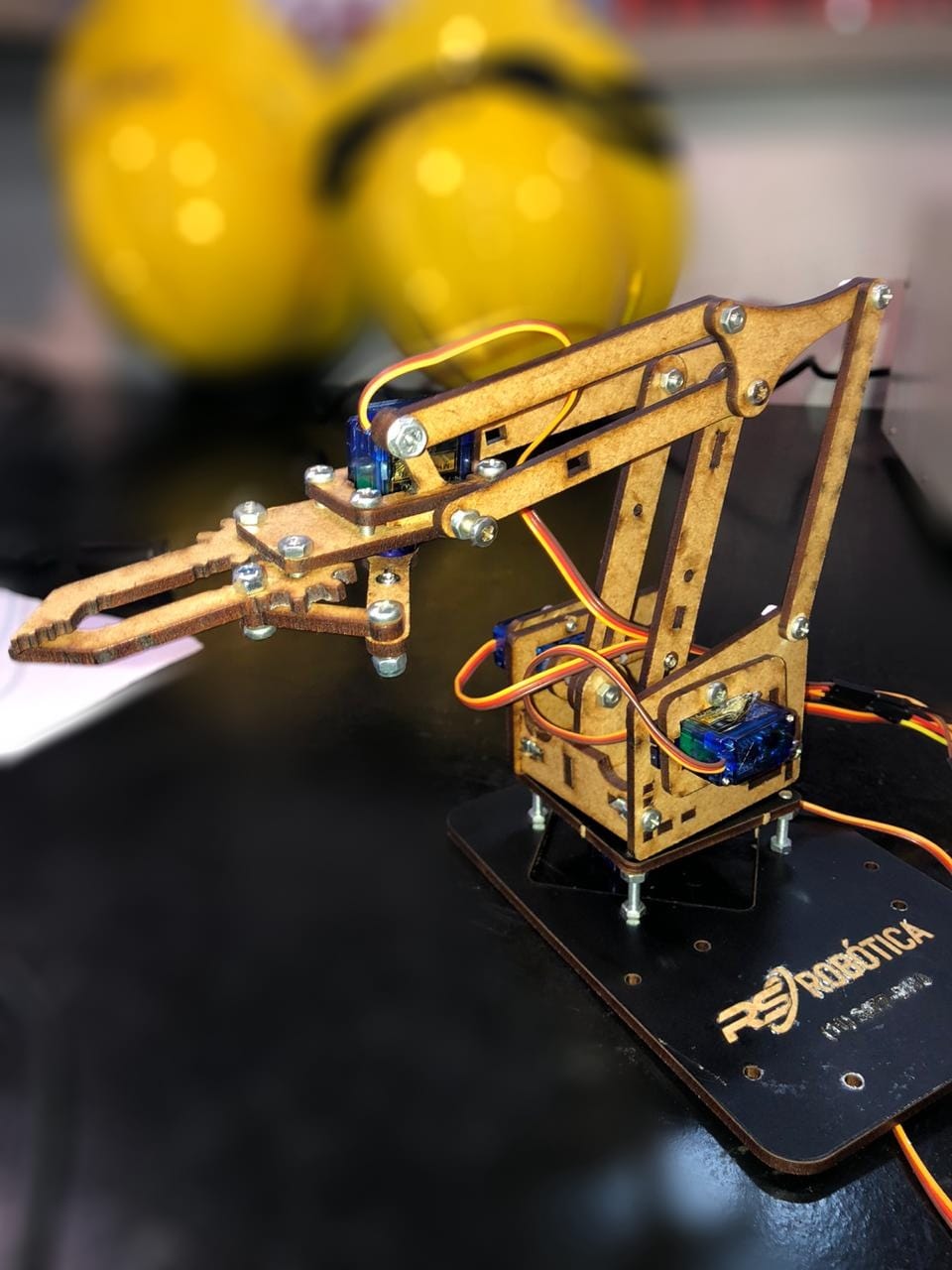


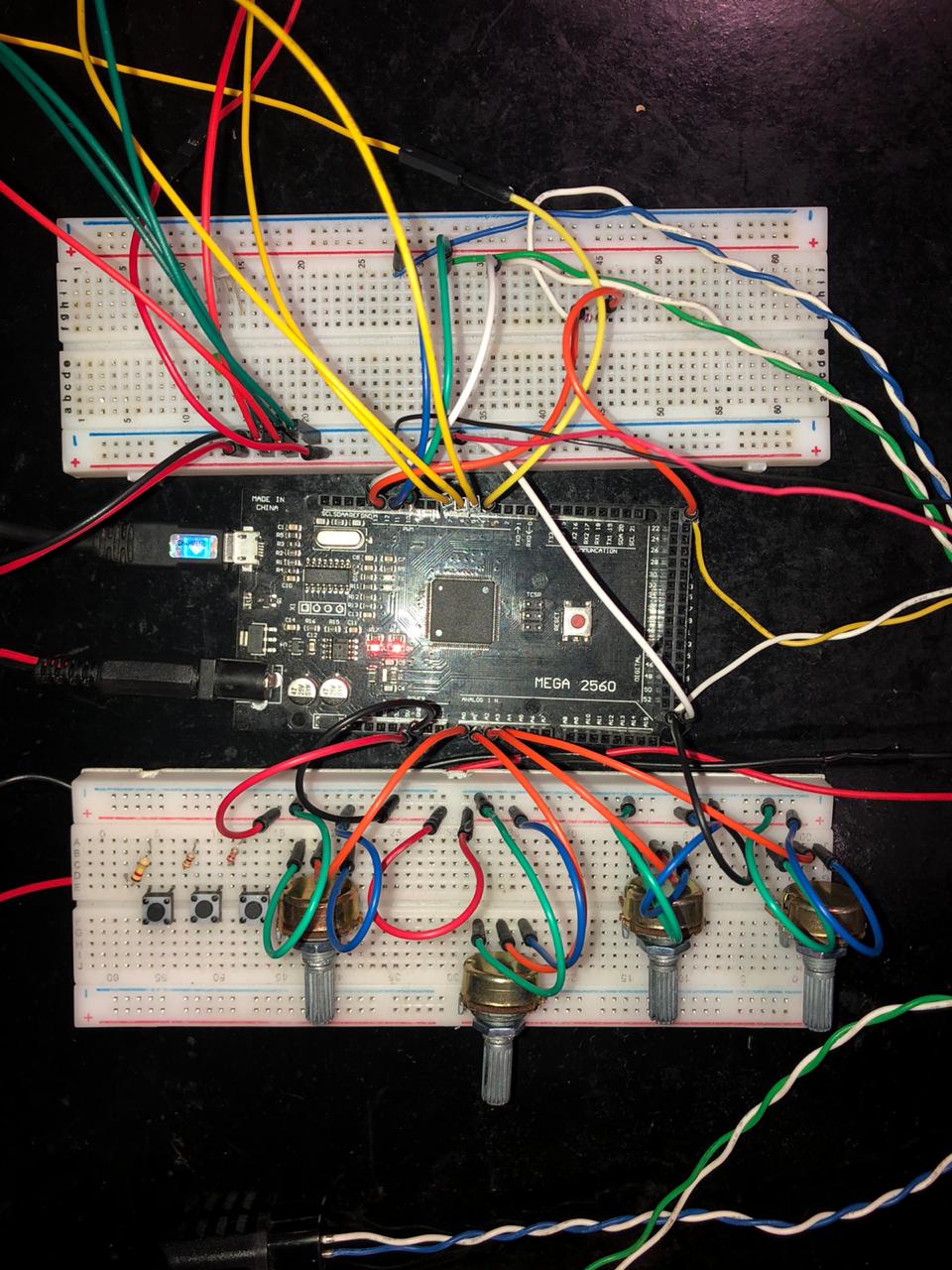
**5.1.2 Circuito Principal**



**5.2 FOTOS**

**5.2.1 Protótipo**

****

**5.2.2 Circuito**

**6.PROGRAMAÇÃO**

#include <Arduino.h>

#include "Servo.h"

#include "lib.h"

#include "PS2X\_lib.h"

PS2X ps2x; // create PS2 Controller Class

// Defines

// Menus

#define PRIMEIRO 0

#define MUDANDO 1

#define EXECULTAR 2

#define EXECULTANDO 3

#define ADICIONAR 4

#define ADICIONANDO 5

#define PERGUNTA\_ADC 6

#define APAGAR 7

// Pinos

// Analógicos - Potênciometros

#define SERVO\_1 map(analogRead(A0), 0, 1023, 0, 180)

#define SERVO\_2 map(analogRead(A1), 0, 1023, 0, 90)

#define SERVO\_3 map(analogRead(A2), 0, 1023, 0, 90)

#define SERVO\_4 map(analogRead(A3), 0, 1023, 50, 0)

// Servos

#define PINO\_SERVO\_1 4

#define PINO\_SERVO\_2 5

#define PINO\_SERVO\_3 6

#define PINO\_SERVO\_4 7

// Controle

#define BOTAO\_X ps2x.ButtonPressed(PSB\_BLUE)

#define BOTAO\_O ps2x.ButtonPressed(PSB\_RED)

#define BOTAO\_A ps2x.ButtonPressed(PSB\_GREEN)

// Variáveis

// Sequencia

estado sequencia[100]; // Vetor da sequência

unsigned char quantidade = 0; // Quantidade de estados na sequência

bool vai\_volta = false; // Configuração se a sequência vai e volta ou só vai e reinicia

bool continua = false; // para ver se continua ou não

// Botões

// Ultimo estado

bool BX\_last = false, // Botão X

BO\_last = false; // Botão O

// Borda de subida

bool BX\_APERTADO = false, // Botão X apertado

BO\_APERTADO = false; // Botão O apertado

// Contador

unsigned char executador = 0; // Contador os estados da sequência

unsigned long contador\_milis = 0; // contador do milis

unsigned int diferenca\_milis = 0; // diferenca do contador

bool comecar\_tempo = false; // Começar a contar tempo

bool frente\_tras = true; // Contagem crescente ou decrescente - false desce - true sobe

// Menu

char menu = 0; // Numero do menu

bool menu\_foi = false; // Menu já apareceu - lógica invertida

// Declaração dos Servos

Servo servo\_1,

servo\_2,

servo\_3,

servo\_4;

// Protótipo de função

// Funções para Menu

void MenuInicial(); // Escrever no Serial o menu inicial

void MenuAtual(); // Atualiza o menu atual

// Funções para botões

void AtulizaBotoes();

void setup() {

// Inicia o Serial

Serial.begin(9600);

// Escreve Inicialização Serial

MenuInicial();

// Define o valor do menu com zero

menu = PRIMEIRO;

// Verifica se o controle está acessível se não erro = 1

if(ps2x.config\_gamepad(13, 11, 10, 12, true, true) != 0) {

Serial.println("Controle não encontrado");

// Se o controle não foi encontrado não execulta mais nada

while (1);

}

// Bagulho do Servo

servo\_1.attach(PINO\_SERVO\_1, 0, 180);

servo\_2.attach(PINO\_SERVO\_2, 0, 90);

servo\_3.attach(PINO\_SERVO\_3);

servo\_4.attach(PINO\_SERVO\_4, 0, 50);

servo\_1.write(100);

servo\_2.write(0);

servo\_3.write(0);

servo\_4.write(0);

}

void loop() {

// Atualiza o estado do controle

ps2x.read\_gamepad(false, false);

// Função para atualizar o Menu

MenuAtual();

// Verficação

// Verfica se o botão X foi apertado

if (BOTAO\_X) {

// Muda para adicionar ou mudar se estiver no menu inicial

if (menu == PRIMEIRO) {

if (quantidade > 0) {

Serial.println("Já configurado");

}

else {

menu = ADICIONAR;

menu\_foi = true;

}

}

// Se estiver no Menu Apagar

if (menu == APAGAR) {

quantidade = 0; // Apagar a sequência

menu = PRIMEIRO; // Voltar ao menu inicial

menu\_foi = true; // Liberar para mostrar o menu

Serial.println("Sequencia apagada\n"); // Fala que a sequencia foi apagada

}

// Se tiver adicionando a sequência, adiciona o estado na sequência

if (menu == ADICIONANDO) {

sequencia[quantidade].ang\_1 = SERVO\_1;

sequencia[quantidade].ang\_2 = SERVO\_2;

sequencia[quantidade].ang\_3 = SERVO\_3;

sequencia[quantidade].ang\_4 = SERVO\_4;

quantidade++;

menu\_foi = true; // Liberar para mostrar o menu

}

// Se tiver execultando a sequência, incrementa o contador para o próximo estado

if (menu == EXECULTANDO) {

// Contador crescente ou decrescente

if (vai\_volta) {

if (frente\_tras)

executador++;

else

executador--;

if (((executador + 1) >= quantidade) && continua) {

frente\_tras = false;

}

if (executador == 0) {

frente\_tras = true;

}

}

else {

if (((executador + 1) >= quantidade) && continua)

executador = 0;

else

executador++;

}

if ((!continua) && (((executador) >= quantidade) || (executador == 1 && frente\_tras))) {

menu = PRIMEIRO;

executador = 0;

}

menu\_foi = true;

}

// Se tiver na pergunta do adiciona, vai e volta recebe true

if (menu == PERGUNTA\_ADC) {

vai\_volta = true;

menu = PRIMEIRO;

menu\_foi = true;

Serial.println("Vai e volta selecionado\n");

}

}

// Verfica se o botão O foi apertado

if (BOTAO\_O) {

// Se a quantidade for zero então não a sequencia não foi configurada

if (quantidade == 0) {

Serial.println("Sequencia nao configurada");

}

// Se a sequencia foi configurada e tiver no 1º menu, vai pro menu execultar

else if (menu == PRIMEIRO) {

executador = 0;

menu = EXECULTAR;

menu\_foi = true; // Liberar para mostrar o menu

}

// Se estiver no Menu apagar

if (menu == APAGAR) {

menu = PRIMEIRO; // Voltar ao menu Inicial

menu\_foi = true; // Liberar para mostrar o menu

Serial.println("Sequencia nao apagada\n"); // Fala que a sequencia não foi apagada

}

// Se tiver na pergunta do adiciona, vai e volta recebe false

if (menu == PERGUNTA\_ADC) {

vai\_volta = false;

menu = PRIMEIRO;

menu\_foi = true;

Serial.println("Vai e comeca de novo selecionado\n");

}

}

// Verfica se o botão A (Triângulo) foi apertado

if (BOTAO\_A) {

// Verifica se está no primeiro menu

if (menu == PRIMEIRO) {

// Quantidade é 0

if (quantidade == 0) {

Serial.println("Nao ha sequencia");

} else {

menu = APAGAR;

menu\_foi = true;

}}

// Verifica se está adicionando a sequência se sim, para de adicionar

if (menu == ADICIONANDO) {

Serial.println("Nao tem");

menu = PERGUNTA\_ADC;

menu\_foi = true;

}

// Verifica se está execultano se sim para de execultar

if (menu == EXECULTANDO) {

continua = false;

}

}

// Vai colocando o angulo

if (menu == ADICIONANDO) {

servo\_1.write(SERVO\_1);

servo\_2.write(SERVO\_2);

servo\_3.write(SERVO\_3);

servo\_4.write(SERVO\_4);

}

}

// Funções

// Menu Inicial

void MenuInicial() {

Serial.println("--------------------");

Serial.println("------Programa------");

Serial.println("------do Braco------");

Serial.println("------Robotico------");

Serial.println("--------------------");

Serial.println("Selecione:");

Serial.println(" |-> [ X ] Adionar Sequencia");

Serial.println(" |-> [ O ] Executar Sequencia");

Serial.println(" |-> [ A ] Apagar Sequencia");

}

// Atualiza o menu atual

void MenuAtual() {

// Se o menu ainda não apareceu

if (menu\_foi) {

// switch para cada menu

switch (menu) {

// Menu Inicial

case PRIMEIRO:

MenuInicial(); // Mostar o menu inicial

menu\_foi = false; // tornar falso o menu\_foi

break;

// Execultar a sequência

case EXECULTAR:

// Começar a execultar

Serial.println("--------------------");

Serial.println("Executando:");

continua = true;

menu = EXECULTANDO;

break;

// Executando a sequência

case EXECULTANDO:

Serial.print("Posicao ");

Serial.print(executador + 1);

Serial.print(": ");

Serial.print(sequencia[executador].ang\_1);

Serial.print(" ");

Serial.print(sequencia[executador].ang\_2);

Serial.print(" ");

Serial.print(sequencia[executador].ang\_3);

Serial.print(" ");

Serial.println(sequencia[executador].ang\_4);

servo\_1.write(sequencia[executador].ang\_1);

servo\_2.write(sequencia[executador].ang\_2);

servo\_3.write(sequencia[executador].ang\_3);

servo\_4.write(sequencia[executador].ang\_4);

menu\_foi = false; // tornar falso o menu\_foi

break;

// Adicionar sequência

case ADICIONAR:

Serial.println("Criar sequencia:");

Serial.println("Coloque na posicao e aperte [ X ]");

Serial.println("Para parar precione [ A ]");

Serial.print("\nPosicao 1: ");

menu = ADICIONANDO;

menu\_foi = false; // tornar falso o menu\_foi

break;

// Adicionando Sequência

case ADICIONANDO:

Serial.print(sequencia[quantidade - 1].ang\_1);

Serial.print(" ");

Serial.print(sequencia[quantidade - 1].ang\_2);

Serial.print(" ");

Serial.print(sequencia[quantidade - 1].ang\_3);

Serial.print(" ");

Serial.println(sequencia[quantidade - 1].ang\_4);

Serial.print("Posicao ");

Serial.print(quantidade + 1);

Serial.print(": ");

menu\_foi = false; // tornar falso o menu\_foi

break;

// Pergunta se vai e volta ou se vai e reinicia

case PERGUNTA\_ADC:

Serial.println("Sequência vai e volta [ X ]");

Serial.println("Sequência vai e comeca de novo [ O ]");

menu\_foi = false; // tornar falso o menu\_foi

break;

// Pergunta se tem certeza, se sim apaga a sequencia

case APAGAR:

Serial.print("Tem certeza que quer apagar a sequencia? [ SIM -> X ] [ NAO -> O ] -> ");

menu\_foi = false; // tornar falso o menu\_foi

break;

default:

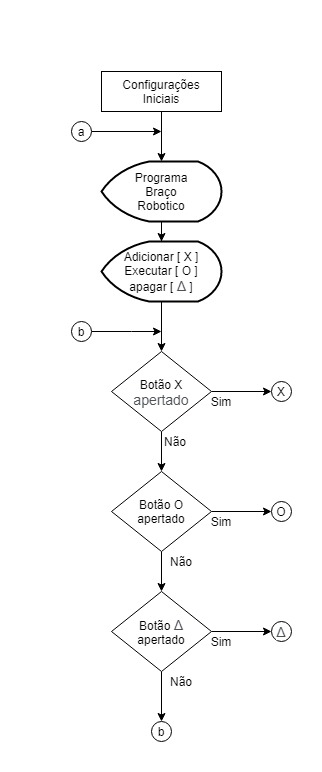
break;

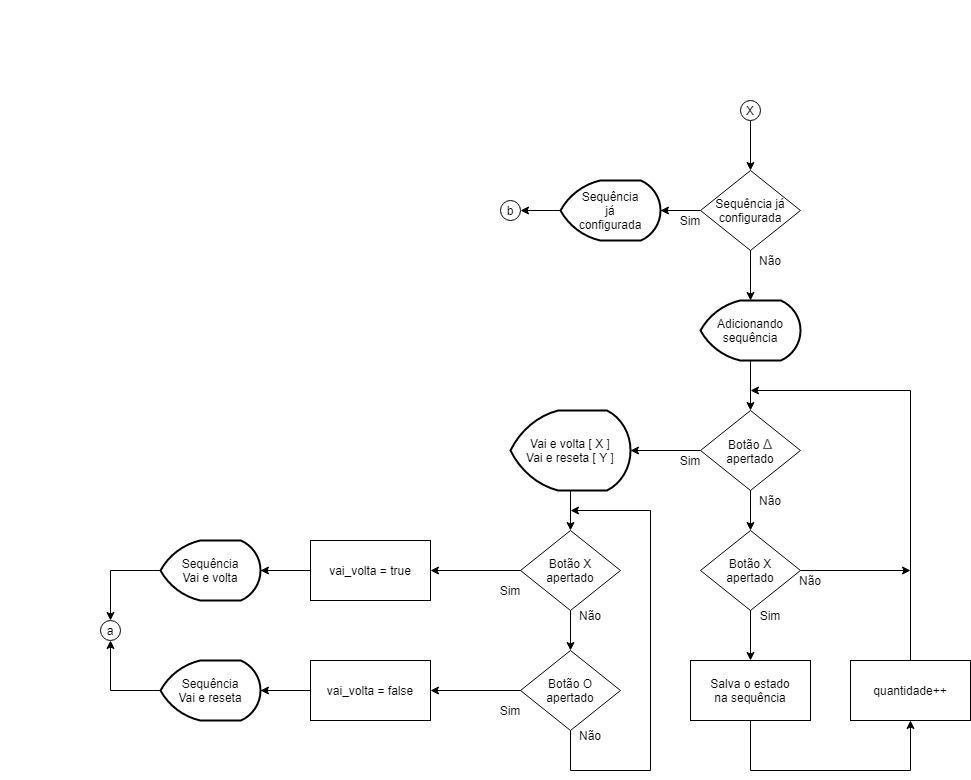
}

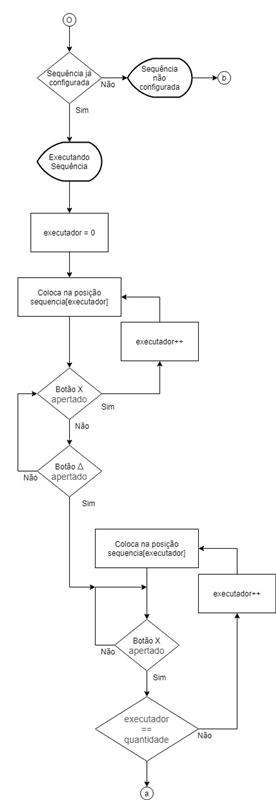
}

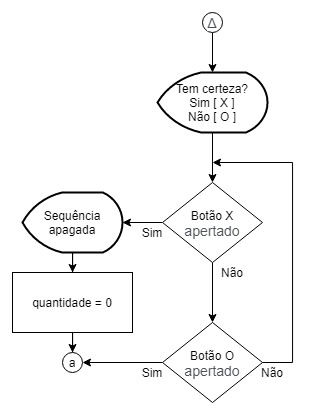
}

**6.3.1 Fluxograma**







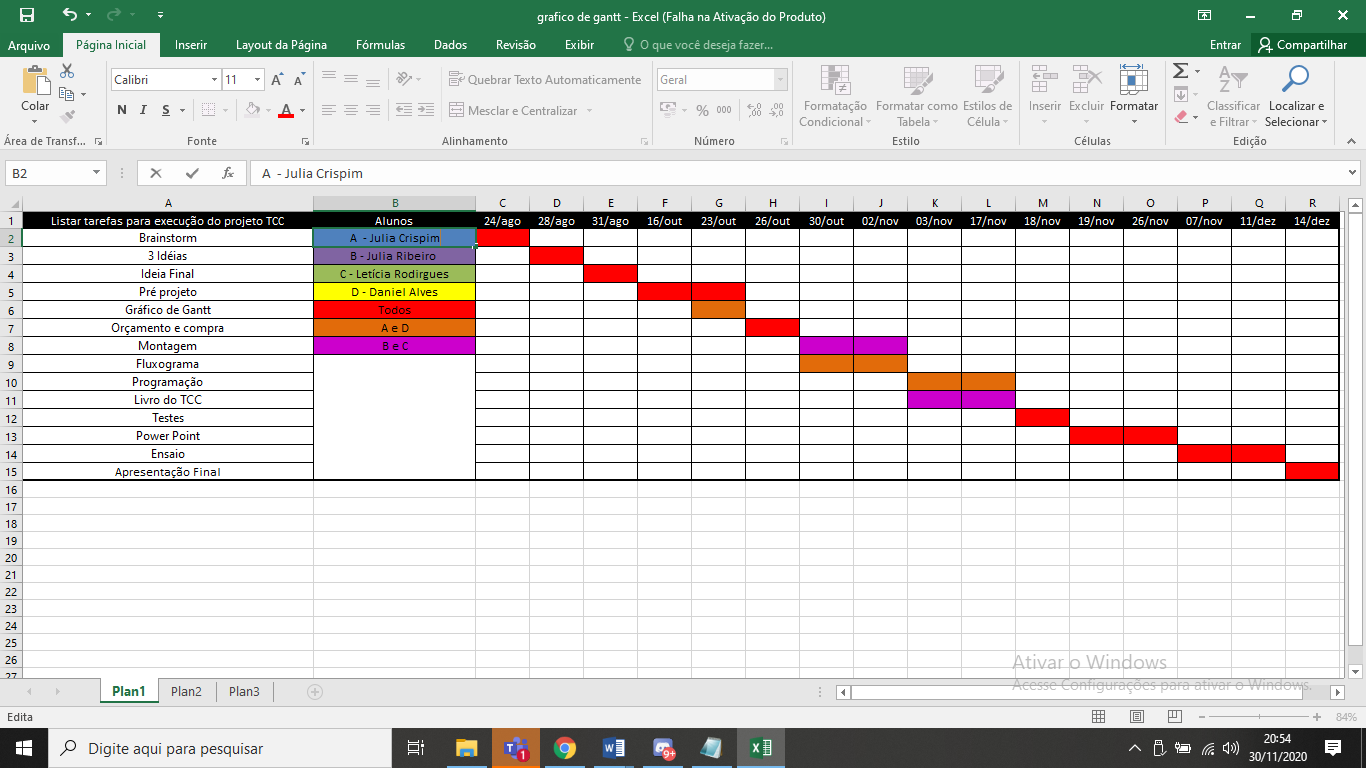


**7. ORÇAMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quant.** | **Unidade** | **Descrição dos Componentes** | **Valor**  **Unitário**  **(R$)** | **Valor**  **Total**  **(R$)** |
| 1 | PÇ | Arduino Mega (Atmega 2560) | R$70,00 | R$70,00 |
| 4 | PÇ | Servo Motor | R$25,90 | R$103,60 |
| 4 | PÇ | Potenciômetro | R$2,50 | R$10,00 |
| 1 | PÇ | Braço-Robótico | R$45,00 | R$45,00 |
| 1 | PÇ | Controle Analógico | R$35,00 | R$35,00 |
| TOTAL | | | | R$263,60 |

**8. CRONOGRAMA DE EXECUÇÃO**

Gráfico de Gantt



**9. CONCLUSÃO**

Diante a realização deste projeto, que foi feito em meio a uma pandemia, todos necessitaram se adaptar para que fosse possível concluir o curso com um excelente TCC.

Mesmo em padrões normais, é necessário o trabalho em grupo e comunicação entre todos, e ainda mais em um momento como esse que estamos distantes um dos outros, dificultando a comunicação. Desta forma, foi preciso um grande esforço de cada um do grupo para que conseguíssemos finalizar este projeto. Cada integrante do grupo ficou responsável, como um “líder”, pela parte na qual se tem facilidade, para que a gestão de realização fosse prática e todos entendessem o projeto por completo.

A realização deste projeto, nos fez refletir a respeito das industrias e sua automatização, como isso afeta a geração de empego, mas também o outro lado da história, como por exemplo a substituição do trabalho da mão de obra humana pela de um robô, além de diminuir o número de acidentes dentro de um setor que necessita de uma rapidez e que frequentemente ocasiona acidentes. Porém, essa tecnologia depois de implementada são necessários técnicos para realizar a manutenção de tais equipamentos, como o do próprio Braço-Robótico. E assim, são gerados empregos, que podem ser facilmente adquiridos, já que atualmente temos diversas escolas técnicas, como o SENAI, que disponibilizam cursos voltados a indústria.

Como melhoria, poderia ser feita a troca do material do braço robótico de MDF por um mais resistente, como o de acrílico. E fazer a montagem utilizando placas ao invés do protoboard.

A maior dificuldade enfrentada foi a montagem do protótipo, mas foi realizado apesar das dificuldades encontradas e finalizado com sucesso.

**10.** **REFERÊNCIAS**

GAMERO, Isis. **História de origem dos Robôs na Industria.**

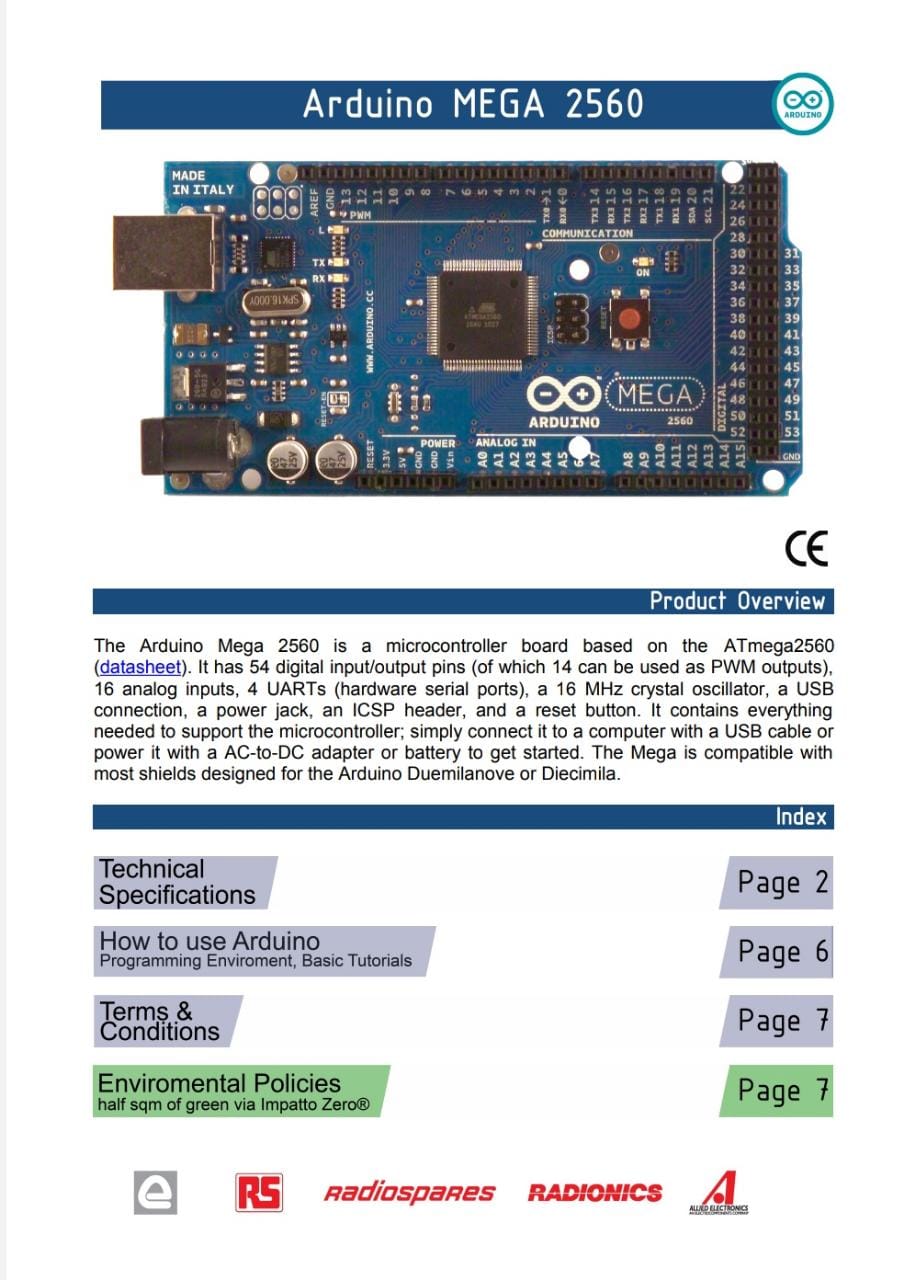
<https://www.pollux.com.br/blog/robos-industriais-tudo-o-que-voce-precisa-saber/#:~:text=O%20primeiro%20rob%C3%B4%20industrial%20surgiu,General%20Motors%20em%20Nova%20Jersey>

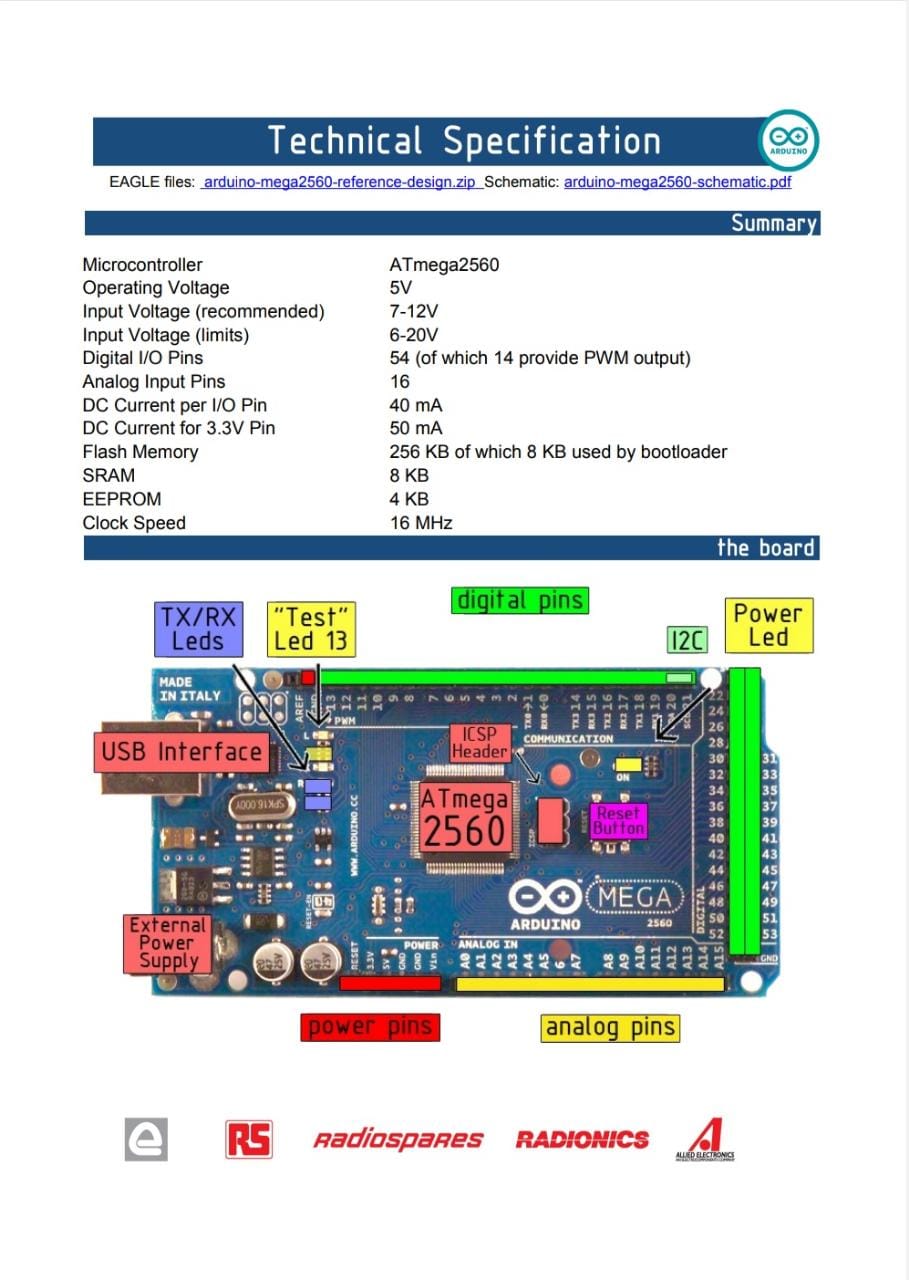
DATASHEET: Arduino e Servo Motor (PDF)

Disponível p/Download no Google

**11.** **ANEXOS DATASHEET’s**

Arduíno





Servo Motor

