

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA  
CATARINA – CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA  
CURSO SUPERIOR DE ENGENHARIA MECATRÔNICA**

**LUIZ HENRIQUE SEIDLER DE MENESES**

**PRÓTESE MECÂNICA DE BAIXO CUSTO CONTROLADA PELO MÉTODO DE  
ELETROMIOGRAFIA E INTERNET DAS COISAS**

**FLORIANÓPOLIS, 2021**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA  
CATARINA – CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA  
CURSO SUPERIOR DE ENGENHARIA MECATRÔNICA**

**LUIZ HENRIQUE SEIDLER DE MENESES**

**PRÓTESE MECÂNICA DE BAIXO CUSTO CONTROLADA PELO MÉTODO DE  
ELETROMIOGRAFIA E INTERNET DAS COISAS**

Trabalho de conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para a obtenção do título de Engenheiro Mecatrônico.

Orientador: Prof. Dr. Eng. Francisco Rafael Moreira da Mota

**FLORIANÓPOLIS, 2021**

Ficha de identificação da obra elaborada pelo autor.

Meneses, Luiz Henrique Seidler  
**PRÓTESE MECÂNICA DE BAIXO CUSTO CONTROLADA PELO MÉTODO  
DE ELETROMIOGRAFIA E INTERNET DAS COISAS** / Luiz Henrique  
Seidler Meneses ; orientação de Francisco Rafael  
Moreira da Mota. - Florianópolis, SC, 2021.  
76 p.

**Trabalho de Conclusão de Curso (TCC)** - Instituto Federal  
de Santa Catarina, Câmpus Florianópolis. Bacharelado  
em Engenharia Mecatrônica. Departamento  
Acadêmico de Metal Mecânica.  
Inclui Referências.

1. Eletromiografia (EMG). 2. Próteses robóticas.  
3. Reconhecimento de padrões. 4. Classificação por aprendizado  
de máquina. I. Moreira da Mota, Francisco Rafael.  
II. Instituto Federal de Santa Catarina. Departamento  
Acadêmico de Metal Mecânica. III. Título.



**MINISTÉRIO DA EDUCAÇÃO**  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS

**ATA DA BANCA FINAL DO TRABALHO DE CONCLUSÃO DO CURSO DE ENGENHARIA  
MECATRÔNICA DO INSTITUTO FEDERAL DE SANTA CATARINA – Nº 0074**

Aos 07 de Abril de 2021, às 10h, o estudante **Luiz Henrique Seidler de Meneses** apresentou o seu Trabalho de Conclusão de Curso para julgamento à Banca Examinadora constituída pelos seguintes integrantes: Prof. Dr. Eng. Francisco Rafael Moreira da Mota (Orientador/Presidente da Banca/IF-SC), Prof. MSc. Eng. Cassiano Bonin (IF-SC), Prof. MSc. Eng. Adriano Regis (IF-SC). A sessão pública de defesa foi aberta pelo Presidente da Banca, que apresentou a Banca Examinadora e deu continuidade aos trabalhos, fazendo uma breve referência ao TCC que tem como título **PRÓTESE MECÂNICA DE BAIXO CUSTO CONTROLADA PELO MÉTODO DE ELETROMIOGRAFIA E INTERNET DAS COISAS**. Na sequência, o estudante teve até 30 minutos para a exposição de seu trabalho, e cada integrante da Banca Examinadora fez a arguição após a apresentação dele. Finalmente, foi aberto um espaço aos presentes para eventuais perguntas ou comentários sobre o trabalho apresentado. Ouvidas as explicações do estudante, a Banca Examinadora, reunida em caráter sigiloso, para proceder à avaliação final, deliberou pelo conceito **10**. Foi dada ciência ao estudante que a versão final do trabalho deverá ser entregue até o dia **07/05/2021**, com as devidas alterações sugeridas pela banca. Nada mais havendo a tratar, a sessão foi encerrada às 11 h 20 min, dela sendo lavrada a presente ata, que, uma vez aprovada, foi assinada por todos os membros da Banca Examinadora e pelo estudante.

---

Prof. Orientador / Presidente: Dr. Eng. Francisco Rafael Moreira da Mota

---

Prof. Avaliador 1: Prof. MSc. Eng. Cassiano Bonin

---

Prof. Avaliador 2: Prof. MSc. Eng. Adriano Regis

---

Acadêmico: Luiz Henrique Seidler de Meneses



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS

## DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o estudante **Luiz Henrique Seidler de Meneses** matrícula nº 151005040-0, do Curso de Engenharia Mecatrônica, defendeu o trabalho intitulado ***PRÓTESE MECÂNICA DE BAIXO CUSTO CONTROLADA PELO MÉTODO DE ELETROMIOGRAFIA E INTERNET DAS COISAS***, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Campus Florianópolis.

Florianópolis, 03 de maio de 2021.

Prof. Orientador do TCC:  
Prof. Dr. Eng. Francisco Rafael Moreira da Mota

## **AGRADECIMENTOS**

Aos meus pais, Gelson Henrique de Meneses e Zenaide Seidler de Meneses, à minha irmã, Ana Clara Seidler de Meneses, por todo apoio, incentivo e carinho durante toda a minha jornada.

Ao professor/orientador Francisco Rafael Moreira de Mota, por todo o apoio e suporte durante a graduação e todo o desenvolvimento deste projeto, sempre demonstrando total profissionalismo e comprometimento.

Aos meus amigos, pelo apoio, incentivo, amizade e aprendizagem durante toda a jornada de graduação, sendo fundamentais para meu crescimento como pessoa.

Ao Instituto Federal de Santa Catarina por toda a estrutura laboratorial de equipamentos, sempre oferecendo oportunidades de crescimento para os acadêmicos.

## RESUMO

A eletromiografia (EMG) é uma técnica capaz de fornecer informações concisas sobre o estado de um determinado músculo humano, que por sua praticidade e eficiência, é um dos métodos mais utilizados para o supervisionamento muscular em áreas médicas e esportivas. Com o grande avanço da tecnologia nas últimas décadas, a eletromiografia vem se tornando uma ferramenta muito útil em aplicações biomecânicas, possibilitando a conversão de movimentos humanos em movimentos de máquina. Uma das principais aplicações nesse contexto é o controle de próteses robóticas, proporcionando ao usuário a capacidade de movimentação a partir de estímulos musculares. Este presente trabalho tem como desígnio o desenvolvimento de uma prótese de baixo custo capaz de simular o movimento dos dedos da mão humana visando o uso de técnicas de eletromiografia para o seu controle. Além disso, o trabalho propõe a utilização de métodos de análise de dados e aprendizado de máquina para a criação de um classificador de movimentos baseado no sinal EMG, estando todo o sistema integrado com tecnologias de internet das coisas (IoT). Por fim, a integração de todas essas técnicas resultou em um classificador de 4 movimentos baseado em *Support Vector Machine* (SVM), com uma assertividade média de 85% e uma prótese de baixo custo capaz de reproduzir os movimentos a partir do estímulo muscular.

**Palavras-chave:** Eletromiografia (EMG). Próteses robóticas. Reconhecimento de padrões. Classificação por aprendizado de máquina.

## ABSTRACT

Electromyography (EMG) is a technique able to providing concise information about the state of a human muscle, due to its practicality and efficiency, is one of the most used methods for muscle supervision in medical and sports areas. With the advance of technology in the last decades, electromyography has become a very useful tool in biomechanical applications, allowing the conversion of human movements into machine movements. One of the main applications in this context is the control of robotic prostheses. This present work aims to develop a low cost prosthesis capable of simulating the movement of the fingers of the human hand using electromyography techniques for its control. In addition, the work proposes the use of data analysis methods and machine learning for the creation of a movement classifier based on the EMG signal, the entire system being integrated with internet of things (IoT). Finally, the integration of all these techniques resulted in a 4-movement classifier based on Support Vector Machine (SVM), with an average accuracy of 85% and a low-cost prosthesis capable of reproducing the movements from a muscle stimulus.

**Keywords:** Electromyography (EMG). Robotic prostheses. Pattern recognition. Machine Learning classification.



## LISTA DE FIGURAS

Figura 1: Representação esquemática da captação e tratamento de sinais EMG ....	20
Figura 2: Exemplo de decomposição do sinal EMG por filtros passa banda .....	20
Figura 3: Posição ideal dos eletrodos em relação à unidade motora .....	21
Figura 4: Princípio do processo FDM .....	26
Figura 5: Empenamento no processo de impressão 3D .....	27
Figura 6: Projeto de Brian Brocken .....	33
Figura 7: Impacto do tensionamento dos cordões sobre os dedos da prótese .....	35
Figura 8: ESP32S.....	38
Figura 9: WeMos D1 R2.....	39
Figura 10: Sensor muscular V3 e posicionamento dos eletrodos.....	40
Figura 11: Circuito de captação do sinal EMG .....	42
Figura 12: Circuito de controle dos motores.....	43
Figura 13: Sinal com 3 estímulos utilizado para o treinamento do algoritmo .....	46
Figura 14: Modelo de arquivo para treinar o algoritmo.....	48
Figura 15: Fluxo de funcionamento do sistema.....	51
Figura 16: Estrutura final da prótese .....	52
Figura 17: Comparação da resolução entre ESP32 e Arduino Mega.....	54
Figura 18: Interface web de comunicação.....	55
Figura 19: WL na extração de características para 7 movimentos.....	56
Figura 20: Demais características para 7 movimentos.....	57
Figura 21: Resultado da combinação de características.....	58
Figura 22: Matriz de confusão para classificação de 7 movimentos .....	59
Figura 23: Matriz de confusão para classificação de 4 movimentos .....	61

## LISTA DE TABELAS

Tabela 1: Precisão do <i>machine learning</i> encontrado no experimento de Too, Abdullah e Saad (2019).....	31
Tabela 2: Precisão do <i>machine learning</i> utilizando múltiplas características no experimento de Too, Abdullah e Saad (2019) .....	32
Tabela 3: Ângulo de fechamento e abertura para cada servo motor.....	51

## LISTA DE ABREVIATURAS

EMG – Eletromiografia.

IoT – *Internet of Things* (Internet das Coisas).

SVM – *Support Vector Machine* (Máquina de Vetores Suporte).

WL – *Wavelength* (Comprimento de Onda).

MAV – *Mean Absolute Value* (Desvio Médio Absoluto).

UM – Unidade Motora.

ZC – *Zero Crossing* (Cruzamento de Zero).

SSC – *Slope Sign Change* (Mudança de Sinal pela Inclinação).

AAC – *Average Amplitude Change* (Mudança de Amplitude Média).

LD – *Log Detector* (Detector Logarítmico).

RMS – *Root Mean Square* (Raiz Quadrada Média).

DASDV – *Difference Absolute Standard Deviation Value* (Valor da Diferença Absoluta do Desvio Padrão).

SSI - *Simple Square Integral* (Integral Quadrada Simples).

VAR – *Variance of EMG Signal* (Variância do Sinal EMG).

MMAV – *Modified Mean Absolute Value* (Desvio Médio Absoluto Modificado).

MMAV2 – *Modified Mean Absolute Value 2* (Desvio Médio Absoluto Modificado 2).

EMAV – *Enhanced Mean Absolute Value* (Desvio Médio Absoluto Melhorado).

EWL – *Enhanced Wavelength* (Comprimento de Onda Melhorado).

CAD – *Computer Aided Design* (Desenho Assistido por Computador).

FDM – *Fused Deposition Modeling* (Modelagem por Fusão e Deposição).

STL – *Standard Tessellation Language* (Linguagem Padrão de Tecelagem).

HTTP – *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertextos).

TCP – *Transmission Control Protocol* (Protocolo de Controle de Transmissão).

UDP – *User Datagram Protocol* (Protocolo de Datagramas de Usuário).

KNN – *K-Nearest Neighbor* (K-ésimo Vizinho mais Próximo).

NB – *Naive Bayes*.

LDA – *Linear Discriminant Analysis* (Análise Discriminante Linear).

FFF – *Fused Filament Fabrication* (Fabricação por Filamento Fundido).

PLA – Ácido Polilático.

AMPOP – Amplificador Operacional.

IP – *Internet Protocol* (Protocolo de Internet).

URL – *Uniform Resource Locator* (Localizador Uniforme de Recursos).

IHM – Interface Humano Máquina.

HTML – *Hypertext Markup Language* (Linguagem de Marcação Hipertexto).

RBF – *Radial Base Function* (Função de Base Radial).

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>15</b>
<b>1.1. Objetivos</b> .....	<b>16</b>
1.1.1. Objetivo Geral .....	16
1.1.2. Objetivos Específicos .....	16
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
<b>2.1. Eletromiografia de Superfície</b> .....	<b>19</b>
<b>2.2. Extração de Características de Sinais EMG</b> .....	<b>22</b>
<b>2.3. Manufatura Aditiva</b> .....	<b>25</b>
2.3.1. Modelagem por Fusão e Deposição (FDM) .....	26
2.3.2. Variáveis do Processo de Impressão 3D .....	27
<b>2.4. Internet das Coisas</b> .....	<b>28</b>
<b>2.5. Machine Learning</b> .....	<b>29</b>
<b>3. MATERIAIS E MÉTODOS</b> .....	<b>33</b>
<b>3.1. Prótese Mecânica</b> .....	<b>33</b>
3.1.1. Modelo Tridimensional .....	33
3.1.2. Fabricação por Manufatura Aditiva .....	34
3.1.3. Sistema de Movimentação .....	35
3.1.4. Montagem .....	36
3.1.5. Teste Mecânico .....	36
<b>3.2. Módulo Eletrônico</b> .....	<b>37</b>
3.2.1. ESP32 .....	37
3.2.2. WeMos D1 R2 .....	38
3.2.3. Sistema de Aquisição do Sinal EMG .....	39
3.2.4. Sistema de Acionamento dos Motores .....	42
<b>3.3. Internet das Coisas</b> .....	<b>43</b>
3.3.1. Servidor ESP32 .....	43
3.3.2. Cliente ESP8266 .....	44
3.3.3. Interface Web .....	44
<b>3.4. Análise dos Dados EMG</b> .....	<b>45</b>
3.4.1. Extração de Características Offline .....	45
3.4.2. Extração de Características Online .....	46

3.4.3. Machine Learning.....	47
<b>4. RESULTADOS E DISCUSSÃO .....</b>	<b>51</b>
<b>4.1. Prótese de Baixo Custo Fabricada por Manufatura Aditiva .....</b>	<b>51</b>
<b>4.2. Módulo Eletrônico .....</b>	<b>53</b>
<b>4.3. Internet das Coisas .....</b>	<b>54</b>
<b>4.4. Tratamento do Sinal EMG.....</b>	<b>55</b>
4.4.1. Extração de Características Offline .....	55
4.4.2. Extração de Características Online .....	58
<b>4.5. Classificação do Sinal EMG .....</b>	<b>59</b>
4.5.1. Modelo com 7 Movimentos .....	59
4.5.2. Modelo com 4 Movimentos .....	60
<b>5. CONCLUSÃO E TRABALHO FUTURO .....</b>	<b>62</b>
<b>REFERÊNCIAS .....</b>	<b>64</b>
Anexo A - Circuito Eletrônico do Muscle Sensor V3 .....	66
Apêndice A - Código do servidor para ESP32 .....	67



## 1. INTRODUÇÃO

Eletromiografia (EMG) é o nome dado ao conjunto de técnicas utilizadas para captar a atividade elétrica associada a contrações musculares (KAMEM; GABRIEL, 2015). Tudo começou no fim do século dezoito, quando Luigi Galvani (1737, Bolonha - 1798, Bolonha) observou a relação entre eletricidade e movimentação muscular ao despolarizar pernas de rãs com hastes de metal, comprovando a existência de potenciais neuromusculares. Em 1849, o francês Du Bois-Reymond (1818, Berlin - 1896, Berlin) forneceu a primeira evidência da geração de corrente elétrica em músculos humanos a partir de uma contração voluntária, utilizando um galvanômetro. No entanto, os sinais dos músculos só puderam ser visualizados a partir de 1920 com a criação do osciloscópio de raios catódicos. Nas décadas de 1930, 1940 e 1950, o desenvolvimento da instrumentação para eletromiografia de superfície propiciou estudos variados envolvendo a musculatura humana. A partir de então, a eletromiografia começou a utilizar dispositivos para detecção, amplificação e apresentação do sinal de EMG em forma gráfica.

Por ser capaz de fornecer informações muito ricas sobre o comportamento e condição dos músculos, o sinal eletromiográfico vem sendo cada vez mais utilizado em aplicações médicas e biomecânicas. O avanço na tecnologia das últimas décadas possibilitou o aumento nos estudos sobre o sinal EMG, já que é necessário uma série de filtros e ajustes para possibilitar o seu uso em sistemas de leitura e controle. A obtenção desses sinais está dividida em dois grandes grupos, a eletromiografia de superfície e a eletromiografia intramuscular, sendo a primeira a mais popular nas aplicações atuais devido a sua alta praticidade. Hoje em dia já é possível captar sinais musculares de forma confiável, transformando-os em sinais digitais que podem ser amplificados e lidos por diferentes controladores, tornando o sinal EMG uma das fontes mais poderosas no controle de próteses em geral (KAMEM; GABRIEL, 2015).

Prótese é um dispositivo implantado no corpo com a função de suprir a falta de um órgão ausente ou até mesmo restaurar uma funcionalidade humana comprometida. A mais antiga prótese conhecida não se trata de uma perna ou braço, mas sim de um dedo do pé encontrado no antigo Egito. Acredita-se que este peculiar objeto, criado entre os anos 950 - 710 A.C. tinha a função de suportar o uso



das tradicionais sandálias egípcias da época. As próteses continuaram sendo comuns por diversas eras da evolução humana, como por exemplo as famosas pernas de pau utilizadas por piratas e as mãos de ferro usadas pelos cavaleiros da idade média. No entanto, a primeira prótese articulada da mão humana foi criada somente no século dezesseis, pelo cirurgião militar Ambroise Paré (1510, Laval – 1590, Paris) (CHAPPEL, 2016).

Para o controle de próteses, muitas vezes é utilizado algoritmos para realizar a classificação de respostas provenientes de determinados sensores. Esses algoritmos são construídos a partir de técnicas de aprendizado de máquina que utilizam sinais reais como parâmetro de classificação. A assertividade dos classificadores depende diretamente do tratamento prévio do sinal, sendo muitas vezes necessário, submetê-lo a filtros e transformações matemáticas. No caso do sinal EMG, existem diversos métodos documentados sobre a sua extração de características, que facilitam e melhoram o reconhecimento de padrões durante a leitura do sensor.

A extração de características do sinal EMG vem sendo estudada ao longo da história, Too, Abdullah e Saad (2019) descreve a utilização de 14 delas e propõe a criação de 2 novas funções baseadas em *Wavelength* (WL) e *Mean Absolute Value* (MAV). Ele afirma que essas duas características são as mais utilizadas para análise de sinais EMG devido à sua simplicidade e eficiência.

## **1.1. OBJETIVOS**

### **1.1.1. OBJETIVO GERAL**

Desenvolver uma prótese robótica de baixo custo capaz de simular movimentos da mão humana utilizando eletromiografia de superfície.

### **1.1.2. OBJETIVOS ESPECÍFICOS**

- a) Fabricar uma prótese a partir do processo de manufatura aditiva;

- b) Utilizar técnicas de Internet das Coisas (IoT) para gerenciar e supervisionar os dados resultantes da leitura do sinal EMG;
- c) Estudar o comportamento do sinal EMG e utilizar técnicas de análise de dados para melhorar a sua leitura por microcontroladores;
- d) Desenvolver um sistema de classificação de movimentos utilizando sinais EMG e aprendizado de máquina;



## 2. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta uma abordagem literária envolvendo os principais fundamentos relacionados à eletromiografia, próteses, manufatura aditiva, internet das coisas, tratamento de dados e aprendizado de máquina.

### 2.1. ELETROMIOGRAFIA DE SUPERFÍCIE

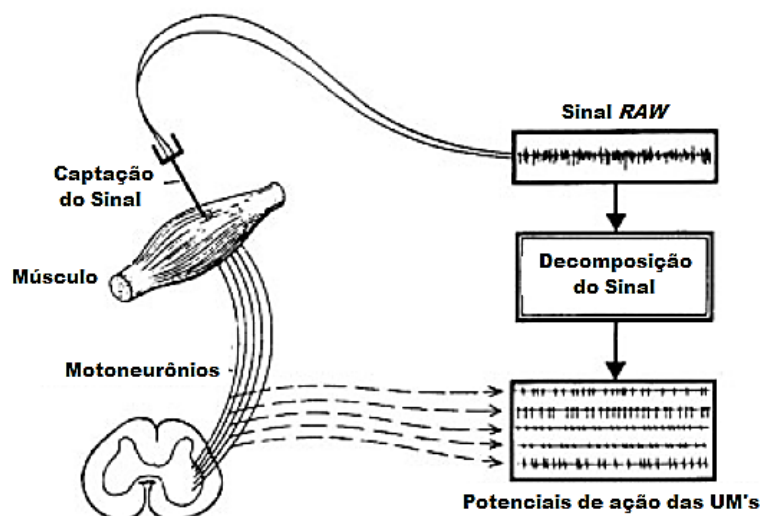
Eletromiografia de superfície é uma técnica não invasiva e indolor (diferente da eletromiografia intramuscular), utilizada para o monitoramento das atividades elétricas das fibras musculares durante o processo de contração e relaxamento (MIOTEC, 2019).

A captação do sinal eletromiográfico por superfície é realizada por meio de sensores posicionados sobre a pele (eletrodos), exatamente acima do músculo a ser analisado. Assim, é possível captar a diferença de potencial que é gerado durante a movimentação muscular.

Segundo Merletti e Parker (2004), durante a captação de sinais através das técnicas de eletromiografia, é possível haver a detecção de diferentes potenciais de ação das Unidades Motoras (UM), resultando em uma sobreposição de sinais que afetam a leitura de saída (*Sinal Raw*).

Com isso, faz-se necessário o uso de artifícios de tratamento de sinais para decompor o *Sinal Raw* em sinais de mais fácil análise e manipulação. Na Figura 1 pode-se analisar o processo de captação e decomposição dos sinais EMG.

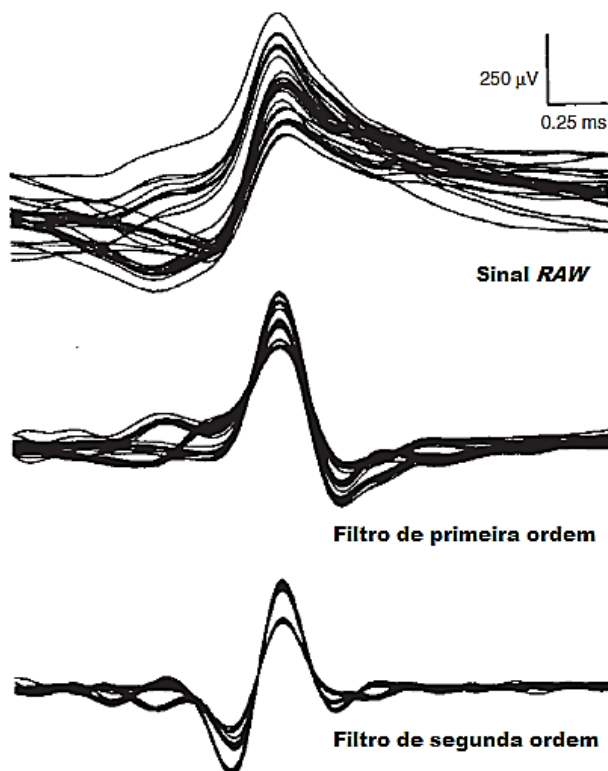
Figura 1: Representação esquemática da captação e tratamento de sinais EMG



Fonte: Adaptado de (MERLETTI, PARKER, 2004)

Existem diversos métodos para aprimorar a leitura do sinal EMG, amplificação, retificação, suavização e filtragem são alguns dos exemplos possíveis. Na Figura 2 é possível notar a transformação do sinal após o uso de filtros passa banda de primeira e segunda ordem.

Figura 2: Exemplo de decomposição do sinal EMG por filtros passa-faixa



Fonte: Adaptado de (MERLETTI, PARKER, 2004)

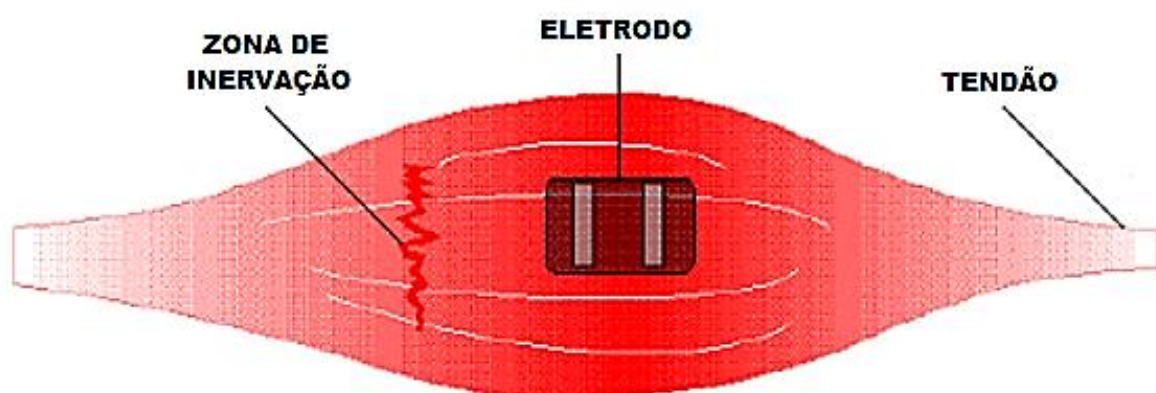
Eletrodos de superfície fornecem uma técnica não invasiva para medição e detecção de sinais EMG. O equilíbrio químico formado entre o eletrodo e a pele, por condução eletrolítica, possibilita a passagem da corrente elétrica resultante do estímulo muscular (JAMAL, 2012).

Diferente dos eletrodos intramusculares, os eletrodos de superfície são de fácil manipulação, não sendo necessário um acompanhamento médico. No entanto, esses eletrodos só podem ser utilizados em músculos superficiais e seu posicionamento deve ser estável em relação à pele, a fim de evitar a distorção do sinal (JAMAL, 2012).

A aplicação de eletrodos de superfície para leitura de sinais EMG necessita de uma preparação prévia da pele a fim de garantir uma melhor qualidade na aquisição dos dados. Para reduzir a impedância, as células mortas e os pelos da pele devem ser totalmente removidos dos locais que serão inseridos os eletrodos. É aconselhável utilizar gel abrasivo e álcool para realização da limpeza, garantindo também a eliminação de qualquer umidade ou suor na pele (JAMAL, 2012).

O posicionamento e orientação dos eletrodos são fundamentais para a garantia de uma boa aquisição de sinais. Os eletrodos devem ser colocados entre a unidade motora e o tendão da extremidade do músculo conforme a Figura 3. A distância entre os centros dos eletrodos deve variar entre 1 e 2 centímetros e o eixo longitudinal formado pelos eletrodos deve ser paralelo ao comprimento das fibras musculares (JAMAL, 2012).

Figura 3: Posição ideal dos eletrodos em relação à unidade motora



Fonte: Adaptado de (JAMAL, 2012)

## 2.2. EXTRAÇÃO DE CARACTERÍSTICAS DE SINAIS EMG

A extração de características é um processo de transformação do sinal bruto em uma estrutura de dados relevantes, removendo ruídos e destacando atributos importantes para análise (SPIEWAK *et al.*, 2018). Esse recurso deve incluir informações significativas que podem melhor descrever os sinais durante o processo de classificação (TOO, ABDULLAH, SAAD, 2019). Neste estudo foram utilizados alguns métodos muito populares no tratamento de sinais EMG. A saber:

- I. *Mean Basolute Value* (MAV) - é um dos métodos mais populares utilizados em aplicações relacionadas à eletromiografia e pode ser definido como a média do somatório dos valores absolutos do sinal EMG. MAV pode ser expresso por:

$$MAV = \frac{1}{L} \sum_{i=1}^L |x_i| \quad (1)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- II. *Wavelength* (WL) - é outro método muito popular no tratamento de sinais EMG e pode ser calculado através do somatório do valor cumulativo do sinal. WL pode ser expresso por:

$$WL = \sum_{i=2}^L |x_{i+1} - x_i| \quad (2)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- III. *Zero Crossing* (ZC) - é um método que mede a frequência de valores ao longo do sinal. Zero crossing pode ser expresso por:

$$ZC = \sum_{i=1}^{L-1} f(x_i) \quad (3)$$

$$f(x_i) = \begin{cases} 1, & \text{se } (x_i > 0 \text{ e } x_{i+1} < 0) \text{ ou } (x_i < 0 \text{ e } x_{i+1} > 0) \\ 0, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- IV. *Slope Sign Change* (SSC) - é um método muito comum na análise de sinais EMG. Ele determina o número de vezes em que ocorre uma alteração relevante do sinal e pode ser calculado por:

$$SSC = \sum_{i=2}^{L-1} f(x_i) \quad (4)$$

$$f(x_i) = \begin{cases} 1, & \text{se } (x_i > x_{i-1} \text{ e } x_i > x_{i+1}) \text{ ou } (x_i < x_{i-1} \text{ e } x_i < x_{i+1}) \\ 0, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- V. *Average Amplitude Change* (AAC) - é outro método muito utilizado em sinais EMG e é capaz de calcular a variação na amplitude do sinal. Ele pode ser expresso por:

$$AAC = \frac{1}{L} \sum_{i=1}^{L-1} |x_{i+1} - x_i| \quad (5)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

- VI. *Log Detector* (LD) - é um método utilizado para analisar a força do sinal e pode ser expresso por:

$$LD = \exp\left(\frac{1}{L} \sum_{i=1}^L \log(|x_i|)\right) \quad (6)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

- VII. *Root Mean Square* (RMS) - é um método muito popular, sendo útil para descrever informações específicas do movimento muscular. Pode ser expresso por:

$$RMS = \sqrt{\frac{1}{L} \sum_{i=1}^L (x_i)^2} \quad (7)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

- VIII. *Difference Absolute Standard Deviation Value* (DASDV) - é responsável por calcular o desvio padrão absoluto do sinal e pode ser calculado por:



$$DASDV = \sqrt{\frac{\sum_{i=1}^{L-1} (x_{i+1} - x_i)^2}{L-1}} \quad (8)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

- IX. *Simple Square Integral* (SSI) - é definido pelo somatório dos quadrados de cada valor do sinal sendo expresso por:

$$SSI = \sum_{i=1}^L (x_i)^2 \quad (9)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

- X. *Variance of EMG Signal* (VAR) - é um bom método para medir a potência do sinal e pode ser expresso por:

$$VAR = \frac{1}{L-1} \sum_{i=1}^L (x_i)^2 \quad (10)$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- XI. *Modified Mean Absolute Value* (MMAV) - é uma extensão do método MAV, modificando o peso dos dados em uma determinado intervalo. Matematicamente, pode ser expresso por:

$$MMAV = \frac{1}{L} \sum_{i=1}^L w_i |x_i| \quad (11)$$

$$w_i = \begin{cases} 1, & \text{se } 0.25L \leq i \leq 0.75L \\ 0.5, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- XII. *Modified Mean Absolute Value 2* (MMAV2) - é outra variação do MAV que atribui um peso contínuo em um intervalo do sinal. Matematicamente, pode ser expresso por:

$$MMAV2 = \frac{1}{L} \sum_{i=1}^L w_i |x_i| \quad (12)$$

$$w_i = \begin{cases} 1, & \text{se } 0.25L \leq i \leq 0.75L \\ \frac{4i}{L}, & \text{se } i < 0.25L \\ \frac{4(i-L)}{L}, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

Segundo Too, Abdullah e Saad (2019), a maioria das informações relevantes são encontradas na parte central do sinal EMG, tornando o início e o fim do mesmo, regiões menos importantes. Para contornar esse problema, foram propostos dois novos métodos que derivam dos clássicos: o *Enhanced Mean Absolute Value* (EMAV) e o *Enhanced Wavelength* (EWL). A saber:

- I. *Enhanced Mean Absolute Value* (EMAV) – é um método que utiliza a mesma lógica aplicada em MAV, porém com foco na região central do sinal EMG. O EMAV pode ser expresso por:

$$EMAV = \frac{1}{L} \sum_{i=1}^L |(x_i)^p| \quad (13)$$

$$p = \begin{cases} 0.75, & \text{se } i \geq 0.2L \text{ e } i \leq 0.8L \\ 0.50, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal.

- II. *Enhanced Wavelength* (EWL) - utiliza a lógica aplicada em WL, também com foco na região central do sinal EMG. O EWL pode ser expresso por:

$$EWL = \sum_{i=2}^L |(x_i - x_{i-1})^p| \quad (14)$$

$$p = \begin{cases} 0.75, & \text{se } i \geq 0.2L \text{ e } i \leq 0.8L \\ 0.50, & \text{caso contrário} \end{cases}$$

Em que  $x$  é o valor do sinal EMG e  $L$  o comprimento do sinal

### 2.3. MANUFATURA ADITIVA

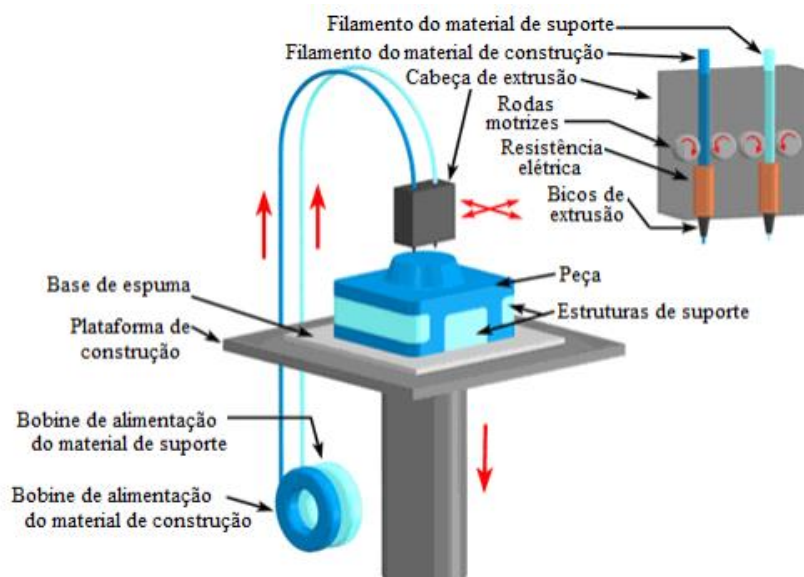
A manufatura aditiva ou impressão 3D pode ser definida como um processo de fabricação através da adição de material por camadas planas sucessivas, assim sendo capaz de produzir geometrias tridimensionais. O processo tem início com o

“fatiamento” do modelo CAD 3D, obtendo-se as “curvas de nível” 2D que indicam onde deve ou não ser depositado o material. A peça final é resultado de um empilhamento dessas camadas que se solidificam durante o processo (VOLPATO, 2017).

### 2.3.1 - MODELAGEM POR FUSÃO E DEPOSIÇÃO (FDM)

Conhecida por ser a mais comum e acessível das tecnologias de impressão 3D, esse processo se baseia no aquecimento de uma resina termoplástica, sob a forma de fio, através de um cabeçote de extrusão que se movimenta no plano x-y. O extrusor traciona o filamento através de componentes rolantes, empurrando-o contra o bico de extrusão devidamente aquecido, formando um fluxo contínuo de polímero fundido. O material é depositado sobre uma plataforma com temperatura inferior ao de extrusão a fim de que ocorra a rápida solidificação do termoplástico. Em seguida, a máquina se movimenta no eixo z para que possa dar continuidade às demais camadas para a fabricação da peça (ABREU, 2015). A Figura 4 mostra os principais componentes do processo FDM.

Figura 4: Princípio do processo FDM



Fonte: Abreu(2015)

Para que o arquivo CAD possa ser processado pela impressora 3D, é necessário o uso de um *software* com a função de fatiar o modelo e fornecer as

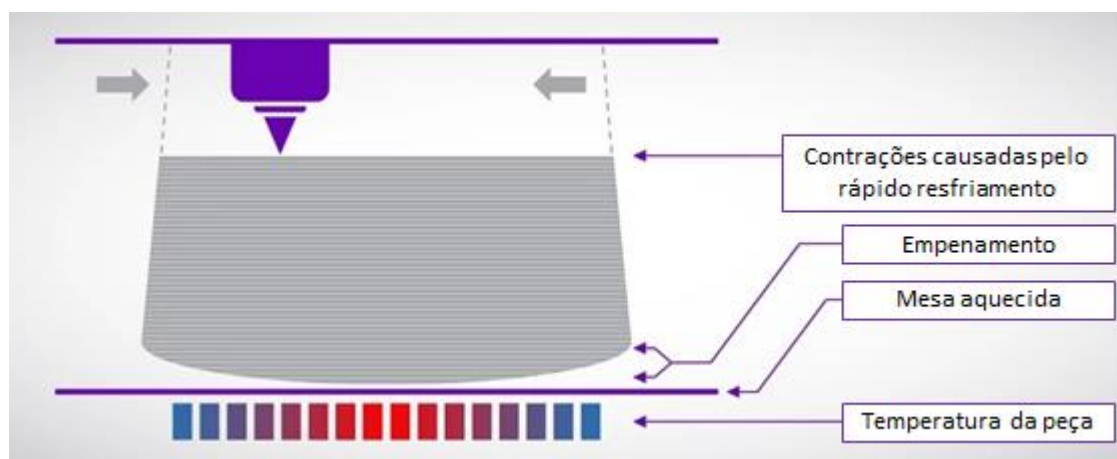
coordenadas de movimentação para o cabeçote de extrusão. Atualmente muitos desses *softwares* são de código aberto, o que justifica o enorme uso dessa tecnologia pelo mundo.

Comumente, os *softwares* de impressão 3D utilizam arquivos no formato STL (Standard Tessellation Language), que se baseia no processo de aproximação por faces planas. Este formato aproxima a superfície do modelo sólido por pequenos triângulos, construindo uma malha que representa a geometria do modelo CAD. Quanto menores os triângulos, mais fiel é a construção da superfície, porém, maior é o tamanho do arquivo, e conseqüentemente maior é o tempo de processamento pelo ficheiro da máquina (ABREU, 2015).

### 2.3.2 - VARIÁVEIS DO PROCESSO DE IMPRESSÃO 3D

Deve-se levar em conta alguns fatores que são de suma importância para a qualidade e acabamento da peça a ser fabricada. Uma delas, é garantir uma boa aderência da resina plástica sobre a mesa de impressão 3D aquecida. Uma má preparação da superfície pode ocasionar empenamentos na peça (*warping*) devido ao desprendimento das extremidades durante o processo de impressão. Existem diversas formas de assegurar uma boa fixação, como a utilização de fitas porosas, *sprays* fixadores, colas e adesivos. Na Figura 5 pode-se observar o resultado de uma má fixação à mesa, que devido ao rápido resfriamento das extremidades da peça, resulta em deformações.

Figura 5: Empenamento no processo de impressão 3D



Fonte: Adaptado de (RIGID INK, 2016).

Outro fator de grande importância ao sucesso de impressão, é a criação de suportes de sustentação. Esses, são utilizados para dar apoio a geometrias que se apresentam suspensas em relação à mesa de deposição. Elas são criadas em colunas finas e perpendiculares ao plano de impressão.

## 2.4. INTERNET DAS COISAS

A internet das coisas (do inglês *Internet Of Things* (IoT)) refere-se ao uso de dispositivos e sistemas conectados de forma inteligente para coletar, analisar e distribuir informações úteis em um determinado processo. Espera-se que o uso de sistemas IoT se espalhe rapidamente nos próximos anos, o que desencadeará em uma nova perspectiva que irá melhorar a qualidade de vida dos consumidores e aumentar a produtividade das empresas (GSMA, 2014).

Segundo Evans (2011), a internet, desde sua criação, teve grandes impactos na construção da sociedade moderna, alterando a forma clássica de educação, comunicação, negócios, ciência, governo e humanidade. Considerando o IoT a próxima evolução da internet, suas funcionalidades serão capazes de tornar dados simplórios em informações, conhecimento e finalmente, sabedoria.

Para que seja possível uma comunicação entre computadores, é necessário uma série de regras e padrões que definem a forma de transmissão de informações, esses são conhecidos como protocolos de rede. Um dos protocolos mais conhecidos e utilizados é o HTTP, que é a base para qualquer troca de dados na web a partir do método cliente-servidor (MDN, 2021).

Clientes e servidores se comunicam através de mensagens individuais. Geralmente, as mensagens enviadas pelo cliente são chamadas de solicitações (*requests*), que tem como objetivo, solicitar alguma informação presente no servidor. Já as mensagens enviadas pelo servidor, são chamadas de respostas (*responses*), que carregam o dado requerido pelo cliente (MDN, 2021).

É importante ressaltar que o HTTP utiliza o protocolo de transporte TCP, que é voltado para conexões web, garantindo a integridade e organização de todos os dados, diferentemente do protocolo UDP (MDN, 2021).

O HTTP possui um conjunto de métodos de requisição que são responsáveis por indicar a ação que deve ser executada durante a comunicação entre o cliente e

o servidor, entre os métodos mais utilizados estão o GET e POST. O primeiro solicita a representação de um determinado recurso especificado, ou seja, pede ao servidor que retorne uma informação específica. Já o método POST, é utilizado para submeter uma entidade a um recurso específico, ou seja, envia dados formatados para o ambiente servidor (MDN, 2021).

## 2.5. MACHINE LEARNING

*Machine Learning* pode ser amplamente definido como métodos computacionais que utilizam a experiência para melhorar o desempenho ou fazer previsões precisas. A experiência se refere às informações disponíveis anteriormente para a criação do algoritmo, que normalmente assume a forma de dados eletrônicos coletados e disponibilizados para análise. Esses dados são obtidos a partir de interações com o ambiente real, formando um conjunto de dados de treinamento capaz de descrever o sistema. Em todos os casos, sua qualidade e tamanho são cruciais para o sucesso das previsões (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

Too, Abdullah e Saad (2019) descrevem quatro métodos distintos de aprendizagem de máquina, sendo eles os mais utilizados para a classificação de sinais EMG. A saber:

- I. *K-Nearest Neighbor* (KNN) - é um dos métodos de classificação mais fundamentais e simples, e deve ser uma das primeiras escolhas para um estudo onde há pouco ou nenhum conhecimento prévio sobre a distribuição dos dados (PETERSON, 2009). Esse método se baseia no cálculo das distâncias euclidianas entre uma amostra de interesse e outras amostras de treinamento previamente especificadas. A classificação ocorre a partir do fracionamento do dado de treinamento, dividindo-o em categorias que serão usadas para classificar o dado de entrada (PETERSON, 2009).
- II. *Naive Bayes* (NB) - Baseado no teorema de Bayes, o método NB é capaz de realizar classificações a partir da probabilidade de eventos, tratando

cada atributo de forma independente, ou seja, sem os relacionar. Esse algoritmo é conhecido por sua matemática simples e a não necessidade de um extenso dado de treinamento, facilitando a implementação e validação do sistema (TOO; ABDULLAH; SAAD, 2019).

- III. *Linear Discriminant Analysis* (LDA) - LDA é o método de aprendizagem de maior robustez em estudos de sinais EMG, oferecendo uma maior velocidade de classificação e menor esforço computacional. A ideia principal desse algoritmo é discriminar o sinal de interesse e compará-lo com os dados de treinamento, classificando-o de acordo com sua semelhança aos grupos de classificação (TOO; ABDULLAH; SAAD, 2019).
- IV. *Support Vector Machine* (SVM) - SVM é um algoritmo de aprendizagem muito conhecido e poderoso para reconhecimento de padrões EMG. A ideia geral desse método é encontrar um hiperplano que corresponde à maior margem possível entre os dados de diferentes classes (TOO; ABDULLAH; SAAD, 2019). Uma das vantagens do *Support Vector Machine* é a possibilidade do uso de um *kernel*, que nada mais é do que um conjunto de funções matemáticas capaz de aumentar a velocidade de processamento e a assertividade do sistema (SPIEWAK *et al.*, 2018).

No estudo realizado por Too, Abdullah e Saad (2019) foram adquiridos diversos conjuntos de sinais EMG a partir da prática de eletromiografia por superfície, com o objetivo de classificá-los em 6 tipos. Em seguida, ele criou 4 diferentes algoritmos de aprendizagem e os aplicou diretamente sobre 16 características extraídas dos sinais. Na Tabela 1 é apresentado a assertividade de cada método, onde o melhor resultado foi encontrado a partir do uso do LDA e EMVA. No entanto, devido a difícil implementação do método LDA para leituras em tempo real, o SVM foi o algoritmo escolhido para classificar o sinal EMG (3.4.3).

Tabela 1: Precisão do *machine learning* encontrado no experimento de Too, Abdullah e Saad (2019)

MÉTODO	PRECISÃO (%)			
	KNN	LDA	SVM	NB
MAV	94,56	96,33	95,11	93,22
WL	94,56	94,56	95,89	93,78
ZC	94,22	95,11	95,00	92,56
SSC	83,89	87,22	89,00	83,67
AAC	92,56	94,56	94,44	93,78
LD	90,22	91,78	92,78	91,56
RMS	94,00	95,67	95,11	93,78
DASDV	93,56	95,89	95,11	93,22
SSI	93,00	90,00	94,78	92,44
VAR	90,56	90,00	92,44	92,44
MMAV	94,33	95,44	95,44	93,00
MMAV2	91,44	93,56	93,56	92,11
EWL	<b>95,00</b>	95,11	<b>96,22</b>	93,56
EMAV	94,67	<b>96,89</b>	96,11	<b>93,89</b>

Fonte: Adaptado de (TOO; ABDULLAH; SAAD, 2019)

Em seguida, realizou um segundo teste utilizando a combinação entre características para alimentar os algoritmos de aprendizagem. Na Tabela 2 são apresentados os resultados para essas configurações, onde o melhor resultado foi encontrado a partir do uso do LDA e da combinação entre EMAV, EWL, MAV e WL.



Tabela 2: Precisão do *machine learning* utilizando múltiplas características no experimento de Too, Abdullah e Saad (2019)

MÉTODO	PRECISÃO (%)			
	KNN	LDA	SVM	NB
EMAV+EWL	95,00	96,89	96,22	94,56
EMAV+EWL+MAV+WL	94,44	<b>97,56</b>	96,22	94,22
EMAV+EWL+SSC+ZC	<b>96,00</b>	95,22	<b>97,22</b>	<b>96,56</b>
EMAV+EWL+RMS+DASDV	95,00	96,67	96,22	94,78
WL+MAV+SSC+ZC	95,33	96,11	96,78	96,44
MAV+MMAV+MMAV2	94,22	94,56	95,67	93,67
AAC+LD+RMS+DASDV	94,22	96,11	95,22	94,00
WL+MAV	94,56	96,33	95,89	94,11
SSI+VAR+MMAV+MMAV2	93,00	95,44	94,67	93,56

Fonte: Adaptado de (TOO; ABDULLAH; SAAD, 2019)

### 3. MATERIAIS E MÉTODOS

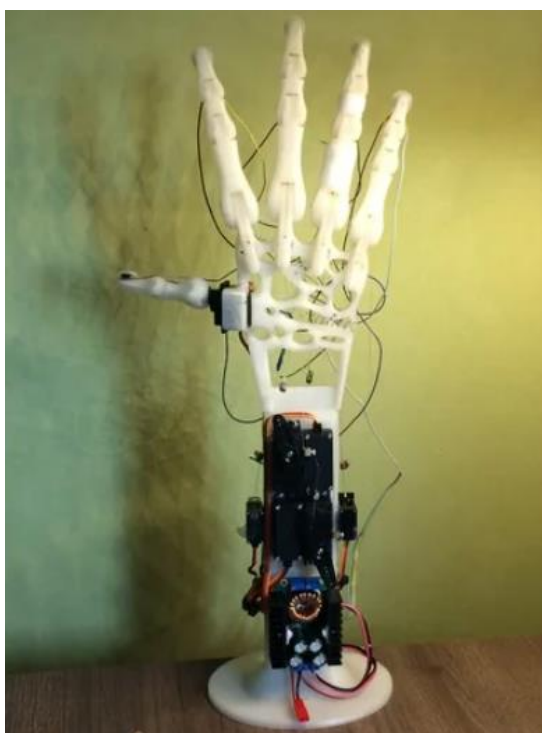
Esta seção apresenta os materiais, métodos e técnicas utilizadas durante o desenvolvimento do trabalho.

#### 3.1. PRÓTESE MECÂNICA

##### 3.1.1. MODELO TRIDIMENSIONAL

O modelo 3D CAD utilizado no projeto foi baseado em um modelo *open source* criado por Brian Brocken em Março de 2018. O projeto está disponível para acesso livre em [www.thingiverse.com](http://www.thingiverse.com), onde é possível realizar o download dos arquivos em formato STL. Na Figura 6 é possível visualizar o modelo construído por Brian.

Figura 6: Projeto de Brian Brocken



Fonte: (BROCKEN, 2018)

A partir da análise feita sobre o projeto de Brian, algumas características foram descartadas da implementação, mostrando-se desnecessárias para os objetivos desse trabalho.

### 3.1.2. FABRICAÇÃO POR MANUFATURA ADITIVA

Para impressão dos componentes da prótese, utilizou-se o processo de fabricação por filamento fundido (FFF) em uma impressora ANET A8. Por ser um material de baixo custo, boa resistência e fácil impressão, o filamento de PLA foi o escolhido para a fabricação.

Para traduzir o modelo STL em informações de movimentação para a impressora, foi utilizado o *software* Simplify3D. Nele, é possível inserir diversos arquivos, fazer simulações de impressão e gerenciar a melhor estratégia para fabricação das peças.

Antes de iniciar a fabricação dos componentes, é necessário preparar alguns parâmetros e detalhes de impressão, a fim de garantir uma geometria mais assertiva em relação ao modelo CAD.

Primeiramente, deve-se garantir a fixação do filamento fundido sobre a mesa da impressora 3D. Para isso, foi utilizado o método de aquecimento da mesa de deposição coberta por fita adesiva (*blue tape*). Apesar de não garantir um bom acabamento na superfície de contato, esse método fornece uma boa fixação para diferentes geometrias e torna a manutenção de impressão mais ágil.

Em seguida, deve-se realizar o nivelamento da mesa aquecida em relação ao extrusor da máquina, no caso da ANET A8, esse processo é manual. Para isso, posicionou-se o bico de extrusão em um dos cantos da mesa até que o espaço entre eles fosse relativo à espessura de uma folha A4. Em seguida, o extrusor foi movimentado até os demais cantos, ajustando os parafusos de nivelamento sempre que necessário.

Por fim, configuraram-se os parâmetros de entrada do *software* Simplify3D que garantem a movimentação da máquina e a extrusão ideal do material. Utilizaram-se camadas com 0,3 mm de altura, 200 °C para a fusão do material, 40 °C para o aquecimento da mesa e 15% de preenchimento interno.

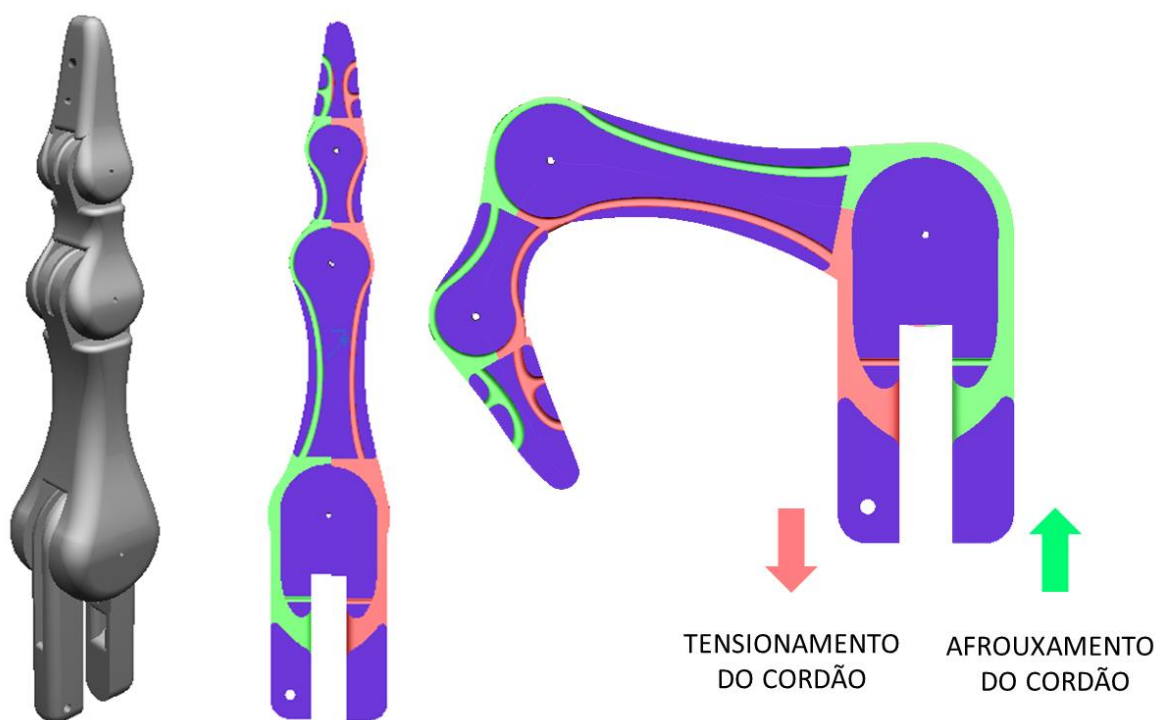
### 3.1.3. SISTEMA DE MOVIMENTAÇÃO

Para realizar a movimentação dos dedos da prótese, utilizou-se 5 servo motores de fácil integração com microcontroladores, sendo eles 2 Hobbico CS-60, 1 TowerPro MG995, 1 TowerPro MG-5010 e 1 TowerPro MG90S.

Para transmitir o movimento dos motores para as articulações, utilizou-se fios de nylon de alta resistência. Cada motor é responsável pela movimentação de um dedo, que através de 2 fios que ligam o eixo de rotação à extremidade do dedo, executam a ação de fechamento e abertura.

Na Figura 7 é apresentada uma vista de seção capaz de mostrar o efeito do fio de nylon sobre o dedo a partir do movimento do motor. É possível visualizar os canais internos presentes nas peças modeladas, que são responsáveis por garantir a livre movimentação dos cordões. A região em verde demonstra o afrouxamento do fio, enquanto a vermelha, o tensionamento do mesmo. Alterando o lado em que o fio é tensionado, é possível criar os movimentos de abertura e fechamento dos dedos.

Figura 7: Impacto do tensionamento dos cordões sobre os dedos da prótese



Fonte: Autor

Levando em consideração que o torque máximo dos motores selecionados em 5V é de 11 Kg.cm e a distância do eixo ao ponto inicial do fio é 3 cm, foi possível dimensionar o fio de nylon ideal para nossa aplicação utilizando a equação:

$$T = r \cdot F \quad (15)$$

Em que  $T$  é o toque do motor,  $r$  a distância e  $F$  a força aplicada.

O valor encontrado foi de 3,66 Kg. Logo, o fio de nylon comercial escolhido para a aplicação foi o Ekilon, com um diâmetro de 0,25 mm que suporta cargas de até 3,8 Kg.

#### **3.1.4. MONTAGEM**

Antes de realizar a montagem do protótipo, foi realizado um processo de lixamento nas peças impressas em PLA a fim de diminuir o atrito entre os componentes móveis da prótese.

Com todas as peças fabricadas, iniciou-se o processo de montagem. Primeiramente, montou-se todos os dedos de forma individual, inserindo o fio de nylon pelo interior das peças e conectando as juntas com pinos de aço. Em seguida, todos os dedos pré-montados foram conectados à palma da mão, que por sua vez foi ligada à base responsável por agregar os motores.

Com a estrutura principal montada, todos os motores foram posicionados de tal forma que a movimentação entre eles não gerasse nenhuma colisão. Em cada servo, foi encaixado a peça responsável por prender os fios de nylon e garantir a distância ideal do centro de giro do motor.

Por fim, todos os fios internos aos dedos foram fixados ao sistema do servo motor garantindo que todos eles não apresentassem uma folga extrema.

#### **3.1.5. TESTE MECÂNICO**

Com o protótipo pronto, iniciou-se a fase de testes mecânicos na movimentação do sistema. Cada motor foi alimentado eletricamente e conectado a um Arduino Mega, responsável por enviar os comandos de movimentação para os servos. Por ser um teste preliminar, os motores foram movimentados de forma individual a fim de encontrar o valor ideal dos ângulos de abertura e fechamento dos

dedos. Durante os testes, notaram-se interferências em algumas juntas de movimentação do sistema, sendo necessário desmontar e realizar um novo processo de lixamento em determinadas peças antes de iniciar a análise dos ângulos.

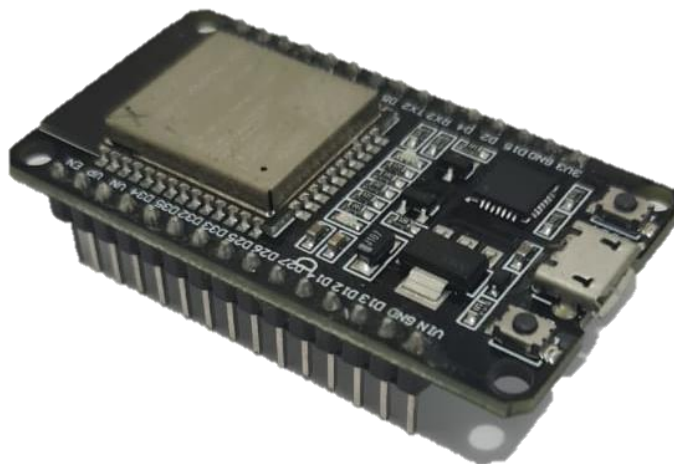
## 3.2. MÓDULO ELETRÔNICO

### 3.2.1. ESP32

O ESP32 é um pequeno microcontrolador capaz de desempenhar funções de comunicação sem fio por possuir tecnologias *bluetooth* e *wifi* integradas. Por possuir um processador de 2 núcleos, um dos seus grandes diferenciais em relação aos microcontroladores concorrentes é a sua alta capacidade de processamento. Esse *chip* vem se destacando cada vez mais em aplicações atuais envolvendo processamento de dados e internet das coisas.

Os tipos de ESP32 encontrados no mercado são variados, apresentando modelos simples com o *chip* unicamente, modelos em módulos com conexão USB e pinagem integrada e até mesmo alguns que apresentam telas e câmeras. O modelo utilizado nesse trabalho é o ESP32S, que possui conexão USB e pinagem externa como apresentado na Figura 8. Além disso, esse microcontrolador possui uma integração direta com a interface Arduino-IDE, possibilitando a sua fácil programação.

Figura 8: ESP32S



Fonte: Autor

### 3.2.2. WEMOS D1 R2

A WeMos D1 R2 é uma placa dedicada a projetos IoT que possui uma interface muito semelhante ao Arduino Uno, porém, controlada pelo módulo ESP8266. Assim como a ESP32, ela também possui um sistema de comunicação *wifi* integrado, apresentando um *hardware* com uma menor capacidade de processamento.

A WeMos também possui conexão USB integrada, como mostrado na Figura 9, e capacidade de comunicação com a interface Arduino-IDE.

Figura 9: WeMos D1 R2



Fonte: Autor

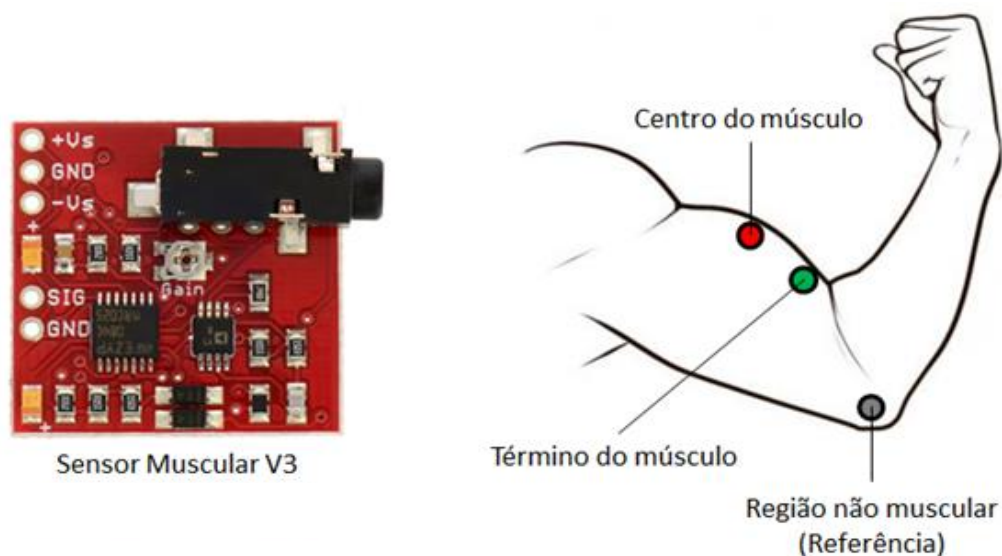
### 3.2.3. SISTEMA DE AQUISIÇÃO DO SINAL EMG

Para realizar a aquisição em tempo real do sinal EMG gerado pela movimentação muscular, foi utilizado o *Muscle Sensor V3*, que é capaz de calcular a diferença de potencial entre dois eletrodos de superfície e comparar com um terceiro eletrodo de referência. Esse componente trabalha a partir de uma tensão de entrada simétrica que pode variar de 3,5 V à 18 V e gera uma saída de tensão positiva que pode variar de zero até o valor máximo de entrada.

O V3 é um sensor de leitura e tratamento de sinais eletromiográficos voltado para aplicações em microcontroladores. Esse circuito é capaz de realizar a leitura da diferença de potencial dos músculos através de três sinais de entrada (três eletrodos). O primeiro eletrodo é responsável por ser a referência do sinal e deve ser posicionado em uma região não muscular do corpo, já os outros dois, são responsáveis por realizar a leitura da contração muscular, um deles deve ser posicionado no centro do músculo em análise, enquanto o outro no fim do mesmo. O sensor V3 e as configurações dos eletrodos são mostrados na Figura 10.



Figura 10: Sensor muscular V3 e posicionamento dos eletrodos



Fonte: Adaptado de (ADVANCER TECHNOLOGIES, 2013)

Por ser um sensor voltado a aplicações em microcontroladores, além de captar sinais eletromiográficos, o V3 também utiliza filtros eletrônicos para melhorar as respostas de saída através do uso de alguns amplificadores operacionais (AMPOPs). Antes de enviar o sinal de saída, o sinal eletromiográfico é retificado e suavizado, melhorando assim o potencial de análise e interpretação pelo microcontrolador. No Anexo A, pode-se observar os circuitos eletrônicos internos ao V3 que são responsáveis pela captação de sinais, retificação, suavização e saída do sinal.

Para a aquisição do sinal é utilizado um amplificador de instrumentação AD8226, responsável por realizar a leitura dos três eletrodos e amplificar o valor resultante. O eletrodo posicionado mais ao centro do músculo é conectado à entrada positiva do amplificador, o eletrodo mais à extremidade do músculo à entrada negativa e o eletrodo posicionado em uma região não muscular no pino de referência. Para configurar a amplificação do sinal, foi utilizado um resistor de  $240\Omega$  posicionado entre os pinos de ganho do amplificador de instrumentação ( $R_g$ ). Com esse valor, adquiriu-se um ganho de aproximadamente 207 vezes (o ganho pode ser calculado a partir da equação 16).

$$R_g = \frac{49,4 \text{ k}\Omega}{G - 1} \quad (16)$$

Onde G é o ganho e  $R_g$  o resistor utilizado.

Para o circuito de retificação de precisão são utilizados dois amplificadores operacionais, dois diodos e alguns resistores para realizar a modificação do sinal. Esse circuito é utilizado pois em aplicações EMG, onde a amplitude do sinal pode ser muito pequena, o uso de retificadores comuns se torna inviável, como por exemplo, o uso de ponte de diodos.

Já para realizar a suavização do sinal, um filtro passa-baixa é utilizado a fim de descartar sinais com frequência acima do limite de corte. Esse sistema é constituído por um amplificador operacional, um capacitor e alguns resistores. Como resultado, tem-se como saída, um sinal amplificado, retificado e suavizado que pode ser facilmente interpretado por um microcontrolador.

A fim de testar a capacidade de resposta do sinal eletromiográfico a partir de estímulos musculares, utilizou-se uma fonte simétrica de 5 V para alimentar o V3 e conectou-se o sinal de saída do mesmo à um osciloscópio digital. Com os eletrodos posicionados no antebraço, notou-se a rápida resposta do sinal em relação a diferentes esforços aplicados.

Em seguida, o sinal de saída foi conectado à uma entrada analógica de um Arduino Mega com o objetivo de testar a aquisição de dados e a resolução na conversão do sinal analógico para digital. Logo, percebeu-se que a resolução da leitura feita pelo microcontrolador era muito baixa, dificultando a diferenciação entre os movimentos musculares. Para calcular a resolução da conversão do sinal foi utilizando a equação:

$$R = \frac{V_{ref}}{2^n} \quad (17)$$

Em que  $R$  é a resolução do sinal,  $V_{ref}$  a tensão de referência e  $n$  o número de bits do microcontrolador.

Sabendo que as portas analógicas de entrada do Arduino Mega são de 10 bits e o sinal EMG máximo é 5 V, concluiu-se que a resolução de leitura era de 4,9 mV/bit. A fim de melhorar esse resultado, mudou-se a placa de aquisição para uma ESP32, cujas portas analógicas são de 12 bits e suportam 3,3 V como tensão de entrada. Sendo assim, a nova resolução de conversão calculada foi de 0,8 mV/bit (cerca de 6 vezes menor do que o anterior).

No entanto, devido a limitação da ESP32, foi necessário ajustar a tensão de entrada da porta analógica do microcontrolador (sinal EMG) de 5 V para 3,3 V. Para

isso, criou-se um divisor de tensão utilizando dois resistores, cujos valores foram calculados a partir da equação:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (18)$$

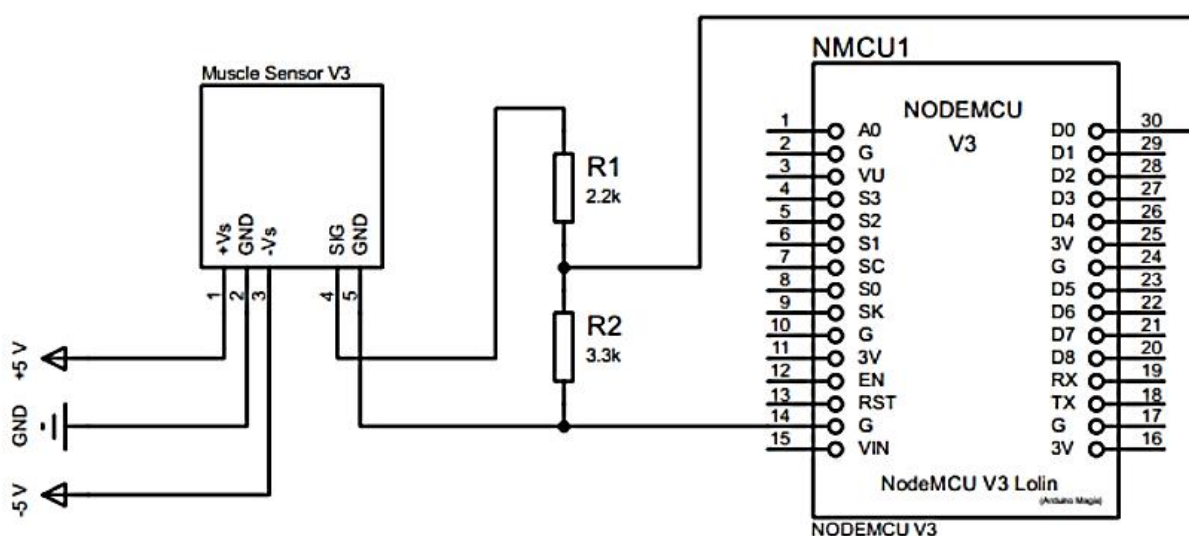
Onde  $V_{out}$  é a tensão de saída do divisor,  $V_{in}$  a tensão de entrada do divisor,  $R_1$  o resistor de menor resistência e  $R_2$  o resistor de maior resistência.

Buscando por resistores disponíveis, foi selecionado um com resistência de  $2200 \Omega$  ( $R_1$ ) e outro de  $3300 \Omega$  ( $R_2$ ).

Com essa nova configuração, foram realizados novos testes para analisar a resolução do sinal EMG, e como esperado, a diferenciação entre os movimentos musculares foi melhorada.

O circuito final do módulo de aquisição do sinal EMG é mostrado na Figura 11, onde a tensão de entrada foi adquirida através de uma fonte simétrica de bancada.

Figura 11: Circuito de captação do sinal EMG



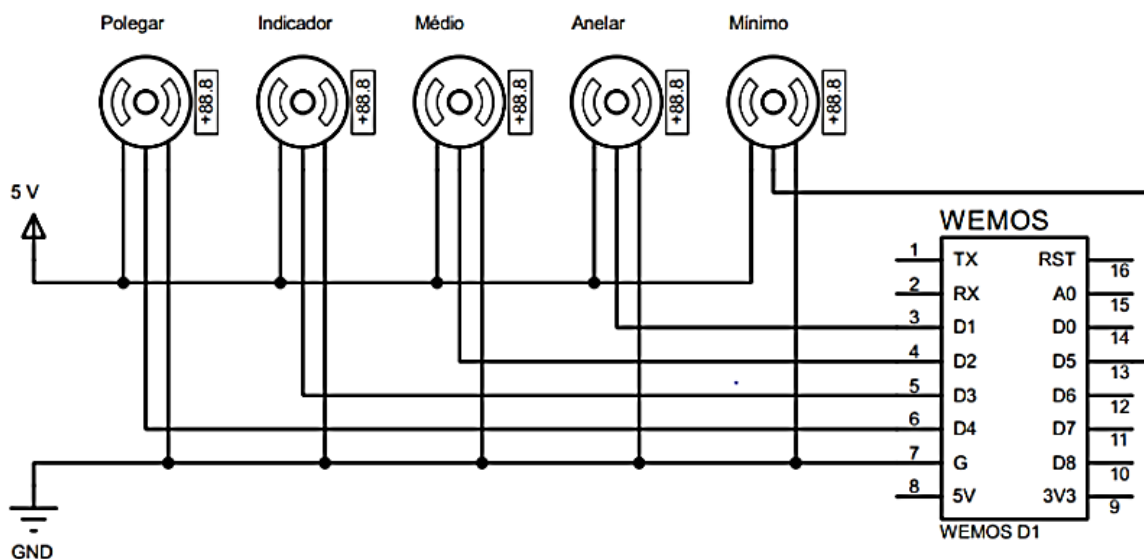
Fonte: Autor

### 3.2.4. SISTEMA DE ACIONAMENTO DOS MOTORES

Para o circuito que será responsável por receber as informações do sistema de aquisição do sinal EMG e controlar os motores do protótipo, foi utilizado a placa WEMOS D1 R2. Ela possui um controlador ESP8266 embarcado que possibilita a

conectividade *wifi*, sendo ideal para projetos IoT. Além disso, a placa possui diversas portas analógicas que podem ser utilizadas para realizar o controle de servo motores. O circuito final do módulo de acionamento dos motores é mostrado na Figura 12:

Figura 12: Circuito de controle dos motores



Fonte: Autor

### 3.3. INTERNET DAS COISAS

#### 3.3.1. SERVIDOR ESP32

Para que a leitura do sinal EMG possa ser disponibilizado na rede, possibilitando a leitura do mesmo por qualquer dispositivo com acesso à internet, utilizou-se a placa NodeMCU ESP32 (mesma responsável pela leitura do sinal eletromiográfico) para criação de um servidor *web*. Para isso, programou-se a interface Arduino IDE garantindo o reconhecimento da placa pelo mesmo.

Primeiramente, foi necessário configurar a conexão da ESP com a rede *wifi* local, garantindo à mesma, o total acesso à rede de internet. Utilizando a biblioteca *WiFi.h* foi possível criar um servidor *web* em uma determinada porta que é capaz de checar, em tempo real, possíveis solicitações de clientes. Com isso, o servidor já se tornou acessível por qualquer dispositivo com acesso à internet utilizando o IP do NodeMCU como URL.

Em seguida, programou-se a leitura do sinal EMG recebido pela porta de conversão analógica-digital. Para verificar a integridade dos valores recebidos, foram realizados testes imprimindo o sinal na saída serial do microcontrolador. Com o teste validado, foi possível disponibilizar o dado no servidor em tempo real.

Além disso, o servidor foi programado para ouvir todas as solicitações provenientes de clientes, retornando sempre a variável correta solicitada. Por exemplo, se o servidor ESP32 recebe um pedido do tipo HTTP GET no URL “\EMG” ele retornará ao cliente a variável presente no destino, no caso o valor do sinal EMG naquele instante.

### **3.3.2. CLIENTE ESP8266**

Para adquirir o sinal EMG presente no servidor *web*, foi necessário configurar um cliente *web* utilizando a placa WEMOS (mesma responsável por controlar os motores). Assim como no servidor, a interface Arduino IDE e a biblioteca *WiFi.h* também foram utilizadas para programar e garantir o acesso à internet respectivamente.

Em seguida, programou-se o cliente de tal forma que fosse possível captar o sinal presente no endereço *web* do servidor, através do método HTTP GET, em tempo real, possibilitando a interpretação do movimento muscular e conseqüentemente o controle dos motores do protótipo. Utilizou-se a biblioteca *Servo.h* para o controle dos servo motores, cujos ângulos variam de acordo com o sinal EMG recebido.

### **3.3.3. INTERFACE WEB**

Para poder realizar a leitura do sinal EMG e supervisionar o movimento da prótese em tempo real, necessitou-se a criação de uma interface web capaz de ser acessada por qualquer aparelho móvel conectado à internet. A IHM foi programada em HTML na interface Arduino IDE no lado do servidor (ESP32).

Foram criadas três páginas web para compor a interface, sendo elas o menu principal para acessar as demais funcionalidades, uma página contendo o sinal em

tempo real de forma gráfica e uma que supervisiona o movimento dos dedos da prótese. Cada uma se tornando uma extensão do endereço do servidor.

Para a primeira, necessitou-se apenas da criação de dois botões que direcionam o usuário para as demais páginas. Essa foi configurada como página inicial da interface *web*.

Já para a segunda, foi necessário criar uma variável em HTML para receber o sinal EMG lido pela placa ESP32, possibilitando a visualização do dado em tempo real. Para criar a interface gráfica do sinal utilizou-se a biblioteca *Highcharts*, uma ferramenta compatível com a linguagem HTML de fácil implementação. Foi utilizado um gráfico simples onde o eixo X é o tempo decorrido desde o início da aplicação e Y o valor do sinal eletromiográfico enviado pelo microcontrolador.

Para a terceira página, foi necessário criar 5 campos capazes de receber o valor do ângulo de cada motor, que são enviados pela aplicação cliente (WEMOS). Foi utilizado o método HTTP Post para enviar os dados do cliente para o servidor *web*.

### **3.4. ANÁLISE DOS DADOS EMG**

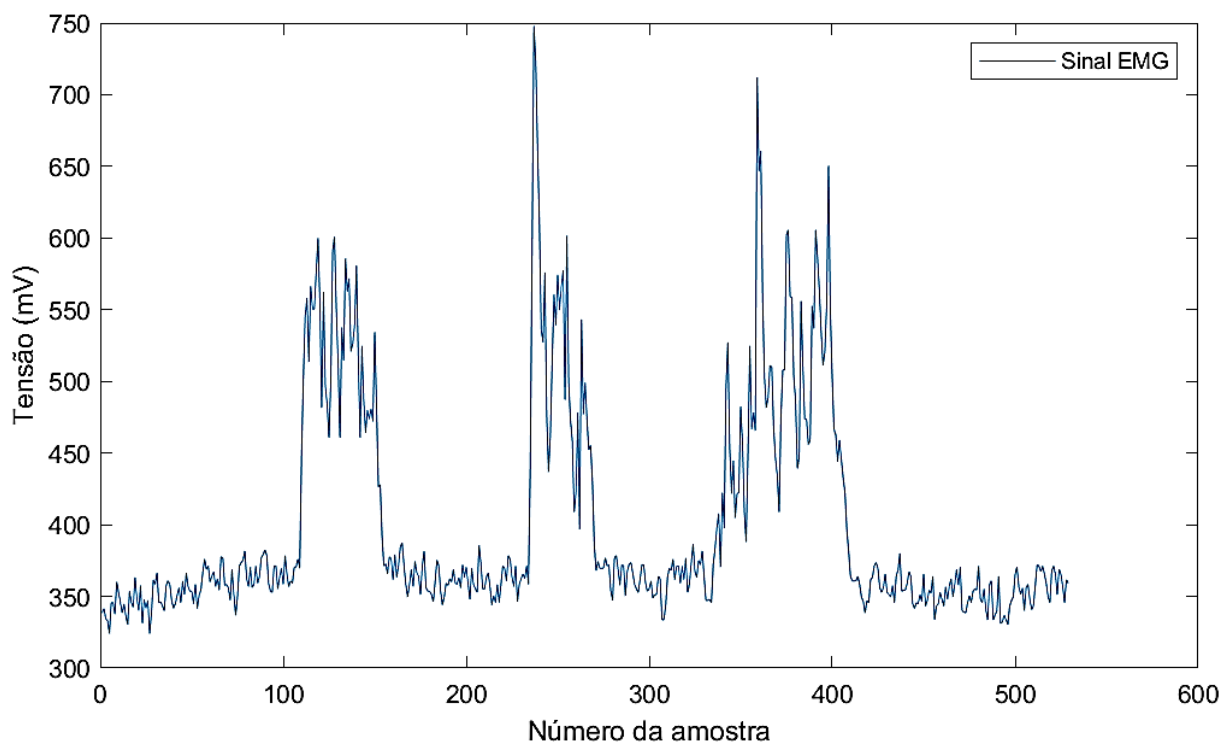
A fim de melhorar a interpretação do sinal proveniente da movimentação muscular, foi realizado um estudo sobre os melhores métodos de extração de características (*Feature Extraction*).

#### **3.4.1. EXTRAÇÃO DE CARACTERÍSTICAS OFFLINE**

Para realizar a análise do impacto de cada característica sobre o sinal EMG, foi feito um estudo de forma *offline* (desconectado do protótipo) utilizando o *software* MATLAB para realizar os cálculos matemáticos.

Primeiramente foi necessário criar um arquivo de texto contendo diferentes movimentos da mão e seus respectivos sinais eletromiográficos. Para isso, utilizou-se o *Muscle Sensor V3* conectado à ESP32 para adquirir os sinais e criar o arquivo de texto através da saída serial do microcontrolador. Foram mapeados 8 movimentos distintos, cada um com 3 estímulos conforme a Figura 13.

Figura 13: Sinal com 3 estímulos utilizado para o treinamento do algoritmo



Fonte: Autor

Para extrair as características de cada movimento no MATLAB, utilizou-se a ferramenta *EMG Feature Extraction Toolbox*, que é capaz de processar sinais EMG de até 40 maneiras diferentes a partir de um simples arquivo de texto. Para esse projeto, foram usadas as funções mais usuais no contexto de eletromiografia, incluindo os métodos *Wavelength*, *Mean Absolute Value*, *Slope Sign Change* e outros.

Para validar a proposta de Too, Abdullah e Saad (2019), foram criados dois novos métodos não existentes na ferramenta citada anteriormente, sendo eles o *Enhanced Mean Absolute Value* (EMAV) e o *Enhanced Wavelength* (EWL). Para criá-los, duas funções foram desenvolvidas no MATLAB seguindo as equações (13) e (14).

### 3.4.2. EXTRAÇÃO DE CARACTERÍSTICAS ONLINE

O projeto em questão propõe realizar o tratamento do sinal EMG e disponibilizá-lo em um servidor web em tempo real. Para isso é necessário extrair as

características durante o processo de aquisição e envio de dados, para posteriormente fornecer ao cliente o sinal devidamente tratado.

Assim, criou-se uma função na interface Arduino IDE da ESP32 capaz de guardar um volume de sinais EMG, lidos pelo microcontrolador, e utilizá-los para calcular o *Mean Absolute Value* (MAV), *Enhanced Mean Absolute Value* (EMAV), *Wavelength* (WL) e *Enhanced Wavelength* (EWL). Utilizando um conjunto de 20 dados conhecidos, foi possível validar a função criada comparando o resultado adquirido com o diretamente calculado pelo MATLAB.

### 3.4.3. MACHINE LEARNING

Com as características extraídas do sinal EMG em tempo real, é possível modelar uma forma de classificar os movimentos através de um algoritmo de *machine learning* alimentado por um sistema de treinamento.

Como visto na seção 2.5., o *machine learning* que obteve o melhor resultado nos experimentos de Too, Abdullah e Saad (2019) foi o *Linear Discriminant Analysis* (LDA), chegando a 97,56% de assertividade. No entanto, a implementação desse tipo de algoritmo em um microcontrolador utilizando a interface Arduino IDE não é tão trivial, diferentemente do *Support-Vector Machine* (SVM), que possui bibliotecas de integração direta com a ESP32. Outro fator levado em consideração foi o resultado da aplicação do SVM com as mesmas características, chegando a uma assertividade de 96,22%, um resultado muito bom e muito próximo ao adquirido pelo LDA.

Para criar o *machine learning* integrado à interface Arduino IDE, utilizou-se duas ferramentas muito conhecidas: LibSVM e Arduino-SVM. A primeira é responsável por receber os dados de treinamento do sinal e utilizá-los para construir o algoritmo de classificação dos dados. Já a segunda é capaz de interpretar o resultado encontrado pela LibSVM e criar um arquivo de integração com microcontroladores.

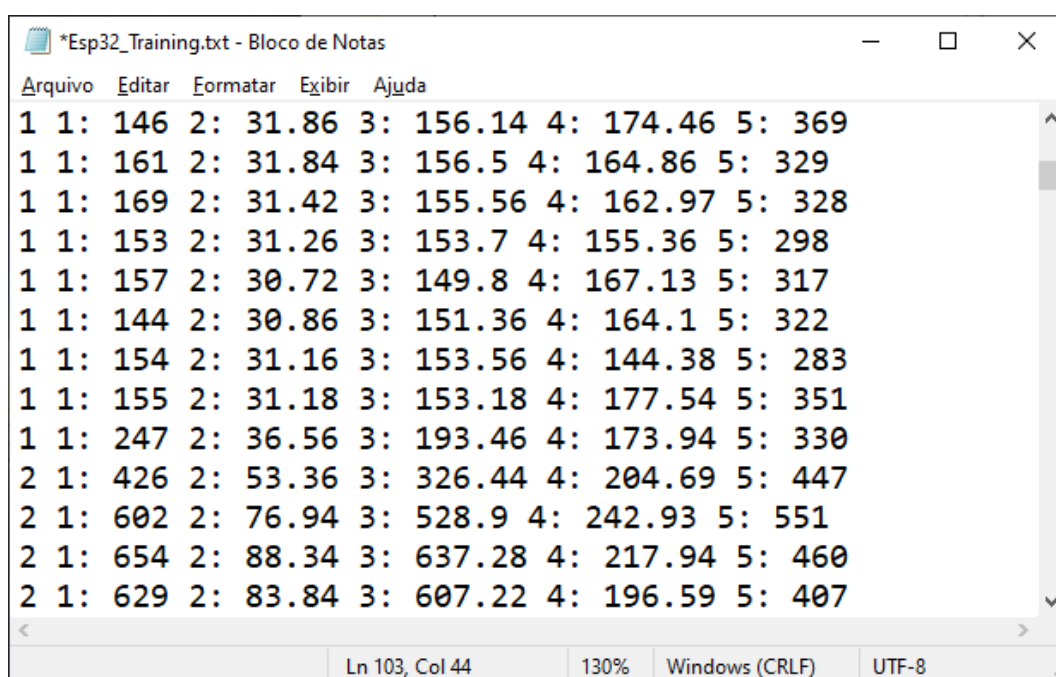
A fim de validar as ferramentas de *machine learning*, utilizou-se um sensor de luminosidade LDR conectado a um Arduino Mega com o objetivo de classificar a luminosidade do ambiente em 3 tipos: escuro, levemente claro e claro. Em seguida, foi criado um arquivo de treinamento para alimentar as ferramentas de criação do *machine learning*.



Com as ferramentas validadas, foi possível construir o modelo para o sistema de classificação do sinal EMG. Primeiramente foi necessário criar um arquivo de treinamento de acordo com as especificações da biblioteca LibSVM, relacionando o movimento executado com o valor do sinal EMG. Levando em consideração que quanto maior o número de dados de treinamento relevantes, mais confiável se torna o *machine learning*, foi criado um modelo com seis pulsos para cada tipo de movimento. Os dados armazenados no arquivo foram os recebidos pelo cliente, assegurando a confiabilidade na transmissão do sinal via internet.

O resultado da criação do arquivo de treinamento é mostrado na Figura 14, onde as colunas mostram o tipo de movimento, o valor do sinal EMG, MAV, EMAV, WL e EWL respectivamente.

Figura 14: Modelo de arquivo para treinar o algoritmo



```
*Esp32_Training.txt - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
1 1: 146 2: 31.86 3: 156.14 4: 174.46 5: 369
1 1: 161 2: 31.84 3: 156.5 4: 164.86 5: 329
1 1: 169 2: 31.42 3: 155.56 4: 162.97 5: 328
1 1: 153 2: 31.26 3: 153.7 4: 155.36 5: 298
1 1: 157 2: 30.72 3: 149.8 4: 167.13 5: 317
1 1: 144 2: 30.86 3: 151.36 4: 164.1 5: 322
1 1: 154 2: 31.16 3: 153.56 4: 144.38 5: 283
1 1: 155 2: 31.18 3: 153.18 4: 177.54 5: 351
1 1: 247 2: 36.56 3: 193.46 4: 173.94 5: 330
2 1: 426 2: 53.36 3: 326.44 4: 204.69 5: 447
2 1: 602 2: 76.94 3: 528.9 4: 242.93 5: 551
2 1: 654 2: 88.34 3: 637.28 4: 217.94 5: 460
2 1: 629 2: 83.84 3: 607.22 4: 196.59 5: 407
Ln 103, Col 44 130% Windows (CRLF) UTF-8
```

Fonte: Autor

Com o arquivo devidamente formatado, foi possível utilizá-lo como dado de entrada para a ferramenta LibSVM, responsável por escalar o sinal e criar o algoritmo de classificação. Tendo em vista a necessidade de classificar o modelo em diferentes movimentos, utilizou-se o tipo multi-classe da ferramenta SVM e a função de base radial (RBF) como *kernel* do *machine learning*.

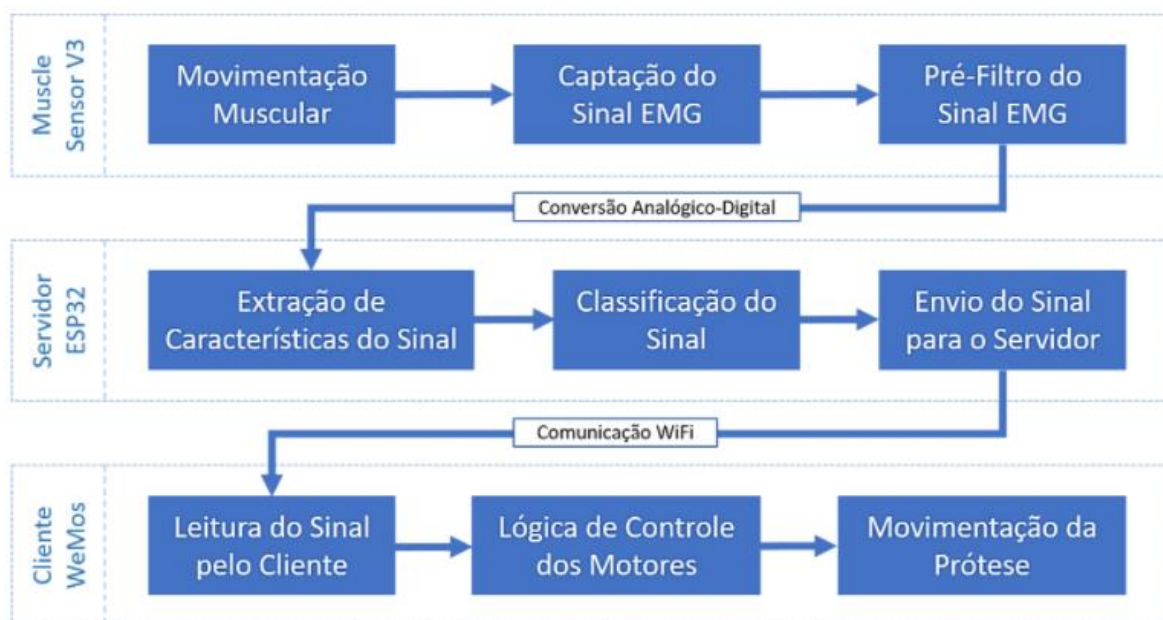
Para integrá-lo à interface Arduino IDE, utilizou-se a ferramenta Arduino-SVM, que tem como entrada o algoritmo de classificação resultante da ferramenta LibSVM e como saída um arquivo de comunicação direta com o *software* utilizado. Configurou-se o código do *machine learning* de tal forma que o mesmo recebesse o sinal EMG e as características extraídas do mesmo.



## 4. RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados encontrados a partir da utilização dos materiais e métodos abordados anteriormente. A Figura 15 apresenta o fluxo de funcionamento do sistema de forma simplificada, dividindo-o em três grandes grupos: a captação e pré-filtro do sinal EMG realizado pelo *Muscle Sensor V3*, a extração de características e classificação do sinal realizada pela ESP32 e o controle dos motores feito pelo WeMos D1 R2.

Figura 15: Fluxo de funcionamento do sistema



Fonte: Autor

### 4.1. PRÓTESE DE BAIXO CUSTO FABRICADA POR MANUFATURA ADITIVA

A fabricação das peças para construção do modelo mecânico através de manufatura aditiva mostrou-se suficiente para suprir o objetivo do projeto. Muitas peças não obtiveram um acabamento superficial perfeito, sendo necessário alguns ajustes após a fabricação. Esse erro está diretamente relacionado com a impressora utilizada (ANET A8) e os parâmetros adotados para a fabricação.

O uso dos fios de nylon de 0,25 mm foi ideal para a movimentação das articulações dos dedos, não apresentando falhas ou folgas durante o tensionamento.

Todos os motores utilizados também se mostraram efetivos, fornecendo o torque necessário para realização dos movimentos da mão.

O resultado final da fabricação e montagem do conjunto é mostrado na Figura 16:

Figura 16: Estrutura final da prótese



Fonte: Autor

Para identificar os ângulos de fechamento e abertura de cada dedo, os servomotores foram acionados de forma individual. Os resultados encontrados são mostrados na Tabela 3. Com a identificação desses valores, é possível criar funções que movimentam um conjunto de dedos para formar um gesto. Para fechar a mão, por exemplo, é necessário informar os valores de fechamento para todos os motores.

Tabela 3 - Ângulo de fechamento e abertura para cada servo motor.

DEDO	ÂNGULO DE FECHAMENTO	ÂNGULO DE ABERTURA
POLEGAR	0°	170°
INDICADOR	80°	165°
MÉDIO	120°	15°
ANELAR	120°	170°
MÍNIMO	70°	10°

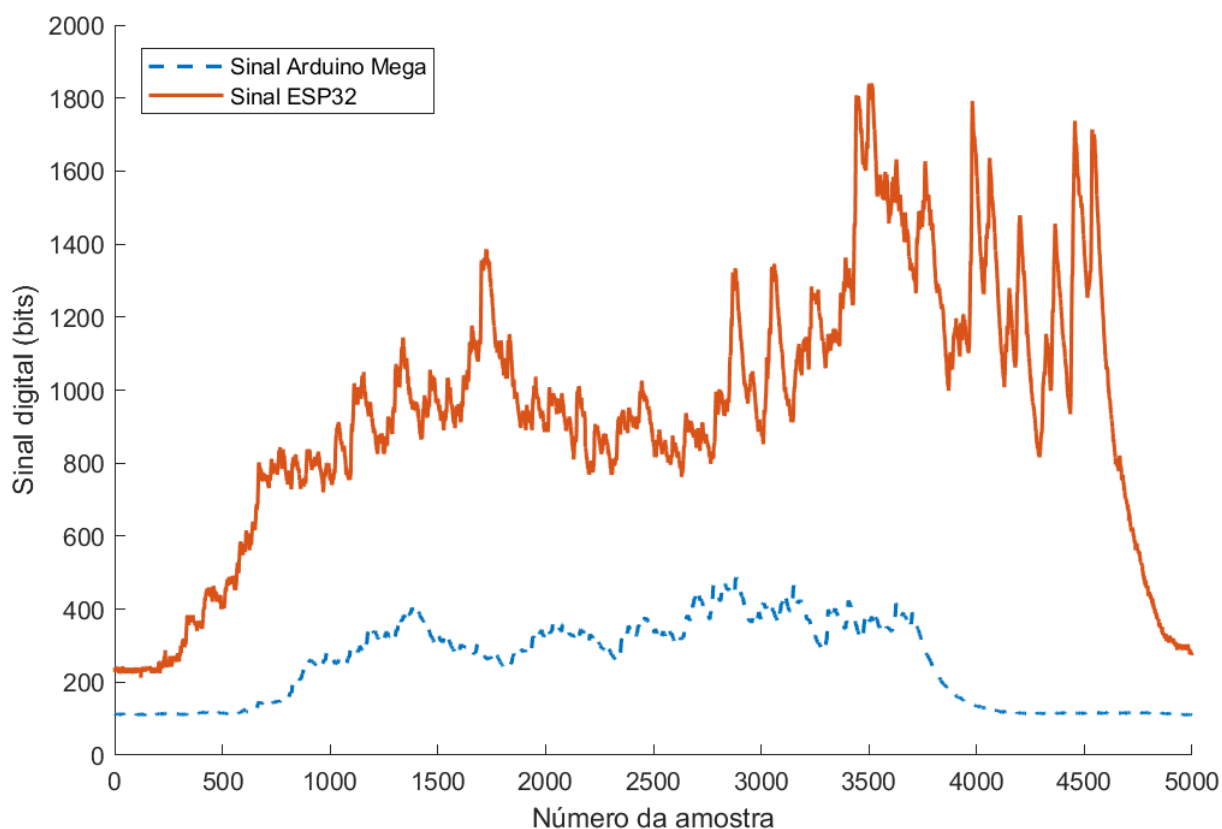
Fonte: Autor

## 4.2. MÓDULO ELETRÔNICO

Para avaliar e comparar o uso do Arduino Mega (10 bits) e o ESP32 (12 bits) durante a captação dos sinais EMG, foram criados 2 arquivos que contêm a resposta do *Muscle Sensor V3* para uma mesma contração muscular. O resultado desse estudo é apresentado na Figura 17, onde em vermelho é apresentado a leitura realizada pela ESP32 e em azul pelo Arduino. Aqui é possível notar que a diferença de resolução entre as placas é muito grande, tendo a primeira uma variação de aproximadamente 1600, enquanto o segundo apenas 300.

É importante ressaltar que nesse experimento foram utilizadas as tensões padrões de alimentação de cada um dos hardwares (fundo de escala), sendo 5V para o Arduino e 3,3V para a ESP32.

Figura 17: Comparação da resolução entre ESP32 e Arduino Mega



Fonte: Autor

A resolução de leitura do sinal EMG é muito importante para poder identificar pequenas variações na contração muscular e posteriormente utilizar essas informações para criar um sistema de classificação mais assertivo. Por esse motivo, o ESP32 foi o microcontrolador escolhido para desempenhar a função de leitura do sinal EMG.

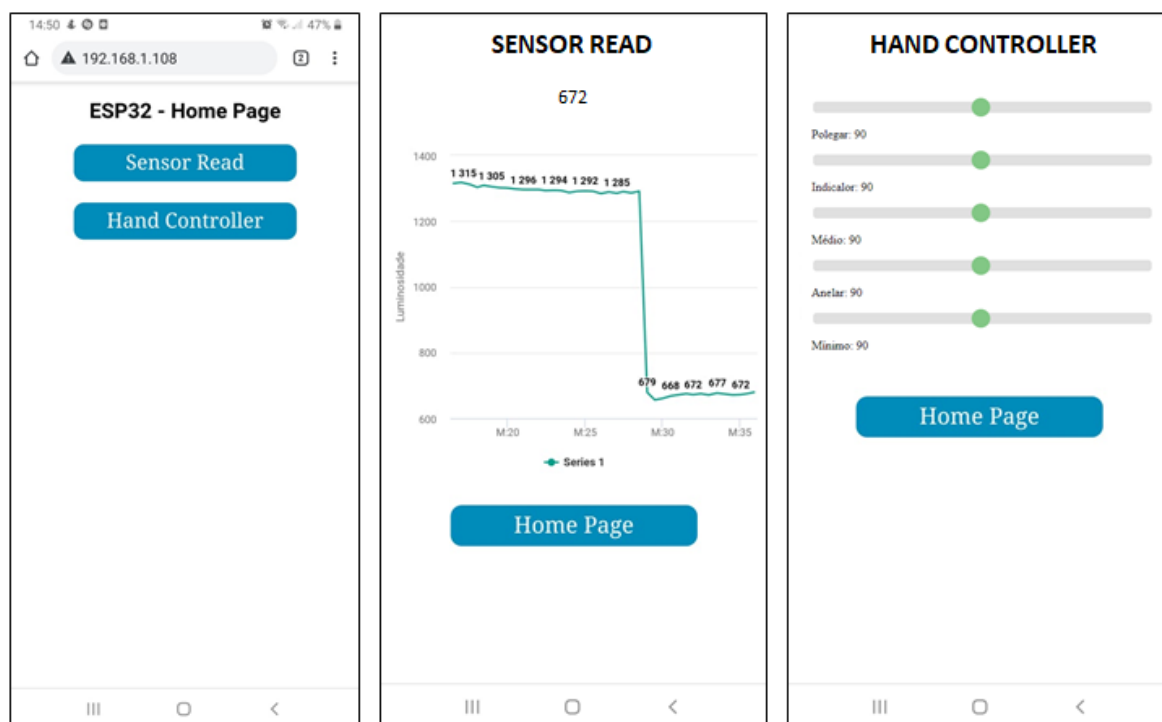
### 4.3. INTERNET DAS COISAS

Após a realização de todas as configurações do servidor (ESP32) e cliente (WeMos), foi realizado alguns teste para avaliar a integridade na comunicação entre as duas placas. O resultado obtido foi satisfatório, concretizando uma transmissão rápida e confiável de dados. O código fonte para estruturação do servidor, utilizando a ESP32, é apresentado no Apêndice A.

Com a estrutura IoT finalizada, testou-se o uso da interface web. Na Figura 18 é possível visualizar o resultado encontrado para as três páginas *web* anteriormente

configuradas. A página de visualização do sinal EMG conseguiu suportar a grande entrada de dados, apresentando-os de forma gráfica. Já a terceira, conseguiu enviar corretamente o sinal para movimentação de cada um dos servos, nesse caso, sem o uso do sinal EMG.

Figura 18: Interface web de comunicação



Fonte: Autor

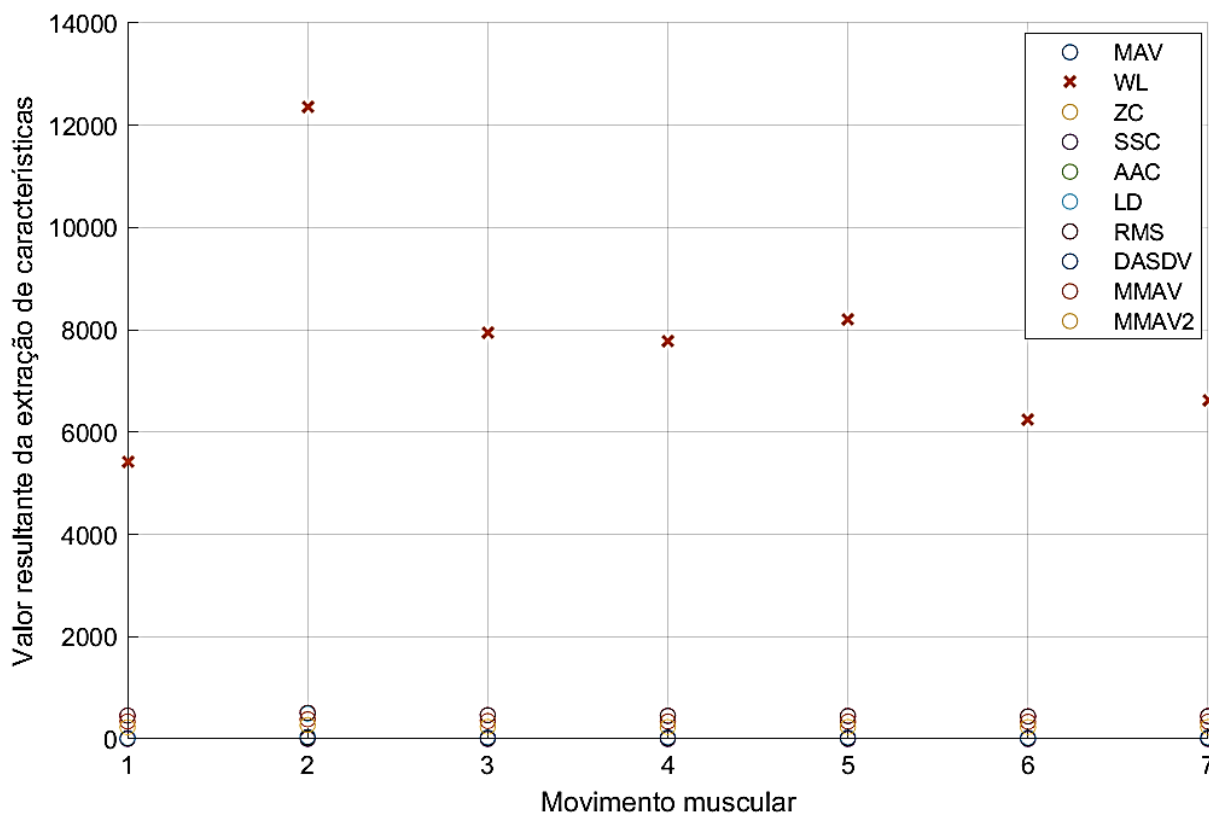
## 4.4. TRATAMENTO DO SINAL EMG

### 4.4.1. EXTRAÇÃO DAS CARACTERÍSTICAS DO SINAL DE FORMA OFFLINE

Como descrito em 3.4.1., utilizou-se a ferramenta EMG *Feature Extraction Toolbox* integrada com o software MATLAB para comparar os diferentes tipos de características do sinal. O resultado encontrado é mostrado na Figura 19, onde o eixo Y é valor resultante da extração de características e o eixo X o tipo do movimento (1 - repouso; 2 - Punho fechado; 3 - Polegar; 4 - Indicador; 5 - Médio; 6 - Anelar; 7 - Mínimo):



Figura 19: WL na extração de características para 7 movimentos

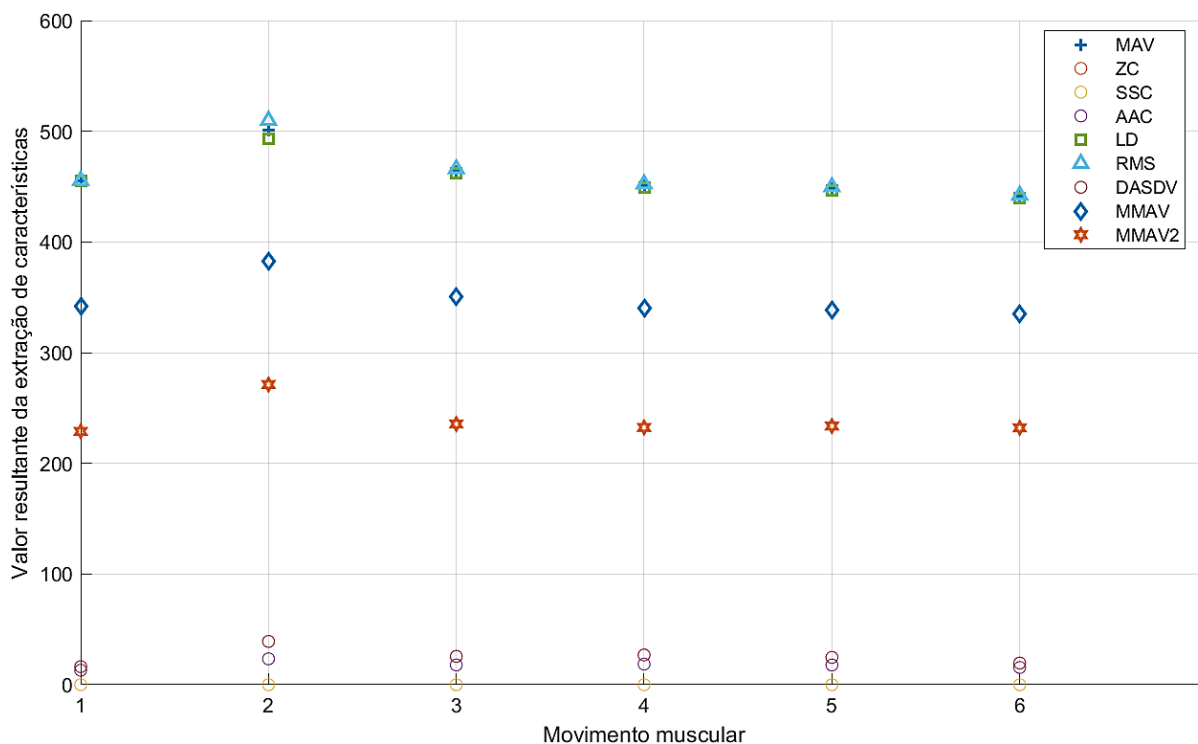


Fonte: Autor

A partir da análise gráfica, é possível perceber que a característica WL se diferencia muito das demais, tendo uma boa diferenciação entre os tipos de movimentos. Entretanto, nota-se que os movimentos 3, 4 e 5 possuem valores muito próximos, o que dificulta a diferenciação entre eles, o mesmo ocorre para os movimentos 6 e 7.

Em seguida, retirou-se a característica WL a fim de analisar o comportamento das demais características. O resultado é mostrado na Figura 20.

Figura 20: Demais características para 7 movimentos

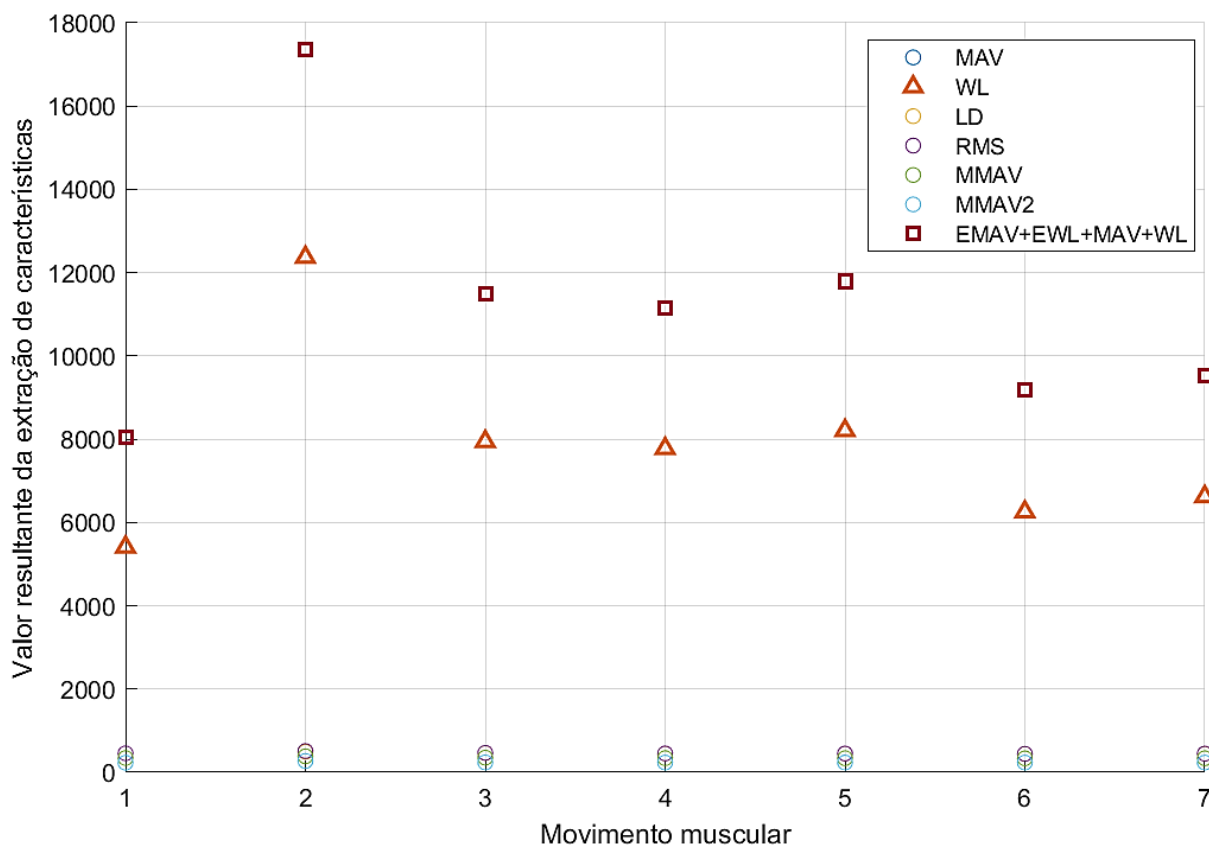


Fonte: Autor

Logo, notou-se que MAV, MMAV, MMAV2, RMS e LD se comportam de forma semelhante em relação a distribuição dos valores referentes aos tipos de movimentos. Também é possível perceber que todos os movimentos, com exceção do 2, estão presentes em uma mesma faixa de valores, o que dificulta a diferenciação entre eles.

Por fim, criou-se uma nova variável combinando 4 diferentes características (MAV, WL, EMAV e EWL), conforme a proposta de Too, Abdullah e Saad (2019). O gráfico resultante é mostrado na Figura 21.

Figura 21: Resultado da combinação de características



Fonte: Autor

Avaliando os pontos criados por essa nova variável, percebeu-se a melhoria no distanciamento entre os tipos de movimentos, tornando-se o melhor dado de classificação até então.

#### 4.4.2. EXTRAÇÃO DAS CARACTERÍSTICAS DO SINAL DE FORMA ONLINE

Como descrito em 3.4.2., foi necessário desenvolver um sistema de extração de características em tempo real (MAV, EMAV, WL e EWL) durante a captação do sinal no módulo ESP32. Para avaliar o funcionamento das funções criadas, criou-se um vetor com 10 valores conhecidos e calculou-se o valor resultante das 4 características através do MATLAB. Em seguida, o mesmo vetor foi inserido como dado de entrada nas funções criadas no Arduino IDE e o resultado extraído foi

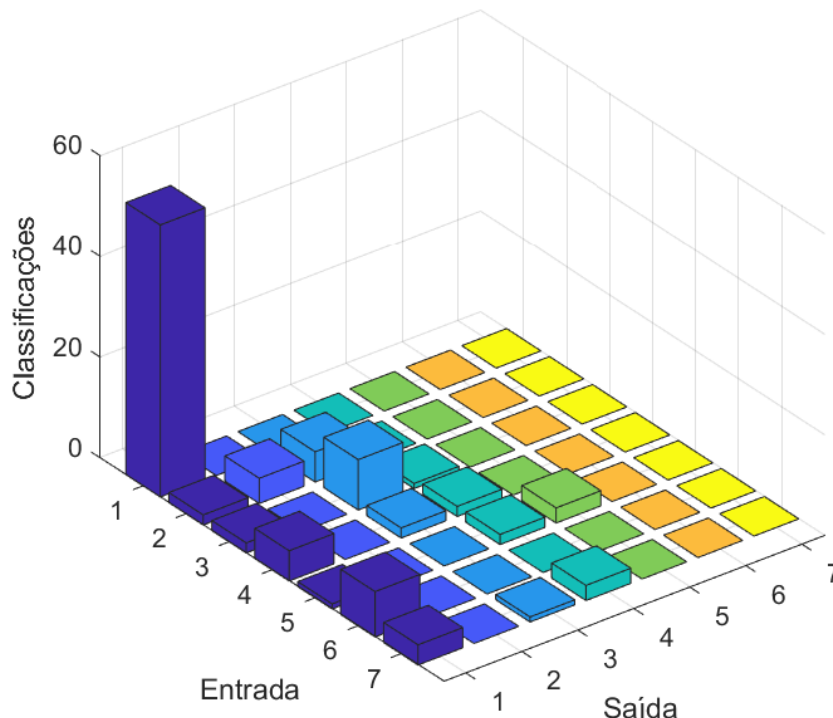
exatamente o mesmo calculado previamente pelo MATLAB, comprovando a efetividade da extração de características em tempo real pela ESP32.

## 4.5. CLASSIFICAÇÃO DO SINAL EMG

### 4.5.1. MODELO COM 7 MOVIMENTOS

Para classificar o sinal EMG em 7 tipos (repouso, punho fechado, polegar, indicador, médio, anelar e mínimo) utilizou-se o método SVM. Para avaliar a assertividade do modelo classificador, criou-se um arquivo de teste com aproximadamente 30% do volume de dados do arquivo de treino. O resultado do classificador sobre o arquivo de teste foi de 65,49%. Na Figura 23 é mostrado a matriz de confusão resultante do processo de classificação.

Figura 22: Matriz de confusão para classificação de 7 movimentos



Fonte: Autor

É possível perceber que a classificação se torna muito ruim dependendo do tipo de movimento executado. Os movimentos 2, 3 e 4, por exemplo, acabam sendo muito semelhantes, o que dificulta a correta classificação. O mesmo pode ser visto nos movimentos 6 e 7, que na maioria das vezes foi classificado como 1.

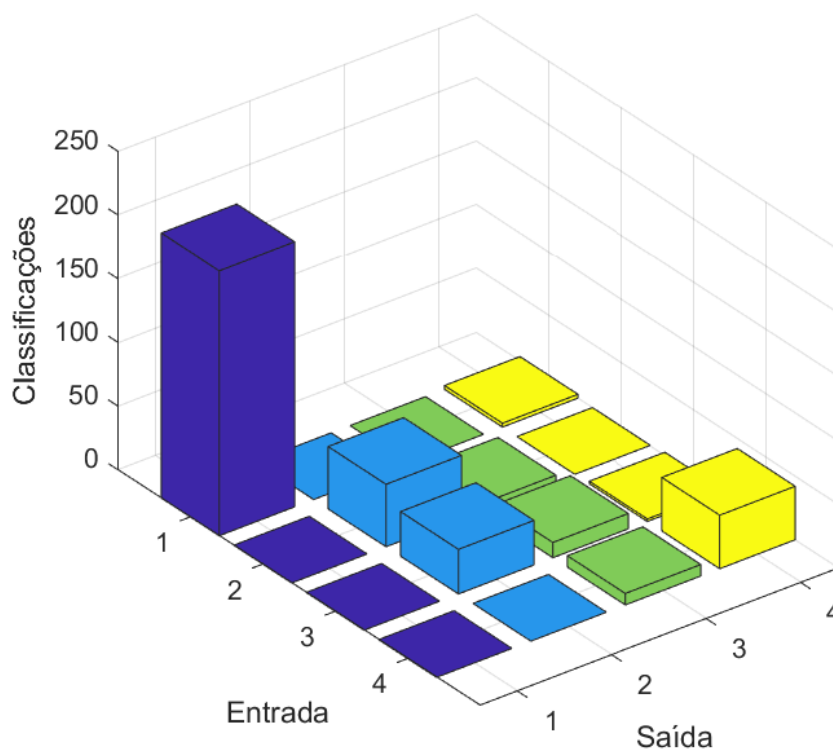
Esse problema de classificação está diretamente ligado ao sensor utilizado para captar o sinal EMG, que no caso, possui apenas um ponto de leitura. Por se tratar de movimentos muito sutis, a análise de apenas um músculo acaba impossibilitando uma melhor classificação. O uso de mais pontos de leitura, em outros músculos do antebraço, potencializaria a capacidade de classificação dos gestos em questão.

#### **4.5.2. MODELO COM 4 MOVIMENTOS**

A fim de criar um classificador mais assertivo utilizando apenas um ponto de leitura do sinal EMG, elaborou-se um novo modelo com 4 movimentos mais distintos entre si (repouso, punho fechado, polegar e polegar/indicador).

Usando a mesma técnica demonstrada em 4.5.2, o resultado da assertividade do modelo foi de 85,21%. Na Figura 24 é mostrado a matriz de confusão resultante do processo de classificação.

Figura 23: Matriz de confusão para classificação de 4 movimentos



Fonte: Autor

Dessa vez, é possível perceber uma maior assertividade na classificação dos movimentos. 1, 2 e 4 tiveram respostas muito boas, tendo quase 100% das vezes, uma resposta correta. Por outro lado, o movimento 3 obteve uma resposta ruim, sendo classificado como 2 na maioria das vezes.

O SVM se mostrou muito eficaz em realizar classificações, no entanto a assertividade encontrada para o reconhecimento dos movimentos não foi muito alta. Isso não está relacionado com o tipo de classificador escolhido, mas sim ao sensor responsável pela captação do sinal EMG. O Muscle Sensor V3 suporta o supervisionamento de apenas um músculo, e os movimentos individuais de alguns dedos acabam gerando respostas muito semelhantes sobre esse ponto de aquisição, impossibilitando uma melhor classificação.

## 5. CONCLUSÃO E TRABALHO FUTURO

Não há dúvidas que o avanço da tecnologia vem proporcionando melhorias em aplicações médicas, possibilitando até mesmo a correção de deficiências motoras a partir do uso de próteses. Esses dispositivos vêm sendo um ponto focal de estudos e melhorias, que acabam integrando diferentes técnicas de análise de dados, captação de sinais, controle e biomecânica. Apesar de ser um conceito antigo, a eletromiografia vem se destacando nas aplicações modernas que envolvem o controle de mecanismos a partir de movimentos humanos.

Nesse contexto, o primeiro objetivo específico foi validar a possibilidade de fabricar um dispositivo de baixo custo, utilizando tecnologias de manufatura aditiva, capaz de realizar movimentos da mão humana. A utilização das técnicas de fabricação e montagem descritas no trabalho foram suficientes para alcançar a meta, tendo como resultado um protótipo de baixo custo, baixo peso e capaz de realizar movimentos essenciais da mão humana.

Como segundo objetivo específico, foi proposto a utilização de técnicas IoT para realizar a transmissão e gerenciamento do sinal EMG. Por possuir uma alta capacidade de processamento e uma boa resolução de leitura, a ESP32 se mostrou um ótimo *hardware* para aquisição e processamento dos dados eletromiográficos, possibilitando também a criação de um servidor web responsável por gerenciar a transmissão de dados.

Estudar o comportamento do sinal EMG e utilizar técnicas de análise de dados para o seu melhoramento foi o terceiro objetivo específico proposto. As extrações de características utilizadas sobre o sinal foram fundamentais para melhorar o reconhecimento de padrões. Como demonstrado nos resultados, a técnica de *Wavelegth* (WL) foi a que mais se destacou, possibilitando um melhor reconhecimento dos movimentos.

Por último, como objetivo específico, foi proposto o desenvolvimento de um classificador de movimentos utilizando aprendizado de máquina. Muitas das referências citadas ao longo do trabalho fazem menção ao uso de múltiplos pontos de aquisição do sinal EMG para criação de um classificador. No entanto, pode-se

concluir que a utilização de apenas um ponto de observação torna possível, mas limitado, a criação de um classificador de movimentos da mão humana.

Com o desfecho dos objetivos anteriores, foi possível realizar a integração de todas as tecnologias propostas, tendo como resultado, uma prótese de baixo custo controlada por eletromiografia e internet das coisas, validando o objetivo geral do presente trabalho.

Propõe-se como trabalho futuro o estudo e aplicação de múltiplos pontos de aquisição do sinal EMG, para que assim seja possível a criação de um classificador mais assertivo e com uma maior capacidade de reconhecimento de padrões.



## REFERÊNCIAS

KAMEN, Gary; GABRIEL, David A.. **Essentials of Electromyography**. Massachusetts: Phorte Editora Ltda, 2015. 266 p.

CHAPPELL, Paul H. **Mechatronic Hands - Prosthetic and Robotic Design**. Londres: Institution of Engineering and Technology, 2016. 167 p.

TOO, Jingwei; ABDULLAH, Abdul Rahim; SAAD, Norhashimah Mohd. **Classification of Hand Movements based on Discrete Wavelet Transform and Enhanced Feature Extraction**. Malaca, Malásia: International Journal Of Advanced Computer Science And Applications, 2019. 7 p.

MIOTEC (org.). **Como utilizar a eletromiografia de superfície na saúde do trabalhador?** 2019. Disponível em: <https://blog.miotec.com.br/eletromiografia-de-superficie-na-saude-do-trabalhador/>. Acesso em: 18 nov. 2020.

MERLETTI, Roberto; PARKER, Philip. **ELECTROMYOGRAPHY: physiology, engineering, and noninvasive applications**. Nova Jersey, Estados Unidos: IEEE, 2004. 494 p.

JAMAL, Muhammad Zahak. **Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis**. Paquistão: Intech, 2012.

SPIEWAK, Christopher *et al.* **A Comprehensive Study on EMG Feature Extraction and Classifiers**. Wisconsin, Estados Unidos: Lupine, 2018.

VOLPATO, Neri. **MANUFATURA ADITIVA: tecnologias e aplicações da impressão 3d**. Sp: Blusher, 2017.

ABREU, Sofia Alexandra Chaves. **Impressão 3D baixo custo versus impressão em equipamentos de elevado custo**. 2015. 235 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade do Porto, Portugal, 2015.

GSMA (org.). **Understanding the Internet of Things (IoT)**. Londres: Gsma, 2014.

EVANS, Dave. **A Internet das Coisas: como a próxima evolução da internet está mudando tudo**. Cisco, 2011.

MDN (org.). **HTTP**. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em: 04 fev. 2021.

MOHRI, Mehryar; ROSTAMIZADEH, Afshin; TALWALKAR, Ameet. **Foundations of Machine Learning**. Massachusetts: Mit, 2012.

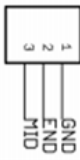
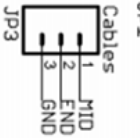
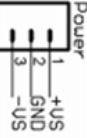
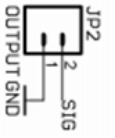
PETERSON, Leif E.. **K-nearest neighbor**. 2009. Disponível em: [http://scholarpedia.org/article/K-nearest\\_neighbor](http://scholarpedia.org/article/K-nearest_neighbor). Acesso em: 25 fev. 2021.

MUKHOPADHYAY, S.C.; HUANG, R.Y.M. **Sensors - Advancements in Modeling, Design Issues, Fabrication and Practical Applications**. Berlim: Springer, 2008.

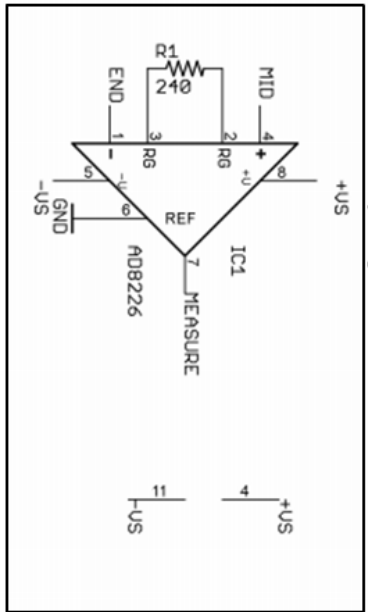
ADVANCER TECHNOLOGIES. **Muscle Sensor V3**: three-lead differential muscle/electromyography sensor for microcontroller applications, 2013.

BROCKEN, Brian. **Robot Hand**: bionic hand prosthesis prototype. Bionic hand prosthesis prototype. 2018. Disponível em: <https://www.thingiverse.com/thing:2838239>. Acesso em: 08 out. 2020.

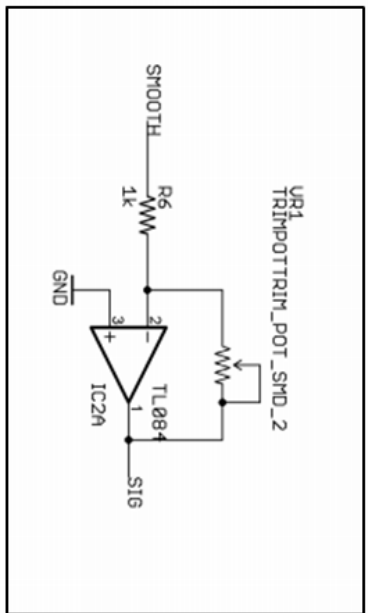
## ANEXO A – Circuito Eletrônico do Muscle Sensor V3



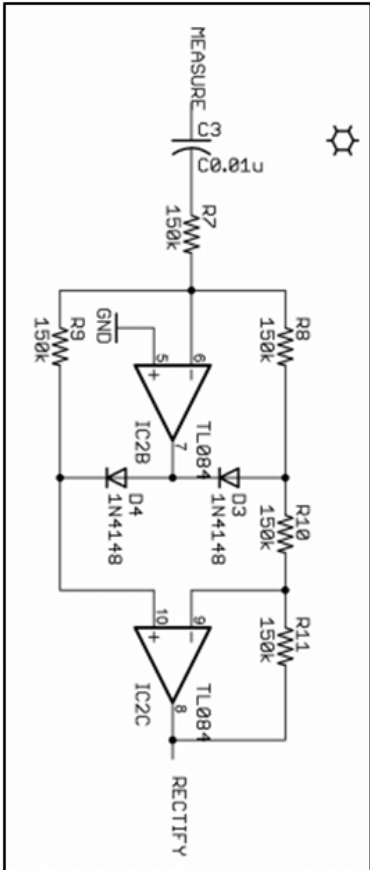
Captação do sinal



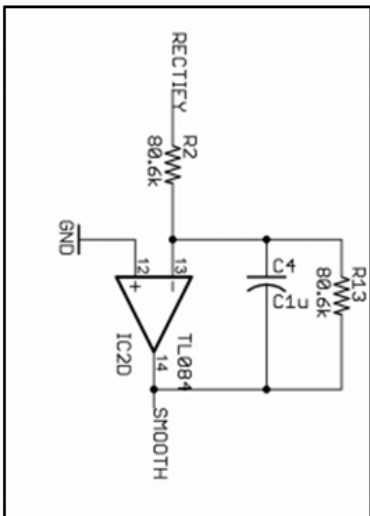
Saída do sinal



Retificação do sinal



Suavização do sinal



Brian E Kaminski

Advancer Technologies

TITLE: Muscle Sensor v3

DATE: 20June2012

SHEET: 1/1

## APÊNDICE A – Código do servidor para ESP32

```

// Bibliotecas WiFi
#include "WiFi.h"
#include "ESPAsyncWebServer.h"

// Acesso WiFi
const char* ssid = "agh";
const char* password = "08011997";

IPAddress gateway(192, 168, 0, 230);
IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(8, 8, 8, 8);

int EMGpin = 34;
String EMG;
String sinalEMG;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

String readEMG() {
  int leitura = analogRead(EMGpin);
  return String(leitura);
}

//Array para a extração de características online
int cont = 0;
const int ArrayLength = 50;
int Array[ArrayLength];

//-----HTML_Página_Inicial-----
char HomePage[] PROGMEM = R"=====(
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://code.highcharts.com/highcharts.js"></script>
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>ESP32 - Home Page</title>
<style>
body {
  background-color: white;
  min-width: 310px;
  max-width: 800px;
  height: 400px;
  margin: 0 auto;
}
p {

```

```

margin: 20px auto;
font-family: verdana;
font-weight: bold;
color: black;
font-size: 25px;
text-align: center;
}
.button {
background-color: #008CBA;
border: none;
color: white;
padding: 8px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 24px;
margin: 25px auto;
display: block;
cursor: pointer;
border-radius: 12px;
width: 200px;
}
</style>
</head>
<body>

```

```

<p>ESP32 - Home Page</p>
<a href="/SensorRead" class="button">Sensor Read</a>
<a href="http://192.168.1.109" class="button">Hand Controller</a>

```

```
<script>
```

```
</script>
```

```
</body>
```

```
</html>
```

```
)=====";
```

```
//-----HTML_Página_Gráfico-----
```

```
char SensorRead[] PROGMEM = R"=====
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<script src="https://code.highcharts.com/highcharts.js"></script>
```

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

```
<title>Sensor Read</title>
```

```
<style>
```

```
body {
```

```
background-color: white;
```

```
min-width: 310px;
```

```

max-width: 800px;
height: 400px;
margin: 0 auto;
}
h1 {
margin: 20px auto;
font-family: verdana;
font-size: 25px;
text-align: center;
}
p {
margin: 20px auto;
font-family: verdana;
font-size: 15px;
text-align: center;
}
.button {
background-color: #008CBA;
border: none;
color: white;
padding: 8px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 24px;
margin: 20px auto;
display: block;
cursor: pointer;
border-radius: 12px;
width: 200px;
}
</style>
</head>
<body>

<h1>Sinal EMG:</h1>
<p>
  <span id="EMG">%EMG%</span>
</p>
<div id="chart-EMG" class="container"></div>

<a href="/" class="button">Home Page</a>

<script>
var chartT = new Highcharts.Chart({
  chart:{ renderTo : 'chart-EMG' },
  title: { text: 'LDR Read' },
  series: [{
    showInLegend: true,
    data: []
  }

```

```

    }],
    plotOptions: {
      line: { animation: true,
        dataLabels: { enabled: true }
      },
      series: { color: '#059e8a' }
    },
    xAxis: { type: 'datetime',
      dateTimeLabelFormats: { second: '%H:%M:%S' }
    },
    yAxis: {
      title: { text: 'EMG' }
    },
    credits: { enabled: false }
  });

```

```

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var x = (new Date()).getTime(),
          y = parseFloat(this.responseText);
      //console.log(this.responseText);
      if(chartT.series[0].data.length > 40) {
        chartT.series[0].addPoint([x, y], true, true, true);
      } else {
        chartT.series[0].addPoint([x, y], true, false, true);
      }
    }
  };
  xhttp.open("GET", "/sinalEMG", true);
  xhttp.send();
}, 500 );
</script>

```

```

</body>
</html>
)=====";
//-----
char webpage[] PROGMEM = R"=====(
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://code.highcharts.com/highcharts.js"></script>
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Chart Js - Graphics</title>
<style>
body {
  background-color: lightblue;

```

```

    min-width: 310px;
    max-width: 800px;
    height: 400px;
    margin: 0 auto;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
    text-align: center;
}
</style>
</head>
<body>

<h1>HTML SERVER - ESP32</h1>
<p>Sinal EMG:</p>
<p>
    <span id="Sinal EMG: ">%EMG%</span>
</p>
<div id="chart-EMG" class="container"></div>

<script>
var chartT = new Highcharts.Chart({
    chart:{ renderTo : 'chart-EMG' },
    title: { text: 'LDR luminosidade' },
    series: [{
        showInLegend: true,
        data: []
    }],
    plotOptions: {
        line: { animation: true,
            dataLabels: { enabled: true }
        },
        series: { color: '#059e8a' }
    },
    xAxis: { type: 'datetime',
        dateTimeLabelFormats: { second: '%H:%M:%S' }
    },
    yAxis: {
        title: { text: 'Chart - LDR' }
    },
    credits: { enabled: false }
});

```



```

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var x = (new Date()).getTime(),
          y = parseFloat(this.responseText);
      //console.log(this.responseText);
      if(chartT.series[0].data.length > 40) {
        chartT.series[0].addPoint([x, y], true, true, true);
      } else {
        chartT.series[0].addPoint([x, y], true, false, true);
      }
    }
  };
}, 500 );
</script>

```

```

</body>
</html>
)=====";
//-----

```

```

String processor(const String& var){
  //Serial.println(var);
  if(var == "EMG"){
    return readEMG();
  }
  return String();
}

```

```

void printWifiStatus() {
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  Serial.print("Gateway: ");
  Serial.println(WiFi.gatewayIP());
  Serial.print("Subnet: ");
  Serial.println(WiFi.subnetMask());
  Serial.print("Primary DNS: ");
  Serial.println(primaryDNS);

  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);

```

```

Serial.println(" dBm");
}

//Funções para extração de características online
//EMAV FEATURE EXTRACTION
double EMAV(int EMAV_array[], int n)
{
  double x = 0;
  double p = 0;
  for(int i=1;i<=n;i++)
  {
    int c = i - 1;
    if((i>=(n*0.2))&&(i<=(n*0.8)))
    {
      p = 0.75;
    }
    else
    {
      p = 0.5;
    }
    x += abs(pow((EMAV_array[c]), p));
  }

  return (x/n);
}

//EWL FEATURE EXTRACTION
double EWL(int EWL_array[], int n)
{
  double x = 0;
  double p = 0;
  for(int i=2;i<=n;i++)
  {
    int c = i - 1;
    if((i>=(n*0.2))&&(i<=(n*0.8)))
    {
      p = 0.75;
    }
    else
    {
      p = 0.5;
    }
    double r = (EWL_array[c] - EWL_array[c-1]);
    double r_abs = abs(r);
    x += pow(r_abs, p);
  }
  return (x);
}

//MAV

```

```

double MAV(int MAV_array[], int n)
{
  double x = 0;
  double p = 0;
  for(int i=1;i<=n;i++)
  {
    int c = i - 1;
    x += abs(MAV_array[c]);
  }
  return (x/n);
}

//WL
double WL(int WL_array[], int n)
{
  double x = 0;
  double p = 0;
  for(int i=2;i<=n;i++)
  {
    int c = i - 1;
    double r = (WL_array[c] - WL_array[c-1]);
    double r_abs = abs(r);
    x += r_abs;
  }
  return (x);
}

void setup() {
  // Serial port for debugging purposes
  Serial.begin(115200);

  pinMode(EMGpin, INPUT);

  // Setting the ESP as an access point
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  //WiFi.softAP(ssid, password);

  delay(1000);

  Serial.println("Connected to wifi");
  printWifiStatus();

  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  //-----Home_Page-----

```

```

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/html", HomePage);
});
//-----Sensor EMG-----
server.on("/sinalEMG", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", readEMG().c_str());
});
//-----Página Gráfico-----
server.on("/EMG_Graph", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/html", SensorRead, processor);
});
// Start server
server.begin();
}
void loop() {
  for(cont=0;cont<ArrayLength;cont++)
  {
    sinalEMG = analogRead(EMGpin);

    Array[cont] = sinalEMG.toInt();

    delay(1);
  }

  //Variáveis das características
  double EMAV_Result = EMAV(Array, ArrayLength);
  double MAV_Result = MAV(Array, ArrayLength);
  double EWL_Result = EWL(Array, ArrayLength);
  double WL_Result = WL(Array, ArrayLength);

  //PRINT 4 Características
  Serial.print(sinalEMG.toInt());
  Serial.print(" ");
  Serial.print(EMAV_Result);
  Serial.print(" ");
  Serial.print(MAV_Result);
  Serial.print(" ");
  Serial.print(EWL_Result);
  Serial.print(" ");
  Serial.print(WL_Result);

  Serial.print("\n");
  //delay(500);
}

```