

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

**USO DE MICROCONTROLADOR NA AUTOMAÇÃO RESIDENCIAL
PARA PESSOAS COM NECESSIDADES ESPECIAIS**

MARCUS VINÍCIUS PEREIRA ARAÚJO

Madson Cruz Machado
Orientador

São Luís (MA)
2018

MARCUS VINÍCIUS PEREIRA ARAÚJO

**USO DE MICROCONTROLADOR NA AUTOMAÇÃO RESIDENCIAL
PARA PESSOAS COM NECESSIDADES ESPECIAIS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação.

Orientador: Madson Cruz Machado

São Luís (MA)
2018

Araújo, Marcus Vinícius Pereira.

Uso de microcontrolador na automação residencial para pessoas com necessidades especiais / Marcus Vinícius Pereira Araújo. – São Luís, 2018.

63 f.

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2018.

Orientador: Prof. Me. Madson Cruz Machado.

1. Automação residencial. 2. Arduino. 3. Deficiência visual. 4. Acessibilidade. I. Título.

MARCUS VINÍCIUS PEREIRA ARAÚJO

**USO DE MICROCONTROLADOR NA AUTOMAÇÃO RESIDENCIAL
PARA PESSOAS COM NECESSIDADES ESPECIAIS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação.

Aprovado em: ____/____/____

BANCA EXAMINADORA:

Prof. Me. Madson Cruz Machado

Orientador

Universidade Estadual do Maranhão

Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva

Universidade Estadual do Maranhão

Prof. Me. Dayse Yanne Caldas Siqueira de Sousa

Universidade Estadual do Maranhão

São Luís (MA)

2018

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, por ter me guiado durante toda essa jornada, dando saúde, bençãos, forças e coragem para questionar realidades e propor sempre um novo mundo de possibilidades, fazendo com que eu jamais desistisse dos meus objetivos.

Aos meus pais Raimundo Marques Araujo e Rozeilde Vale Pereira Araujo por me apoiarem e incentivarem durante toda a vida, nunca deixando eu ser vencido pelo cansaço, principalmente nos momentos mais difíceis, além do mais, sou grato pelos esforços que tiveram em me dar uma boa educação e uma melhor qualidade de vida possível.

Aos meus amigos e colegas de todos os cursos, mas em especial aos de Engenharia de Computação, não obstante somente a turma de 2011.1, pois cada turma teve sua devida importância para fortalecer o sonho geral, ser um Engenheiro de Computação, ressaltando o ditado popular antigo “Um por todos e todos por um”.

Aos professores, especialmente a Madson Cruz Machado, que me deu todo o suporte com suas correções e incentivos para o desfecho desse trabalho de conclusão de curso.

A Universidade Estadual do Maranhão, por me proporcionar um ambiente criativo e amigável para os estudos, melhorando o meu desenvolver pessoal e profissional, estou em gratidão com cada membro do corpo docente, à direção e a administração dessa instituição de ensino que passaram seus conhecimentos durante esse percurso.

*“Quando tudo parecer dar errado em
sua vida, lembre-se que o avião decola
contra o vento, e não a favor dele”
(Henry Ford)*

LISTA DE FIGURAS

Figura 1 - Comparativo do mercado de automação residencial em 2015	18
Figura 2 - Proporção de pessoas com deficiência intelectual.....	21
Figura 3 - Proporção de pessoas com deficiência física.....	21
Figura 4 - Proporção de pessoas com deficiência auditiva.....	22
Figura 5 - Proporção de pessoas com deficiência visual	23
Figura 6 - Proporção de pessoas que nasceram com deficiência visual e proporção de pessoas que adquiriram.....	24
Figura 7 - Proporção de pessoas com grau intenso de limitações ou que não conseguem fazer as atividades habituais	24
Figura 8 - Modelo simples do projeto	25
Figura 9 - Visão superior da placa Arduino UNO.....	26
Figura 10 - Visão superior do Voice Recognition Module V3.....	28
Figura 11 - Visão superior e de perfil dos LED	29
Figura 12 - Visão superior do microfone acoplado ao módulo de reconhecimento de voz	30
Figura 13 - Visão de perfil do potenciômetro de 10K.....	30
Figura 14 - Visão superior do display LCD 16x1	31
Figura 15 - Visão superior e de perfil do buzzer	32
Figura 16 - Visão superior e de perfil do micro servo 9g SG90.....	32
Figura 17 - Visão Frontal da Maquete.....	35
Figura 18 - Visão Lateral da Maquete	35
Figura 19 - Frente da Placa Ideal.....	36
Figura 20 - Construção da Maquete	36
Figura 21 - Costa da Placa Ideal.....	37
Figura 22 - Costa da Placa Real	37
Figura 23 - Ligações físicas do projeto	38
Figura 24 - Legenda dos botões da interface	40
Figura 25 - A interface da IDE	40
Figura 26 - Reconhecimento da palavra Sala	45

LISTA DE TABELAS

Tabela 1 - Comparativo do mercado de automação residencial em 2015	18
Tabela 2 - Especificações técnicas do Arduino Uno	27
Tabela 3 - Especificações técnicas do Voice Recognition Module V3	28
Tabela 4 - Especificações técnicas do Miro Servo 9g SG90 Tower Pro	33
Tabela 5 - Disposição dos pinos em relação aos cômodos.....	39
Tabela 6 - Disposição dos pinos em relação aos outros componentes	39
Tabela 7 - Teste somente para uma entrada	43
Tabela 8 - Teste somente para duas entradas, com portão	43
Tabela 9 - Teste somente para três variáveis, com cozinha.....	43
Tabela 10 - Teste somente para duas variáveis, com sala	44
Tabela 11 - Teste somente para duas variáveis, com cozinha	44
Tabela 12 - Teste somente para duas variáveis, com quarto	44
Tabela 13 - Teste somente para três variáveis, com portão.....	44
Tabela 14 - Teste somente para três variáveis, com sala.....	44
Tabela 15 - Teste somente para quatro variáveis, com portão	45
Tabela 16 - Teste somente para quatro variáveis, com sala	45
Tabela 17 - Teste para todas as entradas	45
Tabela 18 – Custo do Projeto	47

RESUMO

O uso da tecnologia para automatizar tarefas habituais, têm se tornado cada vez mais procurado pela sociedade devido as suas vantagens, e algumas delas são, a segurança, o conforto e a economia, sendo acessível a todos, não importando se a pessoa possui algum tipo de deficiência. Dessa maneira, a proposta deste trabalho é facilitar com que pessoas com necessidades especiais, como deficiente visual, deficiente físico, pessoas que sintam dificuldades para a realização de tarefas simples em uma casa. Com a utilização da automação residencial, por exemplo, a pessoa estará tendo acessibilidade para obter o controle da iluminação e no funcionamento de alguns aparelhos eletrônicos, como a televisão, além do portão automático, e tudo isso utilizando o hardware microcontrolador Arduino interligado com outro hardware de reconhecimento de voz, com funcionamento somente local, ou seja, na residência o qual foi implantado, não fazendo a utilização de nenhum aplicativo, nem da internet.

Palavras-chave: Automação Residencial. Arduino. Deficiência Visual. Acessibilidade.

ABSTRACT

The use of technology to automate routine tasks has become increasingly sought after by society due to its advantages. Some of them are safety, comfort and cost reduction, making it accessible to all, regardless of whether the person has some type of disability. In this context, the proposal of this paper is to make life easier for people with special needs, such as the visually and physically impaired, and people who find it difficult to execute simple domestic tasks. By applying automation, for example, a person could be able to control home lighting and the operation of some electronic appliances, such as the TV and the automatic gate; all this using the microcontroller hardware Arduino, interconnected with another piece of hardware, able to do voice recognition; functioning only locally, that is, in the house where it has been implemented, not making use of any mobile application, nor an internet connection.

Keywords: *Residencial Automation. Arduino. Visual Impairment. Accessibility.*

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivo Específico	13
1.2 METODOLOGIA	14
1.3 ESTRUTURA DO TRABALHO	16
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1 AUTOMAÇÃO RESIDENCIAL.....	17
2.2 MICROCONTROLADORES	19
2.3 IBGE (Instituto Brasileiro de Geografia e Estatística).....	20
3. MODELO PROPOSTO	25
3.1 DESCRIÇÕES DOS COMPONENTES DO SISTEMA	26
3.1.1 Arduino UNO	26
3.1.2 Voice Recognition Module V3.....	27
3.1.3 Sensores e Atuadores	28
3.1.3.1 LED (Diodo Emissor de Luz)	29
3.1.3.2 Microfone	29
3.1.3.3 Potenciômetro.....	30
3.1.3.4 LCD (Display de Cristal Líquido)	31
3.1.3.5 Buzzer.....	31
3.1.3.6 Micro Servo 9g SG90	32
4. DESENVOLVIMENTO DO MODELO PROPOSTO	34
4.1 CONSTRUÇÃO DA MAQUETE.....	34
4.2 LIGAÇÕES FÍSICAS	38
4.3 DESENVOLVIMENTO DO ALGORITMO NO ARDUINO	39
5. TESTES DE HARDWARE E SOFTWARE	43
6. CONCLUSÕES E CONSIDERAÇÕES FINAIS.....	46
6.1 SUGESTÕES PARA TRABALHOS FUTUROS.....	48
REFERÊNCIAS.....	49
APÊNDICE A – Código Fonte Principal do Projeto	51

1 INTRODUÇÃO

A automação residencial, conhecida também como domótica, é uma palavra de origem latina com a junção de duas palavras, *Domus* que significa casa e Robótica que significa ter controle automatizado de alguma coisa. Um marco do uso dessa palavra, ocorreu em meados do século XVIII, em que acontecia a revolução industrial para automatizar processos dentro das indústrias, e com essa influência logo a ideia principal da automatização foi expandida para as outras áreas, principalmente por suas vantagens, economia, segurança, comodidade, acessibilidade, entre outras (MOREIRA, 2013).

Com o passar do tempo, o crescimento da automação e a implantação de novos produtos em vários setores, tornou-se notório que não só as indústrias o queriam, mas os latifundiários e a própria sociedade queriam usufruir desta modernidade tecnológica a preços acessíveis. Sendo assim o termo automação foi expandido para vários setores, sendo industrial, predial, residencial, utilizando na implantação um microcontrolador que é responsável por gerenciar uma determinada ação, de acordo com uma determinada resposta a ser programada, em outras palavras, um microcontrolador também é conhecido como um sistema embarcado.

Os sistemas embarcados são sistemas sem interface com o usuário, sendo um hardware de preço acessível e dedicado a executar uma única tarefa, interagindo com o local ao qual foi instalado, sendo necessária apenas a utilização de uma linguagem de programação, que no caso do trabalho proposto foi escolhido a linguagem C++ por ser uma linguagem orientada a objeto e seu ambiente de programação ser um software livre.

Dessa maneira fica viável a construção de um protótipo de baixo custo, já que a ideia principal é a utilização da voz para ligar e desligar as lâmpadas, abrir e fechar portas de maneira mais amigável ao deficiente, e futuramente ter controle sobre a televisão e outros eletrodomésticos.

1.1 OBJETIVOS

A seguir temos o objetivo geral e específico, isto para que se tenha um controle no desenvolvimento do projeto para que não ocorra desvio do que esta sendo proposto.

1.1.1 Objetivo Geral

Desenvolver um protótipo para o ambiente residencial que envolva a plataforma de prototipagem eletrônica Arduino e o módulo de reconhecimento de voz da Elechouse, ambos sendo integrados para que seja usado junto com o usuário através da voz, já que este será o atuador para o funcionamento do sistema implantado, dando um comando simples de voz para se obter uma resposta já programada pelo desenvolvedor.

1.1.2 Objetivo Específico

- Projeto desenvolvido principalmente para os quatro tipos de deficientes (intelectual, físico, auditivo e visual);
- Aumentar a inclusão social de pessoas que possuem algum tipo de deficiência;
- Acoplar o módulo de reconhecimento de voz da Elechouse com o Arduino;
- Definir e descrever as etapas do processo a serem realizadas;
- Realizar teste para saber se o sistema está funcionando corretamente, de acordo com o esperado;
- Averiguar o custo benefício para o desenvolvimento do sistema proposto;

1.2 METODOLOGIA

Para o desenvolvimento deste projeto, foi necessário dividi-lo em quatro etapas para obtenção do produto final, e essas etapas se encontram logo abaixo:

- 1ª etapa: pesquisar qual placa microcontroladora e qual módulo atendem os requisitos exigidos no projeto;
- 2ª etapa: fazer as conexões físicas entre ambas, integrando-as;
- 3ª etapa: desenvolver no software o algoritmo necessário;
- 4ª etapa: testar e validar o produto final;

Na primeira etapa procurou-se entender o problema e quais eram os requisitos de projeto, em seguida fez-se uma pesquisa mais detalhada sobre qual placa com microcontrolador seria mais adequada para resolver o problema em questão. Sabendo disso, foram encontradas três placas, Arduino Uno Rev 3, Arduino Mega 2560 e a Raspberry, mas como o Raspberry era a mais cara entre as três, e um dos princípios a levar em conta é a economia, foi logo eliminada, restando as duas primeiras. Pesquisando mais a fundo sobre ambas, e o fato de serem da mesma empresa, pôde-se achar o manual com o datasheet, assim facilitando o processo de escolha, no caso, o Arduino Uno Rev 3 foi a escolhida devido está possuir pinagens suficientes, e ser mais econômica em relação ao preço, apesar dela ter a desvantagem de possuir uma memória EEPROM menor, comparado ao do Arduino Mega.

Sabendo qual placa a ser utilizada, continuaram as pesquisas, mas sobre qual o módulo de reconhecimento de voz adequaria ao problema proposto. Não tendo muitas alternativas, a escolhida foi Voice Recognition Module V3 da Elechouse₂, que somando os preços de ambas as placas, obteríamos um total de 150 reais, equivalente ao preço do Raspberry que havia sido eliminado.

Na segunda etapa, obtendo os conhecimentos no manual de ambas, foi realizada a integração através de cabeamentos físicos, não encontrando nenhuma dificuldade no processo, apenas sendo necessário fazer uma organização na fiação exposta, mantendo sempre seguro e

organizado para não ocorrer problemas futuros, além da estética está mais aceitável, mesmo sendo somente um protótipo em fase de testes.

Tendo em vista que as etapas anteriores foram finalizadas, pode-se iniciar a programação no software, na própria IDE do arduino, em linguagem C++ foi feito o código para ser implantado no próprio arduino através do cabo USB, pois para o módulo era somente necessário fazer gravações das palavras chaves ditas pelo usuário para resultar em uma ação por parte do microcontrolador quando for reconhecida pelo módulo.

Chegando à etapa final, ocorreram os testes iniciais, verificando a alimentação e os cabamentos se estavam em ordem, em seguinte, o teste intermediário, averiguando se a voz estava sendo reconhecida pelo módulo, e o teste final, se o microcontrolador estava se comportando de acordo com o programado devido a sua integralização com o módulo, fazendo suas funções de ligar e desligar luz, abrir e fechar portão.

1.3 ESTRUTURA DO TRABALHO

A estrutura do trabalho está dividida em seis capítulos que serão explicados aqui de forma resumida sobre cada um deles. No primeiro capítulo tem-se abordado sobre automação residencial, objetivo geral e específico, além da metodologia, de forma introdutória para ter uma noção do que se trata o trabalho proposto.

No segundo capítulo temos conceitos teóricos sobre automação residencial, microcontrolador e IBGE, para facilitar o entendimento sobre cada um desses assuntos abordados, além de tabela e figuras para compreender de forma mais didática o funcionamento, como se comporta na placa Arduino, além de ter conceitos sobre o módulo de reconhecimento de voz para ser utilizado nos capítulos seguintes.

No terceiro capítulo são detalhadas as descrições de cada componente utilizado, incluindo datasheet nos principais como uma solução mais confiável na montagem do projeto, mesmo utilizando voltagem de 5 Volts para alimentar os componentes, sendo considerada uma voltagem baixa, é preciso ter cuidado também com a amperagem para não ocasionar a queima das peças do protótipo.

No quarto capítulo temos de forma mais organizada e detalhada a construção do projeto, explicando partes principais do algoritmo, incluindo dados extras obtidos durante o desenvolver do trabalho.

No quinto capítulo são realizados os testes de acordo com os resultados obtidos, validando o produto final.

No sexto capítulo a conclusão é feita para informar se os objetivos foram alcançados de acordo com o esperado, e no caso mostrando quais possíveis soluções, e incrementações podem ser realizadas em trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo encontram-se as informações necessárias para a fundamentação teórica do modelo proposto, incluindo dados, tabelas e figuras.

2.1 AUTOMAÇÃO RESIDENCIAL

Segundo Roveri (2012), automação residencial “é o conjunto de serviços realizados através da tecnologia integrada onde podemos satisfazer as necessidades básicas de segurança, comunicação, conforto e gestão energética de uma casa.”, e de acordo com o presidente da Aureside, José Cândido Forti, umas das vantagens da automação residencial são transformarem casas em confortáveis refúgios capazes de oferecer segurança e economia de custos, e para o diretor executivo da Aureside, José Roberto Muratori, afirma que “um indicador normalmente utilizado pelo mercado é estimar o preço de um projeto de automação em algo que varia de 3% a 5% do valor do imóvel para sistemas com maior grau de tecnologia”.

Sabendo desses conceitos é possível afirmar que a automação residencial não é apenas um produto a ser vendida no mercado, ela garante uma qualidade de vida melhor no dia a dia dos usuários, facilitando nos processos dentro da residência, desde algo simples como ter controle das luzes na iluminação do ambiente até ser programado o funcionamento de um determinado produto doméstico (mecânico, eletrônico e eletromecânico).

Um fator importante a ser averiguado é a imprevisibilidade, tendo em vista que a evolução da tecnologia não para no tempo, a sociedade da antiguidade não imaginava que grandes computadores seriam atualmente pequenos ocupando pouco espaço, além disso, que seriam integrados melhorando o funcionamento do sistema, sendo assim, a taxa de comunicação e de processamento tornaram-se mais eficazes.

Para se ter uma noção do mercado de automação residencial de 2015, abaixo na Figura 1 encontra-se gráficos, e na Figura 2 estão os dados mais detalhados em forma de tabela sobre os gráficos da Figura 1, tudo isso para melhorar o entendimento.

Figura 1 - Comparativo do mercado de automação residencial em 2015



Fonte: (AURESIDE, 2015)

Tabela 1 - Comparativo do mercado de automação residencial em 2015

	EUA	ALE	REINO UNIDO	FRANÇA	ESPAÑA	BRASIL (efetivo)	BRASIL (potencial)
TOTAL DE RESIDÊNCIAS (milhões)	134	38	27	25	17	63	63
RESIDÊNCIAS COM AUTOMAÇÃO (milhões)	24,0	6,5	4,4	4,5	2,6	0,3	1,9
RESIDÊNCIAS COM AUTOMAÇÃO (%)	18%	17%	16%	18%	15%	(*)	3%

(*) base estimada atual: 300 mil casas ou menos de 0,5% do total potencial estimado de 3% equivale a cerca de 1,9 milhões de casas

Fonte: (AURESIDE, 2015)

A automação residencial possui três níveis de interação, e estes são classificados como sistemas autônomos, sistemas integrados (controle centralizado) e sistemas complexos. Os sistemas que funcionam de forma independente, exercendo todas as funções para os quais foram projetados, sem a necessidade da intervenção de um controlador centralizado, são conhecidos como sistemas autônomos. Já os sistemas que funcionam exercendo todas as funções para os quais foram projetados, utilizando uma central inteligente programada por um microprocessador sem a necessidade da intervenção de um controlador centralizado, são conhecidos como sistemas integrados. E por último temos os sistemas complexos que precisa de um planejamento a partir do projeto de construção ou da reforma do imóvel, sendo necessário um cabeamento estruturado eficiente (SILVA, 2009).

2.2 MICROCONTROLADORES

Os microcontroladores são dispositivos pequenos, circuitos integrados que possuem um ou mais microprocessadores, memórias (voláteis e não voláteis) e periféricos de entrada e saída. Seu tamanho pequeno se dá através da necessidade do homem de criar dispositivos cada vez menores e econômicos para serem utilizados na área de automação, já que a sociedade cada vez mais necessitava de computadores melhores, rápidos e potentes, para resoluções de cálculos complexos, execução de tarefas (repetitivas ou não), e com o passar dos anos esse recurso foi sendo imerso nas indústrias, mercados, residências.

Os microcontroladores se diferenciam dos processadores, pois além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, o microcontrolador integra elementos adicionais em sua estrutura interna, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, EEPROM para armazenamento permanente de dados, dispositivos periféricos como conversores analógico/digitais (ADC), conversores digitais/analógicos (DAC), portas de entrada e saída digitais (I/O) para propósito geral (OLIVEIRA, 2017).

O microprocessador também recebe informações do próprio produto no qual está embarcado, e controla o aparelho mandando sinais para diversos componentes do mesmo. É

também, de certa forma, resistente, podendo ser aplicado em condições mais extrema onde computadores normais não conseguiriam (AGUILAR, 2008).

O menor custo do Arduino se dá devido a este utilizar materiais que em sua maioria são componentes produzidos em larga escala e baratos, além disso, dependendo do microcontrolador ele varia a frequência do clock de acordo com o funcionamento da aplicação do sistema, por exemplo, para se ter um controle na taxa de processamento de dados é preciso ter um controle da velocidade do clock, aumentando ou diminuindo de acordo com a necessidade, sendo ambas (frequência e taxa de processamento) dependentes proporcionais, quanto maior for ela, mais dados temos e consequente maior será a energia gasta, e o contrário também prevalece, isto é, quanto menor dados, menor será o gasto de energia.

Com o surgimento de sistemas embarcados, o microcontrolador não perdeu espaço no mercado, pelo contrário, acabou ganhando espaço devido sistemas embarcados utilizarem a junção de hardware e software, desse modo, pode-se pensar que a junção de hardware com software equivale a um computador, porém existe um detalhe a ser mencionado, pois computadores são máquinas de multi objetivos, enquanto os sistemas embarcados são projetados para atuar sobre um problema, ter controle de processos, dimensionando seus recursos a um domínio de objetivo bem menor, ou mesmo simples.

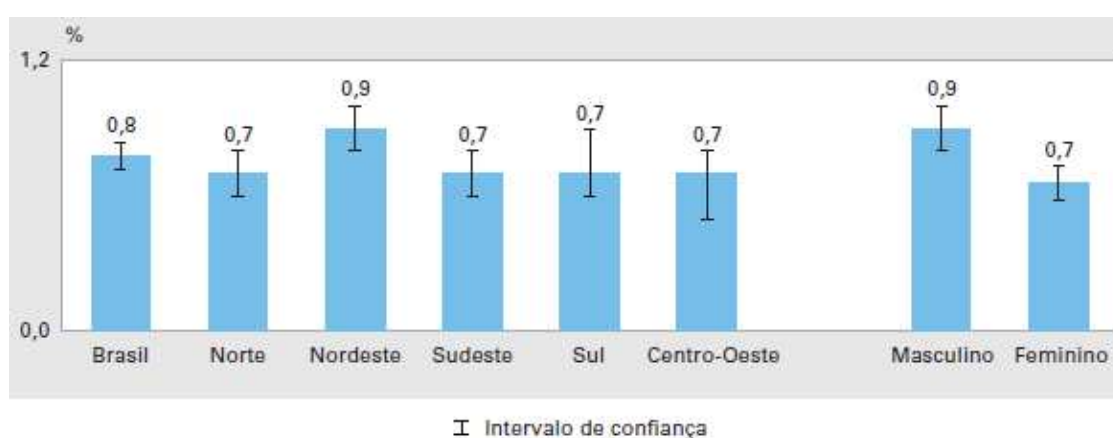
2.3 IBGE (Instituto Brasileiro de Geografia e Estatística)

De acordo com a PNS (Pesquisa Nacional de Saúde) em 2013 foram estimados dados a respeito de quatro tipos de deficiências: intelectual, física, auditiva e visual. Além disso, estimou 200,6 milhões de pessoas residentes em domicílios particulares permanentes, sendo desse total, 6,2% possuía pelo menos uma das quatro deficiências citadas anteriormente. Vale ressaltar que da Figura 2 até a Figura 7 a indicação do intervalo de segurança é de 95%.

A porcentagem de população que desde o nascimento possui deficiência intelectual, com limitações associadas a duas ou mais áreas de habilidades adaptativas é de 0,5%, e 0,3% delas adquiriram devido a doença ou acidente. Dentre todas as deficiências investigadas, essa foi a única em que o percentual de pessoas que nasceu com ela foi superior ao percentual de pessoas que a adquiriram por doença ou acidente.

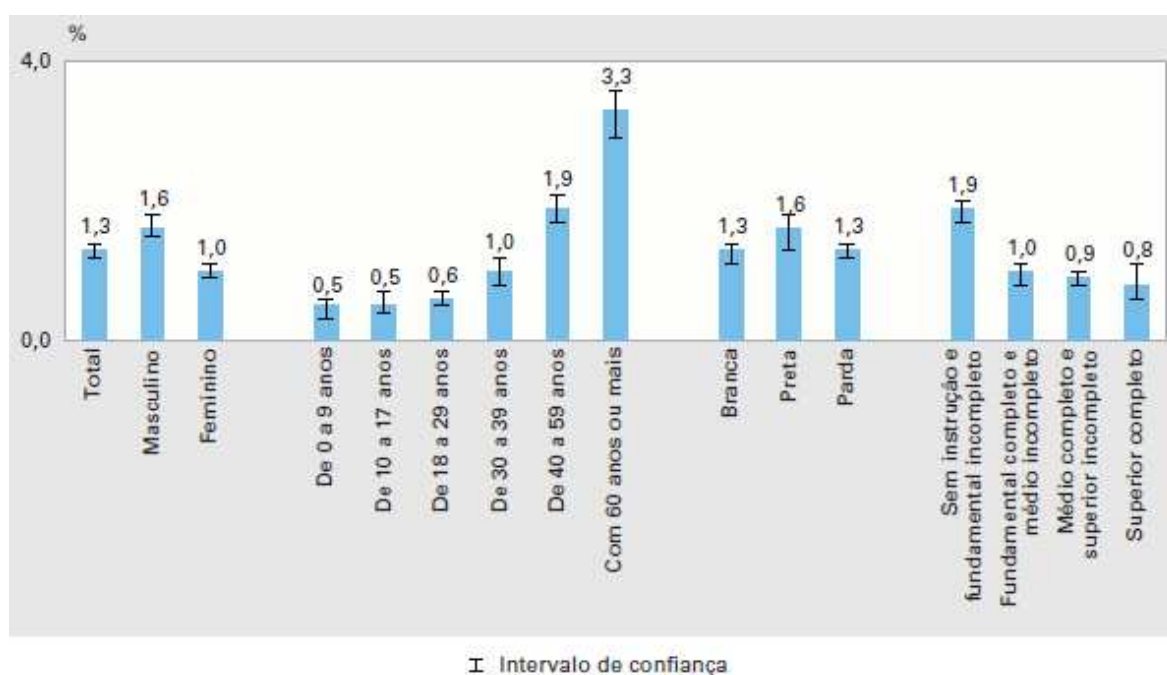
Dessa pesquisa feita em 2013 no Brasil, 1,3% da população declarou possuir deficiência física, tendo comprometimento das funções neurológicas, sensorial e/ou físicas devido existir alteração completa ou parcial de alguma parte do ser humano, sendo que dentro desse 1,3%, a porcentagem para os homens são de 1,6% e para as mulheres é de 1,0%. Na Figura 3 pode ser analisado pelos gráficos de barra que o número de casos aumenta de acordo com uma determinada faixa etária, tendo mais destaque a partir dos 60 anos, além de a maioria ser da cor/raça preta e não ter instrução e fundamental incompleto.

Figura 2 - Proporção de pessoas com deficiência intelectual



Fonte: (IBGE, 2013)

Figura 3 - Proporção de pessoas com deficiência física



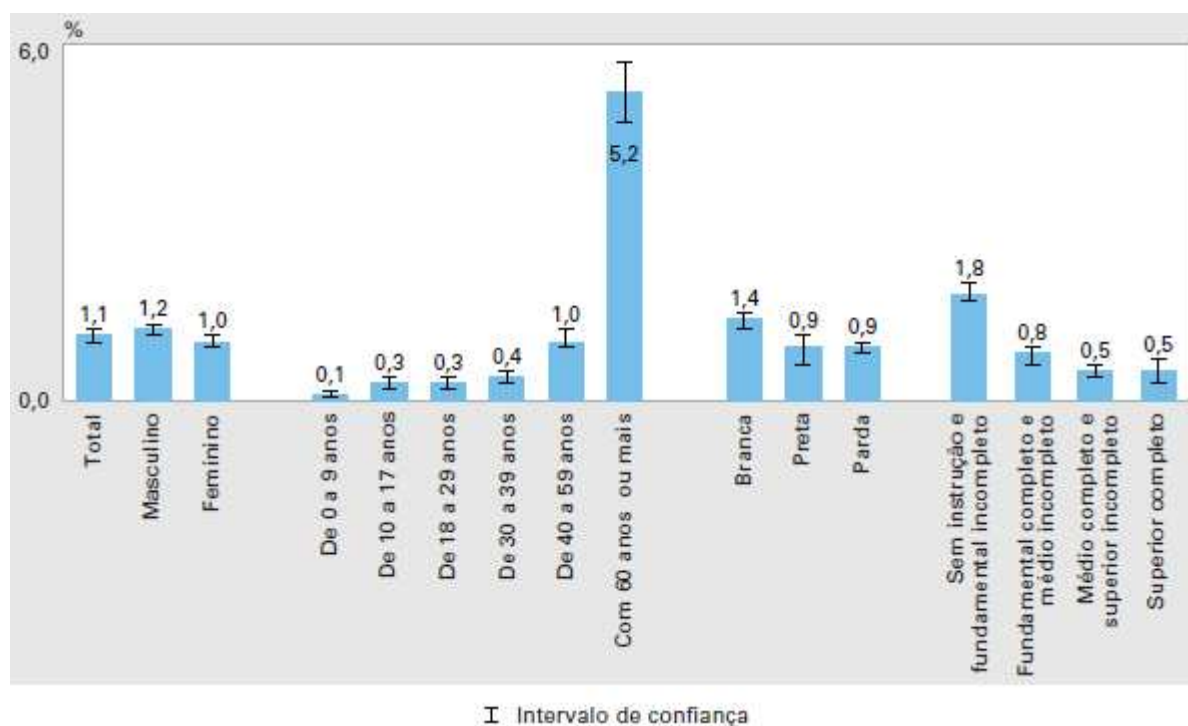
Fonte: (IBGE, 2013)

Ainda sobre a pesquisa da PNS, as características abaixo foram consideradas como caso de deficiência auditiva:

- surdez nos dois ouvidos;
- surdez em um ouvido e audição reduzida no outro;
- audição reduzida de ambos os ouvidos;

Na Figura 4 nota-se que as características das pessoas com deficiência auditiva em relação ao da Figura 3 são as mesmas, diferenciando apenas nas porcentagens, onde o sexo masculino continua com a porcentagem maior do que o feminino, além de pessoas a partir dos 60 anos ter uma frequência com esse tipo de deficiência, sem instrução ou com fundamental incompleto. Neste caso é perceptível a diferença na porcentagem entre a cor/raça, sendo a proporção para os brancos de 1,4% para os pretos e pardos de 0,9% em ambos.

Figura 4 - Proporção de pessoas com deficiência auditiva



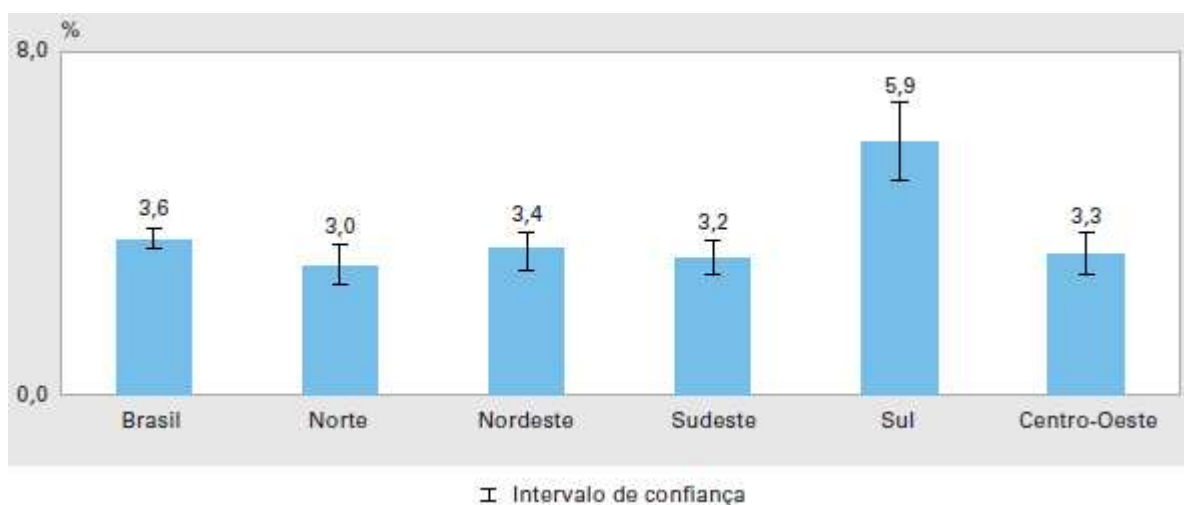
Fonte: (IBGE, 2013)

Segundo a PNS, as características abaixo foram consideradas como caso de deficiência visual:

- cegueira de ambos os olhos;
- cegueira de um olho e visão reduzida do outro;
- cegueira de um olho e visão normal do outro;
- baixa visão de ambos os olhos;

Dos tipos das deficiências investigadas pela PSN em 2013 no Brasil, a que obteve mais destaque na população foi à deficiência visual, com proporção de 3,6%. Na Região Sul do Brasil foi o local com mais destaque com 5,9%. Na Figura 5 podemos averiguar os dados apresentados nesse parágrafo com os demais gráficos das Figuras 6 e 7.

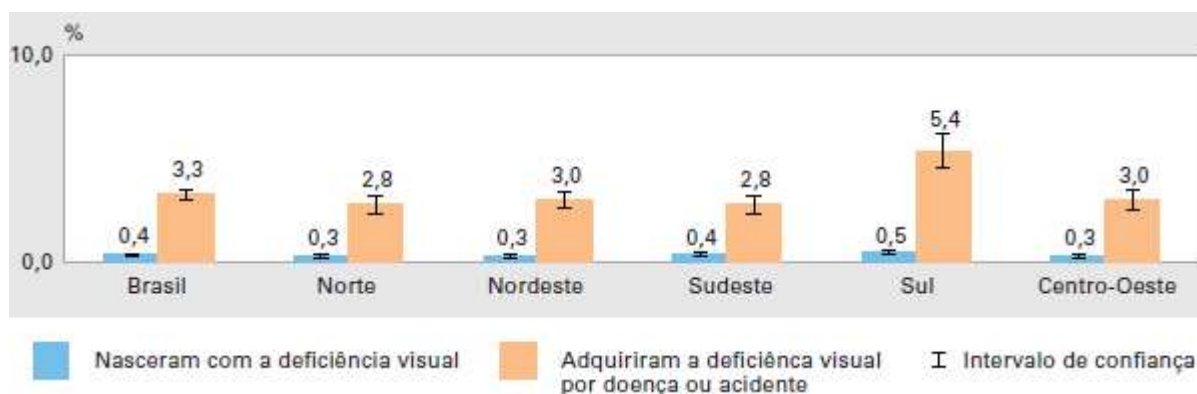
Figura 5 - Proporção de pessoas com deficiência visual



Fonte: (IBGE, 2013)

A que registrou a maior proporção estimada, dentre as quatro deficiências, com maior proporção estimada para as pessoas que o adquiriram por doença ou por acidente foi a de deficiência visual com 3,3%, enquanto, as que possuíam desde o nascimento eram de 0,4%. Não foi notória uma diferença significativa entre as grandes regiões para a população que tinha a deficiência desde o nascimento. Na região Sul obteve a maior proporção de pessoas que adquiriram a deficiência visual por doença ou acidente, com 5,4%.

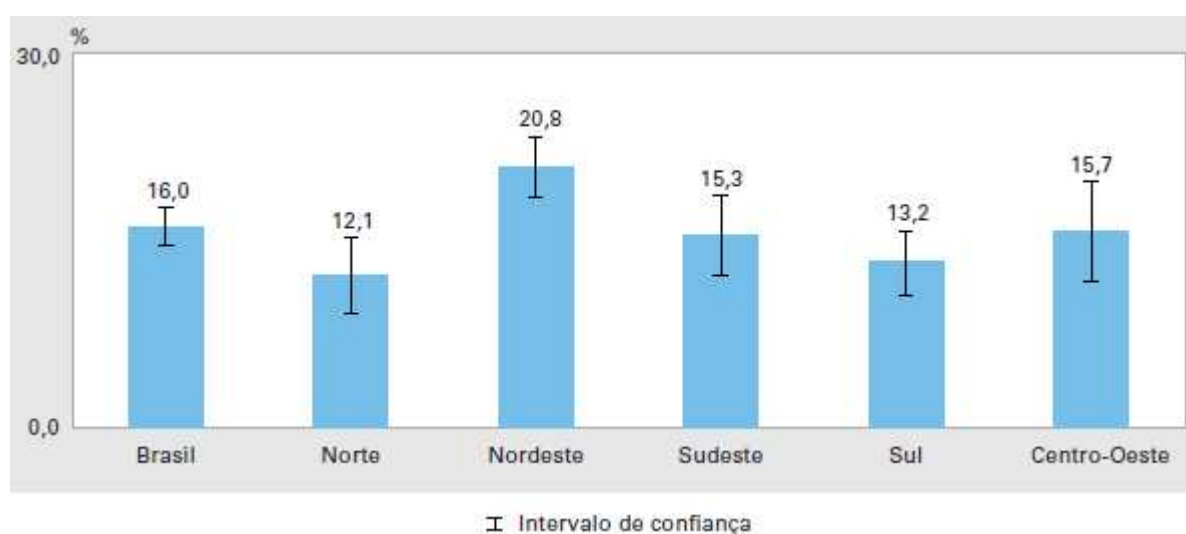
Figura 6 - Proporção de pessoas que nasceram com deficiência visual e proporção de pessoas que adquiriram



Fonte: (IBGE, 2013)

Sabendo que em 2013, demonstrado pela Figura 7, a menor proporção de limitação encontrada nas quatro deficiências investigadas, foi o da população com deficiência visual que apresentou grau intenso ou muito intenso de limitações ou não conseguia realizar as atividades habituais, sendo notório com 16,0%. No caso da região Nordeste tem 20,8%, sendo a única que estava com proporção acima da média nacional observada para esse indicador. Uma informação importante é que 4,8% das pessoas que possuíam deficiência visual, frequentavam o serviço de reabilitação, sendo o menor percentual estimado nas quatro deficiências pesquisadas.

Figura 7 - Proporção de pessoas com grau intenso de limitações ou que não conseguem fazer as atividades habituais



Fonte: (IBGE, 2013)

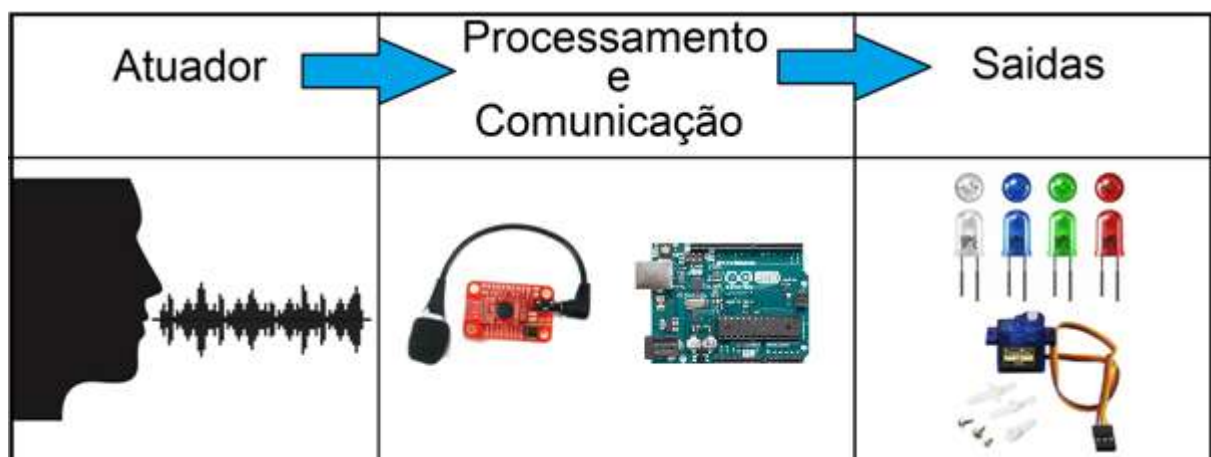
3. MODELO PROPOSTO

O modelo proposto e seu esquema encontram-se logo abaixo na Figura 8, sendo elaborado um projeto que utiliza o microcontrolador na automação residencial para suprir algumas necessidades de pessoas especiais. A voz emite ondas que através do microfone é recebida e convertida em dados tendo uma comunicação entre a placa de reconhecimento de voz e o arduino, esses dados serão processados através da biblioteca disponibilizada pelo fabricante.

O processo é feito pelas gravações de palavras chaves curtas, não podendo passar de 1,5 segundos a duração de cada uma, limitação vista nas especificações da Tabela 2. Após ser processada e comunicada, se a palavra for reconhecida, uma determinada ação é correspondida, ligando ou desligando o LED que está representado as lâmpadas caseiras, e no caso do servo motor representará um portão automático, sendo rotacionado de 0 a 90 graus, sentido horário e anti-horário, além do buzzer representar o alarme emitindo o efeito sonoro.

A linguagem C++ foi usada no ambiente de desenvolvimento integrado (IDE) do próprio Arduino para programar a lógica, além de ser uma linguagem orientada a objeto, o que facilita nos trabalhos futuros para a melhoria do protótipo. Vale lembrar que é uma linguagem antiga, entretanto, ainda vem sendo utilizada no mercado, justamente pela facilidade para o aprendizado, criando projetos, protótipos simples até mais complexos.

Figura 8 - Modelo simples do projeto



Fonte: Autoria própria

3.1 DESCRIÇÕES DOS COMPONENTES DO SISTEMA

A partir desta seção teremos as descrições dos componentes utilizados no sistema, com suas respectivas tabelas de especificações técnicas retiradas dos datasheet e fotos para o reconhecimento de cada um, devido na Figura 23 termos cada um deles com ligações físicas.

3.1.1 Arduino UNO

O Arduino Uno é uma placa de microcontrolador baseada no ATmega328P que possui quatorze pinos de entrada/saída digitais, dos quais seis podem ser utilizados como saídas PWM, além de conter seis entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, uma tomada de energia, um cabeçalho ICSP e um botão de reinicialização. Ele contém tudo o que é necessário para suportar o microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligar com uma bateria externa AC (corrente alternada) para DC (corrente contínua). Você pode mexer com seu UNO sem se preocupar demais em fazer algo errado, o pior caso, você pode substituir o chip por alguns reais e começar de novo. "Uno" significa “um” em italiano e foi escolhido para marcar o lançamento do Software Arduino (IDE) 1.0. Em relação ao Uno e a versão 1.0 do Software Arduino (IDE) foram as versões de referência para versões mais recentes a serem desenvolvidas pela mesma empresa. A placa Uno é a primeira em uma série de placas USB Arduino e o modelo de referência para a plataforma Arduino. (ARDUINO, 2018)

Figura 9 - Visão superior da placa Arduino UNO



Fonte: (ARDUINO UNO, 2018)

Tabela 2 - Especificações técnicas do Arduino Uno

Microcontrolador	ATmega328P
Tensão Operacional	5V
Tensão de Entrada (recomendada)	7-12V
Tensão de Entrada (limite)	6-20V
Pinos de E / S digitais	14 (dos quais 6 oferecem saída PWM)
Pinos de E / S PWM digitais	6
Pinos de entrada analógicos	6
Corrente DC por pino I / O	20 mA
Corrente DC para Pin 3.3V	50 mA
Memória flash	32KB, dos quais 0.5 kB são utilizados na inicialização
SRAM	8KB
EEPROM	1KB
Velocidade de Clock	16MHz
LED_BULTIN	13
Comprimento	68,6 mm
Largura	53,4 mm
Peso	25g

Fonte: (ARDUINO UNO, 2018)

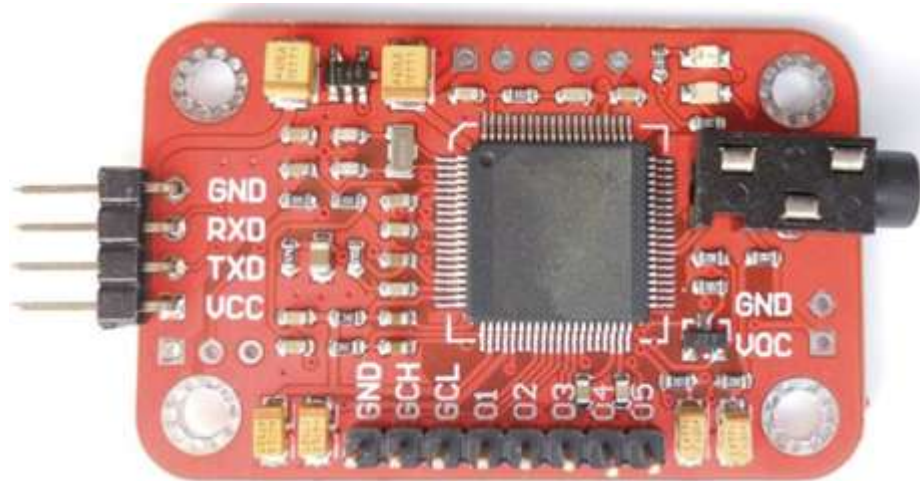
3.1.2 Voice Recognition Module V3

O Módulo de reconhecimento de voz ELECHOUSE é um painel de reconhecimento de fala compacto e de controle fácil. Este produto é um módulo de reconhecimento de voz dependente do falante. Ele suporta até 80 comandos de voz, sendo dividido três grupos, e é capaz de aceitar no máximo 7 comandos de voz funcionando ao mesmo tempo. Qualquer som pode ser treinado como comando. Os usuários precisam treinar o módulo primeiro, usando a biblioteca pronta disponível no github, e o site para download está disponível no manual de usuário do módulo. Após o treinamento ele reconheça qualquer comando de voz. Esta placa possui 2 formas de controle: Porta Serial (função completa), Pinos de Entrada Geral (parte da função). Os pinos de saída na placa poderiam gerar vários tipos de ondas enquanto o comando de voz correspondente era reconhecido. (ELECHOUSE, 2018)

Tabela 3 - Especificações técnicas do Voice Recognition Module V3

Tensão Operacional	5V
Tensão de Entrada (recomendada)	4.5-5.5V
Corrente DC	<40 mA
Comprimento	50 mm
Largura	31 mm
Suporte máximo de comando de voz	80 comandos com 1500 ms cada
Precisão de reconhecimento	99% (sob ambiente ideal)
Interface Analógica	conector de microfone mono canal de 3.5 mm

Fonte: (ELECHOUSE, 2018)

Figura 10 - Visão superior do Voice Recognition Module V3

Fonte: (ELECHOUSE, 2018)

3.1.3 Sensores e Atuadores

A seguir temos os sensores e atuadores utilizados durante o desenvolvimento do projeto acadêmico, sendo considerados sensores todos os equipamentos que recebem do meio externo o comando de ação, no caso o microfone pode ser considerado um sensor por receber uma onda sonora, sendo esta produzida pela boca, enquanto, que os atuadores são os leds e o servo motor, justamente por fazer uma ação de acordo com uma determinada entrada, a voz recebida pelo microfone.

3.1.3.1 LED (Diodo Emissor de Luz)

É um diodo semicondutor que quando energizado emite luz, é uma luz estreita que é produzida pelas interações energéticas do elétron, sendo esse processo chamado de eletroluminescência, além disso, é importante para fixar a ideia de LED (Light Emitting Diode), entender o funcionamento da junção p-n de um semicondutor, que tem como atividade permitir a emissão de luz. Esta junção é responsável por permitir somente a passagem de corrente positiva. Para entender melhor essa passagem, um exemplo simples é imaginar uma onda senoidal entrando em um LED, ele cortará a parte negativa permitindo só a corrente positiva (INFOESCOLA, 2018).

Figura 11 - Visão superior e de perfil dos LED



Fonte: (MUNDODAELETRICA, 2018)

3.1.3.2 Microfone

Com base em um princípio da Física chamado indução magnética, foi possível construir esse equipamento que nos permite falar em público com intensidade de voz maior. Podemos dizer então que o microfone é um dispositivo eletromecânico que transforma as vibrações mecânicas em corrente elétrica, além disso, no microfone existe um diafragma, protegido por uma cobertura perfurada que permite a passagem do som e recebe as vibrações sonoras longitudinais. Quando alguma vibração do ar atinge esse diafragma, ele a transmite para um sistema elétrico, que pode ser uma bobina móvel, um capacitor ou grãos de carvão, dependendo do tipo de microfone utilizado (BRASILESCOLA, 2018).

Figura 12 - Visão superior do microfone acoplado ao módulo de reconhecimento de voz



Fonte: (SMARTKITS, 2018)

3.1.3.3 Potenciômetro

São dispositivos elétricos simples que possuem sua resistência alterada de forma mecânica, muitos usados para controlar volume de um som, brilho da tela, sintonizar estação de rádio, entre outros. O seu mecanismo de funcionamento consta em utilizar os três terminais, normalmente os da extremidade fazem parte da alimentação (GND e VCC), e o terminal central responsável por mandar a resistência desejada, devido a regulagem manual do cursor em cima de uma trilha circular de carvão, que no caso utilizado do projeto, o potenciômetro varia de 0 a 10000 ohms e também considerado do tipo linear.

Figura 13 - Visão de perfil do potenciômetro de 10K



Fonte: (SMARTKITS, 2018)

3.1.3.4 LCD (Display de Cristal Líquido)

São bastante utilizados em aparelhos eletrônicos que precisam da visualização do usuário, a sua tecnologia conhecida como LCD (Liquid Crystal Display), que em português traduzida para Display de Cristal Líquido, é composta por um líquido polarizador de luz comprimido, controlado por via elétrica, dentro de lâminas transparentes e cada uma delas possui pontos de eletricidade que fazem o campo elétrico se unir ao líquido no interior (TECHTUDO, 2018).

No projeto foi utilizado display LCD 16x1, isso significa que possui uma matrix de 16 colunas e 1 linha, sendo necessário mostrar na tela somente a informação principal para o usuário, já que se fosse preciso algo mais avançado como alterar a fonte, tamanho da fonte, cor do fundo, cor da letra, teria que utilizar um display mais adequado para objetivo desejado, não sendo o caso deste projeto.

Figura 14 - Visão superior do display LCD 16x1



Fonte: (SMARTKITS, 2018)

3.1.3.5 Buzzer

É um componente eletrônico com dois terminais para alimentação (GND e VCC), que internamente o buzzer possui três camadas, sendo uma de cristal piezoelétrico e duas de metal, e mesmo o componente sendo pequeno é capaz de emitir um som e para que funcione será necessário que receba uma tensão a uma determinada frequência, dessa maneira, a célula piezoelétrica que tem dentro dele irá vibrar nesta frequência, produzindo ondas sonoras. O buzzer pode ser notado em aparelhos que utilizam o efeito sonoro simples, também conhecidos como som de Beep.

Figura 15 - Visão superior e de perfil do buzzer



Fonte: (MUNDODAELETRIA, 2018)

3.1.3.6 Micro Servo 9g SG90

Figura 16 - Visão superior e de perfil do micro servo 9g SG90



Fonte: (MUNDODAELETRIA, 2018)

O servo pode girar aproximadamente 180 graus (90 em cada direção), é pequeno e leve, tendo apenas nove gramas, dispositivo de malha fechada com alta potência de saída e funcionando como os outros servos de maior porte, mas com suas próprias limitações por ser menor. Pode ser usado qualquer código no servo, hardware ou biblioteca para controlar esses servos. Possui três fios e cada um possui sua função, o vermelho para alimentação, o amarelo para sinais e o preto para o terra.

Tabela 4 - Especificações técnicas do Miro Servo 9g SG90 Tower Pro

Tensão Operacional	5V
Tensão de Entrada (limite)	3,0-7,2V
Ângulo de Rotação	180 graus
Velocidade	0,12s/60 graus (4,8V) sem carga
Torque	1,2kg.cm (4,8V) e 1,6kg.cm (6,0V)
Temperatura de Operação	-30C até + 60C
Tipo de engrenagem	Nylon
Tamanho do cabo	245 mm
Dimensões	32 x 30 x 12 mm
Peso	9g

Fonte: (FILIPEFLOP, 2018)

4. DESENVOLVIMENTO DO MODELO PROPOSTO

Neste capítulo encontram-se as informações necessárias para o desenvolvimento do modelo proposto, incluindo figuras.

4.1 CONSTRUÇÃO DA MAQUETE

Para a construção da maquete foram avaliados dois tipos de materiais principais, o isopor e o papelão, ambos são materiais leves e de custo barato, mas o escolhido para fazer parte de todo o projeto foi o isopor, por ter uma melhor resistência em relação a fixação dos componentes eletrônicos em cada cômodo, e do servo motor na área externa. Além disso, um detalhe a ser observado na Figura 17 são os posicionamentos dos LEDs, todos foram colocados na base, para que durante a demonstração de seu funcionamento fosse mais fácil a visualização de cada um, pois se tivesse colocado no teto, não teria como ver com facilidade se o comando tinha funcionado, mesmo o teto sendo uma parte removível.

Na Figura 18 é notória uma construção de uma sub-base, que foi utilizada apenas para colocar o arduino e suas fiações, um meio de melhorar o aspecto visual do projeto, mostrando somente o essencial. Também é perceptível na Figura 17 que o microfone está isolado em uma das extremidades, enquanto que no outro extremo temos o LCD, o potenciômetro. Eles foram colocados nessa ordem para que o microfone evite receber ruído por parte do buzzer ao ser ativado, e o LCD para ser visualizado pelo usuário o que está sendo ativo ou não pelo comando de voz. Outra maneira de visualizar é através da interface do ambiente de programação localizado na Figura 26 no capítulo de teste de hardware e software.

No desenvolvimento da construção do projeto foi preciso produzir um modelo de placa ideal de circuito integrado, visto na Figura 19 e na Figura 21, desenvolvido no software da EAGLE para evitar a utilização de uma protoboard, já que um dos problemas encontrados foi a quantidade de cabos a ser utilizado. Já na Figura 20 e na Figura 22 temos a produção placa real que foi utilizada.

Figura 17 - Visão Frontal da Maquete



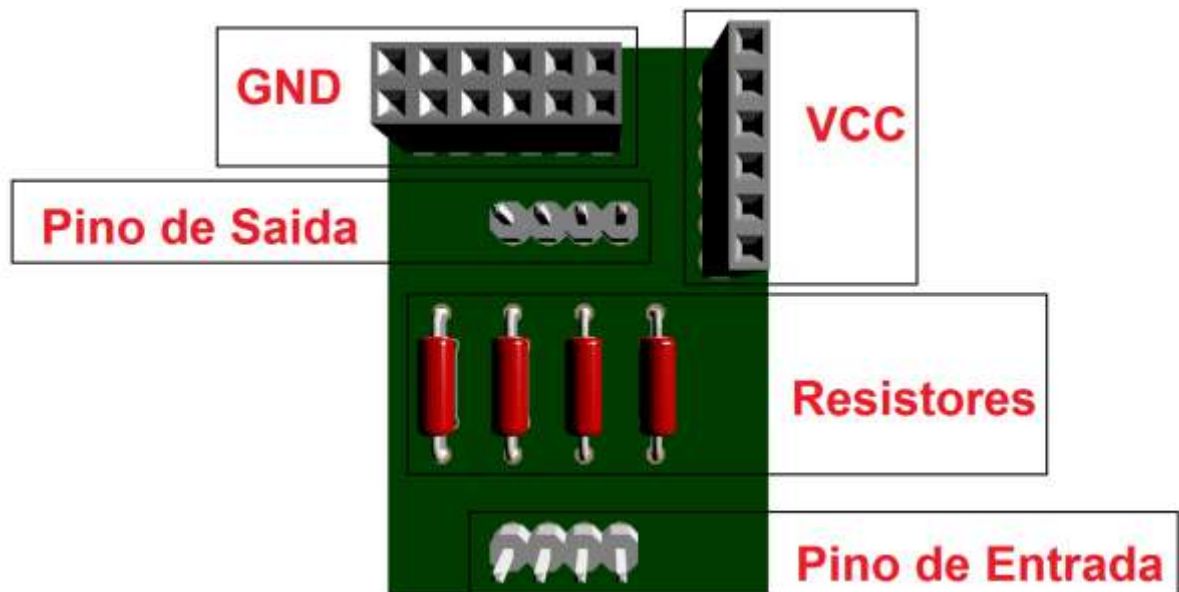
Fonte: Autoria própria

Figura 18 - Visão Lateral da Maquete



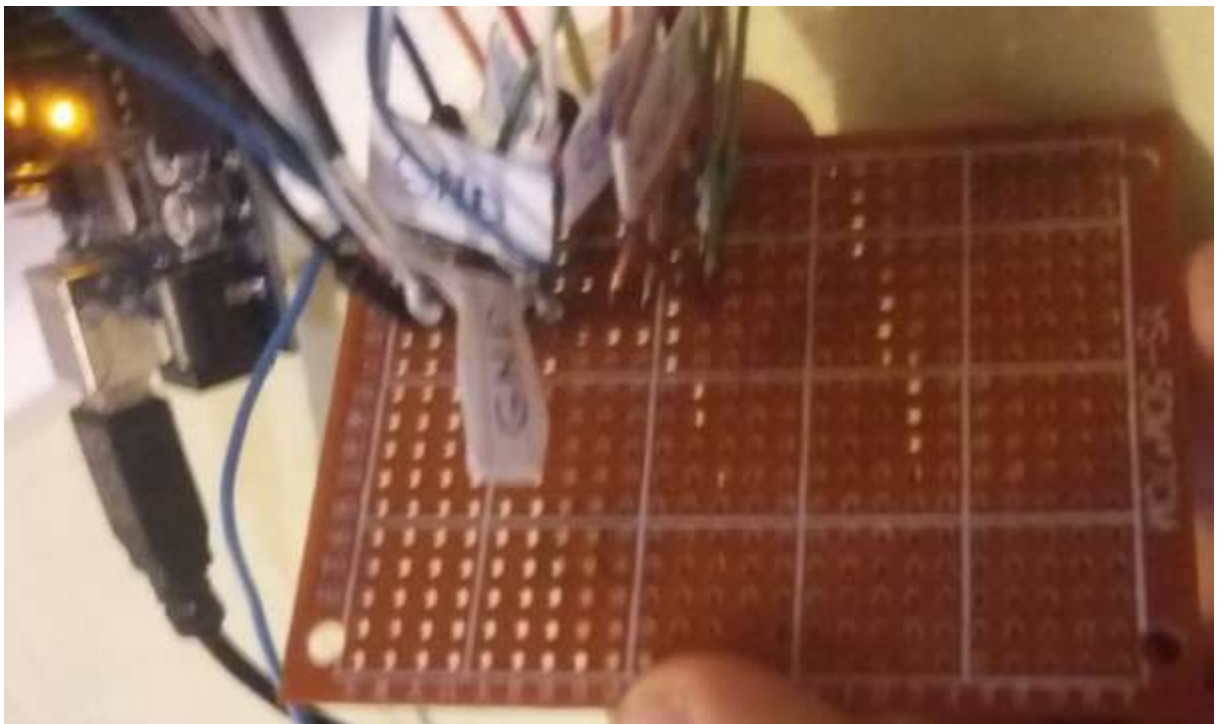
Fonte: Autoria própria

Figura 19 - Frente da Placa Ideal



Fonte: Autoria própria

Figura 20 - Construção da Maquete



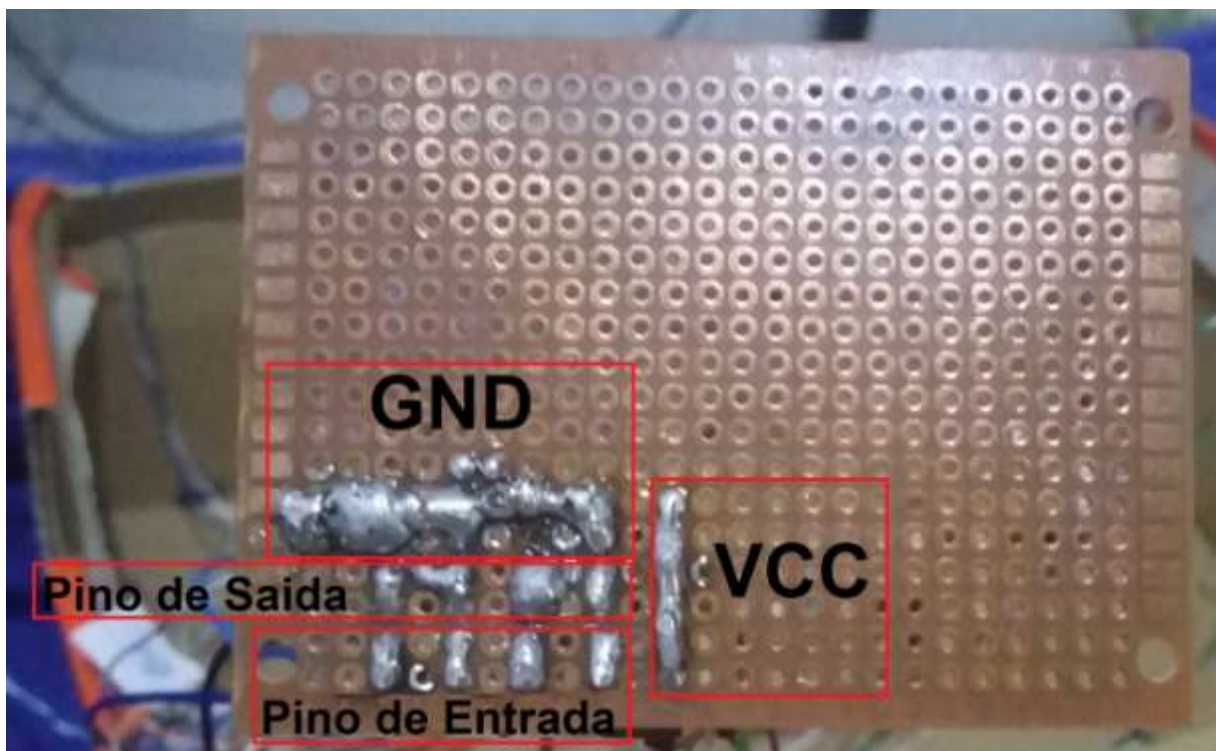
Fonte: Autoria própria

Figura 21 - Costa da Placa Ideal



Fonte: Autoria própria

Figura 22 - Costa da Placa Real

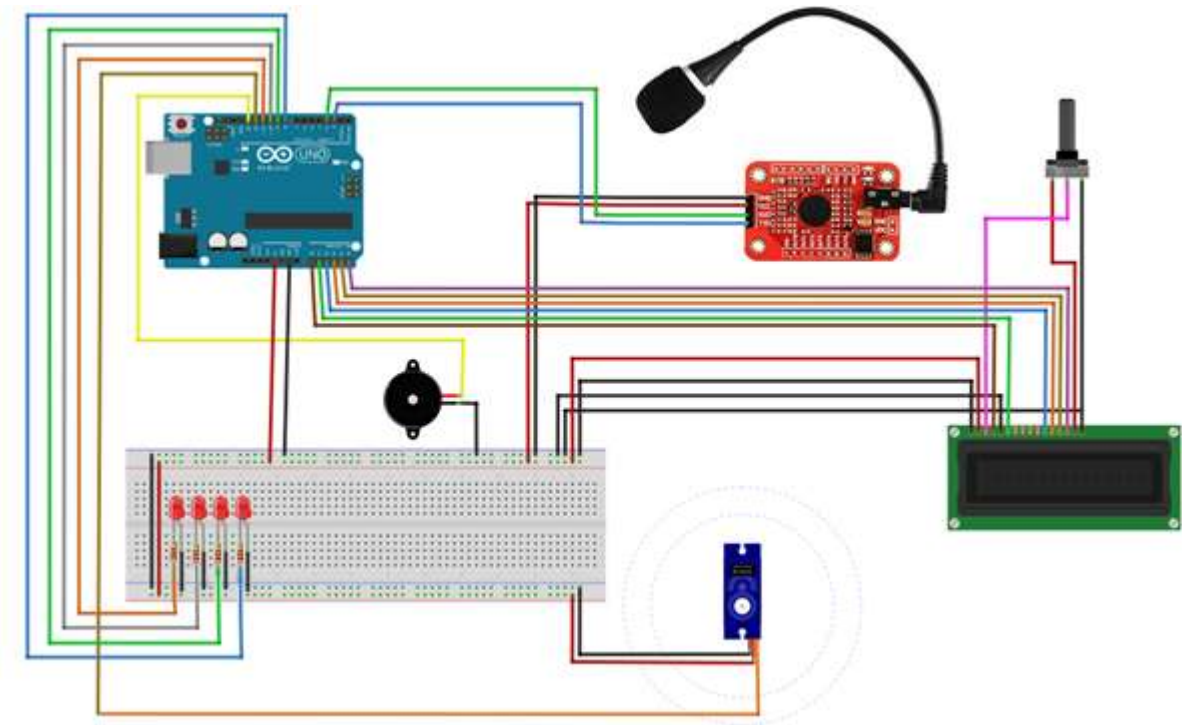


Fonte: Autoria própria

4.2 LIGAÇÕES FÍSICAS

As ligações físicas entre o arduino, atuador e sensores foram feitas em seus respectivos cômodos após o desenvolvimento da maquete, em seguida foram testados cada um dos componentes para ter a garantia que estavam tudo ocorrendo de acordo com o programado, também utilizando o multímetro quando necessário para averiguar a voltagem e corrente elétrica, caso contrário, as devidas alterações seriam feitas durante o progresso do projeto. Além dos testes, cada fio recebeu uma etiqueta para identificá-los melhor, diminuindo as chances de fazer as conexões de forma errônea, facilitando mais o trabalho em sua montagem devido a grande quantidade de fios a serem utilizados. Na Figura 23 está o esquema das ligações físicas entre cada um dos componentes, ressaltando que as cores dos cabeamentos desta figura são apenas ilustrativas, importando somente as ligações entre eles. Um fator não mostrado na figura abaixo, entretanto mencionado na Tabela 1 – Especificações técnicas do Arduino Uno, é que precisa de uma fonte de alimentação, no caso foi utilizado a própria porta serial USB do notebook, por ser de 5 Volts.

Figura 23 - Ligações físicas do projeto



Fonte: Autoria própria

Na Tabela 5 temos uma relação entre o cômodo e um determinado pino do arduino, demonstrando que todos são independentes. Além disso, na Tabela 6 temos as ligações do buzzer e do display LCD.

Tabela 5 - Disposição dos pinos em relação aos cômodos

Sensor	Pino/Arduino	Cômodo
Microfone	4	Sala
	5	Cozinha
	6	Quarto
	7	Banheiro
	8	Portão

Fonte: Autoria própria

Tabela 6 - Disposição dos pinos em relação aos outros componentes

Sensor	Pino/Arduino	Componente Atuador
Microfone	9	Buzzer
Potenciômetro	V0	LCD
Microfone	A0	LCD
	A1	LCD
	A2	LCD
	A3	LCD
	A4	LCD
	A5	LCD

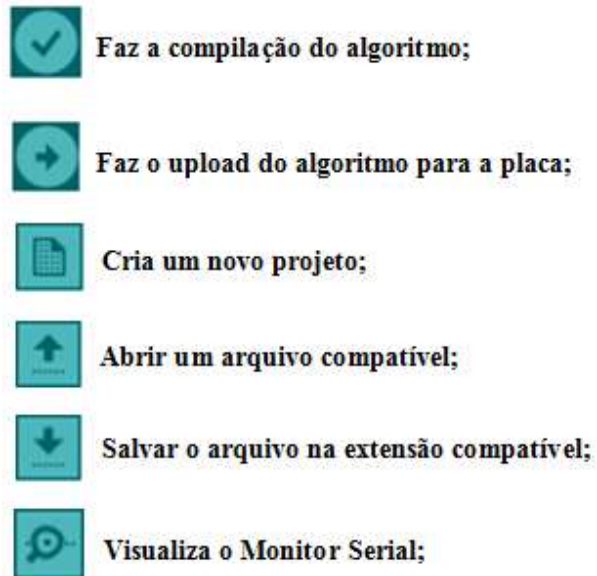
Fonte: Autoria própria

4.3 DESENVOLVIMENTO DO ALGORITMO NO ARDUINO

O ambiente de programação do Arduino possui uma interface amigável ao programador, que pode ser observado as legendas dos botões na Figura 24 e a interface da IDE na Figura 25. Sabendo que é uma plataforma de hardware livre e de fácil utilização, mesmo que uma pessoa tenha apenas o conhecimento básico em programação, isso não tornará um obstáculo, devido ser acessível pelo próprio site do Arduino o funcionamento de

cada função, biblioteca, além de que nas mídias sócias conterem materiais de como utilizar, justamente por ser uma ferramenta considerada de baixo custo econômico que pode ter soluções viáveis para muitos problemas de acordo com suas limitações.

Figura 24 - Legenda dos botões da interface



Fonte: Autoria própria

Figura 25 - A interface da IDE



Fonte: Autoria própria

Os microcontroladores geralmente são programados através de um programador, a menos que você tenha um firmware no seu microcontrolador que permita a instalação de novos firmwares sem a necessidade de um programador externo, isso é chamado de bootloader. Com o bootloader na placa Arduino não é preciso programar o microcontrolador, ou seja, a própria placa Arduino é utilizada como uma gravadora para o microcontrolador, além de possibilitar uma comunicação direta via porta USB entre o computador e a plataforma de prototipagem eletrônica.

Após a instalação adequada do software que é disponibilizado no site do Arduino, tem que ser configurado a porta de comunicação desejada com a placa sendo alimentada pelo cabo USB, entretanto a porta COM 3 não é padrão, podendo sofrer alteração dependendo da máquina utilizada, onde ocorre a variação apenas do número que a identifica, e para saber qual porta esta sendo utilizada, deve-se ir em gerenciador de dispositivos se o sistema operacional a ser utilizado for o Windows, caso contrário, procurar no gerenciador responsável pelas portas do sistema operacional que esta sendo utilizado pelo usuário.

Utilizando o código exemplo de treinamento de voz disponível no manual do módulo de reconhecimento de voz, é possível de forma simples treinar e gravar sete comandos de voz, de acordo com o exemplo utilizado. Após o treinamento e gravação, cada assinatura de voz é armazenada no próprio módulo, não precisando ser gravada também na placa do Arduino, deste modo, tendo o armazenamento todo livre para iniciar a escrever o algoritmo do projeto.

No escopo da interface devemos incluir as bibliotecas que serão utilizadas durante o decorrer da programação, sendo elas a do módulo de reconhecimento de voz, do LCD e a do servo motor, além de declarar as variáveis, criar os módulos das funções e as próprias funções responsáveis por cada parte do algoritmo, processo conhecido por deixar o código mais leve e limpo possível, seguindo também o padrão de um código bem estruturado.

As funções criadas e para que servem estão logo abaixo:

- voidsomLigar(): é emitido um único som de duração curta (BEEP) pelo buzzer, quando é ativado algum cômodo;
- voidsomDesligar(): é emitido dois sons de duração curta (BEEP) pelo buzzer, quando é desativado algum cômodo;

- voidmsgTodosOn(): é ativado quando todos os cômodos estão ativados, com excessão do portão;
- voidmsgTodosOff(): é ativado quando todos os cômodos estão desativados, incluindo o portão;
- voidmsgPortao(): é ativado quando o portão está aberto e a mensagem enviada ao LCD;
- voidmsgSala(): é ativado quando o led da sala está ligado e a mensagem enviada ao LCD;
- voidmsgCozinha(): é ativado quando o led da cozinha está ligado e a mensagem enviada ao LCD;
- voidmsgQuarto(): é ativado quando o led do quarto está ligado e a mensagem enviada ao LCD;
- voidmsgBanheiro(): é ativado quando o led da banheiro está ligado e a mensagem enviada ao LCD;
- voidmsgLCD(): recebe todas as mensagens enviadas por funções ativas que iniciem por “voidmsg” e mostra no display LCD para o usuário;

5. TESTES DE HARDWARE E SOFTWARE

Foi uma etapa exaustiva por envolver todos os componentes, sendo testado cada um em módulos, e conseqüentemente após integrá-los, foi realizado um novo teste para saber se estavam todos funcionando de acordo com a programação realizada. Das Tabelas 7 a Tabela 17, encontram-se todas as possibilidades realizadas durante a etapa de teste de hardware, variando de um cômodo até todos os cômodos sendo ativados, incluindo o portão, além de ajudar no desenvolvimento do algoritmo para mostrar mensagem no display LCD e no serial monitor visto na Figura 26. Um fator a ser observado durante esta etapa foi a capacidade de o microfone captar a onda sonora da palavra treinada por comando de voz a uma distância curta de 50 centímetros, e caso tenha um ruído, a capacidade de ser reconhecida pelo módulo será dificultada.

Tabela 7 - Teste somente para uma entrada

LOCAL	SAIDA
PORTÃO	OK
SALA	OK
COZINHA	OK
QUARTO	OK
BANHEIRO	OK

Fonte: Autoria própria

Tabela 8 - Teste somente para duas entradas, com portão

LOCAL		SAIDA
PORTÃO	SALA	OK
PORTÃO	COZINHA	OK
PORTÃO	QUARTO	OK
PORTÃO	BANHEIRO	OK

Fonte: Autoria própria

Tabela 9 - Teste somente para três variáveis, com cozinha

LOCAL			SAIDA
COZINHA	QUARTO	BANHEIRO	OK

Fonte: Autoria própria

Tabela 10 - Teste somente para duas variáveis, com sala

LOCAL		SAIDA
SALA	COZINHA	OK
SALA	QUARTO	OK
SALA	BANHEIRO	OK

Fonte: Autoria própria

Tabela 11 - Teste somente para duas variáveis, com cozinha

LOCAL		SAIDA
COZINHA	QUARTO	OK
COZINHA	BANHEIRO	OK

Fonte: Autoria própria

Tabela 12 - Teste somente para duas variáveis, com quarto

LOCAL		SAIDA
QUARTO	BANHEIRO	OK

Fonte: Autoria própria

Tabela 13 - Teste somente para três variáveis, com portão

LOCAL			SAIDA
PORTÃO	SALA	COZINHA	OK
PORTÃO	SALA	QUARTO	OK
PORTÃO	SALA	BANHEIRO	OK

Fonte: Autoria própria

Tabela 14 - Teste somente para três variáveis, com sala

LOCAL			SAIDA
SALA	COZINHA	QUARTO	OK
SALA	COZINHA	BANHEIRO	OK

Fonte: Autoria própria

Tabela 15 - Teste somente para quatro variáveis, com portão

LOCAL				SAIDA
PORTÃO	SALA	COZINHA	QUARTO	OK
PORTÃO	SALA	COZINHA	BANHEIRO	OK

Fonte: Autoria própria

Tabela 16 - Teste somente para quatro variáveis, com sala

LOCAL				SAIDA
SALA	COZINHA	QUARTO	BANHEIRO	OK

Fonte: Autoria própria

Tabela 17 - Teste para todas as entradas

LOCAL					SAIDA
PORTÃO	SALA	COZINHA	QUARTO	BANHEIRO	OK

Fonte: Autoria própria

Figura 26 - Reconhecimento da palavra Sala

```

BIP
-----
VR Index      Group  RecordNum  Signature
0             NONE    0           sala
-----

Sala: 1
Cozinha: 0
Quarto: 0
Banheiro: 0
Portão: 0
Todos: 0
Ligados: sala
-----

BIP BIP
-----
VR Index      Group  RecordNum  Signature
0             NONE    0           sala
-----

Sala: 0
Cozinha: 0
Quarto: 0
Banheiro: 0
Portão: 0
Todos: 0
Desligados: Todos
-----

```

Fonte: Autoria própria

6. CONCLUSÕES E CONSIDERAÇÕES FINAIS

De acordo com o projeto proposto é notório que utilizando uma placa de Arduino com o módulo de reconhecimento de voz, e tendo conhecimento de eletrônica e programação o sistema pode ser utilizado tanto para pessoas com ou sem necessidades especiais, precisando a priori da voz e de um sistema de alimentação. Além disso, pode ser verificado na Tabela 18 o gasto de cada componente, o que deve ser notado particularmente é o preço da placa arduino e do modulo de reconhecimento de voz, somando ambos custará 150 reais, sendo esse um valor de preço acessível para as classes sociais devido a sua a usabilidade, em outras palavras, foi um projeto econômico e financeiramente viável.

Outro fator a ser observado no projeto é que ele pode ser utilizado em outras áreas, não limitando apenas a automação residencial, podendo ser também predial, considerando o fato de sempre haver uma voz e uma fonte de alimentação para os componentes eletrônicos. Além disso, é importante desmistificar que a área de automação é complexa, e mostrar que outras áreas podem ser integradas em projetos, no caso demonstrado foram as áreas de computação, automação e elétrica, todas elas funcionando em conjunto, claro que dependendo da atuação e da complexidade de cada projeto serão necessários mais conhecimento em cada área requisitada.

No comparativo de mercado de automação residencial em 2015 visto na Figura 1, percebe-se que nos países desenvolvidos a porcentagem de casas com automação é pouco em relação ao total de residências em cada país comparado, entretanto ao comparar o Brasil e cada um deles, é perceptível que o Brasil precisa melhorar bastante para alcançar uma porcentagem razoável. Sendo assim, o país precisa melhorar não apenas na produção de peças produzidas para diminuir o custo delas, mas também na educação, pois sem ela não seria possível ter ideias inovadoras para serem divulgadas e financiadas, facilitando o aumento da porcentagem de casas automatizadas.

Os objetivos propostos deste trabalho foram alcançados, a fala com a palavra chave é reconhecida se um ruído for baixo, o display LCD mostra a mensagem de qual led do cômodo está ativo (ligado), e o mesmo acontece para o portão. O buzzer indica através do efeito sonoro conhecido como BEEP se o led e/ou portão foi ligado (único beep) ou desligado (dois

beeps). Sendo assim se uma pessoa que tem deficiência visual ela pode falar a palavra chave e saber se o comando foi reconhecido através do efeito sonoro (BEEP) do buzzer. Caso a pessoa possua deficiência auditiva, ela pode reconhecer o comando através da mensagem do display LCD. E caso a pessoa possua tanto deficiência intelectual quanto física, ela pode reconhecer o comando por ambos os dispositivos, o display LCD e o buzzer. Desta maneira demonstrando que é possível incluir automação residencial para pessoas com necessidades especiais por meio de uma placa de prototipagem microcontroladora que no caso a utilizada foi Arduino, podendo variar entre outras placas desde que tenham no mínimo os mesmos requisitos da qual foi usada no projeto. Vale lembrar que essa é a primeira versão do sistema desenvolvido, portanto, melhorias devem ser feitas justamente por estamos na era da internet das coisas, onde tudo é acessível através da internet tendo conexões em conjunto com dispositivos eletrônicos, deste modo, aumentando a capacidade computacional e de comunicação quando conectado a internet

Tabela 18 – Custo do Projeto

Descrição do Item	Valor Unitário (R\$)	Unidade(s)	Valor Total (R\$)
Arduino UNO	50,00	1	50,00
Módulo Elechouse	100,00	1	100,00
Servo Motor	6,00	1	6,00
Display LCD	6,00	1	6,00
LED'S	0,10	4	0,40
Valor Total do Projeto (R\$)			162,40

Fonte: Autoria própria

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

A seguir obtemos algumas sugestões para a realização de trabalhos futuros, não se limitando somente a essa primeira versão do protótipo. Além disso, dependendo do objetivo da aplicação do projeto pode ser expandida para outras áreas:

- Com o advento da Indústria 4.0 é possível mesclar seus conceitos junto com os conhecimentos de programação web para criação de aplicativo utilizando a internet como meio de comunicação entre o smartphone e a casa automatizada, não se limitando somente a rede local, sendo o aplicativo a priori na plataforma Android, justamente por smartphones que possuem essa plataforma ser abundante e acessível na população pelo preço.
- Utilizar um microfone móvel sem necessidade de cabeamento, podendo ser via wireless ou bluetooth, modificando para que aceite a entrada Plug P-2, entrada exigida pelo módulo reconhecimento de voz dependendo do local onde a aplicação será instalada.
- Possibilidade de utilizar o módulo de reconhecimento de voz não só para lâmpadas e portões como demonstrado no projeto, mas também pode ser utilizado em aparelhos domésticos, por exemplo, ter o controle da televisão por voz, sem a necessidade da utilização do controle remoto, apenas precisando de programador para desenvolver o código.
- Melhoria na programação para que a voz seja reconhecida por um determinado grupo criado, não monopolizando as instruções para a pessoa que fez as gravações da assinatura da voz.

REFERÊNCIAS

- [1] AGUILAR, Luís Joyanes. **Fundamentos de Programação - Algoritmos, estruturas de dados e objetos**. Editora LTDA, 3ª edição, 2011.
- [2] ALVES, Jose Luiz Loureiro. **Instrumentação, Controle e Automação de Processos**. Editora LTC, 2ª edição, 2005.
- [3] ARDUINO. **Arduino Uno Rev 3**. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 23 jan. 2018.
- [4] AURESIDE. **Comparativo do mercado de automação residencial em 2015**. Disponível em: <http://www.aureside.org.br/_pdf/potencial_2015.pdf>. Acesso em: 23 jan. 2018.
- [5] BLUM, Jeremy. **Explorando o Arduino - Técnicas e Ferramentas para Mágicas de Engenharia**. Editora Alta Books, 2016.
- [6] CABRAL, Michel Madson Alves; CAMPOS, Antonio Luiz Pereira de Siqueira. **Sistemas de Automação Residencial de Baixo Custo: Uma Realidade Possível**. 2008.
- [7] ELECHOUSE. **Speak Recognition, Voice Recognition Module V3**. Disponível em: <http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=168_170&products_id=2254>. Acesso em: 23 jan. 2018.
- [8] ELECHOUSE. **Voice Recognition Module V3 – Speak to Control**. Disponível em: <https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf>. Acesso em: 23 jan. 2018.
- [9] FILIPEFLOP. **Micro Servo 9g SG90 Tower Pro**. Disponível em: <<https://www.filipeflop.com/produto/micro-servo-9g-sg90-towerpro>>. Acesso em: 05 fev. 2018.
- [10] GIMENEZ, Salvador Pinillos. **Microcontroladores 8051 - Teoria e Prática**. São Paulo, 1 ed., 2005.
- [11] GOMES, Victor Z. Ferreira. **A Domótica como instrumento para a melhoria da qualidade de vida dos portadores de deficiência**. Trabalho de Conclusão de Curso – (Graduação em Tecnologia em Automação Industrial), IFPB. João Pessoa, 2010.
- [12] GROOVER, Mikell P. **Automation, Production Systems and Computer Integrated Manufacturing Systems**. Prentice Hall, 2 edition, 1987.
- [13] IBGE. **Pesquisa Nacional de Saúde 2013 Ciclos de vida**. Disponível em: <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv94522.pdf>>. Acesso em: 23 jan. 2018.
- [14] JUNIOR, Sergio Luiz Stevan; SILVA, Rodrigo Adamshuk. **Automação e Instrumentação Industrial com Arduino - Teoria e Projetos**. Editora Érica Saraiva, 2015.

- [15] KARVINEN, Kimmo; KARVINEN, Tero. **Primeiros Passos com Sensores – Perceba o mundo usando eletrônica, Arduino e Raspberry Pi**. Editora Novatec, 2014.
- [16] MARWEDEL, Peter. **Embedded System Design**. Kluwer Academic Publishers, Dortmund, 1 edition, 2003.
- [17] MOREIRA, Jonathan Rosa. **AutoControl: Uma proposta de acessibilidade e segurança residencial com apoio da plataforma Arduino**; Trabalho de Conclusão de Curso; (Graduação em Sistema de Informação), Faculdade Projeção. São Paulo, 2013.
- [18] OGATA, Katsuhiko. **Engenharia de Controle Moderno**. Editora Pearson, 5ª edição, 2010.
- [19] OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. Editora Novatec, 2017.
- [20] PRUDENTE, Francesco. **Automação Predial e Residencial: Uma Introdução**, Editora LTC, 2010.
- [21] ROVERI, Michael Rubens. **Automação Residencial**. Trabalho de Conclusão de Curso – (Curso Superior de Tecnologia em Redes de Computadores), Faculdade Politec. Santa Bárbara d'Oeste, 2012.
- [22] SILVA, Danise da. **Desenvolvimento e Implementação de um Sistema de Supervisão e Controle Residencial**. Trabalho de Conclusão de Curso – (Pós-Graduação em Engenharia Elétrica), UFRN. Natal, 2009.
- [23] SILVA, Domiciano Correa Marques da. **"Microfone"**; Brasil Escola. Disponível em <<http://brasilescola.uol.com.br/fisica/microfone.htm>>. Acesso em: 12 fev. 2018.
- [24] SILVEIRA, Paulo Rogério da; SANTOS, Winderson E. **Automação e Controle Discreto**. Editora Érica, 2002.
- [25] TOCCI, Ronaldo J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas Digitais - Princípios e Aplicações**. Editora Pearson, 11ª edição, 2011.

APÊNDICE A – Código Fonte Principal do Projeto

```

// bibliotecas
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
#include <Servo.h>
#include <LiquidCrystal.h>

VR myVR(2,3); // 2:RX 3:TX, pino de comunicação entre modulo e o arduino

uint8_t records[7]; // quantidade de gravações
uint8_t buf[64];

// prototipo da funcao
void somLigar();
void somDesligar();
void msgTodosOn();
void msgTodosOff();
void msgPortao();
void msgSala();
void msgCozinha();
void msgQuarto();
void msgBanheiro();
void msgLCD();

// inicializa o servo
Servo myServo;

// define os pinos
LiquidCrystal lcd(14, 15, 16, 17, 18, 19); // LiquidCrystal lcd(<pino RS>, <pino enable>,
<pino D4>, <pino D5>, <pino D6>, <pino D7>)
int sala = 4;
int cozinha = 5;
int quarto = 6;
int banheiro = 7;
int portao = 8;
int buzzer = 9;
int i = 0;
bool status[6] = {false,false,false,false,false,false};

// define os nomes da variaveis de cada gravação
#define gravacao1 (0)
#define gravacao2 (1)
#define gravacao3 (2)
#define gravacao4 (3)
#define gravacao5 (4)
#define gravacao6 (5)
#define gravacao7 (6)

// imprime no serial monitor

```

```

void printSignature(uint8_t *buf, int len)
{
    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("]");
        }
    }
}

// imprime no serial monitor
void printVR(uint8_t *buf)
{
    Serial.println("-----");
    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");
}

// funções
void somLigar(){
    Serial.println("BIP"); // Indica que alguma lampada ou portao foi ligado/aberto
}

```

```

    tone(buzzer,349,300); // nota musical FA
    delay(100);
}

void somDesligar(){
    Serial.println("BIP BIP"); // Indica que alguma lampada ou portao foi
    desligado/fechado
    tone(buzzer,349,300); // nota musical FA
    delay(300);
    tone(buzzer,294,300); // nota musical RE
    delay(200);
}

void msgTodosOn(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Ligado: Lampadas"); //Envia o texto entre aspas para o LCD
}

void msgTodosOff(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Desligado: Todos"); //Envia o texto entre aspas para o LCD
}

void msgPortao(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Aberto: Portao"); //Envia o texto entre aspas para o LCD
    delay(3000);
}

void msgSala(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Ligado: Sala"); //Envia o texto entre aspas para o LCD
    delay(3000);
}

void msgCozinha(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Ligado: Cozinha"); //Envia o texto entre aspas para o LCD
    delay(3000);
}

void msgBanheiro(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Ligado: Banheiro"); //Envia o texto entre aspas para o LCD
}

```

```

    delay(3000);
}

void msgQuarto(){
    lcd.clear(); //Limpa a tela
    lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
    lcd.print("Ligado: Quarto"); //Envia o texto entre aspas para o LCD
    delay(3000);
}

// funcao mostra mensagem LCD
void msgLCD(){
    /*
    * status[0] = Sala;
    * status[1] = Cozinha;
    * status[2] = Quarto;
    * status[3] = Banheiro;
    * status[4] = Portao;
    * status[5] = Ativa todos;
    * status[6] = Desativa todos;
    */

    // Ativa Tudo( Todas as lampadas ligadas )
    if(status[0]== true && status[1]== true && status[2]== true && status[3]== true &&
status[4]== false){
        msgTodosOn();
        Serial.println("Ligados: Lampadas"); // Mostra no Monitor Serial
    }

    // Lampadas e Portao ativos
    else if(status[0]== true && status[1]== true && status[2]== true && status[3]== true
&& status[4]== true){
        lcd.clear(); //Limpa a tela
        lcd.setCursor(0, 0); //Posiciona o cursor na coluna 3, linha 0;
        lcd.print("Ligado: Todos"); //Envia o texto entre aspas para o LCD
        Serial.println("Ligado: Todos"); // Mostra no Monitor Serial
    }

    // Somente Portao aberto
    else if(status[0]== false && status[1]== false && status[2]== false && status[3]==
false && status[4]== true){
        msgPortao();
        Serial.println("Aberto: Portao"); // Mostra no Monitor Serial
    }

    // Somente Sala ligado
    else if(status[0]== true && status[1]== false && status[2]== false && status[3]==
false && status[4]== false){
        msgSala();
        Serial.println("Ligados: Sala"); // Mostra no Monitor Serial
    }

```

```

}

// Somente Cozinha ligado
else if(status[0]== false && status[1]== true && status[2]== false && status[3]==
false && status[4]== false){
    msgCozinha();
    Serial.println("Ligados: Cozinha"); // Mostra no Monitor Serial
}

// Somente Quarto ligado
else if(status[0]== false && status[1]== false && status[2]== true && status[3]==
false && status[4]== false){
    msgQuarto();
    Serial.println("Ligados: Quarto"); // Mostra no Monitor Serial
}

// Somente Banheiro ligado
else if(status[0]== false && status[1]== false && status[2]== false && status[3]==
true && status[4]== false){
    msgBanheiro();
    Serial.println("Ligados: Banheiro"); // Mostra no Monitor Serial
}

// Somente Portao e Sala
else if(status[0]== true && status[1]== false && status[2]== false && status[3]==
false && status[4]== true){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgPortao();
        msgSala();
    }
    Serial.println("Ligados: Portão e Sala"); // Mostra no Monitor Serial
}

// Somente Portao e Cozinha
else if(status[0]== false && status[1]== true && status[2]== false && status[3]==
false && status[4]== true){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgPortao();
        msgCozinha();
    }
    Serial.println("Ligados: Portão e Cozinha"); // Mostra no Monitor Serial
}

// Somente Portao e Quarto
else if(status[0]== false && status[1]== false && status[2]== true && status[3]==
false && status[4]== true){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgPortao();
        msgQuarto();
    }
}

```

```

    Serial.println("Ligados: Portão e Quarto"); // Mostra no Monitor Serial
}

    // Somente Portao e Banheiro
    else if(status[0]== false && status[1]== false && status[2]== false && status[3]==
true && status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgPortao();
            msgBanheiro();
        }
        Serial.println("Ligados: Portão e Banheiro"); // Mostra no Monitor Serial
    }

    // Somente Portao, Sala e Cozinha
    else if(status[0]== true && status[1]== true && status[2]== false && status[3]==
false && status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgPortao();
            msgSala();
            msgCozinha();
        }
        Serial.println("Ligados: Portão, Sala e Cozinha"); // Mostra no Monitor Serial
    }

    // Somente Portao, Sala e Quarto
    else if(status[0]== true && status[1]== false && status[2]== true && status[3]==
false && status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgPortao();
            msgSala();
            msgQuarto();
        }
        Serial.println("Ligados: Portão, Sala e Quarto"); // Mostra no Monitor Serial
    }

    // Somente Portao, Sala e Banheiro
    else if(status[0]== true && status[1]== false && status[2]== false && status[3]==
true && status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgPortao();
            msgSala();
            msgBanheiro();
        }
        Serial.println("Ligados: Portão, Sala e Banheiro"); // Mostra no Monitor Serial
    }

    // Somente Portao, Sala, Cozinha e Quarto
    else if(status[0]== true && status[1]== true && status[2]== true && status[3]== false
&& status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro

```

```

    msgPortao();
    msgSala();
    msgCozinha();
    msgQuarto();
}
Serial.println("Ligados: Portão, Sala, Cozinha e Quarto"); // Mostra no Monitor
Serial
}

    // Somente Portao, Sala, Cozinha e Banheiro
    else if(status[0]== true && status[1]== true && status[2]== false && status[3]== true
    && status[4]== true){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgPortao();
            msgSala();
            msgCozinha();
            msgBanheiro();
        }
        Serial.println("Ligados: Portão, Sala, Cozinha e Banheiro"); // Mostra no Monitor
        Serial
    }

    // Somente Sala e Cozinha
    else if(status[0]== true && status[1]== true && status[2]== false && status[3]==
    false && status[4]== false){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgSala();
            msgCozinha();
        }
        Serial.println("Ligados: Sala e Cozinha"); // Mostra no Monitor Serial
    }

    // Somente Sala e Quarto
    else if(status[0]== true && status[1]== false && status[2]== true && status[3]==
    false && status[4]== false){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgSala();
            msgQuarto();
        }
        Serial.println("Ligados: Sala e Quarto"); // Mostra no Monitor Serial
    }

    // Somente Sala e Banheiro
    else if(status[0]== true && status[1]== false && status[2]== false && status[3]==
    true && status[4]== false){
        for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
            msgSala();
            msgBanheiro();
        }
        Serial.println("Ligados: Sala e Banheiro"); // Mostra no Monitor Serial
    }

```

```

}

// Somente Sala, Cozinha e Quarto
else if(status[0]== true && status[1]== true && status[2]== true && status[3]== false
&& status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgSala();
        msgCozinha();
        msgQuarto();
    }
    Serial.println("Ligados: Sala, Cozinha e Quarto"); // Mostra no Monitor Serial
}

// Somente Sala, Cozinha e Banheiro
else if(status[0]== true && status[1]== true && status[2]== false && status[3]== true
&& status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgSala();
        msgCozinha();
        msgBanheiro();
    }
    Serial.println("Ligados: Sala, Cozinha e Banheiro"); // Mostra no Monitor Serial
}

// Somente Sala, Cozinha, Quarto e Banheiro
else if(status[0]== true && status[1]== true && status[2]== true && status[3]== true
&& status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgSala();
        msgCozinha();
        msgQuarto();
        msgBanheiro();
    }
    Serial.println("Ligados: Sala, Cozinha, Quarto e Banheiro"); // Mostra no Monitor
Serial
}

// Somente Cozinha e Quarto
else if(status[0]== false && status[1]== true && status[2]== true && status[3]==
false && status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgCozinha();
        msgQuarto();
    }
    Serial.println("Ligados: Cozinha e Quarto"); // Mostra no Monitor Serial
}

// Somente Cozinha e Banheiro
else if(status[0]== false && status[1]== true && status[2]== false && status[3]==
true && status[4]== false){

```

```

    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgCozinha();
        msgBanheiro();
    }
    Serial.println("Ligados: Cozinha e Banheiro"); // Mostra no Monitor Serial
}

// Somente Cozinha, Quarto e Banheiro
else if(status[0]== false && status[1]== true && status[2]== true && status[3]== true
&& status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgCozinha();
        msgQuarto();
        msgBanheiro();
    }
    Serial.println("Ligados: Cozinha, Quarto e Banheiro"); // Mostra no Monitor Serial
}

// Somente Quarto e Banheiro
else if(status[0]== false && status[1]== false && status[2]== true && status[3]==
true && status[4]== false){
    for(i=0;i<=1;i++){ // faz sempre que for verdadeiro
        msgQuarto();
        msgBanheiro();
    }
    Serial.println("Ligados: Quarto e Banheiro"); // Mostra no Monitor Serial
}

// Todos desligados
else{
    msgTodosOff();
    Serial.println("Desligado: Todos"); // Mostra no Monitor Serial
}
}

void setup()
{
    /** initialize */
    myVR.begin(9600);

    myServo.attach(portao); // pino 8 do servo motor
    myServo.write(180); // inicia fechado o portao
    lcd.begin(16, 1); // lcd com matriz de 16 colunas e 1 linha

    Serial.begin(115200);
    Serial.println("Elechouse Voice Recognition V3 Module\r\nControl LED sample");

    pinMode(sala, OUTPUT);
    pinMode(cozinha, OUTPUT);
    pinMode(quarto, OUTPUT);

```

```

pinMode(banheiro, OUTPUT);
pinMode(portao, OUTPUT);
pinMode(buzzer, OUTPUT);

if(myVR.clear() == 0){
  Serial.println("Recognizer cleared.");
}else{
  Serial.println("Não foi encontrado o Modulo de Reconhecimento de Voz.");
  Serial.println("Por favor, check a conexão e reinicie o Arduino.");
  while(1);
}

if(myVR.load((uint8_t)gravacao1) >= 0){
  Serial.println("Gravacao1 carregado.");
}

if(myVR.load((uint8_t)gravacao2) >= 0){
  Serial.println("Gravacao2 carregado.");
}

if(myVR.load((uint8_t)gravacao3) >= 0){
  Serial.println("Gravacao3 carregado.");
}

if(myVR.load((uint8_t)gravacao4) >= 0){
  Serial.println("Gravacao4 carregado.");
}

if(myVR.load((uint8_t)gravacao5) >= 0){
  Serial.println("Gravacao5 carregado.");
}

if(myVR.load((uint8_t)gravacao6) >= 0){
  Serial.println("Gravacao6 carregado.");
}

if(myVR.load((uint8_t)gravacao7) >= 0){
  Serial.println("Gravacao7 carregado.");
}

void loop()
{
  int ret;
  ret = myVR.recognize(buf, 50);
  if(ret>0){
    switch(buf[1]){
      case gravacao1: //vetor 0
        /** SALA */
        if(status[0] == false){

```

```

digitalWrite(sala, HIGH);
status[0] = true;
somLigar();
}
else{
digitalWrite(sala, LOW);
status[0] = false;
somDesligar();
}
break;

case gravacao2: //vetor 1
/** COZINHA */
if(status[1] == false){
digitalWrite(cozinha, HIGH);
status[1] = true;
somLigar();
}
else{
digitalWrite(cozinha, LOW);
status[1] = false;
somDesligar();
}
break;

case gravacao3: //vetor 2
/** QUARTO */
if(status[2] == false){
digitalWrite(quarto, HIGH);
status[2] = true;
somLigar();
}
else{
digitalWrite(quarto, LOW);
status[2] = false;
somDesligar();
}
break;

case gravacao4: //vetor 3
/** BANHEIRO */
if(status[3] == false){
digitalWrite(banheiro, HIGH);
status[3] = true;
somLigar();
}
else{
digitalWrite(banheiro, LOW);
status[3] = false;
somDesligar();
}

```

```

}
break;

case gravacao5: //vetor 4
/** PORTAO */
if(status[4] == false){
for(i=180;i>=90;i--) { // abri de forma suave
  myServo.write(i); // aqui I é o angulo específico
  delay(50);
}
status[4] = true;
somLigar();
}
else{
for(i=90;i<=180;i++) { // fecha de forma suave
  myServo.write(i); // aqui I é o angulo específico
  delay(50);
}
status[4] = false;
somDesligar();
}
break;

case gravacao6: //vetor 5
/** ATIVA TODOS */
for(i=0;i<=3;i++){
  if(status[i] == false){
    digitalWrite(4+i, HIGH);
    status[i] = true; }
  }
//myServo.write(90);
somLigar();
status[5] = true;
break;

case gravacao7: //vetor 6
/** DESATIVA TODOS*/
for(i=0;i<=4;i++){
  if(status[i] == true){
    digitalWrite(4+i, LOW);
    status[i] = false; }
  }
myServo.write(180);
somDesligar();
status[5] = false;
break;

default: //caso nao seja nenhum dos vetores anteriores
Serial.println("Gravação não reconhecida.");
break;

```

```
}  
/** voice recognized */  
printVR(buf);  
Serial.println("-----");  
Serial.print("Sala:");  
Serial.println(status[0]);  
Serial.print("Cozinha:");  
Serial.println(status[1]);  
Serial.print("Quarto:");  
Serial.println(status[2]);  
Serial.print("Banheiro:");  
Serial.println(status[3]);  
Serial.print("Portao:");  
Serial.println(status[4]);  
Serial.print("Todos:");  
Serial.println(status[5]);  
msgLCD();  
Serial.println("-----");  
}  
}
```