

PROTÓTIPO DE MOVIMENTAÇÃO DA MÃO DO BRAÇO ROBÓTICO COM CONTROLE VIA WI-FI E MEDIAPIPE

PROTOTYPE OF ROBOTIC ARM HAND MOVEMENT CONTROLLED VIA WI-FI AND MEDIAPIPE

Eduardo De Assis Rodrigues Centro Universitário Geraldo Di Biasi, Volta Redonda/RJ, Brasil
eduassis29@gmail.com

Gabriel Ribeiro Dos Santos Centro Universitário Geraldo Di Biasi, Volta Redonda/RJ, Brasil
si.gabrielribeiro@gmail.com

Luiz Henrique Barbosa Da Silva Centro Universitário Geraldo Di Biasi, Volta Redonda/RJ, Brasil
luizbarbosa785@gmail.com

Weber Monteiro Luna Centro Universitário Geraldo Di Biasi, Volta Redonda/RJ, Brasil
weber.m.luna@gmail.com

André Ricardo Prazeres Rodrigues Centro Universitário Geraldo Di Biasi, Volta Redonda/RJ, Brasil
3918@academicougbr.com.br

Resumo O projeto tem como objetivo o desenvolvimento de uma mão em um braço robótico impresso em 3D, controlar a movimentação dos dedos da mão reconhecidos por visão computacional. A proposta integra Python, OpenCV (MediaPipe) e o microcontrolador ESP32, que realiza a comunicação sem fio com os servomotores responsáveis pela movimentação da mão. A iniciativa busca criar um protótipo funcional, de baixo custo e fácil replicação, que poderá ser utilizado em atividades educacionais e experimentais, além de servir como base para pesquisas futuras na área de robótica, automação e inteligência artificial.

Palavras-chave Robótica. Visão computacional. ESP32. Servomotores. Impressão 3D.

Abstract The project aims to develop a hand for a 3D-printed robotic arm and to control the movement of its fingers based on computer vision recognition. The proposal integrates Python, OpenCV (MediaPipe), and the ESP32 microcontroller, which performs wireless communication with the servomotors responsible for the hand's movements. The initiative seeks to create a functional, low-cost, and easily replicable prototype that can be used in educational and experimental activities, as well as serve as a foundation for future research in robotics, automation, and artificial intelligence.

Keywords Robotics. Computer vision. ESP32. Servomotors. 3D printing.



Licença de Atribuição BY do Creative Commons
<https://creativecommons.org/licenses/by/4.0/>

Aprovado em 30/01/2026
Publicado em 30/04/2026

1 INTRODUÇÃO

O avanço da robótica e da inteligência artificial tem possibilitado a criação de sistemas autônomos e interativos. A combinação de um braço robótico impresso em 3D com um sistema de controle de mão utilizando Python e OpenCV (MediaPipe) oferece uma abordagem para interações humanas-robô. O uso do ESP32 permite a comunicação via Wi-Fi, aumentando a flexibilidade do controle.

Sales Júnior (1997) descreveu um projeto e a implementação de um Sistema de Controle Inteligente para Braços Robóticos que pode ser utilizado para testes no ato de agarrar uma escada. Outro trabalho foi a movimentação do braço robótico em pegar ou soltar as peças do tabuleiro. A inteligência implementada foi a lógica Fuzzy (SHHEIBIA, 2001).

Segundo Bigelli (2021), a Ubiquo Robótica, empresa de robótica que tem como seu primeiro produto o braço robótico de mesa UR-1, que tem todas as suas partes mecânicas produzidas via impressão 3D. O trabalho exercido por um braço robótico ou robô manipulador exemplifica a importância da robótica nas indústrias (SILVA et al., 2017).

A robótica tem revolucionado diversas indústrias, automatizando linhas de produção e entregando ganhos expressivos em velocidade, qualidade e redução de erros operacionais (SIQUEIRA BATISTA et al., 2016; AMARAL et al., 2021). No campo da saúde, sua aplicação também tem sido relevante: tecnologias robóticas têm sido empregadas para automatizar tarefas como transporte de materiais em hospitais, proporcionar apoio a pacientes e viabilizar terapias e reabilitação com maior precisão e segurança (SIQUEIRA BATISTA et al., 2016; SILVA et al., 2025). Assim, a robótica contribui tanto para o aumento da eficiência industrial quanto para a modernização dos cuidados médicos.

Os sistemas robóticos de braços articulados reproduzem com precisão os movimentos do cirurgião, eliminando tremores e reduzindo a fadiga. Essa tecnologia permite movimentos delicados e controlados, semelhantes aos da mão humana. Conforme Araujo et al. (2008), a assistência robótica elimina os tremores e movimentos involuntários, assegurando maior precisão nos gestos cirúrgicos.

Reck e Antonello (2021) construíram um braço robótico de código open source, de baixo custo, através de Impressão 3D no campus do IFC Luzerna, equipamento este a ser utilizado em atividades pedagógicas e em iniciativas de pesquisa e extensão universitária. Outro trabalho apresentou uma ferramenta didática de baixo custo, com alguns detalhes construtivos, de um manipulador robótico com quatro graus de liberdade (MENDONÇA et al., 2016).

Neste contexto, o objetivo é desenvolver um sistema integrado que permita o controle do movimento dos dedos de uma mão de um braço robótico, os gestos reconhecidos por um sistema de visão computacional, utilizando Python, OpenCV e ESP32 para comunicação sem fio entre o cliente e o servidor.

2 METODOLOGIA UTILIZADA NA PESQUISA

Além de um levantamento bibliográfico, o estudo consiste em usar uma impressora 3D para a produção da mão do braço robótico aliados a tecnologias como Python, OpenCV e ESP32.

Foi utilizado um projeto de um braço robótico com uma mão articulada (BARDON, 2018), na qual a mão foi impressa em 3D na Creality Ender 3 S1. O sistema eletrônico foi composto por quatro servos MG996R, um micro servo MG90S e uma placa de desenvolvimento ESP32. Em seguida foram desenvolvidas duas aplicações: uma chamada cliente (código de reconhecimento do movimento dos dedos da mão) e outra chamada (código que recebe do cliente quais dedos movimentar).

A máquina cliente empregada para captura e processamento é uma estação equipada com um processador Intel Core i5-12400 (12ª geração), 16 GB de memória RAM e uma placa de vídeo NVIDIA GeForce GTX 1660 Super. Nela foi instalado o ambiente python e a biblioteca MediaPipe para detectar e reconhecer gestos da mão, capturar o vídeo em tempo real e processar os frames para identificar os gestos. No AP, que é um sistema eletrônico com componentes Esp32, que foi instalado o firmware foi gravado na placa ESP32 o firmware MicroPython v1.26.0 — arquivo ESP32_GENERIC-20250809-v1.26.0.bin para python, na qual se conectam servomotores para movimentação da mão, isso através da implementação lógica para diferentes gestos.

Espera-se que este projeto resulte em um protótipo funcional de um braço robótico impresso em 3D, utilizando Esp32 e Python como base tecnológica. O desenvolvimento desse protótipo permitirá explorar as capacidades da impressão 3D e da programação em Python, bem como a integração de técnicas de inteligência artificial para a realização de tarefas específicas (abrir e fechar os dedos). Além disso, o projeto contribuirá para o aprendizado e a formação dos alunos envolvidos, proporcionando uma experiência prática e multidisciplinar no campo da robótica e automação.

3 RESULTADOS

O protótipo demonstrou funcionalidade na detecção e tradução de gestos para movimento robótico. O sistema cliente (PC) utiliza a biblioteca MediaPipe para reconhecer gestos dos cinco dedos e os traduz em comandos de estado, como "aberto" ou "fechado", que são enviados via Wi-Fi para o microcontrolador ESP32.

Para essa operação vamos dividir o processo em detecção da mão (cliente) e seus movimentos dos dedos, assim como o envio para o ESP32 na mão robótica (servidor)

No lado do cliente, primeiramente, devemos importar as bibliotecas necessárias para o reconhecimento da mão, conforme códigos da Figura 1.

Figura 1: Reconhecimento da mão com MediaPipe.

```

1  import socket
2  import time
3  import cv2
4  import mediapipe as mp
5  import logging
6  import os
7  import threading
8  from datetime import datetime
9
10 # Configuração da captura de vídeo
11 cap = cv2.VideoCapture(0, cv2.CAP_DSHOW if os.name == 'nt' else cv2.CAP_V4L2)
12 cap.set(3, 640) # Largura
13 cap.set(4, 480) # Altura
14
15 mpHands = mp.solutions.hands
16 hands = mpHands.Hands(static_image_mode=False, max_num_hands=1,
17                        min_detection_confidence=0.5, min_tracking_confidence=0.5)
18 mpDraw = mp.solutions.drawing_utils

```

Fonte: Elaborado pelos autores

As linhas 3 e 4 são as bibliotecas responsáveis pela captura e detecção da mão. A linha 11 faz a captura em vídeo e a linha 16 detecta a mão. A detecção da mão vai gerar pontos de identificação ao longo dos dedos e mão possibilitando controlar esses pontos.

Uma vez detectado a mão com seus pontos, iremos se os pontos na ponta dos dedos passaram da linha na base dos dedos, ver código na Figura 2.

Figura 2: Código que controla a posição dos dedos.

```

49  if results.multi_hand_landmarks:
50      for handLms in results.multi_hand_landmarks:
51
52          # Landmarks para cada dedo
53          finger_tips = [4, 8, 12, 16, 20] # Polegar, Indicador, Médio, Anelar, Mínimo
54          finger_mcp = [2, 5, 9, 13, 17] # MCP ou articulação base
55
56          # Detectar se cada dedo está aberto (tip < mcp => aberto)
57          for i, (tip_id, mcp_id) in enumerate(zip(finger_tips, finger_mcp)):
58              is_open = handLms.landmark[tip_id].y < handLms.landmark[mcp_id].y
59              state = "Aberto" if is_open else "Fechado"
60
61          # Só envia se houve mudança no estado do dedo
62          if previous_fingers_state[i] != state:
63              previous_fingers_state[i] = state
64              message = f"{finger_names[i]}: {state}"
65              threading.Thread(target=send_message_to_esp32, args=(message,)).start()
66
67          # Escreve na tela o estado de cada dedo
68          cv2.putText(img, f"{finger_names[i]}: {state}",
69                     (10, 30 + i * 30), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 255), 2)
70
71      mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS)

```

Fonte: Elaborada pelos autores

Nas linhas 53 e 54 definimos os pontos nos topos dos dedos (finger_tips) e os pontos da base

articular, esses valores são dados pela biblioteca MediaPipe. As linhas 57 e 65 são responsáveis em detectar o estado inicial dos dedos e uma possível mudança de estado. Caso ocorra a mudança de estado um thread¹ é enviado para o servidor. Nesse caso não espera retorno se o thread chegou ou não, isso acelera o processo de movimentação dos dedos. Em seguida escreve na tela o estado de cada dedo da mão, conforme Figura 3.

Figura 3: Detecção dos dedos – abertos ou fechados.



Fonte: Elaborada pelos autores

Bem, uma vez reconhecido e enviado o estado de cada dedo para o servidor, vamos analisar os códigos do servidor, que é MicroPython² e algumas bibliotecas já estão instaladas quando o ESP32 é configurado para rodar com python. Segue código na Figura 4.

Figura 4: Biblioteca MicroPython no ESP32.

```

1  import network
2  import socket
3  from machine import Pin, PWM
4
5  # Configuração dos servos para cada dedo
6  # Ajuste os pinos conforme seu hardware
7  servos = {
8      "Polegar": PWM(Pin(25, mode=Pin.OUT)),
9      "Indicador": PWM(Pin(26, mode=Pin.OUT)),
10     "Medio": PWM(Pin(27, mode=Pin.OUT)),
11     "Anelar": PWM(Pin(14, mode=Pin.OUT)),
12     "Minimo": PWM(Pin(12, mode=Pin.OUT))
13 }
14
15 for servo in servos.values():
16     servo.freq(50)

```

Fonte: Elaborado pelos autores

¹ Thread é um fluxo independente de controle que opera dentro do mesmo espaço de endereço que outros fluxos independentes de controles dentro de um processo)

² MicroPython é uma implementação enxuta e eficiente da linguagem de programação Python 3, otimizada para rodar em microcontroladores e sistemas embarcados com recursos limitados (pouca RAM/flash). <https://docs.micropython.org/en/latest/esp32/quickref.html>.

Nas linhas 1 e 2 são importados as bibliotecas responsáveis pela conexão do cliente e recebimento do thread e a linha 3 a biblioteca responsável pela comunicação do ESP32 com os servomotores. Nas linhas 7 à 16 são feito os ajustes dos pinos utilizados no ESP32.

Uma vez ajustados vamos a função que controla o movimento dos dedos através dos servomotores, conforme a Figura 5.

Figura 5: Função para mover os dedos.

```
18 # Funções para mover servos
19 def mover_dedo(dedo, estado):
20     if dedo not in servos:
21         print(f"Dedo desconhecido: {dedo}")
22         return
23
24     if estado == "Aberto":
25         servos[dedo].duty(26) # posição aberta
26     elif estado == "Fechado":
27         servos[dedo].duty(80) # posição fechada
28     else:
29         print(f"Estado desconhecido: {estado}")
```

Fonte: Elaborado pelos autores.

Nas linhas 19 à 29 a função mover_dedos recebe o tipo do dedo que mudou de estado (“Polegar”, “Indicador”, “Medio”, “Anelar” e “Minimo”), assim como o estado (“Aberto” ou “Fechado”). Caso aberto move o motor que está conectada a um cabo de nylon no dedo indicado no tipo.

O diagrama de detecção dos dedos, ver Figura 3, mostra o processo de como um "dicionário com os cinco dedos" recebe um *thread* para enviar o comando de "aberto ou fechado" via Wi-Fi para o servidor. O ESP32, atuando como servidor, recebe esses comandos e aciona os servomotores com precisão para replicar o movimento nos dedos da mão robótica.

4 CONSIDERAÇÕES FINAIS

O desenvolvimento do protótipo de mão robótica controlada por visão computacional demonstrou que é possível integrar, de forma eficiente e de baixo custo, tecnologias acessíveis como impressão 3D, MediaPipe, ESP32 e programação em Python. O sistema conseguiu reconhecer gestos da mão humana em tempo real e reproduzir esses movimentos por meio de servomotores conectados à mão robótica, estabelecendo uma comunicação estável via Wi-Fi entre o módulo cliente e o servidor embarcado.

A impressão 3D mostrou-se fundamental para viabilizar rapidamente peças customizadas, com boa precisão e repetibilidade, permitindo ajustes estruturais ao longo do projeto. Já o ESP32, executando MicroPython, ofereceu simplicidade na programação e controle preciso dos servomotores. O uso da biblioteca MediaPipe confirmou-se como uma solução robusta para detecção articulada dos dedos,

proporcionando respostas rápidas necessárias para interação dinâmica.

De modo geral, o sistema alcançou seus objetivos principais: demonstrou viabilidade técnica, funcionou de forma integrada e apresentou potencial para ampliações futuras em aplicações educacionais, industriais e de pesquisa. Além disso, contribuiu para a formação prática dos alunos envolvidos, ao unir conceitos de robótica, computação visual, manufatura digital e sistemas embarcados.

Como recomendação para etapas futuras, sugere-se a expansão do protótipo para incluir a movimentação articular do cotovelo do braço robótico, permitindo ao sistema executar gestos mais amplos e realistas. A inclusão de um segundo grau de liberdade complementar à movimentação da mão, tornando o braço mais funcional para ações como alcance, empurrar, levantar ou interagir com objetos posicionados em diferentes alturas.

REFERÊNCIAS

- AMARAL, H. N. et al. **Indústria 4.0 e robótica: automação industrial avançada**. *Interface Tecnológica*, v. 18, n. 1, 2021.
- ARAUJO, Sergio Eduardo Alonso et al. **Videocirurgia colorretal com assistência robótica: o próximo passo?** *Revista Brasileira de Coloproctologia*, São Paulo, v. 28, n. 3, p. 373–376, jul./set. 2008.
- BIGELLI, V. G. **Impressão 3D de polímeros: estudo de caso de um braço robótico de mesa**. 24 ago. 2021.
- MENDONÇA, M. et al. **DESENVOLVIMENTO DE UM BRAÇO ROBÓTICO CONTROLADO POR ARDUINO**. *Revista Sinergia*, v. 17, n. 3, p. 191–197, 30 set. 2016.
- MILES BARDON. **Humanoid Robotic Hand by Tohaker - Thingiverse**. Disponível em: <<https://www.thingiverse.com/thing:3105948/files>>. 2018. Acesso em: 26 ago. 2025.
- RECK, M. A. X.; ANTONELLO, R. Projeto para um Braço Robótico de Baixo Custo Baseado em Open Source e Impressão 3D. **Caderno de Propriedade Intelectual e Transferência de Tecnologia**, v. 3, n. 1, p. 65–72, 30 jun. 2021.
- SALES JÚNIOR, E. FERREIRA. **Sistema de controle inteligente para um braço robótico**. 30 dez. 1997.
- SHHEIBIA, T. A. A. EL. **Controle de um braço robótico utilizando uma abordagem de agente inteligente**. 6 jul. 2001.
- SILVA, C. C. da et al. **Inteligência artificial a favor da saúde: uma revisão sistemática sobre a aplicação da robótica em procedimentos cirúrgicos**. *Revista FT*, jan. 2025.
- SILVA, C. C. et al. **BRAÇO ROBÓTICO ARTICULADO CONTROLADO POR ARDUÍNO PARA DEMONSTRAÇÃO DO EMPREGO DA ROBÓTICA NAS INDÚSTRIAS. VI JORNACITEC - Jornada Científica e Tecnológica**, 6 out. 2017.
- SIQUEIRA-BATISTA, R. et al.. **ROBOTIC SURGERY: BIOETHICAL ASPECTS. ABCD. Arquivos Brasileiros de Cirurgia Digestiva (São Paulo)**, v. 29, n. 4, p. 287–290, out. 2016.