



**Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Engenharia Mecânica**

PROJETO DE TCC:

Desenvolvimento de um código computacional para manipulação de um braço robótico de baixo custo por captura de movimento

ALUNO:

Wilson Campos de Almeida Bisneto

PROFESSORA ORIENTADORA:

Aline Victoria Cavalcanti Pereira

**Recife - PE
2022**

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Almeida Bisneto, Wilson Campos de .

Desenvolvimento de um código computacional para manipulação de um braço robótico de baixo custo por captura de movimento / Wilson Campos de Almeida Bisneto. - Recife, 2023.

64 p. : il., tab.

Orientador(a): Aline Victoria Cavalcanti Pereira

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Mecânica - Bacharelado, 2023.

Inclui referências, apêndices.

1. Arduino. 2. Braço robótico. 3. Mecatrônica. 4. impressão 3D. 5. captura de movimento. I. Pereira, Aline Victoria Cavalcanti . (Orientação). II. Título.

620 CDD (22.ed.)



Universidade Federal de Pernambuco
Departamento de Engenharia Mecânica Centro de
Tecnologia e Geociências- CTG/EEP



**ATA DE SESSÃO DE DEFESA DE
TRABALHO DE CONCLUSÃO DE CURSO – TCC2**

Ao 06.º dia do mês de setembro do ano de dois mil e vinte e três, às 15:00 horas, de forma virtual através da plataforma google meet, reuniu-se a banca examinadora para a sessão pública de defesa do Trabalho de Conclusão de Curso em Engenharia Mecânica da Universidade Federal de Pernambuco, intitulado **Desenvolvimento de um código computacional para manipulação de um braço robótico por captura de movimento**, elaborado pelo aluno **Wilson Campos de Almeida Bisneto**, matrícula 117.724.164/10, composta pelos avaliadores Profa. **Aline Victoria Cavalcanti Pereira** (orientadora), Prof. **Pedro Manuel González Del Foyo** (avaliador) e Prof. **João Paulo Cerquinho Cajueiro** (avaliador). Após a exposição oral do trabalho, o candidato foi arguido pelos componentes da banca que em seguida reuniram-se e deliberaram pela sua aprovação, atribuindo-lhe a média 8,5, julgando-o apto() / inapto() à conclusão do curso de Engenharia Mecânica. Para constar, redigi a presente ata aprovada por todos os presentes, que vai assinada pelos membros da banca.

Orientadora: Prof. Aline Victoria Cavalcanti Pereira Nota: 8,5

Assinatura _____

Avaliador Interno: Prof. Pedro Manuel González Del Foyo Nota: 8,5

Assinatura _____

Avaliador Interno: Prof. João Paulo Cerquinho Cajueiro Nota: 8,5

Assinatura _____

Recife, 06 de setembro de 2023.

Prof. Marcus Costa de Araújo
Coordenador de Trabalho de Conclusão de curso - TCC
Curso de Graduação em Engenharia Mecânica – CTG/EEP-UFPE



Emitido em 06/09/2023

ATA DE APROVACAO Nº 421/2023 - DEPEES (11.65.61)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 06/09/2023 16:20)
ALINE VICTORIA CAVALCANTI PEREIRA
PROFESSOR DO MAGISTERIO SUPERIOR
DEPEES (11.65.61)
Matrícula: ###707#8

(Assinado digitalmente em 29/09/2023 13:09)
JOAO PAULO CERQUINHO CAJUEIRO
PROFESSOR DO MAGISTERIO SUPERIOR
DEMEC (11.65.55)
Matrícula: ###197#0

(Assinado digitalmente em 06/09/2023 19:48)
PEDRO MANUEL GONZALEZ DEL FOYO
PROFESSOR DO MAGISTERIO SUPERIOR
DEMEC (11.65.55)
Matrícula: ###212#8

Visualize o documento original em <http://sipac.ufpe.br/documentos/> informando seu número: **421**, ano: **2023**, tipo:
ATA DE APROVACAO, data de emissão: **06/09/2023** e o código de verificação: **d9399a82e3**

RESUMO

A tecnologia de impressão 3D tem se popularizado nos últimos anos, resultando em uma redução significativa nos custos de produção de equipamentos e materiais devido a lei da oferta e da demanda. Nesse contexto, este trabalho apresenta uma prova de conceito de aplicação da tecnologia na construção de um braço robótico de baixo custo com 3 graus de liberdade. O protótipo é controlado por um Arduino e utiliza uma webcam para capturar a posição do braço do usuário, replicando seus movimentos. Para sua construção, foram utilizados 3 servomotores e um Arduino nano, o código computacional foi desenvolvido utilizando bibliotecas de código aberto, o protótipo tem o custo total de R\$155,00, podendo ser reduzido ao se procurar melhores fornecedores. Devido a necessidade de desenvolver o protótipo em tempo limitado, os materiais foram adquiridos localmente, mas foi estimado que o custo seria de R\$87,00 caso os materiais fossem comprados da china. Os resultados demonstram a viabilidade da utilização de tecnologias acessíveis para a construção de braços robóticos com potencial aplicação em diversas áreas, como automação residencial, indústria e educação.

Palavras-chave: Arduino; Braço robótico; Mecatrônica; impressão 3D; captura de movimento.

ABSTRACT

3D printing technology has become popular in recent years, resulting in a significant reduction in equipment production and material costs due to the law of supply and demand. In this context, this work presents a proof of concept of technology application in the construction of a low-cost robotic arm with 3 degrees of freedom. The prototype is controlled by an Arduino and uses a webcam to capture the user's arm position, replicating the movements. For its construction, 3 servomotors and an Arduino nano were used, the computational code was developed using open source libraries, the prototype has a total cost of R\$155,00, which can be reduced when looking for better suppliers. Due to the need to develop the prototype in a limited time, the materials were purchased locally, but it was estimated that the cost would be R\$87,00 if the materials were purchased from China. The results demonstrate the viability of using accessible technologies to build robotic arms with potential application in several areas, such as home automation, industry and education.

Keywords: Arduino; Robotic arm; Mechatronics; 3D printing; motion capture.

LISTA DE FIGURAS

Figura 1. Instalações anuais de robôs por cliente	9
Figura 2. Previsão de instalação anual de robôs.....	9
Figura 3. Articulação glenoumeral.....	11
Figura 4. Manguito rotador	12
Figura 5. Cotovelo	12
Figura 6. Movimentos do punho	13
Figura 7. Braço robótico da empresa Mitsubishi	14
Figura 8. Esquema básico de um braço robótico	14
Figura 9. Tipos de articulação	15
Figura 10. Movimentos das juntas.....	16
Figura 11. Braços com um (à esquerda) e dois graus de liberdade (à direita)	16
Figura 12. Flexão/extensão do ombro	17
Figura 13. Abdução/adução do ombro	17
Figura 14. Rotação interna/externa do ombro	18
Figura 15. Flexão/extensão do cotovelo.....	18
Figura 16. Pronação/supinação do antebraço.....	18
Figura 17. Flexão/extensão do punho	19
Figura 18. Classificação de braço robótico: (a) cartesiano, (b) polar, (c) cilíndrico, (d) articulado, (e) Scara	20
Figura 19. Protótipo finalizado.....	23
Figura 20. Partes do protótipo desmontado	23
Figura 21. Junta do protótipo.....	24
Figura 22. Conexões elétricas.....	24
Figura 23. Arduino NANO.....	25
Figura 24. Fluxograma do código computacional.....	26
Figura 25. Diagrama do sinal	27
Figura 26. Pontos no braço	28
Figura 27. Ângulo de flexão/extensão do cotovelo	28
Figura 28. Pontos e vetores no corpo	29
Figura 29. Vetor altura, vetor ombro e ângulo de abdução/adução do ombro	30

Figura 30. Vetor projeção do braço e vista lateral	31
Figura 31. Vista frontal e Vista superior	32
Figura 32. Mecanismo do braço	33
Figura 33. Ângulos do mecanismo parte 1	34
Figura 34. Ângulos do mecanismo parte 2	35
Figura 35. Visão computacional.	37
Figura 36. Motor encaixado na base com furo ajustado.....	38
Figura 37. Medições dos ângulos.....	39
Figura 38.Diferença do ângulo no experimento 1 em posição aleatória.....	40
Figura 39.Diferença do ângulo no experimento 1 de frente	41
Figura 40.Diferença do ângulo no experimento 1 de lado	41

LISTA DE TABELAS

Tabela 1. Exemplo de comando.....	25
Tabela 2. Custos	39
Tabela 3. Medições dos ângulos de rotação no experimento 1.	41
Tabela 4. Medições dos ângulos de adução no experimento 1.....	42
Tabela 5. Medições dos ângulos do cotovelo no experimento 1.....	42
Tabela 6. Média de erro e desvio padrão dos resultados do experimento 1.	44
Tabela 7. Erro na variação da posição relativa à câmera no experimento 2, medição 1.	45
Tabela 8 .Erro na variação da posição relativa à câmera no experimento 2, medição 2.	45
Tabela 9. Erro na variação da posição relativa à câmera no experimento 2, medição 3.	46

SUMÁRIO

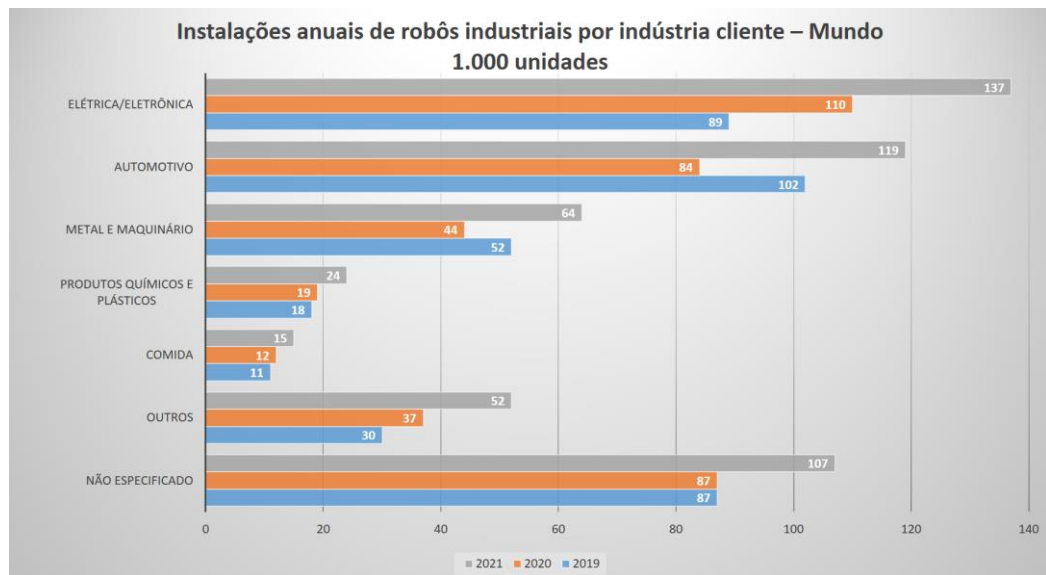
1	INTRODUÇÃO	9
1.1	OBJETIVOS	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	BIOMECÂNICA DO BRAÇO	11
2.1.1	O OMBRO	11
2.1.2	O COTOVELO.....	12
2.1.3	O PUNHO.....	12
2.2	FUNDAMENTOS DE BRAÇOS ROBÓTICOS	13
2.2.1	BRAÇO ROBÓTICO	13
2.2.2	ANATOMIA DOS BRAÇOS MECÂNICOS	13
2.2.3	GRAUS DE LIBERDADE.....	16
2.2.4	CLASSIFICAÇÃO DE BRAÇOS ROBÓTICOS	19
2.3	CAPTURE DE MOVIMENTO DO BRAÇO HUMANO.....	20
2.3.1	VISÃO COMPUTACIONAL.....	20
2.3.2	BIBLIOTECA OPENCV.....	20
2.3.3	BIBLIOTECA MEDIAPIPE	21
2.4	CIRCUITO	21
2.4.1	ARDUINO.....	21
2.4.2	SERVOMOTORES.....	21
2.4.3	COMUNICAÇÃO SERIAL.....	22
3	METODOLOGIA.....	23
3.1	CONSTRUÇÃO DO PROTÓTIPO DE BRAÇO ROBÓTICO	23
3.1.1	PARTE MECANICA	23
3.1.2	PARTE ELETRICA.....	24
3.2	CÓDIGO COMPUTACIONAL PARA ACIONAMENTO DOS MOTORES DO PROTÓTIPO DE BRAÇO ROBÓTICO.	25
3.3	CÓDIGO COMPUTACIONAL QUE IDENTIFICA A POSIÇÃO DO BRAÇO HUMANO 25	
3.4	ÂNGULOS	27
3.4.1	Ângulo de flexão/extensão do cotovelo	27

3.4.2	Ângulo de abdução/adução do ombro	29
3.4.3	Ângulo de rotação do ombro.....	31
3.5	CINEMÁTICA DO BRAÇO ROBÓTICO	32
4	RESULTADOS	37
5	CONCLUSÃO.....	47
5.1	TRABALHOS FUTUROS.....	48
	REFERÊNCIAS BIBLIOGRÁFICAS	49
	APÊNDICE A – Código do Arduino que move os motores.....	51
	APÊNDICE B – Código que comanda o braço.....	55

1 INTRODUÇÃO

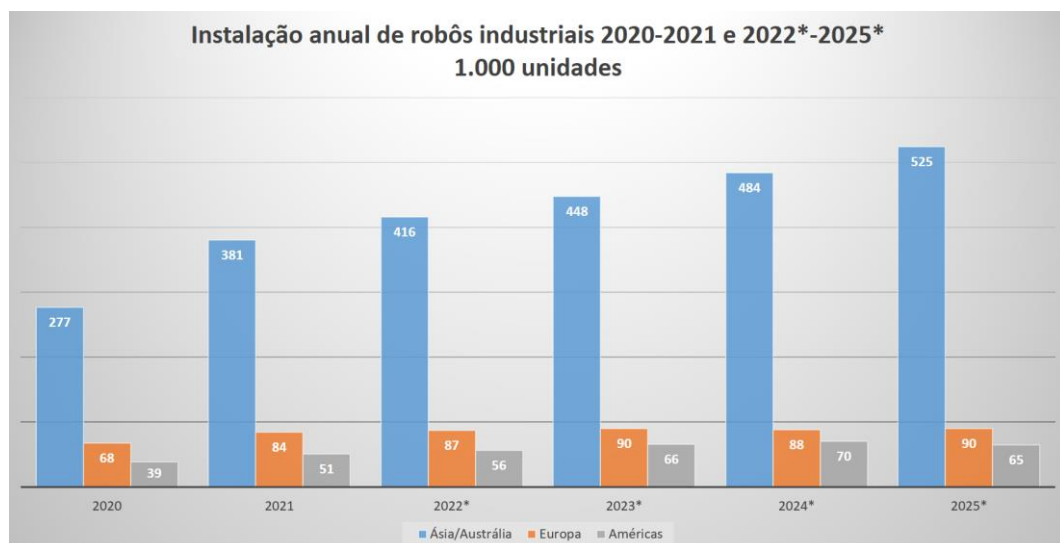
Segundo os dados da IFR (*International Federation of Robotics*), a tendência das indústrias é de automatizar seus processos cada vez mais. O ano de 2021 teve 31% de aumento nas vendas de robôs em comparação ao ano anterior. Na Figura 1, nota-se que em todo mundo, foram vendidas mais de meio milhão de unidades, especialmente nas indústrias automobilísticas e eletrônica. Nos seguintes anos, é esperado crescimento nas vendas devido à continua demanda industrial como mostra a Figura 2. (IFR, 2022).

Figura 1. Instalações anuais de robôs por cliente



Fonte: Adaptado de IFR, 2022.

Figura 2. Previsão de instalação anual de robôs



Fonte: adaptado de IFR, 2022.

Tamanho avanço da automação industrial permitiu o uso de braços robóticos para tarefas repetitivas e precisas como montagem de veículos, entretanto esta tecnologia é pouco explorada num ambiente doméstico devido ao elevado custo de um braço robótico. Contudo, a presença de *low cost robotics* e a democratização de impressão 3D tornou essa tecnologia mais acessível ao público (ITMAN, 2011).

Apesar de ser amplamente utilizada no ambiente industrial, o uso dessas tecnologias em ambientes residenciais ainda é reduzido. Estando presente, em geral, nas residências de pessoas de elevado poder aquisitivo, já as residências de pessoas de baixa renda acabam não tendo acesso a tal tecnologia devido ao seu custo (MOURA, 2020).

A capacidade de controlar robôs de forma natural e intuitiva permite aplicações em diversas áreas e reduz a barreira de entrada para usuários menos experientes. Se é possível fazer um braço robótico identificar e replicar os movimentos do braço humano, com alguns sensores e câmeras é possível programa-lo para fazer outras atividades a partir de gestos, seja para automação de processos industriais, ou até mesmo para automação de tarefas domésticas mais simples (SILVA, 2011). Ainda, a utilização de materiais mais acessíveis ao público possibilita a redução de custo para alcançar cada vez mais residências e empresas (CORRALLO, 2015).

Neste trabalho, será desenvolvido um código para controlar um braço robótico utilizando a captura de movimento, permitindo que o robô execute tarefas de forma intuitiva. A montagem é realizada com câmeras comuns para enviar instruções para um braço robótico cuja fabricação foi realizada através de prototipagem rápida de impressão 3D, movido por servomotores e controlado por um microcontrolador de um Arduino *UNO*.

1.1 OBJETIVOS

O objetivo geral deste trabalho será desenvolver um código computacional para um protótipo de braço robótico para que ele replique os movimentos de um braço humano.

Para isso, como seus objetivos específicos podem-se citar:

- Imprimir e montar o protótipo de braço robótico;
- Elaborar código computacional para movimentação do protótipo de braço;
- Elaborar código computacional para análise da posição atual do braço humano e para enviar instruções para o protótipo de braço.

2 FUNDAMENTAÇÃO TEÓRICA

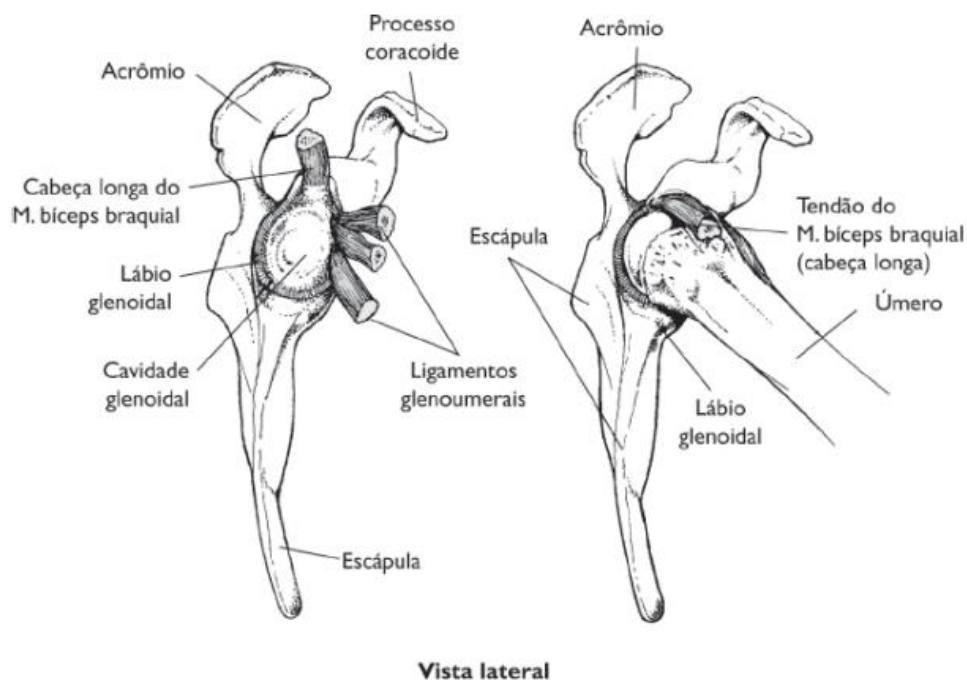
2.1 BIOMECÂNICA DO BRAÇO

A biomecânica do braço humano estuda os movimentos e as forças que atuam no braço e no ombro. Isso inclui a análise dos músculos, tendões, ossos e articulações que compõem o braço humano (HALL, 2016). A análise dos principais movimentos do braço se faz necessária para replicá-los mecanicamente com o protótipo 3D através do código desenvolvido. Para tal as duas principais estruturas são: ombro e cotovelo.

2.1.1 O OMBRO

O ombro é uma das articulações mais complexas do corpo humano devido à combinação de cinco articulações diferentes. A articulação glenoumeral, ilustrada na Figura 3, é referida como a principal articulação do ombro pois permite a mobilidade para a clavícula e a escápula. (HALL, 2016).

Figura 3. Articulação glenoumeral



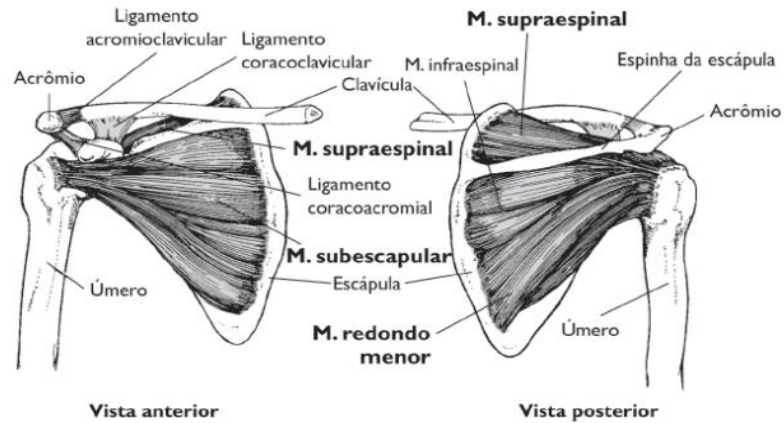
Fonte: HALL, 2016.

Essa é a articulação que produz mais movimentos em todo o corpo humano. É possível realizar flexão, extensão, abdução, adução, rotação interna e rotação externa, estes 6 movimentos são na verdade 3 pares de movimentos onde muda o sentido do movimento. A flexão, extensão e rotação do ombro são definidas pelo conjunto de músculos deltoides. Já na rotação ativa do braço, as forças musculares tendem a limitar as amplitudes de movimento do ombro, reduzindo a translação umeral. (HALL, 2016).

Outra estrutura importante na biomecânica do ombro é o conjunto de tendões com quatro músculos conhecido como manguito rotador, apresentados na Figura 4. Esses

músculos contribuem para a rotação do úmero. Além disso, a estabilidade do ombro é reforçada pelo manguito rotador cuja contração puxa a cabeça do úmero na direção da cavidade glenoidal. (HALL, 2016).

Figura 4. Manguito rotador

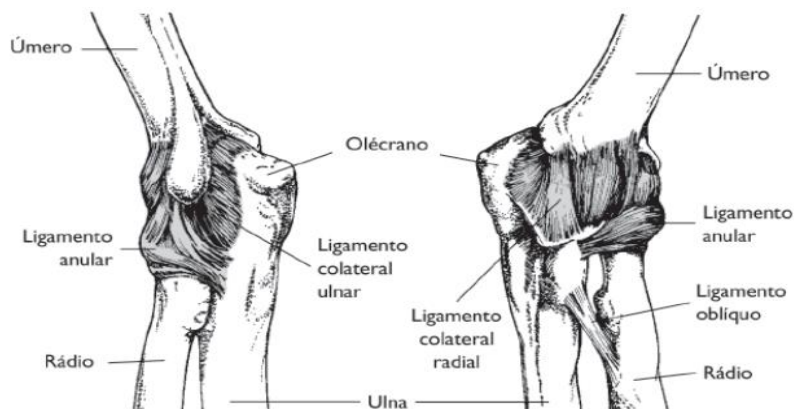


Fonte: HALL, 2016.

2.1.2 O COTOVELO

O cotovelo definido como uma estrutura simples de dobradiça composto por três articulações. A Figura 5 apresenta a principal delas: a articulação umeroulnar. Dentre os músculos que atuam no cotovelo, o bíceps contribui na flexão, enquanto o tríceps atua na extensão. (HALL, 2016).

Figura 5. Cotovelo



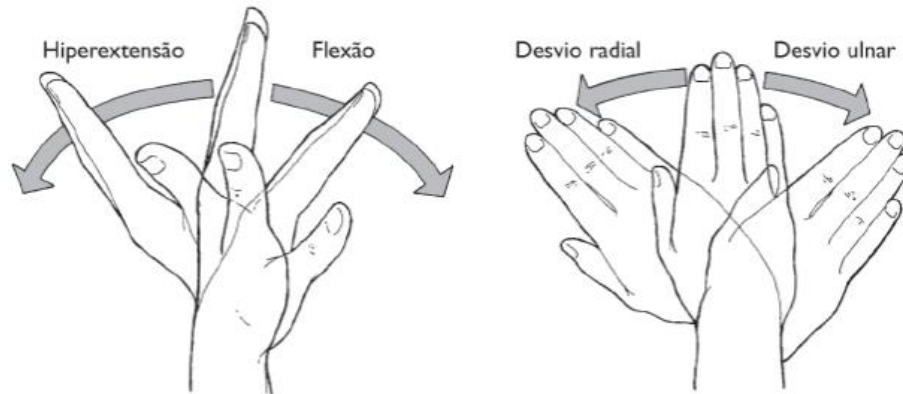
Fonte: HALL, 2016.

2.1.3 O PUNHO

O punho é uma articulação complexa capaz de realizar movimentos nos planos sagital e frontal, além do movimento rotacional. A flexão do punho é definida pelo movimento da superfície palmar da mão na direção da face anterior do antebraço. Enquanto a hiperextensão a superfície dorsal da mão se aproxima da face posterior do antebraço. Os movimentos do plano frontal são determinados pela direção da mão. Quando atua

para o polegar é o desvio radial, e na direção oposta é chamado de desvio ulnar. Esses movimentos descritos são apresentados na Figura 6.

Figura 6. Movimentos do punho



Fonte: HALL, 2016.

2.2 FUNDAMENTOS DE BRAÇOS ROBÓTICOS

2.2.1 BRAÇO ROBÓTICO

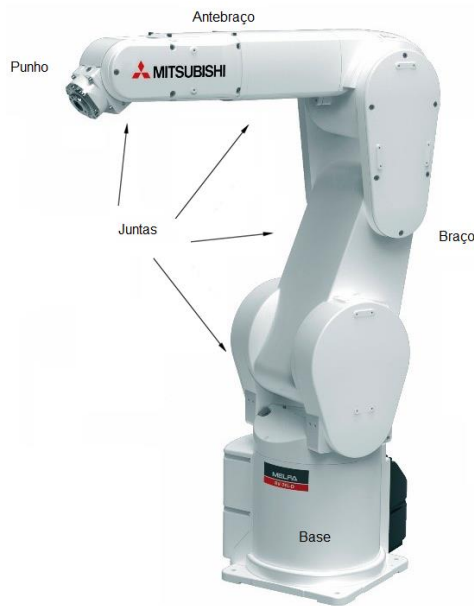
Um braço robótico é um dispositivo mecânico programável desenvolvido para manipular objetos de maneira similar a um braço humano. Em geral, são controlados por computadores e possuem sensores e atuadores que permitem a precisão e a repetitividade dos movimentos.

O projeto é realizado para atuar em tarefas específicas, como pegar objetos, manipulá-los e colocá-los em outro lugar (SHEPLEY, Phil, 2022). Alguns braços robóticos também possuem capacidade de aprendizado de máquina e pode ser ensinado a realizar tarefas específicas através de programação ou ensino por demonstração. (ROSARIO, 2010).

2.2.2 ANATOMIA DOS BRAÇOS MECÂNICOS

Um braço robótico industrial pode ser dividido em cinco elementos: base, braço, punho, antebraço e juntas, uma junta é a junção entre 2 elos, um elo é uma estrutura rígida não móvel. A Figura 7 ilustra um braço robótico fabricado pela empresa Mitsubishi.

Figura 7. Braço robótico da empresa Mitsubishi

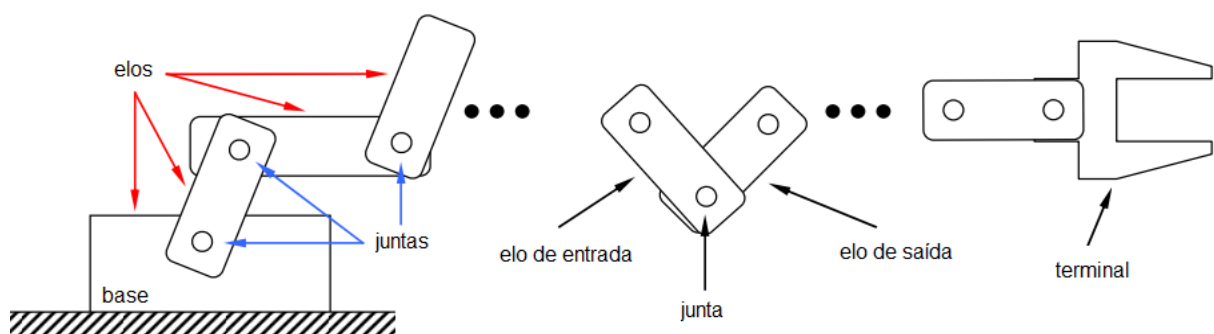


Fonte: autoria própria

O movimento do braço robótico é composto pela rotação das juntas do braço, punho e base. Em grande parte dos braços robóticos, também há uma base que pode ser estacionária ou móvel (GROOVER, 2011).

O braço atua com elos unidos por juntas de movimento relativo acionadas individualmente. Essa estrutura é fixada à base por um lado e ao punho pelo outro. O punho é composto por várias juntas próximas entre si que possibilitam a realização da tarefa. A Figura 8 exemplifica o esquema básico de um braço robótico (CARRARA, 2015).

Figura 8. Esquema básico de um braço robótico



Fonte: Adaptada de CARRARA, 2015.

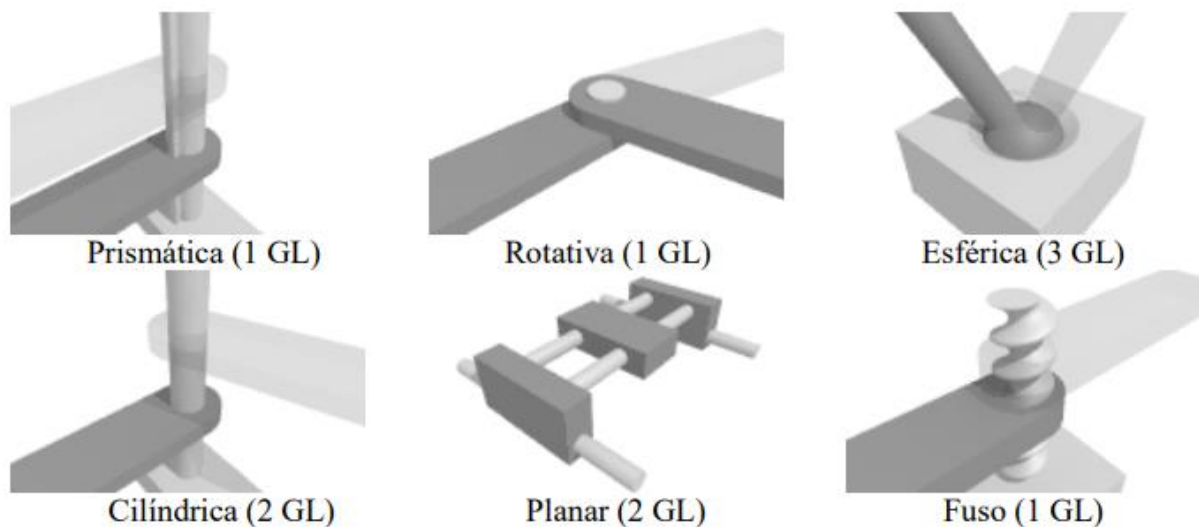
Os elos são denominados de acordo com sua distância da base. O mais próximo da base é chamado de elo de entrada, enquanto o elo mais extremo (próximo ao terminal) é conhecido como elo de saída (CARRARA, 2015).

As juntas do braço robótico podem ser classificadas de acordo com o tipo de movimento: rotação ou translação e quantidade de direções desse movimento (graus

de liberdade, abordado com mais detalhes na seção seguinte). A Figura 9 descreve os tipos de articulação (CARRARA, 2015):

- Prismática ou linear: realiza um movimento deslizante em linha reta.
- Rotativa: permite um movimento de rotação perpendicular em torno de uma linha imaginária estacionária chamada de eixo de rotação. Essa junta movimenta-se tanto como dobradiça quanto como cadeira giratória.
- Esférica: atua como a combinação de três juntas de rotação, o que possibilita rotações em torno de três eixos distintos.
- Cilíndrica: é composta por duas juntas, uma rotacional e uma prismática. Essa combinação permite duas direções de movimento.
- Planar: consiste em duas juntas prismáticas, e atua com movimentos em duas direções.
- Fuso ou parafuso: é formada por um parafuso e uma rosca que executa um movimento semelhante ao da junta prismática, contudo, o movimento realiza uma rotação no eixo central (movimento do parafuso).

Figura 9. Tipos de articulação

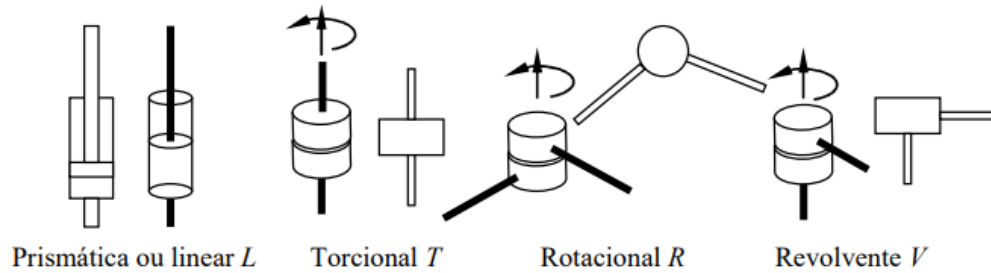


Fonte: CARRARA, 2015.

As juntas prismáticas e rotativas são as mais utilizadas em robôs industriais e também podem ser classificadas conforme o movimento realizado ilustrado na Figura 10 (CARRARA, 2015 e OLIVEIRA, 2014):

- Linear ou prismática: o elo de entrada e o de saída são paralelos entre si e deslizam um sob o outro.
- Torção: o movimento de rotação é perpendicular entre os elos de entrada e de saída com a mesma direção do eixo de rotação da junta.
- Rotacional: o movimento dos elos de entrada e de saída são perpendiculares ao eixo de rotação da junta.
- Revolvente ou ortogonal: o elo de entrada possui a mesma direção do eixo de rotação, mas o elo de saída é perpendicular a este.

Figura 10. Movimentos das juntas



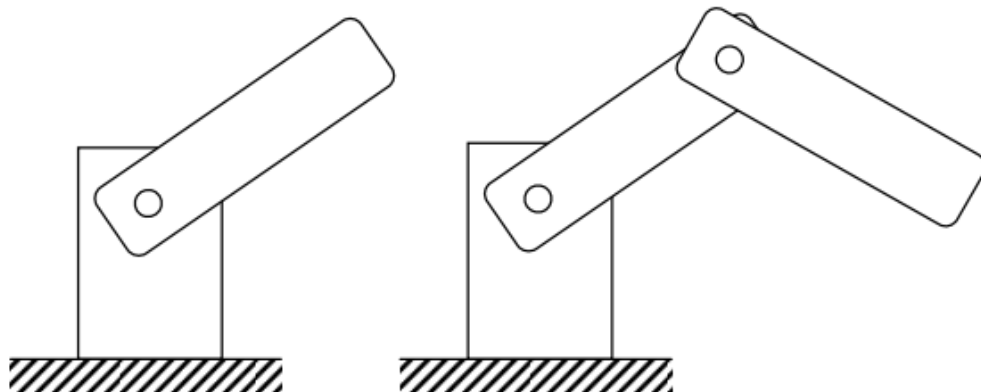
Fonte: CARRARA, 2015.

2.2.3 GRAUS DE LIBERDADE

Segundo Norton, o grau de liberdade de um sistema é um número igual ao número de variáveis independentes necessárias para definir uma posição no espaço em um determinado momento, ou seja, é a quantidade de parâmetros necessários para especificar as condições de mobilidade de uma articulação (NORTON, 2010).

A quantidade de graus de liberdade de um braço robótico é definida pela soma dos graus de liberdade de suas juntas. Se o movimento relativo atua num único eixo, a junta possui apenas um grau de liberdade. Quando permite dois eixos de movimento, apresenta dois graus de liberdade. A Figura 11 exemplifica duas estruturas uma com um grau de liberdade e outra com dois (CARRARA, 2015).

Figura 11. Braços com um (à esquerda) e dois graus de liberdade (à direita)

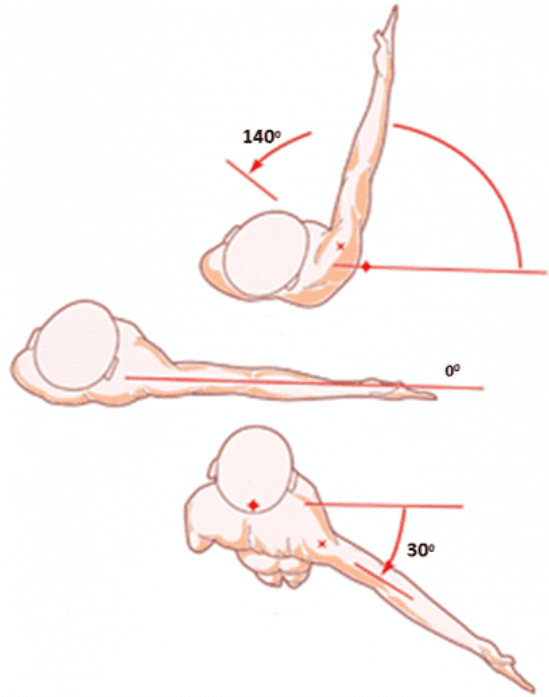


Fonte: adaptada de CARRARA, 2015.

O braço humano possui seis graus de liberdade, cada grau de liberdade permite dois movimentos da articulação, que podem ser entendidos como o mesmo movimento com o sentido contrário, estes movimentos são os seguintes:

- Flexão/extensão do ombro: movimento que permite levar o braço para frente e para trás, o movimento é ilustrado na Figura 12.

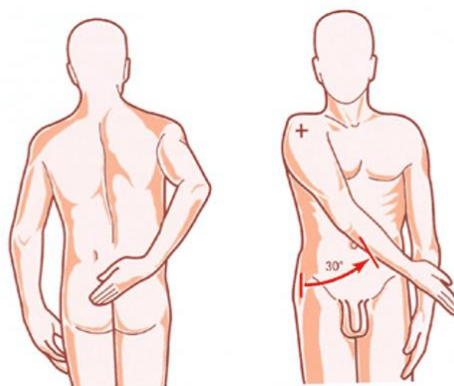
Figura 12. Flexão/extensão do ombro



Fonte: KAPANDJI, 2007.

- Abdução/adução do ombro: movimento que permite levar o braço para fora e para dentro do corpo, o movimento é ilustrado na Figura 13.

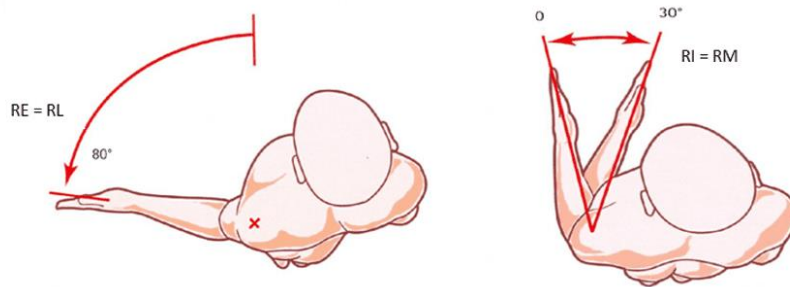
Figura 13. Abdução/adução do ombro



Fonte: KAPANDJI, 2007.

- Rotação interna/externa do ombro: movimento que permite girar o braço no sentido horário e anti-horário, o movimento é ilustrado na Figura 14.

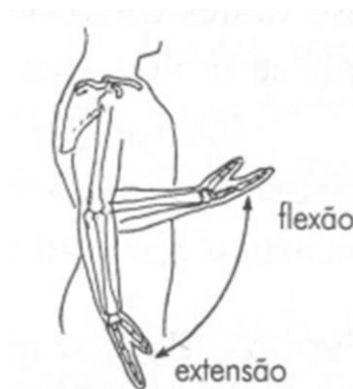
Figura 14. Rotação interna/externa do ombro



Fonte: KAPANDJI, 2007.

- Flexão/extensão do cotovelo: movimento que permite dobrar e estender o cotovelo, o movimento é ilustrado na Figura 15.

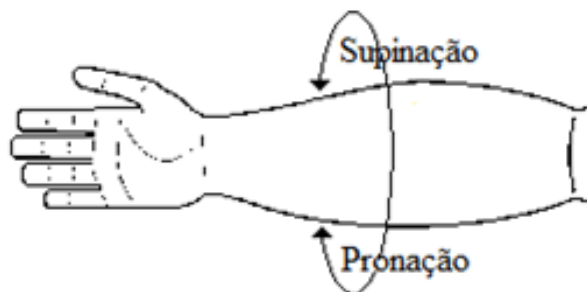
Figura 15. Flexão/extensão do cotovelo



Fonte: ASSUNÇÃO.

- Pronação/supinação do antebraço: movimento que permite girar o antebraço de forma que a face interna ou externa do punho fique voltada para cima, o movimento é ilustrado na Figura 16.

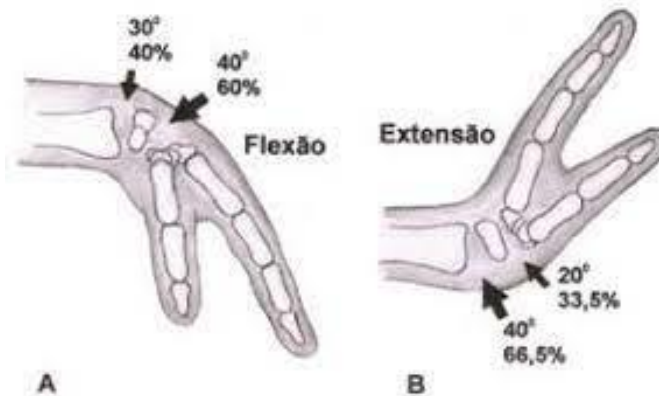
Figura 16. Pronação/supinação do antebraço



Fonte: coelho, 2012.

- Flexão/extensão do punho: movimento que permite dobrar e estender os dedos (HALL, 2016), o movimento é ilustrado na Figura 17.

Figura 17. Flexão/extensão do punho



Fonte: HENRIQUES, 2018.

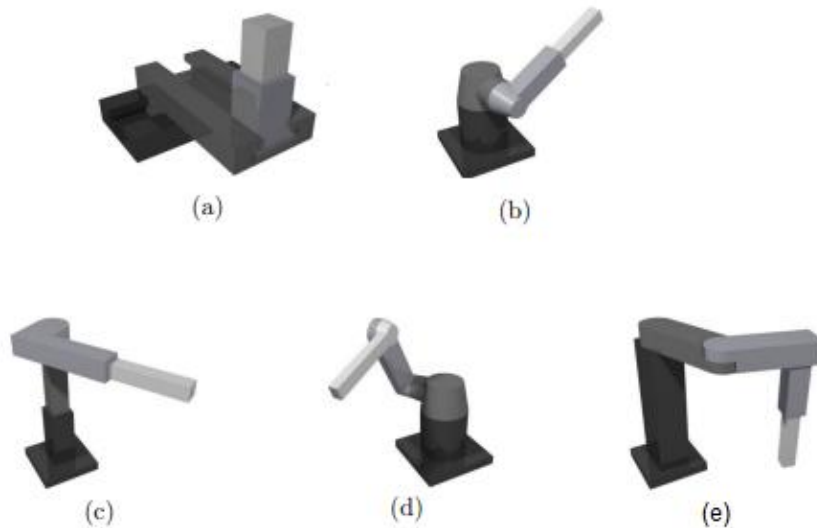
Para simular os movimentos do braço, foi simplificado em dois vetores, braço e antebraço, por conta desta simplificação o movimento de abdução/adução do ombro iria apenas transladar o ponto do ombro não alterando os vetores, o movimento de pronação/supinação do antebraço iria rotacionar o vetor do antebraço no próprio eixo não causando nenhuma alteração, e o movimento de flexão/extensão do punho não alteraria os vetores pois na simplificação não é considerado o punho como parte do braço, devido a simplificação o modelo do braço precisará de apenas 3 graus de liberdade, o primeiro para a flexão/extensão do ombro, o segundo para rotação interna/externa do ombro e o terceiro para flexão/extensão do cotovelo.

2.2.4 CLASSIFICAÇÃO DE BRAÇOS ROBÓTICOS

Groover (2011) define cinco configurações básicas de braços robóticos apresentadas na Figura 18. Essa classificação é determinada pela associação de movimentos realizado pelas juntas (linear, torcional, rotacional ou revolvente).

- Cartesiano: utiliza três juntas lineares com movimento deslizantes. É a configuração mais simples quando comparada com as demais.
- Polar: possui duas juntas rotativas cada um desliza em relação ao corpo, gira em torno de um dos eixos (vertical e horizontal). Foi projetado para suportar grandes cargas e amplo alcance.
- Cilíndrico: é composto de uma coluna vertical onde o braço move para cima e para baixo, direita e esquerda.
- Articulado: consiste em três juntas rotativas e permite o giro sobre uma base. Sua configuração é similar com as movimentações de um braço humano e suas juntas são comparadas ao ombro e cotovelo. Por esse motivo, são os mais usados nas indústrias.
- Scara: assemelha-se ao robô articulado, excerto pelo ombro e o cotovelo que além do movimento de flexão, possuem também o movimento rotacional. Devido a sua grande precisão e repetibilidade, é bastante utilizado na indústria de montagem mecânica ou eletrônica (*pick and place*).

Figura 18. Classificação de braço robótico: (a) cartesiano, (b) polar, (c) cilíndrico, (d) articulado, (e) Scara



Fonte: adaptada de CARRARA, 2015.

2.3 CAPTURA DE MOVIMENTO DO BRAÇO HUMANO

2.3.1 VISÃO COMPUTACIONAL

A visão computacional é um campo da ciência que estuda como os computadores interpretam imagens ou vídeos, os principais focos são em como processar, analisar e entender as imagens para extrair informações (DE MILANO, 2014).

A tecnologia de visão computacional possui algumas abordagens, algumas simples como filtrar cores e tamanho de áreas destacadas, já outras mais complexas como usar aprendizado de máquina para tentar replicar o cérebro humano, os algoritmos de visão computacional usados hoje em dia são baseados em reconhecimento de padrões. Os computadores são treinados com uma quantidade massiva de dados visuais e suas descrições para criar um modelo de rede neural convolucional, assim eles identificam os padrões nas imagens e tentam encontrar estes padrões em outras imagens para classificá-las segundo o modelo. Exemplo disso seria fornecer milhões de fotos de gatos, com isso o algoritmo cria um padrão de um gato, quando receber outras fotos ele tenta identificar o modelo que foi criado nas novas fotos (BABICH, 2020).

Com a visão computacional, pode-se entender que o computador ganhou a habilidade de ver de maneira semelhante a um humano, podendo identificar objetos, formas, movimentos e muito mais.

2.3.2 BIBLIOTECA OPENCV

É uma biblioteca que foi desenvolvida pela empresa intel com o objetivo de tornar a visão computacional mais acessível aos usuários e programadores. A biblioteca de código aberto possui diversas funções para processamento de imagens, análise

estruturas, análise de movimento e rastreamento de objetos, reconhecimento de padrões e calibração de câmera e reconstrução 3D (MARENGONI. 2009).

2.3.3 BIBLIOTECA MEDIAPIPE

É um *framework* de código aberto para *machine learning* que cria *pipelines* para o processamento de dados de serie temporal, como vídeos e áudios. Foi desenvolvido pela Google em 2012, inicialmente para análise de dados de vídeos e áudios no *youtube*, mas que foi integrado em diversos outros produtos como *Gmail*, *Google Photos* e *Google Home* (LUGARESI. 2019).

2.4 CIRCUITO

2.4.1 ARDUINO

Arduino é uma plataforma de prototipagem eletrônica de hardware livre e placa única, ela permite criar programas capazes de processar entradas e saídas entre um dispositivo e os componentes externos conectados a ele. As placas possuem 14 pinos digitais que podem ser definidos individualmente como entrada ou saída, além disso 6 dos 14 pinos digitais podem fornecer uma saída de modulação por largura do pulso (PWM).

O Arduino utilizado no projeto é o Arduino nano, que utiliza o microcontrolador Atmega 328 e possui 14 pinos digitais (entrada/saída) sendo que 6 podem ser usados como saídas PWM (Modulação por largura de pulso, do inglês *Pulse Width Modulation*), 8 pinos analógicos, um cristal oscilador de 16 MHz, entrada USB, entrada de alimentação e um botão reset. As dimensões são reduzidas em comparação aos demais modelos de Arduino.

Diversos protocolos de comunicação estão disponíveis como o protocolo serial, bus serial de interface periférica (SPI) e I2C/TWI. Em cada placa é incluso como recurso padrão um conector de programação serial *in-circuit* (ICSP) e um botão reset. A linguagem de programação usada é essencialmente C/C++ (EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. 2013).

2.4.2 SERVOMOTORES

Os servomotores são dispositivos formados por um motor de corrente continua, engrenagens, um potenciômetro e um circuito de controle. O circuito de controle junto ao potenciômetro forma um sistema de realimentação capaz de controlar a posição do eixo do motor de forma precisa. Servomotores comuns possuem sua rotação limitada de 0° a 180°, sua posição é definida pela entrada do motor, sua entrada é um sinal modulado por largura de pulso (PWM), a cada 20 milissegundos o circuito de controle checa o valor da entrada, o tempo do pulso de entrada dura de 1 a 2 milissegundos, a tensão de entrada varia de 0 a 5 volts (TANNUS, 2018.).

O projeto usa três servos-motores do modelo SG90, que apresentam alimentação nominal de 4,8-6 V, com torque que chega até 2,5 kg/cm, entretanto como o projeto visa apenas o movimento e não o carregamento de cargas, não será usado o limite

de torque do motor, conseqüentemente o motor vai requerer menos energia, permitindo que a tensão de entrada fique mais próxima de seu limite inferior.

2.4.3 COMUNICAÇÃO SERIAL

A interface serial UART (*universal asynchronous receiver transmitter*) do Arduino é usada para receber e enviar dados para o microcontrolador. Para isso são usados os pinos Tx para a transmissão e Rx para recepção. Na comunicação serial usa-se um dispositivo que envia comandos (mestre) e um dispositivo que recebe o comando e o executa (escravo), desta maneira o dispositivo mestre envia instruções bit a bit para os dispositivos escravos (TANNUS, 2018.).

Desta forma é possível enviar e receber dados em forma de caracteres, um caractere de cada vez, assim podendo criar uma lógica que recebe estes caracteres, une eles em forma de uma *string* e trata a informação como for a necessidade do projeto.

3 METODOLOGIA

Este trabalho é uma pesquisa classificada como aplicada quanto a sua finalidade, descritiva quanto aos objetivos, com abordagem qualitativa e quantitativa, e experimental quanto aos procedimentos. O método científico utilizado será o hipotético-dedutivo.

3.1 CONSTRUÇÃO DO PROTÓTIPO DE BRAÇO ROBÓTICO

3.1.1 PARTE MECANICA

Foi escolhido um projeto de braço robótico disponibilizado em um site chamado *thingiverse*, a Figura 19 mostra o protótipo finalizado, é um braço robótico articulado, no projeto ele possui 4 graus de liberdade, mas como o movimento de abrir e fechar a mão não será considerado neste trabalho, serão utilizados apenas 3 graus de liberdade.

Figura 19. Protótipo finalizado



Fonte: autoria própria.

A estrutura do braço foi feita com plástico ABS em uma impressora 3D, as peças desmontadas podem ser vistas na Figura 20.

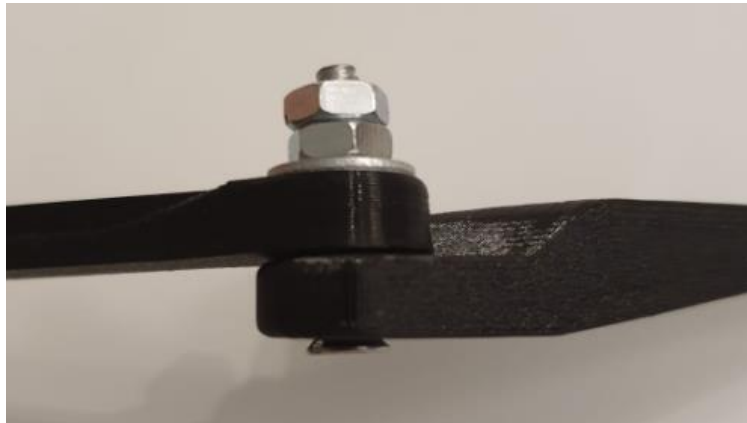
Figura 20. Partes do protótipo desmontado



Fonte: autoria própria.

O braço é composto de 11 juntas rotativas, para as conexões das juntas foram usados parafusos, porcas e arruelas de 1/8 de polegada, como mostrado na Figura 21, as porcas são apertadas uma contra a outra, deixando uma pequena folga para permitir a rotação da junta, mas impedir movimentos laterais.

Figura 21. Junta do protótipo



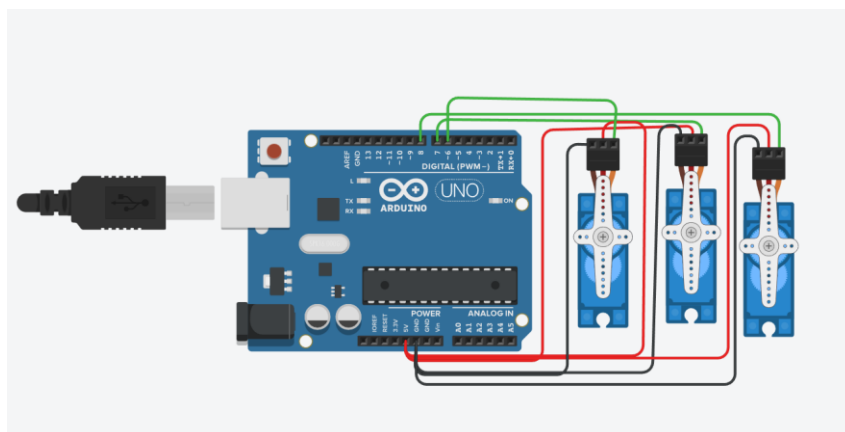
Fonte: autoria própria.

3.1.2 PARTE ELETRICA

A parte elétrica é composta de um Arduino NANO e 3 servomotores SG90. A Figura 22 ilustra a conexão do protótipo, devido às limitações do software usado para a ilustração, foi usado o modelo do Arduino UNO na figura, a Figura 23 mostra o Arduino NANO utilizado.

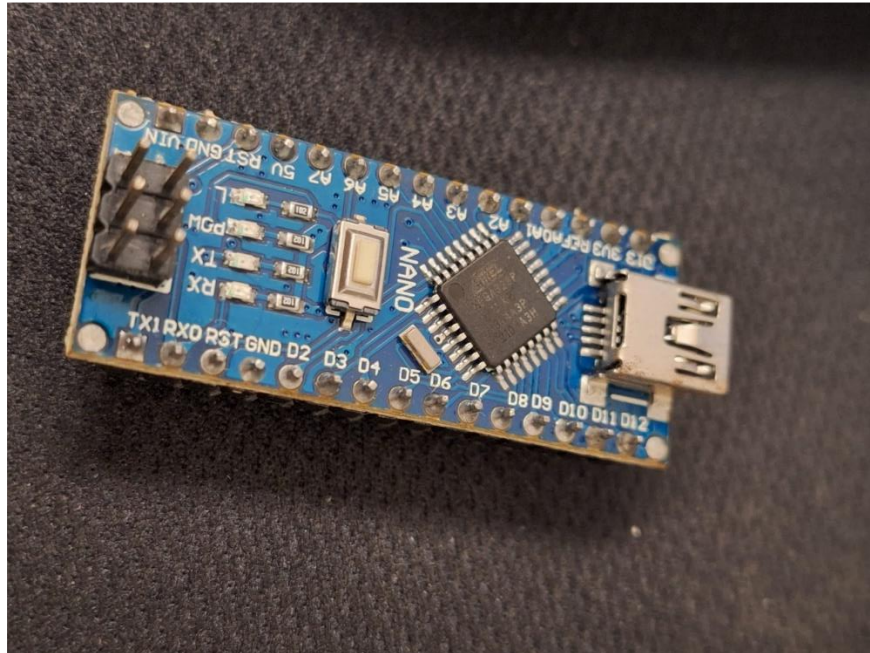
Nela foi ligado os pinos de energia dos motores em paralelo, usando apenas um pino do Arduino, analogamente para os pinos terra (GND), os pinos de informações precisam ser ligados separadamente para permitir o movimento independente de cada motor.

Figura 22. Conexões elétricas



Fonte: autoria própria.

Figura 23. Arduino NANO



Fonte: autoria própria.

3.2 CÓDIGO COMPUTACIONAL PARA ACIONAMENTO DOS MOTORES DO PROTÓTIPO DE BRAÇO ROBÓTICO.

O código computacional que move o braço, disponível no apêndice A, é um código que recebe a informação de como mover o braço através de comunicação serial em formato de *string*, analisa a *string*, identifica se é uma *string* válida e caso seja modifica a posição de cada motor para mover o braço de acordo com as instruções passadas.

A *string* deve estar num formato I+A+XXX+B+YYY+C+ZZZ+F (exemplo: IA090B045C060F) onde I indica o início do comando, F o final do comando, A indica que o ângulo é de Rotação, B indica que o ângulo é de adução, C indica que o ângulo é do cotovelo e os valores XXX, YYY e ZZZ são os valores do ângulo que o motor deverá se posicionar, a Tabela 1 mostra cada parte do comando separadamente.

Tabela 1. Exemplo de comando.

Início	Motor rotação	Ângulo rotação	Motor adução	Ângulo adução	Motor cotovelo	Ângulo cotovelo	Fim
I	A	XXX	B	YYY	C	ZZZ	F
I	A	090	B	045	C	060	F

Fonte: autoria própria.

3.3 CÓDIGO COMPUTACIONAL QUE IDENTIFICA A POSIÇÃO DO BRAÇO HUMANO

O código computacional que captura os movimentos utiliza as bibliotecas *OpenCV*, *cmath*, *mediapipe*, *numpy*, *cv2*, *serial* e *sympy* para com o sinal de entrada

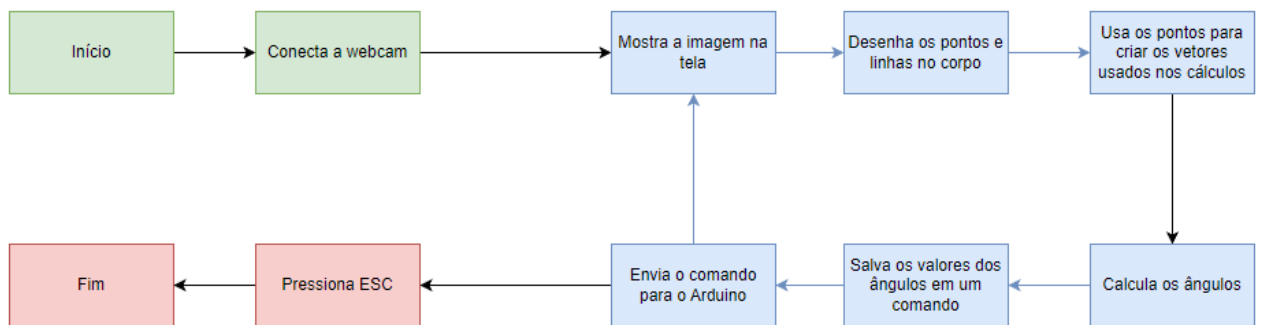
de uma *webcam* com resolução de 640x360 pixels, capturar a posição dos pontos do corpo, calcular os ângulos formados pelo braço e enviar o comando para o Arduino através de comunicação serial.

A biblioteca *mediapipe* capta a pose do corpo humano na imagem e calcula a posição de alguns pontos do corpo humano, ela utiliza a largura da cintura para poder calcular a posição tridimensional em uma imagem bidimensional.

Na figura 24, é possível ver um fluxograma de como o código funciona, nele é possível ver que o código funciona como um *loop* em que captura a imagem, mostra como o computador está interpretando a imagem para o usuário, em seguida calcula os ângulos para serem enviados ao braço. Como o loop ocorre rapidamente, as imagens mostradas formam um vídeo da perspectiva da visão computacional do movimento humano. No apêndice B, é possível encontrar o código e ver detalhadamente como foi feita cada etapa para enviar o comando de como o braço deve se mover, o cálculo dos ângulos é explicado na seção 3.4.

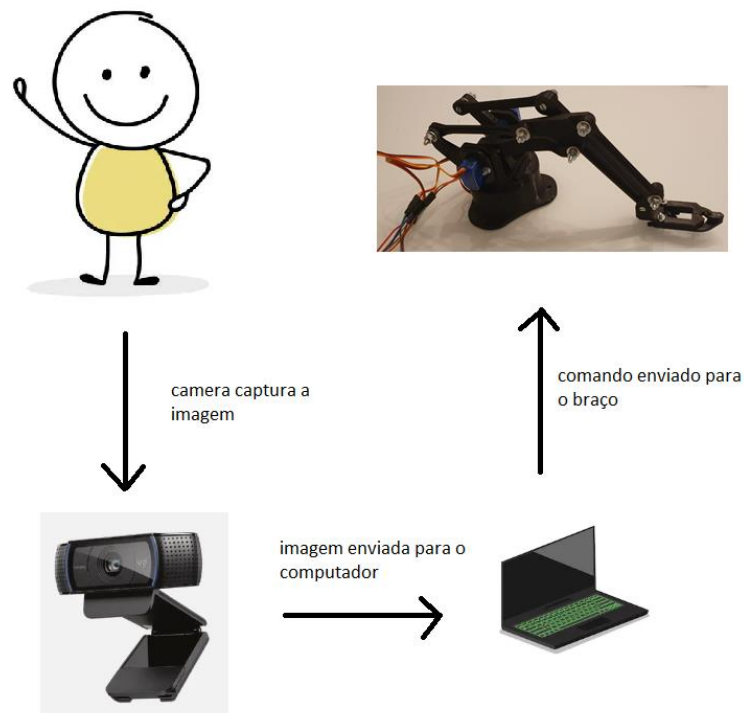
Após realizar o cálculo dos ângulos, devido a cinemática do protótipo, o ângulo de flexão do cotovelo foi incrementado no valor do ângulo de adução do ombro, para que o braço se posicione da forma desejada.

Figura 24. Fluxograma do código computacional



Fonte: autoria própria.

Figura 25. Diagrama do sinal



Fonte: autoria própria.

3.4 ÂNGULOS

Os ângulos calculados no código foram chamados de ângulo cotovelo, ângulo flexão e ângulo rotação, o ângulo do cotovelo é o ângulo de flexão/extensão do cotovelo, o ângulo de flexão é o ângulo de abdução/adução do ombro, e o ângulo de rotação é o ângulo de rotação do ombro.

Devido a cinemática do protótipo, para que o protótipo se movesse da mesma forma que o braço real, o ângulo enviado para o posicionamento do motor que controla o ângulo do cotovelo precisa ser a soma dos ângulos do cotovelo e flexão (flexão do cotovelo, abdução do ombro).

3.4.1 Ângulo de flexão/extensão do cotovelo

O ângulo de flexão/extensão do cotovelo é o ângulo que é formado entre o antebraço e o braço, no código a variável que representa este ângulo é chamada de `angulo_cotovelo`. Ele é calculado pela relação do produto interno pelo produto das normas dos vetores do braço e do antebraço, como mostrado nas Equações (1), (2) e (3).

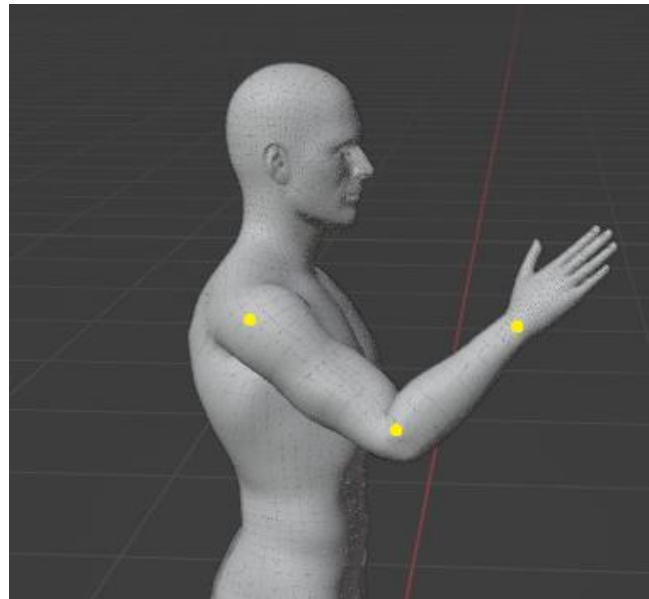
$$\text{Produto interno} = \langle \text{braço}, \text{antebraço} \rangle \quad (1)$$

$$\text{Cosseno do angulo} = \frac{\text{Produto interno}}{|\text{Braço}| * |\text{Antebraço}|} \quad (2)$$

$$\text{Angulo suplementar} = \arccos(\text{Cosseno do angulo}) \quad (3)$$

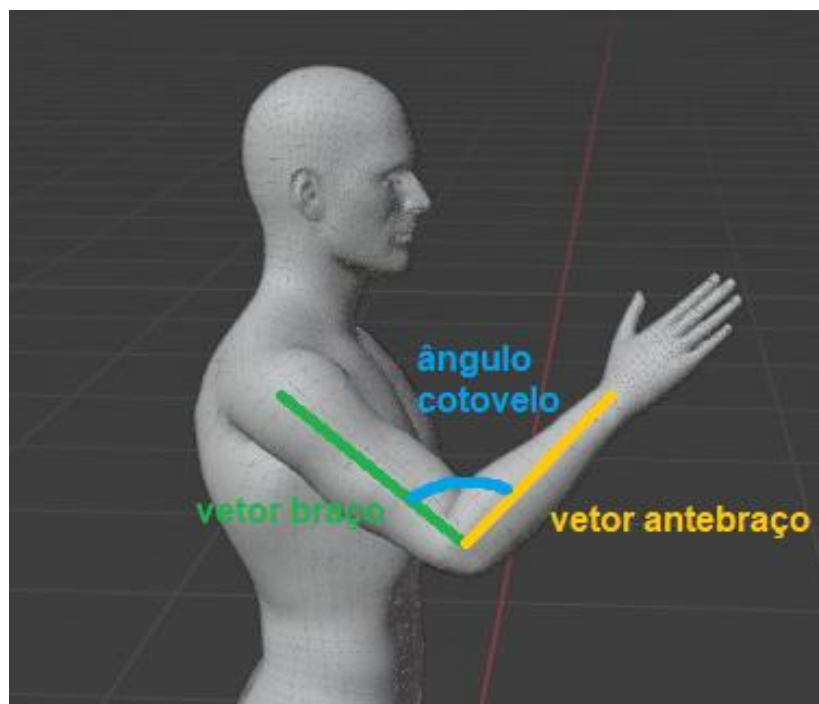
Nas Figuras 26 e 27 são ilustrados os vetores do braço e do antebraço. O vetor do braço é composto pelos pontos do cotovelo e ombro e tem sentido positivo do ombro para o cotovelo. Enquanto, o vetor do antebraço é formado pelos pontos do pulso e cotovelo e a referência do sentido é dada do cotovelo para o pulso.

Figura 26. Pontos no braço



Fonte: autoria própria.

Figura 27. Ângulo de flexão/extensão do cotovelo



Fonte: autoria própria.

Conforme as referências escolhidas, o ângulo calculado pelos vetores do braço e do antebraço é o suplemento do ângulo desejado. Desta forma, o ângulo do cotovelo é determinado pela Equação (4).

$$\text{Angulo do cotovelo} = 180 - \text{Angulo suplementar} \quad (4)$$

3.4.2 Ângulo de abdução/adução do ombro

O ângulo de abdução/adução do ombro, representado no código pela variável chamada de `angulo_flexao`, de maneira semelhante ao cálculo do ângulo de flexão/extensão do cotovelo nas Equações (1), (2), (3) e (4), o ângulo de abdução/adução do ombro pode ser definido pelas Equações (5), (6), (7) e (8), a Figura 28 mostra os pontos e vetores usados para calcular o ângulo.

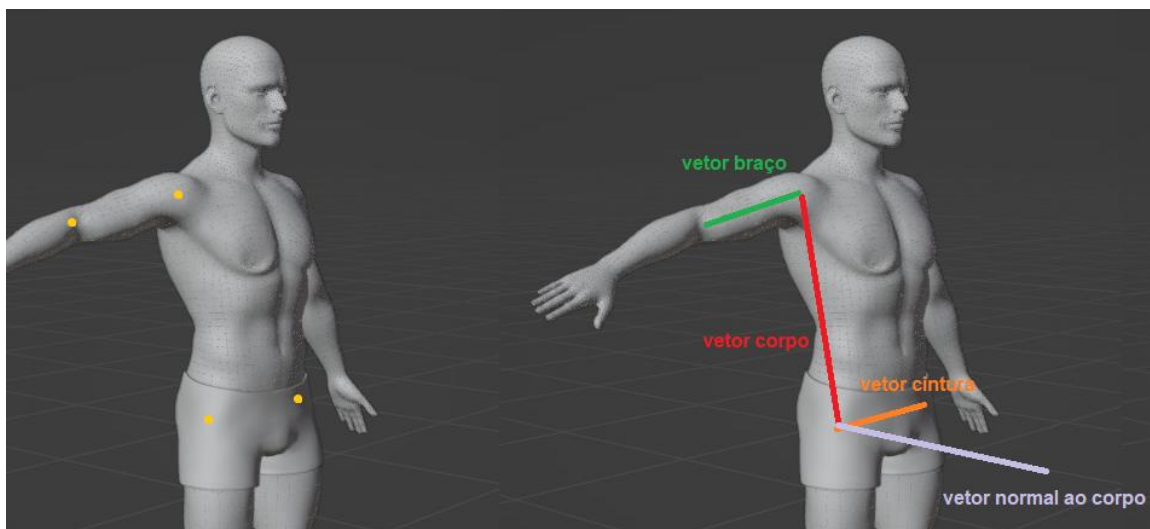
$$\text{Produto interno} = \langle \text{Vetor perpendicular a lateral do corpo}, \text{Braço} \rangle \quad (5)$$

$$\text{Cosseno do angulo} = \frac{\text{Produto interno}}{|\text{Vetor perpendicular a lateral do corpo}| * |\text{Braço}|} \quad (6)$$

$$\text{Angulo suplementar} = \arccos(\text{Cosseno do angulo}) \quad (7)$$

$$\text{Angulo complementar} = 180 - \text{Angulo suplementar} \quad (5)$$

Figura 28. Pontos e vetores no corpo



Fonte: autoria própria.

Contudo, o vetor paralelo ao corpo, chamado vetor corpo na Figura 28, que conecta os pontos do ombro e da cintura possui uma inclinação indesejada devido à diferença da largura da cintura e do ombro, esta inclinação causa erro ao calcular o ângulo. Assim, é preciso definir um novo vetor para se calcular o ângulo de maneira mais

confiável, o chamado vetor altura ilustrado na Figura 29 e representado no código pela variável `vetor_altura`, um vetor que toma o corpo como referencial e não possui a inclinação indesejada para não alterar o ângulo.

Também é necessário calcular um vetor que é perpendicular à lateral do corpo para o cálculo, chamado na Figura 29 de vetor ombro e representado no código pela variável `vetor_ombro`. O movimento de rotação iria alterar o valor do ângulo calculado pelo produto interno entre o vetor braço e o vetor altura, mas ao se usar um vetor perpendicular à lateral do corpo, a rotação do braço não iria mudar o ângulo calculado. Nota-se que o resultado seria o complemento do ângulo desejado.

Para calcular estes vetores foi necessário criar o vetor normal ao corpo, representado no código pela variável `vetor_normal_corpo`, conforme a Equação (9), é calculado pelo produto vetorial de dois vetores: vetor cintura e vetor corpo (apresentados na Figura 28). O vetor da cintura é composto pelos pontos da lateral esquerda e direita da cintura, já o vetor corpo pela lateral da cintura e o ombro. Logo, calcula-se o vetor normal ao corpo.

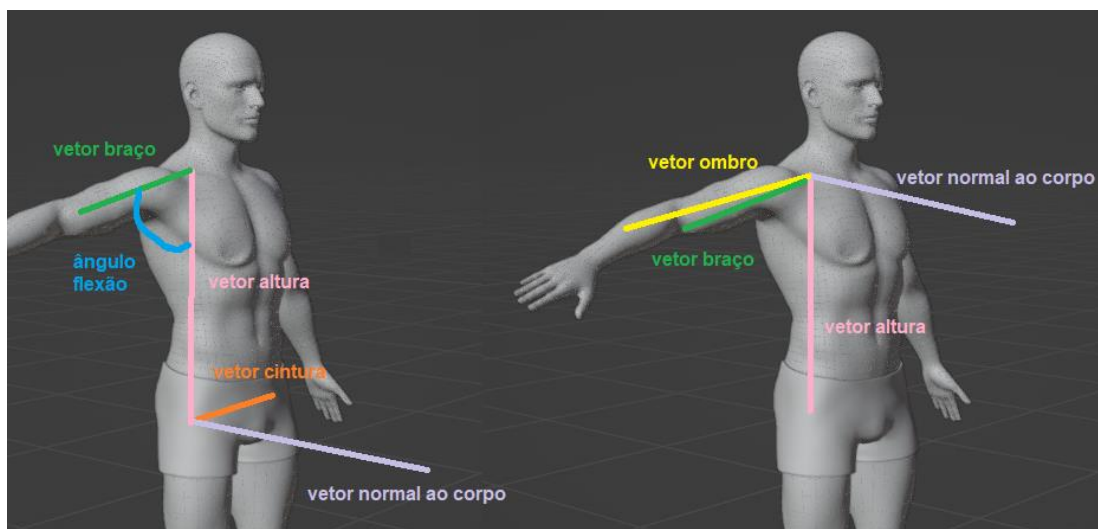
$$\text{Vetor normal ao corpo} = \text{Cintura} \wedge \text{Corpo} \quad (9)$$

Para calcular o vetor altura e o vetor ombro, seus valores são determinados pelo produto vetorial do vetor normal ao corpo com o vetor da cintura e com o vetor altura, respectivamente. As Equações (10) e (11) definem os cálculos e a Figura 29 ilustra os vetores descritos.

$$\text{Vetor altura} = \text{Vetor normal ao corpo} \wedge \text{Cintura} \quad (10)$$

$$\text{Vetor ombro} = \text{Vetor normal ao corpo} \wedge \text{Vetor altura} \quad (11)$$

Figura 29. Vetor altura, vetor ombro e ângulo de abdução/adução do ombro



Fonte: autoria própria.

Conforme as referências escolhidas, o ângulo desejado é o complemento do ângulo calculado entre os vetores de altura e ombro. Portanto, define-se na Equação (12) o ângulo de adução por:

$$\text{Angulo flexão} = 90 - \text{Angulo complementar} \quad (6)$$

3.4.3 Ângulo de rotação do ombro

O ângulo de rotação do ombro, no código a variável que representa este ângulo é chamada de `angulo_rotacao`, de forma similar ao cálculo do ângulo de flexão/extensão do cotovelo nas Equações (1), (2), (3) e (4), o ângulo de rotação do ombro pode ser calculado como mostram as Equações (13), (14), (15) e (16).

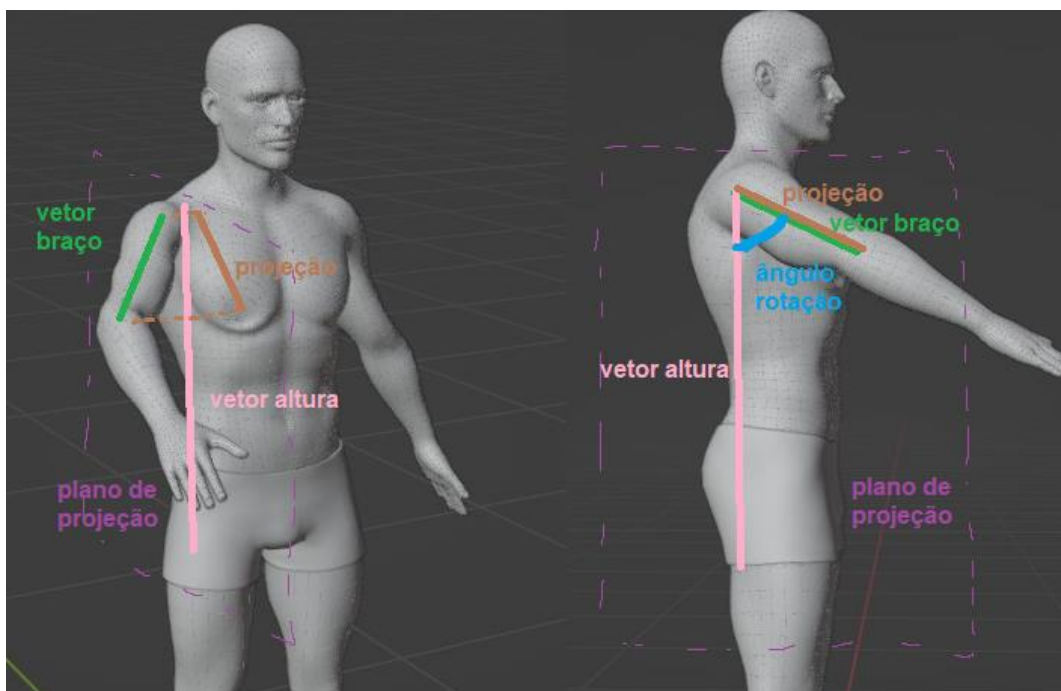
$$\text{Produto interno} = \langle \text{proj}_{\text{braço}}, \text{vetor altura} \rangle \quad (7)$$

$$\text{Cosseno do angulo} = \frac{\text{Produto interno}}{|\text{proj}_{\text{braço}}| * |\text{vetor altura}|} \quad (8)$$

$$\text{Angulo suplementar} = \arccos(\text{Cosseno do angulo}) \quad (9)$$

$$\text{Angulo do rotação} = 180 - \text{Angulo suplementar} \quad (10)$$

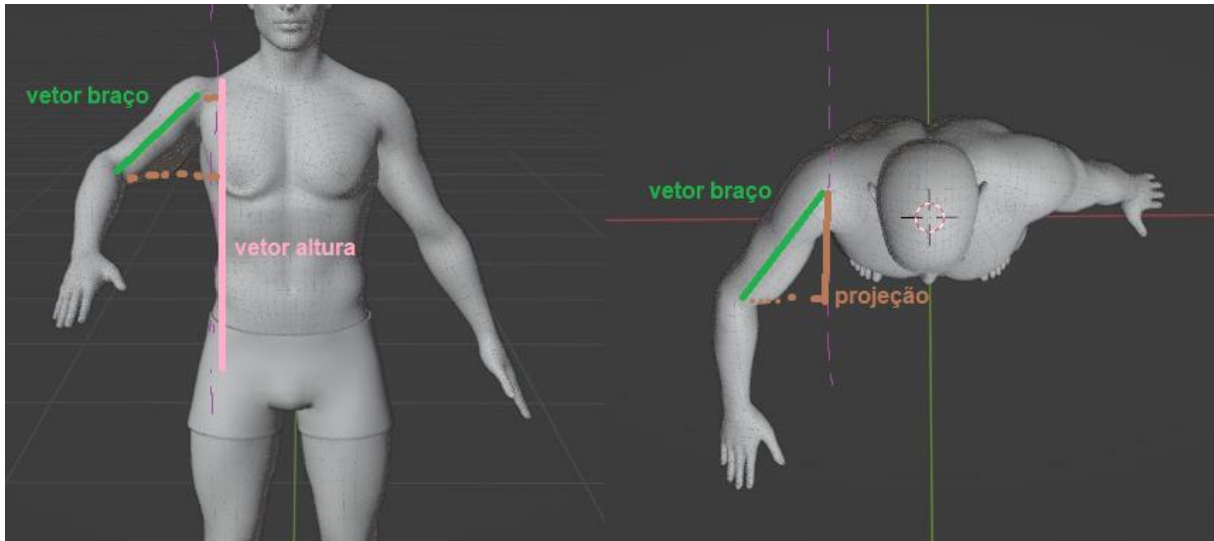
Figura 30. Vetor projeção do braço e vista lateral



Fonte: autoria própria.

Contudo, é necessário utilizar a projeção do vetor do braço no plano paralelo a lateral do corpo e o vetor altura, previamente definido. O vetor projeção é determinado projetando os pontos do ombro e cotovelo num plano qualquer perpendicular ao vetor normal a lateral do corpo. As Figuras 30 e 31 apresentam esse vetor nas vistas principais do corpo.

Figura 31. Vista frontal e Vista superior



Fonte: autoria própria.

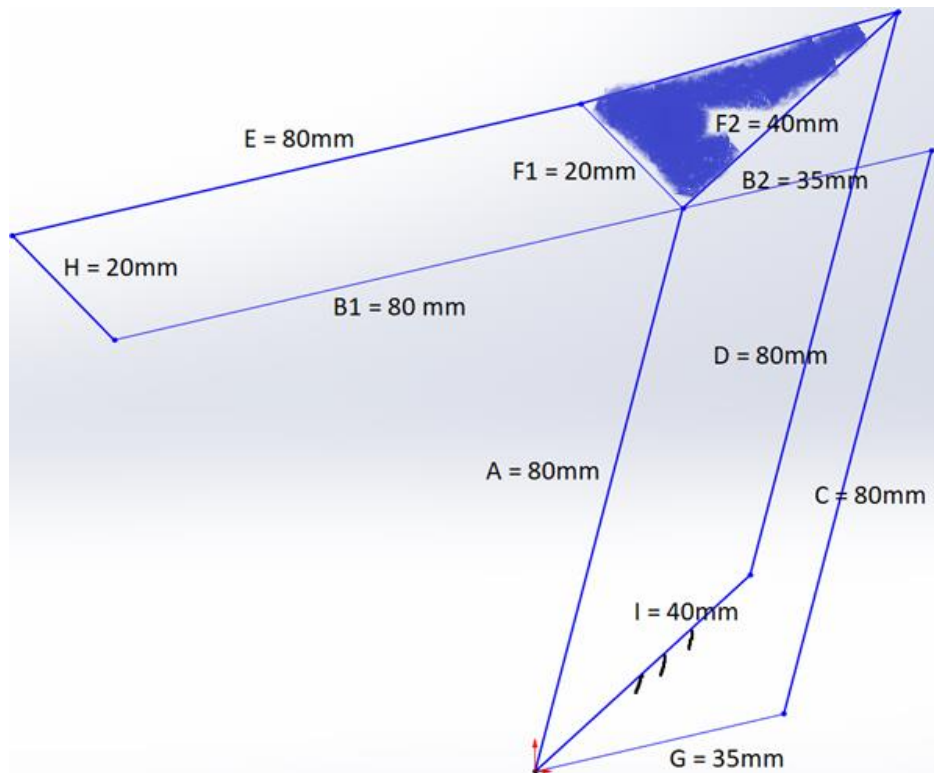
3.5 CINEMÁTICA DO BRAÇO ROBÓTICO

O modelo de braço robótico escolhido possui duas características geométricas de interesse para o projeto, a primeira é que o ângulo de flexão/extensão do cotovelo (ângulo da junta do cotovelo) é dependente do ângulo de abdução/adução do ombro, desta forma, caso o motor que controla o ângulo de adução se mova, para manter o ângulo do cotovelo o mesmo, é preciso que o motor que movimentava o ângulo do cotovelo se mova a mesma quantidade.

A segunda característica relevante é que a garra do braço robótico, que não foi utilizada no projeto, se manteve paralela à base do braço robótico, em caso de fazer alguma alteração para um futuro projeto esta característica pode ser interessante.

Com isso em mente serão demonstradas estas propriedades do braço robótico, a Figura 32 ilustra o mecanismo do braço robótico em que suas partes foram simplificadas em barras que representam seus corpos rígidos, não foi considerado a rotação da base pois não afeta as características mencionadas, a barra I é fixa à base, já a barra F é uma barra terciária, o que significa que possui 3 conexões, ela foi representada como um triângulo.

Figura 32. Mecanismo do braço



Fonte: autoria própria.

As partes do braço robótico foram medidas através do *solidworks*, que é um software de modelagem, os arquivos para impressão 3D foram abertos nele e as medidas das peças foram tomadas, como são muitas medidas, para evitar muitas imagens de pouca relevância, todas as medidas foram inseridas na Figura 32.

Como um paralelogramo pode ser definido pela propriedade de ter os lados opostos congruentes, o que significa que os lados opostos possuem o mesmo comprimento, é possível afirmar que a composição do mecanismo do braço robótico possui 3 paralelogramos, o primeiro formado pelas barras A, I, D e F2, o segundo formado pelas barras A, G, C, B2, e o terceiro formado pelas barras H, B1, F1 e E.

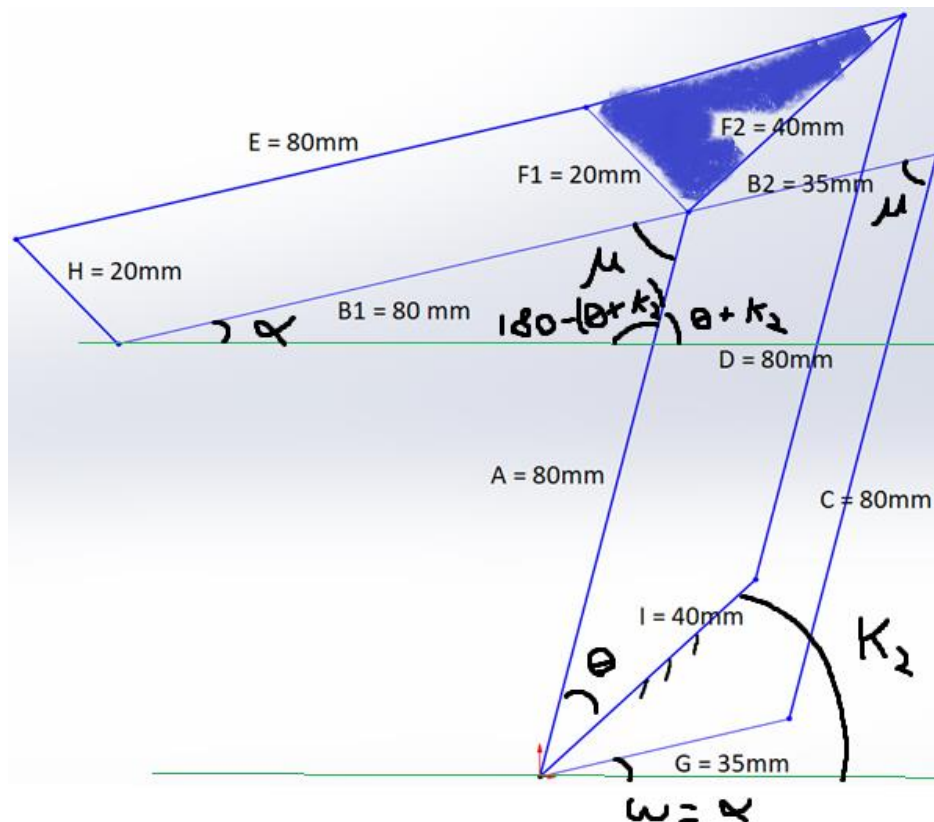
Por serem paralelogramos, é possível inferir as Equações (17 e 18):

$$A//D, \quad I//F2, \quad A//C, \quad B//G, \quad E//B1, \quad H//F1 \quad (17)$$

$$A//D, \quad A//C \rightarrow C//D \quad (18)$$

Traçando retas horizontais (retas verdes) é possível definir os ângulos α e ω , como mostra a Figura 33.

Figura 33. Ângulos do mecanismo parte 1



Fonte: autoria própria.

Como G é paralelo a B, é possível afirmar que α é igual a ω , como A é paralelo a C, é possível afirmar que o ângulo que B forma com C é o mesmo que B forma com A que foi chamado de μ .

μ é o ângulo de flexão/extensão do cotovelo, isso significa que a primeira característica que queremos será descoberta ao definir a relação dos ângulos dos motores e o ângulo μ .

Como mostrado na Figura 33, μ faz parte de um triângulo que inclui μ , α e $(180 - \theta + K2)$, $K2$ sendo o ângulo constante que a barra fixa I faz com a base do braço, desta forma, sabendo que a soma dos ângulos internos de um triângulo é 180, é possível chegar na Equação (19 e 20).

$$\mu = 180 - \alpha - (180 - \theta - K2) \quad (19)$$

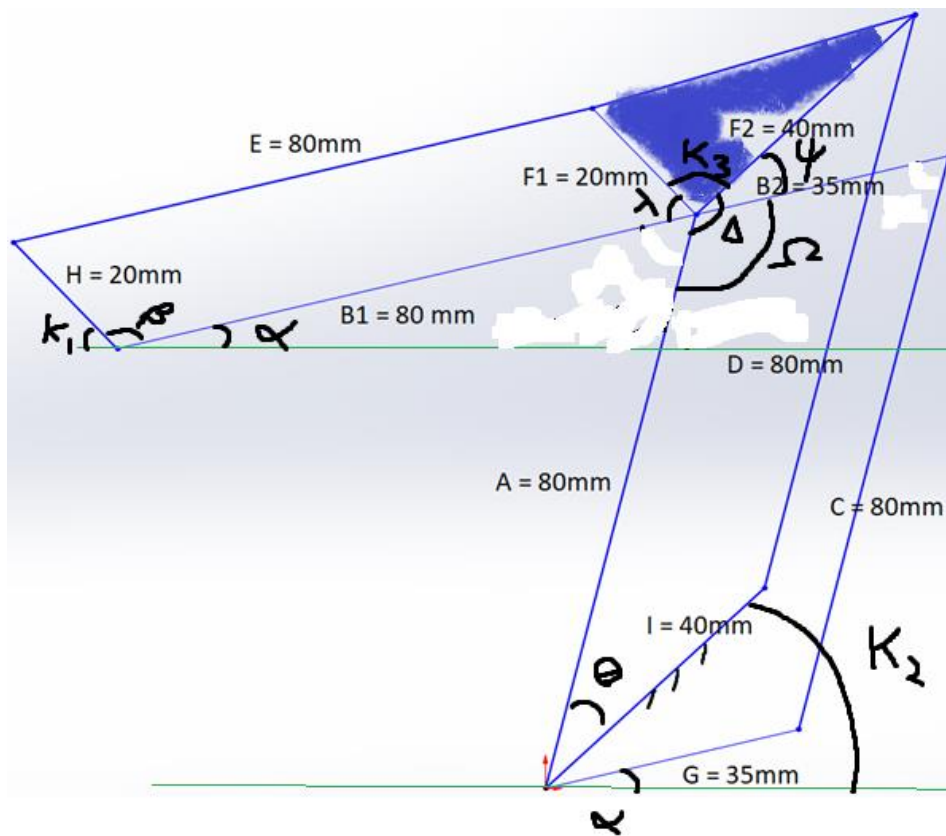
$$\mu = -\alpha + \theta + K2 \quad (20)$$

onde α é o ângulo da posição do eixo do motor que controla o ângulo do cotovelo, e $(\theta + K2)$ o ângulo da posição do eixo do motor que controla o ângulo de flexão, isso significa que para manter o ângulo do cotovelo igual, caso um dos motores se mova,

o outro tem que se mover o mesmo tanto para que a diferença entre os ângulos se mantenha constante.

A Figura 34 ilustra os ângulos que serão usados para provar a segunda característica mencionada do braço robótico, para provar que o garra do braço robótico é sempre perpendicular a base é preciso provar que a soma dos ângulos α e β é sempre a mesma, pois desta forma o ângulo que a barra da garra (H) forma com a base (linha horizontal verde) vai ser sempre o mesmo, pois o ângulo da garra K1 é constante por ser um corpo rígido e como o ângulo α foi definido, a soma desses 3 ângulos vai ser igual a 180 graus.

Figura 34. Ângulos do mecanismo parte 2



Fonte: autoria própria.

Com os novos ângulos definidos, é possível deduzir as Equações (21 a 29):

$$\Psi + K3 + \lambda = 180 \quad (21)$$

$$\lambda = 180 - \Psi - K3 \quad (22)$$

como β e λ são ângulos internos de um paralelogramo, sua soma vai ser 180.

$$\beta + \lambda = 180 \quad (23)$$

$$\beta = 180 - \lambda \quad (24)$$

$$\beta = \Psi + K3 \quad (25)$$

$$\Psi = \Delta - \Omega \quad (26)$$

$$\beta = \Delta - \Omega + K3 \quad (27)$$

$$\beta = 180 - \theta - (180 - (\theta + K2)) + K3 \quad (28)$$

$$\alpha + \beta = K2 + K3 = \text{constante} \quad (29)$$

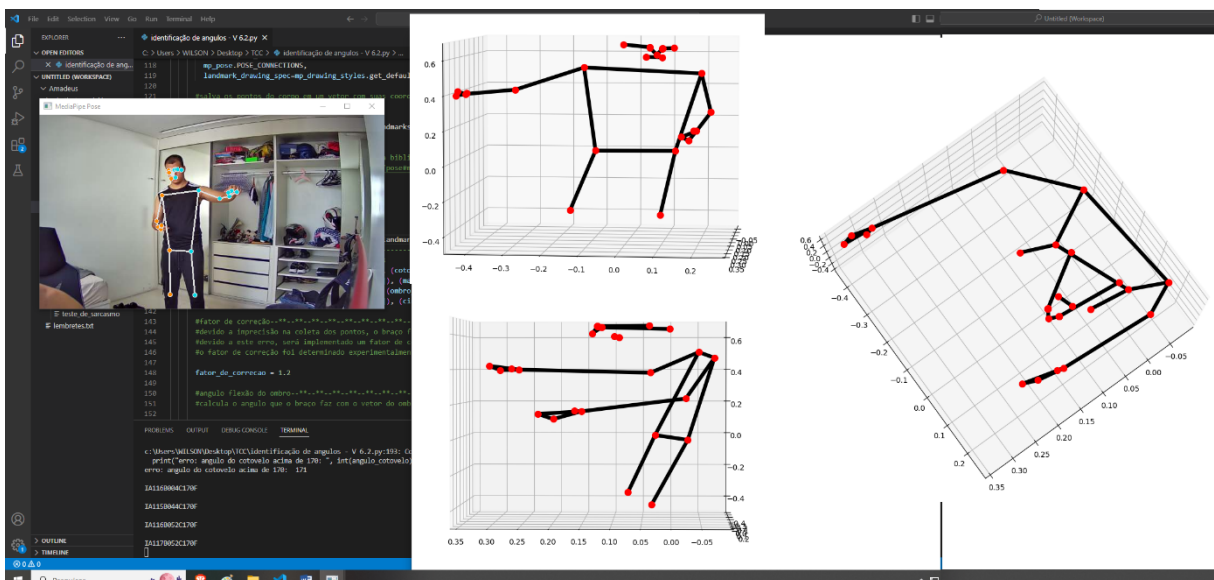
desta forma foi provado que a garra do braço sempre será paralela a base.

4 RESULTADOS

Nesta seção será discutido sobre o processo de desenvolvimento do braço, seus resultados, erros que ocorreram e como foram solucionados.

Inicialmente, foi cogitado o uso de dispositivos como pulseiras ou fitas refletivas para determinar os pontos do braço. Ainda, filtros de imagem seriam utilizados para determinar o ponto médio das áreas e definir os pontos num espaço tridimensional. Entretanto, esse método apresentou dificuldades e falhas na aquisição de dados. Assim, foi definido o uso da biblioteca *mediapipe*. Através dessa, a captura da posição foi simplificada pelo uso de apenas uma câmera (*webcam*), a Figura 35 mostra a perspectiva da visão computacional onde foi plotado os pontos num gráfico tridimensional.

Figura 35. Visão computacional.



Fonte: autoria própria.

Na impressão 3D da base do braço, o espaçamento para o servo motor não se adequou corretamente ao servo SG90. A fim de evitar desperdícios e reimpressão, a alteração foi realizada com um ferro de solda aquecido para derreter o material e possibilitar o encaixe perfeito do servo motor, como mostra a Figura 36.

Figura 36. Motor encaixado na base com furo ajustado



Fonte: autoria própria.

A fixação de algumas articulações cujo movimento gera mais vibração ocasionou o afrouxamento das porcas. Esse problema foi solucionado com adição de uma segunda porca, conforme apresentado na Figura 21. Essa foi pressionada sob a primeira para evitar que as porcas movam com a vibração. Ainda, possibilitou maior liberdade de movimento visto que as porcas não atuam diretamente sob articulação, o que ocasionaria seu travamento.

Como o projeto é uma prova de conceito de um projeto de baixo custo, o custo final foi uma das características a serem consideradas, como mostrado na Tabela 2, o custo total do projeto ficou em torno de R\$155,00, é uma estimativa pois alguns materiais não foram comprados, os servomotores foram comprados por R\$35,00 cada, o Arduino nano foi fornecido pelo laboratório de automação e controle (LAC) da universidade federal de Pernambuco (UFPE), em pesquisas online foi encontrado que o valor fica em torno de R\$40,00 em sites de comércio eletrônico, a impressão 3D do braço também foi fornecida pelo laboratório, como é difícil estimar o preço e normalmente impressões 3D possuem um valor baixo, o custo da impressão foi desprezado, os parafusos e porcas foram comprados por R\$20,00, mas nem todos foram usados pois foram comprados diversos tamanhos diferentes para testar o melhor resultado, por isto foi estimado a metade do valor, a webcam não foi considerada parte do braço por isso seu custo não foi adicionado, a partir destas considerações obtém-se o valor de R\$155,00. O projeto possuía um limite de tempo para ser feito, por isso fornecedores locais foram usados, mas ao se pesquisar por melhores fornecedores, pode-se reduzir drasticamente este valor, pesquisas sites de comércio online chineses foi encontrado os servomotores e o Arduino por aproximadamente R\$77,00 com frete incluso, ao somar com os parafusos comprados localmente, o valor fica de R\$87,00, reduzindo o valor quase pela metade.

Tabela 2. Custos

Produto	preço
Servomotores	R\$35,00 x 3
Arduino	R\$40,00
Impressão 3D	desconsiderado
Parafusos	R\$10,00
Total	R\$155,00

Fonte: autoria própria.

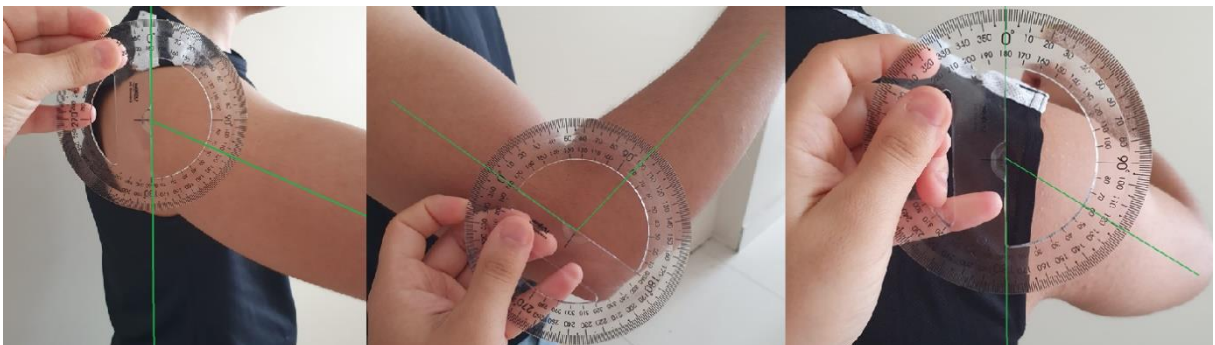
O cálculo dos ângulos foi inicialmente realizado através de um referencial no corpo para calcular a distância entre os pontos, e assim, o ângulo formado por eles. Contudo, essa estratégia não considera as variações de posicionamento de pessoa para pessoa. Por essa razão, foram introduzidos cálculos vetoriais com direção e sentido para definir o ângulo entre os pontos. Esses vetores ainda podem variar com o movimento do corpo, ocasionando imprecisão. A solução encontrada foi a adição de vetores auxiliares, conforme abordado na Seção 3.4.

Nos testes iniciais, os ângulos do braço definidos pelo programa eram elevados e além do limite de atuação do servo motor. Essa ação danificou-o, gerando pequenas fraturas na engrenagem interna. Assim, foi necessário estabelecer via código um limite máximo para grandes valores de ângulos do braço.

As medições dos ângulos foram realizadas com auxílio de um transferidor de acrílico com uma resolução de um grau. Os ângulos definidos estão sujeitos a erros devido ao pequeno movimento do corpo durante conferência do valor, à geometria do braço humano e à dificuldade em definir o ponto médio do braço.

Apesar dos possíveis erros de medição, como o projeto é uma prova de conceito, a análise desses resultados é aceitável para estudar a precisão do protótipo. A Figura 37 ilustra como foram realizadas as medições, as Tabelas 3, 4 e 5 apresentam as medições dos ângulos reais e do protótipo. As médias de erro e desvios padrões dos valores são definidos na Tabela 6.

Figura 37. Medições dos ângulos



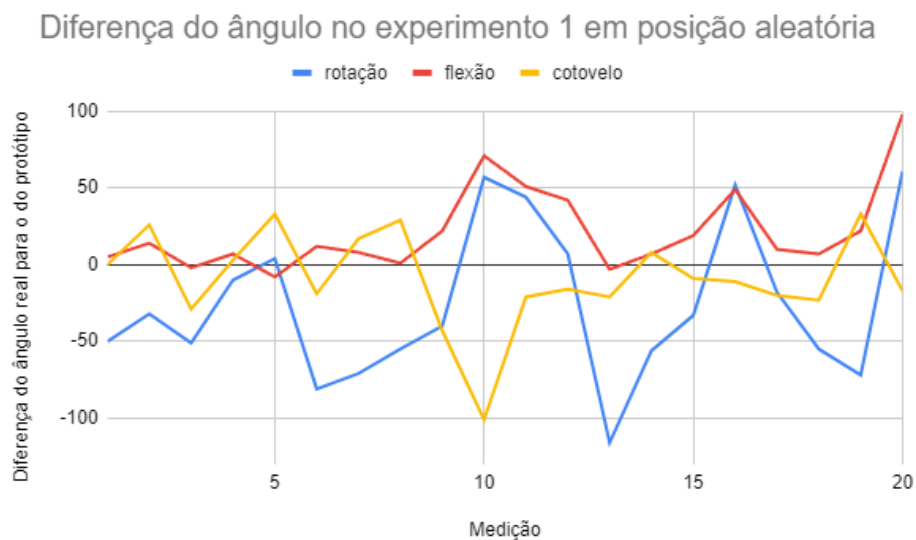
Fonte: autoria própria.

Nota-se que o ângulo de rotação apresenta os maiores erros. Esse fato se deve a grande sequência de cálculos utilizada para definir seu valor. Quanto maior a quantidade de processos, maior a imprecisão no valor final. Além disso, alguns dos

vetores definidos são relativamente pequenos, dificultando a análise da mudança na posição entre dois pontos. Ainda que o tamanho dos vetores não seja considerado, a mudança linear da posição de um ponto causa incerteza em sua direção devido ao acúmulo de imprecisão no erro do ângulo de rotação.

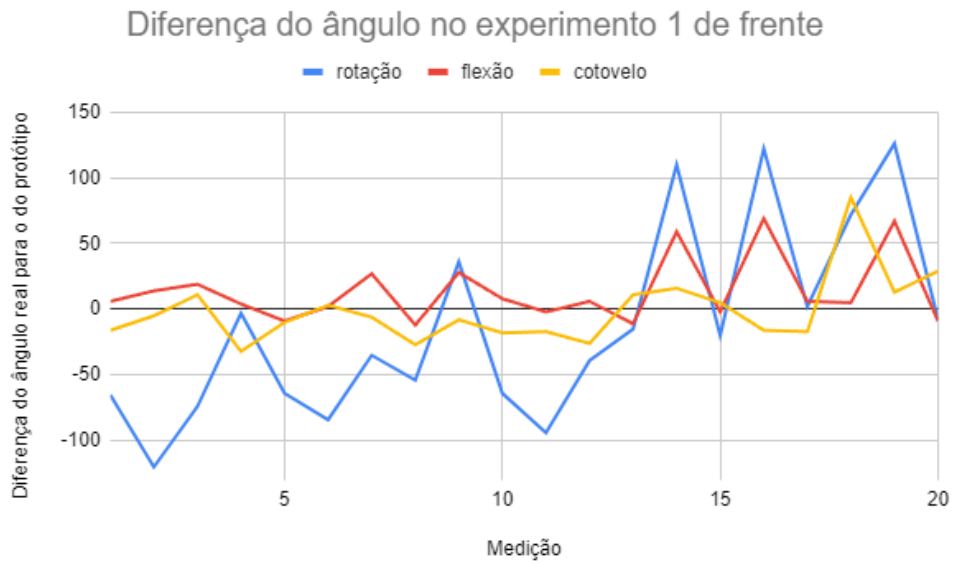
Foram realizados 2 experimentos, no primeiro foram executados 3 tipos de medições para se obter uma amostra de dados mais ampla: 1- o voluntário tomou a posição que preferir e foram realizadas as medições; 2 - foi pedido que o voluntário ficasse de frente para a câmera e 3 - foi estabelecida a posição de lado para a webcam. Assim, foi obtida uma amostra de dados mais diversificada que possibilita uma análise mais profunda, para melhor visualização dos dados, as Figuras 38, 39 e 40 mostram gráficos da diferença do ângulo real e do protótipo.

Figura 38. Diferença do ângulo no experimento 1 em posição aleatória



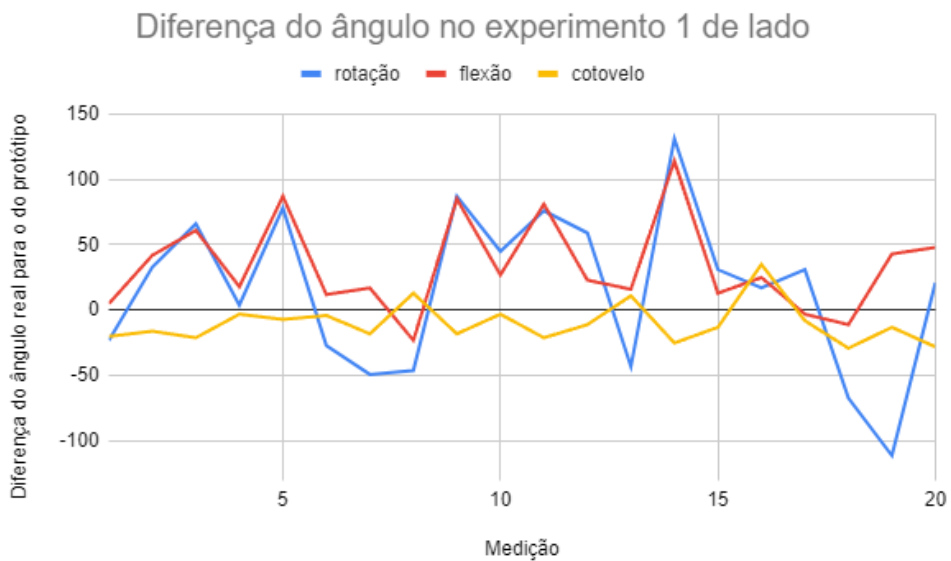
Fonte: autoria própria.

Figura 39. Diferença do ângulo no experimento 1 de frente



Fonte: autoria própria.

Figura 40. Diferença do ângulo no experimento 1 de lado



Fonte: autoria própria.

Tabela 3. Medições dos ângulos de rotação no experimento 1.

Posição aleatória		De frente		De lado	
Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo
20°	70°	5°	70°	-20°	3°
10°	42°	-30°	90°	25°	-8°
20°	71°	-10°	64°	60°	-6°
15°	25°	50°	53°	0°	-4°
65°	61°	10°	74°	70°	-8°
0°	81°	0°	84°	-25°	2°
-20°	51°	15°	50°	-50°	-1°

15°	70°	-5°	49°	-45°	1°
20°	60°	60°	24°	80°	-7°
80°	23°	10°	74°	40°	-5°
70°	26°	-5°	89°	70°	-6°
45°	38°	10°	49°	55°	-4°
-35°	81°	35°	50°	-35°	8°
25°	81°	135°	25°	130°	-1°
30°	63°	5°	25°	30°	-1°
75°	23°	140°	18°	15°	-2°
10°	28°	10°	8°	35°	4°
-10°	45°	90°	18°	-60°	7°
-10°	62°	120°	-6°	-110°	1°
120°	59°	25°	33°	20°	-1°

Fonte: autoria própria.

Tabela 4. Medições dos ângulos de adução no experimento 1.

Posição aleatória		De frente		De lado	
Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo
65°	60°	60°	54°	15°	10°
70°	56°	50°	36°	50°	8°
25°	27°	80°	61°	80°	19°
10°	3°	40°	36°	30°	12°
35°	43°	10°	19°	90°	3°
60°	48°	5°	3°	30°	18°
45°	37°	80°	53°	35°	18°
60°	59°	55°	67°	50°	73°
80°	58°	80°	52°	85°	0°
90°	19°	40°	32°	35°	8°
90°	39°	35°	37°	85°	4°
60°	18°	25°	19°	40°	17°
30°	33°	45°	56°	40°	24°
35°	28°	110°	51°	140°	26°
80°	61°	15°	17°	15°	2°
90°	41°	115°	46°	80°	55°
40°	30°	25°	19°	25°	28°
30°	23°	90°	85°	35°	46°
55°	33°	110°	43°	100°	57°
130°	32°	60°	69°	80°	32°

Fonte: autoria própria.

Tabela 5. Medições dos ângulos do cotovelo no experimento 1.

Posição aleatória		De frente		De lado	
Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo	Ângulo real	Ângulo protótipo
110°	110°	100°	116°	140°	160°
140°	114°	110°	115°	105°	121°
80°	109°	120°	109°	130°	151°
170°	167°	100°	132°	110°	113°
160°	127°	120°	130°	100°	107°

100°	119°	170°	167°	105°	109°
150°	133°	110°	116°	95°	113°
140°	111°	70°	97°	110°	97°
50°	93°	110°	118°	140°	158°
50°	151°	120°	138°	60°	63°
110°	131°	100°	117°	145°	166°
130°	146°	90°	116°	70°	81°
105°	126°	125°	114°	150°	139°
150°	142°	135°	119°	50°	75°
100°	109°	115°	110°	150°	163°
100°	111°	80°	96°	150°	115°
80°	100°	125°	142°	110°	118°
120°	143°	170°	85°	60°	89°
170°	137°	140°	127°	100°	113°
55°	72°	130°	101°	110°	138°

Fonte: autoria própria.

Observa-se que no primeiro experimento, as primeiras medições dos ângulos de flexão e cotovelo possuem média de erro 50% menor que os ângulos de rotação. Nas segundas medições, nas quais o voluntário ficou de frente para a webcam, os resultados para os ângulos de adução e do cotovelo foram melhores. Por outro lado, o ângulo de rotação teve o erro mais elevado. Essa variação pode ser justificada pela quantidade limitada de amostras.

Na terceira medição, com o voluntário posicionado de lado para a webcam, houve uma expressiva melhora no ângulo do cotovelo. Nota-se, então, que a captação de imagem lateral do corpo favorece a captura dos pontos para o cálculo do ângulo do cotovelo. Contudo, o ângulo de adução piorou em relação ao experimento inicial.

Não só o erro médio mudou com os experimentos, o desvio padrão também variou ao longo dos resultados, diminuindo e aumentando junto ao erro médio. Ao analisar os dados chega-se à conclusão de que a orientação do corpo influencia na captação dos pontos do corpo, o que muda o resultado dos ângulos.

Para se obter os melhores resultados nas medições de ângulo do cotovelo deve-se captar a imagem de lado. Enquanto, a imagem de frente a webcam aprimora a medição do ângulo de adução. No caso do ângulo de rotação, a posição intermediária, na qual a orientação do corpo forme algum ângulo com a webcam, obtém a medição mais fidedigna. Como o cálculo do ângulo de rotação é o único que é definido pela projeção de um vetor num plano, percebe-se que essa projeção propaga erros ao determinar o ângulo.

Na tabela 6 é possível ver que apesar de possuir média de erro menor, o desvio padrão do ângulo de adução é similar ao de rotação. Esse fato é justificado pelos cálculos adicionais dos vetores de altura e de ombro. Devido à diferença de largura entre ombro e cintura, uma inclinação indesejada é gerada, sendo necessário definir esses vetores dois para determinar o ângulo de adução. Essa manipulação evita imprecisão dos dados, reduzindo a média de erro, porém o desvio padrão é propagado pela quantidade de pontos necessária ao cálculo.

Tabela 6. Média de erro e desvio padrão dos resultados do experimento 1.

Diferença entre os valores reais e protótipo									
Posição aleatória			De frente			De lado			
rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo	
50°	5°	0°	65°	6°	16°	23°	5°	20°	
32°	14°	26°	120°	14°	5°	33°	42°	16°	
51°	2°	29°	74°	19°	11°	66°	61°	21°	
10°	7°	3°	3°	4°	32°	4°	18°	3°	
4°	8°	33°	64°	9°	10°	78°	87°	7°	
81°	12°	19°	84°	2°	3°	27°	12°	4°	
71°	8°	17°	35°	27°	6°	49°	17°	18°	
55°	1°	29°	54°	12°	27°	46°	23°	13°	
40°	22°	43°	36°	28°	8°	87°	85°	18°	
57°	71°	101°	64°	8°	18°	45°	27°	3°	
44°	51°	21°	94°	2°	17°	76°	81°	21°	
7°	42°	16°	39°	6°	26°	59°	23°	11°	
116°	3°	21°	15°	11°	11°	43°	16°	11°	
56°	7°	8°	110°	59°	16°	131°	114°	25°	
33°	19°	9°	20°	2°	5°	31°	13°	13°	
52°	49°	11°	122°	69°	16°	17°	25°	35°	
18°	10°	20°	2°	6°	17°	31°	3°	8°	
55°	7°	23°	72°	5°	85°	67°	11°	29°	
72°	22°	33°	126°	67°	13°	111°	43°	13°	
61°	98°	17°	8°	9°	29°	21°	48°	28°	
Média do erro	48,25	22,9	23,95	60,35	18,25	18,55	52,25	37,7	15,85
Desvio padrão	26,93	26,16	21,02	40,42	21,51	17,70	32,47	31,88	8,98
Média do erro %	27%	13%	13%	34%	10%	10%	29%	21%	9%

Fonte: autoria própria.

No segundo experimento, com o objetivo de estudar mais a interferência da posição relativa da câmera. O voluntário posicionou o braço na mesma posição e rotacionou o corpo para mudar a direção relativa da câmera, mantendo 2 ângulos do braço constantes e movendo apenas 1 por vez, assim isolando a variável a ser estudada, desta forma foi coletado dados dos ângulos do braço e comparado com o resultado obtido pelo programa. O mesmo experimento foi realizado 3 vezes, onde o voluntário para o experimento foi alterado em 2 medições, assim foi possível analisar o erro que ocorre ao mudar de pessoa. Os resultados podem ser vistos nas Tabelas 7, 8 e 9.

Ao analisar os dados, é possível inferir que para o ângulo de rotação, as medições são mais precisas em ângulos menores e a posição relativa a câmera influenciou minimamente. Enquanto, o ângulo de adução do ombro obteve melhores resultados quando de frente para a câmera. Já nas medições do ângulo de flexão do cotovelo, os resultados variaram bastante tornando inconclusivo a melhor forma de captar seu movimento.

Ao comparar os erros entre as medições feitas no primeiro voluntário com as feitas no segundo voluntário, observa-se uma diferença nos erros maior que ao comparar as duas medições feitas no segundo voluntário, como os ângulos do experimento se mantiveram iguais, o que variou neste experimento foi o porte físico dos voluntários, o que leva à conclusão de que o porte físico da pessoa tem influência na captação dos pontos.

Ao observar os resultados nas Tabelas 7, 8 e 9, nota-se que houve variação dos ângulos que deveriam estar iguais. Assim, conclui-se que a maior fonte de erro do projeto é a captação da posição dos pontos do braço.

Tabela 7. Erro na variação da posição relativa à câmera no experimento 2, medição 1.

ângulo real			Diferença do ângulo real e calculado								
			frente			inclinado			lado		
Rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo
0°	0°	170°	18°	4°	5°	4°	3°	3°	2°	1°	1°
45°	0°	170°	50°	4°	4°	47°	12°	12°	53°	3°	3°
90°	0°	170°	55°	18°	36°	93°	9°	10°	97°	9°	9°
0°	0°	170°	18°	4°	5°	4°	3°	3°	2°	1°	1°
0°	45°	170°	48°	16°	29°	7°	27°	18°	4°	37°	8°
0°	90°	170°	80°	8°	82°	6°	15°	75°	6°	15°	75°
0°	0°	120°	29°	31°	19°	8°	57°	7°	1°	36°	61°
0°	0°	90°	88°	57°	23°	2°	45°	35°	2°	59°	21°
0°	0°	55°	41°	49°	28°	4°	57°	58°	5°	68°	47°

Fonte: autoria própria.

Tabela 8 .Erro na variação da posição relativa à câmera no experimento 2, medição 2.

ângulo real			Diferença do ângulo real e calculado								
			frente			inclinado			lado		
Rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo
0°	0°	170°	34°	3°	3°	8°	0°	0°	3°	3°	3°
45°	0°	170°	1°	2°	3°	46°	0°	0°	48°	2°	2°
90°	0°	170°	77°	8°	14°	98°	0°	0°	99°	13°	13°
0°	0°	170°	34°	3°	3°	8°	0°	0°	3°	3°	3°
0°	45°	170°	80°	7°	38°	5°	45°	0°	0°	30°	15°
0°	90°	170°	61°	10°	80°	2°	90°	0°	1°	35°	55°
0°	0°	120°	29°	6°	41°	1°	0°	14°	7°	6°	24°
0°	0°	90°	42°	9°	43°	3°	0°	13°	5°	1°	31°
0°	0°	55°	1°	9°	54°	3°	0°	44°	7°	9°	16°

Fonte: autoria própria.

Tabela 9. Erro na variação da posição relativa à câmera no experimento 2, medição 3.

ângulo real			Diferença do ângulo real e calculado								
			frente			inclinado			Lado		
Rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo	rotação	adução	cotovelo
0°	0°	170°	18°	4°	4°	1°	9°	9°	2°	7°	7°
45°	0°	170°	40°	33°	33°	45°	19°	19°	48°	6°	6°
90°	0°	170°	97°	14°	14°	96°	3°	3°	96°	6°	6°
0°	0°	170°	18°	4°	4°	1°	9°	9°	2°	7°	7°
0°	45°	170°	65°	18°	27°	1°	37°	8°	4°	33°	12°
0°	90°	170°	62°	11°	79°	2°	29°	61°	3°	15°	75°
0°	0°	120°	4°	8°	32°	0°	1°	17°	4°	7°	27°
0°	0°	90°	21°	5°	16°	3°	7°	19°	5°	10°	23°
0°	0°	55°	8°	12°	48°	6°	5°	60°	5°	15°	36°

Fonte: autoria própria.

5 CONCLUSÃO

O desenvolvimento de um código computacional para manipulação de um braço robótico por captura de movimento foi realizado conforme descrito. Como uma prova de conceito de baixo custo, custando R\$155,00, o trabalho satisfaz aos objetivos iniciais do projeto, contendo alguns problemas que podem ser resolvidos em trabalhos futuros, ou melhorados. O projeto possui estrutura, *hardware* e *software* compatíveis com componentes difundidos no mercado como Arduino e servo motor SG90.

O protótipo encontrou situações adversas na montagem mecânica sendo necessárias pequenas adaptações, como por exemplo, a adição de uma porca extra. Além da abertura para o encaixe do servo motor. Esses problemas são passíveis em qualquer montagem mecânica e foram solucionados.

Para a elaboração do código foram utilizadas bibliotecas fundamentais como *mediapipe* e *serial*. A biblioteca *mediapipe* permitiu a captura dos pontos do corpo humano, enquanto a *serial* possibilitou a comunicação com o Arduino. A captura e da imagem pela câmera foi realizada através da biblioteca *cv2*. Para facilitar o cálculo e reduzir a complexidade do código utilizou-se as bibliotecas *numpy* e *cmath*.

Essas duas bibliotecas também auxiliaram a aquisição dos pontos e a definição dos vetores para obter os ângulos necessários. Com essas informações, os dados de posição são enviados ao Arduino para movimentar os servomotores do braço robótico. O protótipo replica, assim, a posição do usuário capturada pela câmera.

Apesar do esperado, a iluminação não representou dificuldades na definição dos pontos. Uma iluminação de lâmpada elétrica comum permitiu um resultado aceitável independente da iluminação natural do ambiente.

Após a montagem do protótipo de braço robótico, foram realizadas medições dos ângulos de rotação, de adução e de cotovelo do braço real a fim de identificar a precisão do braço robótico. As medições foram feitas utilizando um transferidor com uma resolução de 1 grau. Essas são sujeitas a erros devido ao leve movimento do corpo durante conferência do valor, à geometria do braço humano e à dificuldade de definir o ponto médio do braço.

Com base nos resultados apresentados nas Tabelas 2-9, é possível observar que a orientação do corpo influencia o resultado devido a imprecisão na captura dos pontos do corpo, cada ângulo possui seu melhor posicionamento relativo a webcam para captar os pontos, onde os melhores resultados obtidos foram de um erro de 27% para o ângulo de rotação ao ficar posicionado a um ângulo de aproximadamente 45° da *webcam*, 10% para o ângulo de adução ao ficar de frente para a webcam e 9% para o ângulo do cotovelo ao ficar de lado para webcam.

O elevado erro do ângulo de rotação é justificado pelo método de cálculo que possui mais processos do que os demais. Além do tamanho reduzido dos vetores, o que dificulta a análise na alteração do movimento. Ainda, a imprecisão dos resultados pode ser causada por fatores como a resolução e a distância da câmera, diferença

física do corpo do voluntário, e principalmente pela imprecisão na coleta dos pontos para realizar os cálculos.

5.1 TRABALHOS FUTUROS

Após a conclusão do projeto de manipulação do braço robótico por captura de movimento, é possível sugerir trabalhos futuros para complementar e melhorar o projeto desenvolvido

- Remodelar o braço robótico para reduzir suas limitações.
- Utilizar outra forma de captura de movimento como *knetic* de *Xbox* para obter uma melhor captura dos pontos.
- Melhorar a estrutura de cálculos dos ângulos a fim de obter maior precisão.
- Melhorar a configuração da biblioteca de captura de movimento para ter maior precisão na identificação dos pontos do corpo humano.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSUNÇÃO, Jorge. ENTENDA A OSTEOCONDRITE DISSECANTE DO COTOVELO. Disponível em: <https://ortopediaeombro.com.br/osteocondrite-dissecante-do-cotovelo/>. Acesso em: 22 de janeiro de 2023.

BABICH, Nick. What is computer vision & how does it work? an introduction. Adobe. Disponível em: <https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/>. Acesso em: 27 de abril de 2022.

BECKER, Djeily Tamara. VISÃO COMPUTACIONAL APLICADA EM UM BRAÇO ROBÓTICO ANTROPOMÓRFICO DIDÁTICO. Revista Thêma et Scientia, v. 7, n. 1, p. 87-104, 2017.

CARRARA, Valdemir. Introdução à robótica industrial. INPE–Instituto Nacional de Pesquisas Espaciais, p. 1-101, 2015.

COELHO, Renato Ramos; DANTAS, Estélio Henrique Martin; CUNHA, Gerson Gomes. USO DO CONJUNTO BENDSENSOR/ARDUINO PARA A CAPTURA DE MOVIMENTO DO PUNHO-MÃO. 2012.

CORRALLO, Marcio Vinicius; JUNQUEIRA, A. de C. A Lei de esfriamento de Newton utilizando a automatização da tomada dos dados por meio do Arduino. **XXI Simpósio Nacional de Ensino de Física**, 2015.

DE MILANO, Danilo; HONORATO, Luciano Barrozo. Visao computacional. 2014.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em ação. 1 ed. Connecticut: Greenwich, 2013.

GROOVER, Mikell P. Automação industrial e sistemas de manufatura. Pearson Education do Brasil, 2011.

HALL, Susan J. Biomecânica básica. 7ª edição. Rio de janeiro. Editora Guanabara Koogan. 2016.

HENRIQUES, Marcos T. V. Avaliação fisioterapêutica do punho e da mão. Disponível em: <https://silo.tips/download/e-uma-artic-de-eixo-uniaxial-que-possui-um-grau-de-liberdade-posiao-de-reposuo-1>. Acesso em: 22 de janeiro de 2023.

INTERNATIONAL, Federation of Robotics - IFR. World Robotics 2022. Alemanha. 2022. Disponível em: https://ifr.org/downloads/press2018/2022_WR_extended_version.pdf. Acesso em: 25 de fevereiro de 2023.

ITMAN FILHO, André; SILVA, Rosana Vilarim da; BERTOLO, Roney Valdo. 6 Aixs braço robótico mecânica de alta precisão para carga e descarga com marcação ISO Jzj06c-144. **Dental Press Journal of Orthodontics**, v. 16, n. 4, p. 55-59, 2011.

KAPANDJI, I. A. **Fisiologia articular: esquemas comentados de mecânica humana**. 6. ed. Rio de Janeiro: Guanabara Koogan, 2007. v. 1.

LUGARESI, Camillo et al. Mediapipe: A framework for perceiving and processing reality. In: **Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)**. 2019.

MARENGONI, Maurício; STRINGHINI, Stringhini. Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada*, 2009, 16.1: 125-160.

MOURA, Aloisio Shelldon Rocha. Projeto de automação residencial simplificado: uma proposta para pessoas com necessidades especiais (PNE) de baixa renda. 2020.

NORTON, Robert L. Cinemática e dinâmicas dos mecanismos. 1ª edição. Porto alegre: AMGH, 2010.

OLIVEIRA, Eduardo Lino de. **Construção de um braço robótico didático com três graus de liberdade controlado por Arduino**. 2014. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

ROSARIO, J. M. Robótica Industrial I: Modelagem, utilização e programação. São Paulo: Baraúna, 2010.

SHEPLEY, Phil. What is a Robotic Arm?. EasyTecJunkie. Disponível em: <<https://www.easytechjunkie.com/what-is-a-robotic-arm.htm>>. Acesso em: 27 de abril de 2022.

SILVA, Guilherme Schmitt da. Detecção de movimento na automação dos testes de produção de câmeras CFTV. Engenharia Elétrica Telemática-Pedra Branca, 2011.

TANNUS, Alexandre Moraes. Arduino: Comunicação serial. 2018.

TANNUS, Alexandre Moraes. Arduino: Servomotores. 2018.

APÊNDICE A – Código do Arduino que move os motores

```
//UFPE – DEMEC – TCC – 2021.2
```

```
//Aluno : wilson campos de almeida bisneto
```

```
//objetivo: desenvolver o codigo que recebe instruções e move o braço robotico
```

```
#include <Servo.h>
```

```
// novo nome numero do pino
```

```
#define MOTOR1 6
```

```
#define MOTOR2 7
```

```
#define MOTOR3 8
```

```
int angulo1 = 140;
```

```
int angulo2 = 145;
```

```
int angulo3 = 120;
```

```
bool dadosSalvos = false;
```

```
bool caractereInvalido = false;
```

```
bool comandoIniciado = false;
```

```
int lugarDoCaracter = 0;
```

```
char caractere = ' ';
```

```
long tempoDeResposta = 0;
```

```
char comando [16] = "";
```

```
int j = 0;
```

```
Servo ombroRotacao;
```

```
Servo ombroFlexao;
```

```
Servo cotovelo;
```

```
void setup() {

    //linka os pinos dos motores com suas variaveis
    ombroRotacao.attach(MOTOR1);
    ombroFlexao.attach(MOTOR2);
    cotovelo.attach(MOTOR3);

    //espera a porta serial ser conectada
    while (!Serial){

    }

    Serial.begin(9600);

    //inicia o braço numa posição inicial
    ombroRotacao.write(140);
    ombroFlexao.write(145);
    cotovelo.write(120);
}

void loop(){
    if(dadosSalvos){
        identificaComando();
        dadosSalvos = false;
    }
}

//função que vai mover os motores do braço
void moveMotor(int angulo1, int angulo2, int angulo3){
    ombroRotacao.write((180-angulo1));
```

```

ombroFlexao.write((180-angulo2));
cotovelo.write((angulo3));
}

```

```

void zeraVetor(){
  for (j = 0; j <= 15; j++){
    comando[j] = ' ';
  }
}

```

//exemplo de comando IA140B145C120F

```

void identificaComando(){
  //valida o comando
  if(comando[0]=='I'){
    if(comando[1] == 'A' && comando[5] == 'B' && comando[9] == 'C'){
      //salva os caracteres dos angulos em uma string
      angulo1 = ((comando[2] - '0')*100)+((comando[3] - '0')*10)+(comando[4] - '0');
      angulo2 = ((comando[6] - '0')*100)+((comando[7] - '0')*10)+(comando[8] - '0');
      angulo3 = ((comando[10] - '0')*100)+((comando[11] - '0')*10)+(comando[12] - '0');

      moveMotor(angulo1, angulo2, angulo3);
      dadosSalvos = false;
    }
  }
  else{
    Serial.print("INACKF");
  }
}

```

//salva o comando recebido


```
void serialEvent(){
    if(!dadosSalvos){
        caracter = Serial.read();

        //caso receba uma letra I no buffer, dara inicio ao armazenamento do comando
        if(caracter == 'I' && !comandoIniciado){
            lugarDoCaracter = 0;
            comandoIniciado = true;
            tempoDeResposta = millis();
        }

        if(comandoIniciado && (millis() - tempoDeResposta <= 500)){
            //salva o comando em um vetor de caracteres
            comando[lugarDoCaracter] = caracter;
            lugarDoCaracter++;

            //caso a letra F seja lida, dará fim ao armazenamento do comando
            if(caracter == 'F'){
                //corrige o lugar do ultimo caracter (é adicionado +1 antes de conferir se foi o
                ultimo)
                lugarDoCaracter--;
                comandoIniciado = false;
                dadosSalvos = true;
            }
        }
        else{
            Serial.println("INACKF");
            zeraVetor();
            comandoIniciado = false;
        }
    }
}
```

}

APÊNDICE B – Código que comanda o braço

```
#UFPE – DEMEC – TCC – 2021.2
#Aluno : wilson campos de almeida bisneto
#objetivo: desenvolver o código que utiliza webcam para captar o movimento do
#braço e enviar instruções para mover o prototipo

from cmath import sqrt
import cv2
import mediapipe as mp
import numpy as np
#import matplotlib as mpl
#from mpl_toolkits import mplot3d
#import matplotlib.pyplot as plt
import serial
import serial.tools.list_ports
from sympy import Eq, Symbol, solve

#iniciando comunicação serial, a porta pode mudar dependendo do computador
ser=serial.Serial("COM4", 9600, timeout=0.5)

#fecha a comunicação caso ja esteja iniciada e inicia uma nova para evitar
erro
ser.close()
ser.open()
```

```

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_pose = mp.solutions.pose

#retorna o vetor normal a 2 angulos--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**
def produtoVetorial(u, v):
    #    i j k
    # u  a b c
    # v  x y z

    # <u,v> = (xcj + bzi + ayk) - (zaj + bxk + cyi)
    # <u,v> = [(bz-cy), (cx - az), (ay - bx)]

    vetor_normal = [(u[1]*v[2]-u[2]*v[1]), (u[2]*v[0] - u[0]*v[2]), (u[0]*v[1] -
u[1]*v[0])]

    return vetor_normal

#retorna o angulo formado por 2 vetores--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**
**_**
def produtoInterno(u, v):
    #produto interno <u,v>
    #(vx1 * vx2) + (vy1 * vy2) + (vz1 * vz2)
    produto_interno = (u[0] * v[0]) + (u[1] * v[1]) + (u[2] * v[2])

    #norma do vetor |u|, |v|
    u_n = sqrt((u[0]**2) + (u[1]**2) + (u[2]**2))
    v_n = sqrt((v[0]**2) + (v[1]**2) + (v[2]**2))

    cos_angle = produto_interno/(u_n*v_n)
    angle = np.arccos(cos_angle)

```

```

angle = 180-(angle*180)/3.1415926535

return angle

#transforma o valor numerico do angulo em uma string de 3 digitos--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--
def digitos(angulo):

    digito3 = angulo%10
    digito2 = int((angulo%100)/10)
    digito1 = int((angulo%1000)/100)

    numero = str(digito1) + str(digito2) + str(digito3)

    return numero

#calcula a projeção do ponto num plano--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--
def projecao(vetor_normal_plano, ponto):

    #criando variavel de lambda (h -> lambda)
    h = Symbol('h')

    #obtendo as equações parametricas da reta perpendicular ao plano que passa
    pelo ponto desejado

    # x = x_ponto + (x_vetor * lambda)
    # y = y_ponto + (y_vetor * lambda)
    # z = z_ponto + (z_vetor * lambda)

    x = ponto[0] + vetor_normal_plano[0] * h
    y = ponto[1] + vetor_normal_plano[1] * h
    z = ponto[2] + vetor_normal_plano[2] * h

```

```

#assumindo um plano qualquer (ax+by+cz=0) para criar a projeção
#calcular o valor de lambda jogando os valores de x,y,z na equação do plano

eqa = Eq((vetor_normal_plano[0])*(ponto[0] + vetor_normal_plano[0] *
h)+(vetor_normal_plano[1])*(ponto[1] + vetor_normal_plano[1] *
h)+(vetor_normal_plano[2])*(ponto[2] + vetor_normal_plano[2] * h), 0)

lamb = solve(eqa)
lamb = lamb[0]

# substituindo lambda nas equações parametricas

ponto_projetado = [0, 0, 0]
ponto_projetado[0] = float(ponto[0] + vetor_normal_plano[0] * lamb)
ponto_projetado[1] = float(ponto[1] + vetor_normal_plano[1] * lamb)
ponto_projetado[2] = float(ponto[2] + vetor_normal_plano[2] * lamb)

return ponto_projetado

#captura dos pontos--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**
cap = cv2.VideoCapture(0)

with mp_pose.Pose(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as pose:
while cap.isOpened():
    success, image = cap.read()
    altura, largura, _ = image.shape

    if not success:
        print("Ignoring empty camera frame.")
        continue

#colocar .writeable = false para melhorar performance
image.flags.writeable = False

```

```

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
results = pose.process(image)

#desenha as anotações na imagem
image.flags.writeable = True
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
mp_drawing.draw_landmarks(
    image,
    results.pose_landmarks,
    mp_pose.POSE_CONNECTIONS,
    landmark_drawing_spec=mp_drawing_styles.get_default_pose_landmarks_style())

#salva os pontos do corpo em um vetor com suas coordenadas
pontos = []
if results.pose_world_landmarks:
    for id, lm in enumerate(results.pose_world_landmarks.landmark):
        pontos.append([id, (lm.x), (lm.y), (lm.z)])

#o numero dos pontos se refere ao numero que a biblioteca salva o ponto
#https://google.github.io/mediapipe/solutions/pose#min_detection_confidence
ombro_right = pontos[12]
cintura_right = pontos[24]
cotovelo_right = pontos[14]
mao_right = pontos[16]
cintura_left = pontos[23]

#representando as partes do corpo como vetores
braco = [(cotovelo_right[1] - ombro_right[1]), (cotovelo_right[2] - ombro_right[2]), (cotovelo_right[3] - ombro_right[3])]
antebraco = [(mao_right[1] - cotovelo_right[1]), (mao_right[2] - cotovelo_right[2]), (mao_right[3] - cotovelo_right[3])]

```

```

    corpo = [(ombro_right[1] - cintura_right[1]), (ombro_right[2] -
cintura_right[2]), (ombro_right[3] - cintura_right[3])]

    cintura = [(cintura_right[1] - cintura_left[1]), (cintura_right[2] -
cintura_left[2]), (cintura_right[3] - cintura_left[3])]

    #fator de correção--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**
    #devido a imprecisão na coleta dos pontos, o braço faz o movimento na
direção desejada, mas o ângulo é menor do que o esperado

    #devido a este erro, será implementado um fator de correção que aumenta
o ângulo artificialmente

    #o fator de correção foi determinado experimentalmente

    fator_de_correcao = 1.2

    #ângulo flexão do ombro--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**
    #calcula o ângulo que o braço faz com o vetor do ombro

    #calcular o vetor "perpendicular ao chão (altura)", o referencial é o
corpo, mas assim facilita a entender que vetor estou falando

    vetor_normal_corpo = produtoVetorial(cintura, corpo)

    #o vetor altura seria o vetor do corpo, mas como o vetor do corpo possui
uma pequena inclinação devido a largura dos ombros

    #foi preciso calcular o vetor altura para obter um vetor mais adequado

    vetor_altura = produtoVetorial(vetor_normal_corpo, cintura)

    #calcular o ângulo que a altura faz com o vetor perpendicular ao corpo
para obter o vetor do "ombro"

    #não dá para usar diretamente o vetor formado pelos pontos dos ombros
pois os ombros possuem um pouco de

    #movimento relativo ao corpo (ex mover o ombro para frente e para trás),
este movimento pode alterar a precisão do resultado

```

```

vetor_ombro= produtoVetorial(vetor_normal_corpo, vetor_altura)

#calcular o angulo que o braço faz com o ombro
#não precisa de fator de correção, o resultado esta satisfatorio
angulo_flexao = abs(90 - produtoInterno(vetor_ombro, braco))

#a conta de abs(90-angulo) é precisa pois o produto interno calcula o
angulo complementar do angulo desejado

#o abs acredito ser desnecessario mas é possivel que em algum teste
tenha dado um valor negativo, então vou mante-lo

#proteção em caso de algo inesperado
if(angulo_flexao > 180):
    angulo_flexao = 180
    print("erro: angulo de flexão acima de 180")

if(angulo_flexao < 0):
    angulo_flexao = 0
    print("erro: angulo de flexão abaixo de 0")

#angulo cotovelo (vai de 0-90 graus por limitação do prototipo)--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--
#angulo entre o braço e antebraço
angulo_cotovelo = produtoInterno(braco, antebraço)

#fator de correção
if(angulo_cotovelo >= 45):
    angulo_cotovelo = 45+(angulo_cotovelo-45)*fator_de_correcao
else:
    angulo_cotovelo = 45-(45-angulo_cotovelo)*fator_de_correcao

#limitação do angulo do prototipo

```



```

if(angulo_cotovelo > 170):
    print("erro: angulo do cotovelo acima de 170: ", int(angulo_cotovelo))
    angulo_cotovelo = 170

if(angulo_cotovelo <= (angulo_flexao + 10)):
    angulo_cotovelo = (angulo_flexao + 10)
    print("erro: cotovelo muito proximo do braco")

#faz o angulo do cotovelo ter o referencial do angulo do ombro
#o modelo do projeto cria esta necessidade
if(angulo_cotovelo + angulo_flexao < 170):
    angulo_cotovelo = angulo_cotovelo + angulo_flexao
else:
    #print(angulo_cotovelo, angulo_flexao)
    angulo_cotovelo = 170
    print()

#proteção em caso de algo inesperado
if(angulo_cotovelo < 0):
    angulo_cotovelo = 0
    print("erro: angulo do cotovelo abaixo de 0")

#angulo rotação do ombro--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--
**--**--**--**--**--**
    #calcular o a projeção de 2 pontos do vetor do braço (ombro e cotovelo)
no plano

#vetor normal ao plano de projeção
vetor_normal_plano = produtoVetorial(vetor_normal_corpo, vetor_altura)
projecao1 = projecao(vetor_normal_plano, ombro_right)
projecao2 = projecao(vetor_normal_plano, cotovelo_right)

#calcular uma projeção do vetor do braço no plano perpendicular ao corpo

```

```

    vetor_projetado = [(projecao2[0]-projecao1[0]), (projecao2[1]-
projecao1[1]), (projecao2[2]-projecao1[2])]

    #calcular o angulo que a projeção do vetor faz com o vetor do corpo
(altura)

    angulo_rotacao = produtoInterno(vetor_projetado, vetor_altura)

    #proteção em caso de algo inesperado

    if(angulo_rotacao > 180):

        angulo_rotacao = 180

        print("erro: angulo de rotação acima de 180")

    if(angulo_rotacao < 0):

        angulo_rotacao = 0

        print("erro: angulo de rotação abaixo de 0")

    #mandar o resultado para o Arduino-----
**__**__**__**__**__**__**__**__**__**__**__**__**__**__**__**__
**__**__**__**__**__**__**__**__**__**__**__**__**__**__**__**
**__**__**__**__**__**__**__**__**__**

    #exemplo de comando IA090B045C060F

    #cria o comando que sera enviado

    anguloA = digitos(int(round(angulo_rotacao)))
    anguloB = digitos(int(round(angulo_flexao)))
    anguloC = digitos(int(round(angulo_cotovelo)))

    comando = "IA" + str(anguloA) + "B" + str(anguloB) + "C" + str(anguloC)
+ "F"

    #imprime o comando para testes

    #print(comando)

    #envia o comando para o Arduino

```

```
try:
    ser.write(comando.encode('utf-8'))

except Exception as e:
    print("---abnormal---: ",e)

# Flip the image horizontally for a selfie-view display.
cv2.imshow('MediaPipe Pose', cv2.flip(image, 1))
if cv2.waitKey(5) & 0xFF == 27:
    break
cap.release()
ser.close()
```