

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTOS ACADÊMICOS DE ELETRÔNICA E MECÂNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

ELUANA SIEVERT CARDOSO
MATHEUS HERRERA BEZERRA

**INTEGRAÇÃO DE CONTROLE ELETRÔNICO ATRAVÉS DE
COMANDOS DE VOZ APLICADO EM PRÓTESES MECÂNICAS.**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2018

ELUANA SIEVERT CARDOSO
MATHEUS HERRERA BEZERRA

**INTEGRAÇÃO DE CONTROLE ELETRÔNICO ATRAVÉS DE
COMANDOS DE VOZ APLICADO EM PRÓTESES MECÂNICAS.**

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Rubens Alexandre de Faria

Coorientadora: Prof. Dra. Mariana Antonia Aguiar Furucho

CURITIBA
2018

TERMO DE APROVAÇÃO

ELUANA SIEVERT CARDOSO
MATHEUS HERRERA BEZERRA

INTEGRAÇÃO DE CONTROLE ELETRÔNICO ATRAVÉS DE COMANDOS DE VOZ APLICADO EM PRÓTESES MECÂNICAS.

Este trabalho de conclusão de curso foi apresentado no dia 13 de junho de 2018, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Milton Luiz Polli
Coordenador de Curso
Departamento Acadêmico de Mecânica

Prof. M.Sc. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Simone Crocetti
UTFPR

Ph.D. Gilson Yukio Sato
UTFPR

Marcelo Maldaner
UTFPR

D. Sc. Rubens Alexandre de Faria
Orientador - UTFPR

M. Sc. Mariana Antonia Aguiar Furucho
Co-orientadora - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

“Mas em todas estas coisas somos mais que vencedores,
por meio daquele que nos amou.”

Romanos 8:37

AGRADECIMENTOS

Não é possível expressar em poucos parágrafos todos os nomes de nossos familiares, amigos, apoiadores e professores que nos incentivaram a chegar até aqui. Portanto, desde já, nos desculpamos a estes que não estão presentes aqui em palavras, mas certamente estão em nossos corações e pensamentos, gratidão a vocês.

Dedicamos este trabalho primeiramente a Deus, por ser essencial em nossas vidas, autor dos nossos destinos, Pai e socorro presente na hora da angústia.

Reverenciamos nosso Professor Orientador, Rubens Alexandre de Faria, que mesmo estando em outro país, nos forneceu todo suporte e apoio necessário para chegarmos aqui hoje. De mesmo modo, nossa Professora Co Orientadora, Mariana Antonia Aguiar Furucho, por todo carinho, paciência e sabedoria expressos em todos os momentos que compartilhamos ao longo desta jornada. Também ao Prof. David Kretschek, do DAMEC, e coordenador do Grupo de Impressão 3D (GIP3D) da UTFPR que nos doou a prótese utilizada para a realização deste trabalho, sem ele nós não teríamos concluído este projeto. Por meio deles, nos reportamos a toda à comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) por ter nos dado à chance e todas as ferramentas que nos permitiram chegar ao final desse ciclo de maneira satisfatória.

Aos nossos pais e irmãos, Valter Cardoso, Ester Sievert Cardoso, Isabela Sievert Cardoso, Luiz Vieira Bezerra, Dejair Aparecida Herrera Paim e Marcell Herrera Bezerra, por todo apoio, incentivo. Vocês se sacrificaram, se dedicaram, abdicaram de tempo e de muitos projetos pessoais para que nós tivéssemos a oportunidade de estudar e de ter uma boa formação, não somente profissional, mas também pessoal. Devemos tudo o que somos a vocês, e se sentimos orgulho de nós hoje e do lugar onde chegamos, é porque sabemos que vocês vieram segurando a nossa mão todos esses anos.

Aos nossos cônjuges, Andreas Lickfeld, Marinês Dias Muller e filha Raquel Muller, por toda paciência, compreensão, amor e apoio durante este período tão conturbado de nossas vidas.

Aos nossos amigos-irmãos, sempre presentes na hora do sufoco: Vanessa Romankiv, Kelly Siqueira, Fernanda Werka, Eneile Guimarães, Miriam Reddin, Camila e Felipe de Goes Silva, Luisa Silva, Alexandre Breda, Valério Marochi, Estela Utzig de Moura, Karin Arruda, Ian Rodrigo Cavalli, Andrey Nunes, por todo apoio, incentivo e companhia ao longo da nossa jornada.

E a todos os outros que nos acompanharam, incentivaram com palavras e gestos, vocês com certeza fazem parte desta vitória.

“O pensamento é uma corrente elétrica, uma tempestade se espalhando pelo cérebro. Só isso. É um relâmpago em milivolts, em milissegundos. Tudo o que fazemos vem de uma tempestade elétrica na nossa cabeça. Uma tempestade tão complexa e tão difícil de prever como as que acontecem no céu. Mas move tudo o que nós fazemos, como sonhar, imaginar, pensar, prever, andar, falar, correr, tudo.

Essa é a raiz de toda a humanidade, centenas de bilhões de elementos disparando pequenas descargas elétricas que geram um campo magnético muito pequeno, mas mesmo assim poderoso o suficiente para gerar tudo que a história da humanidade já gerou.

E o sonho é uma atividade do sistema nervoso que, pelo que se debate muito hoje em dia e pelas teorias mais modernas, responde pela recapitulação do que aconteceu recentemente com tudo o que foi acumulado ao longo da vida e gera reverberações elétricas quando dormimos. Ele tem várias funções. Se considerarmos pesquisas mais recentes, nada mais é do que a consolidação das nossas memórias.”

(NICOLELIS, Miguel, 2009.)

RESUMO

CARDOSO, Eluana Sievert, BEZERRA, Matheus Herrera. **Integração de controle eletrônico através de comandos de voz aplicado em próteses mecânicas**. 2018. 71 páginas. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A população de deficientes físicos brasileiros é majoritariamente composta por pessoas de baixa renda, as quais não possuem condições financeiras de adquirir uma prótese funcional de qualidade (IBGE, 2010). Notou-se uma oportunidade para este trabalho: através da manufatura aditiva e da eletrônica, é possível realizar a produção de próteses impressas em 3D, com eletrônica embarcada, o que oferece ao usuário maior funcionalidade e liberdade de uso. Este trabalho apresenta o estudo e desenvolvimento de um firmware controlado por comandos de voz, integrado a uma prótese mecânica, criada através de manufatura aditiva. Também será discorrido sobre a manufatura aditiva, Movimento *Maker* e projetos de próteses *open source*. Tendo, então, estas tecnologias unidas, constata-se que o custo agregado é acessível, de acordo com a problemática apresentada, visto que as classes sociais menos favorecidas podem ser finalmente atingidas, propiciando a estes uma nova oportunidade de liberdade e inclusão na sociedade.

Palavras chave: Prótese. Manufatura aditiva. Sistemas embarcados. Reconhecimento de voz. Movimento *Maker*.

ABSTRACT

CARDOSO, Eluana Sievert, BEZERRA, Matheus Herrera. **Integration of electronic control through voice recognition applied to mechanical prostheses**. 2018. 71 páginas. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The Brazilian disabled population is mostly composed of low-income people, who do not have the financial conditions to acquire a quality functional prosthesis (IBGE, 2010). There was an opportunity for this work: through the additive manufacture and electronics, it is possible to realize the production of 3D printed prostheses with embedded electronics, which offers the user greater functionality and freedom of use. This work presents the study and development of a firmware controlled by voice commands, integrated with a mechanical prosthesis, created through additive manufacturing. Also will be discussed on the additive manufacture, maker movement and open source prosthesis projects. Having these technologies together, it can be seen that the aggregate cost is accessible, according to the problem presented, since the less favored social classes can finally be reached, giving them a new opportunity for freedom and inclusion in society.

Keywords: Prosthesis. Additive Manufacturing. Embedded systems. Voice recognition. Maker Movement.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura óssea da mão humana.	21
Figura 2 - Tipos de preensão definidos por Kapandji.	22
Figura 3 - Preensão bidigital ou tipo pinça.	23
Figura 4 - Preensão pluridigital.....	23
Figura 5 - Preensão digital palmar.	24
Figura 6 - Preensão palmar.....	24
Figura 7 - Preensão centrada.....	24
Figura 8 - Próteses com pés em posição fixa, de pau ou aço.	25
Figura 9 – Prótese de madeira.	26
Figura 10 - Prótese de aço.	26
Figura 11 - Próteses de cobre e silicone.	27
Figura 12 - Braço com articulação de punho funcional.	40
Figura 13 - Prótese modelo Phoenix Hand.	40
Figura 14 - Prótese modelo Raptor Reloaded.	40
Figura 15 - Prótese modelo Osprey Hand.....	41
Figura 16 - Prótese modelo Talon.....	41
Figura 17 - Prótese modelo Cyborg Beast.	41
Figura 18 - Prótese modelo Ody Hand.....	42
Figura 19 - Prótese modelo Flexy Hand.....	42
Figura 20 - Braço com articulação de cotovelo funcional.	42
Figura 21 - Prótese modelo Team Unlimbited Arm.	43
Figura 22 - Prótese modelo RIT Arm.....	43
Figura 23 - Braço com desarticulação de cotovelo e acima.	44
Figura 24 - Prótese modelo Hackberry da EXIII.	44
Figura 25 - Arduino Nano.	46
Figura 26 - Pinos do Arduino Nano.	47
Figura 27 - Módulo de reconhecimento de voz F 0475.	48
Figura 28 - Módulo Relé 5V 4 Canais.	48
Figura 29 - Micro Servo Motor 9g SG90 Tower Pro 180º.....	49
Figura 30 - Bateria MOX 9 Volts 320mAh modelo MO-9V320.	50
Figura 31 - Diagrama de ligação do controle da prótese.....	52
Figura 32 – Projeto eletrônico fixado na placa de acrílico (sem as baterias).	52
Figura 33 – Projeto eletrônico com fiação na placa de acrílico (sem as baterias).	53
Figura 34 – Módulo de controle fixado ao protótipo.	54
Figura 35 – Base em MDF.	54
Figura 36 – Prótese com módulo de controle montado sobre base em MDF.....	55
Figura 37 - Biblioteca vr_sample_train.	57
Figura 38 - Gravando comandos no módulo de voz.....	58
Figura 39 - Comando ABRE.....	61
Figura 40 - Comando FECHA.	62
Figura 41 - Comando PINÇA.....	63
Figura 42 - Comando PALMAR.....	64

LISTA DE TABELAS

Tabela 1 - Conexão dos pinos do Arduino ao módulo de voz.	50
Tabela 2 - Conexão do arduino ao módulo de relés.....	51
Tabela 3 - Orçamento	65

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	O TEMA.....	12
1.2	DELIMITAÇÕES DO ESTUDO.....	13
1.3	PROBLEMA.....	14
1.4	OBJETIVOS.....	15
1.4.1	OBJETIVO GERAL.....	15
1.4.2	OBJETIVOS ESPECÍFICOS.....	15
1.5	JUSTIFICATIVA.....	16
1.6	PROCEDIMENTOS METODOLÓGICOS.....	17
1.7	ESTRUTURA DO TRABALHO.....	19
2	FUNDAMENTAÇÃO TEÓRICA.....	21
2.1	A MÃO HUMANA.....	21
2.1.1	TIPOS DE PEGADA.....	22
2.2	PRÓTESES NO BRASIL E SUA HISTÓRIA PELO MUNDO.....	25
2.3	PESQUISA <i>ONLINE</i> EM COMUNIDADE DE DEFICIENTES FÍSICOS.....	28
2.4	MOVIMENTO MAKER.....	30
2.5	E-NABLE.....	31
2.6	FAB LAB.....	31
2.6	MAO3D - PROGRAMA SOCIAL E COLABORATIVO DA UNIFESP.....	32
2.7	A PROTETIZAÇÃO INFANTIL.....	33
2.8	PRÓTESES AVANÇADAS NO MERCADO.....	34
2.9	ACESSIBILIDADE PARA PESSOAS DE BAIXA RENDA.....	35
2.10	PRÓTESES ATUAIS E DIFERENÇAS NOS VALORES.....	36
2.11	NASCIMENTO DA IMPRESSÃO 3D.....	36
2.12	COMO FUNCIONA A IMPRESSÃO 3D.....	37
2.13	MATERIAIS DA IMPRESSÃO 3D.....	37
2.14	VANTAGENS E DESVANTAGENS DA MANUFATURA ADITIVA.....	38
2.15	MODELOS DE PRÓTESES DE CÓDIGO ABERTO.....	39
2.15.1	MODELOS PARA DESARTICULAÇÃO DE DEDOS E PARCIAL DE PALMA.....	39
2.15.2	MODELOS PARA DESARTICULAÇÃO DE PUNHO E PARCIAL DE ANTEBRAÇO.....	42

2.15.3	MODELOS PARA DESARTICULAÇÃO DE COTOVELO E PARCIAL DE BRAÇO	43
2.15.4	PRÓTESE RAPTOR RELOADED.....	44
3	MÓDULOS PARA PROJETO ELETRÔNICO DE CONTROLE DE MOVIMENTAÇÃO.....	46
3.1	MICROCONTROLADOR ARDUINO NANO	46
3.2	MÓDULO DE RECONHECIMENTO DE VOZ	47
3.3	MÓDULO DE RELÉS	48
3.4	SERVO MOTORES	49
3.5	BATERIAS	49
3.6	PROJETO ELETRÔNICO.....	50
3.7	INTEGRAÇÃO DO PROJETO ELETRÔNICO NO PROTÓTIPO	53
4	PROGRAMAÇÃO DO MÓDULO DE CONTROLE.....	56
4.1	CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ	56
4.2	CÓDIGO PARA CONTROLE DOS SERVOMOTORES	59
5	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	61
5.1	TESTES DO PROTÓTIPO INTEGRADO A ELETRÔNICA	61
5.1.1	COMANDO ABRE E FECHA	61
5.1.2	COMANDO PINÇA.....	63
5.1.3	COMANDO PALMAR.....	64
5.1.4	COMANDO DESLIGAR	64
5.1.5	COMANDO TESTANDO	64
5.2	ORÇAMENTO	65
6	CONSIDERAÇÕES FINAIS	66
	REFERÊNCIAS.....	68
	ANEXO A – ARQUIVOS STL DA PRÓTESE RAPTOR RELOADED	73
	ANEXO B – CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ “VR_SAMPLE_TRAIN”.....	75
	ANEXO C – CÓDIGO PARA CONTROLE DOS SERVOS-MOTORES	94

1 INTRODUÇÃO

A história apresenta relatos de fósseis datados a partir de 2.300 a.C. que já apresentavam próteses improvisadas feitas, por exemplo, com o pé de uma cabra com o encaixe sendo a própria pele dissecada do animal, em pernas e braços humanos (CARVALHO, 2003).

Após milhares de anos, o avanço tecnológico trouxe para a realidade de algumas pessoas as próteses biônicas. Passando de uma prótese rudimentar para as de fibras de carbono com ligas de alumínio, foi alcançado o ápice desta tecnologia desenvolvendo sensores capazes de proporcionar ao usuário percepções táteis (SANCTUARY, 2014).

Muitas pessoas amputadas, ou que sofreram avulsão de membros, desconhecem ainda hoje as novas tecnologias na área de próteses biônicas e, se conhecem, não possuem meios financeiros de adquirir uma prótese de avançada tecnologia, segundo pesquisa realizada pelos autores (CARDOSO; BEZERRA, 2015).

Há muita pesquisa a ser realizada nesta área, como testar novos tipos de materiais biocompatíveis ou sensores neurais que forneçam sensações cada vez mais reais e com motores de alta capacidade de torque e baixo peso. Um exemplo de pesquisa é a apresentada por Halpern (2015), na qual, próteses experimentais feitas com materiais mais leves possibilitam a movimentação quase natural, simulando a dinâmica de articulações, com controle em tempo real, através de microprocessadores e sensores e fabricadas em impressoras 3D. Bem como estas e outras tecnologias, tais como a manufatura aditiva, controle de motores elétricos, comandos de voz e outras tecnologias serão apresentadas ao longo deste trabalho.

1.1 O TEMA

Segundo um estudo realizado por Kunkel (2016a), coordenadora do grupo de pesquisa do projeto Mao3D, através da lei nº 8.080 de 19 de Setembro de 1990 (COLLOR, 1990), o SUS (Sistema Único de Saúde) deveria fornecer uma prótese e um programa de reabilitação a cada indivíduo que precisar, porém o atendimento atual não contempla todos os necessitados. Dentre os motivos que impedem o fornecimento adequado de prótese de membro superior, o principal deles é que as

marcas nacionais são caras, encontradas a partir de R\$ 100.000,00, pesadas e pouco funcionais e os modelos importados, que possuem maior funcionalidade, chegam a custar R\$ 300.000,00 (KUNKEL, 2016).

Abre-se espaço, assim, para uma nova tecnologia: a manufatura aditiva, mais conhecida como impressão 3D, que oferece uma alternativa para se obter próteses de baixo custo. Porém, para se conseguir imprimir uma prótese deste tipo, deve-se efetuar um projeto de engenharia do modelo a ser construído. Contudo, para que o próprio usuário possa produzir sua prótese, ele deverá possuir um conhecimento de nível técnico básico-intermediário nas áreas de mecânica e eletrônica, além de desenvolver um modelo do objeto dentro de um aplicativo de modelagem 3D. Este modelo deve ser enviado para a impressora, através de um *software* que irá “fatiá-lo” em camadas e em duas dimensões, a partir do qual, se iniciará a impressão. Todo o processo de desenvolvimento e impressão é demorado, levando no mínimo 30 dias e exige conhecimentos técnicos que, geralmente, não são acessíveis a maioria das pessoas.

Além da tecnologia de impressão 3D, há necessidade de realizar atividades de maneira mais prática e inteligente. Primeiro os computadores dotados de comando de voz permitiam escrever sem utilizar os dedos, depois, *smartphones* e até mesmo alto-falantes, que realizam tarefas apenas ouvindo o seu comando de voz. Com o conhecimento atual e o grande crescimento do *Machine Learning*, que é a programação de computadores para aperfeiçoar um desempenho usando dados de exemplo programados previamente ou adquiridos através de experiências passadas (ALPAYDIN, 2010), esta tecnologia é notada como uma forma de oferecer maior liberdade ao usuário final, como por exemplo, os de próteses biônicas.

1.2 DELIMITAÇÕES DO ESTUDO

Este trabalho trata dos assuntos relacionados à integração da eletrônica através do acionamento por comandos de voz em próteses mecânicas. O desafio proposto inicialmente, de controlar a prótese via comandos mioelétricos, era complexo, visto a necessidade de importação de alguns itens, especialmente do sensor mioelétrico (sensor que capta sinais provenientes da contração muscular (ORTOLAN, 2002)) e também da dificuldade da captação e filtragem de sinais do mesmo. Após esta experiência, foi decidido então realizar todo o estudo de integração

de um controle eletrônico através de comandos de voz em próteses mecânicas, sendo esta, produzida através da manufatura aditiva, já existente e cedida pelo Departamento Acadêmico de Mecânica da UTFPR.

1.3 PROBLEMA

Segundo o Censo/IBGE 2010, dos 190,7 milhões de brasileiros, 13,3 milhões destes possuem algum tipo de deficiência motora (7% da população do país) (IBGE, 2010). Em relação às deficiências motoras, podem ser classificadas em três grupos: 'Deficiência motora – não consegue de modo algum (734.421 pessoas)'; 'Deficiência motora – grande dificuldade (3.698.929 pessoas)'; 'Deficiência motora – alguma dificuldade (8.832.249 pessoas) (IBGE, 2010). Dentro deste grupo, existem pessoas que sofreram algum tipo de amputação devido a acidente, doença ou outros fatores (IBGE, 2010).

Neste mesmo censo, foi apresentado o rendimento salarial de deficientes motores, no qual é indicado que 82,7% destes deficientes recebem de 0 a 2 salários mínimos (IBGE, 2010). Em 2010, o salário mínimo era de R\$ 510,00, nota-se então que uma pessoa com deficiência motora que trabalhasse e recebesse dois salários mínimos não teria condições de adquirir qualquer prótese biônica oferecida no mercado, mesmo sendo as de mais baixo valor que custam, no mínimo, R\$ 100.000,00. O salário mínimo brasileiro em 2018 é de R\$ 954,00 (DIEESE, 2018), supondo que o deficiente que foi entrevistado pelo IBGE continue trabalhando hoje, mesmo assim, seria inviável para ele adquirir uma prótese.

As próteses biônicas oferecidas no mercado brasileiro ultrapassam o valor de R\$100.000,00 (LOPES, 2015). A manufatura aditiva entra neste caso como uma alternativa. O desenvolvimento de próteses utilizando este método tem alcançado níveis exponenciais com o passar dos anos, com projetos *open sources* e disponibilização *online* de dúvidas de desenvolvedores e projetistas amadores, o termo "*Do it yourself*" (Faça você mesmo) se torna real, dando a oportunidade de pessoas com algum conhecimento técnico, na área de mecânica e eletrônica, produzirem próteses de baixo custo.

1.4 OBJETIVOS

O objetivo deste trabalho, visando alcançar deficientes motores de membros superiores com amputações ou deficiência nata na área de mão e punho, porém com articulação do punho existente e ainda, com situação financeira de baixa renda e sem acesso às próteses que deveriam ser fornecidas pelo governo através do SUS, é oferecer uma alternativa viável economicamente a esses possíveis usuários. Será fornecido, um estudo para o auxílio na integração de uma prótese produzida utilizando a manufatura aditiva e da eletrônica através de comandos de voz, visando à utilização de produtos biodegradáveis, como o PLA (Poliácido Láctico), o que agregará redução no peso e valor do produto final.

1.4.1 OBJETIVO GERAL

Realizar um estudo e projeto de integração de um controle eletrônico, para a movimentação de um protótipo de prótese mecânica de custo acessível (se comparado ao valor de próteses comerciais com funções parecidas ou idênticas), através de comandos de voz.

1.4.2 OBJETIVOS ESPECÍFICOS

- Realizar um estudo sobre a mão humana e seus tipos de pegada para auxílio na implementação de movimentos em uma prótese.
- Identificar a necessidade de desenvolver próteses de custo acessível para portadores de amputação de palma, parcial de palma ou que possuam a articulação do punho preservada.
- Levantar os benefícios e desvantagens do uso da manufatura aditiva para a redução de custo e facilidade de personalização na fabricação de próteses.
- Apresentar os principais modelos de próteses mecânicas disponíveis no mercado para a produção através da manufatura aditiva.
- Confeccionar um protótipo eletrônico, utilizando Arduino e *Shields* (módulos auxiliares de interface), para controle de movimentação de próteses mecânicas através de comandos de voz.

- Apresentar uma solução de integração do projeto desenvolvido no protótipo de uma prótese, já impresso no modelo Raptor Reloaded.
- Apresentar os custos para a produção da prótese no modelo Raptor Reloaded e os custos do projeto de eletrônica para o controle de movimentação do protótipo da mesma.
- Executar os testes na plataforma do protótipo.

1.5 JUSTIFICATIVA

Através de uma pesquisa realizada em uma comunidade *online* de deficientes motores (CARDOSO; BEZERRA, 2015), 50% dos participantes declararam que se sentem excluídos e inválidos para a sociedade em que vivem, por não conseguirem executar algumas tarefas consideradas simples para pessoas sem as limitações deles. Segurar uma garrafa de água, cozinhar, escrever, digitar no seu celular ou *notebook* de forma rápida, tornaram-se tarefas praticamente impossíveis para algumas destas pessoas com deficiência. Até mesmo a vergonha de mostrar em público a ausência de parte ou completa de membro se tornou motivo para simplesmente não se apresentar em público e se ausentar do convívio social.

A utilização de prótese possibilita a reinserção no mercado de trabalho, escola, faculdade e até mesmo no meio social, devolvendo parcialmente a função motora para aqueles que possuem deficiência nata ou adquirida. Tendo como conhecimento a realidade do deficiente físico brasileiro, o desenvolvimento de uma prótese de baixo custo e alta funcionalidade se deu como uma necessidade para este público.

Analisando projetos independentes de próteses biônicas, verifica-se a possibilidade de integrar as diversas áreas do conhecimento explanadas ao longo do curso de mecatrônica. Mecânica, elétrica, eletrônica e programação se unem em um único propósito, desenvolver um protótipo de uma prótese funcional, que sirva de modelo para futuros estudos nesta mesma área, em prol de pessoas com deficiência motora dos membros superiores.

Entra então a possibilidade de cooperar com o desenvolvimento de novas tecnologias em próteses para professores, alunos, desenvolvedores e também para deficientes físicos, para os quais é apresentada uma oferta de possibilidade de um novo estilo de vida, com mais conforto e independência, a baixo custo, visto que o custo para o desenvolvimento de uma prótese impressa em 3D com eletrônica

integrada, segundo orçamentos feitos em empresas de impressão e prototipagem na região de Curitiba – PR, somando os valores dos componentes eletrônicos, os valores iniciais se aproximam em R\$ 1.000,00, que é apenas 1% do valor inicial de uma prótese biônica do mercado brasileiro (LOPES, 2015).

Contudo, é importante realizar a análise do material utilizado na impressão 3D, tendo como foco a biocompatibilidade do material impresso e do usuário, visto que comumente são utilizados os materiais ABS não cirúrgico (Acrilonitrila Butadieno Estireno) e PLA (Poliácido Láctico), havendo alternativas para a impressão de próteses, como os materiais PC ISO (Policarbonato - ISO) e o ABS-M30i (ABS termoplástico disponibilizado pela empresa Stratasys), os quais são biocompatíveis e esterilizáveis (MASCARENHAS, 2016).

Esta análise não está sendo considerada para a execução deste trabalho, visto que o objetivo é apresentar um protótipo o qual deve ser considerado como uma proposta e possível solução para a problemática explanada.

1.6 PROCEDIMENTOS METODOLÓGICOS

Visando identificar a necessidade de produzir próteses de baixo custo, foi desenvolvido um estudo sobre a mão humana, tipos de articulações e formas de utilização da mesma, para um maior entendimento da anatomia e movimentos a serem implementados no protótipo desenvolvido. Além disso, foi realizada uma pesquisa bibliográfica, baseada em dados estatísticos junto a órgãos governamentais e organizações não governamentais para quantificar a população de pessoas portadoras de deficiência motora, especificamente aqueles que sofreram amputação, ou avulsão de membro superior. Da mesma forma, dados socioeconômicos deste público alvo também foram explorados.

Em seguida foi realizado e apresentado um estudo sobre a manufatura aditiva, explanando o que é e como vem sendo difundida no Brasil e no mundo, bem como os benefícios e desvantagens do uso da manufatura aditiva para a redução de custo e facilidade de personalização na fabricação de próteses. Também foram apresentados também alguns modelos de próteses que podem ser produzidos através deste processo, das quais os projetos e arquivos são disponibilizados gratuitamente através

da internet em sites especializados em compartilhar esse tipo de conhecimento de forma legalizada.

Após a apresentação destas próteses, foi realizado um projeto eletrônico (que inclui: servo-motores, microcontrolador Arduino, módulo de relés, módulo de reconhecimento de voz, bateria) e integração do mesmo, devidamente testado em um protótipo de prótese já confeccionado em PLA, no modelo Raptor Reloaded da fabricante e-Nable. Este protótipo apresentado neste trabalho atende os casos específicos de amputação de dedos e parcial de palma, sendo necessária que a articulação de punho esteja presente e seja funcional, essa amputação é conhecida como transradial.

O protótipo utilizado neste trabalho foi cedido pelo Prof. David Kretscheck, do Departamento Acadêmico de Mecânica (DAMEC) e coordenador do Grupo de Impressão 3D (GIP3D) da UTFPR. Neste protótipo, ao flexionar a articulação do punho, os fios que são ligados das pontas dos dedos da prótese, até a peça que é fixada ao antebraço, são esticados, fechando simultaneamente todos os dedos. Este acionamento é muito similar na maioria das próteses mecânicas.

Devido à falta de variedade de movimentos neste modelo de prótese, foi implantado um circuito eletrônico que comandam movimentos mais elaborados para a mesma através de comandos de voz. Os fios já presentes na prótese são acoplados à parte móvel dos motores, dando movimentos independentes aos dedos. Os motores são acionados por um módulo de relés, para não sobrecarregar as saídas do microcontrolador Arduino Nano. Este foi programado para receber comandos da plataforma de reconhecimento de voz, capaz de acionar os motores através do módulo de relés. A plataforma de reconhecimento de voz, que já conta com um microfone, foi acoplada também ao microcontrolador e todo este conjunto é alimentado por três baterias de 9V. O conjunto foi integrado ao protótipo da prótese, que posteriormente poderá ser fixado ao antebraço de um usuário, entretanto este conjunto final foi fixado a uma plataforma de MDF a qual possui a modelagem de um antebraço que se encaixa perfeitamente ao protótipo da prótese, podendo assim ser facilmente demonstrado todo seu funcionamento em bancada.

O funcionamento do conjunto foi testado da seguinte maneira: os programas são testados através de simulador próprio para encontrar e corrigir possíveis erros de código. Depois de corrigidos, foi inserido o código no interior da memória de programa

do microcontrolador. O teste do funcionamento dos motores foi executado através de comandos enviados pela interface do programa.

Já o módulo de reconhecimento de voz foi testado da seguinte forma: os vocábulos de um locutor (um dos pesquisadores deste trabalho) foram gravados no mesmo e cadastrados no interior do código. O acionamento dos servos-motores foi realizado através do reconhecimento dos comandos cadastrados. Estes movimentam um ou mais servo-motores, executando o movimento desejado, referente àquele vocábulo mencionado. Uma vez integradas às partes eletrônica e mecânica, cada vocábulo dito pelo locutor gera um movimento específico no protótipo da prótese, concluindo assim os testes de integração.

Foram disponibilizados os arquivos em formato STL (Stereolithography), fornecidos pelo desenvolvedor da prótese (E-nable), além do projeto eletrônico e códigos desenvolvidos pelos autores deste trabalho.

Detalhes como tempo estimado de produção, quantidade de material PLA e componentes utilizados para a fabricação de uma prótese em tamanho real, no modelo Raptor Reloaded da E-Nable, foram obtidos através de orçamentos realizados com escritórios de impressão 3D. Além disso, foram disponibilizados os custos para a realização do projeto eletrônico aplicado a um protótipo da mesma prótese, o qual foi desenvolvido neste trabalho.

1.7 ESTRUTURA DO TRABALHO

O trabalho possui a estrutura apresentada a seguir.

Capítulo 1 - Introdução: É apresentado o tema deste trabalho, a delimitação de estudo encontrada para a conclusão do mesmo, o problema que motivou a execução do mesmo, os objetivos específicos e geral, além da justificativa e procedimentos metodológicos aplicados.

Capítulo 2 – Fundamentação Teórica: São apresentados tópicos que explanam a mão humana, tipos de pegada e preensões. Além de um estudo sobre as próteses no Brasil, uma pesquisa realizada pelos autores deste trabalho com uma comunidade de deficientes físicos brasileiros, Movimento *Maker*, empresas que realizam projetos de próteses e protótipos no formato *open source*, impressão 3D e modelos de próteses fabricadas através deste tipo de manufatura.

Capítulo 3 – Módulos Para Projeto Eletrônico De Controle De Movimentação: São apresentados todos os módulos utilizados para o correto funcionamento do protótipo proposto para este trabalho, além de detalhes sobre o projeto eletrônico, a integração do mesmo no protótipo da prótese escolhida e como os testes foram executados.

Capítulo 4 – Códigos: São apresentados os códigos desenvolvidos pelos autores para o funcionamento e integração dos módulos entre si.

Capítulo 5 – Apresentação e análise dos resultados: São apresentados todos os resultados após a conclusão do projeto e uma breve análise sobre os mesmos além do orçamento do projeto.

Capítulo 6 – Considerações finais: São apresentadas as conclusões do projeto, comparação entre os resultados esperados e encontrados e previsões para próximos estudos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 A MÃO HUMANA

A mão é um membro próximo ao antebraço, que permite movimentos de empunhadura, ou preensões. Ela é formada por três conjuntos ósseos: carpo, metacarpo e falanges. Também é composta pelo punho, que a proporciona movimentos, pela palma que fornece uma superfície de aderência e também pelo dorso das mãos e dedos que possibilita que os movimentos sejam completos e precisos. A mão humana é suprida por terminações sensoriais, estas são responsáveis por nos fornecerem os sentidos, como dor, sensibilidade a temperaturas e é claro, o toque (MOORE et al.,2014). Uma mão humana possui 14 falanges: duas no polegar e outras 12 distribuídas, conforme apresentado na Figura 1 (LIMA,2015).

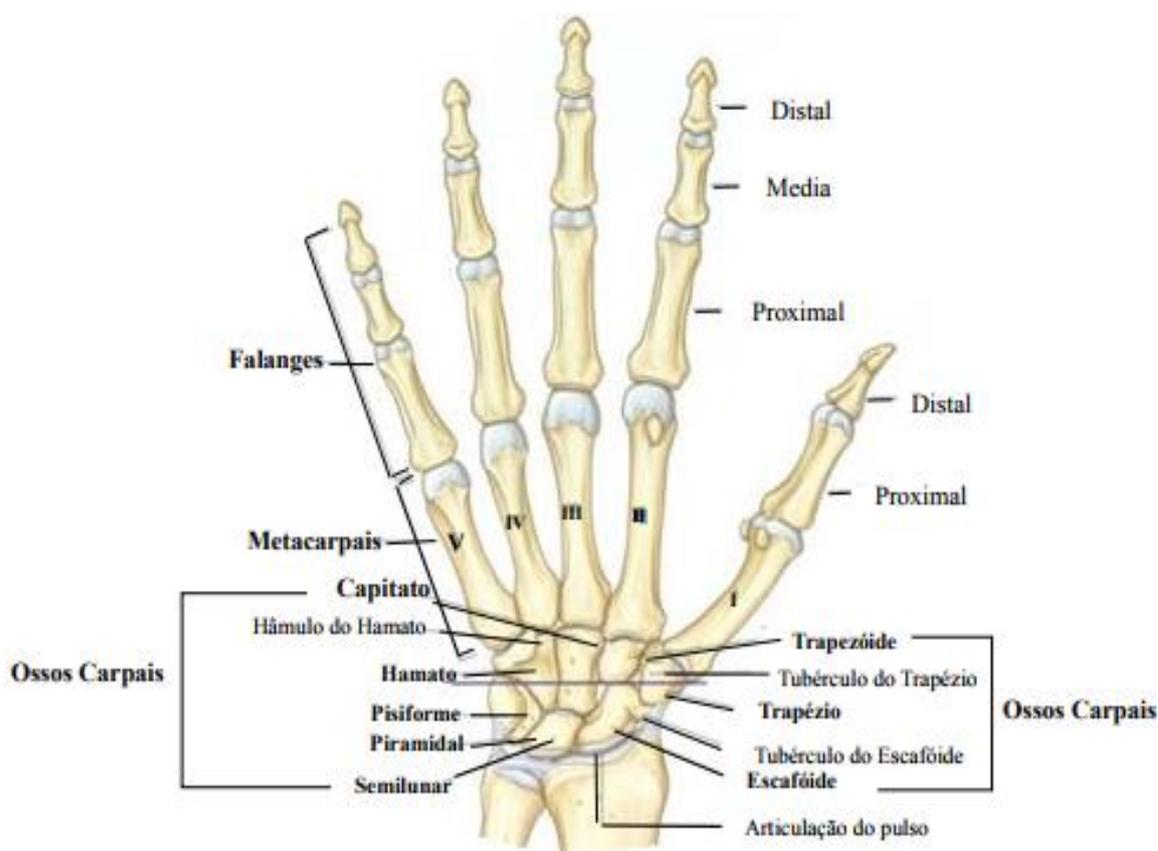


Figura 1 - Estrutura óssea da mão humana.

Fonte: Moore et al. 2014, p. 709.

2.1.1 TIPOS DE PEGADA

A aplicação da força muscular aliada a um tipo de preensão gera o desempenho de uma atividade física ou manipulação de objetos. Esta ação é conhecida como pegada (PASCHOARELLI, LC.; MENEZES, MS.,2009).

Em 1956, Napier tornou-se uma das primeiras referências nesse campo. Ele estudou os movimentos da mão e os classificou em dois grupos: preensões de força (preensões palmares) e preensões de precisão (preensões digitais). A.I. Kapandji (2002) foi o primeiro autor a ampliar a gama das preensões: preensões puras (palmares, digitais e centradas), preensões com peso (auxiliadas pela gravidade) e preensões-ações (associadas a movimentos), conforme na Figura 2 (PASCHOARELLI, LC.; MENEZES, MS.,2009).

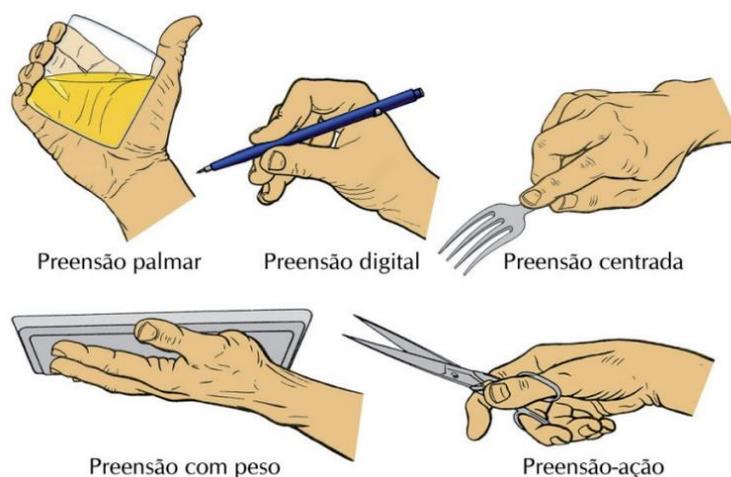


Figura 2 - Tipos de preensão definidos por Kapandji.

Fonte: (PASCHOARELLI, LC.; MENEZES, MS.,2009).

2.1.1.1 PREENSÕES DIGITAIS

Preensões digitais são classificadas como preensões de precisão e se divide em outros três tipos de preensões: Bidigitais, pluridigitais e tridigital (KAPANDJI, 2002).

- **Bidigitais:** Nesta preensão os dedos polegar e indicador são unidos com o objetivo de realizar uma pegada em forma de pinça, conforme a Figura 3.

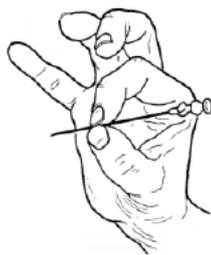


Figura 3 - Preensão bidigital ou tipo pinça.

Fonte: Kapandji (2002).

- **Tri ou Pluridigitais:** É um modo de preensão que exerce maior força sob o objeto que sofre a pegada. Ela é realizada com a combinação do polegar e outros dois ou mais dedos, conforme a Figura 4.

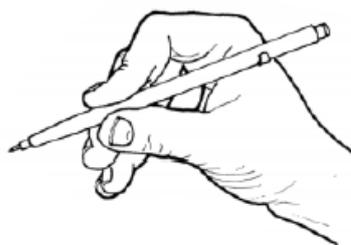


Figura 4 - Preensão pluridigital.

Fonte: Kapandji (2002).

2.1.1.2 PREENSÕES PALMARES

Preensões palmares são classificadas como preensões de força e se dividem em outros dois tipos de preensões: digital palmar e palmar (KAPANDJI, 2002).

- **Digital palmar:** É quando a força da palma é oposta em conjunto com os últimos quatro dedos, conforme apresentado na Figura 5. Esta preensão pode ser observada quando se está dirigindo, segurando o volante.



Figura 5 - Preensão digital palmar.

Fonte: Kapandji (2002).

- **Palmar:** É quando ocorre a união de todos os dedos, realizando um movimento como o de “mão fechada”, conforme apresentado na Figura 6. Por exercer grande força, esta preensão é utilizada quando os objetos a serem pegos são pesados ou grandes.



Figura 6 - Preensão palmar.

Fonte: Kapandji (2002).

2.1.1.3 PREENSÕES CENTRADAS

Estas preensões são classificadas como preensões de força. É o ato de prolongar a mão com extrapolação do indicador e flexão dos três últimos dedos. Pode ser observada ao manusear uma chave de fenda, conforme apresentado na Figura 7 (KAPANDJI, 2002).

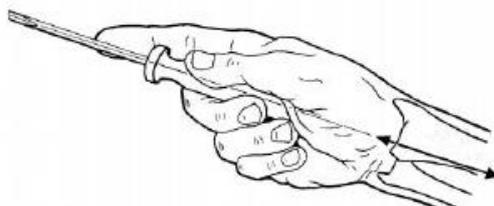


Figura 7 - Preensão centrada.

Fonte: Kapandji (2002).

2.2 PRÓTESES NO BRASIL E SUA HISTÓRIA PELO MUNDO

Segundo o censo (IBGE, 2000), 5,5% da população brasileira possuía algum tipo de deficiência motora e no censo de 2010 este número subiu para 7%, ou seja, 13.265.599 brasileiros deficientes motores (IBGE, 2000). Este último censo ainda apresentou que a maior fatia desse grupo, era composta por mulheres de 15 a 64 anos. Visto que este número está se tornando cada vez maior, diversas pesquisas estão sendo realizadas na área de tecnologia assistiva, tendo em vista a tentativa de dar assistência e uma melhor aparência para o usuário, além é claro de contribuir para a realização de tarefas diárias. (HUNOLD et al., 2014)

Norton (2017) relatou que a história das próteses é longa, passando desde a ideia de próteses com pé em posição fixa, até ideias que foram totalmente esquecidas, como utilizar aço para a construção das mesmas, conforme a Figura 8. Este caminho se iniciou em 1500 a.C e só tem evoluído desde então. As primeiras próteses de mão foram realizadas utilizando ganchos e hoje próteses de alta tecnologia com movimentos individuais dos dedos e falanges tomam conta do mercado.

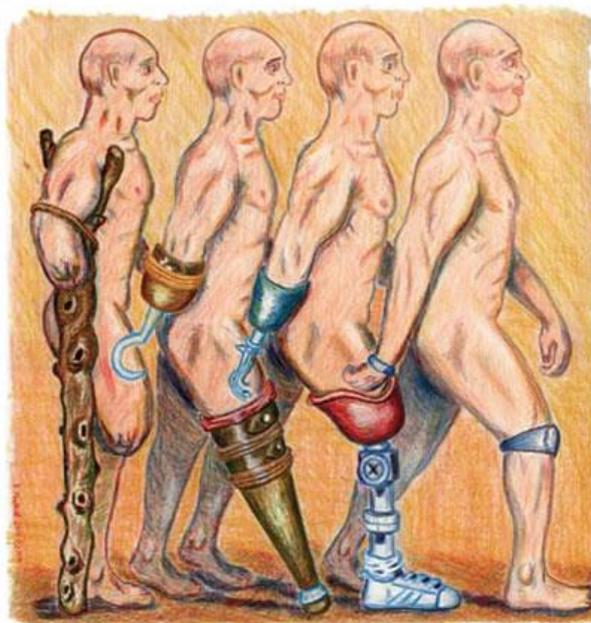


Figura 8 - Próteses com pés em posição fixa, de pau ou aço.

Fonte: Norton, 2007.

Os egípcios, pioneiros no ramo de protetização, criaram modelos feitos com fibra, com sentido de totalidade e não de funcionalidade. Estudos apresentaram ainda

que uma múmia foi encontrada com, a princípio, a primeira prótese totalmente funcional de dedo (NORTON, 2007).

Avançando na história, encontram-se relatos da época da renascença, próximos aos anos de 1500-1800. Este período marcou o início de novas perspectivas da arte, filosofia, ciência e medicina. Ao retornar para as descobertas médicas dos gregos e romanos sobre próteses, esta época apresentou um avanço na história das próteses. Durante este período elas foram feitas de aço, cobre e madeira, conforme apresentado na Figura 9 e Figura 10.



Figura 9 – Prótese de madeira.

Fonte: Norton, 2007.



Figura 10 - Prótese de aço.

Fonte: Norton, 2007.

Já avançando, no período da guerra civil americana, nasce um novo formato de próteses. Mais leve e sendo um dispositivo paciente-moldado, o retorno ao estilo de vida foi ofertado novamente aos amputados, deixando de lado próteses que apenas forneciam a funcionalidade básica ou uma aparência mais agradável. Nesta época, as próteses se tornam mais realistas. São moldadas em silicone e cobre, conforme a Figura 11, e capazes de imitar melhor a função de um membro natural (NORTON, 2007).



Figura 11 - Próteses de cobre e silicone.

Fonte: Norton, 2007.

Em 2015, na Áustria, o cirurgião Oskar Aszmann reuniu três pacientes com graves lesões no braço e mãos amputadas (ASZMANN, 2015). Estes pacientes já haviam tentado outros tratamentos com próteses controladas por ondas cerebrais, mas sem sucesso. Aszmann (2015) realizou um treinamento com os pacientes e enxertou uma parte do músculo da perna no braço do paciente, de forma que o pequeno sinal captado pelos nervos fosse amplificado. O desenvolvimento dos nervos nos músculos enxertados demorou três meses. Após esta fase e o longo treinamento de controle do músculo, o membro foi amputado e foi implantada a mão biônica de Aszmann. O procedimento foi bem sucedido e foi a primeira vez na história que pacientes escolheram ter suas mãos amputadas para receberem um membro biônico. O time do cirurgião visa criar agora uma prótese biônica que transmita a sensação de tato através de sinais elétricos aos nervos (GARCIA, 2015).

Atualmente existe tecnologia o suficiente para fabricar uma prótese caseira, usando impressão 3D, por exemplo, e até mesmo integrar componentes eletrônicos que proporcionam funcionalidade.

Este trabalho visa apresentar a inclusão de comandos de voz e eletrônica em uma prótese fabricada via manufatura aditiva. Para isso, é necessário compreender alguns assuntos relacionados com esta tarefa, tais como:

- Quais são os tipos de próteses que os deficientes físicos brasileiros conhecem hoje, se eles possuem acesso a elas e como eles se sentem inseridos na sociedade. Estas questões foram exploradas por uma pesquisa realizada pelos autores deste trabalho e os seus resultados podem ser encontrados no capítulo 2.3 PESQUISA ONLINE EM COMUNIDADE DE DEFICIENTES FÍSICOS;
- O movimento *maker*, o que são os *FabLab* e o *Mao3D*;
- A protetização infantil;
- Como são as próteses avançadas no mercado;
- Acessibilidade às pessoas de baixa renda;
- O nascimento da impressão 3D e como ela funciona;
- Quais são os materiais utilizados para esse tipo de impressão;
- As vantagens e desvantagens da manufatura aditiva;
- A diferença de valores entre próteses comerciais e próteses produzidas através da manufatura aditiva;
- Arduino e *Shields* e,
- O controle por reconhecimento de voz.

2.3 PESQUISA ONLINE EM COMUNIDADE DE DEFICIENTES FÍSICOS

Neste tópico são apresentados os resultados obtidos por pesquisa explanatória eletrônica realizada em uma comunidade *online* de deficientes físicos, realizada pelos autores no período de 2015 a 2017.

Foram feitas seis perguntas usando um formulário do *site* Survio para a comunidade “Amputados BRASIL” disponível na rede social *Facebook*, dentre os membros, apenas quatro se propuseram a responder o questionário.

Para a primeira pergunta “Você é usuário de próteses mecânicas/eletrônicas?” na qual, 50% dos respondentes confirmam serem usuários de próteses.

A segunda pergunta foi “Você já foi convidado a testar um modelo de prótese com controle eletrônico?” na qual, 75% dos respondentes informa não saber o que é uma prótese com controle eletrônico e 25%, nunca foi convidado a testar um modelo de prótese com controle eletrônico.

Na terceira pergunta, “Você se importaria em dar seu depoimento detalhado, sobre a experiência e adaptação com modelos de próteses?”, foi realizada uma tentativa de encontrar possíveis usuários que gostariam de dar um depoimento sobre a experiência que tiveram ou possuem com diversos modelos de próteses. Dos respondentes, 75% se propuseram a relatar sobre suas experiências.

A quarta pergunta foi “Há quanto tempo você utiliza prótese?”, foram oferecidas quatro opções: Não uso prótese de nenhum modelo, Menos de um ano, 1 a 2 anos, 2 a 3 anos ou Mais de 3 Anos. Sendo que 25% respondeu que não utiliza prótese, 25% utiliza a menos de um ano, 25% utiliza prótese entre 1 e 2 anos e os outros 25%, utiliza prótese a mais de três anos.

A quinta pergunta foi aberta e foi solicitado aos respondentes o seguinte: “Deixe sua opinião sobre o avanço tecnológico nas próteses, fale sobre sua experiência com próteses, o que você espera e o que pode melhorar etc.” Foram obtidas quatro respostas. São elas:

1. *“Ainda não sou usuário de próteses e isto faz com que eu não compareça a encontros de amigos ou aniversários... Tenho vergonha de me apresentar sem uma prótese, sou o centro das atenções. Não consigo fazer atividades simples no meu dia, como segurar duas coisas ao mesmo tempo ou digitar de forma rápida. Espero que com uma prótese esta situação seja revertida.”;*
2. *“Utilizo próteses há 33 anos e já passei por todo desenvolvimento possível de próteses. Atualmente utilizo a quatro anos a mesma prótese e pretendo adquirir uma melhor no ano que vem. O processamento eletrônico faz toda a diferença.”;*
3. *“Não conheço muito sobre a tecnologia e não posso opinar sobre o avanço. Porém, minha prótese dada pelo SUS foi a mais básica possível*

e para mim foi uma decepção, já que eu sou hiperativo e pretendia fazer muitas coisas normalmente. A prótese não é a ideal para mim e acabou me limitando. Folgada me causa dor e já quebrou o pé que veio com ela fixo. Agora estou andando com o pé amarrado em arame. Adoraria poder experimentar uma prótese descente que realmente me proporcionasse um desafio a ir além.”;

4. *“Não tenho muito conhecimento, mas sei de próteses com vários graus de liberdade, movimentos e até mesmo sensações táteis. queria ter a chance de usar uma prótese assim porque me sinto muito excluído de atividades normais e as vezes até mesmo esquecido no meio em que vivo... Espero que pessoas trabalhem mais e pesquisem mais para mudarem nosso futuro, o futuro dos deficientes físicos brasileiros.”;*

Com essa pergunta foi possível notar o desejo dos usuários sentirem suas funções motoras retornarem, entretanto com o que eles possuem atualmente isto não é possível. Muitos se sentem incapazes ou limitados e alguns, sofrem com dor.

A última pergunta foi um agradecimento e uma solicitação de comentários, entretanto, não foram coletados dados nesta pergunta.

2.4 MOVIMENTO MAKER

O conceito Movimento Maker (*Maker Movement*) - termo que em inglês significa aquele que cria ou aquele que faz muito difundido nos EUA e vem crescendo no Brasil, alterando a forma como as pessoas se relacionam com os objetos e processos de fabricação. Novas formas de produção, novos ambientes de trabalho e novas formas de se trabalhar estão sendo criadas, aumentando a colaboração e compartilhamento de informações ao dia-a-dia dos alunos (COMUNIDADE DE EDUCADORES DA MICROSOFT, 2015).

O Movimento Maker é uma tendência moderna de agregar pessoas, para que elas possam construir ou modificar em conjunto, de forma artesanal (ou seja, com as próprias mãos), qualquer objeto, este movimento busca facilitar essas reuniões, promovendo o “faça você mesmo”, termo que em inglês é muito popular, o “*do it yourself*” e segundo Neves (2016), esse movimento sempre existiu, porém não haviam os espaços apropriados para a propagação desse estilo de vida.

2.5 E-NABLE

A e-NABLE é uma comunidade global formada por pessoas dispostas a colaborar para melhorar e difundir projetos de impressão 3D para próteses de mãos e braços, visando ajudar em especial a crianças, mas também adultos, que nasceram com limitações nos membros superiores ou sofreram amputação seja por doença, guerra, acidente ou desastre natural (e-NABLE BRASIL).

Na página *online* da e-Nable (<http://enablingthefuture.org/which-design/>) são apresentados nove modelos de próteses mecânicas diferentes para os distintos tipos de amputações, que vão desde a amputação total dos dedos até a amputação acima do cotovelo. Para a desarticulação de cotovelo e amputações acima do cotovelo a página indica outros três desenvolvedores que trabalham com próteses mio elétricas, as quais são comandadas por impulsos elétricos musculares (ENABLING THE FUTURE).

As próteses indicadas para amputações que ainda contam com um punho funcional são: Phoenix Hand, Raptor Reloaded, Osprey Hand, Talon, Cyborg Beast, Ody Hand e Felxy Hand. As próteses com a articulação de cotovelo funcional são: Team Unlimbited Arm e RIT Arm (ENABLING THE FUTURE). As opções indicadas para desarticulação do cotovelo e acima são as dos desenvolvedores: Limbitless Solution Arm, Open Bionics e EXIII.

2.6 FAB LAB

Dentro do movimento *maker*, a rede *Fab Lab* (*Fabrication Laboratory*), em português “Laboratório de Fabricação”, com sua filosofia de fabricação local vem expandindo e oferecendo novas oportunidades auxiliada por uma rede global. Dentro desta filosofia, o pesquisador Paulo Blikstein vem implantando laboratórios dentro de escolas através do projeto *FabLab@School* (Laboratório de Fabricação na Escola), como objetivo de inserir os alunos nesse movimento, através de espaços desenhados e construídos especialmente para comportar instrumentos de robótica, impressoras 3D e cortadoras a laser dentro das escolas, com o foco em desenvolver o aprendizado

e a criatividade das crianças (COMUNIDADE DE EDUCADORES DA MICROSOFT, 2015).

O Movimento Maker se caracteriza pelo uso de *hardware* aberto (*hardware* cujos projetos são disponibilizados publicamente de modo que qualquer um possa estudar, modificar, distribuir, fabricar e vender o projeto ou o *hardware* baseado no projeto (OSHW, 2015), cortadoras lasers, impressoras 3D, para desenvolver os mais diversos produtos eletrônicos e robóticos.

Em uma *Fab Lab*, pode se fazer desde uma caixinha de som artesanal até uma prótese. Um exemplo disso foi uma prótese de mão infantil totalmente projetada e fabricada pela Garagem FabLab. (DIALOGANDO, 2016).

2.6 MAO3D - PROGRAMA SOCIAL E COLABORATIVO DA UNIFESP

A tecnologia de impressão 3D não é relativamente nova, a mesma data desde 1984 (DEURSEN, 2016). Entretanto, vem sendo aprimorada com o passar dos anos e é um recurso tecnológico com o qual pode se produzir os mais diversos objetos físicos com materiais diferentes, como o plástico, partindo de um modelo digital, e o que não era possível fazer até pouco tempo.

Na última década já estão sendo realizados diversos modelos de prótese de braço e mão para adultos e crianças, graças à esta tecnologia e à algumas comunidades – como a e-Nable – que disponibilizam na internet modelos abertos de impressoras 3D de baixo custo e *softwares* livres de modelagem 3D. As próteses disponibilizadas pela e-Nable são esteticamente agradáveis, de baixo peso e podem ser totalmente personalizáveis (KUNKEL, 2016a).

O Instituto de Ciência e Tecnologia da Universidade Federal de São Paulo (UNIFESP), que faz parte da Rede Nacional de Núcleos de Pesquisa em Tecnologia Assistiva, em parceria com o Centro de Reabilitação Lucy Montoro, ambos na cidade de São José dos Campos conceberam o Programa de Extensão Mão3D. O principal objetivo do programa é utilizar a impressão 3D para produzir próteses para pessoas com amputação ou má formação dos membros superiores, protetizando e reabilitando 100 pessoas da região do Vale do Paraíba (UNIVERSIDADE FEDERAL DE SÃO PAULO, 2016).

“Na região do Vale do Paraíba, cerca de 47 mil pessoas são deficientes motores com ausência ou malformação de braços, mãos ou dedos. Com a perda da capacidade manipulatória, o indivíduo pode ter uma vida menos funcional, mais dependente e passa a ser excluído da sociedade.” (UNIVERSIDADE FEDERAL DE SÃO PAULO, 2016).

O projeto visa atender em sua maioria crianças, pois são as que ficariam sem prótese até alcançar a idade adulta, pois no Brasil não existem modelos infantis funcionais, portanto, os modelos utilizados são de *design* aberto fornecidos pela ONG *E-Nable*, e a prioridade será dada à região do Vale do Paraíba devido à proximidade, para que seja dado o devido acompanhamento, reabilitação e possíveis ajustes que podem surgir neste processo, e isso é feito em parceria com um Instituto de Saúde local que atende pelo SUS (3DPRINTING, 2016).

Segundo a pesquisadora Maria Elizete Kunkel em entrevista para o site *3DPrinting*, ainda a literatura nesta área está no início, não havendo muitos artigos científicos sobre próteses e impressão, por este motivo, foi criado um blog do Mao3D para difundir o conhecimento sobre a cultura *Maker*, impressão 3D e próteses no Brasil, assim contribuindo com essa literatura, escrevendo artigos científicos e oferecendo treinamento e capacitação para a fabricação das próteses e para a reabilitação dos usuários, na mesma entrevista Kunkel fala também sobre as impressoras utilizadas pela instituição Mao3D, explicitando que as impressoras utilizadas são de baixo custo, pela quantidade ainda grande de problemas que as impressoras 3D apresentam (3DPRINTING, 2016).

2.7 A PROTETIZAÇÃO INFANTIL

O programa Mão3D produz próteses de membros superiores baseadas nos modelos livres da e-Nable, que são adaptadas e personalizadas para a realidade de cada criança. Essas próteses são do tipo mecânico, sem componentes eletrônicos de controle, sendo o movimento de abrir e fechar a mão controlado pela articulação mais próxima ao coto, o punho ou cotovelo. O “coto” é a parte do membro que permanece após a amputação, ou a parte malformada em pessoas que nascem sem parte do membro superior. Quando a distância da extremidade do coto até a próxima articulação é muito pequena a dificuldade de se produzir uma prótese aumenta, pois é necessário que haja uma parte rígida o suficiente para sustentar e movimentar a

prótese (KUNKEL, 2016b). Em casos em que a amputação é muito próxima ao cotovelo, existem somente duas outras opções:

1. O uso de uma prótese mecânica acionada por tirantes pelo movimento do outro ombro;
2. O uso de uma prótese mio elétrica. Este tipo de prótese utiliza sensores que captam sinais elétricos provenientes dos músculos do protetizado, esses sinais são lidos por um microcontrolador que filtra e amplifica o sinal, acionando motores que controlam os movimentos de abrir e fechar da mão da prótese, sendo todo esse sistema alimentado por baterias recarregáveis, o sistema é complexo, ainda não existindo no mercado nacional, próteses mio elétricas adequadas para o uso infantil.

A e-Nable está no processo de desenvolvimento de alguns modelos de próteses mio elétricas infantis feitas por impressão 3D, mas eles ainda não estão prontos (KUNKEL, 2016b).

2.8 PRÓTESES AVANÇADAS NO MERCADO

A *Steeper Group*, uma empresa britânica, criou a “*BeBionic*”, uma mão biônica que promete ser a mais “anatomicamente precisa” do mercado, oferecendo um “incomparável nível de precisão e movimento natural” (MALLEY, 2015).

Utilizando uma tecnologia presente nos carros de Fórmula 1, a *Steeper Group* construiu, com exatidão, uma estrutura de esqueleto e componentes em miniatura. Desta forma, ela não é uma prótese volumosa e pode realizar as funções de uma mão real, com 14 diferentes formas de segurar objetos.

A mão funciona com sensores mio elétricos que são ativados com os movimentos dos músculos no braço de protetizado. “Cada dedo possui motores próprios, assim podendo se mover independentemente dos outros; ao todo, a mão possui 337 partes mecânicas que a fazem funcionar.” (MALLEY, 2015).

Os dedos são feitos de ímãs para “melhorar o desempenho e o equilíbrio entre velocidade e força”. A mão pode aguentar até 45 kg. Para mantê-la ainda mais real, as pontas dos dedos da prótese contêm bolhas de ar, para imitar o fato de dedos de verdade serem um pouco esponjosos. O valor da

mão biônica, entretanto, ainda é uma realidade distante para muitos: ela custa aproximadamente US\$ 58.000 por unidade — caro, mas dá a pessoas como Nicky movimentos que elas nunca nem imaginaram ter (MALLEY, 2015).

2.9 ACESSIBILIDADE PARA PESSOAS DE BAIXA RENDA

A Universidade Federal de Goiás (UFG) está ajudando quem precisa de uma prótese. Os pesquisadores produzem as peças das próteses utilizando a tecnologia de impressão 3D, conseguindo produzi-las com um custo bem abaixo do que o usual (GLOBO, 2015). Os pesquisadores da Universidade Federal de Goiás desenvolveram um aplicativo que move a mão eletrônica através da fala ou do apertar de um botão. O projeto visa fornecer próteses para pessoas com amputação ou má formação da mão como uma alternativa mais barata em relação às que já existem no mercado.

Segundo o coordenador do projeto da UFG, Marcelo Stoppa, a prótese eletrônica comercial, que tem um custo de aproximadamente R\$ 150 mil é produzida na íntegra, com motor e eletrônica funcional, por menos de R\$ 5 mil, sendo produzida com polímero derivado do milho, buscando conforto e leveza no produto (GLOBO, 2015).

A empresa japonesa EXII disponibilizou o que diz ser a mudança para quem necessita de um braço, o *HACKberry*, um projeto de membro biônico, cujo *design* e código-fonte foram disponibilizados gratuitamente na internet. O projeto utiliza um *smartphone* como microprocessador e usa baterias de câmeras como alimentação de energia (STONE, 2015).

O dispositivo é composto, em sua maior parte, por peças feitas em impressora 3D, que podem ser retiradas e trocadas, além das peças eletrônicas, parafusos e componentes que precisam ser adquiridos para a produção da mesma. A empresa também disponibiliza todas as especificações destes componentes. Ao fazer isso, a EXIII colabora com a comunidade criadora, estimulando a transparência em todo o desenvolvimento e abre um precedente, buscando trazer inovações de maneira mais rápida à este meio (STONE, 2015).

A prótese da EXIII tem algumas características impressionantes, como os dedos e pulsos flexíveis, o que permite ao usuário da mão biônica fazer movimentos extraordinariamente expressivos, como segurar objetos grandes e pequenos, folhear revistas e até amarrar sapatos. “Apesar de o modelo ainda não estar à venda, o

protótipo custa cerca de US\$ 300 para produzir. Mesmo se uma versão futura chegar ao mercado com um preço bem maior, ainda será muito mais em conta que outras próteses.” (STONE, 2015).

2.10 PRÓTESES ATUAIS E DIFERENÇAS NOS VALORES

A pesquisa realizada pela Agência Nacional de Saúde Suplementar (ANS) apresentou uma variação de até 3.108% em preços de órteses (AGÊNCIA NACIONAL DE SAÚDE SUPLEMENTAR, 2016). O levantamento avaliou os preços pagos por operadoras de planos de saúde por dois grupos de dispositivos implantáveis, usados para melhorar a circulação sanguínea arterial: a endoprótese vascular e os *stents* farmacológicos. Os resultados da pesquisa inédita revelam a variação de custos entre os estados do país e as formas de aquisição.

De acordo com a ANS, a variação entre os valores de comercialização é esperada, considerando aspectos como transporte, armazenamento, tributação e poder de negociação. No entanto, os níveis observados de variação de preços no mercado de dispositivos médicos implantáveis “alcançam proporções extremamente elevadas, o que pode estar associado a condutas antiéticas, anticoncorrenciais ou ilegais” (AGÊNCIA NACIONAL DE SAÚDE SUPLEMENTAR, 2016).

2.11 NASCIMENTO DA IMPRESSÃO 3D

A impressão 3D está revolucionando diversos setores da economia. Esta tecnologia é aplicada em diversas áreas e é encontrada desde a fabricação de chocolates até mesmo próteses biônicas. Entretanto, esta tecnologia não é recente. Sua primeira máquina funcional completou 30 anos em 2014 (3DECIA, 2017).

A primeira impressora 3D foi inventada por Chuck Hull, na Califórnia em 1984, utilizando a estereolitografia, tecnologia precursora da impressão 3D. Hull já havia desenvolvido um ano antes a tecnologia do que viria a ser a máquina, quando ela tinha duas funções principais, sendo uma delas a criação de lâmpadas para solidificação de resinas, primeiro objeto criado pela ferramenta. A principal, entretanto, foi a confecção de partes de plástico de forma rápida, já que o processo tradicional levava de seis a oito semanas, e as peças ainda precisavam ser refeitas diversas vezes devido a problemas na manufatura. Poucos anos depois, Chuck Hull fundou a

3D Systems Corp., patenteando sua criação e diversas formas de impressão, assim como iniciando a comercialização da tecnologia. Para se ter uma ideia do sucesso, a empresa permanece como uma das líderes do segmento até hoje (DUARTE, 2014).

2.12 COMO FUNCIONA A IMPRESSÃO 3D

Para se imprimir em 3D, primeiramente deve-se realizar um projeto do objeto a ser impresso. O usuário deverá desenvolver um modelo do objeto dentro de um aplicativo de modelagem 3D em seu próprio computador. Este modelo deve ser enviado para a impressora, através de um *software* que irá fatiar o modelo desenhado em de camadas e em duas dimensões. A partir deste momento, se inicia a impressão 3D. A impressora será acionada e fará com que o bico injetor aqueça e puxe o filamento plástico (PLA ou ABS), derretendo o material e o depositando em camadas na base. A plataforma desce, ou o bico injetor subirá, dependendo do modelo da impressora, o que permite a criação da próxima camada. Ao fim do processo de depósito de camadas de plástico, a peça estará pronta. O tempo de impressão varia desde poucos minutos até mesmo dias, o que determina isto é o tamanho e definição da peça modelada.

2.13 MATERIAIS DA IMPRESSÃO 3D

Existem disponíveis no mercado diversos materiais para a impressão 3D, sendo que os mais utilizados são o PLA e o ABS. Eles serão explicados a seguir.

O plástico PLA é um dos materiais mais utilizados no mundo da impressão 3D por ter custo benefício e durabilidade. É um material de base biodegradável e atóxico, disponível em várias cores. Seus benefícios são diversos os de destaque são: Acabamento brilhoso, ótimo custo benefício, atóxico. Porém, a impressão realizada com este material apresenta as suas camadas visíveis pós-impressão e o material pode sofrer alterações com a influência da umidade do ar (CAMMADA, 2018).

Assim como o PLA, o ABS é um material bastante popular. Apresenta boa resistência mecânica e térmica, sendo bastante utilizado em protótipos. Por ser derivado do petróleo não é indicado para contato intenso com a pele, podendo causar irritação. O ABS possui diversos benefícios, entre eles podem citar a alta resistência mecânica, facilidade em colar as após o processo e facilidade em pintar a mesma. O

ABS também apresenta camadas visíveis pós-impressão, somado a uma aparência fosca. Este material não apresenta uma grande gama de cores e além de ser propenso a influência da umidade do ar, ele se degrada ainda por excesso de exposição a luz. (CAMMADA, 2018).

Ainda podem ser utilizados outros materiais menos divulgados para a impressão 3D, como o PETG (Polietileno Tereftalado modificado com Glicol), Nylon, Resinas e até mesmo, gesso. (CAMMADA, 2018).

2.14 VANTAGENS E DESVANTAGENS DA MANUFATURA ADITIVA

Como foi visto, a manufatura aditiva tem ganhado um grande espaço no meio dos *makers*. Entretanto, não há que se esperar apenas um lado positivo desta nova facilidade. Logo, são apresentados os prós e contras do uso desta tecnologia.

Prós:

- Possibilita o percurso da teoria à prática (do arquivo digital ao objeto) em poucos passos;
- Possibilidade de criar objetos únicos, customizados e personalizados a um custo baixo;
- Variedade de materiais aceitos pelas impressoras e de aplicações da tecnologia;
- Sustentabilidade e economia, já que há diminuição de resíduos e produção mais eficiente;
- Facilita a inovação e conseqüentemente a criatividade;

Contras:

- Lentidão nos processos (pode-se levar mais de uma hora para imprimir objetos do tamanho de uma xícara);
- Complexidade na operação da impressora e no desenvolvimento dos arquivos;
- Dificuldade de operação, calibração e manutenção (DESTINO NEGÓCIO, 2015).

2.15 MODELOS DE PRÓTESES DE CÓDIGO ABERTO

A comunidade não comercial sem fins lucrativos *E-Nable*, cujo lema é, em tradução livre do inglês, *Habilitando o Futuro – Uma rede global de voluntários apaixonados usando a impressão 3D para dar ao mundo uma “mão amiga”, (Enabling the Future - A global network of passionate volunteers using 3d printing to give the world a "helping hand."*), desenvolve e disponibiliza diversos modelos de próteses, incluindo os arquivos para a fabricação e instruções de montagem necessários para a produção de uma prótese completa.

Os modelos são separados em conjuntos, de acordo com os distintos tipos de amputações. Os modelos do primeiro conjunto são para a amputação total dos dedos até parcial de palma, mas com a articulação do punho funcional; o segundo conjunto para desarticulação desde o punho até a parcial de antebraço, mas contando com a articulação de cotovelo funcional; e para o terceiro conjunto, quando se tem a amputação acima do cotovelo, sendo que para este último grupo a página indica outros três desenvolvedores que trabalham com próteses mio elétricas, as quais são comandadas por impulsos elétricos musculares.

Como este trabalho visa utilizar o reconhecimento de voz para adaptar um modelo de prótese puramente mecânico para que atue de forma eletrônica, o tipo da amputação não deverá afetar o intuito deste projeto, sendo que uma vez funcional em uma prótese, deverá ser facilmente adaptado para outras próteses, visto a necessidade de cada usuário. Desta forma, serão apresentados alguns modelos de próteses mecânicas disponíveis no mercado para a manufatura aditiva que melhor atendam às necessidades dos usuários e os requisitos desta proposta, que são: boa usabilidade, baixo custo, facilidade de fabricação, e possibilidade de integração com a eletrônica através do reconhecimento de voz.

2.15.1 MODELOS PARA DESARTICULAÇÃO DE DEDOS E PARCIAL DE PALMA

Para amputações que possuam desarticulação de dedos e parcial de palma que ainda contam com um punho funcional, conforme demonstrado na Figura 12, são apresentados alguns modelos de próteses, entre elas: Phoenix Hand (Figura 13),

Raptor Reloaded (Figura 14), Osprey Hand (Figura 15), Talon (Figura 16), Cyborg Beast (Figura 17), Ody Hand (Figura 18) e Flexy Hand (Figura 19).

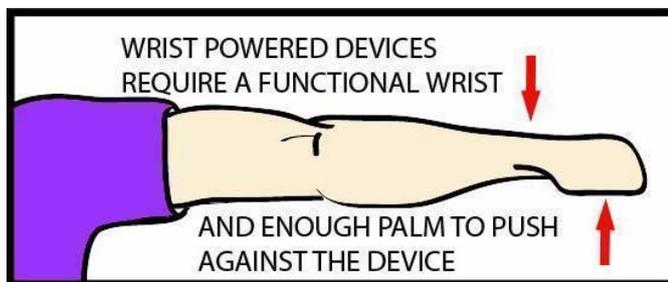


Figura 12 - Braço com articulação de punho funcional.

Fonte: E-NABLE COMMUNITY.



Figura 13 - Prótese modelo Phoenix Hand.

Fonte: E-NABLE COMMUNITY.



Figura 14 - Prótese modelo Raptor Reloaded.

Fonte: E-NABLE COMMUNITY.

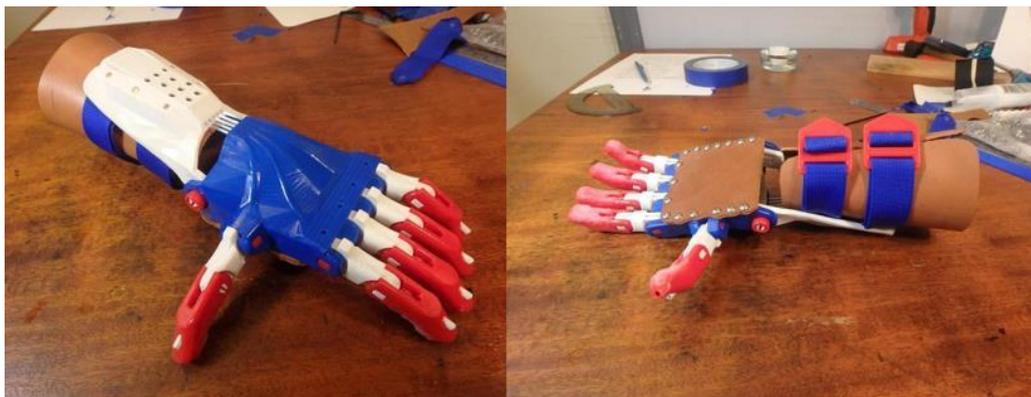


Figura 15 - Prótese modelo Osprey Hand.

Fonte: E-NABLE COMMUNITY.



Figura 16 - Prótese modelo Talon.

Fonte: E-NABLE COMMUNITY.



Figura 17 - Prótese modelo Cyborg Beast.

Fonte: E-NABLE COMMUNITY.



Figura 18 - Prótese modelo Ody Hand.

Fonte: E-NABLE COMMUNITY.



Figura 19 - Prótese modelo Flexy Hand.

Fonte: E-NABLE COMMUNITY.

2.15.2 MODELOS PARA DESARTICULAÇÃO DE PUNHO E PARCIAL DE ANTEBRAÇO

Para amputações que possuem a articulação de cotovelo funcional, conforme apresentado na Figura 20, as próteses mais indicadas são: Team Unlimbited Arm (Figura 21) e a RIT Arm (Figura 22).

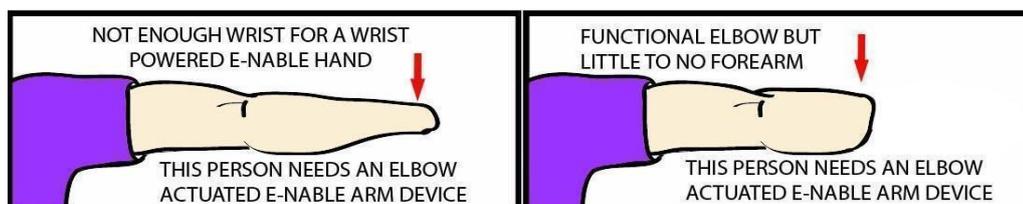


Figura 20 - Braço com articulação de cotovelo funcional.

Fonte: E-NABLE COMMUNITY.



Figura 21 - Prótese modelo Team Unlimbited Arm.

Fonte: E-NABLE COMMUNITY.



Figura 22 - Prótese modelo RIT Arm.

Fonte: E-NABLE COMMUNITY.

2.15.3 MODELOS PARA DESARTICULAÇÃO DE COTOVELO E PARCIAL DE BRAÇO

As opções indicadas para amputações que sofreram desarticulação do cotovelo e acima (Figura 23), são as dos desenvolvedores: Limbitless Solution Arm, Open Bionics e EXIII, sendo o modelo Hackberry apresentado na Figura 24, da EXIII, a primeira opção proposta para este trabalho, contendo 63 peças impressas em 3D, e a necessidade de um alto acabamento destas peças para seu bom funcionamento, o custo da impressão, orçado em Curitiba, ficou em aproximadamente R\$ 3.000,00, portanto, pelo alto custo e devido à maior quantidade de componentes necessários para a sua fabricação, além dos componentes eletrônicos mais caros e indisponíveis nacionalmente, como por exemplo o sensor mioelétrico, que sozinho custa em média mais de R\$ 300,00 (trezentos reais), a prótese base do estudo deste trabalho foi alterada para o modelo Raptor Reloaded, com controle de voz, devido à sua disponibilidade e custo mais acessível dos componentes.

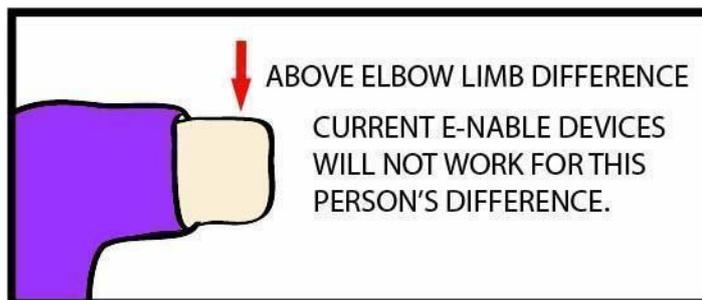


Figura 23 - Braço com desarticulação de cotovelo e acima.

Fonte: E-NABLE COMMUNITY.



Figura 24 - Prótese modelo Hackberry da EXIII.

Fonte: Exiii-hackberry.com.

2.15.4 PRÓTESE RAPTOR RELOADED

Dentre as próteses apresentadas, a que será utilizada neste trabalho é a do modelo Raptor Reloaded, que requer a impressão em 3D de 12 tipos de peças distintas nas seguintes quantidades listadas a seguir, totalizando 31 peças para cada mão completa:

- 1x palma esquerda ou palma direita;
- 2x pino de junta;
- 1x pino de polegar;
- 5x falange proximal;
- 5x pino de dedo;
- 5x falange distal;
- 1x luva;
- 2x pino de pulso;
- 2x capa do pino do pulso;

- 1x tensionador;
- 1x clip de retenção;
- 5x pino tensionador.

A montagem de uma prótese no modelo Raptor Reloaded requer também os seguintes materiais:

- (1,22m) Velcro, dupla face (0,05m de largura);
- (2,44m) Cabo de nylon não elástico, 0,9 mm;
- (1,55m) Cabo elástico flexível;
- (5) Parafusos tensores;
- (3) Parafusos de fixação de velcro de palma com porcas de parada;
- (0,30m) Revestimento de espuma firme (0,15m de largura);
- (5) Lee Tippi Micro Gel Fingertip Grips (opcional).

Os *links* para os arquivos em formato “.stl” estão disponíveis na seção Anexos (ANEXO A – ARQUIVOS STL DA PRÓTESE RAPTOR RELOADED).

3 MÓDULOS PARA PROJETO ELETRÔNICO DE CONTROLE DE MOVIMENTAÇÃO

3.1 MICROCONTROLADOR ARDUINO NANO

O microcontrolador que será utilizado para o controle da prótese é o Arduino Nano (Figura 25). Este possui dimensões de 43 x 15 mm, uma entrada USB para conexão a um computador para a transferência do código do programa. O Arduino Nano possui saídas de tensão de 3,3V e 5V, respectivamente nos pinos 3V3 e 5V. As entradas analógicas são os pinos A0 a A7 que podem receber valores entre 0 e 5V, enquanto as entradas digitais D0 a D13 reconhecem valores na faixa de 0V para nível lógico baixo e de 5V para alto. Os pinos A0 a A5 podem operar também como saídas digitais, enquanto os pinos D3, D5, D6, D9, D10 e D11, sinalizados na placa com um ponto, suportam saída analógica através de PWM (técnica de modulação na qual a característica modificada é o pulso da onda.) (ELETRÔNICA PARA ARTISTAS, 2017).

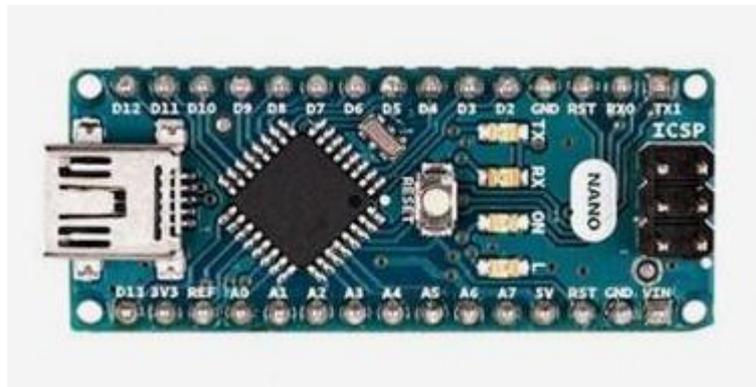


Figura 25 - Arduino Nano.

Fonte: Eletrônica para artistas, 2017.

A correspondência dos pinos do Arduino e suas funções estão representadas na Figura 26.

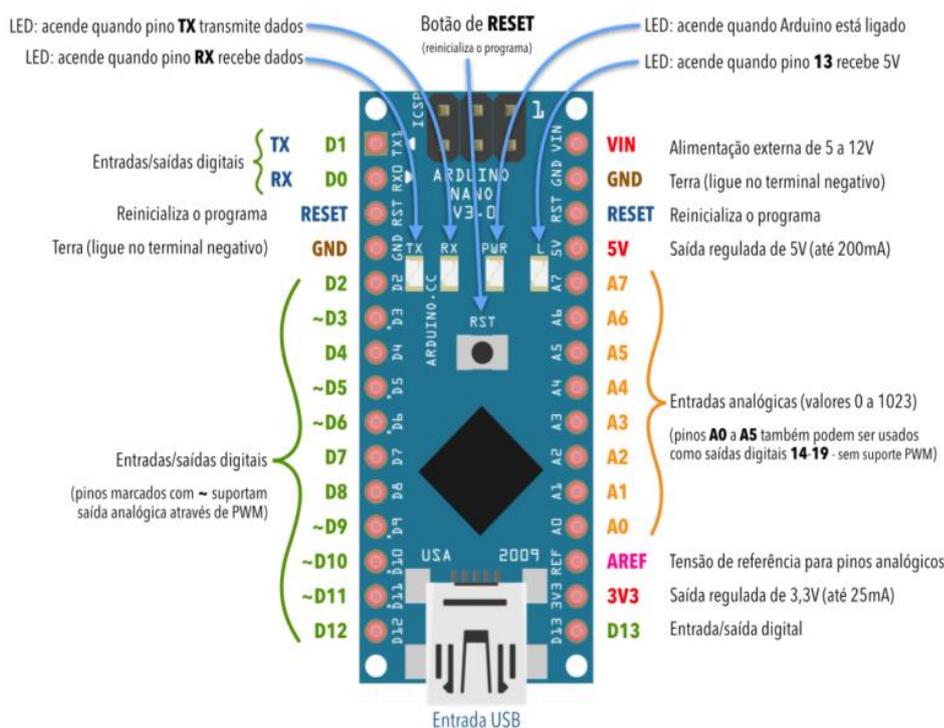


Figura 26 - Pinos do Arduino Nano.

Fonte: Eletrônica para artistas, 2017.

3.2 MÓDULO DE RECONHECIMENTO DE VOZ

O processo de reconhecimento de voz consiste em diversas etapas: primeiramente o computador capta as vibrações criadas pela voz e faz a conversão dessas ondas em dados digitais utilizando um conversor analógico-digital. Na sequência, as ondas captadas são mensuradas e filtradas, separando a voz de ruídos e interferências, e nas próximas etapas, as frequências contidas na voz são computadas, sincronizadas, se necessário e depois separadas em sons fonéticos não maiores que uma sílaba, para só então os sons captados serem comparados com fonemas já gravados no banco de dados (PEREIRA, 2009).

O módulo de reconhecimento de voz utilizado neste trabalho é o modelo FZ0475 com microfone, o mesmo é apresentado na Figura 27 e permite gravar até 80 comandos, sendo que somente sete deles poderão ser utilizados ao mesmo tempo. É importante salientar que as alterações significantes no tom de voz e no modo de falar sofrem variações dependendo do locutor, portanto é necessário que os comandos sejam gravados pela pessoa que fará a utilização do módulo (STRAUB, 2015).



Figura 27 - Módulo de reconhecimento de voz F 0475.

Fonte: SmartKits, 2018.

3.3 MÓDULO DE RELÉS

O módulo de relés utilizado é o Módulo Relé 5V 10A, de quatro canais com optoacopladores, por meio da qual é possível controlar até quatro dispositivos de forma independente, sendo compatível com o Arduino. O mesmo é apresentado na Figura 28. Sua tensão de entrada é de 5VDC e pode controlar sistemas de corrente contínua e alternada, nos quais a tensão de operação é de até 30VDC para corrente contínua e 250VAC para corrente alternada, desde que a corrente não ultrapasse o valor de 10A para ambos os casos (FILIPEFLOP, 2018).

O diferencial deste módulo são os optoacopladores, que isolam a região de entrada (do microcontrolador Arduino), da região de saída (servo-motores), protegendo aquele de possíveis sobrecargas causadas pelo servo-motores, desempenhando a função de um sistema de segurança para o Arduino e módulo de reconhecimento de voz.



Figura 28 - Módulo Relé 5V 4 Canais.

Fonte: FilipeFlop, 2018.

3.4 SERVO MOTORES

O servo-motor é utilizado em aplicações para robótica, sistemas com microcontroladores, aeromodelismo e outras aplicações que requerem um controle preciso dos movimentos, sendo um dispositivo de malha fechada que verifica a posição atual e se move para a posição desejada através do recebimento de um sinal de controle.

Neste trabalho será utilizado o modelo Micro Servo Motor 9g SG90 Tower Pro 180°, apresentado na Figura 29, que pesa somente nove gramas e oferece um torque máximo de 1,2kgf.cm em 4,8V de alimentação até um torque de 1,8kgf.cm alimentado em 6V, com um ângulo de movimentação de até 180° (FILIFELOP, 2018).

Este servo possui três fios de alimentação, sendo dois para alimentação e outro para o recebimento do sinal de controle, podendo ser encontrado com a sequência de cores marrom, vermelho e amarelo ou preto, vermelho e branco, dos quais o marrom ou preto são conectados ao pino GND, o vermelho ao pino 5VCC e o amarelo ou branco é conectado a um pino de saída de sinal digital com PWM.



Figura 29 - Micro Servo Motor 9g SG90 Tower Pro 180°.

Fonte: Eletrogate, 2018.

3.5 BATERIAS

Neste trabalho serão utilizadas três baterias recarregáveis de 9 Volts da marca MOX, essas baterias são de hidreto metálico de níquel (Ni-MH), com

capacidade de 320mAh e têm uma vida útil de 500 ciclos de carga (MOXGROUP, 2018). O modelo é o MO-9V320 apresentado na Figura 30.



Figura 30 - Bateria MOX 9 Volts 320mAh modelo MO-9V320.

Fonte: Moxgroup, 2018.

3.6 PROJETO ELETRÔNICO

A conexão dos módulos eletrônicos deve ser executada seguindo as ligações exatamente como o descrito nesta seção, para que não haja o mau funcionamento ou danos temporários, ou permanentes a algum módulo.

Os pinos do módulo de reconhecimento de voz deverão ser conectados ao Arduino de acordo com a Tabela 1.

Tabela 1 - Conexão dos pinos do Arduino ao módulo de voz.

Módulo de Reconhecimento de Voz	Microcontrolador Arduino Nano
GND	GND
VCC	5V
RXD	3
TXD	2

Fonte: Usina Info, 2018.

Da mesma forma o módulo de relés deve ser conectado ao Arduino como na Tabela 2.

Tabela 2 - Conexão do arduino ao módulo de relés.

Módulo de Relés	Microcontrolador Arduino Nano
GND	GND
VCC	5V
IN1	13
IN2	12
IN3	11

Fonte: Usina Info, 2018.

Os servos-motores receberão o sinal de posição através dos fios amarelos (ou brancos, dependendo do modelo) conectados ao Arduino nos pinos 6, 9 e 10, e a alimentação destes se dará pela segunda fonte de energia, sendo os fios pretos ou marrons conectados aos polos negativos das baterias, enquanto os fios vermelhos do servo-motores serão conectados aos terminais NF (normalmente fechados) dos módulos relés K1, K2 e K3 respectivamente, e o comum destes mesmos relés estarão conectados ao polo positivo das baterias.

A alimentação do Arduino se dará por uma primeira fonte de energia, uma bateria que será conectada aos pinos VIN e GND do Arduino, o primeiro ligado ao polo positivo da bateria, passando por um interruptor ou chave para ligar e desligar o Arduino enquanto o GND será conectado ao polo negativo da bateria.

A alimentação dos motores será independente da alimentação do Arduino, porém, os polos negativos de ambas as fontes de energia devem estar ligados um ao outro, para que os sinais dos motores, que vêm do Arduino funcionem corretamente, do contrário os motores ficam sem controle algum.

Todas as conexões dos componentes eletrônicos estão no projeto eletrônico de controle da prótese, detalhado no diagrama de blocos apresentado na Figura 31, com a fiação em cores diferentes e de acordo com o projeto realizado, para facilitar na montagem e identificação da mesma.

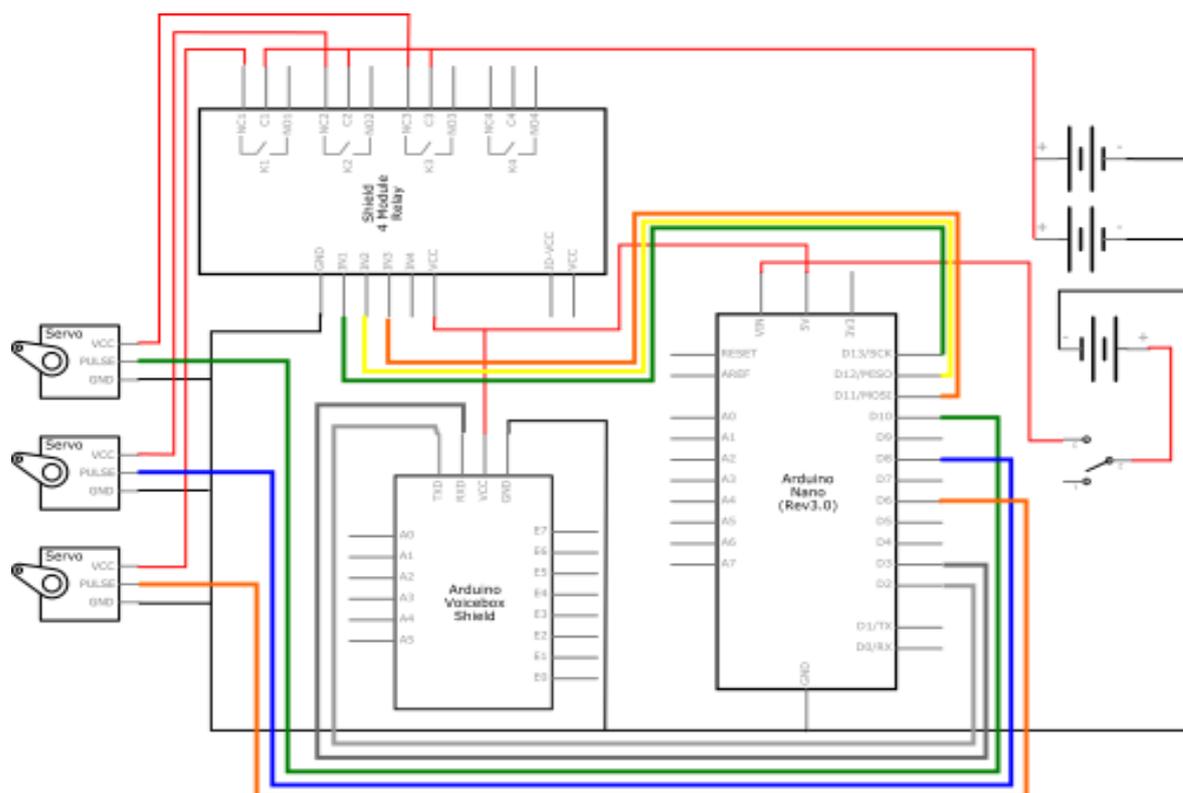


Figura 31 - Diagrama de ligação do controle da prótese.

Fonte: Os Autores, 2018.

Para a montagem dos circuitos eletrônicos foi utilizada como base uma chapa de acrílico de 100mm x 200mm x 5 mm. Em outra chapa de acrílico de 45mm x 70mm foi feito um recorte retangular para encaixe dos servos, de forma que estes fiquem ajustados na posição desejada. Todos os componentes foram fixados através de parafusos M3x12mm com porcas, com exceção do Arduino Nano, o qual foi fixado através de fio de Poliamida, conforme a Figura 32.

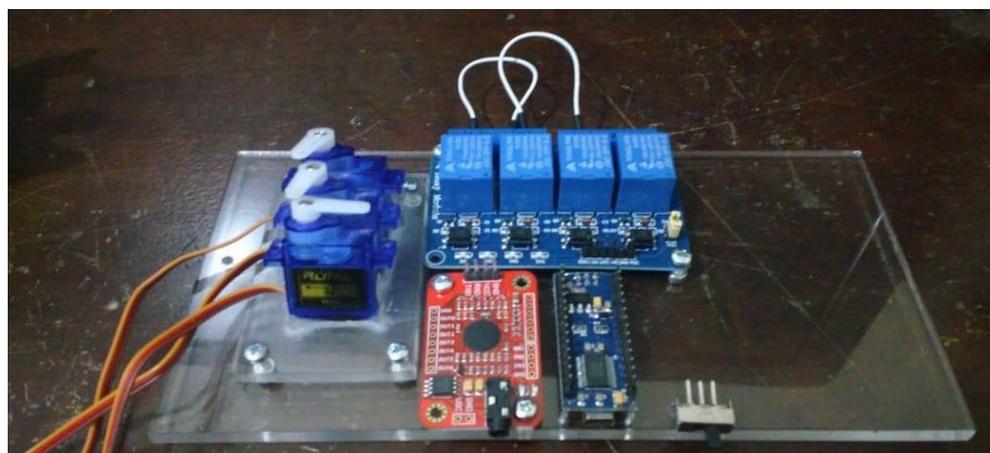


Figura 32 – Projeto eletrônico fixado na placa de acrílico (sem as baterias).

Fonte: Os Autores, 2018.

Após a ligação dos componentes, segundo o especificado neste capítulo sem as baterias, o projeto eletrônico é mostrado na Figura 33.

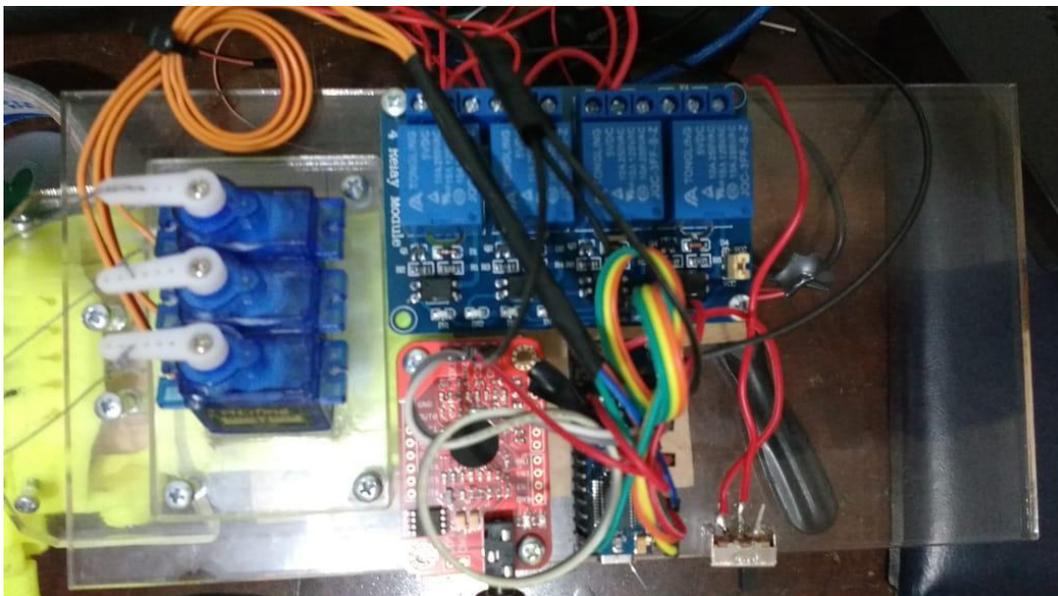


Figura 33 – Projeto eletrônico com fiação na placa de acrílico (sem as baterias).

Fonte: Os Autores, 2018.

3.7 INTEGRAÇÃO DO PROJETO ELETRÔNICO NO PROTÓTIPO

Estando toda a parte eletrônica fixada em uma placa de acrílico, tem-se um módulo de controle de movimentos, o qual pode ser adaptado em diversos modelos de próteses. Neste trabalho será utilizado o modelo *Raptor Reloaded*, ao qual foi fixado através de furação já existente no protótipo. O módulo de controle e após a fixação e os fios que fecham os dedos da prótese foram conectados às hélices, ou braços. Foram devidamente testados para que pudesse ser feito o ajuste dos ângulos de movimentação dos motores na programação do Arduino, a fim de que os comandos executem o movimento devido em cada dedo, conforme a Figura 34.

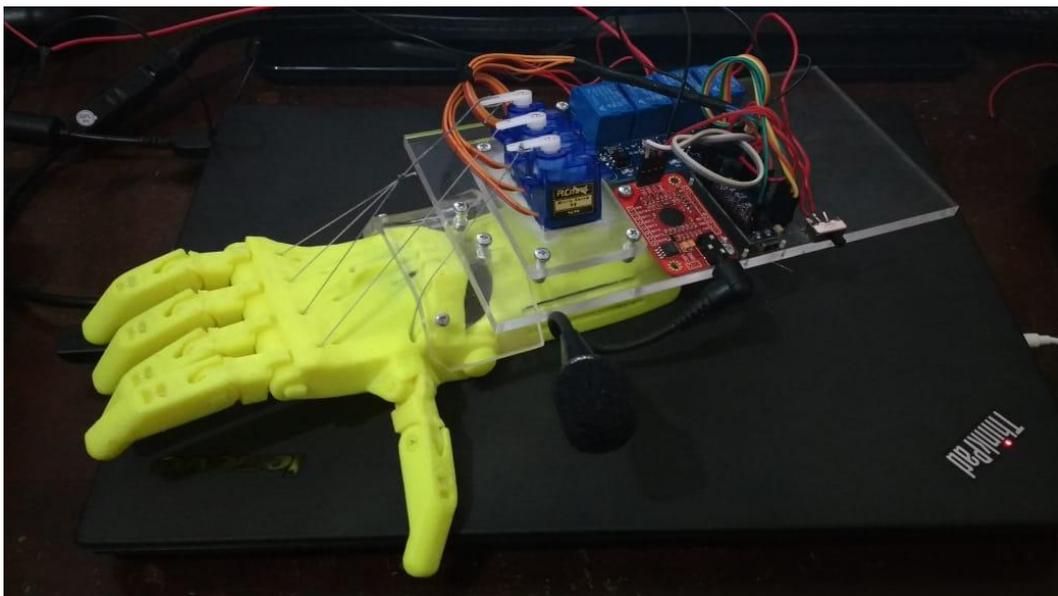


Figura 34 – Módulo de controle fixado ao protótipo.

Fonte: Os Autores, 2018.

Foi construída, também, uma base em MDF, para que a prótese pudesse ter os movimentos de seus dedos livres de obstrução e também uma boa apresentação (Figura 35). Em uma prótese para a efetiva utilização por uma pessoa, esta base não é necessária.



Figura 35 – Base em MDF.

Fonte: Os Autores, 2018.

Na Figura 36 o protótipo de prótese está sobre a base de MDF com todo o módulo de controle integrado. Sua remoção e instalação sobre a base é feita com quatro parafusos, não tendo sido feita nenhuma interferência permanente no protótipo.

Em posição de descanso, ou desligado, a mão permanece aberta, pois são utilizados fios 100% poliuretano, que são elásticos e altamente resistentes para

manter os dedos abertos, enquanto o movimento de fechar dos dedos é realizado por fios não elásticos fixados aos servos-motores.

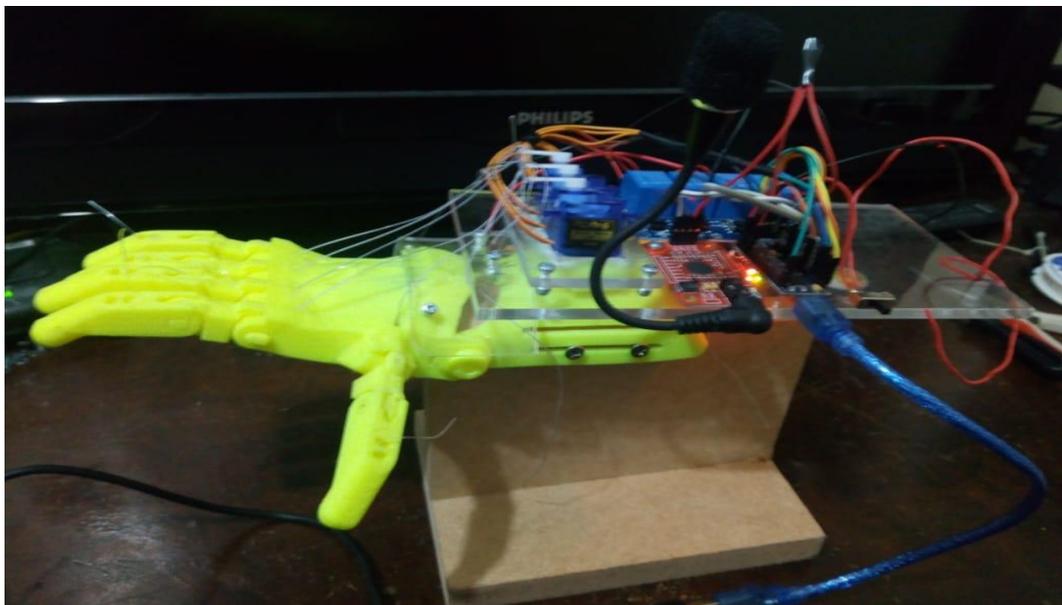


Figura 36 – Prótese com módulo de controle montado sobre base em MDF.

Fonte: Os Autores, 2018.

4 PROGRAMAÇÃO DO MÓDULO DE CONTROLE

A programação do módulo de controle consiste em dois arquivos distintos. O primeiro dos arquivos é um código para a gravação dos vocábulos no módulo de reconhecimento de voz, sendo que este arquivo, denominado “*vr_sample_train*” tem 1151 linhas de código e faz parte de uma biblioteca, a qual está disponível para baixar no endereço eletrônico da Usina Info, que é uma empresa nacional que comercializa através da *internet* componentes eletrônicos. Nesse site também são disponibilizados alguns artigos, como o “Módulo de reconhecimento de voz e comando para Arduino”, no qual se encontra o endereço para realizar o *download* desta biblioteca. Este código, “*vr_sample_train*”, foi testado e seu funcionamento está correto, sendo que o *link* está disponível nas referências e o código está na seção de Anexos (ANEXO B – CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ “VR_SAMPLE_TRAIN”). As instruções de como utilizar este código para gravar os comandos está no capítulo 4.1 CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ (USINA INFO, 2018).

Após o carregamento dos arquivos de áudio para o módulo de reconhecimento de voz, é necessário carregar o segundo arquivo, o qual faz o reconhecimento dos termos falados pelo locutor, comparando-os com os áudios gravados, e executando os movimentos referentes a cada comando, movimentando cada motor para a posição desejada. Este arquivo foi desenvolvido pelos autores deste trabalho e o código deste se encontra, já com as linhas comentadas, em Anexos (ANEXO C – CÓDIGO PARA CONTROLE). As rotinas essenciais para seu funcionamento se encontram explicadas no Capítulo 4.2 CÓDIGO PARA CONTROLE DOS SERVOMOTORES.

4.1 CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ

Ao baixar o arquivo “*vr_sample_train*” ou copiar seu código da seção ANEXO B para a pasta Arduino de seu disco local, abra o programa Arduino e em seguida navegue até “Arquivo->Exemplos->Módulo de Reconhecimento de Voz->*vr_sample_train*”. Depois de carregada a Biblioteca, no canto inferior direito do Monitor Serial defina Nova-linha e 115200 velocidade, assim como na Figura 37.

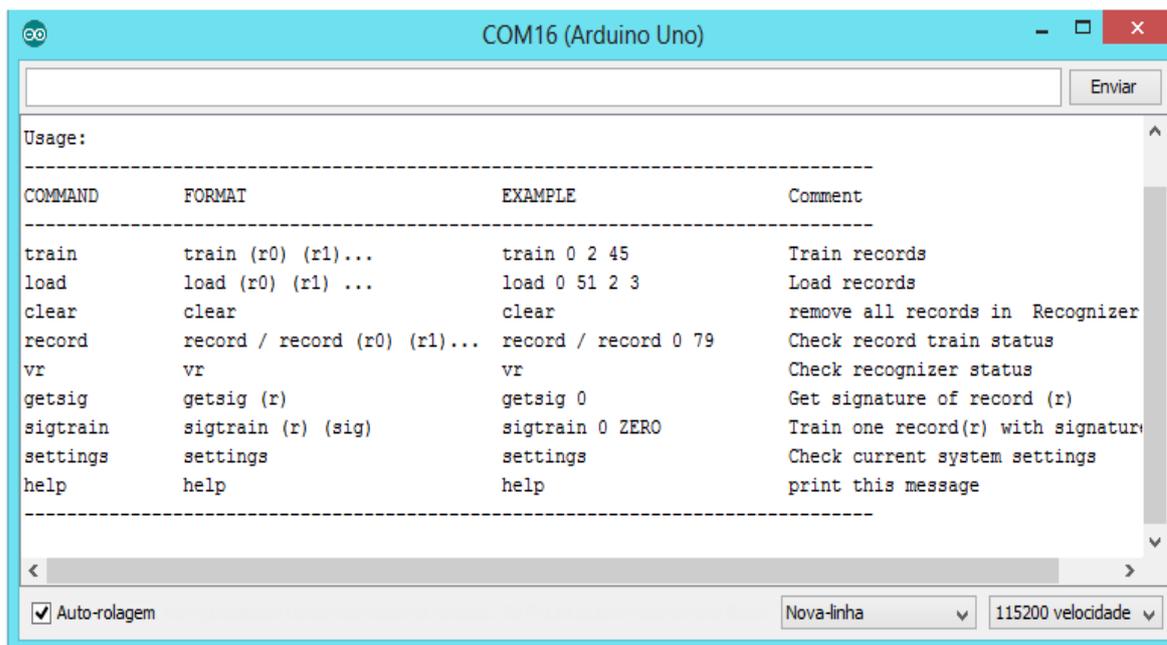


Figura 37 - Biblioteca vr_sample_train.

Fonte: UsinaInfo, 2018.

Isto feito, uma lista de comandos é apresentada em seu monitor, cada comando possui uma função específica. Esses comandos são:

- Comando “train (r0) (r1)...” – Grava interruptamente a sequencia estipulada de áudios.
- Comando “load (r0) (r1)...” – Carrega e anexa os áudios em sua biblioteca (7 de cada vez).
- Comando “clear” – Remove todos os áudios arquivados.
- Comando “record” – Verifica o estado treinado/não treinado.
- Comando “vr” – Verifica sequencia de áudios.
- Comando “getsing” –Obtém a assinatura de registro.
- Comando “sigtrain” – Realiza a gravação de um registro específico junto de sua assinatura.
- Comando “settings” – Verifica a configuração do módulo.
- Comando “help” – Imprime a tabela de comandos novamente.
- O comando “sigtrain r nome”, será utilizado para gravar os áudios pelo locutor, os quais serão reconhecidos posteriormente onde “r” é o número do registro utilizado, “sigtrain” o comando para gravar o áudio e “nome” é a assinatura do áudio gravado.

Os comandos gravados são:

- Sigtrain 0 abre
- Sigtrain 1 fecha
- Sigtrain 2 pinça (não será utilizado acento ou cedilha na programação, portanto o comando falado “pinça” será escrito como “pinca”)
- Sigtrain 3 palmar
- Sigtrain 4 testando
- Sigtrain 5 desligar

Sempre que um comando for válido, os mesmos irão aparecer no Monitor Serial, ao usar o comando de gravação *sigtrain*, por consequência o *Speak now* aparecerá. Neste momento deverá se falar a palavra que quer que seja gravada, em seguida o Monitor Serial mostrará a mensagem *Speak again*, no qual deve ser repetida a palavra dita anteriormente. Repetindo este processo até obter sucesso através da mensagem *Success* que irá aparecer na tela, conforme Figura 38.

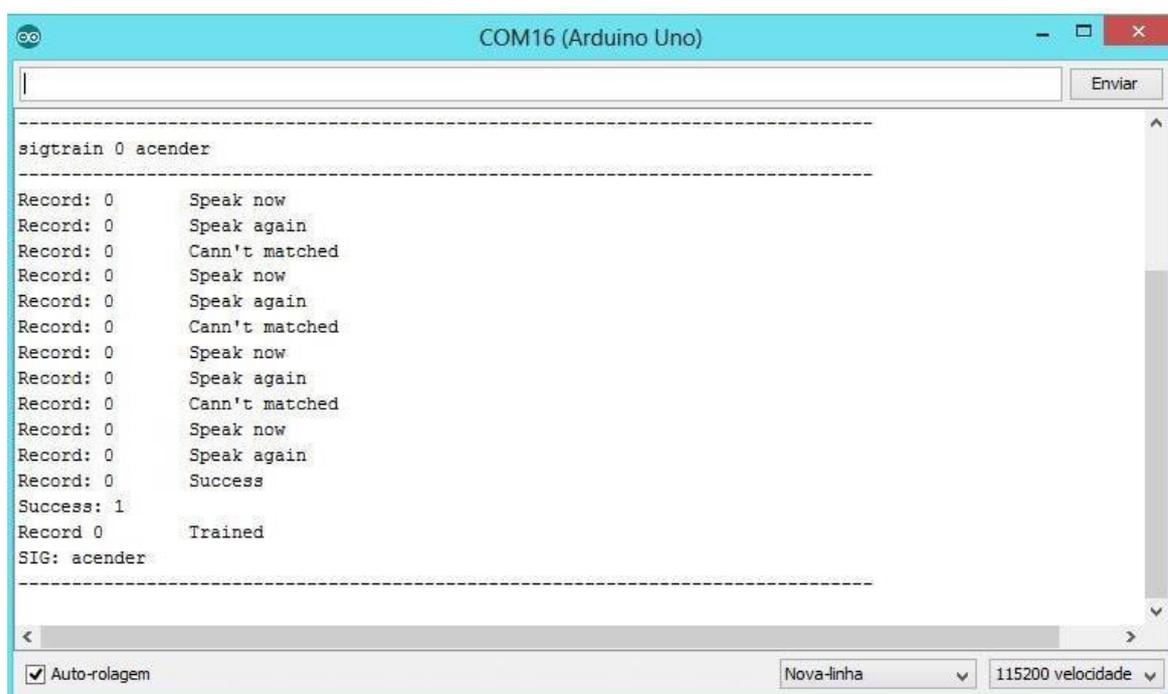


Figura 38 - Gravando comandos no módulo de voz.

Fonte: UsinalInfo, 2018.

Repetindo o mesmo processo com os registros 0, 1, 2, 3, 4 e 5, depois de todos os arquivos de áudios já gravados, é digitado o comando “load 0 1 2 3 4 5” para carregar os novos comandos de voz para o projeto.

Depois de feito isto e seguido todos os passos apresentados anteriormente, tem-se um projeto com simples conexão e fácil instalação. O Módulo de Reconhecimento de Voz pode ser utilizado para uma gama extremamente grande de aplicações, porém um fator importante que se percebe neste projeto é a alta sensibilidade que o produto apresenta, aceitando o comando de voz de um só indivíduo, no caso a pessoa que gravou.

4.2 CÓDIGO PARA CONTROLE DOS SERVOMOTORES

Com os comandos já gravados, é criado um novo *sketch* do Arduino e em seguida copiado o código abaixo dentro do arquivo, este fará todo o controle da prótese, de acordo com os comandos gravados. O arquivo denominado como “*code_TCC_final_comentado*” será responsável pelo reconhecimento dos comandos de voz treinados anteriormente e a realização do controle dos relés, que estão instalados juntos as portas 11, 12 e 13 e dos servos-motores que estão conectados nas portas 6, 9 e 10 do Arduino.

Para evitar o uso de muitos “*if*” no código, foi utilizada uma estrutura com o comando “*switch case*”, onde caso um comando de voz seja reconhecido o trecho de código específico é realizado, no exemplo abaixo, o comando “*abre*” liga os relés 1 à 3, movimenta os servo-motores para a posição inicial, observando que o motor 3 se movimenta no sentido anti-horário, por questão de espaço útil na montagem, então sua posição inicial é 180°, enquanto os demais iniciam na posição 20°, pois os motores tem uma limitação de controle de movimentação de 180°, sendo que, quando na posição 0° o motor fica rotacionando sem controle, para tal, foi usada a posição de 20° como inicial.

```
switch (buf[1]) {
  case abre:
    digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
    digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
    digitalWrite(rele3, HIGH); // Define estado do rele 2 como Ligado
    s1.write(20); // Movimenta servo1 para posição zero
    s2.write(20); // Movimenta servo2 para posição zero
    s3.write(180); // Movimenta servo3 para posição zero
```

Para garantir que os comandos sejam executados corretamente, cada um é iniciado com o trecho acima, pois não são conhecidos os estados atuais dos relés e as posições dos servos-motores, garantindo que os relés estejam ligados e que a posição dos motores esteja no ângulo inicial, os quais mantêm a prótese com os dedos abertos. Logo após são executados os comandos de movimentação específicos de cada comando. No exemplo abaixo, o servo-motor 1, que movimenta o polegar da prótese, vai da posição 20° até a posição de 90° em sentido horário, ficando semiaberto, com um *delay* de 5 milissegundos entre graus e ao final deste movimento, o segundo motor sai da sua posição inicial até a posição de 180°, também em sentido horário, levando o dedo indicador até a posição totalmente fechado.

```

for(pos = 20; pos < 90; pos++) // Movimenta servo1 em 70 graus horário
{
  s1.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
for(pos = 20; pos < 180; pos++) // Movimenta servo2 em 160 graus horário
{
  s2.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}

```

Sendo que para cada locução, movimentos distintos são executados pela prótese, além de outras linhas de programação que fazem parte da programação para o total funcionamento da mesma, o código completo se torna muito extenso para este capítulo, estando na íntegra na seção de Anexos.

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste tópico são apresentados os resultados dos testes aplicados ao protótipo com a eletrônica integrada e o orçamento final do projeto.

5.1 TESTES DO PROTÓTIPO INTEGRADO A ELETRÔNICA

Para a conferência da total funcionalidade conforme o planejado diversos testes foram executados, os quais consistiam em o autor da gravação do módulo de voz da eletrônica do projeto falar os comandos gravados e aguardar o acionamento do módulo Arduino para o mesmo enviar os sinais para os servos-motores e por fim, esses realizarem a movimentação dos dedos do protótipo.

5.1.1 COMANDO ABRE E FECHA

A posição do protótipo aguardada ao comando “ABRE” ser enviado ao módulo de voz, é apresentada na Figura 39. E a posição do comando “FECHA” é apresentada na Figura 40.

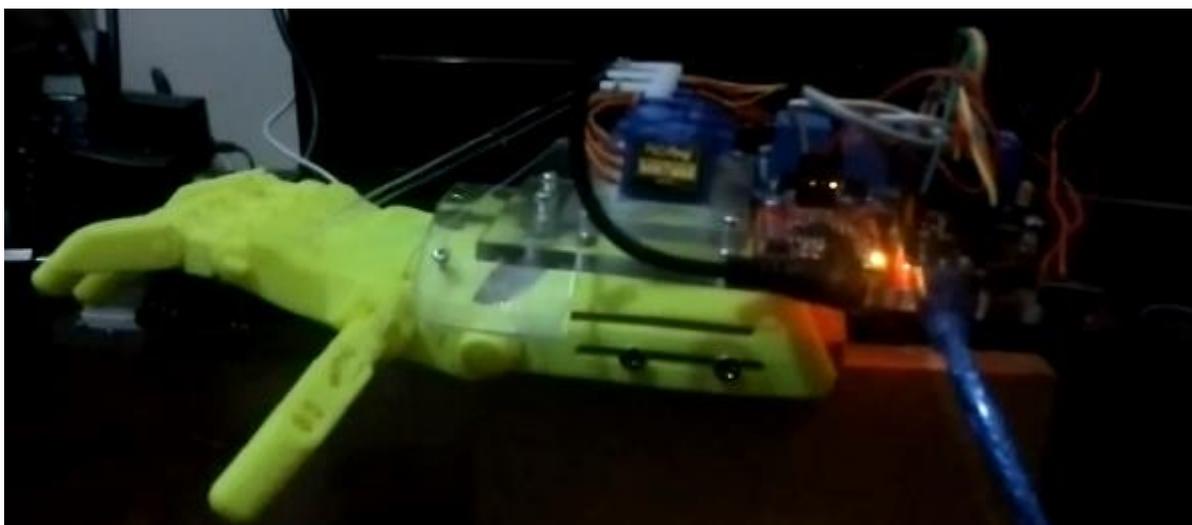


Figura 39 - Comando ABRE.

Fonte: Os Autores, 2018.



Figura 40 - Comando FECHA.

Fonte: Os Autores, 2018.

O comando “ABRE” posiciona todos os motores na “posição zero”, isto significa que nenhum dos motores é acionado e o protótipo mantém esta posição através dos fios de silicone 100% poliuretano.

O comando “FECHA” aciona todos os motores, contudo, o motor acoplado ao polegar realiza um leve movimento de 70° e os outros dois motores (um acoplado ao dedo indicador e outro acoplado aos dedos médio, anelar e mínimo), realizam movimentos de fechamento, os quais são de 160° e 150° , respectivamente. Assim que este movimento é completado, o servo-motor do polegar volta a se movimentar em mais 70° no mesmo sentido que o movimento anterior, fechando o polegar sobre o dedo indicador. O protótipo realiza o posicionamento apresentado anteriormente na Figura 40.

5.1.2 COMANDO PINÇA

A posição do protótipo ao comando “PINÇA” é apresentada na Figura 41.



Figura 41 - Comando PINÇA.

Fonte: Os Autores, 2018.

O comando “PINÇA” aciona os servos-motores do dedo indicadores e polegar do protótipo. O movimento de pinça é realizado pelos dois dedos simultaneamente, sendo o fechamento de 140°. Enquanto isto, os outros dedos são mantidos na posição de descanso, ou seja, abertos. Este posicionamento proporciona segurança para segurar, por exemplo, uma folha de papel, um cartão ou um convite por exemplo.

5.1.3 COMANDO PALMAR

A posição do protótipo ao comando “PALMAR” é apresentada na Figura 42.



Figura 42 - Comando PALMAR.

Fonte: Os Autores, 2018.

O comando “PALMAR” aciona os três servo-motores do protótipo. Em um primeiro momento, o servo-motor do polegar é acionado e se move em 130° . Logo em seguida, o motor do dedo indicador também em 130° . Por fim, o último servo motor aciona os dedos médio, anelar e mínimo em 140° , finalizando assim o movimento. Este movimento palmar é firme e proporciona segurança para segurar objetos maiores, como um estilete ou talheres.

5.1.4 COMANDO DESLIGAR

O comando “DESLIGAR” retorna os dedos à posição de descanso e desliga os servos-motores, abrindo os contatos dos relés, para economia de bateria, ou simplesmente para poder manipular os servos sem a necessidade de desligar o Arduino.

5.1.5 COMANDO TESTANDO

Foi criado um comando “TESTANDO”, o qual realiza todos os movimentos da prótese, abrindo, fechando, pinçando, realizando o movimento palmar e desligando a

prótese. Este comando tem a função de testar os movimentos, não sendo um movimento útil no dia-a-dia do usuário, mas a sua substituição é possível, lembrando que é possível gravar até sete comandos de voz na prótese.

5.2 ORÇAMENTO

Para a realização deste trabalho, foram investidos alguns valores referentes à criação do protótipo da prótese e deslocamento dos autores. Estes dados são apresentados na Tabela 3.

Tabela 3 - Orçamento

Descrição	Qtd.	Valor Unitário (R\$)	Valor Total (R\$)
Impressão da Prótese Raptor Reloaded	1	350,00	350,00
Microcontrolador Arduino Nano	1	40,00	40,00
Módulo de reconhecimento de voz	1	150,00	150,00
Módulo Relé 5V 10A de 4 canais	1	30,00	30,00
Micro Servo Motor 9g SG90	4	19,00	76,00
Chapas de acrílico	2	10,00	20,00
Chapa de MDF	1	10,00	10,00
Xerox de documentos técnicos/livros	25	0,12	3,00
Baterias 9V Recarregáveis	3	23,00	69,00
Kit de Jumpers Fêmea/Macho	1	12,00	12,00
Deslocamentos (Uber)	6	25,00	150,00
Deslocamentos (Ônibus)	20	4,25	85,00
Rolo de Elásticos / Fios	2	10,00	20,00
TOTAL (R\$)			1.015,00

Fonte: Os Autores, 2018.

6 CONSIDERAÇÕES FINAIS

Reconhecendo a falta de próteses com controle eletrônico e valor acessível no mercado, nasceu este projeto, inicialmente um protótipo, de uma prótese totalmente criada a partir da cultura *Maker* e de projetos *open source*.

O objetivo principal foi atingir o baixo custo de produção para que a classe brasileira menos favorecida fosse atingida, oferecendo uma oportunidade de mudança na rotina e por que não, de vida. Após a execução deste projeto, de Junho de 2017 a Junho de 2018, notou-se que é possível criar uma prótese através da manufatura aditiva, com baixo peso (se comparada a próteses comerciais), alta resistência, de custo acessível. Aliada a esta, foram integrados módulos eletrônicos, incluindo um comando de voz, o qual proporcionou ao protótipo total funcionalidade através de um código implantado no microcontrolador Arduino, que comanda os servos-motores, os quais recebem sinais elétricos e através do movimento das aletas acopladas a eles fazem com que os dedos do protótipo executem movimentos típicos e próximos dos reais para o usuário final.

O estudo proposto sobre a mão humana e seus tipos de pegada foi importante para a conclusão deste trabalho, visto que o mesmo foi utilizado como base para a escolha dos movimentos executados pelo protótipo. Além deste estudo, foram levantados vários valores dos módulos propostos para a execução do protótipo e isto possibilitou a melhor escolha para que este projeto fosse concluído com um custo acessível. Para isto ser viabilizado, optou-se também pelo uso de um protótipo de prótese produzido através da manufatura aditiva que mostrou ser uma alternativa econômica, resistente e também inovadora.

A relevância do projeto pode ser concluída através da pesquisa *online* realizada e também, através de dados divulgados pelo IBGE e ANS. Entre outros portais de pesquisa e apresentação de projetos com a mesma finalidade, entretanto nenhum apresentava uma prótese com eletrônica embarcada e comandos de voz como o proposto neste projeto, o que mostrou ser uma ideia inovadora e que na visão dos autores, proporcionaria um novo estilo de vida para possíveis usuários.

Por fim, os testes com a plataforma de MDF, o protótipo da prótese e a eletrônica se apresentaram consistentes e abre um novo leque de pesquisas e projetos futuros, nos quais poderão ser pensadas formas de transformar este protótipo

em uma prótese que realmente poderá ser utilizada por pessoas com deficiência físico-motora.

REFERÊNCIAS

3DECIA, 2017. **A história da impressão 3D - de sonho a realidade**. 2017. Disponível em: <<https://www.3decia.com.br/historia>>. Acesso em 03 Jan. 2018.

Agência Nacional De Saúde Suplementar, 2016. **Relatório final do grupo de trabalho externo de órteses, próteses e materiais especiais (GTE OPME) ANS/ANVISA**. 2016. Disponível em: <http://www.ans.gov.br/images/stories/Participacao_da_sociedade/2016_gt_opme/gt-opme-relatoriointegral.pdf>. Acesso em 20 Dez 2017.

ASZMANN Oskar C, Bionic reconstruction to restore hand function after brachial plexus injury: a case series of three patients, **The Lancet**. 24 Fev. 2015. Disponível em: <[http://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(14\)61776-1/fulltext](http://www.thelancet.com/journals/lancet/article/PIIS0140-6736(14)61776-1/fulltext)>. Acesso em 01 Mar. 2018

CAMMADA, 2018. **Materiais para impressão 3D - Entenda como são os materiais para impressão 3D**. 2018. Disponível em: <<http://cammada.com/materiais>>. Acesso em 01 Mar. 2018.

CARDOSO, Eluana Sievert; BEZERRA, Matheus Herrera, 2015. **Candidatos a opinar/declarar suas experiências com próteses**. 01 Jun. 2015. Disponível em: <<https://www.survio.com/survey/d/A2T5E8Q1E3L9H9I9W>>. Acesso em 10 Mar. 2018.

CARVALHO, J. A. **Amputações de membros inferiores: em busca da plena reabilitação**. Manole. 2ª ed. São Paulo. 2003. p.363.

COLLOR, Fernando, 1990. **Lei nº 8.080**. 19 Set. 1990. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/L8080.htm>. Acesso em 18 de Jun. 2018.

DEURSEN, van Felipe. **A revolução das impressoras 3D**. 31 Out. 2016. Disponível em: <<https://super.abril.com.br/tecnologia/a-revolucao-das-impressoras-3d/>>. Acesso em 03 Mar. 2017.

DIEESE, 2018. **Salário mínimo nominal e necessário**. 01 Jan. 2018. Disponível em: <<https://www.dieese.org.br/analisecestabasica/salarioMinimo.html>> Acesso em: 23 Fev. 2018.

ELETRÔNICA PARA ARTISTAS. **Arduino 2: ConFiguração do Arduino Nano**. Disponível em: <<http://eletronicaparaartistas.com.br/arduino-2-conFiguracao-do-arduino-nano/>>. Acesso em 30 Nov 2017.

E-NABLE BRASIL. **E-nable Brasil - Habilitando o Futuro**. Disponível em: <<http://e-nablebrasil.org/>>. Acesso em 20 Dez 2017.

E-NABLING THE FUTURE. **Wich Design?**. Disponível em:<<http://enablingthefuture.org/which-design/>> . Acesso em 20 Dez 2017.

E-NABLE COMMUNITY. **A Global Network Of Passionate Volunteers Using 3D Printing To Give The World A "Helping Hand."** Disponível em:<<http://enablingthefuture.org/>>. Acesso em 20 Dez 2017.

EXIII-HACKBERRY. **3D printable open-source bionic arm.** Disponível em:<<http://exiii-hackberry.com/>>. Acesso em 20 Dez 2017.

FILIFELOP. **Micro Servo 9g SG90 TowerPro.** Disponível em:<<https://www.filipeflop.com/blog/micro-servo-motor-9g-sg90-com-arduino-uno/>>. Acesso em 25 Jan 2018.

FILIFELOP. **Módulo Relé 5V 4 Canais.** Disponível em:<<https://www.filipeflop.com/produto/modulo-rele-5v-4-canais/>>. Acesso em 25 Jan 2018.

GARCIA, Gabriel. **Três pessoas ganham implantes de próteses de mãos na Áustria.** 25 Fev. 2015. Disponível em:<<https://exame.abril.com.br/ciencia/tres-pessoas-ganham-implantes-de-protese-de-maos-na-austria/>>. Acesso em: 14 Fev. 2018.

LIMA, Mariana. **Ossos do membro Superior. Anatomia Humana.** Disponível em:<<http://mundoeducacao.bol.uol.com.br/biologia/ossos-membro-superior.htm>>. Acesso em: 23 mar. 2015.

LOPES, José Marcos. **Nova mão biônica permite mexer os 5 dedos.** Disponível em:<<http://www.gazetadopovo.com.br/vida-e-cidadania/nova-mao-bionica-permite-mexer-os-5-dedos-bytcg678vw18buag16riskjf2>>. Acesso em: 24 Fev. 2018

KAPANDJI, Albert. **The Physiology of the Joints** – 2nd ed. – British Library Cataloguing in Publication Data. 18 Out. 1973.

KUNKEL, Maria Elizete. Disponível em:<<https://mao3d.wordpress.com/2016/07/27/mao3d-programa-social-e-colaborativo-da-unifesp/>>. Acesso em 21 Abr. 2017.

_____. Disponível em:<<https://mao3d.wordpress.com/2016/07/30/mao3d-nem-sempre-e-possivel-fazer-a-protetizacao-infantil/>>. Acesso em 21 Abr. 2017.

HUNOLD, Murilo; LIMA, Vivian.; CASTRO, Yves.; AUGUSTO, Carlos; YOSHIMURA, Felipe; MATIAS, Jean; RODRIGUES, Manuela; KUNKEL, Maria. **Protótipo de prótese de mão robótica de lego controlada por sistema Android para bi-amputado.** XXIV Congresso Brasileiro de Engenharia Biomédica – CBEB, 2014. Disponível

em:<http://www.canal6.com.br/cbeb/2014/artigos/cbeb2014_submission_714.pdf>. Acesso em 10 Fev. 2018.

MASCARENHAS, Rafael. **Tecnologia de impressão 3D FDM**. CADguru, 23 Jun. 2016. Disponível em:< <http://cad.cursosguru.com.br/novidades/tecnologia-de-impressao-3d-fdm/>>. Acesso em 05 Abr. 2017

MOORE, Keith; DALLEY, Arthur; AGUR, Anne. **Clinically Oriented Anatomy**, 7ª edição, Lippincott Williams & Wilkins, 2014.

NEVES, Heloisa. **Movimento Maker: Faça você mesmo**. 29 Set. 2016. Disponível em:<<https://dialogando.com.br/movimento-maker-faca-voce-mesmo/>>. Acesso em 26 Fev. 2018.

NORTON, Kim M. **A Brief History of Prosthetics**. 20 Nov. 2007. Disponível em:<<https://www.amputee-coalition.org/resources/a-brief-history-of-prosthetics/>>. Acesso em 19 Fev. 2018.

NICOLELIS, Miguel. **A utopia é científica**. Julho 2009. Disponível em:<<http://www.redebrasilatual.com.br/revistas/37/a-utopia-e-cientifica>> Acesso em 07 Mar. 2018

ORTOLAN, Rodrigo Lício. **Estudo e avaliação de técnicas de processamento do sinal mioelétrico para o controle de sistemas de reabilitação**. 2002. Disponível em:<<http://www.teses.usp.br/teses/disponiveis/18/18133/tde-19112002-153337/pt-br.php>>. Acesso em 18 Jun. 2018.

OSHW. **Open Source Hardware**. 29 Dez. 2015. Disponível em:<<http://freedomdefined.org/OSHW/translations/pt>>. Acesso em 03 Jan. 2018.

PASCHOARELLI, LC., and MENEZES, MS., orgs. **Design e ergonomia: aspectos tecnológicos [online]**. São Paulo: Editora UNESP; São Paulo: Cultura Acadêmica, 2009. 279 p. ISBN 978-85-7983-001-3. Disponível em:<<http://books.scielo.org/id/yjxnr/pdf/paschoarelli-9788579830013-05.pdf>> Acesso em 09 Mar. 2018.

PEREIRA, Ana Paula. **Como funciona o reconhecimento de voz?**. 24 Nov. 2009. Disponível em:<<https://www.tecmundo.com.br/curiosidade/3144-como-funciona-o-reconhecimento-de-voz-.htm>> Acessado em 24 fev. 2018.

SANCTUARY, Hillary. **Amputee Feels in Real-Time with Bionic Hand**. 05 Fev. 2014. Disponível em:<<https://actu.epfl.ch/news/amputee-feels-in-real-time-with-bionic-hand/>>. Acessado em 18 Jun. 2018.

STRAUB, Matheus Gebert. **Módulo De Reconhecimento De Voz E Comando Para Arduino**. 04 Dez. 2015. Disponível em:<<http://blog.usinainfo.com.br/modulo-de-reconhecimento-de-voz-para-arduino/>> Acessado em 24 fev. 2018.

STRAUB, Matheus Gebert. **Download da Biblioteca de Reconhecimento de Voz.**
04 Dez. 2015. Disponível em:
<http://www.usinainfo.com.br/index.php?controller=attachment&id_attachment=248>
Acessado em 24 fev. 2018.

WEIR, R. **Design of artificial arms and hands for prosthetic applications.** In:
Standard Handbook of Biomedical Engineering and Design. Nova York: McGraw-
Hill, 2004.

ANEXO(S)

ANEXO A – Arquivos stl da prótese Raptor Reloaded

ANEXO B – Código para gravação dos comandos de voz
“vr_sample_train”

ANEXO C – Código para controle dos servo-motores

ANEXO A – ARQUIVOS STL DA PRÓTESE RAPTOR RELOADED

Seguem abaixo os nomes de cada peça, juntamente com o link direto para o download dos seus respectivos arquivos em formato stl. Bem como o endereço para a página instructables, onde se encontra um tutorial detalhado de como montar a prótese:

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Palma esquerda.
Disponível em: <<https://www.thingiverse.com/download:959712>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Palma direita.
Disponível em: <<https://www.thingiverse.com/download:959713>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Pino de junta.
Disponível em: <<https://www.thingiverse.com/download:959701>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Pino de polegar.
Disponível em: <<https://www.thingiverse.com/download:959705>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Falange proximal.
Disponível em: <<https://www.thingiverse.com/download:959900>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Pino de dedo.
Disponível em: <<https://www.thingiverse.com/download:999964>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Falange distal. Disponível em: <<https://www.thingiverse.com/download:970024>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Luva. Disponível em: <<https://www.thingiverse.com/download:959709>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Pino de pulso. Disponível em: <<https://www.thingiverse.com/download:959704>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Capa do pino do pulso. Disponível em: <<https://www.thingiverse.com/download:959706>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Tensionador. Disponível em: <<https://www.thingiverse.com/download:959702>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Clip de retenção. Disponível em: <<https://www.thingiverse.com/download:959703>>. Acesso em 22 Mai 2018.

ENABLE COMMUNITY, Raptor Reloaded by e-NABLE - Pino tensionador. Disponível em: <<https://www.thingiverse.com/download:959708>>. Acesso em 22 Mai 2018.

BASTIAN, Andreas. Assembly of E-Nable's Raptor Reloaded Hand. Disponível em: <<http://www.instructables.com/id/Assembly-of-e-NABLEs-Raptor-Reloaded-Hand/>>. Acesso em 22 Mai 2018.

ANEXO B – CÓDIGO PARA GRAVAÇÃO DOS COMANDOS DE VOZ “VR_SAMPLE_TRAIN”

```

/**
*****
* @file vr_sample_train.ino
* @author JiapengLi
* @brief This file provides a demonstration on
* how to train VoiceRecognitionModule to record your voice
*****
* @note:
* Use serial command to control VoiceRecognitionModule. '
* All commands are case insensitive. Default serial baud rate 115200.
*
* COMMAND      FORMAT          EXAMPLE      Comment
*
* train        train (r0) (r1)...  train 0 2 45  Train records
* load         load (r0) (r1) ...  load 0 51 2 3  Load records
* clear        clear                clear          remove all records in Recognizer
* record       record (r0) (r1)... record 0 79     Check record train status
* vr           vr                    vr             Check recognizer status
* getsig       getsig (r)            getsig 0       Get signature of record (r)
* sigtrain     sigtrain (r) (sig)  sigtrain 0 ZERO Train one record(r) with signature(sig)
* settings     settings            settings       Check current system settings
*****
* @section HISTORY
*
* 2013/06/13 Initial version.
*/
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
/**
* Connection
* Arduino VoiceRecognitionModule
* 2 -----> TX
* 3 -----> RX
*/
VR myVR(2,3); // 2:RX 3:TX, you can choose your favourite pins.
/*****/
/** declare print functions */
void printSeperator();
void printSignature(uint8_t *buf, int len);
void printVR(uint8_t *buf);
void printLoad(uint8_t *buf, uint8_t len);
void printTrain(uint8_t *buf, uint8_t len);
void printCheckRecognizer(uint8_t *buf);
void printUserGroup(uint8_t *buf, int len);
void printCheckRecord(uint8_t *buf, int num);
void printCheckRecordAll(uint8_t *buf, int num);
void printSigTrain(uint8_t *buf, uint8_t len);
void printSystemSettings(uint8_t *buf, int len);
void printHelp(void);
/*****/
// command analyze part
#define CMD_BUF_LEN 64+1
#define CMD_NUM 10
typedef int (*cmd_function_t)(int, int);
uint8_t cmd[CMD_BUF_LEN];

```

```

uint8_t cmd_cnt;
uint8_t *paraAddr;
int receiveCMD();
int checkCMD(int len);
int checkParaNum(int len);
int findPara(int len, int paraNum, uint8_t **addr);
int compareCMD(uint8_t *para1 , uint8_t *para2, int len);
int cmdTrain(int len, int paraNum);
int cmdLoad(int len, int paraNum);
int cmdTest(int len, int paraNum);
int cmdVR(int len, int paraNum);
int cmdClear(int len, int paraNum);
int cmdRecord(int len, int paraNum);
int cmdSigTrain(int len, int paraNum);
int cmdGetSig(int len, int paraNum);
int cmdSettings(int len, int paraNum);
int cmdHelp(int len, int paraNum);
/** cmdList, cmdLen, cmdFunction has correspondence */
const char cmdList[CMD_NUM][10] = { // command list table
    {
        "train" }
    ,
    {
        "load" }
    ,
    {
        "clear" }
    ,
    {
        "vr" }
    ,
    {
        "record" }
    ,
    {
        "sigtrain" }
    ,
    {
        "getsig" }
    ,
    {
        "Settings" }
    ,
    {
        "test" }
    ,
    {
        "help" }
    ,
};
const char cmdLen[CMD_NUM]= { // command length
    5, // {"train"},
    4, // {"load"},
    5, // {"clear"},
    2, // {"vr"},
    6, // {"record"},
    8, // {"sigtrain"},
    6, // {"getsig"},
    8, // {"Settings"},
    4, // {"test"},

```

```

4, // {"help"}
};
cmd_function_t cmdFunction[CMD_NUM]={ // command handle fuction(function pointer table)
  cmdTrain,
  cmdLoad,
  cmdClear,
  cmdVR,
  cmdRecord,
  cmdSigTrain,
  cmdGetSig,
  cmdSettings,
  cmdTest,
  cmdHelp,
};

/*****
/** temprory data */
uint8_t buf[255];
uint8_t records[7]; // save record

void setup(void)
{
  myVR.begin(9600);
  /** initialize */
  Serial.begin(115200);
  Serial.println(F("Elechouse Voice Recognition V3 Module \"train\" sample."));
  printSeperator();
  Serial.println(F("Usage:"));
  printSeperator();
  printHelp();
  printSeperator();
  cmd_cnt = 0;
}

void loop(void)
{
  int len, paraNum, paraLen, i;
  /** receive Serial command */
  len = receiveCMD();
  if(len>0){
    /** check if the received command is valid */
    if(!checkCMD(len)){
      /** check parameter number of the received command */
      paraNum = checkParaNum(len);
      /** display the received command back */
      Serial.write(cmd, len);
      /** find the first parameter */
      paraLen = findPara(len, 1, &paraAddr);
      /** compare the received command with command in the list */
      for(i=0; i<CMD_NUM; i++){
        /** compare command length */
        if(paraLen == cmdLen[i]){
          /** compare command content */
          if( compareCMD(paraAddr, (uint8_t *)cmdList[i], paraLen) == 0 ){
            /** call command function */
            if( cmdFunction[i](len, paraNum) != 0){
              printSeperator();
              Serial.println(F("Command Format Error!"));
              printSeperator();
            }
          }
        }
      }
    }
  }
}

```

```

        break;
    }
}
}
/** command is not supported*/
if(i == CMD_NUM){
    printSeperator();
    Serial.println(F("Unkonwn command"));
    printSeperator();
}
}
else{
    /** received command is invalid */
    printSeperator();
    Serial.println(F("Command format error"));
    printSeperator();
}
}
/** try to receive recognize result */
int ret;
ret = myVR.recognize(buf, 50);
if(ret>0){
    /** voice recognized, print result */
    printVR(buf);
}
}
/**
 * @brief receive command from Serial.
 * @param NONE.
 * @retval command length, if no command receive return -1.
 */
int receiveCMD()
{
    int ret;
    int len;
    unsigned long start_millis;
    start_millis = millis();
    while(1){
        ret = Serial.read();
        if(ret>0){
            start_millis = millis();
            cmd[cmd_cnt] = ret;
            if(cmd[cmd_cnt] == '\n'){
                len = cmd_cnt+1;
                cmd_cnt = 0;
                return len;
            }
            cmd_cnt++;
            if(cmd_cnt == CMD_BUF_LEN){
                cmd_cnt = 0;
                return -1;
            }
        }
        if(millis() - start_millis > 100){
            cmd_cnt = 0;
            return -1;
        }
    }
}
}
/**

```

```

* @brief compare two commands, case insensitive.
* @param para1 --> command buffer 1
* para2 --> command buffer 2
* len --> buffer length
* @retval 0 --> equal
* -1 --> unequal
*/
int compareCMD(uint8_t *para1 , uint8_t *para2, int len)
{
    int i;
    uint8_t res;
    for(i=0; i<len; i++){
        res = para2[i] - para1[i];
        if(res != 0 && res != 0x20){
            res = para1[i] - para2[i];
            if(res != 0 && res != 0x20){
                return -1;
            }
        }
    }
    return 0;
}
/**
* @brief Check command format.
* @param len --> command length
* @retval 0 --> command is valid
* -1 --> command is invalid
*/
int checkCMD(int len)
{
    int i;
    for(i=0; i<len; i++){
        if(cmd[i] > 0x1F && cmd[i] < 0x7F){
        }
        else if(cmd[i] == '\t' || cmd[i] == ' ' || cmd[i] == '\r' || cmd[i] == '\n'){
        }
        else{
            return -1;
        }
    }
    return 0;
}
/**
* @brief Check the number of parameters in the command
* @param len --> command length
* @retval number of parameters
*/
int checkParaNum(int len)
{
    int cnt=0, i;
    for(i=0; i<len; ){
        if(cmd[i]!='\t' && cmd[i]!=' ' && cmd[i] != '\r' && cmd[i] != '\n'){
            cnt++;
            while(cmd[i] != '\t' && cmd[i] != ' ' && cmd[i] != '\r' && cmd[i] != '\n'){
                i++;
            }
        }
        i++;
    }
    return cnt;
}

```

```

}
/**
 * @brief Find the specified parameter.
 * @param len --> command length
 * @param paraIndex --> parameter index
 * @param addr --> return value. position of the parameter
 * @param retval length of specified parameter
 */
int findPara(int len, int paraIndex, uint8_t **addr)
{
    int cnt=0, i, paraLen;
    uint8_t dt;
    for(i=0; i<len; ){
        dt = cmd[i];
        if(dt!='\t' && dt!=' '){
            cnt++;
            if(paraIndex == cnt){
                *addr = cmd+i;
                paraLen = 0;
                while(cmd[i] != '\t' && cmd[i] != ' ' && cmd[i] != '\r' && cmd[i] != '\n'){
                    i++;
                    paraLen++;
                }
                return paraLen;
            }
        }
        else{
            while(cmd[i] != '\t' && cmd[i] != ' ' && cmd[i] != '\r' && cmd[i] != '\n'){
                i++;
            }
        }
    }
    return -1;
}

int cmdHelp(int len, int paraNum)
{
    if(paraNum != 1){
        return -1;
    }
    printSeperator();
    printHelp();
    printSeperator();
    return 0;
}

/**
 * @brief Handle "train" command
 * @param len --> command length
 * @param paraNum --> number of parameters
 * @param retval 0 --> success
 * @param retval -1 --> Command format error
 */
int cmdTrain(int len, int paraNum)
{
    int i, ret;
    if(paraNum < 2 || paraNum > 8){
        return -1;
    }
}

```

```

for(i=2; i<=paraNum; i++){
    findPara(len, i, &paraAddr);
    records[i-2] = atoi((char *)paraAddr);
    if(records[i-2] == 0 && *paraAddr != '0'){
        return -1;
    }
}
printSeperator();
ret = myVR.train(records, paraNum-1, buf);
// ret = myVR.train(records, paraNum-1);
if(ret >= 0){
    printTrain(buf, ret);
}
else if(ret == -1){
    Serial.println(F("Train failed."));
}
else if(ret == -2){
    Serial.println(F("Train Timeout."));
}
printSeperator();
return 0;
}
/**
 * @brief Handle "load" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdLoad(int len, int paraNum)
{
    int i, ret;
    if(paraNum < 2 || paraNum > 8){
        return -1;
    }
    for(i=2; i<=paraNum; i++){
        findPara(len, i, &paraAddr);
        records[i-2] = atoi((char *)paraAddr);
        if(records[i-2] == 0 && *paraAddr != '0'){
            return -1;
        }
    }
    // myVR.writehex(records, paraNum-1);
    ret = myVR.load(records, paraNum-1, buf);
    printSeperator();
    if(ret >= 0){
        printLoad(buf, ret);
    }
    else{
        Serial.println(F("Load failed or timeout."));
    }
    printSeperator();
    return 0;
}
/**
 * @brief Handle "clear" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error

```

```

*/
int cmdClear(int len, int paraNum)
{
    if(paraNum != 1){
        return -1;
    }
    if(myVR.clear() == 0){
        printSeperator();
        Serial.println(F("Recognizer cleared."));
        printSeperator();
    }
    else{
        printSeperator();
        Serial.println(F("Clear recognizer failed or timeout."));
        printSeperator();
    }
    return 0;
}
/**
 * @brief Handle "vr" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdVR(int len, int paraNum)
{
    int ret;
    if(paraNum != 1){
        return -1;
    }
    ret = myVR.checkRecognizer(buf);
    if(ret<=0){
        printSeperator();
        Serial.println(F("Check recognizer failed or timeout."));
        printSeperator();
        return 0;
    }
    printSeperator();
    printCheckRecognizer(buf);
    printSeperator();
    return 0;
}
/**
 * @brief Handle "record" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdRecord(int len, int paraNum)
{
    int ret;
    if(paraNum == 1){
        ret = myVR.checkRecord(buf);
        printSeperator();
        if(ret>=0){
            printCheckRecordAll(buf, ret);
        }
    }
    else{

```

```

    Serial.println(F("Check record failed or timeout.));
}
printSeperator();
}
else if(paraNum < 9){
for(int i=2; i<=paraNum; i++){
    findPara(len, i, &paraAddr);
    records[i-2] = atoi((char *)paraAddr);
    if(records[i-2] == 0 && *paraAddr != '0'){
        return -1;
    }
}
ret = myVR.checkRecord(buf, records, paraNum-1); // auto clean duplicate records
printSeperator();
if(ret>=0){
    printCheckRecord(buf, ret);
}
else{
    Serial.println(F("Check record failed or timeout.));
}
printSeperator();
}
else{
    return -1;
}
return 0;
}
/**
 * @brief Handle "sigtrain" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdSigTrain(int len, int paraNum)
{
    int ret, sig_len;
    uint8_t *lastAddr;
    if(paraNum < 2){
        return -1;
    }
    findPara(len, 2, &paraAddr);
    records[0] = atoi((char *)paraAddr);
    if(records[0] == 0 && *paraAddr != '0'){
        return -1;
    }
    findPara(len, 3, &paraAddr);
    sig_len = findPara(len, paraNum, &lastAddr);
    sig_len +=( (unsigned int)lastAddr - (unsigned int)paraAddr );
    printSeperator();
    ret = myVR.trainWithSignature(records[0], paraAddr, sig_len, buf);
    // ret = myVR.trainWithSignature(records, paraNum-1);
    if(ret >= 0){
        printSigTrain(buf, ret);
    }
    else{
        Serial.println(F("Train with signature failed or timeout.));
    }
    printSeperator();
    return 0;
}

```

```

}
/**
 * @brief Handle "getsig" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdGetSig(int len, int paraNum)
{
    int ret;
    if(paraNum != 2){
        return -1;
    }
    findPara(len, 2, &paraAddr);
    records[0] = atoi((char *)paraAddr);
    if(records[0] == 0 && *paraAddr != '0'){
        return -1;
    }
    ret = myVR.checkSignature(records[0], buf);
    printSeperator();
    if(ret == 0){
        Serial.println(F("Signature isn't set.));
    }
    else if(ret > 0){
        Serial.print(F("Signature:));
        printSignature(buf, ret);
        Serial.println();
    }
    else{
        Serial.println(F("Get sig error or timeout.));
    }
    printSeperator();
    return 0;
}
/**
 * @brief Handle "test" command
 * @param len --> command length
 * paraNum --> number of parameters
 * @retval 0 --> success
 * -1 --> Command format error
 */
int cmdTest(int len, int paraNum)
{
    printSeperator();
    Serial.println(F("TEST is not supported.));
    printSeperator();
    return 0;
}
int cmdSettings(int len, int paraNum)
{
    int ret;
    if(paraNum != 1){
        return -1;
    }
    ret = myVR.checkSystemSettings(buf);
    if( ret > 0){
        printSeperator();
        printSystemSettings(buf, ret);
        printSeperator();
    }
}

```

```

}
else{
    printSeperator();
    Serial.println(F("Check system settings error or timeout"));
    printSeperator();
}
return 0;
}
/*****
/**
 * @brief Print signature, if the character is invisible,
 * print hexible value instead.
 * @param buf --> command length
 * len --> number of parameters
 */
void printSignature(uint8_t *buf, int len)
{
    int i;
    for(i=0; i<len; i++){
        if(buf[i]>0x19 && buf[i]<0x7F){
            Serial.write(buf[i]);
        }
        else{
            Serial.print(F("["));
            Serial.print(buf[i], HEX);
            Serial.print(F("]"));
        }
    }
}
/****
 * @brief Print signature, if the character is invisible,
 * print hexible value instead.
 * @param buf --> VR module return value when voice is recognized.
 * buf[0] --> Group mode(FF: None Group, 0x8n: User, 0x0n: System
 * buf[1] --> number of record which is recognized.
 * buf[2] --> Recognizer index(position) value of the recognized record.
 * buf[3] --> Signature length
 * buf[4]~buf[n] --> Signature
 */
void printVR(uint8_t *buf)
{
    Serial.println(F("VR Index\tGroup\tRecordNum\tSignature"));
    Serial.print(buf[2], DEC);
    Serial.print(F("\t\t"));
    if(buf[0] == 0xFF){
        Serial.print(F("NONE"));
    }
    else if(buf[0]&0x80){
        Serial.print(F("UG "));
        Serial.print(buf[0]&(~0x80), DEC);
    }
    else{
        Serial.print(F("SG "));
        Serial.print(buf[0], DEC);
    }
    Serial.print(F("\t"));
    Serial.print(buf[1], DEC);
    Serial.print(F("\t\t"));
    if(buf[3]>0){
        printSignature(buf+4, buf[3]);
    }
}

```

```

}
else{
  Serial.print(F("NONE"));
}
Serial.println(F("\r\n"));
}
/**
 * @brief Print seperator. Print 80 '-'.
 */
void printSeperator()
{
  for(int i=0; i<80; i++){
    Serial.write('-');
  }
  Serial.println();
}
/**
 * @brief Print recognizer status.
 * @param buf --> VR module return value when voice is recognized.
 * buf[0] --> Number of valid voice records in recognizer
 * buf[i+1] --> Record number.(0xFF: Not loaded(Nongroup mode), or not set (Group mode)) (i= 0, 1,
... 6)
 * buf[8] --> Number of all voice records in recognizer
 * buf[9] --> Valid records position indicate.
 * buf[10] --> Group mode indicate(FF: None Group, 0x8n: User, 0x0n:System)
 */
void printCheckRecognizer(uint8_t *buf)
{
  Serial.print(F("All voice records in recognizer: "));
  Serial.println(buf[8], DEC);
  Serial.print(F("Valid voice records in recognizer: "));
  Serial.println(buf[0], DEC);
  if(buf[10] == 0xFF){
    Serial.println(F("VR is not in group mode.));
  }
  else if(buf[10]&0x80){
    Serial.print(F("VR is in user group mode:));
    Serial.println(buf[10]&0x7F, DEC);
  }
  else{
    Serial.print(F("VR is in system group mode:));
    Serial.println(buf[10], DEC);
  }
  Serial.println(F("VR Index\tRecord\t\tComment"));
  for(int i=0; i<7; i++){
    Serial.print(i, DEC);
    Serial.print(F("\t\t"));
    if(buf[i+1] == 0xFF){
      if(buf[10] == 0xFF){
        Serial.print(F("Unloaded\tNONE"));
      }
      else{
        Serial.print(F("Not Set\t\tNONE"));
      }
    }
    else{
      Serial.print(buf[i+1], DEC);
      Serial.print(F("\t\t"));
      if(buf[9]&(1<i)){
        Serial.print(F("Valid"));
      }
    }
  }
}

```

```

    }
    else{
        Serial.print(F("Untrained"));
    }
}
Serial.println();
}
}
/**
 * @brief Print record train status.
 * @param buf --> Check record command return value
 * buf[0] --> Number of checked records
 * buf[2i+1] --> Record number.
 * buf[2i+2] --> Record train status. (00: untrained, 01: trained, FF: record value out of range)
 * (i = 0 ~ buf[0]-1 )
 * num --> Number of trained records
 */
void printCheckRecord(uint8_t *buf, int num)
{
    Serial.print(F("Check "));
    Serial.print(buf[0], DEC);
    Serial.println(F(" records.));
    Serial.print(num, DEC);
    if(num>1){
        Serial.println(F(" records trained.));
    }
    else{
        Serial.println(F(" record trained.));
    }
    for(int i=0; i<buf[0]*2; i += 2){
        Serial.print(buf[i+1], DEC);
        Serial.print(F("\t-->\t"));
        switch(buf[i+2]){
            case 0x01:
                Serial.print(F("Trained"));
                break;
            case 0x00:
                Serial.print(F("Untrained"));
                break;
            case 0xFF:
                Serial.print(F("Record value out of range"));
                break;
            default:
                Serial.print(F("Unknown Stauts"));
                break;
        }
        Serial.println();
    }
}
/**
 * @brief Print record train status.
 * @param buf --> Check record command return value
 * buf[0] --> Number of checked records
 * buf[2i+1] --> Record number.
 * buf[2i+2] --> Record train status. (00: untrained, 01: trained, FF: record value out of range)
 * (i = 0 ~ buf[0]-1 )
 * num --> Number of trained records
 */
void printCheckRecordAll(uint8_t *buf, int num)
{

```

```

Serial.print(F("Check 255"));
Serial.println(F(" records."));
Serial.print(num, DEC);
if(num>1){
  Serial.println(F(" records trained."));
}
else{
  Serial.println(F(" record trained."));
}
myVR.writehex(buf, 255);
for(int i=0; i<255; i++){
  if(buf[i] == 0xF0){
    continue;
  }
  Serial.print(i, DEC);
  Serial.print(F("\t-->\t"));
  switch(buf[i]){
    case 0x01:
      Serial.print(F("Trained"));
      break;
    case 0x00:
      Serial.print(F("Untrained"));
      break;
    case 0xFF:
      Serial.print(F("Record value out of range"));
      break;
    default:
      Serial.print(F("Unknown Stauts"));
      break;
  }
  Serial.println();
}
}
}
/**
 * @brief Print check user group result.
 * @param buf --> Check record command return value
 * buf[8i] --> group number.
 * buf[8i+1] --> group position 0 status.
 * buf[8i+2] --> group position 1 status.
 * ...
 * buf[8i+6] --> group position 5 status.
 * buf[8i+7] --> group position 6 status.
 * (i = 0 ~ len)
 * len --> number of checked groups
 */
void printUserGroup(uint8_t *buf, int len)
{
  int i, j;
  Serial.println(F("Check User Group:"));
  for(i=0; i<len; i++){
    Serial.print(F("Group:"));
    Serial.println(buf[8*i]);
    for(j=0; j<7; j++){
      if(buf[8*i+1+j] == 0xFF){
        Serial.print(F("NONE\t"));
      }
      else{
        Serial.print(buf[8*i+1+j], DEC);
        Serial.print(F("\t"));
      }
    }
  }
}

```

```

    }
    Serial.println();
  }
}
/**
 * @brief Print "load" command return value.
 * @param buf --> "load" command return value
 * buf[0] --> number of records which are load successfully.
 * buf[2i+1] --> record number
 * buf[2i+2] --> record load status.
 * 00 --> Loaded
 * FC --> Record already in recognizer
 * FD --> Recognizer full
 * FE --> Record untrained
 * FF --> Value out of range"
 * (i = 0 ~ (len-1)/2 )
 * len --> length of buf
 */
void printLoad(uint8_t *buf, uint8_t len)
{
  if(len == 0){
    Serial.println(F("Load Successfully. "));
    return;
  }
  else{
    Serial.print(F("Load success: "));
    Serial.println(buf[0], DEC);
  }
  for(int i=0; i<len-1; i += 2){
    Serial.print(F("Record "));
    Serial.print(buf[i+1], DEC);
    Serial.print(F("\t"));
    switch(buf[i+2]){
      case 0:
        Serial.println(F("Loaded"));
        break;
      case 0xFC:
        Serial.println(F("Record already in recognizer"));
        break;
      case 0xFD:
        Serial.println(F("Recognizer full"));
        break;
      case 0xFE:
        Serial.println(F("Record untrained"));
        break;
      case 0xFF:
        Serial.println(F("Value out of range"));
        break;
      default:
        Serial.println(F("Unknown status"));
        break;
    }
  }
}
/**
 * @brief Print "train" command return value.
 * @param buf --> "train" command return value
 * buf[0] --> number of records which are trained successfully.
 * buf[2i+1] --> record number
 * buf[2i+2] --> record train status.

```

```

* 00 --> Trained
* FE --> Train Time Out
* FF --> Value out of range"
* (i = 0 ~ len-1 )
* len --> length of buf
*/
void printTrain(uint8_t *buf, uint8_t len)
{
    if(len == 0){
        Serial.println(F("Train Finish."));
        return;
    }
    else{
        Serial.print(F("Train success: "));
        Serial.println(buf[0], DEC);
    }
    for(int i=0; i<len-1; i += 2){
        Serial.print(F("Record "));
        Serial.print(buf[i+1], DEC);
        Serial.print(F("\t"));
        switch(buf[i+2]){
            case 0:
                Serial.println(F("Trained"));
                break;
            case 0xFE:
                Serial.println(F("Train Time Out"));
                break;
            case 0xFF:
                Serial.println(F("Value out of range"));
                break;
            default:
                Serial.print(F("Unknown status "));
                Serial.println(buf[i+2], HEX);
                break;
        }
    }
}
/**
 * @brief Print "sigtrain" command return value.
 * @param buf --> "sigtrain" command return value
 * buf[0] --> number of records which are trained successfully.
 * buf[1] --> record number
 * buf[2] --> record train status.
 * 00 --> Trained
 * F0 --> Trained, signature truncate
 * FE --> Train Time Out
 * FF --> Value out of range"
 * buf[3] ~ buf[len-1] --> Signature.
 * len --> length of buf
 */
void printSigTrain(uint8_t *buf, uint8_t len)
{
    if(len == 0){
        Serial.println(F("Train With Signature Finish."));
        return;
    }
    else{
        Serial.print(F("Success: "));
        Serial.println(buf[0], DEC);
    }
}

```

```

Serial.print(F("Record "));
Serial.print(buf[1], DEC);
Serial.print(F("\t"));
switch(buf[2]){
case 0:
  Serial.println(F("Trained"));
  break;
case 0xF0:
  Serial.println(F("Trained, signature truncate"));
  break;
case 0xFE:
  Serial.println(F("Train Time Out"));
  break;
case 0xFF:
  Serial.println(F("Value out of range"));
  break;
default:
  Serial.print(F("Unknown status "));
  Serial.println(buf[2], HEX);
  break;
}
Serial.print(F("SIG: "));
Serial.write(buf+3, len-3);
Serial.println();
}
/**
 * @brief Print "settings" command return value.
 * @param buf --> "settings" command return value
 * buf[0] --> number of records which are trained successfully.
 * buf[1] --> record number
 * buf[2] --> record train status.
 * 00 --> Trained
 * F0 --> Trained, signature truncate
 * FE --> Train Time Out
 * FF --> Value out of range"
 * buf[3] ~ buf[len-1] --> Signature.
 * len --> length of buf
 */
const unsigned int io_pw_tab[16]={
  10, 15, 20, 25, 30, 35, 40, 45,
  50, 75, 100, 200, 300, 400, 500, 1000
};
void printSystemSettings(uint8_t *buf, int len)
{
  switch(buf[0]){
  case 0:
  case 3:
    Serial.println(F("Baud rate: 9600"));
    break;
  case 1:
    Serial.println(F("Baud rate: 2400"));
    break;
  case 2:
    Serial.println(F("Baud rate: 4800"));
    break;
  case 4:
    Serial.println(F("Baud rate: 19200"));
    break;
  case 5:
    Serial.println(F("Baud rate: 38400"));

```

```

    break;
default:
    Serial.println(F("Baud rate: UNKONOWN"));
    break;
}
switch(buf[1]){
case 0:
case 0xFF:
    Serial.println(F("Outpu IO Mode: Pulse"));
    break;
case 1:
    Serial.println(F("Outpu IO Mode: Toggle"));
    break;
case 2:
    Serial.println(F("Outpu IO Mode: Clear(When recognized) "));
    break;
case 3:
    Serial.println(F("Outpu IO Mode: Set(When recognized)"));
    break;
default:
    Serial.println(F("Output IO Mode: UNKONOWN"));
    break;
}
if(buf[2] > 15){
    Serial.println(F("Pulse width: UNKONOWN"));
}
else{
    Serial.print(F("Pulse Width: "));
    Serial.print(io_pw_tab[buf[2]], DEC);
    Serial.println(F("ms"));
}
if(buf[3] == 0 || buf[3] == 0xFF){
    Serial.println(F("Auto Load: disable"));
}
else{
    Serial.println(F("Auto Load: enable"));
}
}
switch(buf[4]){
case 0:
case 0xFF:
    Serial.println(F("Group control by external IO: disabled"));
    break;
case 1:
    Serial.println(F("Group control by external IO: system group selected"));
    break;
case 2:
    Serial.println(F("Group control by external IO: user group selected"));
    break;
default:
    Serial.println(F("Group control by external IO: UNKNOWN STATUS"));
    break;
}
}
}
void printHelp(void)
{
    Serial.println(F("COMMAND      FORMAT          EXAMPLE      Comment"));
    printSeperator();
    // Serial.println(F("-----"));
-");
    Serial.println(F("train      train (r0) (r1)...  train 0 2 45    Train records"));

```

```

Serial.println(F("load      load (r0) (r1) ...  load 0 51 2 3    Load records"));
Serial.println(F("clear      clear                clear          remove all records in Recognizer"));
Serial.println(F("record    record (r0) (r1)... record 0 79      Check record train status"));
Serial.println(F("vr        vr                  vr             Check recognizer status"));
Serial.println(F("getsig    getsig (r)          getsig 0        Get signature of record (r)"));
Serial.println(F("sigtrain  sigtrain (r) (sig) sigtrain 0 ZERO Train one record(r) with
signature(sig)"));
Serial.println(F("settings  settings           settings        Check current system settings"));
Serial.println(F("help      help              help           print this message"));
}

```

ANEXO C – CÓDIGO PARA CONTROLE DOS SERVOS- MOTORES

```

#include <SoftwareSerial.h>
#include <Servo.h>
#include "VoiceRecognitionV3.h"
/**
  Connection
  Arduino  VoiceRecognitionModule
  2 -----> TX
  3 -----> RX
*/
VR myVR(2, 3); // 2:RX 3:TX, define os pinos 2 e 3 para comunicação serial.
Servo s1; // Variável Servo1
Servo s2; // Variável Servo2
Servo s3; // Variável Servo3
uint8_t records[7]; // save record
uint8_t buf[64];
int rele1 = 13; // Define Rele 1 no Porta Digital 13
int rele2 = 12; // Define Rele 2 no Porta Digital 12
int rele3 = 11; // Define Rele 3 no Porta Digital 11
int pos; // Posição Servo
int pos1; // Posição Servo1
int pos2; // Posição Servo2
int pos3; // Posição Servo3
#define abre (0) // Define gravação 0 como comando abre
#define fecha (1) // Define gravação 1 como comando fecha
#define pinca (2) // Define gravação 2 como comando pinca
#define palmar (3) // Define gravação 3 como comando palmar
#define testando (4) // Define gravação 4 como comando testando
#define desligar (5) // Define gravação 5 como comando desligar
#define SERVO1 6 // Porta Digital 6 PWM
#define SERVO2 9 // Porta Digital 9 PWM
#define SERVO3 10 // Porta Digital 10 PWM
/**
  @brief Print signature, if the character is invisible,
         print hexible value instead.
  @param buf --> command length
         len --> number of parameters
*/
void printSignature(uint8_t *buf, int len)
{
  int i;
  for (i = 0; i < len; i++) {
    if (buf[i] > 0x19 && buf[i] < 0x7F) {
      Serial.write(buf[i]);
    }
    else {
      Serial.print("[");
      Serial.print(buf[i], HEX);
      Serial.print("]");
    }
  }
}
/**
  @brief Print signature, if the character is invisible,
         print hexible value instead.
  @param buf --> VR module return value when voice is recognized.

```

```

    buf[0] --> Group mode(FF: None Group, 0x8n: User, 0x0n:System
    buf[1] --> number of record which is recognized.
    buf[2] --> Recognizer index(position) value of the recognized record.
    buf[3] --> Signature length
    buf[4]~buf[n] --> Signature
*/
void printVR(uint8_t *buf)
{
  Serial.println("VR Index\tGroup\tRecordNum\tSignature");
  Serial.print(buf[2], DEC);
  Serial.print("\t\t");
  if (buf[0] == 0xFF) {
    Serial.print("NONE");
  }
  else if (buf[0] & 0x80) {
    Serial.print("UG ");
    Serial.print(buf[0] & (~0x80), DEC);
  }
  else {
    Serial.print("SG ");
    Serial.print(buf[0], DEC);
  }
  Serial.print("\t");
  Serial.print(buf[1], DEC);
  Serial.print("\t\t");
  if (buf[3] > 0) {
    printSignature(buf + 4, buf[3]);
  }
  else {
    Serial.print("NONE");
  }
  Serial.println("\r\n");
}
void setup()
{
  /** initialize */
  myVR.begin(9600);
  Serial.begin(9600);
  Serial.println("Controle de Prótese Carregado");
  pinMode(rele1, OUTPUT); // Define pino do rele 1 como saída
  pinMode(rele2, OUTPUT); // Define pino do rele 2 como saída
  pinMode(rele3, OUTPUT); // Define pino do rele 3 como saída
  s1.attach(SERVO1); // Anexa valor de s1 para porta do Servo 1
  s2.attach(SERVO2); // Anexa valor de s2 para porta do Servo 2
  s3.attach(SERVO3); // Anexa valor de s3 para porta do Servo 3
  digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
  digitalWrite(rele2, HIGH); // Define estado do rele 1 como Ligado
  digitalWrite(rele3, HIGH); // Define estado do rele 1 como Ligado
  s1.write(20); // Inicia servo1 posição zero
  s2.write(20); // Inicia servo2 posição zero
  s3.write(180); // Inicia servo3 posição zero

  if (myVR.clear() == 0) {
    Serial.println("Recognizer cleared.");
  } else {
    Serial.println("Not find VoiceRecognitionModule.");
    Serial.println("Please check connection and restart Arduino.");
    while (1);
  }
}
if (myVR.load((uint8_t)abre) >= 0) {

```

```

    Serial.println("abre loaded");
}
if (myVR.load((uint8_t)fecha) >= 0) {
    Serial.println("fecha loaded");
}
if (myVR.load((uint8_t)pinca) >= 0) {
    Serial.println("pinca loaded");
}
if (myVR.load((uint8_t)palmar) >= 0) {
    Serial.println("palmar loaded");
}
if (myVR.load((uint8_t)testando) >= 0) {
    Serial.println("testando loaded");
}
if (myVR.load((uint8_t)desligar) >= 0) {
    Serial.println("desligar loaded");
}
}
}
void loop()
{
    int ret;
    ret = myVR.recognize(buf, 50);
    if (ret > 0) {
        switch (buf[1]) {
            case abre:
                digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
                digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
                digitalWrite(rele3, HIGH); // Define estado do rele 2 como Ligado
                s1.write(20); // Movimenta servo1 para posição zero
                s2.write(20); // Movimenta servo2 para posição zero
                s3.write(180); // Movimenta servo3 para posição zero
                delay(30); // Atraso em milisegundos
                break;
            case fecha:
                digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
                digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
                digitalWrite(rele3, HIGH); // Define estado do rele 3 como Ligado
                s1.write(20); // Movimenta servo1 para posição zero
                s2.write(20); // Movimenta servo2 para posição zero
                s3.write(180); // Movimenta servo3 para posição zero
                delay(300); // Atraso em milisegundos
                for(pos = 20; pos < 90; pos++) // Movimenta servo1 em 70 graus horário
                {
                    s1.write(pos);
                    delay(5); // Atraso a cada grau, quanto maior mais lento
                }
                for(pos = 20; pos < 180; pos++) // Movimenta servo2 em 160 graus horário
                {
                    s2.write(pos);
                    delay(5); // Atraso a cada grau, quanto maior mais lento
                }
                for(pos = 180; pos > 30; pos--) // Movimenta servo3 em 150 graus anti-horário
                {
                    s3.write(pos);
                    delay(5); // Atraso a cada grau, quanto maior mais lento
                }
                for(pos = 90; pos < 160; pos++) // Movimenta servo1 em mais 70 graus horário
                {
                    s1.write(pos);
                    delay(5); // Atraso a cada grau, quanto maior mais lento
                }
            }
        }
    }
}

```

```

}
  break;
  case pinca:
    digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
    digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
    digitalWrite(rele3, HIGH); // Define estado do rele 2 como Ligado
    s1.write(20); // Movimenta servo1 para posição zero
    s2.write(20); // Movimenta servo2 para posição zero
    s3.write(180); // Movimenta servo3 para posição zero
    delay(300); // Atraso em milisegundos
    for(pos = 20; pos < 160; pos++) // Movimenta servo1 e servo2 em 140 graus horário
    {
      s1.write(pos);
      s2.write(pos);
      delay(5); // Atraso a cada grau, quanto maior mais lento
    }
    break;
    case palmar:
      digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
      digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
      digitalWrite(rele3, HIGH); // Define estado do rele 2 como Ligado
      s1.write(20); // Movimenta servo1 para posição zero
      s2.write(20); // Movimenta servo2 para posição zero
      s3.write(180); // Movimenta servo3 para posição zero
      delay(300); // Atraso em milisegundos
      for(pos = 20; pos < 150; pos++) // Movimenta servo1 em 130 graus horário
      {
        s1.write(pos);
        delay(5); // Atraso a cada grau, quanto maior mais lento
      }
      for(pos = 20; pos < 150; pos++) // Movimenta servo2 em 130 graus horário
      {
        s2.write(pos);
        delay(5); // Atraso a cada grau, quanto maior mais lento
      }
      for(pos = 180; pos > 40; pos--) // Movimenta servo3 em 140 graus anti-horário
      {
        s3.write(pos);
        delay(5); // Atraso a cada grau, quanto maior mais lento
      }
    break;
    case testando:
      digitalWrite(rele1, HIGH); // Define estado do rele 1 como Ligado
      digitalWrite(rele2, HIGH); // Define estado do rele 2 como Ligado
      digitalWrite(rele3, HIGH); // Define estado do rele 2 como Ligado
      s1.write(20); // Movimenta servo1 para posição zero
      s2.write(20); // Movimenta servo2 para posição zero
      s3.write(180); // Movimenta servo3 para posição zero
      delay(500); // Atraso em milisegundos
      for(pos = 20; pos < 90; pos++) // Movimenta servo1 em 70 graus horário
      {
        s1.write(pos);
        delay(5); // Atraso a cada grau, quanto maior mais lento
      }
      for(pos = 20; pos < 180; pos++) // Movimenta servo2 em 160 graus horário
      {
        s2.write(pos);
        delay(5); // Atraso a cada grau, quanto maior mais lento
      }
      for(pos = 180; pos > 30; pos--) // Movimenta servo3 em 150 graus anti-horário

```

```

{
  s3.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
for(pos = 90; pos < 160; pos++) // Movimenta servo1 em mais 70 graus horário
{
  s1.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
delay(500);
s1.write(20); // Movimenta servo1 para posição zero
s2.write(20); // Movimenta servo2 para posição zero
s3.write(180); // Movimenta servo3 para posição zero
delay(300); // Atraso em milisegundos
for(pos = 20; pos < 160; pos++)
{
  s1.write(pos);
  s2.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
delay(500);
s1.write(20); // Movimenta servo1 para posição zero
s2.write(20); // Movimenta servo2 para posição zero
s3.write(180); // Movimenta servo3 para posição zero
delay(300); // Atraso em milisegundos
for(pos = 20; pos < 150; pos++) // Movimenta servo1 em 130 graus horário
{
  s1.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
for(pos = 20; pos < 150; pos++) // Movimenta servo2 em 130 graus horário
{
  s2.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
for(pos = 180; pos > 40; pos--) // Movimenta servo3 em 140 graus anti-horário
{
  s3.write(pos);
  delay(5); // Atraso a cada grau, quanto maior mais lento
}
delay(500); // Atraso em milisegundos
s1.write(20); // Movimenta servo1 para posição zero
s2.write(20); // Movimenta servo2 para posição zero
s3.write(180); // Movimenta servo3 para posição zero
delay(500); // Atraso em milisegundos
digitalWrite(rele1, LOW); // Define estado do rele 1 como Desligado
digitalWrite(rele2, LOW); // Define estado do rele 2 como Desligado
digitalWrite(rele3, LOW); // Define estado do rele 3 como Desligado
break;
case desligar:
s1.write(20); // Movimenta servo1 para posição zero
s2.write(20); // Movimenta servo2 para posição zero
s3.write(180); // Movimenta servo3 para posição zero
delay(500); // Atraso em milisegundos
digitalWrite(rele1, LOW); // Define estado do rele 1 como Desligado
digitalWrite(rele2, LOW); // Define estado do rele 2 como Desligado
digitalWrite(rele3, LOW); // Define estado do rele 3 como Desligado
break;
default:
  Serial.println("Record function undefined");
}

```

```
        break;
    }
    /** voice recognized */
    printVR(buf);
}
}
```