

UNIVERSIDADE SÃO JUDAS TADEU

Carlos Damian Viasoli

Carlos Henrique Ferreira Souza

Erick Tadayoshi de Oliveira Kurota

Mateus Ferreira Dias

Vanessa Venancio Sobrinho

Robô autônomo para separação e entrega automática de pedidos
realizados via aplicativo

São Paulo

2021

Carlos Damian Viasoli
Carlos Henrique Ferreira Souza
Erick Tadayoshi de Oliveira Kurota
Mateus Ferreira Dias
Vanessa Venancio Sobrinho

Robô autônomo para separação e entrega automática de pedidos
realizados via aplicativo

Monografia apresentada à
Universidade São Judas Tadeu,
como requisito parcial para a
obtenção do título de Engenheiro de
Controle e Automação.

Orientador: Prof. Dr. Carlos López
Noriega

São Paulo

2021

FICHA DE APROVAÇÃO

Data: _____

Horário: _____

Sala: _____

Título: Robô autônomo para separação e entrega automática de pedidos realizados via aplicativo

Nome completo dos alunos	RA
Carlos Damian Viasoli	81726042
Carlos Henrique Ferreira Souza	816114021
Erick Tadayoshi De Oliveira Kurota	818117403
Mateus Ferreira Dias	819118015
Vanessa Venancio Sobrinho	81726041

Observações sobre o trabalho:

Professores da Banca examinadora	Assinatura
Prof. Dr. Carlos López Noriega	
Prof. Fernando Russo Júnior	
Prof. Me. Mairlos Parra Navarro	

Resultado: _____

DEDICATÓRIA

Dedico este trabalho aos meus amigos de turma que se dedicaram e se empenharam para que fosse possível terminar o projeto, juntamente com os professores que auxiliaram durante todo o projeto, e aqueles que nos apoiaram a todo momento fazendo com que este trabalho fosse possível.

AGRADECIMENTOS

A Deus por ter nos dado saúde e força para enfrentarmos essa longa jornada. Aos nossos familiares, pelos incentivos nas horas difíceis, de desânimo e cansaço. Aos nossos colegas que nos ajudaram no decorrer da realização deste trabalho e a todos que nos auxiliaram, direta ou indiretamente.

Agradecemos e dedicamos o nosso projeto ao nosso orientador Prof. Carlos López Noriega por ter nos orientado e nos ajudado durante todo o trabalho.

O nosso muito obrigado!

RESUMO

Este trabalho de conclusão de curso constitui-se de um sistema inteligente de uma célula robótica móvel responsável pela separação e entrega automática de pedidos realizados via aplicativo, esse sistema é capaz de receber as informações do aplicativo via comunicação *wi-fi* com a placa microcontrolador Esp32, tratar e armazenar de forma a realizar as movimentações, separações e entregas de acordo com o pedido desejado do cliente. A abordagem utilizada na concepção do sistema para realizar a movimentação da célula robótica móvel foi a utilização de um braço robótico acoplado em um carrinho e a linguagem de programação C++, para controlar o braço robótico foi utilizado uma placa que utiliza comunicação I2C para facilitar no controle dos eixos da célula, além de portas PWM para fazer o controle dos servos para a movimentação do robô, já o aplicativo foi desenvolvido na plataforma MIT App Inventor do Google, para estabelecer comunicação entre o aplicativo e o microcontrolador, foi utilizado a plataforma de banco de dados em nuvem Google Firebase. O Firebase recebe as informações de compra realizado no aplicativo e redireciona os dados para que o microcontrolador possa separar os produtos.

Palavras-chave: Célula robótica; Separação automática; Internet das Coisas; Mercado Autônomo; Braço robótico.

ABSTRACT

This course conclusion work consists of an intelligent system of a mobile robotic cell responsible for the automatic separation and delivery of orders made via the application, this system is capable of receiving application information via wi-fi communication with an Esp32 microcontroller board, treat and store in a way to carry out movements, separations and deliveries according to the customer's desired order. The approach applied in the design of the system to carry out the movement of the mobile robotic cell was the use of a robotic arm coupled to a cart and the programming language C ++, control for the robotic arm was used a board that uses I2C communication to facilitate no control of the cell axes, in addition to PWM ports to control the servos for robot movement, the application was developed on Google's MIT App Inventor platform, to establish communication between the application and the microcontroller, using the platform of Google Firebase cloud database. Firebase receives the purchase information made in the application and redirects the data so that the microcontroller can separate the products. The application and redirects the data so that the microcontroller can separate the products.

Keywords: Robotic cell; Automatic separation; Internet of Things; Autonomous Market; Robotic arm.

LISTA DE FIGURAS

Figura 1 - Partes de um braço robótico (Fonte: https://slideplayer.com.br)	22
Figura 2 - Tipos de robôs cartesianos (Fonte: https://www.citisystems.com.br/).....	23
Figura 3– Robô SCARA (Fonte: https://www.citisystems.com.br)	24
Figura 4 – Tipos de robôs articulados (Fonte: https://www.citisystems.com.br).....	24
Figura 5 - Robô KMR <i>iiwa</i> da KUKA (Fonte: https://face-aluminium.com).....	25
Figura 6- Tipos de sistemas de AGVs (Fonte: https://www.arpapress.com).....	27
Figura 7 - Modelo da lanchonete <i>fast food</i> Bionicook (Fonte: https://exame.com).....	29
Figura 8 - Robô separador KUKA da Bionicook (Fonte: https://www.gazetadopobo.com.br).....	30
Figura 9 - Peça para fixar os servos dos eixos 1,2,3 e 4 (Fonte: Toppingmall).....	36
Figura 10 - Fixação entre o eixo 2 e 3 (Fonte: Toppingmall).....	36
Figura 11 - Formato em L para fixação entre os eixos 3 e 4 (Fonte: Toppingmall)	37
Figura 12 - Base para fixação de todos os eixos do braço robótico (Fonte: Toppingmall)	37
Figura 13 - Disco de alumínio para conectar na engrenagem dos servos com as estruturas de alumínio (Fonte: Toppingmall).....	38
Figura 14 - Garra para pegar as peças, acoplada ao eixo 6 do robô (Fonte: Toppingmall)	38
Figura 15 - Abertura máxima do robô (necessário desconsiderar parte do meio que foi removida) (Fonte: Toppingmall).....	39
Figura 16 - Servo motor TD-8325MG utilizado nos eixos 2 e 3 (Fonte: Toppingmall).....	40
Figura 17 - Servo motor MG996R utilizado nos eixos 1,4,5 e 6 (Fonte: Toppingmall).....	40

Figura 18 – Desenho técnico: Tamanho do servo (Fonte: Toppingmall)	41
Figura 19 - Base para fixar no carrinho (Fonte: Elaborado pelo autor)	42
Figura 20 - Prateleira de acrílico para componentes eletrônicos (Fonte: Elaborado pelo autor)	43
Figura 21 – Motor DC 6V com redução + roda (Fonte: https://www.baudaeletronica.com.br/)	44
Figura 22 - Regulador de tensão (Fonte: https://alexnlid.com/)	45
Figura 23 - Placa de desenvolvimento <i>Wi-Fi e Bluetooth</i> ESP32 (Fonte: Elaborada pelo autor).....	47
Figura 24 - Composição de um motor de corrente contínua (Fonte: https://www.citisystems.com.br)	48
Figura 25 – Motor de corrente contínua (Fonte: https://www.citisystems.com.br)	49
Figura 26- Comportamento da corrente em um motor CC (Fonte: https://www.citisystems.com.br).....	49
Figura 27 - Representação do campo magnético em um motor CC (Fonte: https://www.citisystems.com.br)	50
Figura 28 – Torque em um motor CC (Fonte: https://www.citisystems.com.br)	50
Figura 29 - Regra da mão esquerda (Fonte: https://www.citisystems.com.br).....	51
Figura 30 - Célula Li-ion (Fonte: https://www.embarcados.com.br)	52
Figura 31 - Partes internas do servo motor (Fonte: https://www.feis.unesp.br)	56
Figura 32 - Sinal de controle do servo (Fonte: https://miro.medium.com).....	57
Figura 33 - Ângulos de movimentação do servomotor (Fonte: http://repositorio.aee.edu.br).....	57
Figura 34 - Servo motor submetido a um torque de 1 kg-cm (Fonte: https://eletroeletronicaetec.wordpress.com)	58
Figura 35 - Tipos de servos motores (Fonte: https://cdn.shopify.com)	59
Figura 36 - Imagem do servo <i>TowerPro</i> MG996R (Fonte: https://www.botnroll.com).....	59
Figura 37 - Sensor óptico por difusão (Fonte: Catálogo de sensores industriais da WEG)	62

Figura 38 - <i>Layout</i> no aplicativo APP Inventor (Fonte: Elaborada pelo autor)	63
Figura 39 - <i>Layout</i> no aplicativo APP Inventor (Fonte: Elaborada pelo autor)	65
Figura 40- <i>Layout</i> do aplicativo (Fonte: Elaborada pelo autor).....	66
Figura 41 – <i>Layout</i> do aplicativo no <i>smartphone</i> (Fonte: Elaborada pelo autor)	66
Figura 42 - Programação em bloco do APP Inventor (Fonte: Elaborada pelo autor)	67
Figura 43 - Tela do <i>firebase</i> (Fonte: Elaborada pelo autor).....	68
Figura 44 - Circuito de movimentação da célula robótica (Fonte: Elaborado pelo autor)	70
Figura 45 – Disposição das células de lítio (Fonte: https://www. produto.mercadolivre.com)	72
Figura 46 - Célula robótica: Ângulo 1 (Fonte: Elaborado pelo autor).....	73
Figura 47 - Célula robótica: Ângulo 2 (Fonte: Elaborado pelo autor).....	73
Figura 48 – Motor DC com caixa de redução (Fonte: https://www.baudaeletronica.com.br)	74
Figura 49 - Dimensões do braço robótico (Fonte: Elaborado pelo autor)	75
Figura 50 - Espaço de trabalho (Fonte: Toppingmall).....	76
Figura 51 - Forças aplicadas ao braço (Fonte: Elaborado por autor).....	77
Figura 52 - Plano XYZ – ângulo θ_1 (Fonte: Elaborado pelo autor)	78
Figura 53 - Plano YZ – ângulo θ_3 (Fonte: Elaborado pelo autor)	79
Figura 54 - Plano YZ – ângulo θ_2 (Fonte: Elaborado pelo autor)	81
Figura 55 - Plano YZ – ângulo θ_4 (Fonte: Elaborado pelo autor)	82
Figura 56 - Gráfico de previsão dos movimentos do braço robótico (Fonte: Elaborado pelo autor).....	84
Figura 57 – Programação usando a biblioteca <i>Wire.h</i> (Fonte: Elaborado pelo autor)	87

Figura 58 – Programação usando a biblioteca <i><analogWrite.h></i> (Fonte: Elaborado pelo autor).....	87
Figura 59 - Esquema elétrico das ligações entre a esp32 e todos os periféricos para funcionamento do projeto (Fonte: Elaborada pelo autor com o <i>software Fritzing</i>).....	88
Figura 60 - Módulo Seguidor de Linha TCRT5000 - 5 Canais (Fonte: https://www.baudaeletronica.com.br/) ...	89
Figura 61 - Placa L298N (Fonte: https://uploads.filipeflop.com/)	89
Figura 62 - Driver PWM/Servo 16 canais 12-bit com <i>interface</i> I2C (Fonte: https://produto.mercadolivre.com.br/)	90
Figura 63 - Função para parametrizar os servos (Fonte: https://diyi0t.com)	94
Figura 64 - Posições dos servos (Fonte: Elaborada pelo autor).....	95
Figura 65 - Placa desenvolvida para regular os ângulos dos servos (Fonte: Elaborada pelo autor)	96
Figura 66 - Esquema elétrico da placa de circuito impresso e suas ligações (Fonte: Elaborada pelo autor com ajuda do <i>software fritzing</i>).....	96
Figura 67 - Montagem do circuito elétrico na célula robótica (Fonte: Elaborado pelo autor).....	97
Figura 68 - Célula robótica: Garra (Fonte: Elaborado pelo autor).....	97
Figura 69 - Circuito (Fonte: Elaborado pelo autor)	99
Figura 70 - Prateleira (Fonte: Elaborado pelo autor).....	99
Figura 71 - Produto (Fonte: Elaborado pelo autor)	99
Figura 72 - Célula no ponto A do circuito (Fonte: Elaborado pelo autor).....	100
Figura 73 - Google <i>Firebase</i> (Fonte: Elaborada pelo autor).....	102
Figura 74 – Tela inicial do aplicativo (Fonte: Elaborado pelo autor).....	102
Figura 75 - Tela de login (Fonte: Elaborado pelo autor)	103
Figura 76 - Tela de cadastro (Fonte: Elaborado pelo autor)	103

Figura 77 – Tela: Lista dos produtos (Fonte: Elaborado pelo autor).....	104
Figura 78 – Tela: Lista dos produtos selecionados (Fonte: Elaborado pelo autor).....	105
Figura 79 – Tela pedido finalizado (Fonte: Elaborado pelo autor).....	106
Figura 80 – Tela: <i>Check in</i> (Fonte: Elaborado pelo autor)	107
Figura 81 - Tela: Pedido finalizado (Fonte: Elaborado pelo autor)	108

LISTA DE TABELAS

Tabela 1– Lista de materiais (Fonte: Elaborado pelo autor).....	34
Tabela 2 – Componentes do Braço Robótico (Fonte: Elaborado pelo autor)	35
Tabela 3 – Especificação do modelo usado do braço robótico (Toppingmall, 2021).....	35
Tabela 4 – Componentes do carrinho da célula (Fonte: Elaborado pelo autor).....	39
Tabela 5 – Componentes do carrinho da célula (Fonte: Elaborado pelo autor).....	41
Tabela 6 – Componentes do braço robótico (Fonte: https://www.baudaeletronica.com.br/).....	43
Tabela 7 – Regulador de tensão (Fonte: https://alexnl.com/)	44
Tabela 8 – Microcontrolador: ESP32 (Fonte: Elaborado pelo autor)	46
Tabela 9 – Motor (Fonte: Elaborado pelo autor)	48
Tabela 10 – Bateria (Fonte: https://www.produto.mercadolivre.com)	52
Tabela 11 – Servos motor (Fonte: ReadyTosky).....	54
Tabela 12– Servos motor (Fonte: https://www.robocore.net/)	55
Tabela 13 – Sensores (Fonte: Elaborado pelo autor).....	61
Tabela 14 – Circuito de movimentação da célula robótica (Fonte: Elaborado pelo autor).....	69
Tabela 15 - Consumo do sistema (Elaborado pelo autor).....	71
Tabela 16 – Relação de transmissão e perímetro da roda (Elaborado pelo autor).....	74
Tabela 17 – Dimensões (Fonte: Elaborado pelo autor)	75
Tabela 18 – Massa (Fonte: Elaborado pelo autor).....	77
Tabela 19 – Gráfico de previsão dos movimentos do braço robótico (Fonte: Elaborado pelo autor).....	83

Tabela 20 - Gráfico de previsão dos movimentos do braço robótico: Ângulos (Fonte: Elaborado pelo autor)..... 83

SUMÁRIO

1. Introdução.....	17
1.1. Justificativa.....	19
1.2. Objetivos	20
2. Revisão bibliográfica.....	21
2.1. Braço robótico.....	21
2.1.1. Estrutura do braço.....	22
2.1.2. Modelos.....	23
2.2. Célula robótica móvel	25
2.3. AGV (<i>Automated Guided Vehicle</i>)	26
2.4. <i>Fast food</i> robotizada	28
2.5. Mercado 4.0	30
2.6. Aplicativo.....	32
3. Metodologia.....	33
3.1. Célula robótica móvel	33
3.1.1. Modelo do braço	33
3.1.2. Lista de equipamentos	34
3.2. Braço robótico.....	35
3.2.1. Peça para fixar os servos	36
3.2.2. Modelo do servo motor	40
3.3. Carrinho da célula	41
3.3.1. Peça para fixar o carrinho móvel.....	42
3.3.2. Motor DC.....	43
3.3.3. Regulador de tensão	44
3.4. Componentes da célula robótica móvel	45
3.4.1. Microcontrolador	45
3.4.2. Motor DC.....	47
3.4.3. Baterias de Lítio-ion	51
3.4.4. Servo motores	54
3.4.5. Comunicação I2C.....	60
3.4.6. Porta PWM	60
3.4.7. Sensores	61

3.5.	Estrutura lógica da célula robótica.....	62
3.5.1.	Linguagem C++	62
3.6.	Aplicativo.....	64
3.6.1.	Linguagem de programação visual de bloco	64
3.6.2.	MIT App Inventor	65
3.6.3.	Google Firebase.....	68
3.7.	Circuito de movimentação da célula robótica	69
4.	Resultados e discussão	71
4.1.	Cálculo da corrente, tensão e potência usada no circuito	71
4.1.1.	Consumo do circuito de controle	71
4.1.2.	Cálculo da fonte de energia	72
4.1.3.	Cálculo dos motores utilizados.....	72
4.2.	Cálculo dos motores utilizados, torque, rotação, relação de transmissão	75
4.2.1.	Cinemática do braço robótico	75
4.2.2.	Cálculo de Torque do Braço	77
4.2.3.	Cinemática Inversa	78
4.2.4.	Previsão Comportamental do Braço Robótico	83
4.3.	Bibliotecas utilizadas na programação.....	85
4.3.1.	<WiFi.h>.....	85
4.3.2.	<Adafruit_PWMServoDriver.h>.....	85
4.3.3.	<IOXhop_FirebaseESP32.h>.....	85
4.3.4.	<Wire.h>	86
4.3.5.	<analogWrite.h>.....	87
4.4.	Circuito elétrico	88
4.5.	Programação da esp32.....	90
4.6.	Regulagem dos ângulos dos servos, comprimento de onda mínimo e máximo do PWM....	94
4.7.	Placa para teste das posições dos servos, com potenciômetro.....	95
4.8.	Circuito que a célula robótica irá fazer	98
4.9.	Desenvolvimento do Aplicativo	101
5.	Conclusões.....	109
	Referências	110

1. Introdução

Segundo Paletta e Marin (2014), às primeiras ideias da criação de um supermercado em formato de autoatendimento, onde o cliente entrava no estabelecimento e pegava os produtos que deseja direto das prateleiras surgiram em 1912, porém apenas em 1930 a ideia trazida por Michael Kuller se tornava realidade.

Em Long Island, New York se inaugurou a primeira loja de autoatendimento no mundo e conseguia o sucesso para o começo do autoatendimento nos supermercados.

Ainda por Paletta e Marin (2014), no Brasil esse feito apenas se iniciou em 1953 com o supermercado “Sirva-se”, construído na Rua consolação, São Paulo. Adotando o mesmo estilo americano de autoatendimento, outros supermercados em diferentes regiões do país foram surgindo ao longo dos anos, porém esse sistema apenas começou a crescer definitivamente na década de 60, onde segundo a Fundação ABRAS (2002), em 1966 já existiam 997 supermercados espalhados por todo o país e em 1970 já existiam 2527.

Albuquerque (2007), diz que a partir do final da década de 80, os mercados necessitavam de uma nova maneira para gerenciar os mercados, utilizando para tal ferramentas de alta tecnologia e inovação, para conseguir assim, obter preços mais baixos para seus clientes, um melhor gerenciamento de estoques e sistemas de logística para a aquisição e distribuição de seus produtos.

De acordo com Chiavenato (2000), após a revolução industrial, os sistemas de mercados melhoraram a forma de comercializar os produtos, investindo nas áreas de compras e estoques, alterando fortemente os sistemas de estocagem, distribuição e venda em grandes proporções para atender às grandes demandas.

Com a chegada da indústria 4.0 ou revolução industrial, em 2012 surge a transformação digital em um cenário de constantes mudanças tecnológicas, com a execução de produtos conectados, clientes conectados, cadeia de suprimentos inteligente e operações inteligentes impulsionada por uma vasta rede de sistemas *ciberfísicos*.

Entre às tecnologias aplicadas em redes de distribuição de alimento a mais usada é a inteligência artificial (IA), que nada mais é que um ramo da ciência da computação que estuda e desenvolve máquinas e softwares inteligentes. No varejo, está presente através de tecnologia como, *Chef Bot*, uma ferramenta de receitas que oferece sugestões de preparo a partir dos

produtos que o cliente tem em sua geladeira, um carrinho de compras que usa IA para medir, pesar e contabilizar cada item colocado dentro dele, *machine learning* e modelos preditivos para desenvolver listas de produtos personalizadas para os clientes, aplicativos e *web site* para realizar compras e o sistema de gerenciamento robótico que usa IA para resolver problemas de estoque, fazer escaneamento de prateleiras, limpezas de pisos e contagem de inventário utilizando câmeras com sensor a laser.

O gerenciamento de sistema robótico funciona através de microcontrolador e envolve diversas áreas do conhecimento, tais como Engenharia de Controle e Automação, Engenharia Elétrica e Eletrônica, Engenharia Mecânica, Computação, entre outras. Várias ferramentas e tecnologias foram propostas e estão sendo utilizadas para se ter robôs mais autônomos e eficazes.

Atualmente a empresa KUKA desenvolvedora de produtos e serviços para a indústria 4.0, desenvolveu um robô que é utilizado em diversas áreas, como, centros logísticos, onde flexibilidade, precisão, rapidez e adaptação são fatores fundamentais, unindo um braço robótico de alta eficiência a uma plataforma móvel e autônoma. O intuito é substituir os tradicionais AGV (do português: veículo guiado autônomo) e as esteiras transportadoras, focando na etapa de carregamento, locomoção e entrega de produtos na linha de produção.

Com a chegada da pandemia em 2020, o mundo acabou enfrentando diversas crises financeiras e sanitárias, um dos locais mais afetado por esse vírus, foram os supermercados que seguem sendo um dos locais mais propícios à contaminação por Covid-19, além de outros vírus e bactérias, segundo estudo publicado em setembro de 2020 pela Associação Médica do Texas (TMA, na sigla em inglês).

Foi pensando nisso, que muitas empresas começaram a buscar maneira de otimizar o comércio, e melhorar o armazenamento e controle de estoque, um exemplo disso é a empresa Walmart, que durante a pandemia adotou robôs de escaneamento de prateleiras e outros sistemas automatizados para rastreamento de estoque, a empresa colocou as máquinas para trabalhar em 1000 lojas da franquia, o resultado dessa otimização foi divulgada na convenção da *National Retail Federation* (Federação Nacional de Varejo) em 2020, onde o Sarjoun Skaff fundador e CEO da empresa Bossa Nova, empresa que desenvolveu os robôs, diz que foi um sucesso, e que a empresa conseguiu solucionar os problemas de estoque que não era possível resolver a anos.

Outro método usado para diminuir o contato entre as pessoas, foi a compra virtual e retirada no estabelecimento, onde o cliente faz a compra pelo aplicativo ou *website* e retira o pedido pelo *drive-thru*, porém esse método não estava sendo eficiente já que ainda existia contato humano e exposição ao vírus, tanto para o cliente como para o funcionário do estabelecimento.

Muitas empresas e pesquisadores começaram a desenvolver robôs para ser aplicado nesses locais, porém uma das linhas mais importantes na área da robótica no setor varejista é o posicionamento do braço robótico para execução de determinadas tarefas e a movimentação autônoma. Essas tarefas devem ser executadas em um tempo determinado e com uma trajetória muito precisa. É por isso que se deseja do robô um comportamento inteligente durante a execução das tarefas.

Um exemplo deste tipo de problema seria a locomoção e retirada dos produtos das prateleiras. Para isso, pode-se utilizar linguagem C++ para realizar a programação e microcontroladores como ESP32 para controlar o braço robótico e uma placa que utiliza a comunicação I2C para facilitar no controle dos eixos da célula, além de portas PWM para fazer o controle dos servos para a movimentação do robô.

Sendo assim, neste trabalho foi feito a montagem da célula robótica e o desenvolvimento do aplicativo e a estrutura lógica de código que permite controlar a célula robótica com mais eficiência e precisão.

1.1. Justificativa

Durante a pandemia do Covid-19, as indústrias varejistas sofreram muito com as novas medidas de segurança para evitar a propagação do vírus, um exemplo disso, são os supermercados, apesar de ser um setor essencial, existe uma grande manipulação humana, desde o estoque até a entrega final ao cliente, mesmo diminuindo esse contato com diversas restrições e aderindo ao uso de *delivery* e *drive-thru*, os supermercados seguem sendo um dos locais mais propícios à contaminação por Covid-19, além de outros vírus e bactérias, segundo estudo publicado em Setembro de 2020 pela Associação Médica do Texas (TMA, na sigla em inglês). Com o avanço da automação no cotidiano da população, onde você pode escolher os produtos que deseja direto da sua residência, aplicativos como o ifood que disponibiliza vários restaurantes e mercados em um só lugar, é possível observar que esses tipos de tecnologia farão

cada vez mais parte da rotina do ser humano. Pensando no cenário atual e na implementação de tecnologias atuais de indústria 4.0 nesse setor, foi decidido criar célula robótica móvel, que poderá ser implementada em um mercado totalmente autônomo, realizando a separação, empacotamento, autenticação e entrega ao cliente sem nenhum contato que possa disseminar o vírus. Dessa forma, esse projeto irá trazer segurança e praticidade para os clientes, e, diminuir gastos a longo prazo das empresas com sistemas de gestão, transporte, separação e entrega dos produtos para o cliente.

1.2. Objetivos

O objetivo do trabalho é desenvolver uma célula robótica móvel, que irá receber um pedido através de um aplicativo, irá coletar os itens, depositando em uma cesta e entregar para o cliente, através de uma comporta giratória quando o mesmo chegar no local. O projeto será pensado para ser implementado em mercados totalmente automatizados e que funcione com um sistema *drive-thru*, a fim de diminuir o risco por contaminação e aumentar a agilidade e praticidade nas compras efetuadas pelos clientes. Para cumprir o objetivo geral, foram definidos os objetivos específicos descritos a seguir:

1. Identificar os parâmetros que caracterizam a solicitação do pedido de um cliente e desenvolver a estrutura lógica de código que permita controlar uma célula robótica;
2. Estabelecer a comunicação via *wi-fi* entre o aplicativo desenvolvido e a célula robótica;
3. Prototipar a estrutura que caracteriza a célula robótica;

2. Revisão bibliográfica

O crescimento da automação nos últimos anos é um fator cada vez mais presente nos dias atuais, um estudo realizado pela GSI Brasil - Associação Brasileira de Automação juntamente com a empresa de pesquisa Gfk, que realiza todo ano uma pesquisa referente ao crescimento da automação em três principais setores, sendo, consumidores, indústria, comércio e serviços, mostrou que desde 2017 até 2020 a automação cresceu no setor de comércio e serviços um total próximo de 11%, mostrando como cada ano que se passa a automação se torna mais presente no nosso cotidiano e na maneira como é realizado a gestão do setor comercial.

Outro estudo realizado pelo IPEA sobre o comércio mundial em 2020, publicado em janeiro de 2021, mostra que o ano de 2020 foi um ano atípico dos outros, devido a pandemia do COVID-19, que afetou drasticamente a economia mundial, mas principalmente as atividades que necessitavam do contato direto entre as pessoas. De acordo com o estudo, a queda foi próxima de 9,2% no setor comercial.

Fazendo uma comparação entre os dois estudos, é possível perceber que mesmo com os efeitos da pandemia afetando a economia no setor comercial, o avanço de novas tecnologias continuou crescendo, mostrando como a implementação de novas tecnologias nesse setor são de fundamental importância e estão sendo cada vez mais empregadas.

Neste capítulo serão enfatizados os aspectos teóricos dos braços robóticos que foram a base para o desenvolvimento deste projeto, apresentando os tipos de braços e as principais empresas desenvolvedoras.

2.1. Braço robótico

O braço robótico é um dispositivo programável que foi desenvolvido para fazer as mesmas movimentações que um braço humano. O robô é composto por 6 graus de liberdade (GL ou DOF, segundo as iniciais em inglês) sendo três para o posicionamento dentro do espaço de trabalho, e três para obter uma orientação adequada e segura para o objeto.

Segundo artigo “O braço robótico industrial ganha espaço no armazém” publicado pela empresa Mecalux, em março de 2021 “*Com as habilidades do braço robótico é possível desempenhar as mesmas funções que um operador poderia desempenhar, mas com capacidade*

para lidar com cargas mais pesadas, com maior velocidade e sem cansaço que a repetição de movimentos implica para um ser humano”.

2.1.1. Estrutura do braço

Assim, como um braço humano, o braço robótico é formado por juntas, responsável pela movimentação do braço, base, antebraço, punho e o órgão terminal que é uma parte do braço responsável por realizar a tarefa desejada, ex: braço com garras, dispositivos de sucção, ponteiros para soldagem etc.

O movimento do robô é dotado de 3 acionadores e uma configuração 3GL. Três servo motores serão montados sobre sua estrutura para efetuar os movimentos desejados. Eles são articuladamente ligados e podem ser selecionadas suas extensões e retrações através da utilização do microcontrolador.

Antebraço e punho também são compostos de 3 graus de liberdade responsáveis por movimentos de maior precisão na tarefa.

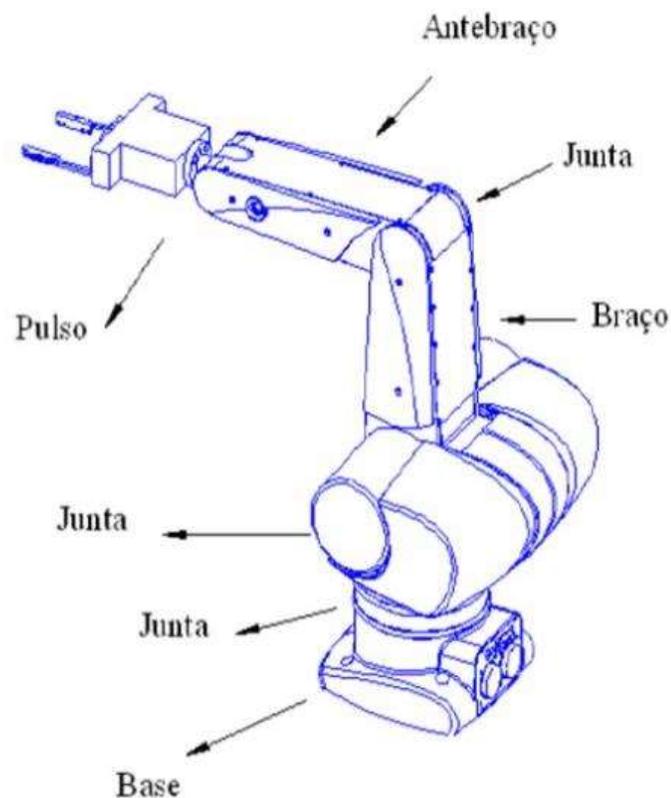


Figura 1 - Partes de um braço robótico (Fonte: <https://slideplayer.com.br>)

2.1.2. Modelos

O artigo, “*Os seis principais tipos de robôs industriais*” publicado pela empresa de automação indústria Citisystems, em 2019 mostrou os seis principais braços robóticos usado no mercado, sendo eles: **Cartesiano, Scara, Cilíndrico, Delta, Polar e Verticalmente Articulado.**

Segundo o autor do artigo Cristiano Bertulucci Silveira, os robôs mais usados no ramo alimentício são:

Cartesiano

Robô usados em operações de escolher e colocar, carga e descarga, montagem e manipulações de materiais nucleares, o robô possui três juntas prismáticas com o objetivo de realizar movimentação linear deslizantes entre os eixos X, Y e Z. Apesar de fácil programação *offline*, menor custo e operação simples, ele possui movimento limitado e montagem complexa.



Figura 2 - Tipos de robôs cartesianos (Fonte: <https://www.citisystems.com.br/>)

Scara

Esse modelo de robô é usado em carregamento de máquinas, embalagem, paletização, montagem de circuitos etc. Ele é formado por duas juntas paralelas tem alta velocidade e fácil programação, porém sua desvantagem é que ele é limitado a superfícies planas.



Figura 3– Robô SCARA (Fonte: <https://www.citisystems.com.br>)

Verticalmente articulado

Esse é o modelo de robô mais usado na indústria, sendo aplicado em manuseio de materiais, máquinas, montagem automotiva, corte de aço, paletização, embalagem de alimentos, soldagem etc. O número de juntas pode variar de duas a dez juntas dependendo da marca do fabricante e o grau de liberdade. Apesar de possuir alta velocidade, melhor alinhamento de coordenadas, ele requer maior controle e programação mais complicada.



Figura 4 – Tipos de robôs articulados (Fonte: <https://www.citisystems.com.br>)

2.2. Célula robótica móvel

Os robôs estão presentes atualmente atendendo às necessidades dos clientes, eles são capazes de executar tarefas que podem representar perigo às pessoas, reduzem os custos, trabalham de forma contínua e melhoram o desempenho das empresas.

Pensando nisso, a empresa KUKA desenvolvedora de produtos e serviços para a indústria 4.0, desenvolveu um robô chamado KMR *iiwa*, figura 5. Esse robô é utilizado em diversas áreas, como, centros logísticos, onde flexibilidade, precisão, rapidez e adaptação são fatores fundamentais, unindo um braço robótico de alta eficiência a uma plataforma móvel e autônoma. O intuito é substituir os tradicionais AGV (do português: veículo guiado autônomo) e as esteiras transportadoras, focando na etapa de carregamento, locomoção e entrega de produtos na linha de produção.



Figura 5 - Robô KMR *iiwa* da KUKA (Fonte: <https://face-aluminium.com>)

De acordo com o site da empresa KUKA, as vantagens do robô móvel KMR *iiwa* podem ser associadas as seguintes características:

- **Autônomo:** A plataforma móvel tem capacidade de navegar de forma totalmente autônoma graças aos scanners a laser. O monitoramento do ambiente é contínuo e se uma pessoa ou um objeto estiver no trajeto de deslocamento, ela reage imediatamente.
- **Sensitivo:** Possui sete sensores especiais de torque localizados nas articulações em cada eixo do robô. Isto o torna ainda mais sensível para o seu ambiente. Ele navega com segurança e sem proteção já que conta com dispositivos instalados na sua plataforma capazes de detectar contato externo com pessoas e objetos.
- **Preciso:** O robô KMR *iiwa* consegue precisão de posicionamento de até +/- 5 milímetros, mesmo em espaço limitados.
- **Móvel:** Especialmente desenvolvidas as rodas *Mecanum* possibilitam que a plataforma móvel efetue movimentos omnidirecionais e rotações em 360 graus o que gera movimentos precisos e controlados. A roda é construída com vários rolos, dispostos respectivamente em ângulo de 45 graus em relação ao eixo. Essa mobilidade máxima reduz os tempos de percurso e diminui os tempos inativos no processo de fabricação.
- **Inteligente:** Com ajuda do KUKA *Navigation Solution* o KMR *iiwa* consegue modificar sua trajetória e procura se esquivar com segurança de qualquer obstáculo.
- **Amigável ao usuário:** Conta com uma unidade chamada KUKA *Sunrise Cabinet* capaz de comandar o veículo e o robô, simplifica para você a operação e o uso do KMR *iiwa* na prática.
- **Independente:** A plataforma e o robô são alimentados diretamente com energia através de baterias Li-Ion.

2.3. AGV (*Automated Guided Vehicle*)

Os AGVs (*Automated Guided Vehicle*) ou veículos guiados automaticamente, são veículos cada vez mais utilizados na indústria, principalmente empregados em sistemas logísticos, tem o objetivo de transportar cargas ao longo da fábrica de maneira autônoma, como mostrado na figura 6.



Figura 6- Tipos de sistemas de AGVs (Fonte: <https://www.arpapress.com>)

De acordo com um relatório de pesquisa de mercado, elaborado pela empresa *Markets and Markets*, publicado em julho de 2020, diz que os investimentos no mercado de AGVs pretende crescer cerca de 9,3% até 2025, com um investimento de 3,6 bilhões de dólares nesse mercado, visando a integração com a indústria 4.0 e a grande demanda pelo transporte de materiais pela indústria com segurança e personalização da produção.

Segundo YAGHOUBI os AGVs podem ser divididos em três principais tipos de orientação para a sua implementação, sendo elas:

- **Orientação por laser:** Permite uma maior flexibilidade para implementações de rotas complexas, já que não necessita de nenhuma indicação para realizar a rota, apenas sendo programado via software.
- **Orientação por fio:** Permite uma troca de informações por meio de fios, rádio ou por um computador principal, onde o veículo é guiado por um fio inserido no chão.
- **Orientação por fita (faixa):** É realizado a troca de informações por meio de uma câmera que envia as informações via rádio, o veículo segue a trajetória através de uma fita ou uma linha pintada no chão.

Além dos tipos de orientações, os AGVs ainda são divididos por tipos de operação, levando a sua necessidade e capacidade de transportar o objeto pelo caminho estipulado, os seus principais tipos também são definidos por YAGHOUBI como, veículo de carga unitária, veículos de carrinho, veículo de empilhadeira, veículo de paletes, veículos de carga leve e veículos de reboque, variando principalmente pelo seu tipo de carga, rota a realizar e complexidade do percurso.

Visando o desenvolvimento de um protótipo para a coleta de produtos e o transporte ao longo do seu percurso até o cliente final, será utilizado um sistema de orientação por fita com um tipo de operação para um veículo de carrinho, porém como será utilizado um braço robótico para realizar a função de pegar os objetos das prateleiras, ele funcionará de uma maneira um pouco diferente de um AGV convencional.

2.4. Fast food robotizada

Pensando no atual cenário causado pela pandemia da covid-19, inúmeras empresas estão buscando maneiras de segurança para evitar a propagação do vírus. Um exemplo disso é a rede McDonald's, maior franquia de fast food do mundo, segundo ranking divulgado pela Entrepreneur (2020), atualmente o restaurante está presente em 119 países e com 37 mil pontos de venda (Fonte: Arcosdourados.com).

Mesmo com essa magnitude, em 2020 o McDonald's teve uma queda de 68% nas vendas globais, analistas previam uma queda de 23,24%, segundo dados da Refinitiv. Para ajudar a alavancar o lucro da empresa, eles colocaram 96% dos restaurantes para operar por meio de *delivery* e *drive-thru*.

"Nossa forte presença no *drive-thru* e os investimentos que fizemos nas áreas digital e de entregas nos últimos anos nos serviram bem nesses tempos de incerteza", disse o presidente-executivo do McDonald's, Chris Kempczinski em entrevista para o site Exame em julho de 2020.

Essa situação também afetou a startup brasileira Bionicook, um fast food que funcionava dentro da Universidade de Caxias do Sul (UCS) no Rio Grande do Sul/ Brasil, porém em 2019 com os fechamentos das universidades por conta da pandemia, a startup teve que encerrar seu trabalho no local.

Todavia, o fundador e CEO da Bionicook, Fabio Rezler percebeu que existia uma procura por estabelecimentos que não necessitavam de contato físico entre pessoas, pensando nisso, ele desenvolveu o primeiro fast food 100% robotizado do mundo. Os aeroportos são locais de grande circulação de pessoas, e muito das vezes essas pessoas estão sempre com pressa, logo existe uma grande busca por alimentação rápida, sem filas e de boa qualidade. Foi pensando nisso que a Bionicook instalou sua loja em um aeroporto no estado de São Paulo, a loja foi inaugurada em abril de 2021, com esse sucesso e a procura por estabelecimentos assim,

segundo o CEO está previsto a instalação de até 1.300 lojas iguais a figura 2 em vários países até 2026.



Figura 7 - Modelo da lanchonete *fast food* Bionicook (Fonte: <https://exame.com>)

“Nós iniciamos com atendimento tradicional, como todos os *fast foods*, mas no decorrer do tempo percebemos que seríamos apenas mais um no mercado e isso me incomodava. Devíamos fazer algo surpreendente, algo que pudesse trazer alguma experiência de consumo incomparável, sem perder o foco naquilo que é essencial, um lanche rápido e de alta qualidade sempre. Precisava ser muito atraente ao consumidor e aos investidores”, conta Fabio Rezler em uma entrevista em abril de 2021, para o site Isto é Dinheiro, filiada do Portal Terra.

Na loja é possível encontrar 18 opções de lanches e 12 de bebidas. Para efetuar a compra, o consumidor faz seu pedido por meio de uma tela digital *touch screen* e, imediatamente após a confirmação do pagamento, o robô inicia o preparo imediatamente e sem a intervenção humana.

A célula robótica como mostrada na figura 8 foi desenvolvida pela empresa Kuka, porém toda a programação foi feita pelo fundador da empresa.



Figura 8 - Robô separador KUKA da Bionicook (Fonte: <https://www.gazetadopobo.com.br>)

Para realizar a instalação dos equipamentos e configuração do robô foi preciso duas semanas, além dos processos legais junto à Anvisa.

“Os agentes estiveram na nossa loja e viram que nosso sistema é muito confiável e seguro. Foi bacana a reação de susto, porque nunca viram um modelo como esse, instalado no meio do aeroporto, sem água corrente ou exaustão. E já estamos acostumados a isso. Também entenderam como é feita a higienização e que os recursos estão dentro da própria máquina. Só precisamos de energia elétrica”. Diz Fabio Rezler, em uma entrevista para o site Exame em abril de 2021

“Cada unidade tem o custo estimado de 500.000 reais para a instalação. E, segundo as previsões da empresa, o objetivo é inaugurar outras 19 lojas no mercado brasileiro até o fim deste ano. já há negociações para levar a iniciativa a outros países da América do Norte, Europa e Oriente Médio. Com a expansão da rede, a previsão é que o faturamento bruto para 2021 pode variar entre 8 e 12 milhões de reais.” (Fonte: site Exame, publicado em abril de 2021).

2.5. Mercado 4.0

A automação no comercial vem mudando a forma de trabalhar de grandes corporações e até mesmo de microempresas, pois simplifica a operação e mitiga retrabalhos e erros. Hoje em dia o mercado oferece soluções de automação que abrange empresas de todos os tamanhos,

assim as tornando menos dependente do trabalho braçal e conseqüentemente reduzindo os custos de operação, fazendo com elas se tornem mais competitivas.

Um ramo do setor comercial, que é o setor varejista almeja pela automação total, com a adoção de lojas e supermercados autônomos, assim dando total autonomia ao cliente e oferecendo uma experiência personalizada para cada um, também reduzindo os custos de operação, uma vez que não se tem mais o fator humano na hora da venda.

Dando mais um passo, rumo à automação comercial total, nasce a Zaitt, empresa responsável por implantar o primeiro supermercado totalmente autônomo do Brasil, ele foi o projeto piloto implantado em 2018, na cidade de Vitória no Espírito Santo, o funcionamento do sistema consiste em: o cliente baixar o aplicativo da Zaitt em seu smartphone, e nele acessa uma chave que abre a loja através de um leitor ótico posicionado na porta da loja.

Dentro da loja os produtos estão dispostos em categorias, como bebidas, comidas, legumes, carnes e mais alguns *snacks*. Após escolher os produtos, o cliente aproxima o celular para ler o código e registrar sua compra, esse processo deve ser feito para todos os itens que foram pegos, por fim o aplicativo confirma o total e debita o valor do cartão de crédito.

O sistema retira o contato humano nas compras de supermercado, mas depende da honestidade do cliente, pois não há um sistema que valida se os produtos lidos pelo aplicativo são realmente os que serão levados.

Um ano depois a Zaitt fez melhorias em seu sistema e seu conceito de supermercado autônomo, então implantou uma nova unidade na cidade de São Paulo, corrigindo as brechas que o sistema piloto havia deixado.

A unidade funciona 24 horas no bairro do Itaim Bibi e opera em parceria com o Carrefour, responsável pelo suporte logístico e de abastecimento de parte dos produtos. Diferentemente da primeira unidade da Zaitt, em operação no Espírito Santo, que faz uso da leitura do QR *Code* do produto via aplicativo, no mercado de São Paulo o cliente conta uma nova experiência chamada RFID.

Trata-se de um sistema de identificação de produtos automático por meio de sinais de rádio. Por meio dele não é necessário o escaneamento produto por produto, pois faz a leitura da cesta do cliente e identifica todos os produtos, assim não mais dependendo da honestidade do cliente em ler todos os produtos pegos, minimizando as perdas e tornando o modelo viável para

implantação em grande escala, o processo de pagamento e liberação da saída se manteve o mesmo da loja piloto.

2.6. Aplicativo

Com o advento dos smartphones, foi possível uma universalização do uso da internet, sendo possível acessar através de uma pequena tela, uma infinidade de assuntos e temas. De acordo com pesquisa realizada pelo Centro de Tecnologia de Informação Aplicada (FGVcia) da Escola de Administração de Empresas de São Paulo da Fundação Getulio Vargas (FGV EAESP), existe atualmente 234 milhões de celulares inteligentes em uso no Brasil, ou seja, mais de um dispositivo por habitante (FGV, 2020).

Segundo dados divulgados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), no ano de 2019, verificou-se que em 99,5% dos domicílios com acesso aos serviços de internet, o principal equipamento utilizado eram os smartphones (IBGE, 2019).

Diante do panorama apresentado, verifica-se que são diversos os serviços que se aproveitam dessas oportunidades para alavancar seus negócios. Conforme Chiavenato (2008) "O mundo está mudando com uma rapidez incrível. E com intensidade cada vez maior. A mudança sempre existiu na história da humanidade, mas não com o volume, rapidez e impacto com que ocorre hoje." Pode-se compreender que com a ampla concorrência de mercado, se cria uma necessidade constante de aprimoramento, para não correr o risco de ser excluído, portanto, a empresa deve-se manter atualizada com as últimas tendências e tecnologias.

Segundo Takiy (2006) é perceptível a preocupação das empresas com relação à melhoria de seus procedimentos, sejam eles gerenciais, operacionais ou de produção. Dessa forma, tem-se utilizado muito os recursos computacionais para melhoria de seus processos, para garantir um maior controle nas operações comerciais.

O mundo presencia uma mudança na maneira como consome produtos e serviços. Através de sistemas e aplicativos, que conectam pessoas diretamente, milhões de dados são trocados pela internet, promovendo a economia compartilhada, tornando-a mais rápida e de fácil acesso.

Nos dias de hoje, é possível constatar a presença cada vez maior do número de serviços oferecidos por plataformas mobile. Aplicativos como Ifood, Uber, Airbnb, Rappi, entre outros, alteraram a forma como as pessoas consomem. Esta realidade é uma comprovação de como as

tecnologias digitais podem revolucionar os resultados no ambiente empresarial e social. A inovação que estes aplicativos trouxeram demonstra como a inserção da tecnologia altera todo o mercado, que cada vez mais necessita de se manter atualizado às novas tendências para não ser deixado para trás.

De acordo com Camarotto (2009), com as novas tecnologias que vão surgindo, novos mercados emergiram, muito em função do estilo de vida e do padrão social da população. Sendo assim, surge a necessidade de as organizações acompanharem essas mudanças, pois elas transformam o modo que os clientes vivem, priorizam seus gastos e escolhem seus produtos e serviços.

3. Metodologia

3.1. Célula robótica móvel

O protótipo foi desenvolvido para ser colocado em um mercado totalmente automatizado, o intuito é trazer mais rapidez, praticidade e segurança tanto para o cliente como para o dono do estabelecimento.

Para dar vida ao nosso protótipo, escolhemos usar a linguagem C++ como programação e o ESP32 como microcontrolador, essa placa será responsável por fazer toda a comunicação entre o aplicativo, o carrinho, o braço e as prateleiras.

Já o aplicativo foi desenvolvido pelo MIT App Inventor, nessa plataforma é possível fazer a programação visual online e realizar a criação de aplicativos com foco em aparelhos móveis que utilizam o sistema operacional Android, além disso também foi usado a plataforma Google Firebase, que serve como integração e implantação de aplicativos web

3.1.1. Modelo do braço

O modelo usado no projeto é o Verticalmente Articulado de seis eixos da empresa Toppingmall, esse modelo foi escolhido, pois possui alta velocidade de movimentação e melhor flexibilidade. O braço foi adquirido através do site aliexpress, o kit contém: Estrutura do braço

de liga de metal para arduino, seis peças, modelo MG996R, seis servos digitais, modelo TD-8325MG e cabeamento mecânico.

3.1.2. Lista de equipamentos

Tabela 1– Lista de materiais (Fonte: Elaborado pelo autor)

Componentes	Quantidade	Valor
6 Servo motor MG996R + estrutura robô	1	R\$ 353,60
2 Servo TD-8325MG	1	R\$ 133,82
Kit motor DC + roda para Robô	4	R\$ 71,12
Módulo seguidor de linha TCRT5000 - 5 Canais	1	R\$ 27,55
Módulo sensor de obstáculo infravermelho IR	1	R\$ 8,90
Potenciômetro linear de 1K	6	R\$ 13,26
Bornes	2	R\$ 2,76
Regulador de tensão L7805	1	R\$ 1,86
Capacitor eletrolitico 1000uF	1	R\$ 0,84
Capacitor eletrolitico 100uF	2	R\$ 0,28
Barra de pinos 40 vias 15mm 180 graus	2	R\$ 3,44
Barra de soquete header fêmea 40 vias 180 graus	2	R\$ 3,98
Placa padrão tipo ilha 10x5 cm	1	R\$ 5,90
Módulo <i>driver</i> ponte H L298N	1	R\$ 32,80
Módulo I2C para servo motor - PCA9685 - 16 canais	1	R\$ 53,25
Parafusos	60	R\$ 12,00
Porcas	60	R\$ 9,00
Estrutura de aluminio	1	R\$ 20,00
Pilha eletrolítica	6	R\$ 162,00
Bateria Samsung ICR18650-22P	4	R\$ 31,50
Placa ESP32	1	R\$ 42,20
Fios	-	-
Resistor	-	-
Total		R\$ 990,06

3.2. Braço robótico

O material usado na estrutura do braço foi o alumínio, o braço robótico possui seis eixos, seis servo motor, sendo dois servos, modelo TD-8325MG, utilizado nos eixos 2 e 3, e quatro servos, modelo MG996R, utilizado nos eixos 1,4,5 e 6.

Tabela 2 – Componentes do Braço Robótico (Fonte: Elaborado pelo autor)

Componentes	Quantidade
Peça para fixar os servos dos eixos 1, 2, 3 e 4	4
Fixação entre o eixo 2 e 3	2
Servo motor TD-8325MG utilizado nos eixos 2 e 3	2
Servo motor MG996R utilizado nos eixos 1,4,5,6	4
Formato em L para fixação entre os eixos 3 e 4	2
Base para fixação de todos os eixos do braço robótico	2
Garra	1
Disco de alumínio para conectar na engrenagem dos servos com as estruturas de alumínio	1
Placa ESP 32	1
Parafuso	-

Tabela 3 – Especificação do modelo usado do braço robótico (Toppingmall, 2021)

Modelo	MG996R
Volts	4.8 - 7.2V
Carga	100mA
Torque	13Kg.cm
Ângulo Limite	180°
Comprimento do fio	30cm
Material	Engrenagem de metal
Tempo de área morta	5uS

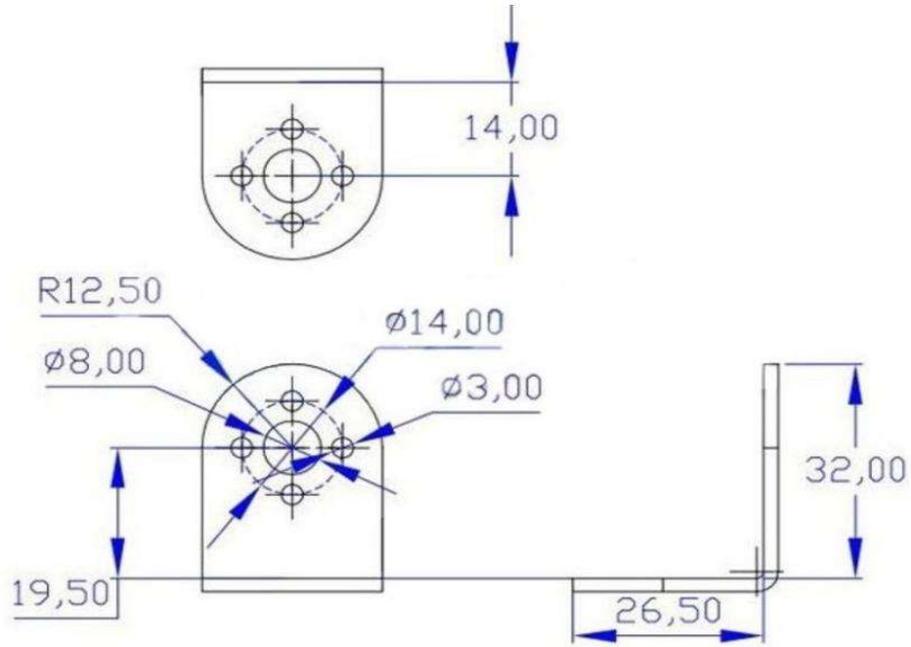


Figura 11 - Formato em L para fixação entre os eixos 3 e 4 (Fonte: Toppingmall)

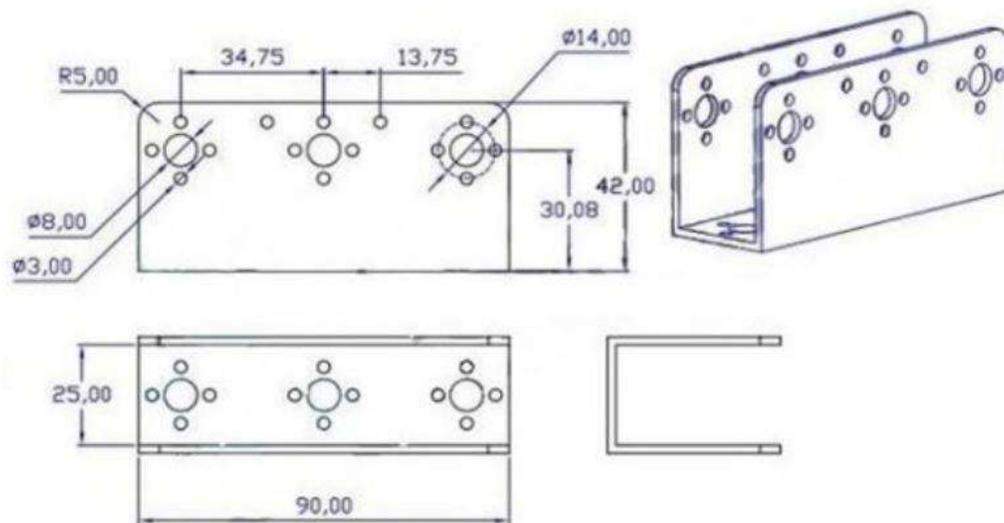


Figura 12 - Base para fixação de todos os eixos do braço robótico (Fonte: Toppingmall)

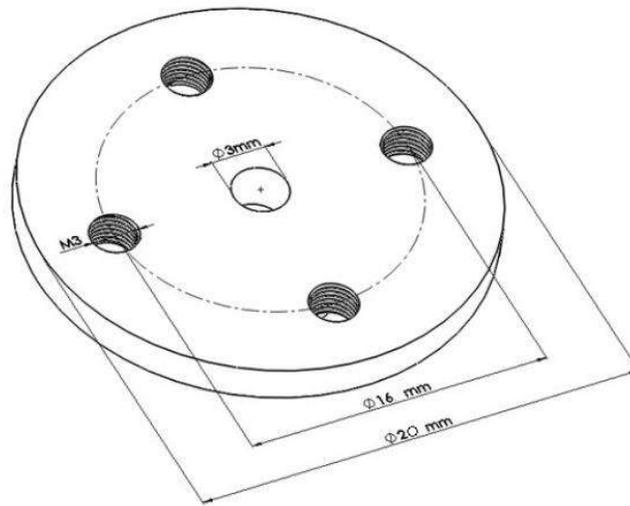


Figura 13 - Disco de alumínio para conectar na engrenagem dos servos com as estruturas de alumínio (Fonte: Toppingmall)

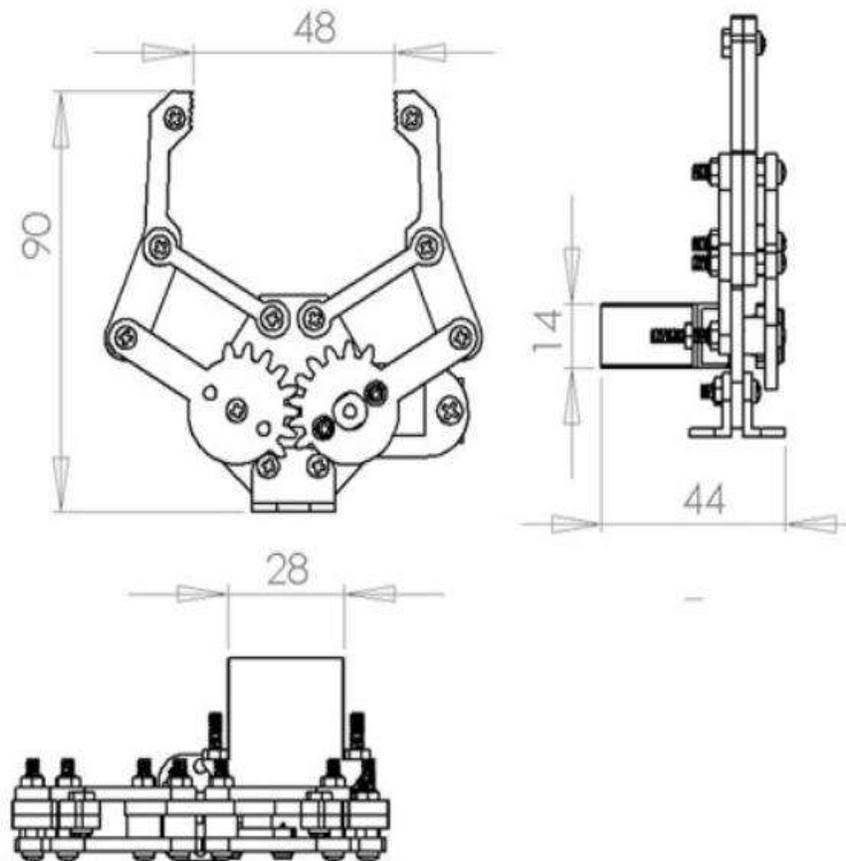


Figura 14 - Garra para pegar as peças, acoplada ao eixo 6 do robô (Fonte: Toppingmall)

Tabela 4 – Componentes do carrinho da célula (Fonte: Elaborado pelo autor)

Dimensão da garra	
Dimensões da garra fechada	11x8cm
Peso	68g

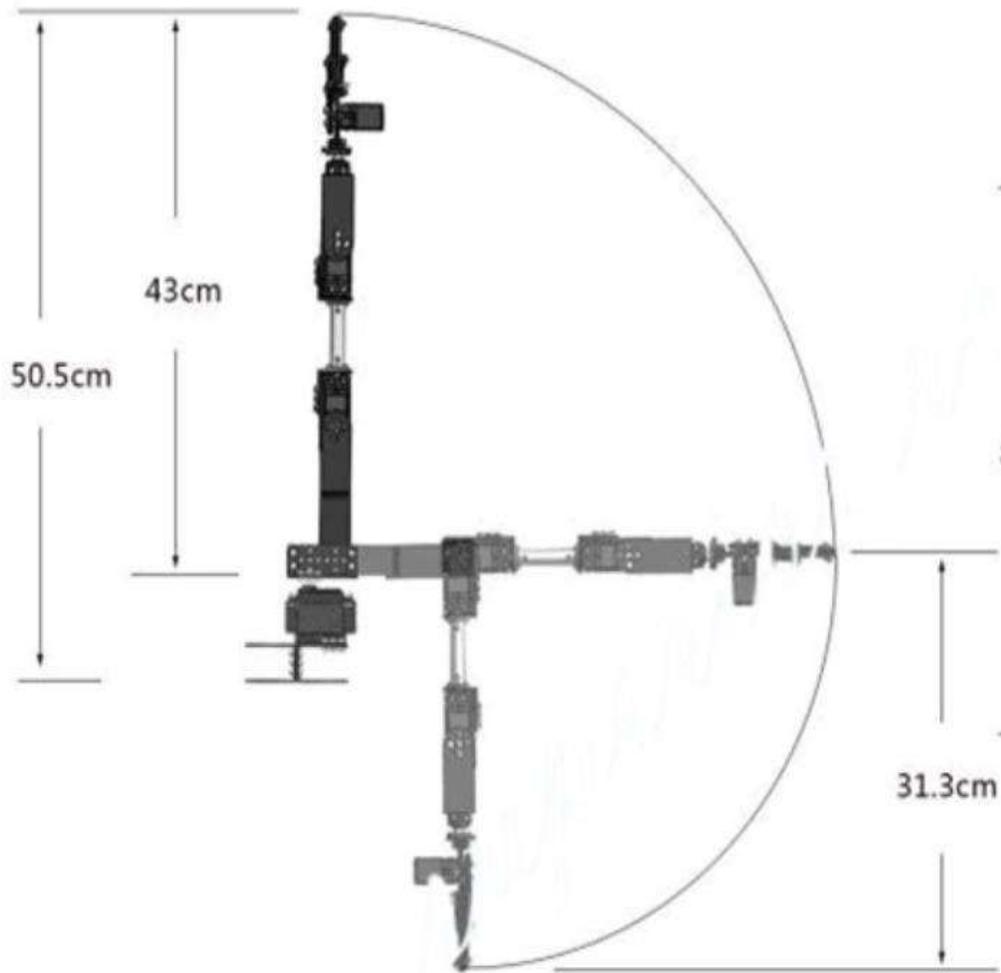


Figura 15 - Abertura máxima do robô (necessário desconsiderar parte do meio que foi removida) (Fonte: Toppingmall)

3.2.2. Modelo do servo motor

MG996R High Torque Metal Gear Dual Ball Bearing Servo

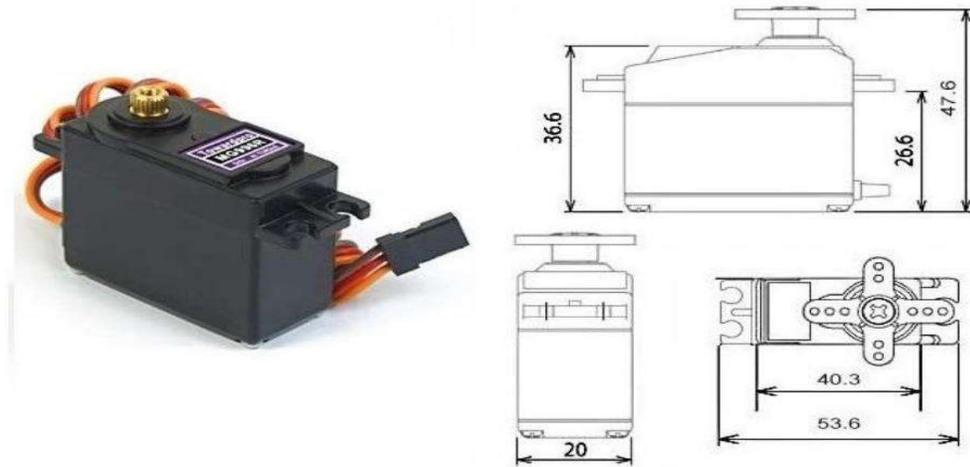


Figura 16 - Servo motor TD-8325MG utilizado nos eixos 2 e 3 (Fonte: Toppingmall)

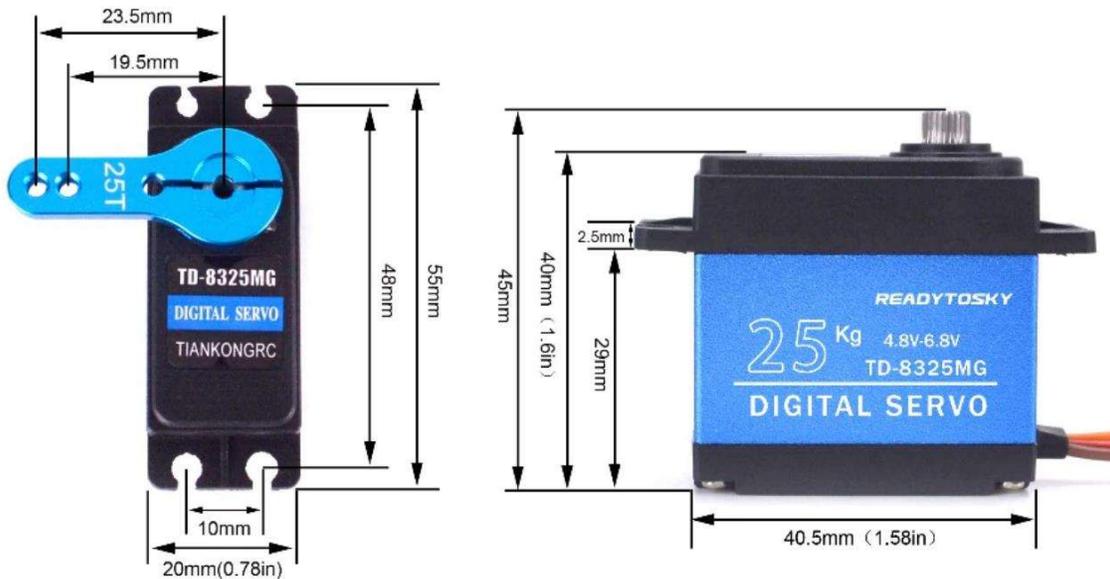


Figura 17 - Servo motor MG996R utilizado nos eixos 1,4,5 e 6 (Fonte: Toppingmall)



Figura 18 – Desenho técnico: Tamanho do servo (Fonte: Toppingmall)

3.3. Carrinho da célula

Tabela 5 – Componentes do carrinho da célula (Fonte: Elaborado pelo autor)

Componentes	Quantidade
Base	1
Rodas	4
Bateria	4
Prateleira	1
Regulador de tensão	1
Placa ESP32	1

O funcionamento do coletor automático de produto é baseado no deslocamento automático para coletar objetos, ditos movimentos são efetuados através de um dos dispositivos, neste caso, o carro móvel. É através de uma faixa principal ou circuito de

prateleiras, sinalizados no chão, que o dispositivo vai se locomover até chegar ao ponto de coleta do objeto indicado. Os diferentes pontos de parada são indicados também no chão por segmentos de faixas adicionais e paralelas à faixa principal. Uma vez posicionado o carro, os movimentos posteriores são efetuados pelo segundo dispositivo, o braço robótico. Os movimentos serão os de rotação se for preciso, os do braço, antebraço, punho e abertura da garra para pegar o objeto. Começando por fechar a garra os movimentos são executados de forma retrativa até depositar o objeto no cesto situado também na parte superior do carro com alta precisão e movimentos corretos.

Com a instalação de uma célula de carga no fundo do cesto o dispositivo enxerga de forma eficiente se tal objeto foi depositado dentro do recipiente. Uma vez confirmado que o objeto desejado foi depositado, o carro continua até sua próxima parada onde efetuará outro movimento para coletar ou se a tarefa foi concluída se dirige até o ponto de entrega previamente designado. Dessa forma o coletor automático de produtos fica a disposição para executar uma nova tarefa quando for indicado.

3.3.1. Peça para fixar o carrinho móvel

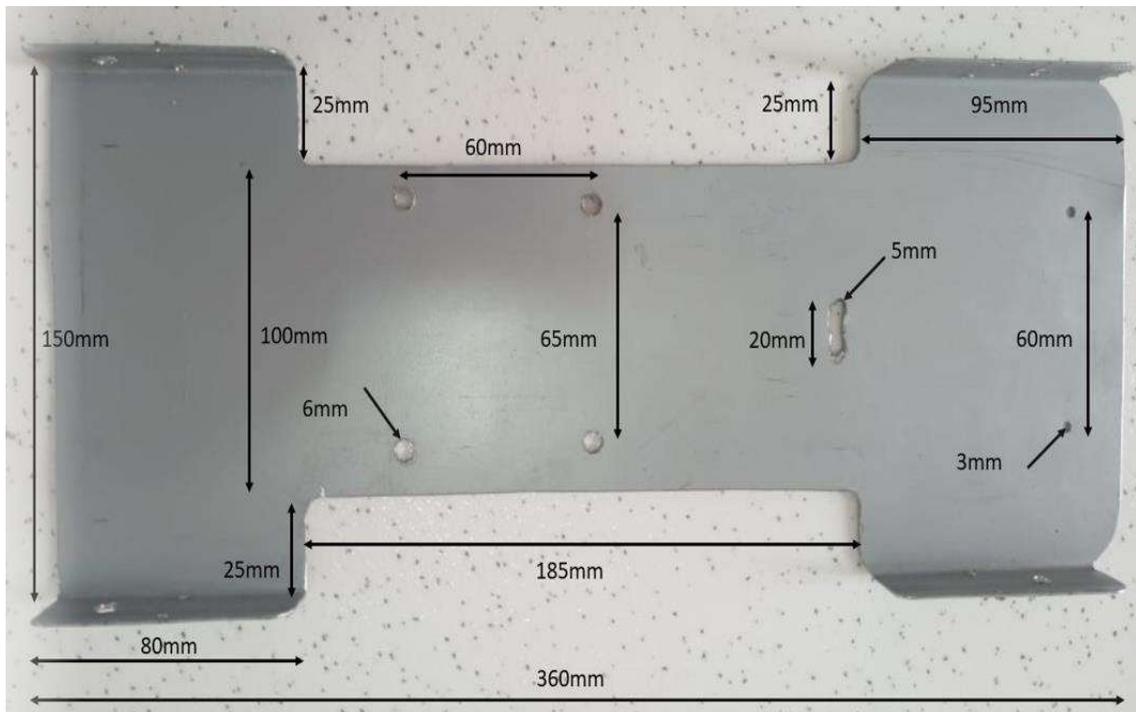


Figura 19 - Base para fixar no carrinho (Fonte: Elaborado pelo autor)



Figura 20 - Prateleira de acrílico para componentes eletrônicos (Fonte: Elaborado pelo autor)

3.3.2. Motor DC

Tabela 6 – Componentes do braço robótico (Fonte: <https://www.baudaeletronica.com.br/>)

Motor DC	6V com redução + roda
Tensão de alimentação dos motores	3 ~ 6VDC
Corrente máxima	120mA
Relação da redução	48:1
Rotação de motor a 6VDC	260 rpm
Velocidade do carro a 6VDC	1 m/s
Dimensões do chassi	22 x 14,7 cm
Dimensões da roda	7 x 7 x 2,6 cm
Perímetro da roda	22cm

Motor DC de 6V com caixa de redução de 48:1 para aumentar o torque do motor na saída para girar a roda.



Figura 21 – Motor DC 6V com redução + roda (Fonte: <https://www.baudaeletronica.com.br/>)

3.3.3. Regulador de tensão

O regulador de tensão, modelo: DC DC_DC XH-M401 é responsável por regular a tensão de entrada em uma tensão estável de saída.

Tabela 7 – Regulador de tensão (Fonte: <https://alexnlid.com/>)

Regulador de tensão DC	DC_DC XH-M401
Modo de regulação	modulação PWM
Tensão de entrada	DC4-40V
Faixa de saída	DC1,25-36V
Corrente máxima	8A
Potência máxima	200W
Eficiência de conversão	94%
Frequência de comutação	180KHz
Tamanho	61 x 41 x 27mm

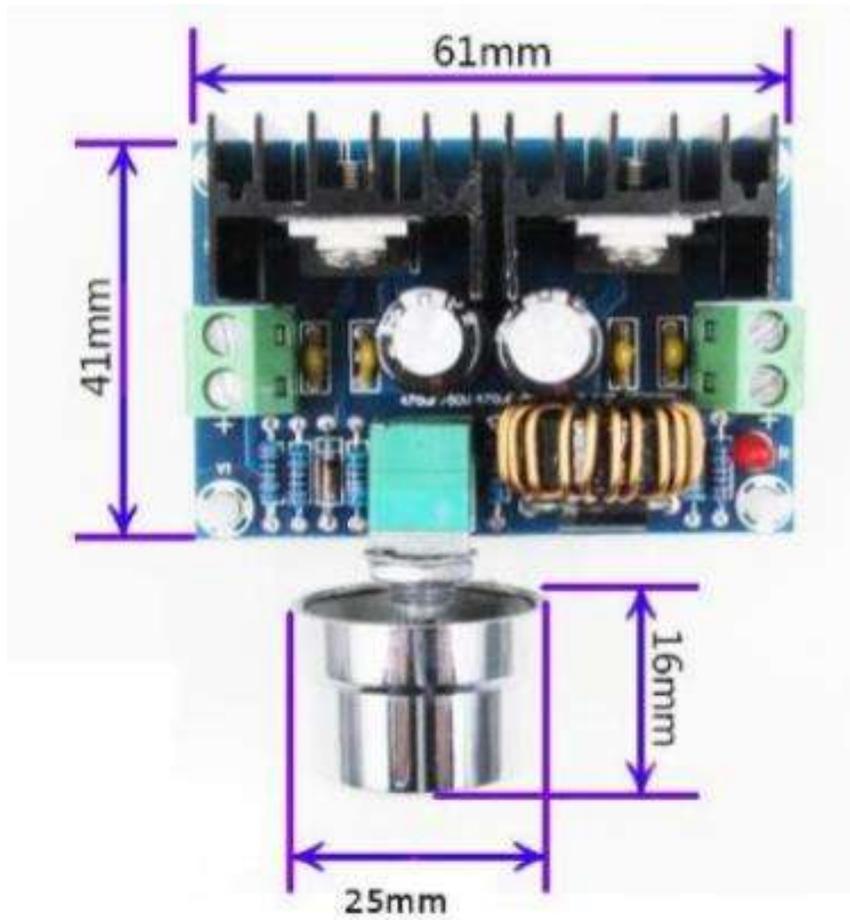


Figura 22 - Regulador de tensão (Fonte: <https://alexnlid.com/>)

3.4. Componentes da célula robótica móvel

3.4.1. Microcontrolador

Um microcontrolador é um componente eletrônico inteligente que possui um processador, pinos de entrada e saída (I/O) e memória. Através dos valores de entrada e com uma programação pré-estabelecida consegue-se controlar os pinos de saída de acordo com a programação. Para realizar a programação, os comandos são feitos em um software que fazem a conversão da linguagem de programação utilizada, para uma linguagem de máquina, dessa forma é possível fornecer informações para realizar as tarefas.

O primeiro microcontrolador foi lançado pela Intel em 1977 e recebeu a sigla “8048”. Por ser um dos percussores, é utilizado em muitas aplicações de automação em diversas áreas. Existem vários tipos de microcontroladores e o que diferencia um do outro é o tamanho da memória, a quantidade de pinos I/O, a velocidade de processamento, os tipos de periféricos e a forma de alimentação deles.

Segundo um artigo publicado no site da Universidade Federal de Campina Grande – UFCP na Paraíba/ Brasil em 23 de setembro de 2020, as principais empresas de microcontrolador são: A linha PIC da *Microchip*, Intel MCS da Intel e Atmel AVR da Atmel. A empresa Atmel é a linha responsável pelo Arduino, plataforma bastante usado por estudante devido à sua simplicidade e acessibilidade).

ESP32

Tabela 8 – Microcontrolador: ESP32 (Fonte: Elaborado pelo autor)

Modelo	ESP32
Cores	2
Arquitetura	32 bits
<i>Clock</i>	160MHz
<i>Wi-Fi</i>	Sim
<i>Bluetooth</i>	Sim
RAM	512KB
<i>Flash</i>	16Mb
GPIO	36
<i>Interfaces</i>	SPI/ I2C/ UART / I2S / CAN
ADC	18
DAC	2

O modelo utilizado nesse projeto é o ESP32, essa placa é um dispositivo perfeito para projetos de IoT (*Internet of Things* em português, Internet das Coisas) por sua capacidade de conectar-se a internet via *Wi-Fi* ou a outros dispositivos, via *Bluetooth*, já integrado na placa. O poder de processamento do ESP32 é fantástico, com três núcleos em seu processador.

O ESP32 (como mostrado na figura 4) é uma evolução do módulo ESP8266, portanto vários de seus erros e problemas foram melhorados nesse módulo. Para programar o dispositivo é possível utilizar várias IDEs, como a do Arduino, *CodeBlocks*, *Netbeans*, lua e muitas outras. As portas GPIO da placa são capazes de fornecer até 12mA. Dessa forma, elas são utilizadas como entradas e saídas digitais. A placa também conta com 10 sensores de toque capacitivos, que reagem a toques e enviam a informação para a placa.

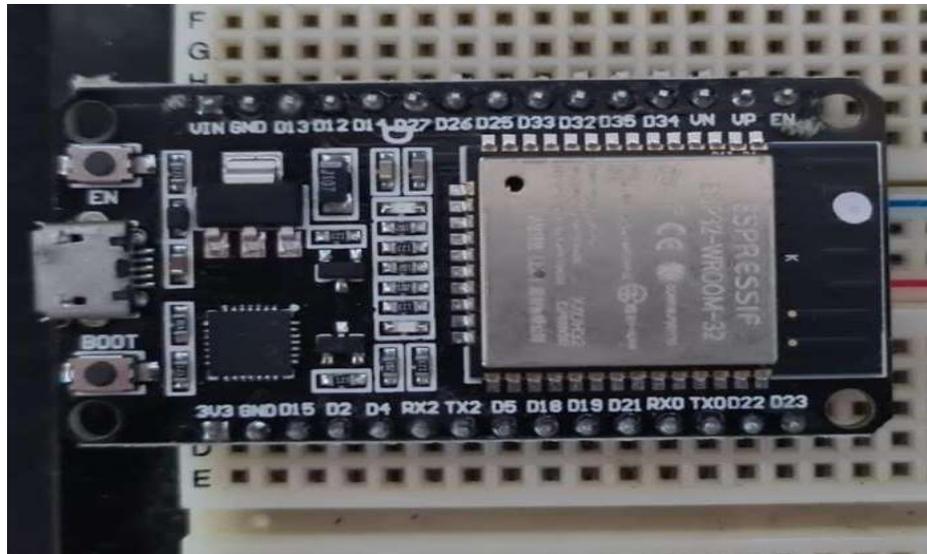


Figura 23 - Placa de desenvolvimento *Wi-Fi e Bluetooth* ESP32 (Fonte: Elaborada pelo autor)

O ADC embutido na placa serve para converter grandezas analógicas em digitais, sendo 16 pinos com essa funcionalidade. A placa conta com portas 5V e 3.3V, GND, pinos de comunicação serial TX e RX, e outros. Assim, em alguns casos, um único pino da placa pode ser utilizado para várias funcionalidades. O ESP32 ainda conta com um sensor de temperatura embutido para controlar a temperatura de operação da placa.

3.4.2. Motor DC

Sua rotação se dá através da troca de energia entre o rotor e o estator, sendo ela através de escovas (motor escovado) ou sem escovas (*brushless*) e com a relação à velocidade, a rotação do motor pode ser controlada variando a tensão de operação, sendo algo mais simples em relação ao motor de corrente alternada, que a velocidade é em função da frequência.

Tabela 9 – Motor (Fonte: Elaborado pelo autor)

Motor DC	6V com redução + roda
Relação da redução	48:01:00
Rotação de motor a 6VDC	260 rpm
Velocidade a 6VDC	1 m/s
Torque a 6VDC	0,8kgf/cm
Perímetro da roda	22cm
Motor DC	6V com redução + roda

motor de corrente contínua é composto por um eixo acoplado ao rotor que é a parte girante do motor. O estator é composto por um imã e o comutador é responsável por transferir a corrente elétrica.

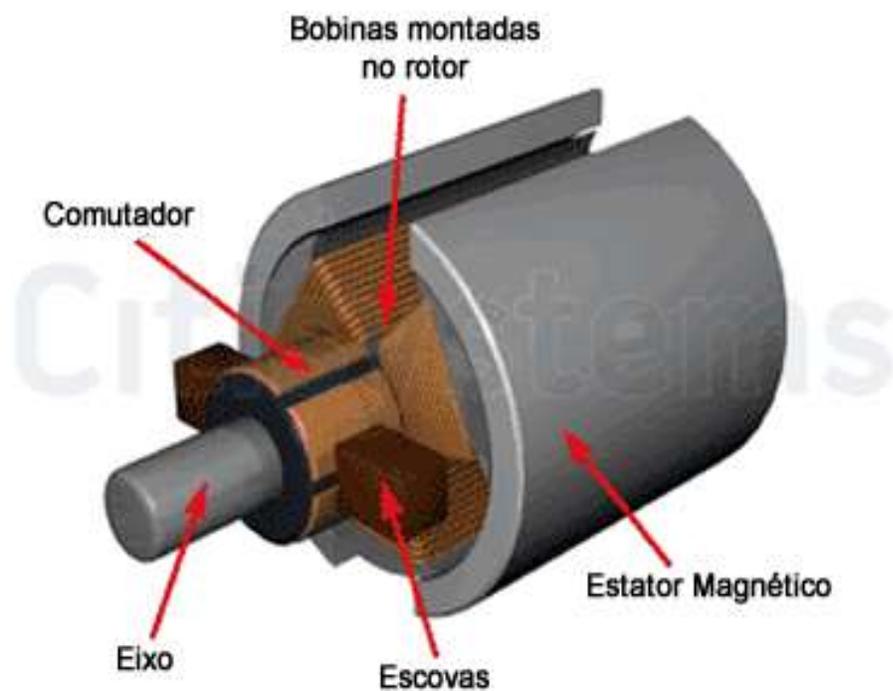


Figura 24 - Composição de um motor de corrente contínua (Fonte: <https://www.citisystems.com.br>)

O estator é constituído por dois ímãs com polaridades opostas, o rotor está sendo representado por uma bobina alimentada pela corrente proveniente do comutador.

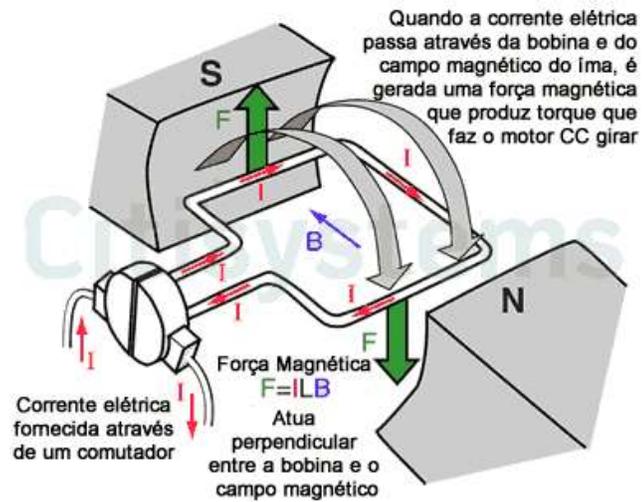


Figura 25 – Motor de corrente contínua (Fonte: <https://www.citisystems.com.br>)

Sempre que uma corrente elétrica passa através de um condutor em um campo magnético, o condutor sofre a ação de uma força mecânica, assim o torque é gerado no eixo do motor e ele entra em movimento.

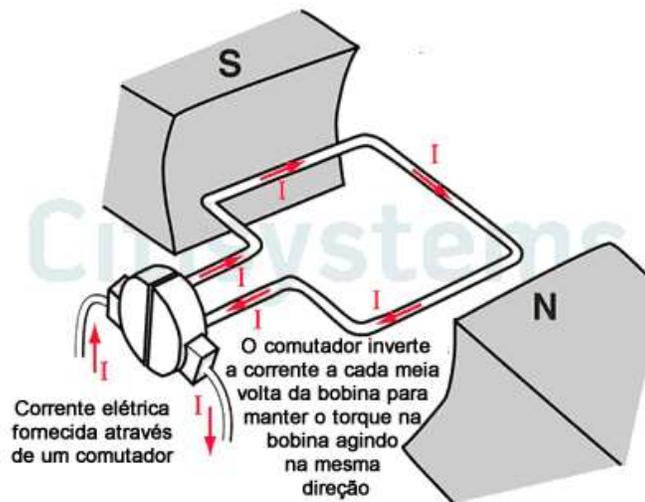


Figura 26- Comportamento da corrente em um motor CC (Fonte: <https://www.citisystems.com.br>)

Ao alimentar o comutador com corrente contínua chamada de I, que é transmitida a bobina através do contato das escovas com o comutador, ele faz a ponte entre a fonte de alimentação e o rotor, o rotor é composto por escovas que fazem o contato com o eixo girando do motor.

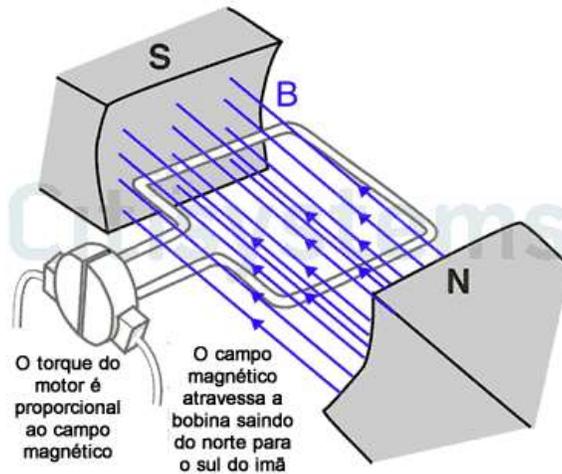


Figura 27 - Representação do campo magnético em um motor CC (Fonte: <https://www.citisystems.com.br>)

O campo magnético é gerado entre os polos do ímã, o sentido parte do Norte em direção ao polo sul. O torque que vai impulsionar o rotor será proporcional ao campo magnético entre os ímãs. A densidade do fluxo magnético é chamada de B.

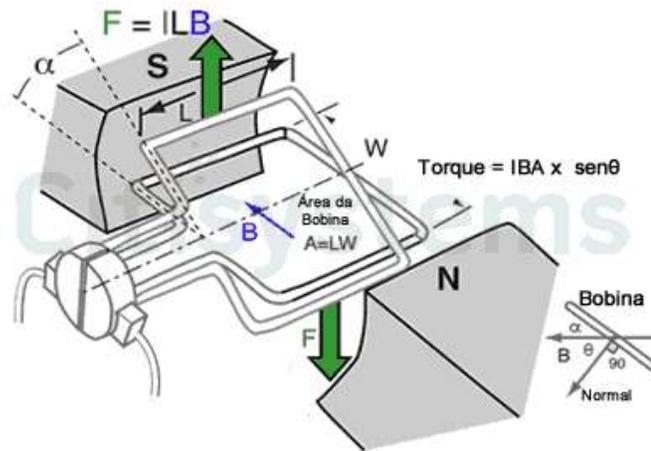


Figura 28 – Torque em um motor CC (Fonte: <https://www.citisystems.com.br>)

A direção da força mecânica e sua magnitude são dadas pela regra da mão esquerda de *Fleming*:

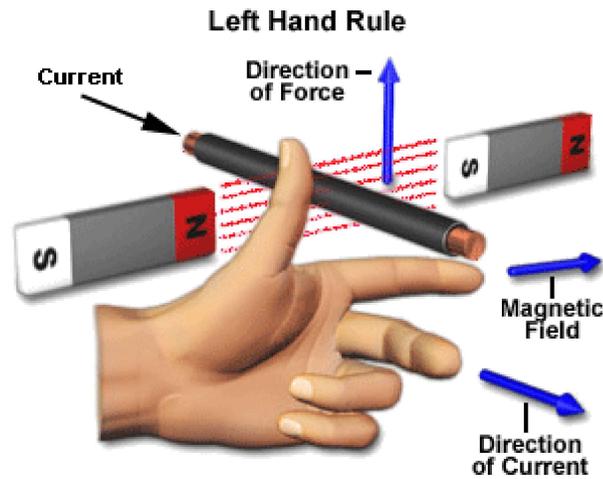


Figura 29 - Regra da mão esquerda (Fonte: <https://www.citisystems.com.br>)

$F = ILB$, onde:

B = densidade de fluxo magnético,

I = corrente da bobina,

L = comprimento do condutor no campo magnético

3.4.3. Baterias de Lítio-íon

Baterias compostas por células de íons de Lítio têm como principal característica a grande vida útil, ausência do efeito memória (degradação do ânodo e/ou do eletrólito) e grande densidade volumétrica de energia. Elas se tornaram economicamente viáveis, podendo ser fabricadas em diferentes associações de células, em busca de maior capacidade, corrente máxima, tensão ou ambas as coisas.

A tecnologia permite que a bateria seja fabricada para diferentes fins, como ter uma grande taxa de descarga, como as baterias desenvolvidas para utilização em drones, automodelo, e entre outros. Nesse caso, em geral, a célula tem maior taxa de autodescarga (perda de energia proveniente das reações químicas internas da bateria) e a possibilidade de ser recarregada de forma rápida com uma alta corrente elétrica em um dos estágios de carga. Baterias voltadas a um fornecimento contínuo de energia tem taxas de descarga/recarga menor e pouco efeito de autodescarga e vida útil prolongada, os smartphones geralmente usam esse tipo de bateria.

As baterias recarregáveis de Lítio se apresentam basicamente em duas configurações, as de Lítio-Polímero (Li-Po) e as chamadas de Lítio-Íon (Li-Ion). Além de algumas diferenças

químicas, as Li-Po geralmente se apresentam como “pacotinhos à vácuo” e as de Li-Íon em invólucros metálicos semelhantes a uma pilha alcalina (Figura 30).

Toda bateria possui tensão nominal, capacidade e corrente de descarga limite determinados pelo fabricante.



Figura 30 - Célula Li-ion (Fonte: <https://www.embarcados.com.br>)

Tabela 10 – Bateria (Fonte: <https://www.produto.mercadolivre.com>)

Modelo	Samsung ICR18650-22P
Voltagem nominal	3,7V
Voltagem com caga completa	4,2V
Voltagem de corte mínima	3V
Capacidade nominal	2200mAh
Capacidade mínima	2100mAh
Corrente de carga	2,2A
Corrente de descarga máxima contínua	4,4A
Altura	65,00mm
Diâmetro	18,40mm

Tensão nominal das baterias

A tensão nominal das células de Li-Po e Li-Íon são 3.7 e 3.6V respectivamente, com pequenas variações dependendo do fabricante e tecnologia empregada. Contudo a tensão pode variar de 4.2 à 3V conforme a descarga da célula, existe também as células de Li-Po High *Voltage*, cuja tensão máxima é de 4.35V. Ultrapassando os limites de descarga pode haver um desgaste prematuro da célula e mesmo um incêndio ou explosão, para o caso de descarga profunda, há um desgaste nos eletrodos, diminuindo a vida útil.

Taxa de carga/ descarga

A taxa de carga, descarga e corrente de curto-circuito é definida pelo fabricante e pode ser dada C, um múltiplo da capacidade nominal. Por exemplo: uma célula de 2200 mAh e corrente de curto-circuito de 30 C pode ser descarregada a 30 vezes 2200 mA, ou seja, 66A. Essa taxa limite não deve ser aplicada por muito tempo sob risco de incêndio ou explosão, por conta do aumento da temperatura das células, entretanto diversos projetos fazem uso dessa alta taxa de descarga, nesse caso deve-se monitorar e até mesmo atuar sobre a temperatura das células, para que a temperatura não exceda o limite recomendado, como é o caso de alguns drones da DJI que usa resfriamento passivo e monitoramento de temperatura em tempo real e os automóveis da Tesla Motors, que utiliza um sistema de refrigeração líquida.

Recarga da bateria

A recarga das baterias de Lítio é talvez o ponto que requer maior cuidado, pois a vida útil da célula depende de uma recarga precisa. A recarga é realizada em 2 ou 3 estágios, sendo os dois primeiros estágios de corrente constante e o último um estágio de tensão constante.

Os estágios de corrente são necessários para carregar a célula de forma segura quando a tensão está baixa, situação em que a célula se comporta como um capacitor com uma resistência em série muito baixa, a corrente de recarga tende a valores altos, o aquecimento pode danificar a célula e mesmo levar ao colapso.

Em bancos de baterias, e mesmo em packs, as células precisam não só serem carregadas, mas também balanceadas, por isso a maioria dos packs tem chicotes de recarga, que dão acesso aos pontos de associação das células. Um desbalanceamento pode afetar profundamente o desempenho, podendo gerar aquecimento desigual nas células mais descarregadas agravando

ainda mais o quadro. Em casos extremos, o banco pode se perder por uma célula bastante afetada ou mesmo morta.

Vida útil

A vida útil das células de íons de Ion-Lítio costuma ser a mais longa entre as células que tem ta taxa de descarga menor, como as usadas em smartphones e notebooks, já as que são projetadas para ter uma taxa de descarga alta, as Li-Po, sua vida útil costuma ser menor, pois elas não podem ficar 100% recarregadas por muito tempo, pois isso acumula gases no pack de células e faz ela inchar, a tornando extremamente volátil e a inutilizando, o nível da mesma também não pode ficar muito baixo, pois diminui muito a vida útil, as fabricantes de drones de filmagem adotam uma tecnologia chamada “*Smart Bateria*”, que são baterias de Li-Po com um microprocessador que monitora, a temperatura e o nível de carga, caso a bateria fique carregada em 100% por um determinado tempo, o sistema aciona algumas resistências para drenar a carga da bateria e a manter em um nível seguro de carga, também monitora a temperatura durante o carregamento, cortando imediatamente a alimentação caso esteja muito quente.

3.4.4. Servo motores

Os servos motores são usados em diversas aplicações quando se deseja movimentar algo de forma precisa e controlada. Sua característica principal é a sua capacidade de movimentar os seus braços até uma posição e mantê-lo, mesmo quando for exercida uma força em outra direção.

Para a construção do braço foi usado dois tipos de servos, sendo:

- Dois servos motores, modelo TD-8325MG utilizado nos eixos 2 e 3.
- Quatro servos motores, modelo MG996R utilizado nos eixos 1,4,5 e 6.

Tabela 11 – Servos motor (Fonte: ReadyTosky)

Modelo	TD-8325MG
Sistema de controle	Controle pwm (1500 u-sec é o ponto central)
Tensão de funcionamento	4.8v ~ 6v
Temperatura de trabalho	-20c ° ~ + 60c °

Torque	4.8v/17.2kg.cm 6.0V /20.3kg.cm
Velocidade ociosa	4.8v/0.15sec / 60 deg 6.0v/0.13sec / 60 graus
Corrente ociosa	4.8v/470ma 6.0v/650ma
Corrente de carga	4.8v/750ma 6.0v/950ma Banda morta: 2usec
Ângulo de movimento	60 ° 10 10 °
Tipo da movimentação	movimentação de mos
Tipo do motor	motor de núcleo
Tipo do potenciômetro	movimentação direta
Tipo do impulsor do sinal	digital Tamanho: 40.5X20.2X40.3mm
Peso	62g (sem acessórios)
Rolamento de esferas	2BB
Material da engrenagem	engrenagem de metal
Material da caixa	pa66 + fiber40 %
Comprimento do fio	260mm
Calibre de fio não e tipo de tomada	26awg, <i>plugue jr</i>
Fio marrom	Negativo
Fio vermelho	Positivo
Fio Laranja	<i>Singal</i>

Tabela 12– Servos motor (Fonte: <https://www.robocore.net/>)

Modelo	MG996R
Tensão de operação	4,8 a 6,0 V
Tensão de funcionamento	4.8v ~ 6v
Tipo de Engrenagem	Metálica
Modulação	Digital
Velocidade de operação	4.8v/0.15sec / 60 deg 6.0v/0.13sec / 60 graus
Torque	9,4 kgf.cm (4,8 V) e 11,0 kgf.cm (6 V)
Faixa de Rotação	180°
Tamanho cabo	300 mm
Dimensões	40 x 19 x 43 mm
Peso	69 g

Partes do servo motor

Para entender melhor seu funcionamento é necessário o conhecimento de suas partes interna, como mostrado na figura 5, que são formadas por:

- **Circuito de controle:** é o responsável pelo monitoramento do potenciômetro e acionamento do motor visando obter uma posição pré-determinada.
- **Potenciômetro:** ele está ligado ao eixo de saída do servo e monitora a posição do mesmo.
- **Motor:** efetua a movimentação das engrenagens e do eixo principal do servo.
- **Engrenagens:** elas reduzem a rotação do motor, são responsáveis por transferir mais torque ao eixo principal de saída e movimentam o potenciômetro em conjunto com o eixo.
- **Caixa do servo:** a caixa serve para alocar as diversas partes do servo.

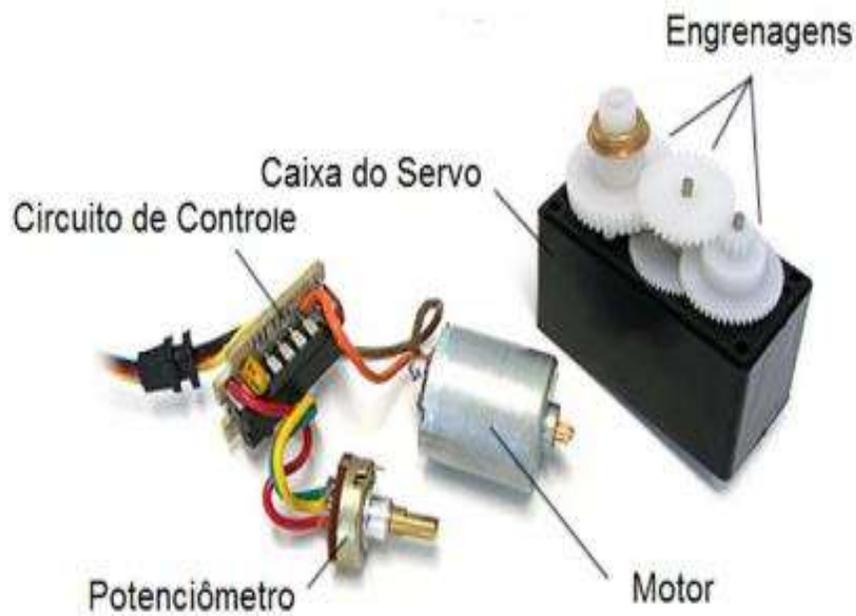


Figura 31 - Partes internas do servo motor (Fonte: <https://www.feis.unesp.br>)

Controle do servo motor

A tensão de alimentação do servo motor é de 5 V e recebe um sinal no formato PWM (*Pulse Width Modulation*). Este sinal é 0 V ou 5 V. O circuito de controle do servo fica monitorando este sinal em intervalos de 20 ms. Se neste intervalo de tempo, o controle detecta uma alteração do sinal na largura do sinal, a posição do eixo é alterada para que a sua posição coincida com o sinal recebido. Um sinal com largura de pulso de 1ms corresponde a posição do servo todo a esquerda ou 0 grau. Um sinal com largura de pulso de 1,5ms corresponde a posição central do servo é dizer, 90 graus. Um sinal com largura de pulso de 2 ms corresponde a posição do servo todo a direita ou 180 graus.

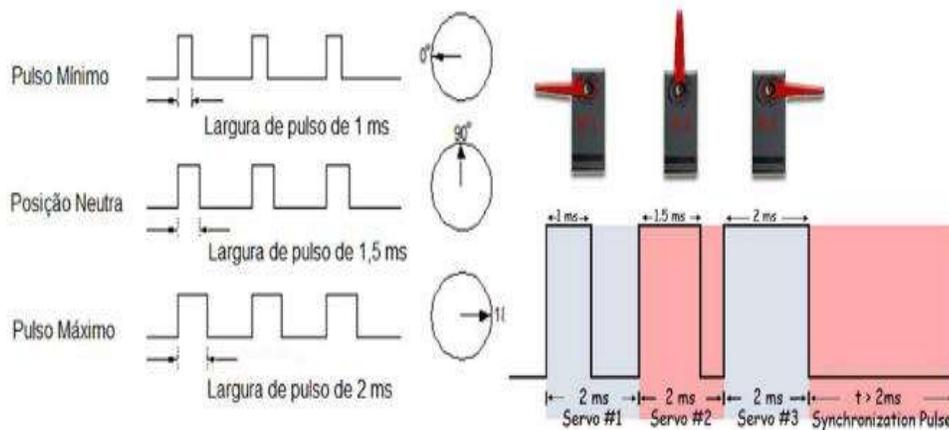


Figura 32 - Sinal de controle do servo (Fonte: <https://miro.medium.com>)

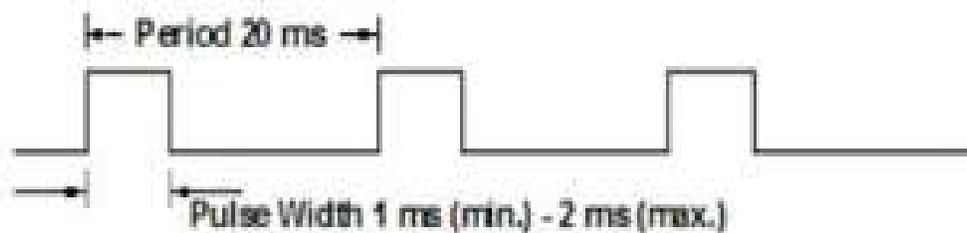


Figura 33 - Ângulos de movimentação do servomotor (Fonte: <http://repositorio.aee.edu.br>)

Uma vez que o servo recebe um sinal de 1,5ms (por exemplo), ele verifica se o potenciômetro encontrasse na posição correspondente, se ele estiver nessa posição nada é feito. Se o potenciômetro não estiver na posição correspondente ao sinal recebido, o circuito de

controle aciona o motor até que o potenciômetro esteja na posição correta. A direção de rotação do servomotor depende da posição do potenciômetro. O motor vai girar na direção leva o potenciômetro mais rapidamente na posição correta.

Ao se tentar alterar a posição do servo motor, verifica-se uma resistência feita pelo motor. Esta resistência é chamada de torque. O torque é uma das principais características do servo motor. Mede-se o torque em KG-cm (Kilo grama por centímetro) ou oz-in (onça por polegada). Uma alternativa para se obter maior robustez é utilizar engrenagens de metal.



Figura 34 - Servo motor submetido a um torque de 1 kg-cm (Fonte: <https://eletroeletronicaetec.wordpress.com>)

Classificação dos servos motores

Os servos motores podem ser classificados em:

- **Standard**, são os servos mais comuns são grandes e robustos, pesam entre 69 de 35 gramas.
- **Mini**, são menores que os standard com um peso aproximado de 20 a 28 gramas.
- **Micro**, são pequenos e leves com um bom torque e pesam entre 6 a 20 gramas

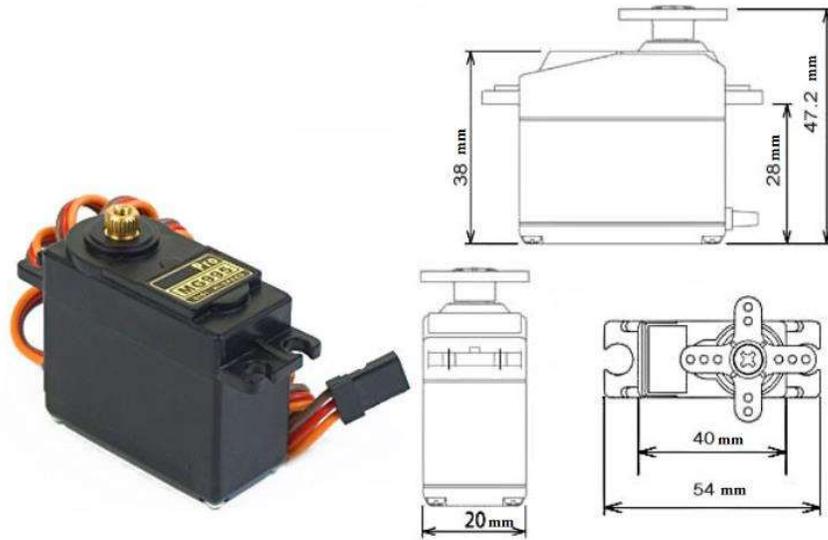


Figura 35 - Tipos de servos motores (Fonte: <https://cdn.shopify.com>)

Servo motor *TowerPro* MG996R

O Servo Motor *TowerPro* MG996R é um servo motor de alto torque, com 13kg/cm a 4.8V e com 15kg/cm a 6V, sua rotação pode ser de 180 graus, devido a essas características, podemos observar sua utilização frequente em projetos de robótica, por exemplo, para a movimentação dos membros de um robô, também é muito utilizado no aeromodelismo para o movimento das asas dos aeromodelos entre outras inúmeras aplicações.

Ele possui engrenagens metálicas de alta durabilidade e confiabilidade. Esse modelo pode ser utilizado com as principais plataformas microcontroladas do mercado, seja ela, *pic*, arduino, *arm* entre outros. As conexões do servo motor MG996R segue o padrão dos principais fabricantes do mercado, são elas: *Hitec*, *GWR*, *Futaba*, *Cirrus* entre outros.



Figura 36 - Imagem do servo *TowerPro* MG996R (Fonte: <https://www.botnroll.com>)

3.4.5. Comunicação I2C

O protocolo de comunicação Inter Integrated Communication (I2C) permite a conexão de diversos equipamentos em uma mesma rede de comunicação. Desenvolvida pela empresa Phillips, é um dos protocolos mais utilizados na comunicação de dispositivos dentro de um mesmo circuito ou equipamento eletrônico (PEREIRA, 2007).

Trata-se de um protocolo síncrono, com a utilização de somente dois fios, sendo um para o *Clock* e outra de Dados. Possibilita a conexão de um equipamento mestre a diversos módulos escravos, contudo também suporta a presença de outros mestres, desde que apenas um esteja ativo, para evitar uma colisão no barramento

Neste trabalho, foi utilizado o protocolo I2C para comunicar um módulo PWM ao microcontrolador ESP32, permitindo assim, a utilização de mais servo motores para o controle da célula robótica.

3.4.6. Porta PWM

A modulação por largura de pulso (PWM) é uma técnica onde é utilizada uma saída digital de um circuito microcontrolado com a finalidade de se obter uma resposta analógica. Seu funcionamento consiste na modulação de pulsos de ondas quadradas entregues a carga.

Com o estabelecimento dos dispositivos baseados em semicondutores, o PWM se tornou um método muito utilizado em equipamentos de eletrônica de potência, sendo aplicada em diversas atividades, desde controle de motores a conversão de energia.

A aplicação dos sinais de PWM consiste no valor médio de uma onda sendo o seu controle realizado através do tempo em que a onda permanece no nível lógico alto. A razão entre o tempo em que o sinal se encontra em nível lógico alto e o tempo total da oscilação é denominado ciclo ativo (*duty cycle*). A alteração do *duty cycle* ocasiona na alteração do valor médio da onda, variando desde 0% em tensão 0 V a 100% com tensão máxima.

Em projeto de programação, o ciclo ativo a ser determinado equivale a um número inteiro, sendo este armazenado em um registrador de 8 bits. Dessa forma, o seu valor vai de 0 (0%) a 255 (100%).

3.4.7. Sensores

Os sensores são uma parte importante da automação, já que é através deles que é capaz de identificar algum objeto, existindo uma gama diversificada de sensores, cada um, para sua devida aplicação, para o desenvolvimento do projeto atual, iremos utilizar os sensores ópticos, para identificar o movimento da célula robótica ao longo do seu percurso e a onde ela deverá parar para realizar a operação programada.

Tabela 13 – Sensores (Fonte: Elaborado pelo autor)

Modelo	LM393
Sensor de obstáculo	IR
Tensão de operação	3.3 à 5V DC
Emissor e receptor	IR
Distância de detecção	2 à 80 cm
Ajuste de Distância	Via Potenciômetro
Dimensões	37 x 14 x 6 mm

Sensor óptico

De acordo com o catálogo de sensores da empresa WEG que desenvolve esses sensores, os sensores ópticos são utilizados para detectar objetos através da emissão de um raio infravermelho, que é recebido por um receptor, identificando assim objetos quando essa emissão é interrompida. Seu funcionamento consiste na utilização de um diodo emissor, que transmite uma luz infravermelha a uma frequência próxima de 1,5kHz, essa mesma frequência é recebida pela foto transistor, que recebe esse sinal e realiza uma comutação eletrônica. Existem algumas maneiras diferentes de aplicar esse tipo de sensor, sendo elas, sistema por barreira, sistema reflexivo, sistema por difusão e sistema por difração.

Iremos utilizar o sistema por difusão no nosso projeto. Esse sistema é utilizado para identificar algum objeto, tanto o emissor quando o receptor está do mesmo lado, o feixe de luz que é emitido pelo emissor é refletido na superfície do objeto e assim recebida pelo receptor, acionando o sensor, desde que o objeto não seja totalmente fosco, como na figura 37.

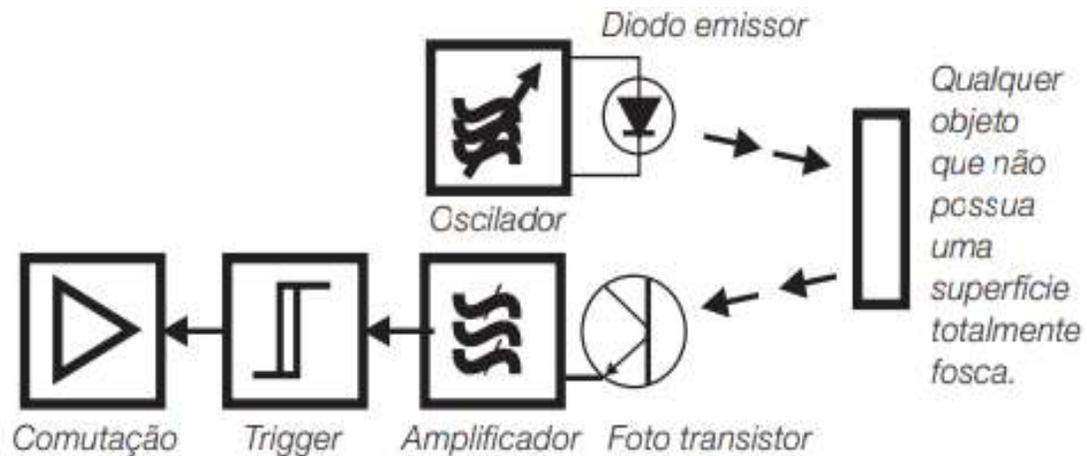


Figura 37 - Sensor óptico por difusão (Fonte: Catálogo de sensores industriais da WEG)

Com o intuito de identificar a mudança de estado do sinal recebido pelo sensor óptico, durante a passagem dele entre uma faixa escura e uma mais clara, será utilizado esse sensor com a função de identificar essa variação durante o movimento da célula robótica no percurso, funcionando assim como um sistema por difusão, como na figura 37.

3.5. Estrutura lógica da célula robótica

Uma linguagem de programação é um recurso padronizado que tem como função expressar instruções para realização de um programa. Para sua estruturação é necessário seguir regras sintáticas e semânticas de forma a definir o programa de computador. Dessa forma, é definido que as regras sintáticas indicam a maneira de escrita e regras semânticas ao conteúdo. Por meio das especificações de uma linguagem de programação demonstra-se quais informações um computador irá utilizar, de que maneira estes dados serão tratados, armazenados e transmitidos e quais operações devem ser realizadas em determinadas circunstâncias.

3.5.1. Linguagem C++

A linguagem de programação C++ é frequentemente denominada de linguagem de médio nível. Contudo, isso não significa que ela seja menos poderosa ou menos desenvolvida.

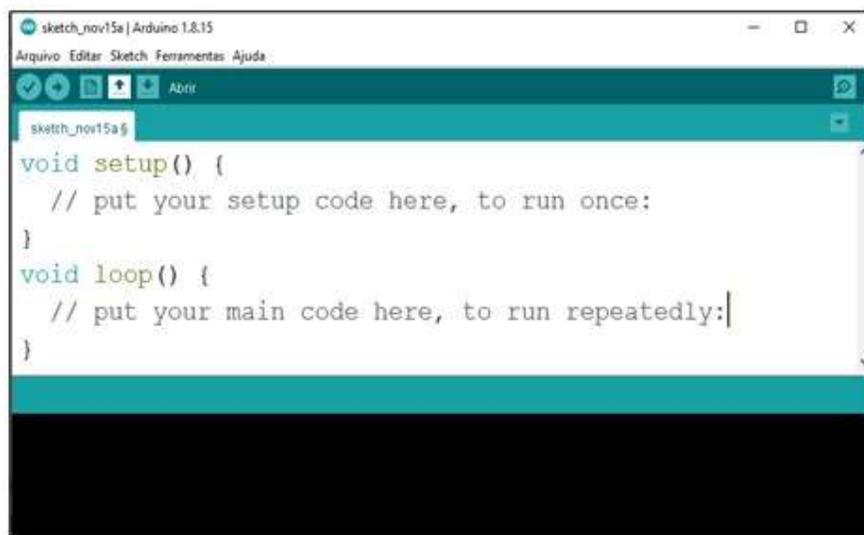
A linguagem C++ é tratada como uma linguagem de médio nível pois relaciona elementos de linguagens de alto nível com as funcionalidades da linguagem *assembly*.

Sendo uma linguagem de médio nível, permite manipular *bits*, *bytes* e endereços, que são elementos básicos de funcionamento de um computador. A linguagem C++ possui uma portabilidade elevada e muito eficiente em termos de desempenho, permitindo ser utilizada para desenvolver os mais variados tipos de programas.

O principal componente estrutural de C++ é a função. Todo programa desenvolvido em C++ consiste em uma ou mais funções. Necessariamente, existe uma função que deve constar em todo programa, que é denominada *main()*, que é a primeira função a ser chamada no momento da execução do programa. Em essência, um código em C, a função *main()* contém um esboço do que o programa faz.

É possível o desenvolvimento de um programa que consista apenas com os comandos feitos pelo programador. Contudo, existe uma coleção de funções, chamada Biblioteca C++ Padrão. Diversas funções básicas ao desenvolvimento de um programa já estão contidas nesta biblioteca, facilitando assim a sua execução.

Para executar um programa em C++, primeiramente é necessário compilar o mesmo. Dessa forma, um compilador realiza a leitura do programa e converte-o no chamado código-objeto, que é uma tradução do código-fonte em um modelo que pode ser executado diretamente pelo computador. Também é conhecido como código binário ou código de máquina. Para isso foi utilizado a IDE do Arduino como mostra na figura 38.



```
sketch_nov15a | Arduino 1.8.15
Arquivo  Editar  Sketch  Ferramentas  Ajuda
sketch_nov15a.g
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Figura 38 - *Layout* no aplicativo APP Inventor (Fonte: Elaborada pelo autor)

3.6. Aplicativo

Dentre as mais variadas plataformas de criação de aplicativos, o presente trabalho optou pela escolha do MIT App Inventor, pois trata-se de um ambiente visual que utiliza a programação em blocos, o que possibilita o desenvolvimento de aplicações para dispositivos móveis Android de forma prática, simples, principalmente se comparada as outras plataformas de desenvolvimento de aplicativos.

Para que o usuário final do aplicativo possa utilizar o mesmo em qualquer local, para isso foi escolhido a plataforma de armazenamento em nuvem Google *Firebase*, pois ela possibilita uma total integração com App Inventor, utilizando ferramentas de comunicação nativas nessas duas plataformas.

3.6.1. Linguagem de programação visual de bloco

A linguagem de programação por meio da utilização de blocos de códigos facilita um maior entendimento no estudo de programação. Este tipo de programação se assemelha a forma de montar objetos com bloco de montagem, como os brinquedos Lego, onde se utilizam peças pré-determinadas que ao ser encaixadas, criam algo funcional.

Por volta do final dos anos 2000, diversas ferramentas para desenvolvimento de programas que se utilizam da linguagem em blocos começaram a surgir. Neste modelo temos como exemplos o *Scratch*, MIT App Inventor, *Blockly* e algumas plataformas de programação para o microcontrolador Arduino.

Segundo Pasternak (2009), essa forma de programação se limita aos encaixes disponibilizados entre os blocos, entretanto este método tem como objetivo diminuir a quantidade de erros que possam ocorrer no processo de elaboração do código, impedindo a possibilidade de o programa possuir incoerências e erros de sintaxe.

Uma programação visual possibilita utilizar todos os elementos de programação, como laços de repetição, estruturas de condição, variáveis entre outras estruturas. Estes objetos são representados em formas de blocos para serem arrastados e encaixados entre si, dessa forma é possível formar funções para solucionar determinado problema. Em uma plataforma de programação, estes os blocos já foram desenvolvidos e estão disponíveis aos programadores para serem utilizados na elaboração de um algoritmo. No momento da criação de um código é

possível verificar quando um bloco encaixado não é compatível ao outros blocos, assim, eliminando problemas que possam ocorrer no programa.

3.6.2. MIT App Inventor

A plataforma de desenvolvimento App Inventor é um ambiente de programação visual online onde é possível realizar a criação de aplicativos com foco em aparelhos moveis que utilizam o sistema operacional Android. Foi criada pelo MIT (*Massachusetts Institute of Technology*) em conjunto com a Google. Sua programação é baseada através de blocos de código, que possibilita o desenvolvedor não necessitar de conhecimentos avançados em programação. Além disto, disponibiliza recursos para que esses aplicativos explorem as várias funcionalidades um dispositivo Android, como reconhecimento de voz, GPS, acelerômetro e conexão com a *Web*.

Basicamente, o ambiente de desenvolvimento do App Inventor é constituído por duas abas denominadas *Designer e Blocks*. Na primeira aba, é possível criar toda a interface visual do aplicativo. Para isto, diversos elementos, os chamados *Components*, localizados na seção *Palette*, estão disponíveis para a criação das funções do aplicativo. Estes elementos podem ser desde botões, imagens, textos, ou até mesmo objetos mais complexos, tais como o *Firebase*, objeto que será discutido mais adiante, mas, basicamente é uma plataforma de banco de dados online. Para utilizar os objetos disponibilizados, basta selecionar o mesmo e arrastar para o *Viewer*, que nada mais é do que a tela de um *Smartphone*.

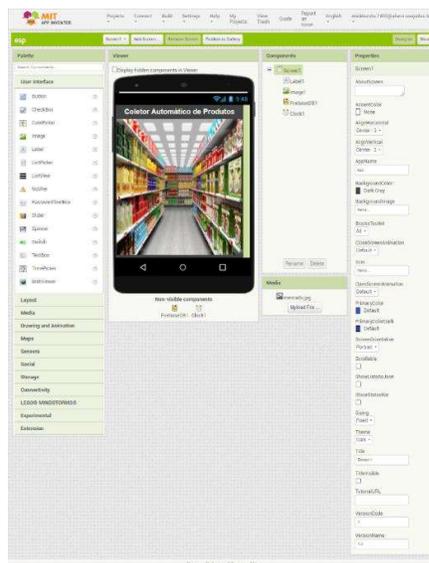


Figura 39 - *Layout* no aplicativo APP Inventor (Fonte: Elaborada pelo autor)

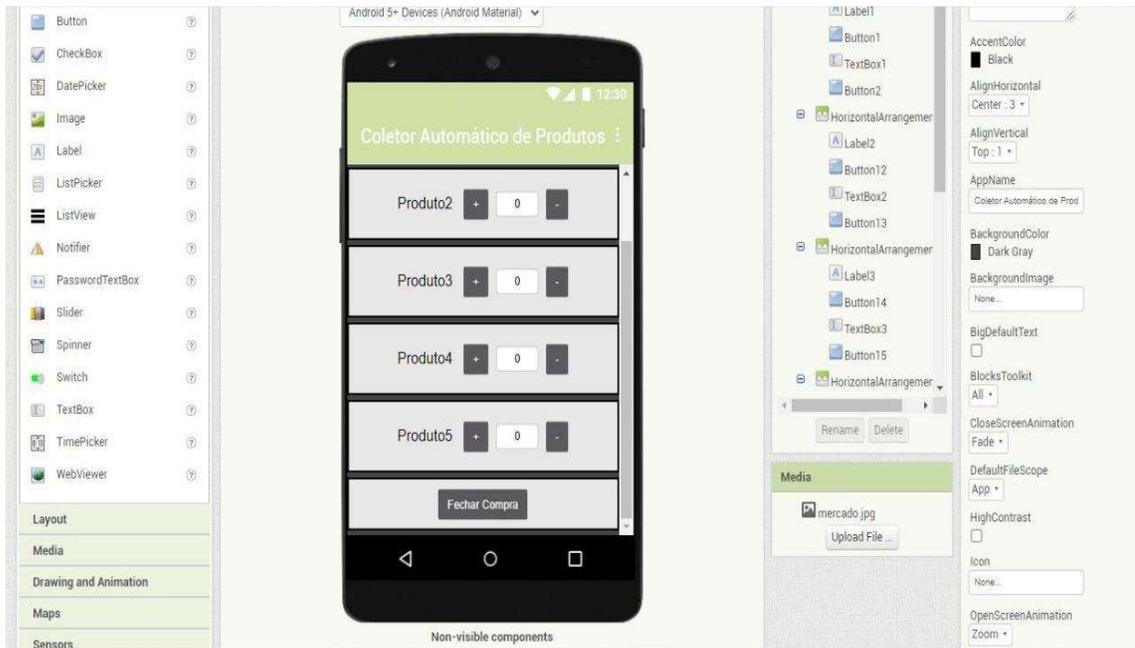


Figura 40- *Layout* do aplicativo (Fonte: Elaborada pelo autor)

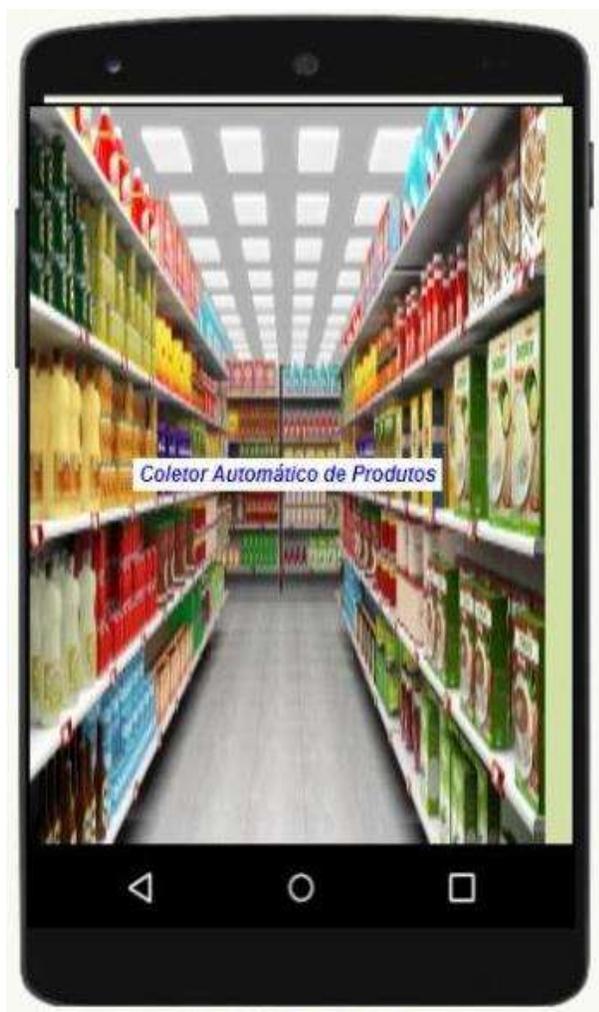


Figura 41 – *Layout* do aplicativo no *smartphone* (Fonte: Elaborada pelo autor)

Na segunda aba, é realizada a programação do aplicativo, ou seja, define-se o comportamento dos componentes criados na primeira aba. Aqui, o desenvolvedor encontra diversos blocos de códigos conectáveis, onde é possível definir eventos ou métodos. Estes blocos operam *strings* e listas, fazem ações de controle, como IF, *ELSE*, etc., e até mesmo operações matemáticas, além de outras funcionalidades. Do mesmo modo, para utilizar um bloco, basta selecionar o mesmo e arrastar para o ambiente de programação.

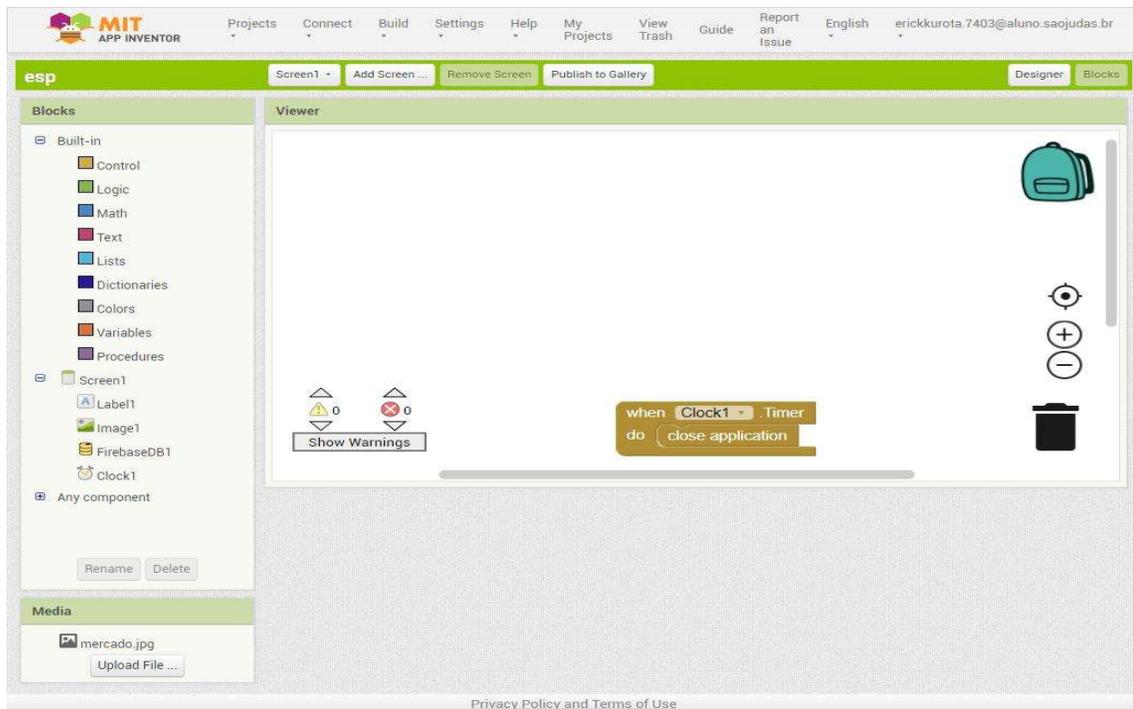


Figura 42 - Programação em bloco do APP Inventor (Fonte: Elaborada pelo autor)

Finalizado o desenvolvimento do software, é possível realizar um teste de execução do mesmo diretamente em um dispositivo *Android*. Para isso, o App Inventor possibilita a execução em um emulador ou até mesmo em um aparelho físico, onde é gerado um *QR Code*, com a aplicação para ser instalada no aparelho.

O site oficial do App Inventor disponibiliza uma grande variedade de materiais de referência, que demonstram as suas mais diversas utilizações. Além disto, em sua comunidade, encontra-se um repositório com aplicações criadas por desenvolvedores do mundo todo, sendo possível realizar o download dos blocos de códigos e dos aplicativos criados.

O App Inventor proporciona diversas possibilidades de desenvolvimento de aplicativos, mesmo que o desenvolvedor não possua um conhecimento técnico elevado em programação. Porém, deve-se observar uma certa limitação na criação do *software*, uma vez que se utiliza

blocos pré-definidos de código. Outro fator é que não é possível exportar os blocos em linguagem de código, o que seria útil para desenvolvedores mais experientes.

3.6.3. Google Firebase

O Firebase é uma plataforma pertencente a Google, cujo objetivo está no desenvolvimento, integração e implantação de aplicativos *Web*, fornecendo aos desenvolvedores diversas funcionalidades e serviços relacionados à internet das coisas, como hospedagem, estatísticas de utilização e diversas ferramentas avançadas, como análise de performance e erros, aprendizado de máquina, testes e projeções de escala.

Uma das diversas ferramentas disponibilizadas pelo Firebase é o *Realtime Database*. Esta ferramenta é um banco de dados hospedado em nuvem, onde os dados são armazenados como JSON e sincronizados em tempo real para todos os usuários conectados nesta rede.

O *Firebase Realtime Database* utiliza a sincronização de dados, para que sempre que os são alterados, todo dispositivo que esteja conecta a ele irá receber essa atualização em milissegundos.

Os dados são armazenados localmente e, mesmo quando o dispositivo conectado a rede esteja offline, os eventos em tempo real permanecem sendo acionados. Quando o dispositivo retoma a conexão, todas as alterações nos dados locais são sincronizadas, mesclando qualquer conflito que possa ocorrer de maneira automática.

Por meio da ferramenta *Firebase Authentication*, é possível ao desenvolvedor definir quem terá acesso, a quais dados e como esses dados podem lidos e gravados.

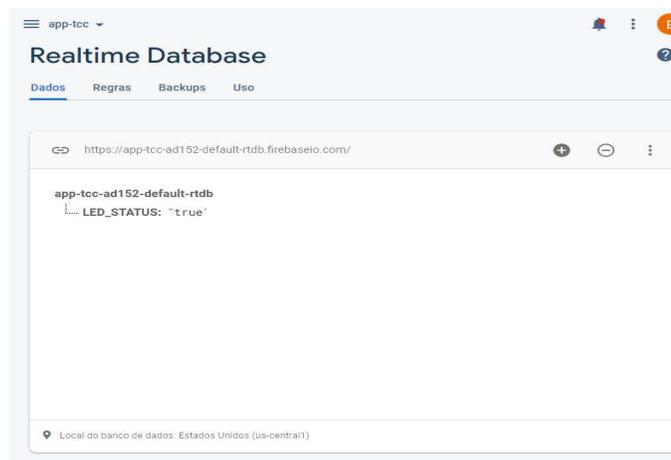


Figura 43 - Tela do *firebase* (Fonte: Elaborada pelo autor)

3.7. Circuito de movimentação da célula robótica

Tabela 14 – Circuito de movimentação da célula robótica (Fonte: Elaborado pelo autor)

Ponto de parada	Descrição
Ponto A	Ponto inicial do trajeto de separação dos produtos
Ponto B	Primeiro ponto de parada - Prateleira 1
Ponto C	Segundo ponto de parada - Prateleira 2
Ponto D	Check in
Ponto E	Entrega

Após a célula robótica receber as informações do pedido realizado via aplicativo, a célula robótica irá armazenar essas informações e começará o processo de separação dos produtos.

A célula robótica sairá do ponto A e irá percorrer o trajeto pré-estabelecido, até chegar no ponto B, onde irá checar se o pedido que foi feito está na prateleira 1, caso o pedido esteja nessa prateleira o braço irá começar a se movimentar para pegar o produto e inserir na sexta dentro do carrinho.

Saindo do ponto B, a célula robótica irá continuar a trajetória até chegar no ponto C, nesse ponto a célula irá parar novamente, e através da lógica de programação, irá checar se algum pedido do cliente se encontra na prateleira 2, se tiver ele irá pegar e depositar no carrinho, caso não, à célula irá continuar o percurso até o ponto D.

No ponto D, será onde a célula tomará algumas decisões com base nas informações que estiver recebendo, esse ponto serve como *check in*, após a finalização da separação dos produtos a célula irá armazenar o pedido dentro da caixa (ponto D) e ficará esperando o sinal, serão dois sinais:

Sinal 1: Caso o cliente que efetuou o pedido via aplicativo, tenha chegado no local de entrega e efetuado a confirmação para retirada, a célula irá retirar o pedido do ponto D e irá prosseguir o trajeto até o ponto E e irá efetuar a entrega para o cliente.

Sinal 2: Caso o cliente ainda não tenha efetuado a confirmação para retirada, a célula robótica irá continuar no ponto D, até que ela receba a informação de um novo pedido. Caso

ela receba a informação de um novo pedido, a célula irá se movimentar para o ponto inicial (Ponto A) e irá começar a fazer a trajetória de separação dos produtos novamente.

Possíveis causas:

Causa 1: Caso o pedido 1 esteja no ponto D esperando a confirmação de retirada e a célula robótica esteja efetuando a separação dos produtos do pedido 2 e o cliente do pedido 1 confirma no aplicativo que está no local para retirada do pedido. A célula robótica irá continuar realizando a separação do pedido 2 e ao chegar no ponto D, irá armazenar o pedido 2 que ficará aguardando a confirmação do cliente para retirada, feito isso a célula irá pegar o pedido 1 e irá efetuar a entrega para o cliente.

Causa 2: Caso a célula chega no ponto D com o pedido 2 e o cliente desse pedido já tenha feito a confirmação para retirada, antes do cliente do pedido 1, a célula irá realizar a entrega do pedido de quem confirmou no aplicativo a retirado do pedido primeiro, nesse caso o pedido 2.

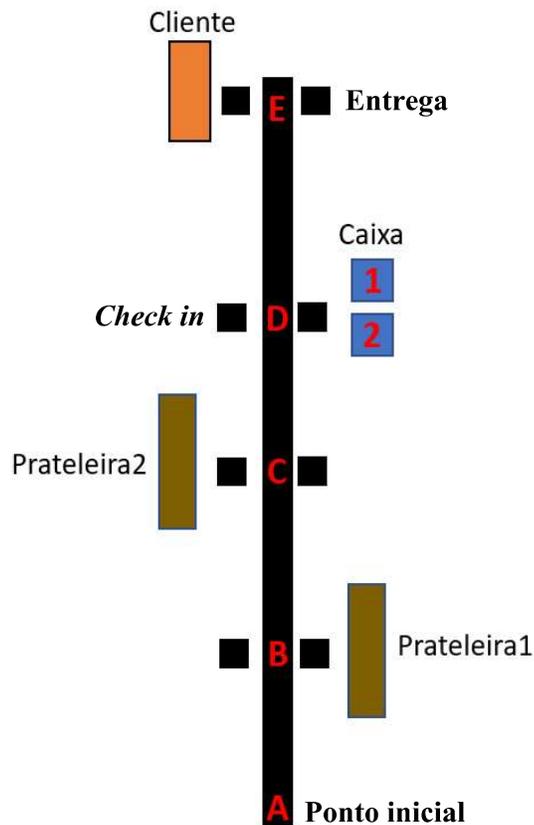


Figura 44 - Circuito de movimentação da célula robótica (Fonte: Elaborado pelo autor)

4. Resultados e discussão

4.1. Cálculo da corrente, tensão e potência usada no circuito

4.1.1. Consumo do circuito de controle

Para mensurar o consumo do sistema, foi utilizado um amperímetro de corrente contínua para aferir o consumo individual de cada componente do sistema. Assim, ao final das medições foi possível constatar o consumo total demandado pelo sistema para que a fonte de energia fosse dimensionada de acordo.

Cálculo do consumo do sistema

Com as informações de tensão e corrente de cada componente descrita na tabela 15, é possível calcular a potência do sistema inteiro, conforme a equação:

$$P=V \cdot A$$

$$P=(V1 \cdot A1)+(V2 \cdot A2)+(V3 \cdot A3)+(V4 \cdot A4)+(V5 \cdot A5)+4 \cdot (V6 \cdot A6)+2 \cdot (V7 \cdot A7)+4 \cdot (V8 \cdot A8)$$

$$P=(6 \cdot 0,1)+(6 \cdot 0,2)+(6 \cdot 0,2)+(5 \cdot 0,2)+(6 \cdot 0,2)+4 \cdot (6 \cdot 0,9)+2 \cdot (6 \cdot 0,2)+4 \cdot (6 \cdot 0,2)$$

$$P = 34W$$

$$I = 5,7A$$

Tabela 15 - Consumo do sistema (Elaborado pelo autor)

Posição	Componente	Quantidade Utilizada	Corrente (A)	Tensão (V)	Potência (W)
1°	Módulo seguidor de Linha TCRT5000	1	0,1	6	0,6
2°	Placa L298N	1	0,2	6	1,2
3°	Driver PWM/Servo 16 Canais	1	0,2	6	1,2
4°	ESP 32	1	0,2	5	1
5°	DC_DC_Converter_XL4016	1	0,2	6	1,2
6°	Servo MG996R	4	0,9	6	21,6
7°	Servo TD-8325MG	2	0,2	6	2,4
8°	Motor DC com caixa de redução 48:1	4	0,2	6	4,8
	Consumo Total	-	5,7 (A)	-	34 (W)

4.1.2. Cálculo da fonte de energia

A fonte de energia que alimentará o circuito terá que entregar uma tensão de 6 volts e ser capaz de entregar uma corrente acima de 6 amperes. Tendo em vista essa necessidade, foi escolhido utilizar 4 células de lítio SAMSUNG ICR18650-22P, com corrente máxima de descarga de 10 amperes e tensão entre 3,62V e 4,2V, a tensão máxima é atingida quando a célula está totalmente carregada. As células estão dispostas em 2 conjuntos com duas células cada, os conjuntos estão ligados em série como mostra a figura abaixo:



Figura 45 – Disposição das células de lítio (Fonte: <https://www.produto.mercadolivre.com>)

O conjunto nessa disposição entrega uma tensão entre 7,24V a 8,4V que é regulado pelo DC_DC_Converter_XL4016 que mantém a tensão do circuito estável em 6V.

4.1.3. Cálculo dos motores utilizados

O sistema completo pesa 3,5kg, assim o torque combinado dos quatro motores utilizados no projeto tem que ser maior que 3,5kgf com a disposição conforme a figura abaixo.

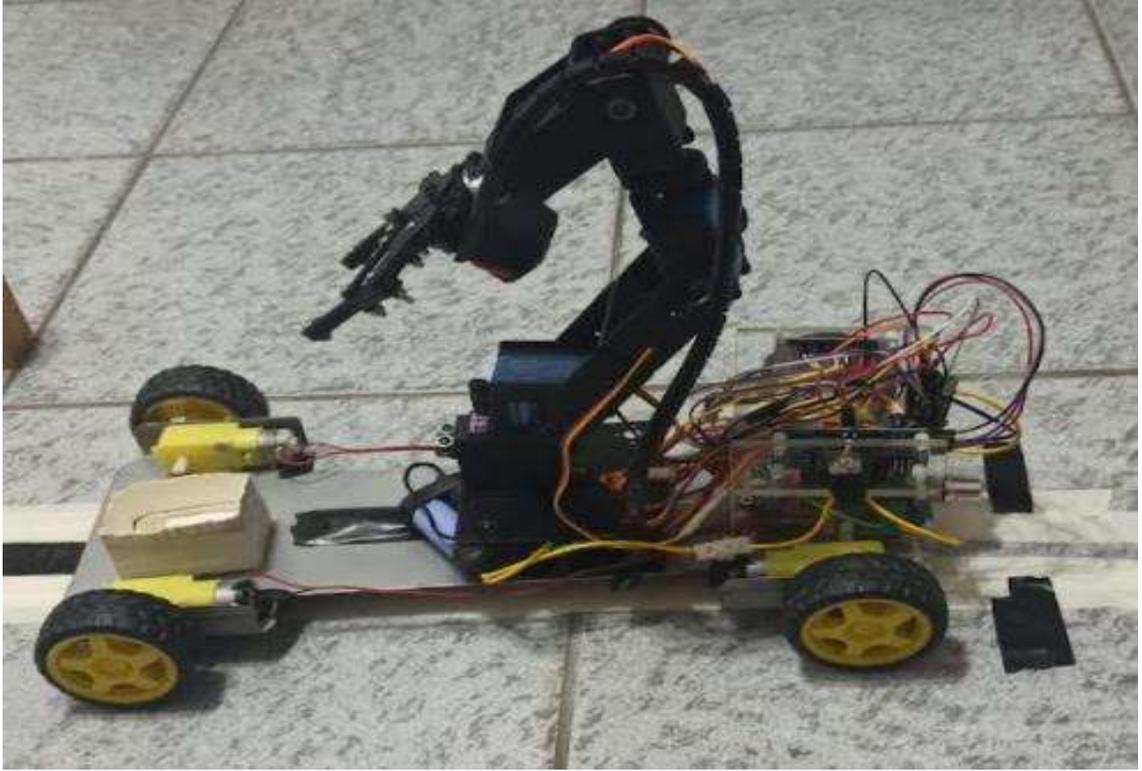


Figura 46 - Célula robótica: Ângulo 1 (Fonte: Elaborado pelo autor)



Figura 47 - Célula robótica: Ângulo 2 (Fonte: Elaborado pelo autor)

Para atender essa demanda optou-se por utilizar um motor DC de 6V com uma caixa de redução acoplado a uma roda.



Figura 48 – Motor DC com caixa de redução (Fonte: <https://www.baudaeletronica.com.br>)

Tabela 16 – Relação de transmissão e perímetro da roda (Elaborado pelo autor)

Motor DC	6V com redução + roda
Relação da redução	48:01:00
Rotação de motor a 6VDC	260 rpm
Velocidade a 6VDC	1 m/s
Torque a 6VDC	0,8kgf/cm
Perímetro da roda	22cm

Com as informações de torque, perímetro da roda e relação de redução é possível calcular o torque na roda acoplada ao motor, conforme as seguintes equações:

$$0,8\text{kgf/cm} = 7,84\text{N}$$

$$T = F \cdot 48$$

$$T = 7,84 \cdot 48$$

$$T = 376,32\text{Ncm}$$

Mas esse é o torque no eixo do motor, é necessário calcular o torque na periferia da roda conforme as equações abaixo:

$$T = F/R$$

$$T = 376,32/11$$

$$T = 34,21\text{N} = 3,488\text{kgf/cm}$$

4.2. Cálculo dos motores utilizados, torque, rotação, relação de transmissão

4.2.1. Cinemática do braço robótico

Neste item são descritas um conjunto de características importantes para o planejamento dos movimentos do braço robótico. A determinação do espaço de operação, torque dos motores e a análise da cinemática inversa são aspectos fundamentais e que merecem atenção.

Espaço de trabalho

Tendo como base as medidas úteis dos elos do braço robótico, é possível determinar o volume do espaço de trabalho do braço. Na Figura 48 estão detalhadas as dimensões dos elos do braço:

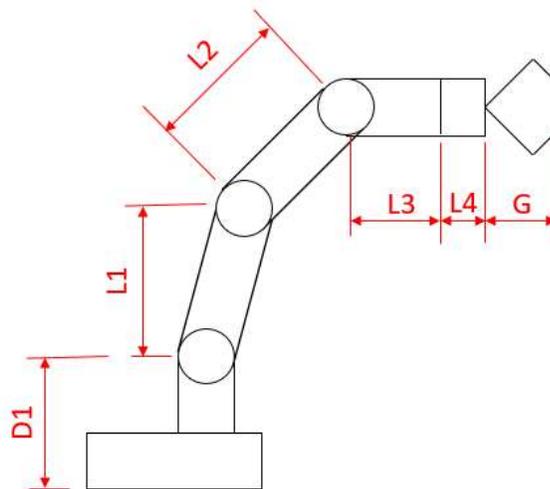


Figura 49 - Dimensões do braço robótico (Fonte: Elaborado pelo autor)

Tabela 17 – Dimensões (Fonte: Elaborado pelo autor)

Motor DC	6V Com redução + roda
D1	6,5
L1	10,5
L2	9,0
L3	8,0
L4	3,0
G	7,0

A área de trabalho deste tipo de braço robótico é uma semiesfera, dessa forma é possível determinar o volume da região de trabalho através da seguinte fórmula:

$$V = \int_0^r \pi(r^2 - x^2)dx$$

$$V = \pi \left(r^3 - \frac{x^3}{3} \right)$$

$$V = \frac{2}{3} \pi r^3$$

A 49 demonstra o arco de operação do braço robótico.

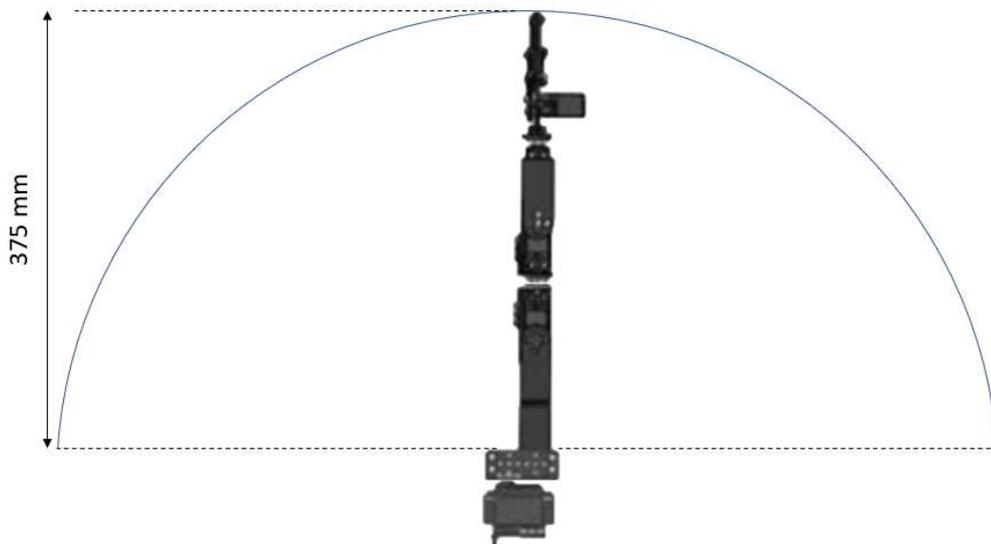


Figura 50 - Espaço de trabalho (Fonte: Toppingmall)

Inserindo o valor do raio r da semiesfera na fórmula, obtêm-se o valor do volume do espaço de trabalho:

$$V = \frac{2}{3} \pi r^3$$

$$V = \frac{2}{3} \pi 0,375^3$$

$$V = 0,1104 \text{ m}^3$$

Devido às limitações impostas para o seu funcionamento, este braço robótico irá trabalhar apenas a metade da semiesfera, dessa forma:

$$V = \frac{0,1104}{2}$$

$$V = 0,0552 \text{ m}^3$$

4.2.2. Cálculo de Torque do Braço

Para identificar o torque necessário dos servomotores a serem utilizados no braço robótico, primeiramente é preciso conhecer as forças aplicadas ao braço, conforme a Figura 51.

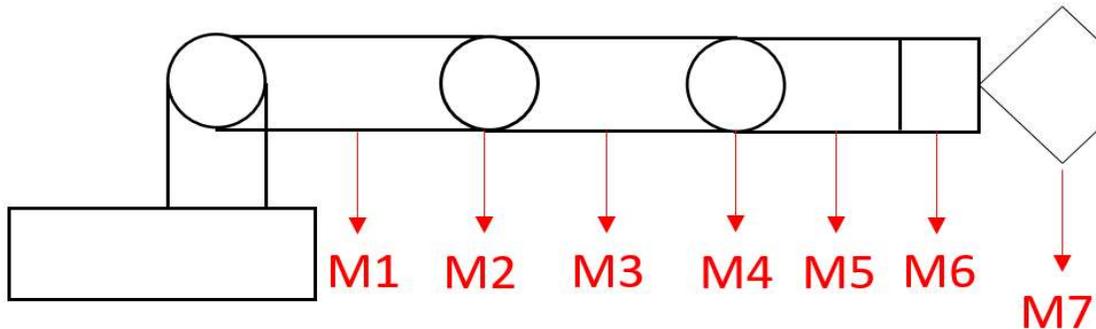


Figura 51 - Forças aplicadas ao braço (Fonte: Elaborado por autor)

Tabela 18 – Massa (Fonte: Elaborado pelo autor)

Dados	Massa (kg)
M1	0,045
M2	0,065
M3	0,035
M4	0,060
M5	0,020
M6	0,060
M7	0,123

Na Figura 51, é possível verificar que quando o braço robótico se encontra paralelo a superfície é a posição que se demanda maior torque em todo o sistema, sendo exercida diretamente no servo da base

Com as informações de massa e as dimensões contidas na Tabela 18, é possível calcular os torques necessários para os servomotores A, B e C, conforme as seguintes equações:

$$TA - M7 \times \left(L1 + L2 + L3 + L4 + \frac{G}{2} \right) - M6 \times \left(L1 + L2 + L3 + \frac{L4}{2} \right) - M5 \times \left(L1 + L2 + \frac{L3}{2} \right) - M4 \times (L1 + L2) - M3 \times \left(L1 + \frac{L2}{2} \right) - M2 \times L1 - M1 \times \frac{L1}{2} = 0$$

$$TA = 4,182 + 1,74 + 0,47 + 1,17 + 0,525 + 0,6825 + 0,2362$$

$$TA = \mathbf{9,0057 \text{ kg.cm}}$$

$$TB - M7 \times \left(L2 + L3 + L4 + \frac{G}{2} \right) - M6 \times \left(L2 + L3 + \frac{L4}{2} \right) - M5 \times \left(L2 + \frac{L3}{2} \right) - M4 \times L2 - M3 \times \frac{L2}{2} = 0$$

$$TB = 2,8905 + 1,11 + 0,26 + 0,54 + 0,1575$$

$$TB = 4,9580 \text{ kg. cm}$$

$$TC - M7 \times \left(L3 + L4 + \frac{G}{2} \right) - M6 \times \left(L3 + \frac{L4}{2} \right) - M5 \times \left(\frac{L3}{2} \right) = 0$$

$$TC = 1,7835 + 0,57 + 0,09$$

$$TC = 2,4435 \text{ kg. cm}$$

4.2.3. Cinemática Inversa

Para calcular os ângulos das juntas do braço robótico em função das coordenadas em que se deseja que a garra alcance foi utilizado a cinemática inversa. Para isso, foi determinado os pontos P_x , P_y e P_z , que são as coordenadas com relação ao seu respectivo eixo. Também foi determinado a inclinação garra em relação a base, denominada por ϕ .

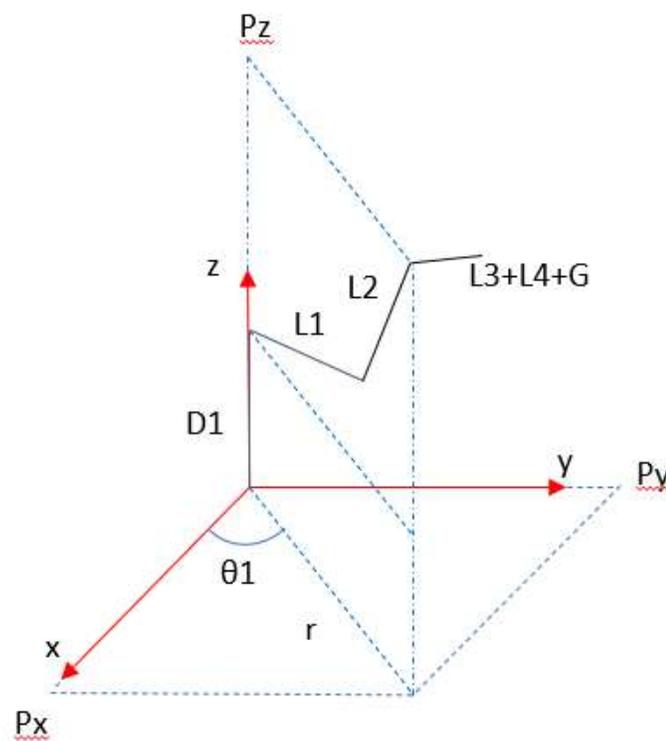


Figura 52 - Plano XYZ – ângulo θ_1 (Fonte: Elaborado pelo autor)

Conforme verificado na Figura 52 - Plano XYZ, é possível obter, através do triângulo formado, o ângulo da base, sendo utilizada a seguinte fórmula:

$$\theta_1 = \arctan\left(\frac{Py}{Px}\right)$$

Ainda na Figura 52, é possível observar que o comprimento da origem do manipulador até a garra, forma a hipotenusa do triângulo retângulo, dessa forma:

$$r = \sqrt{(Px^2 + Py^2)}$$

Relacionando esta informação com a figura 53, vemos que o valor r refere-se ao eixo y, dessa forma, constituindo um triângulo retângulo, em conjunto com d e hipotenusa h1.

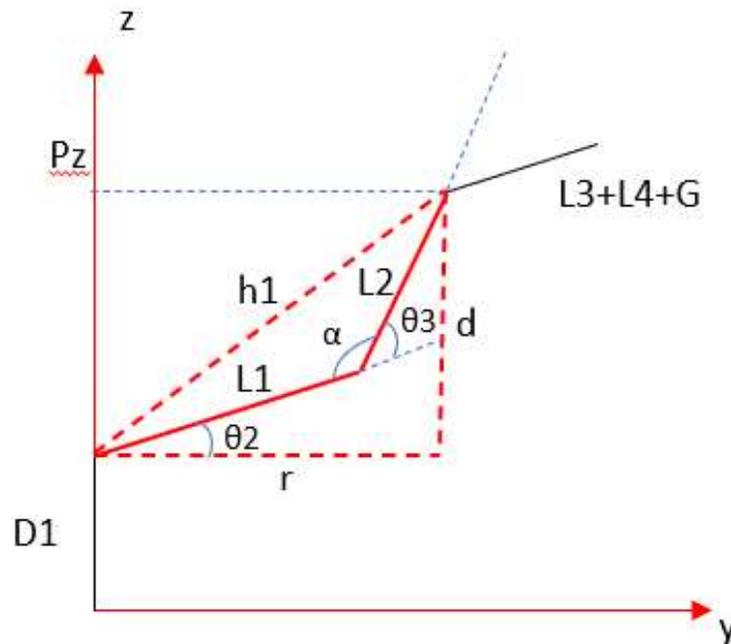


Figura 53 - Plano YZ – ângulo θ_3 (Fonte: Elaborado pelo autor)

Temos que d é:

$$d = Pz - D1$$

Dessa maneira, podemos obter o valor da hipotenusa h_1 , pela equação:

$$h_1^2 = r^2 + (Pz - D1)^2$$
$$h_1^2 = Px^2 + Py^2 + (Pz - D1)^2$$

Para determinar θ_3 , podemos obter através da lei dos cossenos, aplicada no triângulo L_1, L_2 e h_1 .

Primeiramente, sabemos que:

$$\alpha = 180 - \theta_3$$

Então:

$$h_1^2 = L_1^2 + L_2^2 - 2L_1L_2\cos\alpha$$
$$Px^2 + Py^2 + (Pz - D1)^2 = L_1^2 + L_2^2 - 2L_1L_2\cos\alpha$$
$$-\cos\alpha = \frac{Px^2 + Py^2 + (Pz - D1)^2 - L_1^2 - L_2^2}{2L_1L_2}$$

Sabemos que:

$$\cos(\pi - \beta) = -\cos\beta$$

Substituindo:

$$-\cos(180 - \theta_3) = \frac{Px^2 + Py^2 + (Pz - D1)^2 - L_1^2 - L_2^2}{2L_1L_2}$$
$$\cos\theta_3 = \frac{Px^2 + Py^2 + (Pz - D1)^2 - L_1^2 - L_2^2}{2L_1L_2}$$

Então, para definir θ_3 , basta calcular o arco cosseno:

$$\theta_3 = \arccos\left(\frac{Px^2 + Py^2 + (Pz - D1)^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

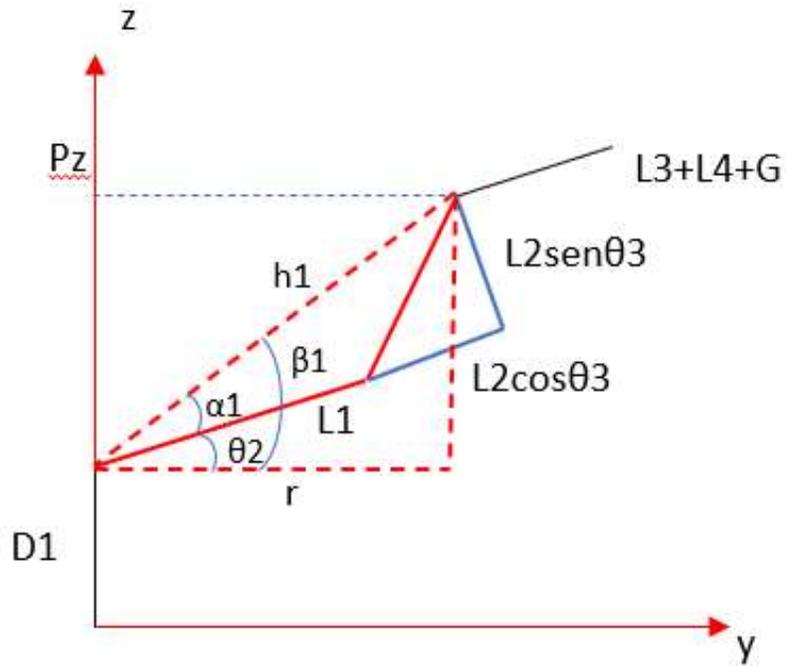


Figura 54 - Plano YZ – ângulo θ_2 (Fonte: Elaborado pelo autor)

De acordo com a Figura 54, para encontrarmos θ_2 , temos que:

$$\theta_2 = \beta_1 - \alpha_1$$

Para obter o valor de β_1 , utilizamos o triângulo formado por d , r e h_1 :

$$\beta_1 = \text{atan} \left(\frac{Pz - D1}{\sqrt{(Px^2 + Py^2)}} \right)$$

E para obtermos o valor de α_1 :

$$\alpha_1 = \text{atan} \left(\frac{L2 \text{sen } \theta_3}{L1 + L2 \text{cos } \theta_3} \right)$$

Substituindo os valores:

$$\theta_2 = \text{atan}\left(\frac{Pz - D1}{\sqrt{(Px^2 + Py^2)}}\right) - \text{atan}\left(\frac{L2 \sin \theta_3}{L1 + L2 \cos \theta_3}\right)$$

O método geométrico da cinemática inversa possui algumas limitações com relação ao número de graus de liberdade.

Para que seja possível a utilização deste método em um braço com três graus de liberdade, é preciso ter conhecimento previamente de um deles. Essa análise fica evidente através da determinação de θ_2 e θ_3 , em que os valores são obtidos através da localização desejada do pulso.

O que se pretende agora é estimar uma posição da garra em relação ao pulso. Sendo assim:

$$\phi = \theta_2 + \theta_3 + \theta_4$$

Onde ϕ é o valor resultante do posicionamento devido os ângulos das três juntas. Dessa maneira, ao manter o valor de ϕ constante, é possível manter sempre a relação entre o pulso e a garra.

Dessa forma, podemos calcular θ_4 :

$$\theta_4 = \phi - (\theta_2 + \theta_3)$$

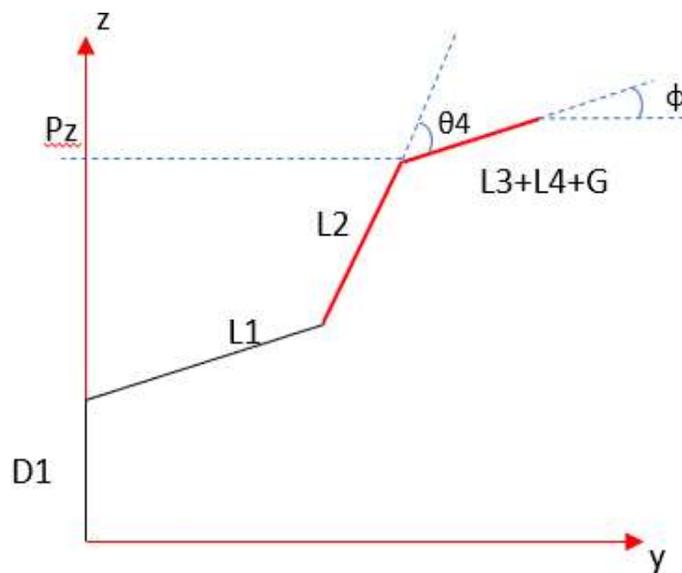


Figura 55 - Plano YZ – ângulo θ_4 (Fonte: Elaborado pelo autor)

4.2.4. Previsão Comportamental do Braço Robótico

Com base nos cálculos de cinemática inversa demonstrados, foi desenvolvido no software Microsoft Excel um gráfico interativo com a previsão do comportamento dos movimentos do braço robótico.

Neste gráfico interativo, basta indicar as coordenadas no plano cartesiano em que se deseja posicionar a garra do robô. Dessa forma, sendo possível verificar a trajetória teórica que o braço robótico irá percorrer.

Tabela 19 – Gráfico de previsão dos movimentos do braço robótico (Fonte: Elaborado pelo autor)

Cinemática inversa	
Braço robótico	
Elos	cm
D1	6.5
L1	10.5
L2	9
L3	8
L4	3
G	7
$Lx(L3+L4+G)$	18
Coordenadas	
X	0
Y	30
Z	0

Tabela 20 - Gráfico de previsão dos movimentos do braço robótico: Ângulos (Fonte: Elaborado pelo autor)

Ângulos	Radiano	Graus
01	0	0
02	1.1599	66.4565
03	-0.6974	-39.9589
04	-0.4625	-26.4977

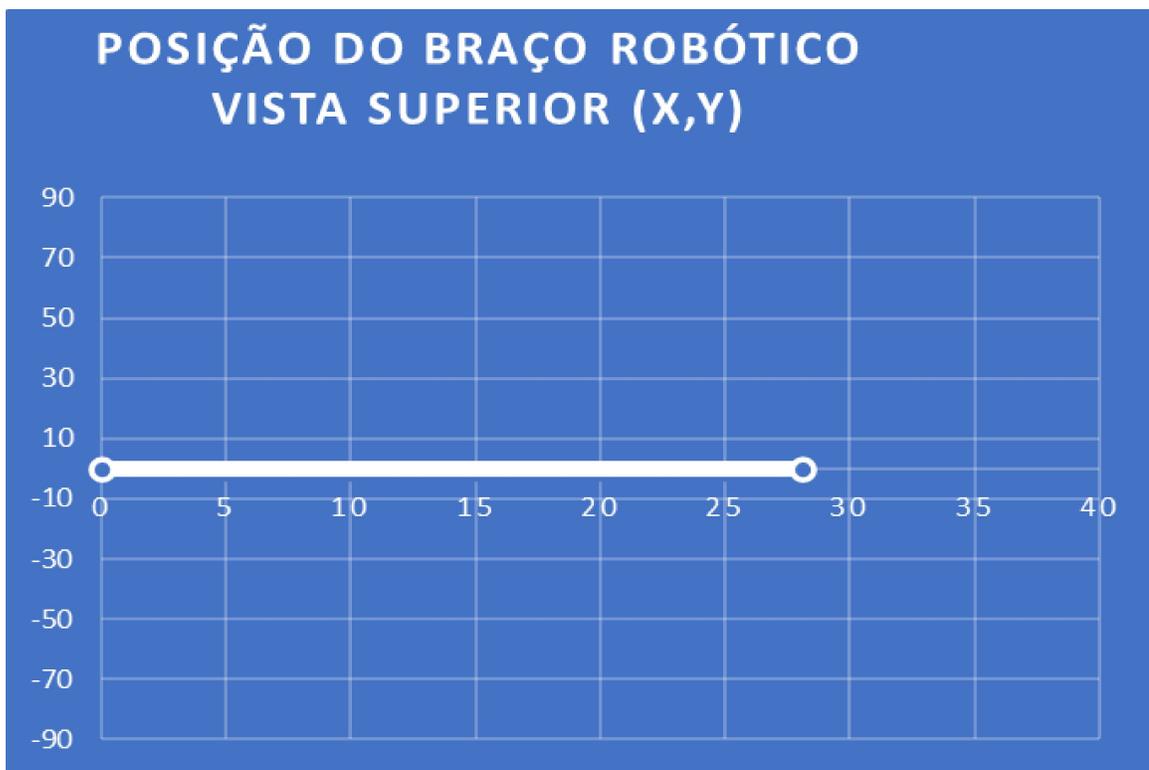
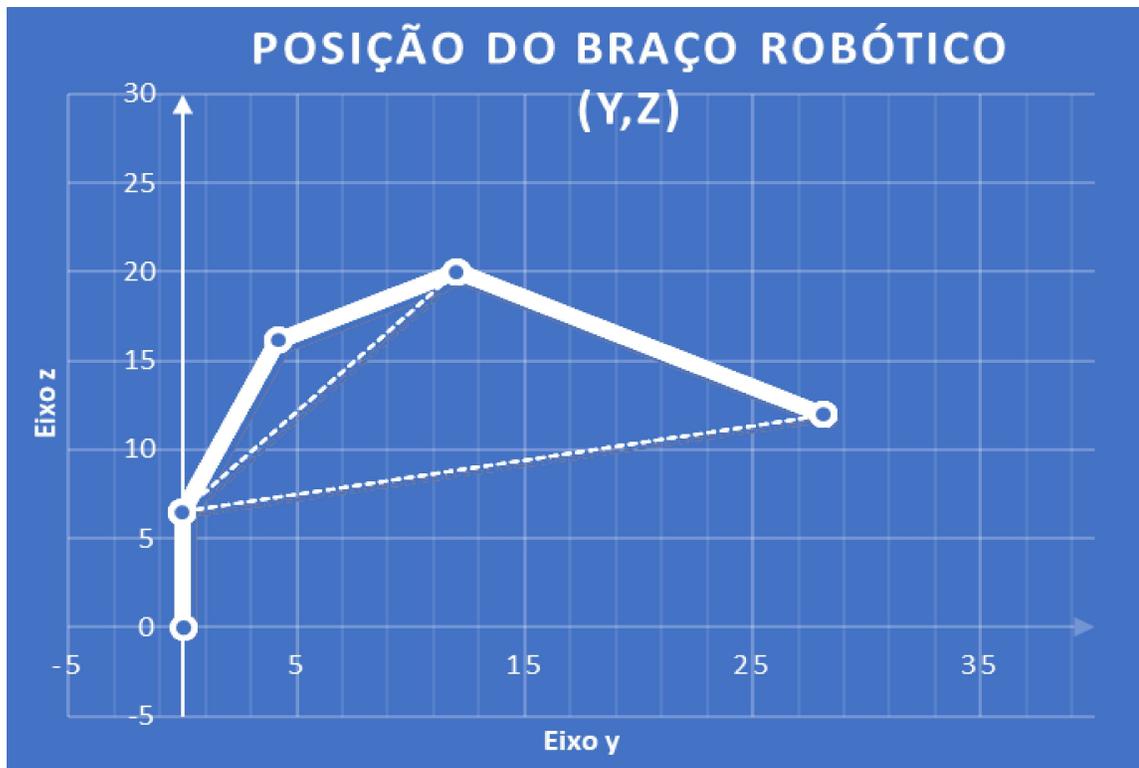


Figura 56 - Gráfico de previsão dos movimentos do braço robótico (Fonte: Elaborado pelo autor)

4.3. Bibliotecas utilizadas na programação

4.3.1. <WiFi.h>

A biblioteca é a que permite a comunicação *Wi-Fi* entre o microcontrolador e o aplicativo de compras.

No início do código se define nome e *password* do usuário.

```
#define WIFI_SSID "Define o nome de usuário da conexão Wi-fi".
```

```
#define WIFI_PASSWORD "password do usuário".
```

Após uma série de impressões seriais obtemos a confirmação se a comunicação entre microcontrolador e aplicativo foi estabelecida.

Através desta biblioteca podemos mapear os estados das diferentes entradas e saídas de forma online, subir dados para a nuvens etc.

4.3.2. <Adafruit_PWMServoDriver.h>

Esta biblioteca permite controlar os servos motores e os ângulos simultaneamente. Em nosso caso, iremos controlar seis servo motores.

No loop desta biblioteca vamos ler os servos declarados um a um como também seus ângulos. A função -for- vai fazer uma varredura dos servos motores e incrementar valores aos ângulos estabelecidos, ditos ângulos podem ser incrementados de 0 a 180 graus, dessa forma conseguimos controlar o servo e o ângulo desejado.

A biblioteca admite protocolo de comunicação I2C utilizada para a troca de dados entre a ESP32 e a placa Shields PCA9685 utilizada para o controle dos servos. O envio de dado é através das portas SDA e SCL, GPIO 21 e GPIO 22 no microcontrolador.

4.3.3. <IOXhop_FirebaseESP32.h>

Biblioteca encarregada de fazer a comunicação entre a ESP32 e o banco de dados *Firebase*.

Um ponto importante do código é sua inicialização, onde é preciso informar o nome e a senha da conexão *Wi-Fi*, também o link do seu banco de dados e a senha.

```
#define FIREBASE_HOST " ";  
  
#define FIREBASE_AUTH " ";  
  
#define WIFI_SSID " ";  
  
#define WIFI_PASSWORD " ";
```

As funções principais são 3, a função *Get* lê os dados do *Firebase* em tempo real e envia a informação para o microcontrolador. Enquanto a função *Set* é encarregada por escrever no banco de dados do *Firebase*. Por último, a função *Push* cria um novo índice para o objeto.

Desta forma conseguimos interagir entre nosso aplicativo destinado a efetuar pedidos e o microcontrolador ESP32 encarregado pelo processamento e execução das tarefas.

4.3.4. <Wire.h>

A biblioteca permite o protocolo de comunicação I2C entre dispositivos.

O master enviara a informação para o *slave* através do barramento I2C. Dito barramento pode conter até 128 dispositivos. As portas utilizadas para a comunicação são SDA (Serial data) e SCL (Serial *clock*).

Na Rotina de envio de dados se definem os endereços em hexadecimal, se inicializa a transmissão e se indica para qual endereço deve ser efetuada a transmissão, por último encerra-se a transmissão de dados. Desta forma conseguimos integrar diferentes dispositivos através do barramento I2C.

```

#include "Wire.h"

#define d1 0x101
#define d2 0x102
#define d3 0x103

void setup() {
  Wire.begin();
}

void loop() {
  Wire.beginTransmission(d1);
  Wire.write(1);
  Wire.endTransmission();
}

```

Figura 57 – Programação usando a biblioteca *Wire.h* (Fonte: Elaborado pelo autor)

4.3.5. <*analogWrite.h*>

Controla a velocidade dos motores através do PWM nos pinos 16 e 17 da esp32 o valor varia entre 0 e 255.

```

//Variáveis para os motores
//Controlar velocidade Lado esquerdo
#define M1_pwm 16
//Controlar velocidade Lado direito
#define M2_pwm 17

```

Figura 58 – Programação usando a biblioteca <*analogWrite.h*> (Fonte: Elaborado pelo autor)

4.4. Circuito elétrico

O circuito é alimentado por baterias de lítio de +-6V com corrente aproximada de 8A, que é regulado por um DC_DC_Converter_XL4016 para manter a tensão estável em 6V, e alimentar todo o circuito, sendo seus componentes a esp32, e um driver I2C -PCA9685, Drive Ponte H L298N, e um regulador de tensão LM7805 para abaixar a tensão em 5V, para alimentar o Módulo Seguidor de Linha TCRT5000 e um resistor de PullUp para a comunicação I2C que funcionam com tensões menores no circuito. A saída A da ponte H é ligado em dois motores do lado esquerdo do carrinho, e a saída B é ligado em dois motores do lado direito, as entradas de controle de sentido de rotação da ponte H são ligadas nas saídas 13,27,26 e 25 da esp32 que irá mandar um sinal alto ou baixo para controlar a rotação, e as saídas 16 e 17 de PWM da esp32 nas entradas MA e MB da ponte H para controlar a velocidade dos motores.

O driver PCA9685 será utilizado para controlar os 6 servos do robô, para isso, ele utiliza uma comunicação I2C com a esp32 através das entradas SDA e SCL do I2C com as portas 21 e 22 respectivamente da esp32 que já são configuradas para a comunicação I2C. O modulo Seguidor de linha TCRT5000 possui 5 sensores ópticos infravermelhos utilizados para detectar as variações de reflexão em uma superfície clara e escura, mandando um sinal para as entradas 34,32,36 e 39 da esp32. A esp32 receberá sinais via wi-fi.

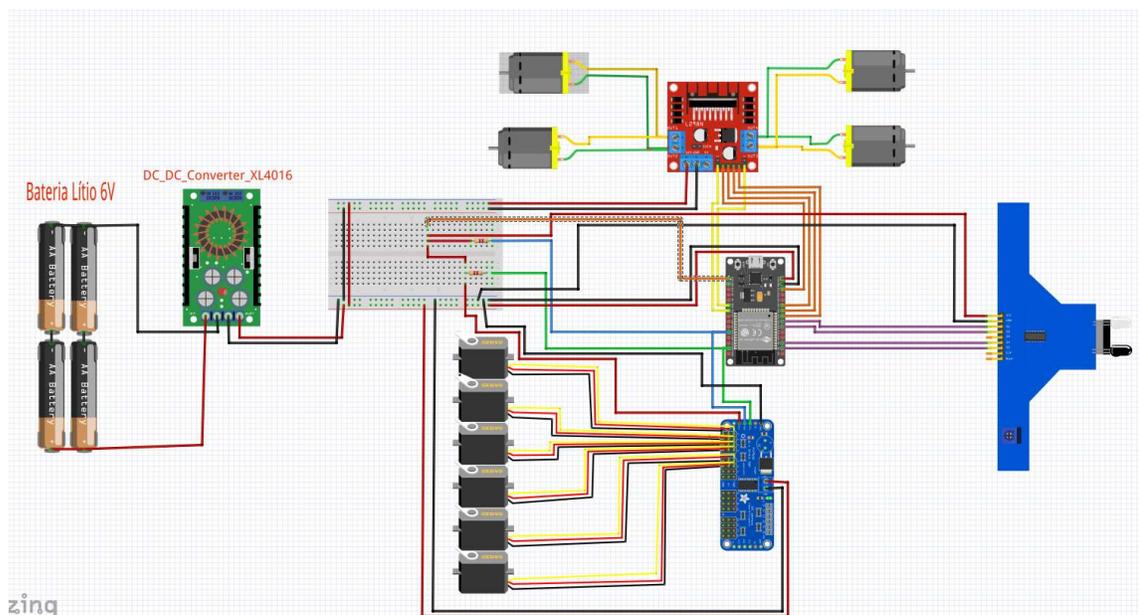


Figura 59 - Esquema elétrico das ligações entre a esp32 e todos os periféricos para funcionamento do projeto (Fonte: Elaborada pelo autor com o software Fritzing)

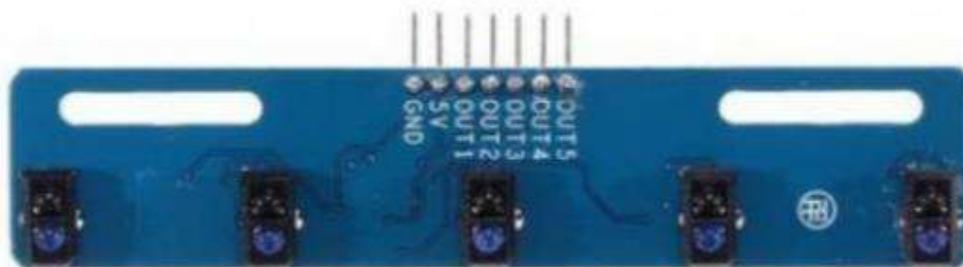


Figura 60 - Módulo Seguidor de Linha TCRT5000 - 5 Canais (Fonte: <https://www.baudaeletronica.com.br/>)

O L298N é responsável por controlar o sentido de rotação através das entradas IN1, IN2, IN3 e IN4, e a velocidade através das entradas MA e MB, controlando os motores A e B respectivamente, o circuito é alimentado pela entrada input por uma fonte de 6V a 35V, e ligado os motores nos outputs para o motor A e motor B.

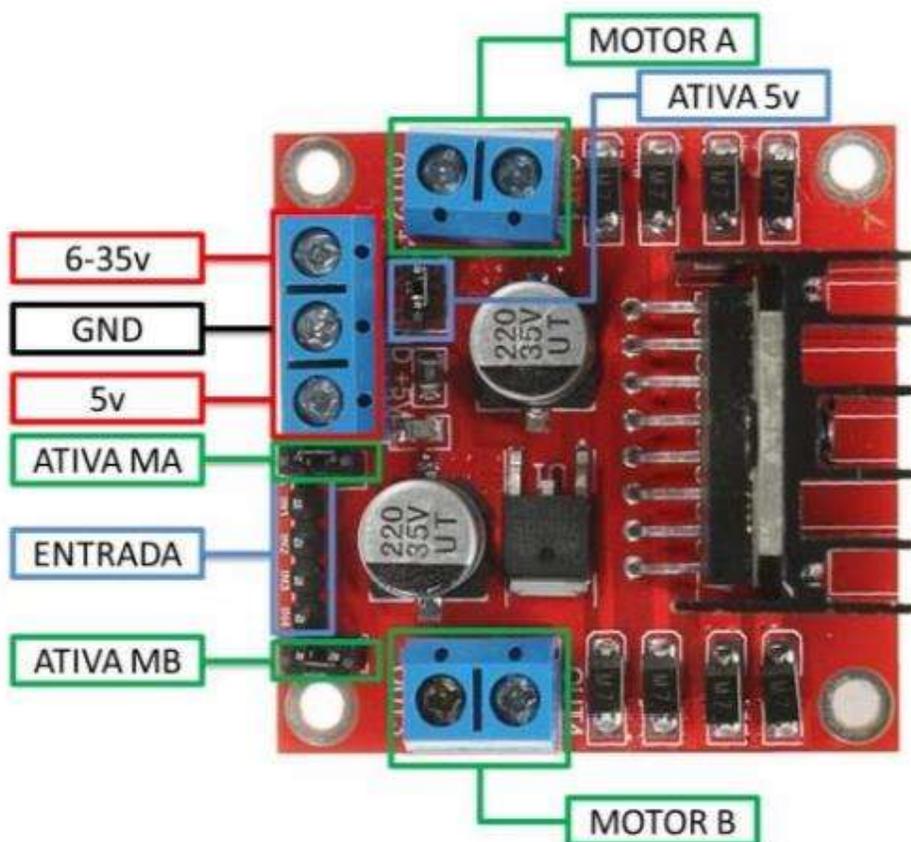


Figura 61 - Placa L298N (Fonte: <https://uploads.filipeflop.com/>)

O driver PCA9685 é responsável por controlar até 16 servos motores simultaneamente, através de uma comunicação I2C com o dispositivo.

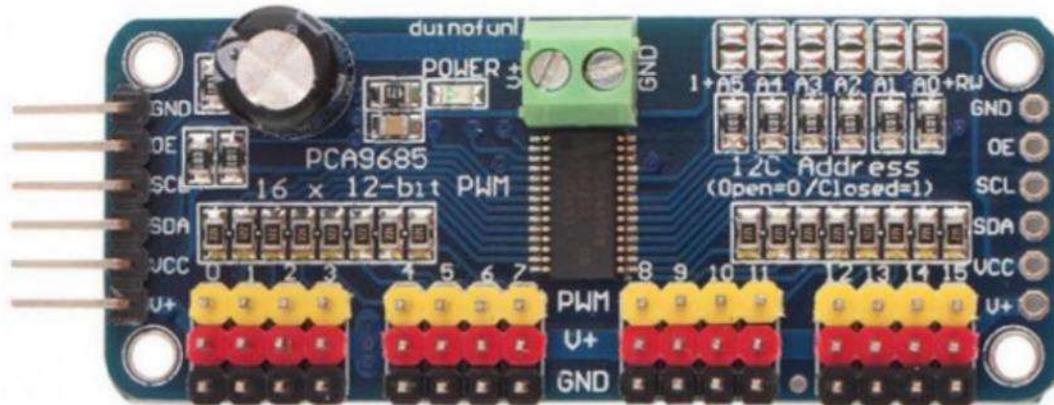


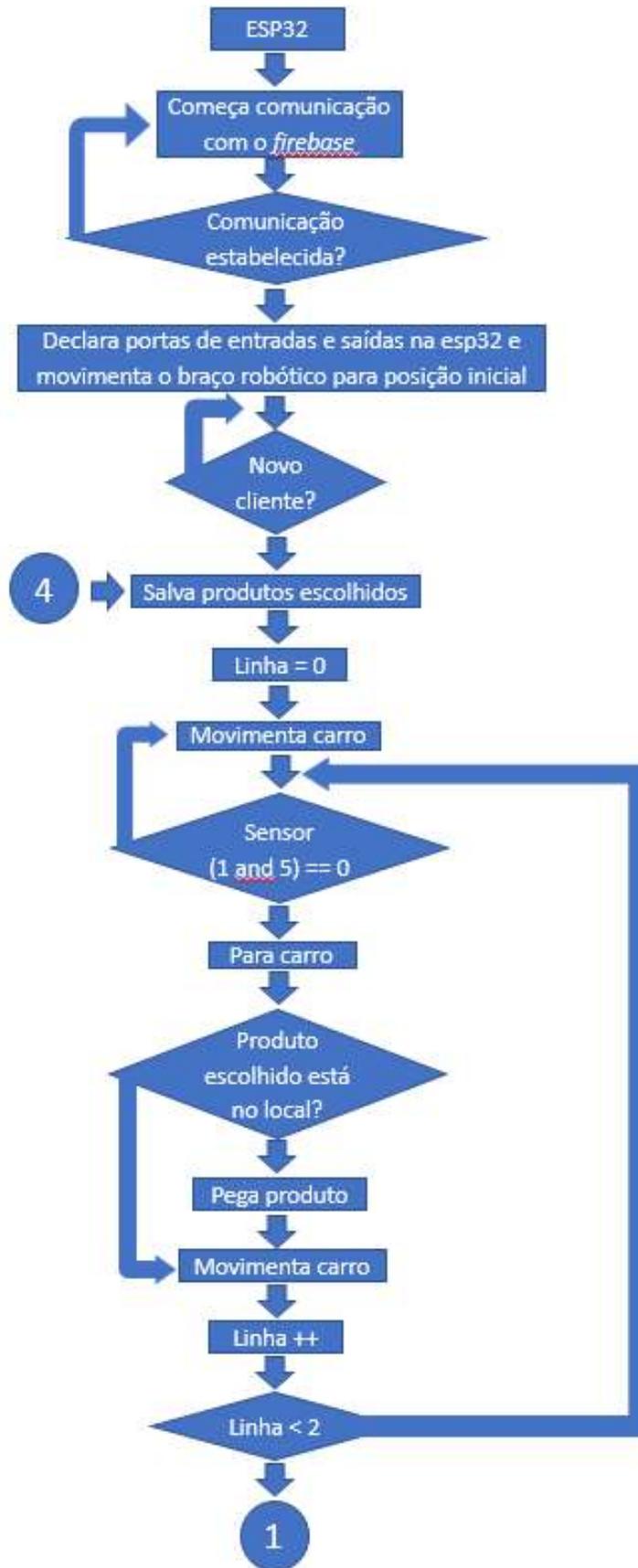
Figura 62 - Driver PWM/Servo 16 canais 12-bit com interface I2C (Fonte: <https://produto.mercadolivre.com.br/>)

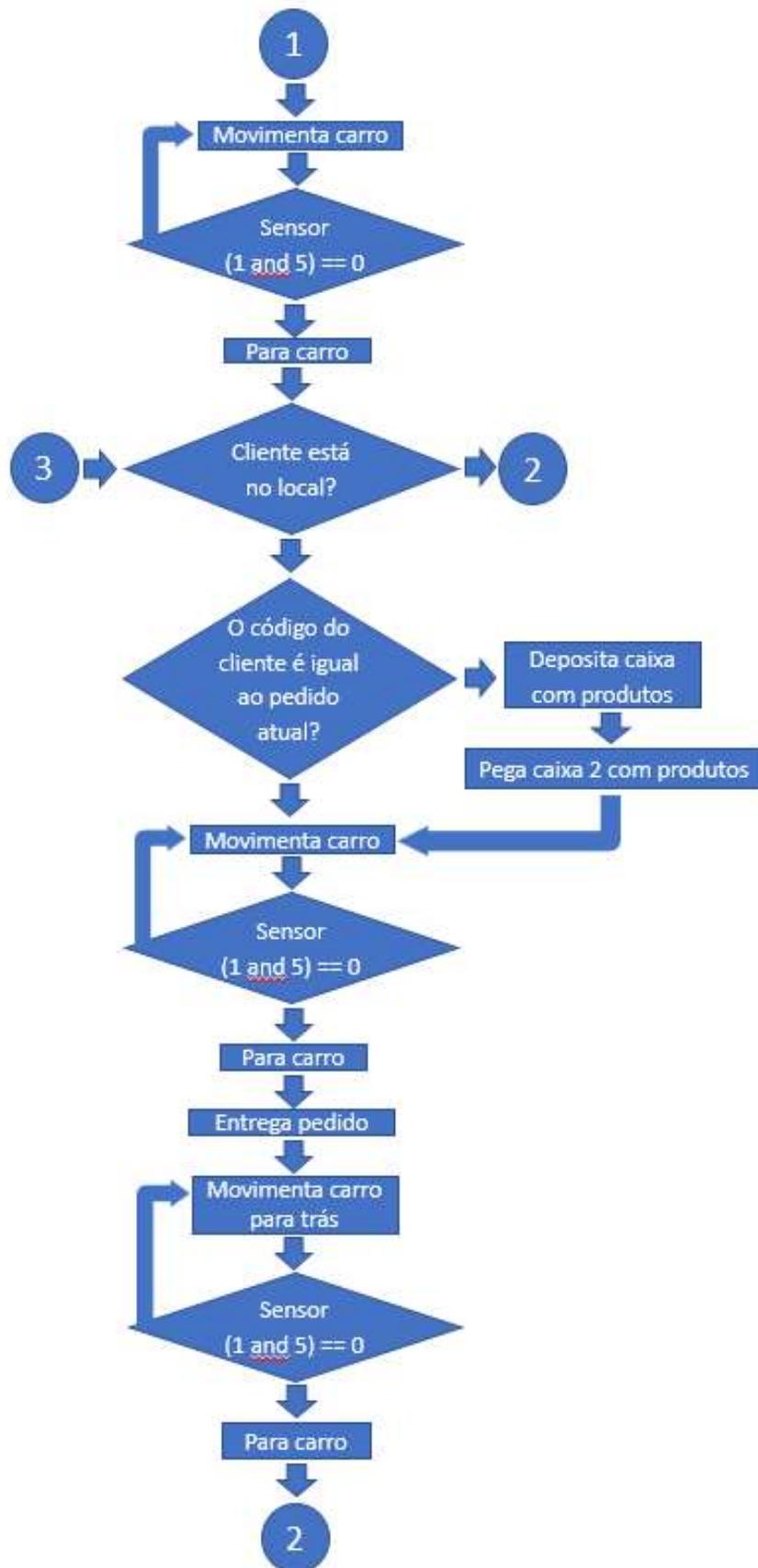
4.5. Programação da esp32

Para a programação da esp32 foi utilizado todas as bibliotecas explicadas anteriormente e diversas funções para facilitar a organização e entendimento do código, deixando apenas no programa principal (*void loop*) a chamada dessas funções, e fazendo a primeira inicialização (*void setup*) da comunicação com o *firebase*, identificando as entradas e saídas que irá ser utilizada na esp32 e a posição inicial do braço robótico.

O programa principal recebe primeiro qual cliente que fez o pedido, e armazena esse cliente em uma variável para ser utilizada futuramente, em seguida recebe e armazena os produtos que irá pegar no trajeto, começando então a se movimentar, onde quando identificado os pontos de paradas, para o movimento do carro e começa a movimentar o braço robótico, depois de deixar o determinado produto na caixa sobre o carro, ele volta a se movimentar até o próximo ponto de parada. Depois de coletar todos os produtos, ele volta a se movimentar até chegar no momento que irá entregar o pedido ou preparar um novo pedido de outro cliente novamente.

O fluxograma abaixo mostra as decisões que a célula robótica poderá tomar, dependendo das informações que tiver em determinados momentos.







4.6. Regulagem dos ângulos dos servos, comprimento de onda mínimo e máximo do PWM

Um dos ajustes que foi necessário fazer para os servos reconhecerem exatamente as posições de 0° a 180° foi a parametrização do comprimento de onda mínimo e máximo, que será utilizado no programa para controlar as posições dos servos, para isso foi utilizado o código abaixo.

```
int pulsewidth(int angle) {
  int pulse_wide, analog_value;
  pulse_wide = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  analog_value = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  return analog_value;
}

void loop() {
  // find MIN_PULSE_WIDTH: set angle to 0
  // find MAX_PULSE_WIDTH: set angle to 180
  pwm.setPWM(0,0,pulsewidth(0)); // 0 or 180
}
```

Figura 63 - Função para parametrizar os servos (Fonte: <https://diyi0t.com>)

O código utiliza uma função para pegar o ângulo que se deseja obter (marcado em vermelho), esse valor então é utilizado na função *map* da biblioteca do *Arduino*, que pega o valor escolhido, e irá fazer um cálculo, utilizando o intervalo de 0 a 180 e os valores de mínimo e máximo de comprimento de onda (MIN_PULSE_WIDTH, MAX_PULSE_WIDTH), esse resultado então é utilizado em uma fórmula, que irá pegar esse resultado que se obteve na função *map* e dividir por $1 \cdot 10^6$, para resultar o valor em segundos, multiplicado pela frequência do servo e por 4096 que é o valor em uma resolução de 12 bits que a placa PCA9685 trabalha.

Os valores de comprimento mínimo e máximo de comprimento de onda variam de acordo com o modelo de servo, fazendo com que os valores para o servo MG966R foi de MIN_PULSE_WIDTH igual a 550 e o MAX_PULSE_WIDTH de 2250, respectivamente, se encontrando nas posições 0° e 180°, como mostrado na figura.

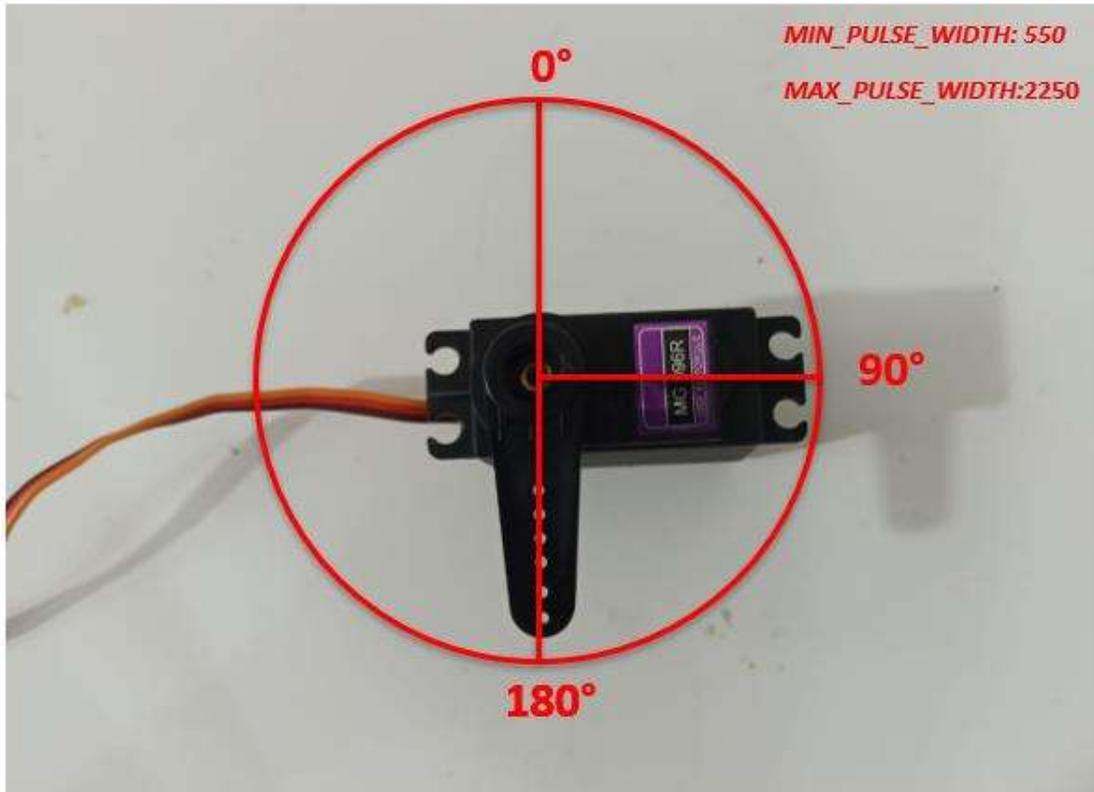


Figura 64 - Posições dos servos (Fonte: Elaborada pelo autor)

4.7. Placa para teste das posições dos servos, com potenciômetro

Foi desenvolvido uma placa de circuito impresso de *fenolite*, que constitui de 6 potenciômetros, 1 capacitor, 1 borne, barras de pinos e jumpers para fazer as ligações.

Uma esp32 foi utilizada para fazer a lógica das informações coletadas, ela é responsável por monitorar os valores analógicos que são recebidos pelos potenciômetros, e assim controlar os servos.

Nas barras de pinos foi inserido os 6 servos que serão utilizados para controlar o braço robótico, cada potenciômetro é responsável por controlar um servo, os sinais analógicos coletados pela esp32 são convertidos através da função *map* na lógica da esp32 para transformar o sinal de 0 a 1023 que é recebido pelo potenciômetro, para um ângulo de 0° a 180°, que corresponde aos valores que serão enviados para os servos, para enviar os sinais para os servos, foi utilizado uma biblioteca chamada `<ESP32Servo.h>` que contém funções necessárias para configurar e controlar os servos.

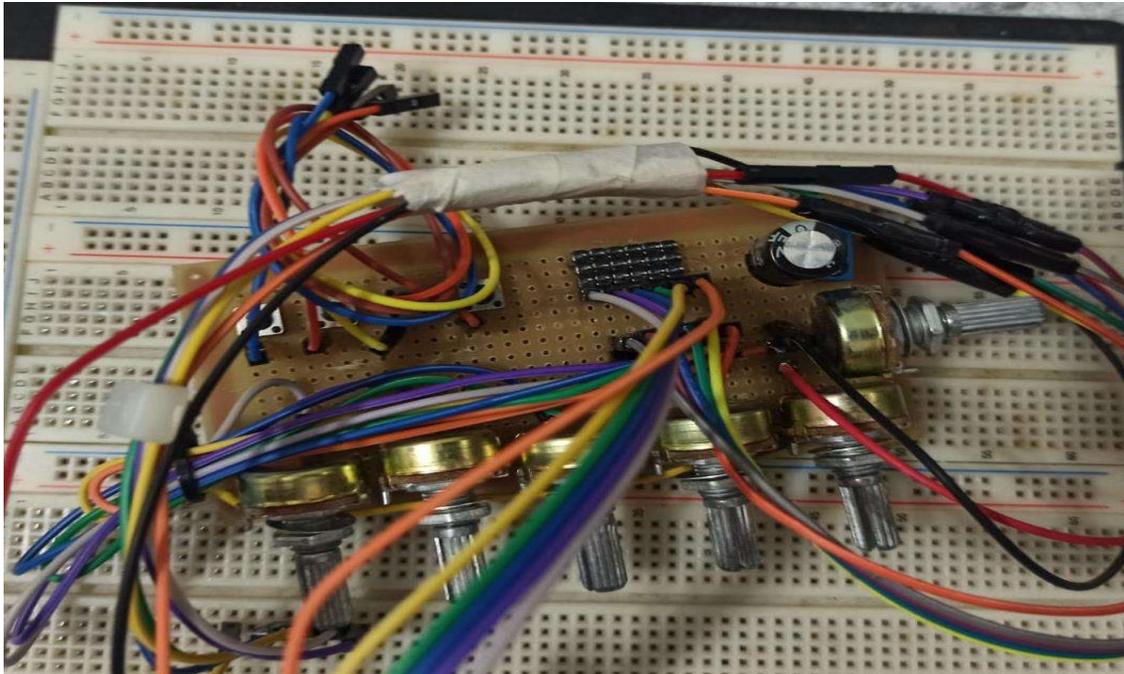


Figura 65 - Placa desenvolvida para regular os ângulos dos servos (Fonte: Elaborada pelo autor)

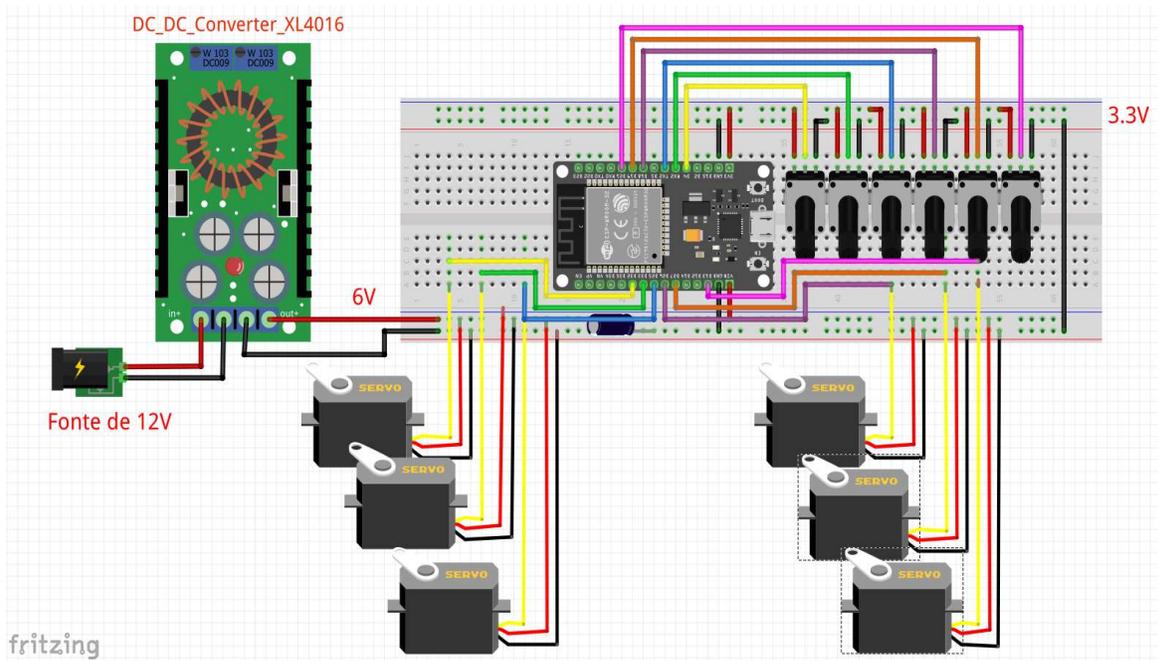


Figura 66 - Esquema elétrico da placa de circuito impresso e suas ligações (Fonte: Elaborada pelo autor com ajuda do *software fritzing*)

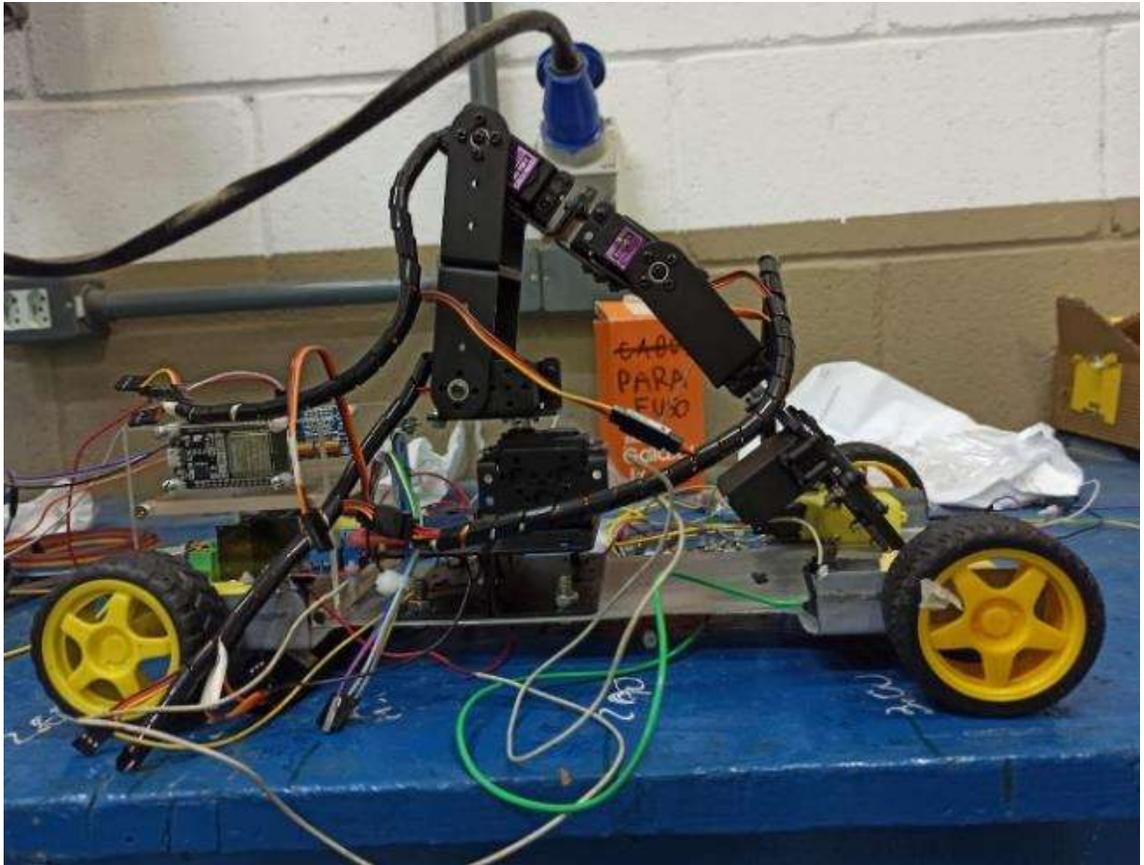


Figura 67 - Montagem do circuito elétrico na célula robótica (Fonte: Elaborado pelo autor)

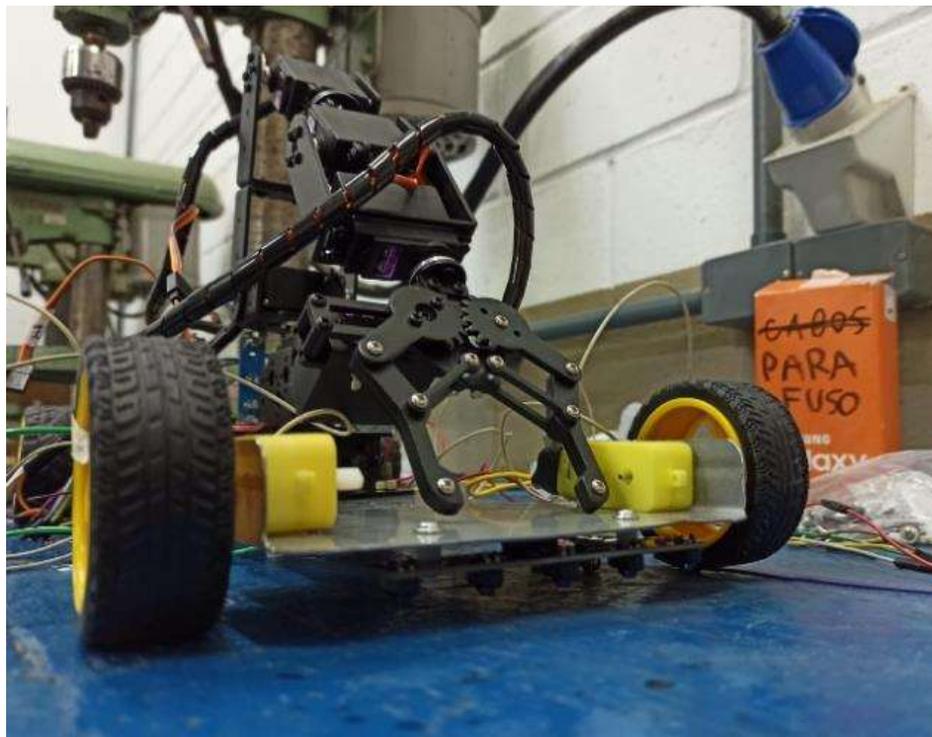


Figura 68 - Célula robótica: Garra (Fonte: Elaborado pelo autor)

4.8. Circuito que a célula robótica irá fazer

Os primeiros testes foram feitos utilizando o monitor serial do Arduino, para checar se as decisões dentro da programação estavam sendo feitas corretamente de acordo com as possibilidades de informações que iria ser recebido.

Os servos foram testados individualmente para garantir se estavam se movimentando totalmente e depois integrados na placa PCA9685, e testados novamente para garantir sua funcionalidade.

Os motores DC foram testados individualmente alimentando com uma tensão de 6V para garantir seu funcionamento e depois integrados com a placa L298N para ser possível controlar o sentido de rotação e a velocidade dos motores.

Os sensores foram alimentados com a alimentação de saída da esp32 de 3.3V e testados, para ver se todos os sensores estavam mandando os sinais de ligado e desligado quando estavam sendo acionados, esses valores foram monitorados pelo monitor serial da IDE do Arduino.

Após os testes individuais de cada componente, foi feito a integração entre todos os módulos, conectando a alimentação das placas através de uma bateria de lítio e um regulador de tensão, que garantia a saída de 6V para todo o circuito, todos os cabos foram conectados nas suas respectivas entradas e saídas da esp32 e a placa PCA9685 conectada através da comunicação I2C com a esp32 nos pinos 21 e 22 da placa.

Em seguida foi testado o código principal para ver se o carro estava se movimentando sobre a faixa e parando nas posições desejadas.

Depois foi feito teste no braço robótico, ajustando os movimentos que não estavam muito precisos, e testando se o robô estava pegando as peças e depositando sobre a caixa corretamente, e por último foi feito o teste de todo o conjunto, observando se a célula estava fazendo as lógicas e tomando as decisões de acordo com as informações que eram recebidas, seguindo a lógica do fluxograma desenvolvido.



Figura 69 - Circuito (Fonte: Elaborado pelo autor)



Figura 70 - Prateleira (Fonte: Elaborado pelo autor)

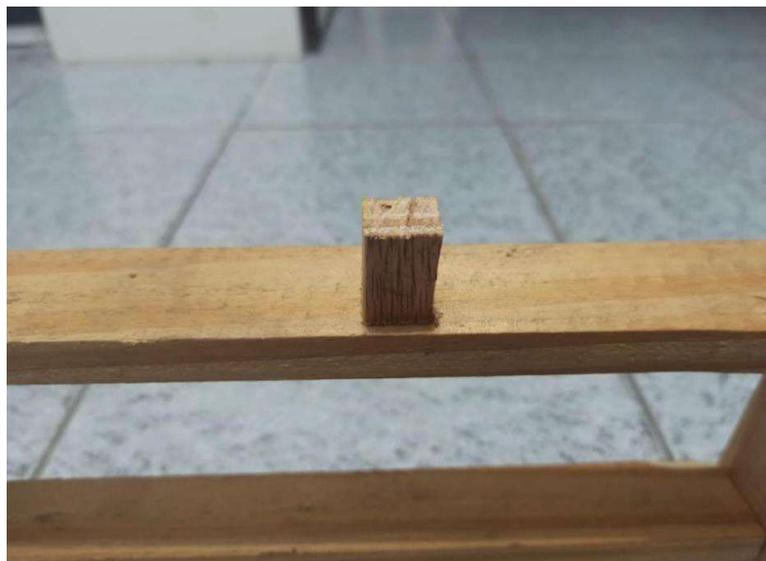


Figura 71 - Produto (Fonte: Elaborado pelo autor)

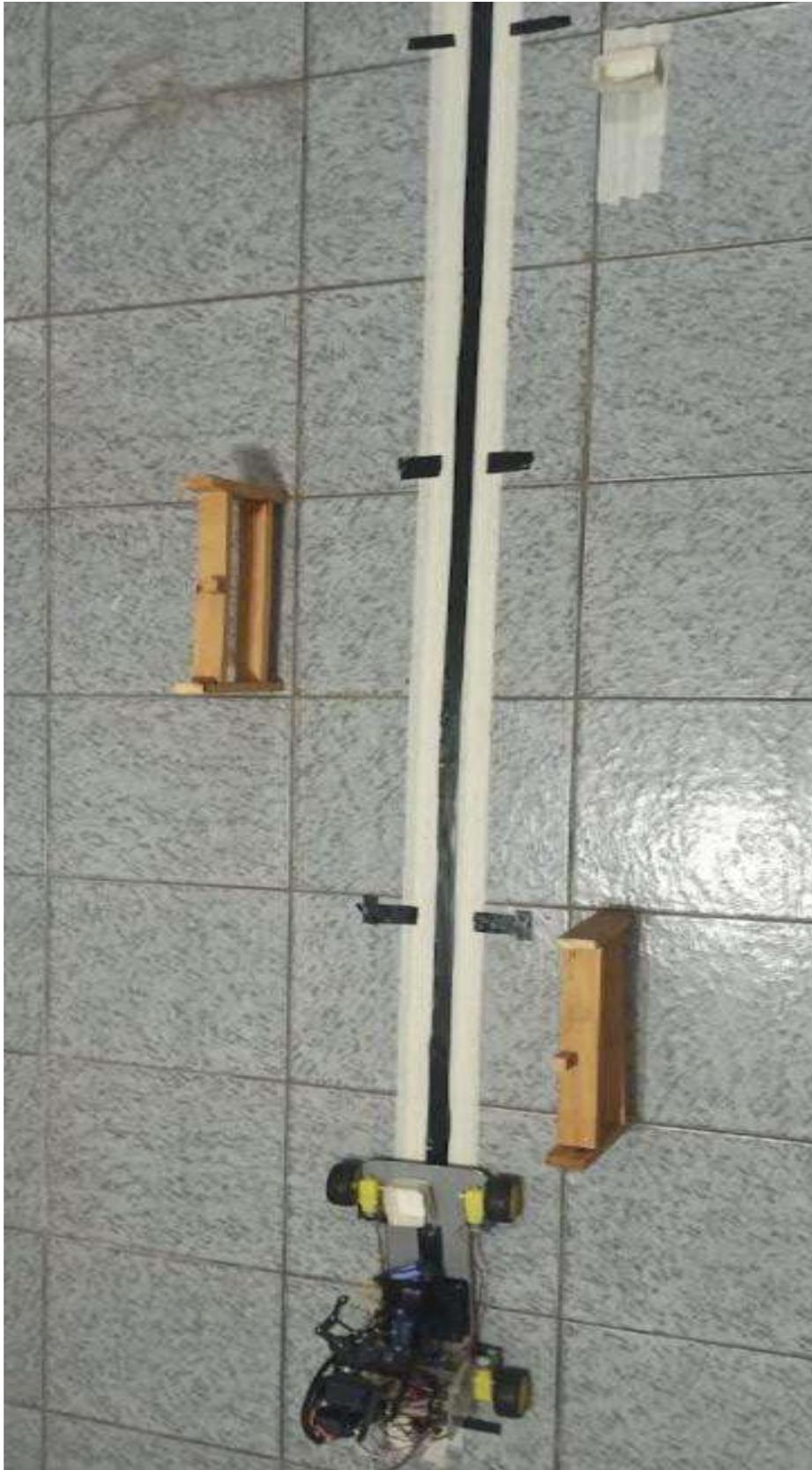
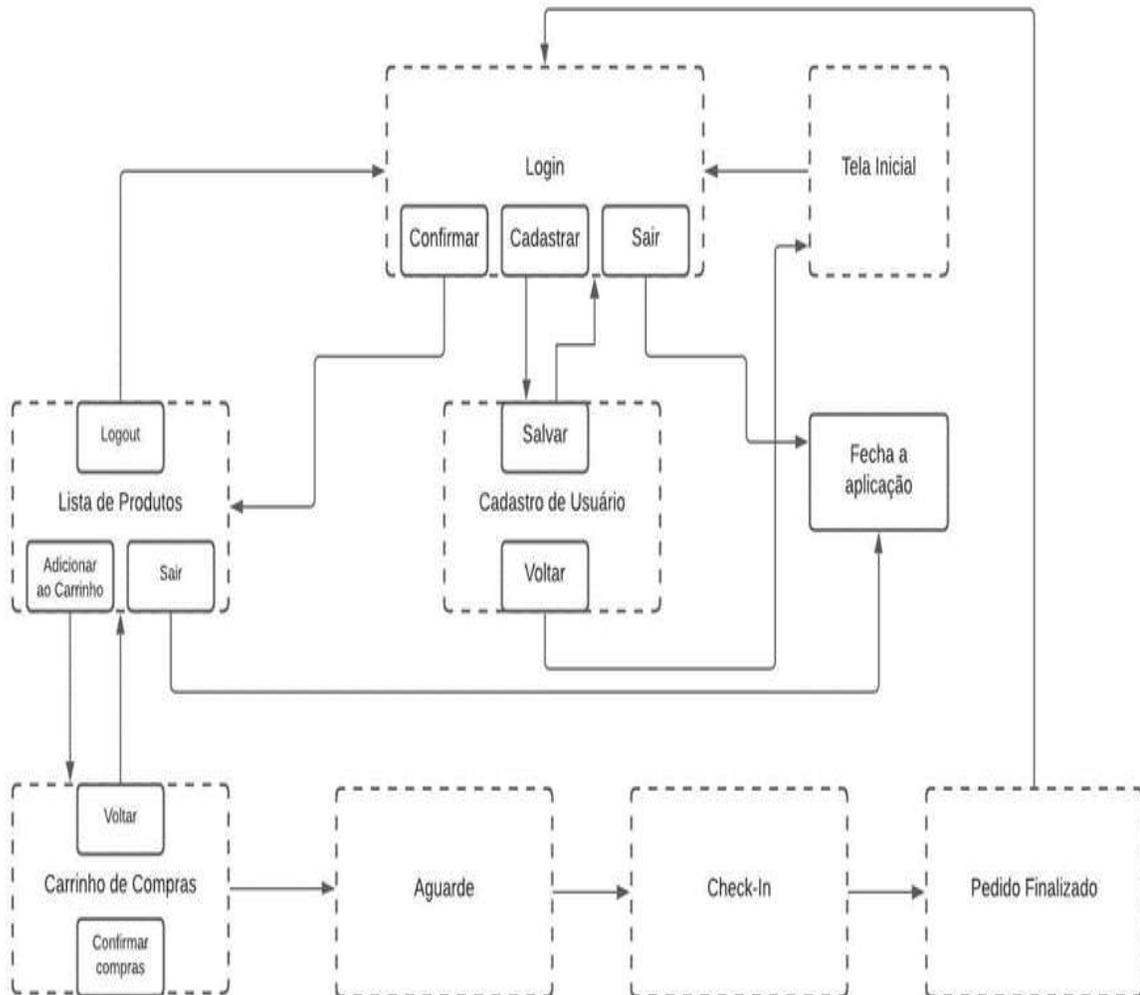


Figura 72 - Célula no ponto A do circuito (Fonte: Elaborado pelo autor)

4.9. Desenvolvimento do Aplicativo

O aplicativo foi desenvolvido com base no seguinte fluxograma:



Para estabelecer comunicação entre o aplicativo e o microcontrolador, foi utilizado a plataforma de banco de dados em nuvem Google *Firebase*. O *Firebase* recebe as informações do pedido de compra realizado no aplicativo (figura 73) e redireciona os dados para que o microcontrolador possa separar os produtos. Dessa forma, não se faz necessidade de que o usuário esteja conectado na mesma rede *wi-Fi* do microcontrolador.

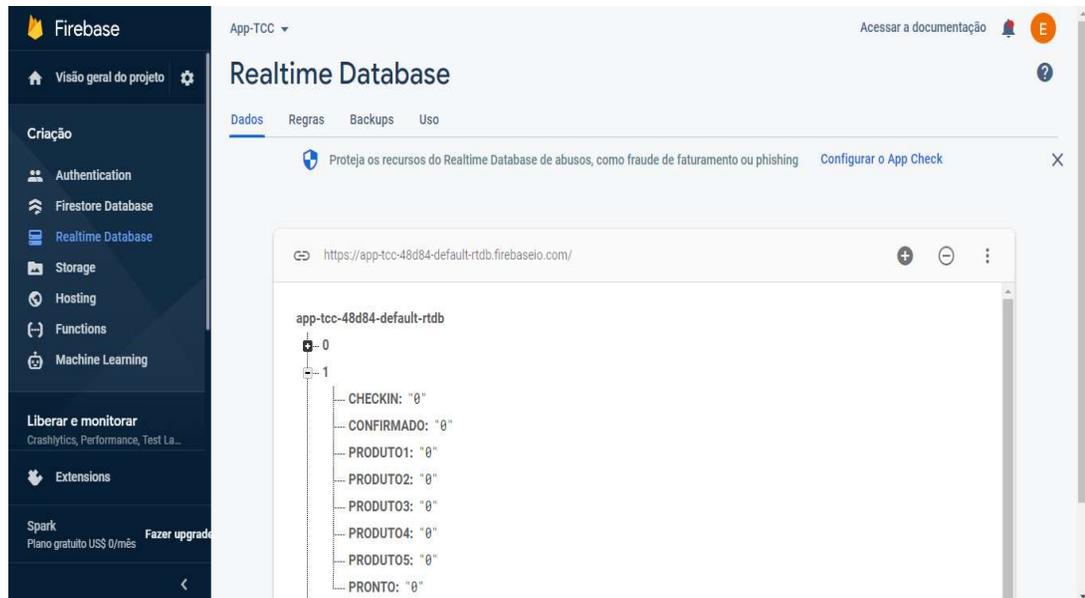


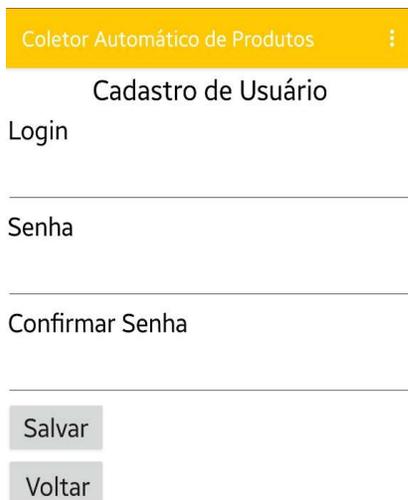
Figura 73 - Google *Firebase* (Fonte: Elaborada pelo autor)

Após a concepção do modelo de funcionamento do aplicativo, foi criada uma aplicação para realização de pedidos de compra. A figura 73 representa a tela inicial do aplicativo, que serve como apresentação do sistema.



Figura 74 – Tela inicial do aplicativo (Fonte: Elaborado pelo autor)

Depois de 5 segundos que o aplicativo foi aberto, automaticamente será aberta a página para que o cliente realize o seu Login na aplicação (Figura 75). Aqui o usuário cadastrado deverá colocar o seu nome de usuário e a senha nos campos indicados e, em seguida, selecionar o botão “Confirmar”, para dar continuidade ao processo.



Coletor Automático de Produtos

Cadastro de Usuário

Login

Senha

Confirmar Senha

Salvar

Voltar

Figura 75 - Tela de login (Fonte: Elaborado pelo autor)

Caso a pessoa não possua um cadastro, basta pressionar o botão “Cadastrar”, onde será aberta a tela de Cadastro (Figura 76). Para realizar o cadastramento, deve-se inserir um login, senha e pressionar o botão “Salvar”. Ao concluir, o aplicativo retorna a página de login.



Figura 76 - Tela de cadastro (Fonte: Elaborado pelo autor)

Em sequência do login, é possível visualizar os produtos disponíveis para compra (Figura 77). Basta o cliente escolher qual produto desejar, sua quantidade e pressionar o botão “Adicionar ao carrinho”.



Figura 77 – Tela: Lista dos produtos (Fonte: Elaborado pelo autor)

Será solicitada uma confirmação, onde em seguida será aberta a página do carrinho, onde consta os produtos escolhidos e as quantidades referentes (Figura 78).

Carrinho

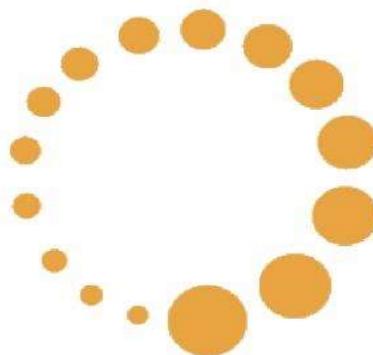
Produto	Quantidade
 Produto 1	1
 Produto 2	1
 Produto 3	1
 Produto 4	1
 Produto 5	1

Confirmar Compra

Voltar

Figura 78 – Tela: Lista dos produtos selecionados (Fonte: Elaborado pelo autor)

Ao pressionar o botão de confirmação da compra, irá ser informado que o pedido está em separação (Figura 79), bastando o cliente aguardar o término do procedimento pelo mercado.



Aguarde, seu pedido está sendo separado...

Figura 79 – Tela pedido finalizado (Fonte: Elaborado pelo autor)

Assim que o mercado finaliza a separação dos produtos, o cliente é informado que já pode se encaminhar ao local para retirar a compra.

Ao chegar ao estabelecimento, o usuário deve realizar um check-in, para o mercado liberar a encomenda (Figura 80).

Coletor Automático de Produtos



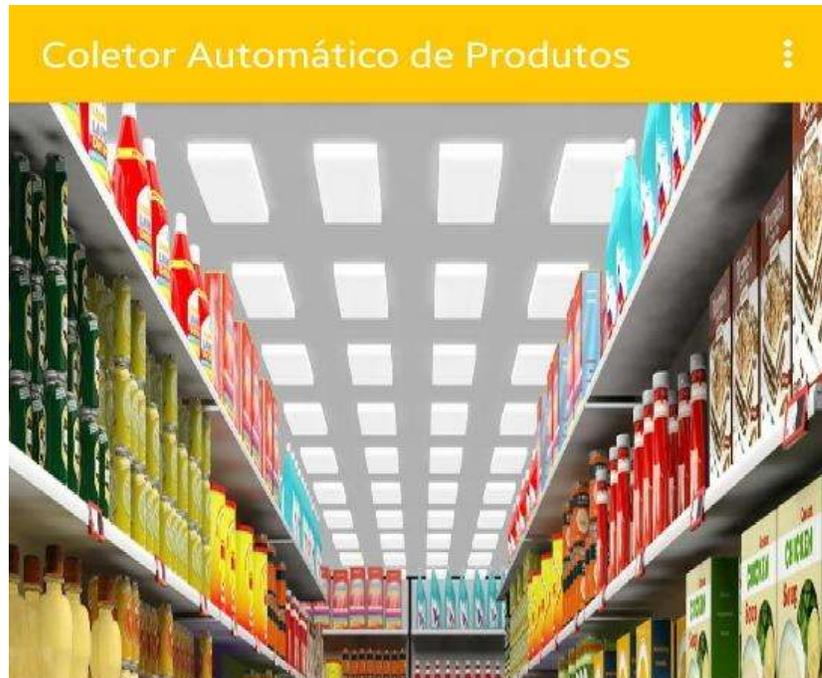
Clique no botão Check-In para retirar seu produto.



Clique apenas se já estiver no local da retirada.

Figura 80 – Tela: *Check in* (Fonte: Elaborado pelo autor)

Após o usuário confirmar que retirou o pedido, a seguinte página será aberta (Figura 81), finalizando assim o processo de compra.



Pedido finalizado!
Agradecemos a preferência!
Tenha um bom dia!!!

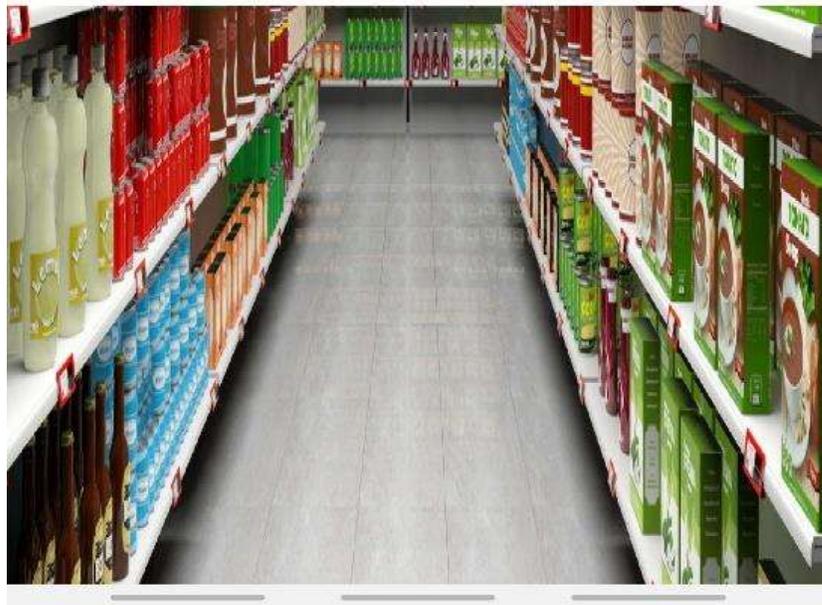


Figura 81 - Tela: Pedido finalizado (Fonte: Elaborado pelo autor)

5. Conclusões

Através da estrutura lógica desenvolvida foi possível realizar a movimentação da célula robótica e a separação e entrega de pedidos realizados via aplicativo.

O robô autônomo é um sistema de coleta e transporte de produtos que capaz de receber as informações do aplicativo via comunicação *wi-fi* com a esp32, tratar e armazenar de forma a realizar as movimentações, separações e entregas de acordo com o pedido desejado do cliente.

Já na concepção do aplicativo, foi utilizado a plataforma MIT App Inventor, pois apesar de ser um site voltado para trabalhos didáticos, atendeu perfeitamente ao que estávamos propondo para o funcionamento do aplicativo.

Com isso, foi realizado a comunicação do aplicativo com o microcontrolador ESP32 instalado na célula robótica, porém para facilitar a utilização do aplicativo em qualquer lugar, foi utilizado um banco de dados em nuvem, o app inventor que possui em sua biblioteca uma ferramenta específica para comunicação com o banco de dados Google Firebase, o que permitiu, com poucas configurações estabelecer a comunicação com a ESP32 de qualquer lugar, desde que o celular esteja conectado com a internet.

Também foi possível definir a confirmação por parte da célula robótica que o pedido estava separado e pronto para retirada, além de conclusão do pedido realizado pelo aplicativo.

Referências

ABRAS F. **Cinquenta Anos de supermercado no Brasil**. São Paulo: Fundação ABRAS. 2002.

AGUIAR, Gabriel. **Conheça a lanchonete fast-food que trocou funcionários por robôs no Brasil**. 2021. Disponível em: <<https://exame.com>>. Acesso em 9 mai 2021.

ALBUQUERQUE, M. P. de C. **Análise da evolução do setor supermercadista brasileiro: uma visão estratégica**. 2007. Dissertação (Mestrado profissionalizante em administração). Faculdade de economia e finanças IBMEC, Rio de Janeiro, 2007.

Are High Voltage LiPo (LiHV) Batteries Worth it? Disponível em: <<https://www.radiocontrolinfo.com/are-high-voltage-lipo-lihv-batteries-worth-it/>>. Acesso em 20 maio 2021.

Athos electronics. **ESP32 – Especificações e projetos**. 2019. Disponível em: <<https://athoselectronics.com>>. Acesso em 05 jun 2021

Baterias de Lítio-Íon: Um guia completo. Disponível em: <<https://www.embarcados.com.br/baterias-de-ion-um-guia-completo>>. Acesso em 4 maio 2021.

BIONICOOK. **O futuro chegou**. 2019. Disponível em: <<https://www.bionicook.com/home>>. Acesso em: 9 mai 2021.

CAMAROTTO, M. R. **Estratégia de Marketing**. 1. ed. Curitiba: IESDE, 2009.

CHIAVENATO, I. **Administração: teoria, processo e prática**. São Paulo: Makron Books. 3. ed. 2000

CHIAVENATO, Idalberto. **Gestão de Pessoas**. 3. ed. São Paulo: Elsevier, 2008.

CURVELLO, André. **ESP32 – Um grande aliado para o Maker IoT**. 2018. Disponível em: <<https://www.filipeflop.com/>>. Acesso em 06 jun 2021

EletronJun. . **O que são microcontroladores? Descubra suas aplicações**. 2020. Disponível em: <<https://eletronjun.com.br/>>. Acesso em 06 jun 2021

Empresa Zaitt. Disponível em: <<https://www.zaitt.com.br/>>. Acesso em: 09 de mai. 2021

EXAME. **Nem drive-thru e delivery salvam resultado do McDonald's e lucro cai 68%. 2020.** Disponível em: <<https://exame.com>>. Acesso em 9 mai 2021.

Firestore (2018a). **Documentação do firestore.** Disponível em: <<https://firebase.google.com/docs/>>. Acessado em 03/06/2021.

Gomes, T. S. G. e Melo, J. C. B. de (2013) **App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação.** Em: Anais do Congresso Brasileiro de Informática na Educação - CBIE, Campinas, Brasil.

GS1 Brasil - **Associação Brasileira de Automação.** Disponível em: <<https://www.gs1br.org/>>. Acesso em: 07 de mai. 2021.

HEROLD, Tracy Stapp. **The Top Food Franchises of 2020.** 2020. Disponível em: <<https://www.entrepreneur.com/>>. Acesso em 9 mai. 2021.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Pesquisa Nacional por Amostra de Domicílios Contínua:** Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal, 2019.

IEEE. **O Que É Um Microcontrolador.** 2020. Disponível em: <<https://edu.ieee.org>>. Acesso em 30 mai 2021

ISTO É DINHEIRO. **Rede de fast food 100% robotizada chega ao Brasil.** 2021. Disponível em: <<https://www.istoedinheiro.com.br>>. Acesso em 9 mai 2021.

MEIRELLES, F. S. **Pesquisa Anual do Uso de TI nas Empresas,** FGV cia: Centro de Tecnologia de Informação Aplicada da EAESP, 31ª edição, 2020.

MIOLO, Michel Miola et al. **Desenvolvimento de componentes para App Inventor e servidor para dispositivos IoT.** 2019.

Motor CC: Saiba como Funciona e de que forma Especificar. Disponível em: <<https://www.citisystems.com.br/motor-cc/>> Acesso em 05 maio 2021.

Notícia do projeto da empresa zaitt. Disponível em: <<https://www.consumidormoderno.com.br/>>. Acesso em: 09 de mai. 2021.

Notícia do projeto da empresa zaitt. Disponível em: <<https://www.hypeness.com.br>>. Acesso em: 09 de mai. 2021

PALETTA, D. B.; MARIN, S. S. **Do armazém ao supermercado.** São Paulo: Paulus, 2014.

PASTERNAK, E. (2009). **Visual programming pedagogies and integrating current visual programming language features**. Master's thesis, Robotics Institute, Carnegie Mellon University.

REIS, Fábio. **Introdução aos Microcontroladores**. 2015. Disponível em: <<http://www.bosontreinamentos.com.br>>. Acesso em 06 jun 2021

Relatório de pesquisa de mercado. **Automated Guided Vehicle (AGV) Market**. 2020. <<https://www.marketsandmarkets.com/PressReleases/automated-guided-vehicle.asp>>. Acesso em 5 jun. 2021.

Robô KMR iiwa da empresa KUKA. Disponível em:<<https://www.kuka.com>>. Acesso em: 05 de mai. 2021.

RODRIGUES, Pedro Henrique Barbosa et al. **Comparação entre ferramentas de ensino de programação orientada a blocos**. MIT APP inventor vs SCRATCH. 2017.

SANTOS, Thiago dos. **A pandemia do COVID-19 no Brasil e seus reflexos na economia compartilhada**. 2021. 69f. Dissertação (Programa de Pós-Graduação em Linguagens, Mídia e Arte) - Pontifícia Universidade Católica de Campinas, Campinas-SP.

SCHILDT, **Herbert**. **C completo e total**. Makron, 1997.

SUA FRANQUIA. **Conheça uma das primeiras redes de franquias fast food robotizada do mundo**. 2019. Disponível em: <<https://www.suafranquia.com/>>. Acesso em 9 mai 2021.

TAKIY, Nelson Hitoshi. **Desenvolvimento de software aplicativo para controle de custos por atividades identificáveis nos processos de fabricação**. 2006. ix, 81 f. Dissertação (mestrado) - Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, 2006.

YAGHOUBI, S. et al. **Designing and Methodology of Automated Guided Vehicle Robots/ Self Guided Vehicle Systems, Future Trends**. International Journal of Research and Reviews in Applied Sciences, Ahar, Outubro 2012. 6.

PEREIRA, Fábio. **Microcontroladores PIC: programação em C**. Saraiva Educação SA, 2007.

SHHEIBIA, Tarig Ali Abdurahman El. **Controle de uma braço robótico utilizando uma abordagem de agente inteligente**. 2001. Monografia (Programa de Pós-Graduação em informática) - Universidade Federal da Paraíba, Campina Grande - PB.

SKELLY, Lupin e Sidesm Rodney R. **O futuro do varejo está nas mãos dos consumidores.** 2021. Disponível em: < <https://www2.deloitte.com/>>. Acesso em 28 set 2021.

RABELO, Agnes. **Transformação Digital: o que é e quais os seus impactos na sociedade.** 2020. Disponível em: < <https://www.rockcontent.com.com/>>. Acesso em 28 set 2021.

Portal da Industria. **Indústria 4.0: Entenda seus conceitos e fundamentos.** 2020. Disponível em: < <https://www.portaldaindustria.com.br/>>. Acesso em 28 set 2021.

TREJO, Araceli Rendón; ALQUICIRA, Andrés Morales; MONDRAGÓN, Irene Juana Guillén. **Indústria 4.0 e a indústria de alimentos.** 2020. Disponível em: <<https://www.riico.net/>>. Acesso em 02 out 2021.

TELES, Jhonata. **Indústria 4.0 – Tudo que você precisa saber sobre a Quarta Revolução Industrial.** 2020. Disponível em: < <https://www.engeteles.com.br/>>. Acesso em 02 out 2021.

ABRAS. **Inteligência Artificial muda compra de alimentos em supermercados.** 2021. Disponível em: < <https://www.abras.com.br/>>. Acesso em 02 out 2021.

VERDON, Joan. **Varejistas apostam em robotização para resolver problemas de estoque.** 2020. Disponível em: < <https://forbes.com.br/>>. Acesso em 02 out 2021.

SILVEIRA, Cristiano Bertulucci. **Os 6 Principais tipos de robôs industriais.** 2019. Disponível em: < <https://forbes.com.br/>>. Acesso em 02 out 2021.

FERREIRA, Erick da Penha; ALVES, Nicholas de Lacerda Alcides. **Braço articulado com controle proporcional de movimento comandado via Bluetooth por um aplicativo desenvolvido para plataformas Android.** 2013. Monografia (Faculdade de engenharia, arquitetura e urbanismo) – Universidade do Vale do Paraíba, São José dos Campos - SP.

SCHIAVONI, Eduardo. **Brasileiro cria braço mecânico capaz de captar e imitar movimentos.** 2015. Disponível em: < <https://www.uol.com.br/>>. Acesso em 09 out 2021.

MECALUX. **O braço robótico industrial ganha espaço no armazém.** 2021. Disponível em: < <https://www.mecalux.com.br/>>. Acesso em 09 out 2021.

KATO, Lucas Kenzo. **Projeto d eum braço robótico para fums didáticos.** 2015. Monografia (Departamento de Automação e Sistemas) – Universidade Federal de Santa Catarina, Florianópolis - SC.

MARINO STORE. **Suporte de alumínio para servo motor Robô Humanóide.** 2021. Disponível em: < <https://www.marinostore.com/>>. Acesso em 04 nov 2021.