



**UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA**

PAULO VICTOR NASCIMENTO DOS SANTOS

**SISTEMA IN-DOOR DE MONITORAMENTO E CONTROLE PARA
ACESSIBILIDADE DE PESSOAS COM DEFICIÊNCIA PELO RECONHECIMENTO
DE FALA UTILIZANDO RASPBERRY PI**

**MANAUS - AM
2022**



PAULO VICTOR NASCIMENTO DOS SANTOS

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia da Universidade Federal do Amazonas, como parte do requisito para obtenção do Título de Bacharel em Engenharia Elétrica na ênfase de Eletrônica.

Orientador: Prof. Me. Francisco de Assis Pereira Januário

**MANAUS – AM
2022**

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S237s Santos, Paulo Victor Nascimento dos
Sistema in-door de monitoramento e controle para acessibilidade de pessoas com deficiência pelo reconhecimento de fala utilizando Raspberry Pi / Paulo Victor Nascimento dos Santos . 2022
63 f.: 31 cm.

Orientador: Francisco de Assis Pereira Januário
TCC de Graduação (Engenharia Elétrica - Eletrônica) -
Universidade Federal do Amazonas.

1. Reconhecimento de fala. 2. Automação residencial. 3. Pessoas com deficiência. 4. Mobilidade reduzida. I. Januário, Francisco de Assis Pereira. II. Universidade Federal do Amazonas III. Título

PAULO VICTOR NASCIMENTO DOS SANTOS

**SISTEMA IN-DOOR DE MONITORAMENTO E CONTROLE PARA
ACESSIBILIDADE DE PESSOAS COM DEFICIÊNCIA PELO RECONHECIMENTO
DE FALA UTILIZANDO RASPBERRY PI**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Tecnologia da Universidade
Federal do Amazonas, como parte do requisito
para obtenção do Título de Bacharel em
Engenharia Elétrica na ênfase de Eletrônica.

Aprovado em 25 de abril de 2022.

Francisco de Assis Pereira Januário

Prof. MSc. Francisco de Assis Pereira Januário
Departamento de Eletrônica e Computação
Universidade Federal do Amazonas

André Luiz Duarte Cavalcante

Prof. Dr. André Luiz Duarte Cavalcante
Departamento de Eletrônica e Computação
Universidade Federal do Amazonas

Frederico da Silva Pinagé

Prof. Dr. Frederico da Silva Pinagé
Departamento de Eletrônica e Computação
Universidade Federal do Amazonas

RESUMO

A evolução das tecnologias utilizadas em automações residenciais tem levantado diversas discussões sobre a sua utilização em favor da acessibilidade, visto que pessoas com deficiência ou com mobilidade reduzida seriam grandemente beneficiadas com a implantação de tais tecnologias. Dentre os mecanismos emergentes, destaca-se a interface de comando por fala, pois possibilita que algumas ações motoras sejam executadas utilizando a voz. Neste contexto, este projeto visa abordar o desenvolvimento de um sistema *in-door* com interface de comando via fala, que tem como objetivo controlar dispositivos domésticos, como televisão, ar-condicionado e lâmpadas, para auxiliar pessoas com mobilidade reduzida. É utilizada a plataforma de desenvolvimento *Raspberry Pi*, associada ao circuito de transmissão de sinais infravermelhos e o relé de chaveamento da lâmpada. Estes módulos são gerenciados por um script desenvolvido na linguagem de programação *Python*, fazendo uso da *Google Speech API*. Para enviar comandos via sinais infravermelhos, a biblioteca LIRC é utilizada. O sistema foi avaliado quanto à eficiência do reconhecimento de fala quando os comandos são pronunciados a 10, 60 e 100 cm de distância do microfone, apresentando métricas de 100%, 87% e 73%, respectivamente, com uma média geral de 87% de conversões corretas. São abordados os aspectos que influenciam a queda de eficiência no reconhecimento dos comandos, e soluções para este problema são apresentados. Apesar deste fator, a proposta de controlar os dispositivos foi cumprida, sendo possível expandi-la ao incluir novos meios de acionar diferentes aparelhos de uma residência.

Palavras-chave: Reconhecimento de fala, automação residencial, pessoas com deficiência, mobilidade reduzida.

ABSTRACT

The evolution of technologies used in residential automation has raised several discussions about their use in favor of accessibility, since people with disabilities or reduced mobility would greatly benefit from the implementation of such technologies. Among the emerging mechanisms, the speech command interface stands out, as it allows some motor actions to be performed using the voice. In this context, this project aims to address the development of an in-door system with a command interface via speech, which aims to control domestic devices, such as television, air conditioning and lamps, to help people with reduced mobility. The Raspberry Pi development platform is used, associated with the infrared signal transmission circuit and the lamp switching relay. These modules are managed by a script developed in the Python programming language, making use of the Google Speech API. To send commands via infrared signals, the LIRC library is used. The system was evaluated for the efficiency of speech recognition when commands are pronounced at 10, 60 and 100 cm away from the microphone, presenting metrics of 100%, 87% and 73%, respectively, with an overall average of 87% of correct conversions. Aspects that influence the drop in efficiency in command recognition are discussed, and solutions for this problem are presented. Despite this factor, the proposal to control the devices was fulfilled, with the possibility of expanding it by including new means of activating different devices in a residence.

Keywords: Speech recognition, home automation, disabled people, reduced mobility.

LISTA DE FIGURAS

Figura 1.1: Diagrama básico do funcionamento do sistema.....	14
Figura 2.1: Diagrama em blocos básico de um sistema de casa inteligente	19
Figura 2.2: Sistema básico de reconhecimento de fala.....	20
Figura 2.3: Raspberry Pi 3B	23
Figura 2.4: Descrição dos pinos do Raspberry Pi 3 B	24
Figura 2.5: LED Emissor Infravermelho	25
Figura 2.6: Receptor infravermelho 1838B	26
Figura 2.7: Interface do software Raspberry Pi Imager.....	27
Figura 2.8: Script básico para reconhecimento de fala com arquivos de áudio	31
Figura 2.9: Saída após execução do script.....	31
Figura 2.10: Resultado aplicando os parâmetros offset e duration.....	31
Figura 2.11: Script básico para captação com microfone.....	32
Figura 2.12: Resultado do reconhecimento utilizando microfone.....	32
Figura 2.14: Exemplo de arquivo de configuração do controle da televisão	34
Figura 3.1: Arquitetura proposta por Mtshali e Khubisa.....	36
Figura 3.2: Estrutura geral do laboratório remoto	37
Figura 3.3: Diagrama de funcionamento da cadeira de rodas automatizada.....	37
Figura 4.1: Arquitetura geral do desenvolvimento do projeto.....	39
Figura 4.2: Diagrama de funcionamento do sistema	40
Figura 4.3: Circuito de amplificação de corrente para o LED infravermelho	41
Figura 4.4: Esquema de ligação entre o relé e a lâmpada	42
Figura 4.5: Trecho do <i>script</i> referente ao reconhecimento de fala.....	43
Figura 4.6: Trecho do script para controle da televisão	44
Figura 4.7: Trecho do script para acionamento da lâmpada.....	44
Figura 5.1: Ambiente de testes do sistema	45
Figura 5.2: Porcentagem de acertos dos comandos com a distância de 10 cm	47
Figura 5.3: Porcentagem de acertos dos comandos com a distância de 60 cm	47
Figura 5.4: Porcentagem de acertos dos comandos com a distância de 100 cm	48
Figura 5.5: Tendência da média de eficiência em função da distância	49

LISTA DE TABELAS

Tabela 2.1: Tecnologias e usos para pessoas com deficiência	17
Tabela 2.2: Descrição dos terminais do receptor infravermelho	26
Tabela 4.1: Custo para implementação do sistema.....	42
Tabela 5.1: Quantidade de acertos dos comandos com base nas distâncias entre usuário e microfone.....	46

LISTA DE ABREVIATURAS E SIGLAS

ASR – *Automatic Speech Recognition*

USB – *Universal Serial Bus*

GPIO – *General Purpose Input/Output*

IBGE – Instituto Brasileiro de Geografia e Estatística

PcD – Pessoa com Deficiência

SMS – *Short Message Service*

RAF – Reconhecimento Automático de Fala

LPC – *Linear Predictive Coding*

MFCC – *Mel-frequency cepstral coefficients*

HMM – *Hidden Markov Model*

DICT – *Dictionary*

AVI – Assistentes virtuais inteligentes

RAM – *Random Access Memory*

HDMI – *High-Definition Multimedia Interface*

I/O – *Input/Output*

SPI – *Serial Peripheral Interface*

UART – *Universal Asynchronous Receiver/Transmitter*

EEPROM – *Electrically Erasable Programmable Read-Only Memory*

PWM – *Pulse Width Modulation*

LED – *Light Emitting Diode*

GND – *Ground*

VDC – Tensão em corrente contínua

VAC – Tensão alternada

OS – *Operating System*

IDE – *Integrated Development Environment*

API – *Application Programming Interface*

LIRC – *Linux Infrared Remote Control*

DC – *Direct current*

GPS – *Global Positioning System*

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Descrição do problema	13
1.2	Objetivos	14
1.2.1	Objetivos específicos	14
1.3	Descrição da solução	14
1.4	Organização do trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	PcD e acessibilidade	16
2.2	Automação residencial.....	18
2.3	Reconhecimento de fala.....	19
2.4	Assistentes virtuais	21
2.5	Raspberry Pi 3B	22
2.6	LED Infravermelho.....	24
2.7	Receptor Infravermelho	25
2.8	Sistema operacional: Raspberry Pi OS	27
2.9	Linguagem de programação: Python	27
2.10	<i>SpeechRecognition</i>	29
2.10.1	Transcrição com arquivos de áudio	30
2.10.2	Transcrição com microfone	31
2.11	LIRC	32
3	TRABALHOS RELACIONADOS.....	35
3.1	Sistema de controle de eletrodomésticos com Alexa.....	35
3.2	Laboratório remoto operado por comando de fala.....	36
3.3	Cadeira de rodas inteligente baseada em comando de voz e GPS	37
4	PROJETO.....	39

4.1	Arquitetura do sistema	39
4.2	Sequência de ações	40
4.3	Materiais utilizados no hardware	41
4.4	Desenvolvimento do Script Python	43
4.5	Ambiente de testes	45
5	RESULTADOS	45
5.1	Discussão	49
6	CONCLUSÕES.....	51
6.1	Trabalhos futuros	51
	REFERÊNCIAS	53
	APÊNDICES	56
	APÊNDICE A – Arquivo de configuração dos controles remotos da biblioteca LIRC.....	57
	APÊNDICE B – Código Python.....	61

1 INTRODUÇÃO

A tecnologia tem proporcionado à sociedade, ao decorrer dos anos, diversas possibilidades de promover a evolução da qualidade de vida, seja por permitir que várias atividades diárias sejam feitas simultaneamente, seja por assegurar execuções simplificadas das mesmas [1]. Nesse contexto, a automatização de atividades domésticas pode trazer benefícios que incentivam o conforto, a conveniência e a segurança dos usuários [2].

As *Smart Homes*, do inglês “Casa inteligente”, são baseadas na integração entre dispositivos eletrônicos e eletrodomésticos, que podem ser configurados de acordo com o interesse de seus residentes. A automação constitui grande parte do sistema de uma casa inteligente, em que os dispositivos interconectados permitem aos usuários a configuração de aspectos do ambiente de acordo com o seu desejo, como iluminação e temperatura do ambiente [3].

Casas inteligentes podem ter um papel significativo para pessoas idosas ou com algum grau de incapacidade física. Poder fazer atividades – que normalmente requereriam auxílio externo – de maneira independente, utilizando os mecanismos de automação residencial, garantiria a melhoria da qualidade de vida [4]. Os ambientes deste tipo devem ser projetados para atender as necessidades especiais de cada pessoa de maneira individual, de forma que os comandos sejam interpretados de maneira adequada, e as atividades sejam executadas propriamente.

Com o objetivo de prover tais facilidades a esse público, o emprego da voz como interface de controle tem sido amplamente utilizado, em associação com a automação residencial. Sistemas com Interface de Usuários por Voz – IUV (do inglês, *Voice User Interface*) fazem uso da voz para realizar comandos ou efetuar transações de maneira mais prática, utilizando o Reconhecimento Automático de Fala - RAF (do inglês, *Automatic Speech Recognition*). Aplicações com o RAF minimizam a necessidade de realizar comandos através de métodos usuais que dependem do uso das mãos, como teclados e mouses, apresentando uma alternativa que favorece o cotidiano de pessoas com limitações motoras [5].

O trabalho tem como motivação proporcionar a essas pessoas a possibilidade de executar tarefas do dia a dia de maneira simples e independente, por meio da automatização de atividades que podem ser executadas com o uso da fala, utilizando um sistema de baixo custo. O sistema de reconhecimento foi desenvolvido na plataforma Raspberry Pi, em associação com um microfone USB e com a biblioteca *Speech Recognition* em linguagem Python, utilizando

comandos únicos pré-definidos em português para executar as ações de ligar a televisão, ligar o ar-condicionado e ligar uma lâmpada. Para acionar a televisão e o ar-condicionado, comandos via infravermelho foram emitidos individualmente, enquanto a lâmpada é ativada por meio do chaveamento de um relé, controlado pelo pino de entrada e saída de propósito geral - GPIO (do inglês, *General Purpose Input/Output*) do Raspberry. Foi desenvolvido um script Python para realizar a integração dos módulos e executar os comandos definidos.

1.1 Descrição do problema

O número de pessoas com algum tipo de deficiência, no Brasil, é expressivo. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), o percentual chega a cerca de 24% da população do país, representando aproximadamente 46 milhões de brasileiros [6]. A deficiência visual estava presente em 3,4% da população nacional; no caso da deficiência motora, em 2,3%; já a deficiência auditiva, em 1,1%; e a deficiência mental/intelectual, em 1,4%.

De acordo com a Lei Federal nº 13.146/2015, em seu artigo 2º [7], uma pessoa com deficiência é aquela que possui impedimentos a longo prazo, seja de natureza física, mental, intelectual ou sensorial, que tem sua participação reduzida na sociedade devido a estes impedimentos. Já as pessoas com mobilidade reduzida são definidas como aquelas que apresentam algum tipo de dificuldade de locomoção, temporária ou permanente, que reduz sua mobilidade, flexibilidade e coordenação motora [8]. Dentre esses casos, podem ser incluídas as pessoas idosas, cuja população segue a tendência dos últimos anos de aumentar [9]. Entre 2012 e 2017, de acordo com o IBGE, o número aumentou em 4,8 milhões de pessoas acima dos 60 anos, com um total acima de 30,7 milhões de pessoas [9].

Nesse contexto do número crescente de pessoas que têm dificuldades motoras, pode-se afirmar que o conceito de casas inteligentes e automação residencial aplicado a este público tem sido considerado importante por diversas instituições, em países tecnologicamente avançados. É perceptível que ambientes adaptados especificamente para essas necessidades promoveriam conforto, praticidade e um grau de independência para essas pessoas, facilitando atividades cotidianas. [4]

1.2 Objetivos

O objetivo do trabalho é desenvolver um sistema de reconhecimento de fala de baixo custo usando Raspberry Pi e linguagem Python, capaz de executar atividades domésticas simples – como acionamento e desligamento de dispositivos eletroeletrônicos – que podem apresentar dificuldades para pessoas com mobilidade reduzida em seu dia a dia.

1.2.1 Objetivos específicos

- a. Desenvolver um protótipo para identificação dos comandos pela fala, que possibilita o controle de eletrodomésticos da residência, como televisão, ar-condicionado e lâmpadas via infravermelho e relé;
- b. Avaliar o desempenho do sistema no que diz respeito à eficiência de entendimento dos comandos, identificando os fatores que influenciam na interpretação da sentença pronunciada;
- c. Avaliar a viabilidade da implementação do sistema considerando praticidade, utilidade e benefícios proporcionados.

1.3 Descrição da solução

O sistema é formado por três componentes principais: o microfone, a unidade de controle e os periféricos de acionamento e desligamento dos dispositivos. A Figura 1.1 ilustra o diagrama básico do funcionamento do projeto, identificando a sequência de ações.

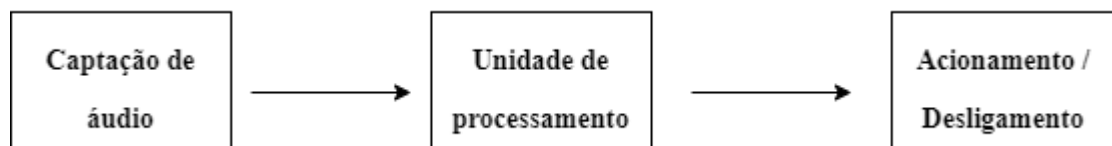


Figura 1.1: Diagrama básico do funcionamento do sistema. Fonte: Autoria própria.

- a. **Captação de áudio** – Responsável pela captação da fala do usuário, convertendo o sinal de áudio em sinais elétricos que podem ser interpretados pela unidade de controle;
- b. **Unidade de processamento** – Tem o Raspberry Pi como central de processamento, com a função de realizar a transcrição do sinal elétrico que representa a fala em texto

(desenvolvido utilizando a biblioteca *Python Speech Recognition* [9]), sendo possível identificar o comando solicitado; a partir da transcrição, aciona os periféricos de controle para acionar ou desligar os dispositivos eletrônicos;

c. **Periféricos de acionamento e desligamento** – Recebem os sinais de ativação da unidade de controle, e dependendo do comando solicitado, envia informações via infravermelho (no caso da televisão e do ar-condicionado) ou via GPIO para chaveamento do relé.

1.4 Organização do trabalho

O trabalho segue a organização por capítulos, em que este inicial apresenta uma introdução apontando os problemas que motivaram o desenvolvimento do projeto, além dos objetivos a serem alcançados.

O Capítulo 2 trata dos fundamentos que embasaram a pesquisa, como o reconhecimento de fala e os dispositivos e tecnologias que o cercam, o Raspberry Pi e seu sistema operacional utilizado, Raspbian, além de módulos que possibilitam o controle de dispositivos eletrônicos

Já o Capítulo 3 aborda sobre os trabalhos relacionados à aplicação da automação, em associação com o reconhecimento de fala, para melhoria da qualidade de vida de pessoas com limitações motoras, seja no âmbito doméstico ou até mesmo no trabalho e estudos.

No Capítulo 4 são listados os materiais e métodos utilizados para realização do projeto, seja para a elaboração do script de processamento, seja no desenvolvimento do hardware utilizado para acionamento e desligamento dos dispositivos. São apresentados a arquitetura e a sequência de ações do sistema.

O Capítulo 5 apresenta o cenário utilizado para coleta dos dados, os resultados obtidos quanto à eficiência do reconhecimento de fala, e a análise e discussão em torno dos mesmos.

Por fim, no Capítulo 6 são apresentadas as conclusões obtidas e os trabalhos propostos para o futuro.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são explanados os conceitos básicos necessários para a compreensão do funcionamento do sistema. As noções de acessibilidade e PcD (Pessoa com Deficiência), em conjunto com os estudos sobre automação residencial, são fundamentais para entender de que forma a tecnologia pode promover melhorias na qualidade de vida destas pessoas. O reconhecimento de fala e os mecanismos de conversão de áudio em texto também são apresentados. Por fim, os elementos que constituem o sistema, nos âmbitos de hardware (plataforma de desenvolvimento, sensores, atuadores) e software (linguagem de programação e bibliotecas utilizadas) serão abordados.

2.1 PcD e acessibilidade

O termo deficiência é definido como uma restrição física, mental ou sensorial, que pode ser permanente ou temporária, e que ocasiona dificuldades em exercer atividades do cotidiano [10]. Dentre os tipos de deficiências existentes, este trabalho visa auxiliar pessoas com deficiência física, que são aquelas que possuem algum tipo de limitação de movimentação, devido a perda completa ou parcial de membros do corpo relativos à função locomotora [11].

Além de PcDs, indivíduos que encontram dificuldades em seu cotidiano para executar atividades simples são aqueles que possuem a mobilidade reduzida: apresentam dificuldades de movimentar-se de maneira definitiva ou temporária, prejudicando sua mobilidade, coordenação motora e percepção. Nem todas as pessoas com mobilidade reduzida se enquadram na definição de PcD, como crianças, idosos e gestantes, pois a dificuldade do movimento, que reduz a mobilidade, faz parte do desenvolvimento dos seres humanos, em todas as suas fases [12].

Nesse contexto, é necessário prover meios que concedam ambientes iguais a este público através da acessibilidade, que pode ser entendida como uma condição que possibilita a superação das barreiras para que estas pessoas tenham participação de maneira igualitária na sociedade em que vivem [13]. É um atributo que permite a melhoria da qualidade de vida das pessoas, garantindo que todos tenham as mesmas condições de uso de produtos, serviços e informações [14]. Para proporcionar tais condições, é necessário identificar e eliminar os fatores que dificultam para os seres humanos a realização de atividades ou funções na sociedade em que vivem, assegurando os direitos a todos os indivíduos [13].

A tecnologia pode ter papel fundamental como ferramenta usada para eliminar as barreiras impostas pela dificuldade motora, permitindo que estas pessoas possam participar de

forma igualitária na sociedade em que vivem. A Tabela 2.1 elenca as possibilidades trazidas pelo uso da tecnologia por pessoas com deficiência ou com mobilidade reduzida. Estas promovem o acesso à informação, à saúde e à educação, de forma a melhorar sua integração social e econômica na sociedade, já que amplia a quantidade de atividades que estas pessoas podem participar [15].

Tabela 2.1: Tecnologias e usos para pessoas com deficiência [15].

Tipos de tecnologia	Usos potenciais para pessoas com deficiência
Internet	<ul style="list-style-type: none"> • <i>Sites</i>: educação à distância e cursos de treinamento, redes sociais, compras, bancos e outros serviços online; • Teletrabalho; • Telemedicina e saúde eletrônica.
Aparelhos e serviços móveis	<ul style="list-style-type: none"> • <i>Smartphones e tablets</i>; • SMS; • Serviços de emergência em formato de voz, texto ou linguagem de sinais; • Aplicativos de saúde móvel.
TV e serviços	<ul style="list-style-type: none"> • Serviços e aplicações interativos de multimídia; • Serviços de acesso: legenda de texto, legenda de áudio, audiodescrição; • Interpretação de linguagem de sinais.
Softwares e aplicativos	<ul style="list-style-type: none"> • Interpretação de linguagem de sinais pela Internet; • Sistemas de bate-papo; • Software de acessibilidade: leitores de tela, conversores de voz em texto, digitação em telas; • Livros e documentos eletrônicos acessíveis;

	<ul style="list-style-type: none"> • Aplicativos para educação e recreação especiais.
Tecnologias emergentes	<ul style="list-style-type: none"> • Casas inteligentes; • <i>Wearables;</i> • Inteligência Artificial; • Conversor de voz em texto, texto em voz, texto/voz em linguagem de sinais.

2.2 Automação residencial

A automação residencial, também definida pelo termo “casa inteligente”, tem se apresentado como conceito largamente difundido em diversos locais do mundo. Pode-se definir uma casa inteligente como “um conjunto de serviços proporcionados por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto de uma habitação” [16]. Um sistema de automação residencial inclui diversos componentes [2], dentre os quais pode-se destacar:

- **Interface de usuário:** um monitor, computador ou celular, por qual é possível dar comandos para controlar o sistema;
- **Modo de transmissão:** conexões cabeadas, como Ethernet; ou conexões sem fio, como *bluetooth*, ondas de rádio ou infravermelho;
- **Controle central:** Interface de hardware que faz a comunicação com a interface do usuário para receber os comandos e controlar os serviços domésticos;
- **Dispositivos eletrônicos:** Dispositivos que serão controlados pela unidade central e que são compatíveis com o modo de transmissão, como lâmpadas, ar-condicionados e televisões.

A Figura 2.1 ilustra a organização básica de um sistema de automação residencial, com seus principais componentes.

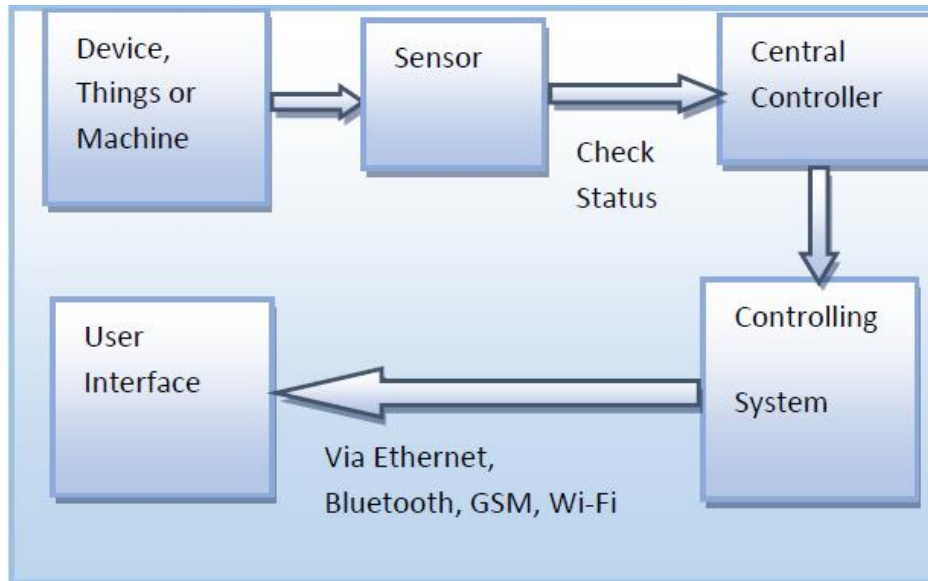


Figura 2.1: Diagrama em blocos básico de um sistema de casa inteligente [2]

O conceito de casa inteligente tem sido aplicado, frequentemente, a PcDs e pessoas com mobilidade reduzida, com o desenvolvimento de diversos tipos de automações, dependendo das necessidades e limitações físicas em cada tipo. O projeto de uma casa inteligente que tem como foco auxiliar pessoas com mobilidade reduzida ou PcDs deve prever a instalação de dispositivos que facilitem a locomoção e manipulação de objetos, como sistemas robóticos para assistência de movimento e interfaces humano-máquina especializadas para cada indivíduo. Além disso, o controle do ambiente deve ser feito de maneira simplificada, de forma que uma pessoa com dificuldades motoras tenha a possibilidade de configurar seu cômodo facilmente, conforme seu desejo [4].

2.3 Reconhecimento de fala

A automação de uma residência pode ser feita de diversas formas, com vários tipos de interface com os usuários, dependendo do perfil de uso em cada pessoa. Dentre estas formas, uma das mais difundidas é o reconhecimento de fala, que possibilita a execução de comandos sem necessidade de usar as mãos. O fato de ser altamente customizável e personalizável torna os sistemas com Reconhecimento Automático de Fala (RAF) favorecidos para desenvolvimento de aplicações, visto que suas características permitem uma adaptação fácil, de acordo com o especificado [5].

O processo de reconhecimento de fala se dá, basicamente, como um mapeamento do sinal de áudio (convertido para um sinal elétrico, por meio de um transdutor) que foi

pronunciado em uma sequência de informações discretas [17]. Os elementos que compõem um sistema genérico de reconhecimento de fala são mostrados na Figura 2.2.

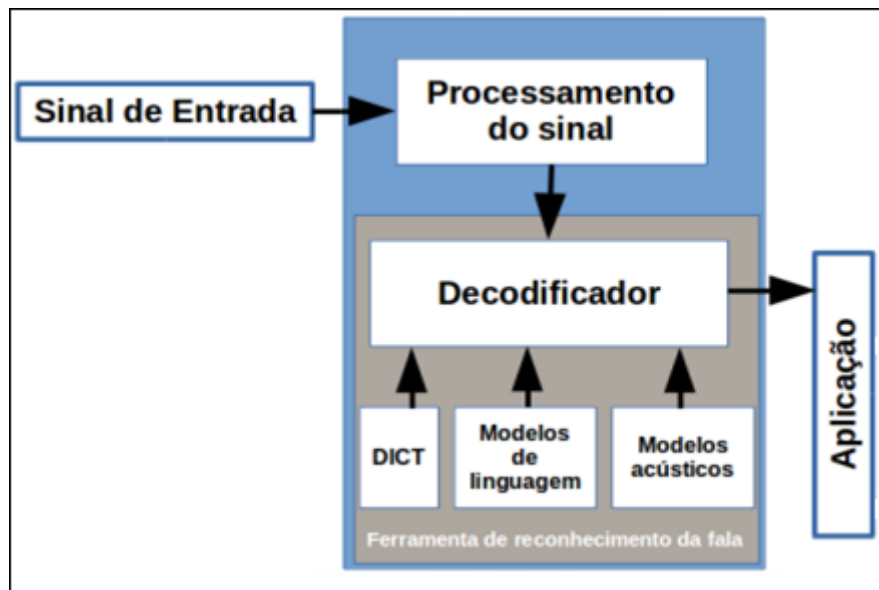


Figura 2.2: Sistema básico de reconhecimento de fala [17].

O processamento de sinal é o estágio em que o sinal de entrada é convertido para uma representação de vetores que carregam características da voz. Diversas técnicas podem ser utilizadas para realizar esta extração, das quais as que se destacam pela eficiência são a análise do banco de filtros, a codificação preditiva linear - LPC (do inglês, *Linear Predictive Coding*) e os coeficientes mel-cepstrais – MFCC (do inglês, *Mel-frequency cepstral coefficients*) [17].

As ferramentas de reconhecimento de fala têm como componentes:

- a. Os modelos de linguagem, que listam as palavras que o sistema é capaz de reconhecer, quais destas palavras têm possibilidade de aparecerem juntas, seguindo alguma ordem. Cada palavra listada é associada com os fonemas que a formam;
- b. Os modelos acústicos, que mostram diversos parâmetros fonéticos e de variabilidade de sinal, e normalmente são apresentados por um modelo oculto de Markov - HMM (do inglês, *Hidden Markov Model*);
- c. O decodificador, que utiliza as informações presentes nos modelos acústico e de linguagem para indicar a sequência de fonemas que tem maior probabilidade de representar o vetor característico apresentado na entrada.

Para gerar o vocabulário equivalente, é feito uso de um dicionário – representado como DICT na Figura 2.2 – com as palavras identificáveis pelo sistema, em que o modelo de

linguagem verifica quais palavras têm maior similaridade com aquela sequência de fonemas gerados [17].

O reconhecimento de fala tem como objetivo principal conseguir identificar o significado de cada palavra pronunciada, o que o difere do reconhecimento de voz, que prioriza compreender quem pronunciou as palavras. Para obter sucesso em encontrar qual palavra foi pronunciada, sendo possível distinguir uma da outra, é preciso extrair e modelar as diversas características que dependem da fala, como a forma que foi pronunciada [18].

Existem diversos motivos que podem reduzir a eficiência para compreender as palavras em um sistema de reconhecimento de fala. Dentre os fatores que influenciam o resultado, pode-se destacar [16]:

- a. O sotaque da pessoa que profere as palavras influencia na pronúncia, o que pode tornar complexo desenvolver um sistema de RAF utilizando português brasileiro, visto que este apresenta uma grande variação de pronúncias em diferentes partes do país;
- b. Em idiomas como o português, que possui diversas palavras com significados diferentes, porém pronúncias idênticas (como sessão e seção, concerto e concertos, dentre outras), é possível obter um resultado diferente do esperado;
- c. O nível de ruído no ambiente em que se pronuncia as palavras interfere no resultado obtido, já que quanto mais ruído próximo à captação do áudio de entrada, pior será o reconhecimento, e vice-versa.

2.4 Assistentes virtuais

Dentro do contexto de evolução tecnológica, é notável a ascensão do uso de interfaces que tem a possibilidade de interagir com o ambiente. As pessoas têm adotado, no seu cotidiano, o uso de tecnologias que possam facilitar suas atividades. Nesse aspecto, as assistentes virtuais inteligentes (AVI) têm se sobressaído, visto que proporcionam praticidade na execução de tarefas, das mais simples como fazer um lembrete, até as mais complexas, como controlar a temperatura de um cômodo. Essa praticidade se dá pelo fato de que a execução das atividades pode ser solicitada via comandos no *smartphone*, seja por meio de texto ou por comandos usando a voz.

As AVI mais difundidas no mercado, atualmente, são: Alexa, desenvolvida pela Amazon; Siri, feita pela Apple; Google Assistant, do Google; e a Cortana, feita pela Microsoft. Elas se baseiam em tecnologias como inteligência artificial e aprendizado de máquina, que permitem a elas aprender as rotinas dos usuários e até mesmo associar a voz de quem solicita os comandos com suas preferências, tornando a execução de tarefas do dia a dia descomplicadas.

Estes softwares inteligentes têm a capacidade de beneficiar pessoas com dificuldades motoras ou mesmo visuais, já que apresentam a possibilidade de execução de comandos por meio da voz, promovendo acessibilidade às pessoas que não têm condições iguais aos demais indivíduos.

2.5 Raspberry Pi 3B

O Raspberry Pi 3B, ilustrado na Figura 2.3, é um minicomputador que possui componentes periféricos integrados, utilizado principalmente no âmbito educacional. O processamento não é o mesmo de um computador convencional, principalmente pelo seu tamanho, porém é suficiente para o desenvolvimento de projetos de pequeno porte, sendo a combinação ideal para a construção de projetos de sistemas embarcados. Utiliza sistemas operacionais baseado em Linux, o que permite uma alta customização do ambiente. Este modelo tem como principais características [19]:

- Processador *Quad Core* 1.2GHz Broadcom BCM2837 64bits;
- 1GB RAM;
- BCM43438 *wireless* LAN e *Bluetooth Low Energy* (BLE) integrado;
- 40 pinos de GPIO;
- 4 portas USB 2.0;
- Saída HDMI;
- *Slot* Micro SD para armazenar arquivos e instalar o sistema operacional;
- Suporte a Micro USB para fonte de alimentação até 2.5 A



Figura 2.3: Raspberry Pi 3B [19].

Os pinos de I/O são numerados conforme é mostrado na Figura 2.4. Dentre os 40 pinos presentes na placa, tem-se:

- 2x pinos de alimentação 5 V (2, 4);
- 2x pino de alimentação 3.3 V (1, 17);
- 8x pinos de alimentação 0 V, GND (6, 9, 14, 20, 25, 30, 34, 39);
- 2x pinos de comunicação via interface I2C, usada para conexão de periféricos de baixa velocidade (3, 5);
- 8x pinos que podem ser usados para comunicação SPI – *Serial Peripheral Interface* – (19, 21, 23, 24, 26, 35, 38, 40);
- 2x pinos de comunicação UART - *Universal Asynchronous Receiver/Transmitter* – (8, 10);
- 2x pinos de ID EEPROM, que é um tipo de memória que pode ser programada e apagada diversas vezes, através de tensão elétrica interna ou externa (27, 28);
- 1x pinos de PWM (*Pulse Width Modulation*), que pode ser usado para controle de motores (12)
- 13x pinos de entrada/saída de dados digitais - GPIO (11, 13, 15, 18, 22, 29, 31, 32, 33, 36, 37).

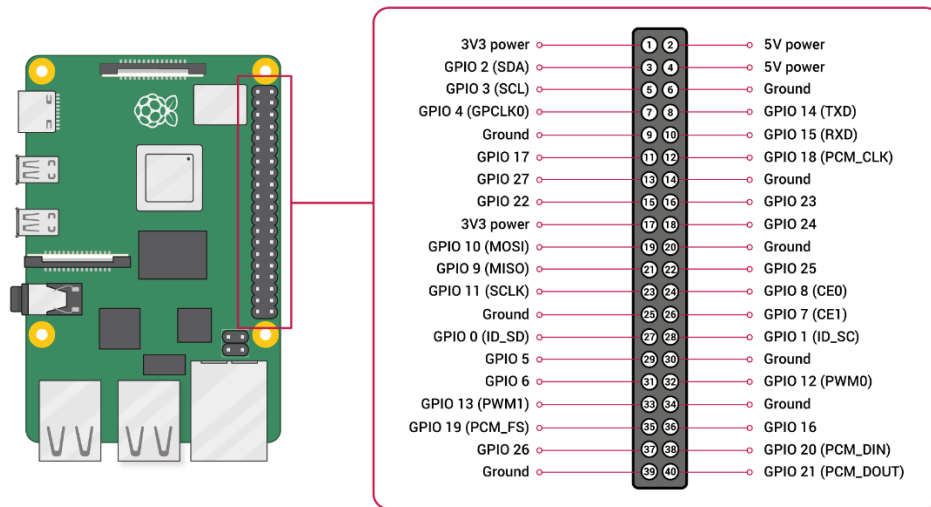


Figura 2.4: Descrição dos pinos do Raspberry Pi 3 B [19].

Os pinos de GPIO, quando configurados como saída, apresentam valor de tensão de 3,3 V quando estão em nível lógico alto, e 0 V em nível lógico baixo. Quando definidos como entrada, é possível configurar, via software, resistores de *pull-up*.

2.6 LED Infravermelho

O LED Emissor infravermelho, mostrado na Figura 2.5, é utilizado principalmente em circuitos eletrônicos em que se deseja controlar outros dispositivos, sejam eles relés, alarmes, aparelhos domésticos, ou qualquer outro que possa ser controlado via infravermelho. Este controle é feito usualmente em conjunto com microcontroladores, que podem ser programados para enviar, através do emissor, comandos codificados para os equipamentos, sendo uma boa alternativa para utilizar como controle remoto.



Figura 2.5: LED Emissor Infravermelho [20].

As especificações do componente são:

- Tensão de operação: 1,2 V a 1,4 V;
- Diâmetro: 5mm;
- Comprimento de onda: 940 nm;
- Terminais: 2.

2.7 Receptor Infravermelho

O receptor infravermelho, modelo 1838B, ilustrado na Figura 2.6, é utilizado amplamente em projetos em que é necessária a compatibilidade com sinais infravermelhos de controles remotos. Através deste componente é possível ler estes sinais e adquirir os padrões dos códigos utilizados em diversos controles do mercado, procedimento que foi utilizado neste projeto para aquisição dos padrões referentes aos controles remotos da televisão e do ar-condicionado.



Figura 2.6: Receptor infravermelho 1838B. Adaptado de [21].

De acordo com a numeração mostrada na Figura 2.6, as funções de cada terminal do receptor infravermelho são:

Tabela 2.2: Descrição dos terminais do receptor infravermelho [21]

Número do terminal	Função do terminal
1	Sinal
2	GND
3	VCC

O receptor tem como especificações técnicas:

- Tensão de operação: 2,7 V a 5 V;
- Frequência de operação: 38 KHz;
- Tensão de nível lógico alto: 4,5 V;
- Tensão de nível lógico baixo: 0,4 V;
- Ângulo de +/- 45°.

2.8 Sistema operacional: Raspberry Pi OS

O sistema operacional (SO) escolhido para o desenvolvimento do projeto foi o Raspberry Pi OS, que é baseado em *Debian*, sendo otimizado para o hardware do Raspberry Pi. É um conjunto de programas básicos e utilidades que permitem ao usuário aproveitar ao máximo o que a plataforma tem a oferecer, dentre os quais destacam-se: as interfaces de desenvolvimento integrado - IDEs (do inglês, *Integrated Development Environment*) para o desenvolvimento de programas, softwares para execução de mídias, *browsers* para navegação *online*, dentre outros.

O SO deve ser instalado em um cartão de memória, com no mínimo 8 GB de espaço. Para isso, é possível fazer uso do software Raspberry Pi Imager, que tem a função de formatar o cartão de memória de acordo com o padrão do SO e de gravar os arquivos do mesmo. A Figura 2.7 ilustra a interface do software, em sua versão 1.2.



Figura 2.7: Interface do software Raspberry Pi Imager. Fonte: Autoria própria.

2.9 Linguagem de programação: Python

Para executar comandos que controlam a ação dos dispositivos, faz-se necessário o uso de instruções que podem ser programadas e que sejam compreendidas pelo microprocessador. Neste caso, são utilizadas as linguagens de programação de alto nível, que permitem que o

usuário gere sequências de instruções – ou algoritmos – de acordo com seu objetivo, de maneira simplificada, e que pode ser compilado, transformando os comandos em combinações de palavras binárias, que podem ser interpretadas pelas unidades de processamento.

Para o desenvolvimento deste projeto, optou-se por usar o Python: uma linguagem interpretada, dinâmica e de alto nível. Esta teve sua origem em 20 de fevereiro de 1991, criada por Guido Van Rossum. Ele almejava obter, com o Python [22]:

- a. Uma linguagem usada em atividades cotidianas, que permitisse menor tempo de desenvolvimento;
- b. Um código que fosse de fácil leitura e compreensão, tão simples quanto a língua inglesa;
- c. Uma linguagem que fosse *open source*, ou seja, todos que tivessem o desejo poderiam contribuir com a evolução do programa.

O Python é uma linguagem orientada a objetos, o que significa que se baseia na interação e criação de atributos, métodos, objetos, entre outros [23]. Conceitos como classes, objetos, métodos, instâncias e outros são fundamentais na compreensão da linguagem e no uso de suas bibliotecas.

O Python foi escolhido para esta aplicação devido ao grande número de bibliotecas relacionadas ao reconhecimento de fala, dentre as quais se destacam [24]:

- *apiai*;
- *assemblyai*;
- *google-cloud-speech*;
- *pocketsphinx*;
- *SpeechRecognition*;
- *watson-developer-cloud*;
- *wit*.

Devido à facilidade do uso da plataforma, bem como o grande uso e participação da comunidade, foi escolhida a biblioteca *SpeechRecognition*, que será abordada na seção 2.8.

2.10 *SpeechRecognition*

SpeechRecognition é uma biblioteca que possibilita realizar o reconhecimento de fala na linguagem *Python* de maneira simplificada. Ela permite que sejam utilizadas diferentes APIs (*Application Programming Interface* – Interface de programação de aplicações), sejam elas *online* ou *offline*. São elas [25]:

- *CMU Sphinx (offline)*;
- *Google Speech Recognition*;
- *Google Cloud Speech API*;
- *Wit.ai*;
- *Microsoft Bing Voice Recognition*;
- *Houndify API*;
- *IBM Speech to Text*;
- *Snowboy Hotword Detection (offline)*.

Para que seja possível fazer uso completo da biblioteca, é necessário cumprir alguns requisitos, sendo eles:

- *PyAudio 0.2.11+* (obrigatório, se uma entrada de microfone for utilizada);
- *PocketSphinx* (obrigatório somente se for feito uso do *Sphinx recognizer*);
- *Google API Client Library para Python* (obrigatório, se for usada a *Google Cloud Speech API*);
- Codificador *FLAC* (obrigatório, se o sistema utilizado não for baseado em *Windows/Linux/OS X x86*).

Para fazer uso de todas as funcionalidades oferecidas pela biblioteca, é necessário criar uma instância da classe *Recognizer*. Cada instância desta classe pode utilizar sete diferentes métodos de reconhecimento de fala, utilizando as APIs mencionadas anteriormente. São eles:

- *recognize_bing()*: *Microsoft Bing Speech*;
- *recognize_google()*: *Google Web Speech API*;
- *recognize_google_cloud()*: *Google Cloud Speech*;
- *recognize_houndify()*: *Houndify*;
- *recognize_ibm()*: *IBM Speech to Text*;
- *recognize_sphinx()*: *CMU Sphinx*;

- `recognize_wit(): Wit.ai`.

Todos estes métodos requerem que uma fonte de áudio seja fornecida para que seja feito o reconhecimento de fala. Esta fonte precisa ser obtida como uma instância da classe da *SpeechRecognition*, a *AudioData*. Existem duas formas possíveis de criar instâncias da classe *AudioData*: via arquivos de áudio e via microfone.

2.10.1 Transcrição com arquivos de áudio

No caso do uso de arquivos de áudio, é necessário que existam áudios pré-gravados e com formatos específicos aceitos pela biblioteca, que são:

- WAV;
- AIFF;
- AIFF-C;
- FLAC.

Para trabalhar com os arquivos, é feito uso da classe *AudioFiles*, utilizada na linha 4 da Figura 2.9, que apresenta uma interface que facilita a leitura e o gerenciamento dos conteúdos dos arquivos de áudio. Ela pode ser inicializada com o caminho em que se encontram os arquivos. O resultado da leitura é gravado em uma instância da classe *AudioData* através do método *record*, representado na linha 7 da Figura 2.8, que pertence à classe *Recognizer*. Assim, é possível utilizar os métodos de reconhecimento, listados anteriormente, para obter o resultado da fala em texto.

O método *record* também permite que apenas partes menores do áudio em questão sejam decodificados, fazendo uso dos argumentos *duration* e *offset*. No caso de *duration*, é interrompida a gravação após o número de segundos especificado no parâmetro. Já com o *offset*, a gravação é iniciada após o número de segundos escolhido. Desta forma, apenas o intervalo de segundos maior que o *offset* e menor que o *duration* são transcritos. Um exemplo desta aplicação é ilustrado na Figura 2.8.

```

speech_test.py > ...
1  import speech_recognition as sr
2
3  r = sr.Recognizer()
4  test_file = sr.AudioFile('TestAudio3.wav')
5
6  with test_file as source:
7      audio = r.record(source, offset=2, duration=3.1)
8
9  print(r.recognize_google(audio))

```

Figura 2.8: Script básico para reconhecimento de fala com arquivos de áudio. Fonte: Autoria própria.

O idioma padrão utilizado pelo método *recognize_google*, mostrado na linha 9 da Figura 2.8, é o inglês dos Estados Unidos, porém este é um parâmetro que pode ser configurado, a depender do idioma com que se deseja trabalhar. A saída do script, sem a aplicação dos parâmetros *offset* e *duration*, é mostrada na Figura 2.9.

```

[Running] python -u "f:\Users\Pichau\Documents\Paulo\Paulo - UFAM\TCC\Scripts\speech_test.py"
turn on the lights turn off the lights turn on all devices turn off all devices

```

Figura 2.9: Saída após execução do script. Fonte: Autoria própria.

O resultado do uso de *offset* e *duration* pode ser visto na Figura 2.10, em que foram transcritas somente duas frases, ao contrário do que foi visualizado no resultado da Figura 2.9, com quatro frases reconhecidas do arquivo de áudio.

```

[Running] python -u "f:\Users\Pichau\Documents\Paulo\Paulo - UFAM\TCC\Scripts\speech_test.py"
turn off the lights turn on all devices

```

Figura 2.10: Resultado aplicando os parâmetros *offset* e *duration*. Fonte: Autoria própria.

Outra funcionalidade importante seria o método *adjust_for_ambiente_noise*, que verifica o nível de ruído presente no arquivo de áudio e calibra o reconhecedor para se adequar a tal nível. Dependendo do nível de ruído, não é possível fazer uma transcrição precisa, sendo necessário utilizar filtros externos à biblioteca para amenizar o nível de interferência.

2.10.2 Transcrição com microfone

De maneira semelhante à transcrição com arquivos, é necessário criar instâncias de classes específicas que gerenciam a captura via microfone. Neste caso, é feito uso de uma

instância da classe *Microphone*, também da biblioteca *SpeechRecognition*. Ao criar uma instância desta classe, é possível especificar o microfone desejado usando o parâmetro *device_index*. Caso o argumento não seja explicitado, será feito uso do microfone padrão do sistema.

Para capturar a entrada de áudio vinda do microfone, deve-se usar o método *listen*, conforme apresentado na linha 8 da Figura 2.11, que recebe como argumento a fonte de áudio escolhida. Também pode ser usado o método *adjust_for_ambient_noise*, ilustrado na linha 7 da Figura 2.11, para reduzir os efeitos de ruído na transcrição da fala. A estrutura do script que permite transcrever a fala por meio do microfone é similar ao visto na seção 2.10.1, desta vez realizando o reconhecimento para o português do Brasil.

```

speech_test.py > ...
1  import speech_recognition as sr
2
3  r = sr.Recognizer()
4  mic = sr.Microphone()
5
6  with mic as source:
7      r.adjust_for_ambient_noise(source)
8      audio = r.listen(source)
9
10 print(r.recognize_google(audio, language='pt-br'))

```

Figura 2.11: Script básico para captação com microfone. Fonte: Autoria própria.

O resultado é mostrado na Figura 2.12. Dependendo do nível de ruído no ambiente e da forma que foram pronunciadas as palavras, é possível que o script retorne uma *exception* chamada *UnknownValueError*. Isto significa que a API não foi capaz de compreender o áudio falado, portanto deve ser tratado com um bloco *try-exception*.

```

[Running] python -u "f:\Users\Pichau\Documents\Paulo\Paulo - UFAM\TCC\Scripts\speech_test.py"
reconhecimento de fala

```

Figura 2.12: Resultado do reconhecimento utilizando microfone. Fonte: Autoria própria.

2.11 LIRC

A biblioteca LIRC (*Linux Infrared Remote Control*) [26] permite decodificar e enviar sinais infravermelhos de diversos controles remotos encontrados no mercado, possibilitando

desenvolver aplicações que envolvem aparelhos controlados por este tipo de sinal. É um serviço executado em plano de fundo, em sistemas Linux, que permite enviar informações através de comandos programáveis pelo usuário.

Para enviar os comandos desejados, deve-se ter um arquivo de configuração contendo o mapeamento dos botões e seus respectivos códigos, para cada controle remoto. É possível listar, através do terminal, os controles configurados com o comando “`irsend LIST <nome_do_controle>`”. Para enviar o comando de um botão do controle, é utilizado o comando “`irsend SEND_ONCE <nome_do_controle> <nome_do_botão>`”.

Para obter os códigos de sinais infravermelhos referentes aos controles remotos dos dispositivos, é necessário executar o comando *irrecord*. Este comando apresenta uma série de passos necessários para a captação dos sinais de cada controle, em que se faz necessário nomear o controle em uso e seus botões a serem mapeados. Esta captação é feita utilizando um sensor infravermelho, que envia seus dados para o *Raspberry*. O resultado do procedimento é um arquivo de configuração “`<nome_do_controle>.conf`”, com os parâmetros do controle e seus botões mapeados. A Figura 2.15 mostra um exemplo de arquivo de configuração gerado pelo comando *irrecord*.

```

1
2 # Please take the time to finish this file as described in
3 # https://sourceforge.net/p/lirc-remotes/wiki/Checklist/
4 # and make it available to others by sending it to
5 # <lirc@bartelmus.de>
6 #
7 # This config file was automatically generated
8 # using lirc-0.10.1(default) on Tue Mar  1 16:18:55 2022
9 # Command line used: -d /dev/lirc0 /home/pi/lircd.conf
10 # Kernel version (uname -r): 5.10.63-v7+
11 #
12 # Remote name (as of config file): tv
13 # Brand of remote device, the thing you hold in your hand:
14 # Remote device model nr:
15 # Remote device info url:
16 # Does remote device has a bundled capture device e. g., a
17 #   usb dongle? :
18 # For bundled USB devices: usb vendor id, product id
19 #   and device string (use dmesg or lsusb):
20 # Type of device controlled
21 #   (TV, VCR, Audio, DVD, Satellite, Cable, HTPC, ...) :
22 # Device(s) controlled by this remote:
23
24 begin remote
25
26   name tv
27   bits          32
28   flags SPACE_ENC|CONST_LENGTH
29   eps           30
30   aeps          100
31
32   header        9018  4567
33   one           504   1742
34   zero          504   622
35   ptrail        506
36   repeat        9027  2296
37   gap           108645
38   toggle_bit_mask 0x0
39   frequency     38000
40
41   begin codes
42     KEY_POWER          0x20DF10EF 0xEAD43F00
43   end codes
44
45 end remote

```

Figura 2.13: Exemplo de arquivo de configuração do controle da televisão. Fonte: Autoria própria.

É possível interagir com o *daemon* (processo que é executado em segundo plano, sem que o usuário o controle diretamente) LIRC através da biblioteca *LIRC Python Package* [27], possibilitando a integração com o script desenvolvido no projeto. É necessário instanciar um cliente para interagir com o serviço LIRC, e a partir de então, pode-se fazer uso de seus comandos. São enviados sinais infravermelhos através do método *send_once*, que recebe dois parâmetros: o primeiro especifica qual controle remoto presente no arquivo de configuração será usado, e o segundo indica qual botão deste controle será enviado. Há a opção de enviar múltiplas vezes o mesmo comando, adicionando o argumento *repeat_count* do método *send_once*.

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados estudos relacionados ao reconhecimento de fala aplicado ao aumento da acessibilidade, permitindo que as pessoas que convivem com deficiências físicas tenham suas atividades diárias, seja no âmbito profissional ou doméstico, simplificadas, para que possam conviver de maneira igual entre todos os membros da sociedade. São listados três projetos que usam essa tecnologia: um sistema de controle de eletrodomésticos por voz [3], um laboratório remoto operado pela fala [28] e uma cadeira de rodas equipada com controle via voz e monitoramento de localização [29].

3.1 Sistema de controle de eletrodomésticos com Alexa

O estudo de Mtshali e Khubisa [3] apresenta uma solução baseada na API Amazon Alexa para controle de dispositivos eletroeletrônicos em um contexto doméstico, com o objetivo de facilitar atividades diárias para pessoas com deficiência física. O sistema desenvolvido faz uso de dispositivos que permitem integração com a Alexa, sendo eles *smart plugs*, ou tomadas inteligentes e câmeras inteligentes, de forma a acionar aparelhos da casa e monitorar possíveis danos que essas pessoas possam sofrer. As tomadas foram conectadas aos dispositivos comuns em uma residência, que não possuem integração à rede. Os eletrodomésticos usados foram: fogão, panela elétrica, abajur, ventilador, torradeira, geladeira, máquina de lavar e forno de micro-ondas.

O trabalho apresenta três cenários de uso dos dispositivos, conforme a Figura 3.1: no primeiro, cada equipamento eletroeletrônico é ligado a uma tomada inteligente, sendo que cada tomada pode ser nomeada de acordo com o dispositivo a ela associado. Por exemplo, é atribuído o nome “geladeira” ao *plug* em que é conectada a geladeira. Desta forma, os comandos enviados através da *Alexa* são facilitados, permitindo ao usuário simplesmente dizer: “Alexa, ligue o fogão”; já no segundo cenário, são adicionadas as câmeras inteligentes, que estariam conectadas a uma televisão ou a uma tela de computador, de forma que a pessoa tenha a possibilidade de visualizar o ambiente que desejar, seja ele interno ou externo, simplesmente utilizando comandos como “Alexa, mostre o ambiente externo”; por fim, o terceiro cenário apresenta uma solução utilizando câmeras DeepLens, que podem ser configuradas para detectar quando a pessoa sofre uma queda, e imediatamente emite um comando de voz para a *Alexa*, que pode

solicitar auxílio rapidamente. Este caso seria interessante para pessoas com deficiência física que moram sozinhas, pois não teriam acesso a uma rápida assistência.



Figura 3.1: Arquitetura proposta por Mtshali e Khubisa [3].

3.2 Laboratório remoto operado por comando de fala

O artigo desenvolvido por Yayla *et al* [28] apresenta uma aplicação de laboratório remoto que visa aumentar a acessibilidade de estudantes de eletrônica com deficiência física, permitindo que estes participem de atividades de projetos de circuitos utilizando a tecnologia do reconhecimento de fala. O projeto possibilita que as operações sejam feitas sem o uso das mãos, e que o estudante consiga realizar análises de circuitos eletrônicos utilizando apenas a voz. Os experimentos abrangem as aplicações de um amplificador operacional na eletrônica, e pode-se escolher entre seis diferentes experimentos.

O sistema é baseado em uma arquitetura cliente/servidor, conforme mostra a Figura 3.2. Há um gerador de sinais que pode ser controlador via computador, e desta forma seus principais parâmetros (amplitude, forma de onda, frequência) podem ser ajustados pelo usuário por meio da voz. Um Raspberry Pi foi usado como interface entre o servidor de reconhecimento de fala e a placa de experimentos, sendo possível escolher qual experimento será realizado e quais os valores dos parâmetros (resistência e nível de tensão DC de referência) serão escolhidos. Um osciloscópio é conectado à placa de experimentos, sendo possível observar os sinais de entrada e saída simultaneamente através da câmera associada ao Raspberry Pi. Ao finalizar os

procedimentos experimentais utilizando o laboratório remoto, um e-mail é gerado e enviado automaticamente pelo sistema, contendo o usuário do estudante, o horário em que foi feito *login*, as capturas de tela do osciloscópio e os resultados finais do experimento.

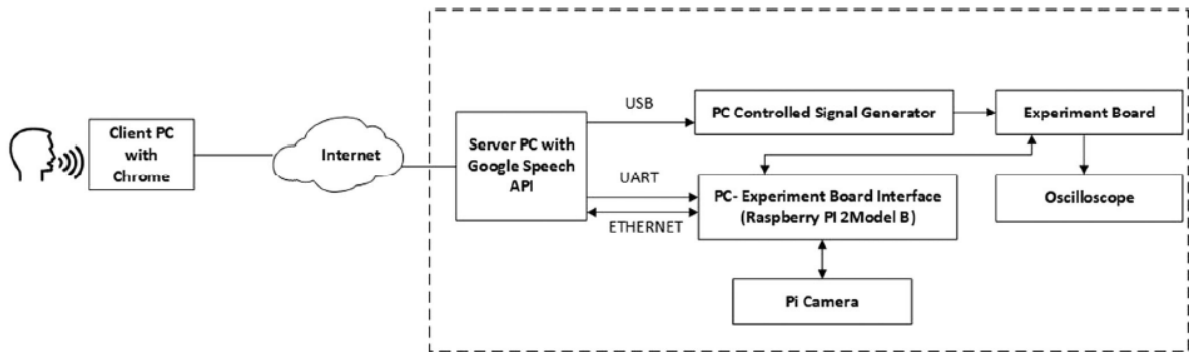


Figura 3.2: Estrutura geral do laboratório remoto [28].

3.3 Cadeira de rodas inteligente baseada em comando de voz e GPS

O trabalho desenvolvido por Aktar *et al* [29] apresenta uma cadeira de rodas controlada via comandos de voz, que tem sua localização monitorada via GPS e enviada para um aplicativo de *smartphone*. O motor que movimenta a cadeira pode assumir três velocidades: baixa, média e alta. Além disso, a cadeira conta com um sistema de detecção de obstáculos, realizado utilizando sensores de infravermelho. A Figura 3.3 mostra os elementos que compõem do sistema.

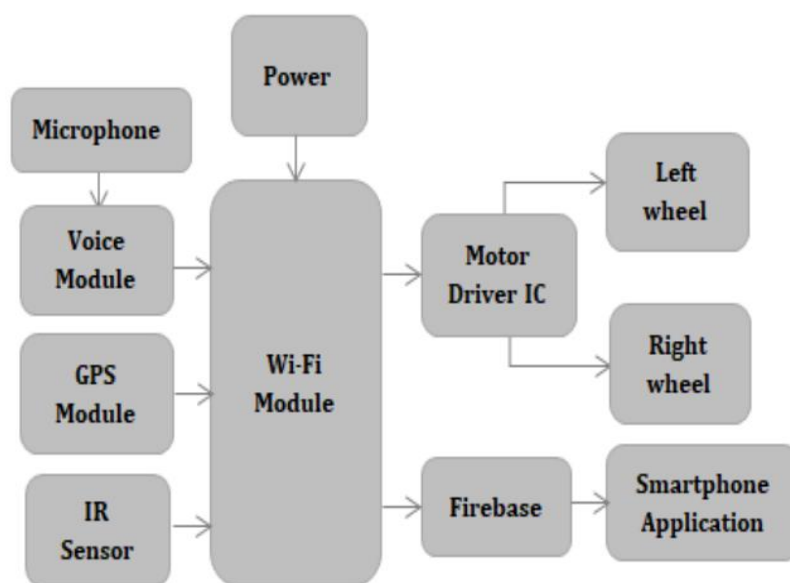


Figura 3.3: Diagrama de funcionamento da cadeira de rodas automatizada [29].

O módulo de reconhecimento de voz utilizado possibilita a gravação de até 80 comandos, que devem ser gravados previamente com a voz do usuário para que seja feito o reconhecimento. Já o sensor infravermelho tem o objetivo de identificar obstáculos, e quando encontrar uma barreira, para a cadeira imediatamente, independente do comando atual. O módulo de GPS é responsável por obter as coordenadas da localização da pessoa com deficiência. O módulo *Wi-Fi* é usado como unidade de processamento, sendo conectado com os demais módulos periféricos, direcionando os comandos enviando as informações para cada unidade do sistema. O *Firebase* é um servidor *online* que é responsável de receber as informações de geolocalização fornecidas pelo GPS e enviá-las para o aplicativo no *smartphone*. O motor, associado a um *driver* de controle, define as velocidades da cadeira, além de sua direção, em ambas as rodas. E a aplicação Android no *smartphone* possibilita que a pessoa com deficiência tenha sua localização acompanhada por parentes ou pessoas de confiança.

4 PROJETO

Este capítulo trata sobre o projeto e implementação do sistema, tratando dos materiais e métodos utilizados. Inicialmente, é abordada a arquitetura do sistema e a sequência de ações para uso do mesmo. A seguir, é explicado o processo referente ao desenvolvimento do hardware, relatando os circuitos eletrônicos necessários. Posteriormente, são abordados os aspectos pertinentes à composição do código de controle, como o *script* em *Python* e as bibliotecas envolvidas para a criação do mesmo.

4.1 Arquitetura do sistema

A arquitetura básica utilizada para o desenvolvimento é ilustrada na Figura 4.1. Para realizar o controle dos dispositivos, são utilizadas três etapas gerais: a captação do comando emitido via fala, o processamento da ação solicitada e o acionamento dos aparelhos conforme a solicitação feita pelo usuário.

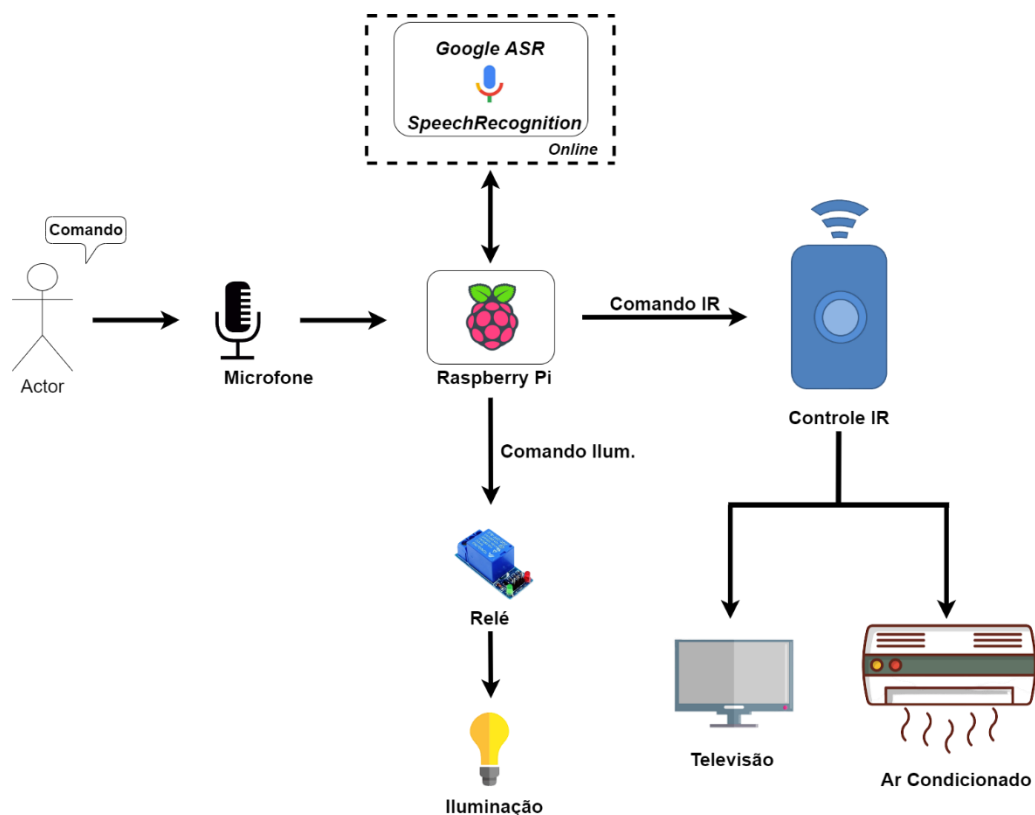


Figura 4.1: Arquitetura geral do desenvolvimento do projeto. Fonte: Autoria própria.

Primeiro, é necessário realizar a transdução do sinal de áudio, referente ao comando emitido pelo usuário, em sinal elétrico, de forma que este possa ser interpretado pela unidade de processamento. Este passo é realizado utilizando um microfone USB, que possui fácil uso devido à presença da tecnologia *plug and play*, funcionando sem a instalação de *drivers* adicionais. Após, é necessário interpretar o comando enviado, convertendo-o para o formato de texto. Nesta etapa, é feito o uso do Raspberry Pi 3B, associado ao Raspberry Pi OS. Na plataforma, o script *Python* consegue coletar o áudio captado pelo microfone utilizando a classe *Microphone*, da biblioteca *SpeechRecognition*. Então, através do método *recognize_google*, que faz uso da *Google Speech API*, é feito o reconhecimento das palavras pronunciadas. Com o resultado da transcrição, é feita uma comparação e, dependendo do comando reconhecido, diferentes dispositivos são acionados. Por fim, para que o comando solicitado seja efetuado, os dispositivos periféricos (via infravermelho ou relé) são acionados. No caso do comando infravermelho, são utilizados os códigos adquiridos dos controles remotos de televisão e ar-condicionado. No caso da lâmpada, seu acionamento é feito por meio do chaveamento do relé, que é controlado pela saída GPIO do Raspberry Pi. As seções 4.2 e 4.3 detalham os procedimentos e componentes utilizados para o desenvolvimento de hardware e software, respectivamente.

4.2 Sequência de ações

O diagrama ilustrando a sequência de ações para entender o funcionamento do projeto é mostrado na Figura 4.2. Após processar o comando, a unidade controladora executa as ações de acionamento ou desligamento do dispositivo, de acordo com o que foi solicitado pelo usuário.

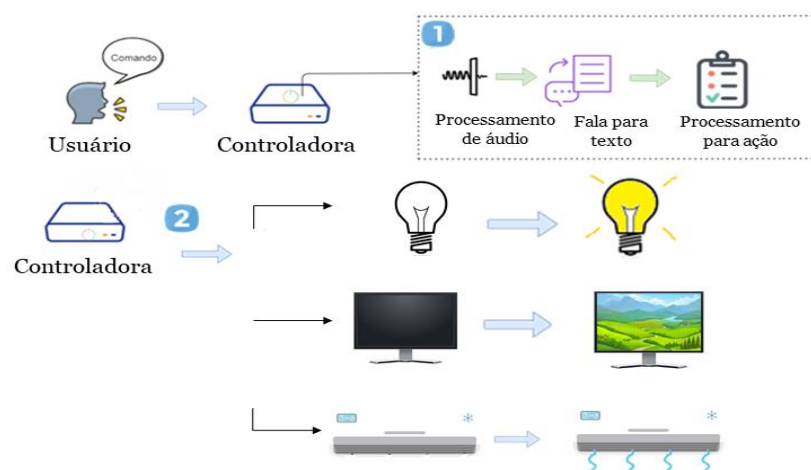


Figura 4.2: Diagrama de funcionamento do sistema. Fonte: Autoria própria.

Os comandos de acionamento foram escolhidos com o objetivo de facilitar o uso do sistema, e por isso são utilizadas palavras simples e de fácil pronúncia. No caso dos aparelhos controlados via infravermelho, são exemplos: “ligar televisão”, “desligar ar-condicionado”, e vice-versa. Já no caso da iluminação, é necessário dizer: “acender lâmpada” ou “apagar lâmpada”. Além do fator da praticidade de uso dos comandos, foi necessário atentar-se à questão da pronúncia das palavras, visto que, dependendo do sotaque, há produção de ruído quando sons de “s” são pronunciados no meio ou no fim de palavras. Desta forma, dependendo do nível de ruído emitido, o tratamento de áudio do sistema pode remover este som e realizar uma transcrição da fala de maneira incorreta.

4.3 Materiais utilizados no hardware

Em questão de componentes de hardware, além do já mencionado Raspberry Pi e seus pinos de GPIO, são utilizados: um circuito de infravermelho, um relé de controle 5 VDC e os dispositivos a serem controlados, todos eles sendo conectados à unidade de processamento principal.

Para o circuito de infravermelho, foi feito o uso do LED transmissor de infravermelho de 940 nm, que permite o envio das informações de controle do dispositivo. Porém, como a corrente de saída dos pinos de GPIO é baixa (em torno de 16 mA), o alcance do transmissor seria limitado, o que proporcionaria dificuldade de operação dos dispositivos a grandes distâncias. Desta forma, optou-se pela adição de um circuito a transistor para amplificação da corrente, que é mostrado na Figura 4.3. O diagrama esquemático foi desenvolvido no software Altium Designer. O circuito foi montado em *protoboard* para testes e validação do sistema.

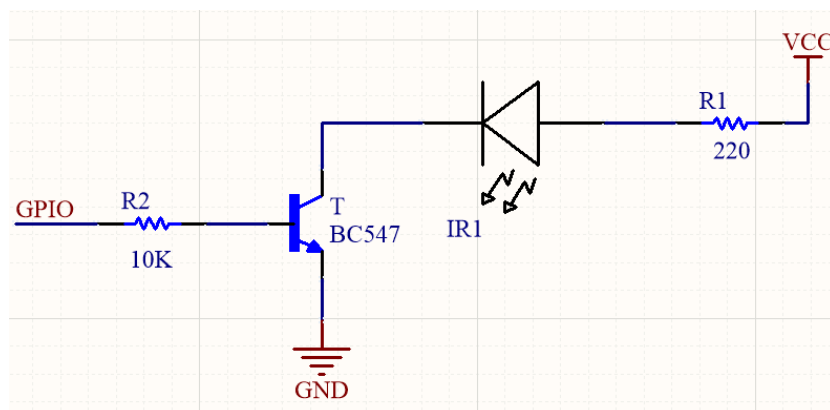


Figura 4.3: Circuito de amplificação de corrente para o LED infravermelho. Fonte: Autoria própria.

Para a captação dos padrões de códigos dos controles remotos, foi utilizado um receptor de infravermelho modelo VS 1838B. Além dos terminais de alimentação (Vcc e GND), é conectado o sensor ao pino 17 do Raspberry Pi para a leitura dos dados. Como o componente foi utilizado de maneira provisória para a aquisição dos dados dos controles remotos, não se fez necessária a inclusão do mesmo no projeto da placa de circuito eletrônico.

O módulo relé de 5 VDC, mostrado na Figura 4.4, foi usado para o acionamento e desligamento da lâmpada através de seu chaveamento, que tem seu controle por meio do pino 12 de GPIO do Raspberry Pi. A lâmpada é conectada ao relé em seu terminal normalmente aberto (NA), e a alimentação 127 VAC vinda da rede é conectada ao terminal comum. Para chavear o relé, é necessário alimentá-lo nos seus pinos de Vcc e GND.

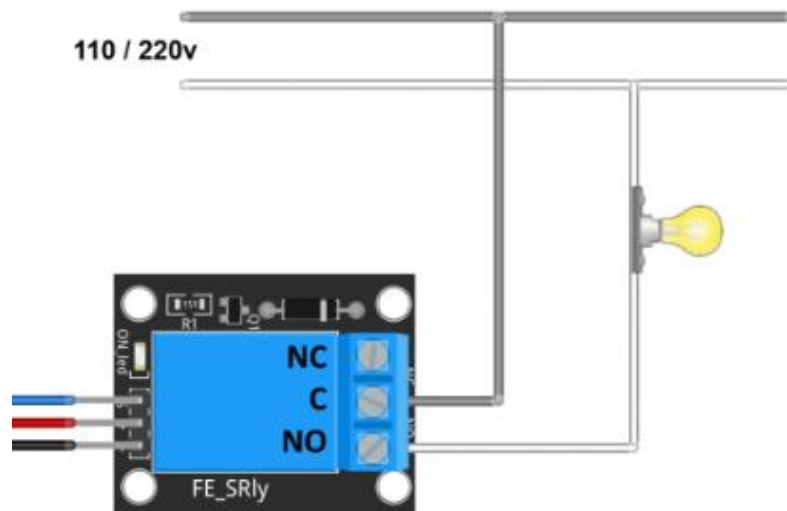


Figura 4.4: Esquema de ligação entre o relé e a lâmpada [30].

O custo dos equipamentos utilizados no desenvolvimento do projeto está listado na Tabela 4.1. O kit mencionado inclui a *case* para a placa, fonte de alimentação, cartão de memória e dissipadores para os *chips* do Raspberry.

Tabela 4.1: Custo para implementação do sistema. Fonte: Autoria própria.

Nº	Componente	Preço
1	<i>Kit Raspberry Pi 3B</i>	R\$ 489,00
2	Microfone USB	R\$ 39,90
3	Módulo relé	R\$ 10,00
4	Circuito infravermelho	R\$ 5,20
Total		R\$ 544,10

4.4 Desenvolvimento do Script Python

Para o desenvolvimento do script de gerenciamento, foi escolhida a linguagem de programação Python, que oferece grande quantidade de bibliotecas que permitem fazer uma interface de comando via fala. Dentre as possíveis, foi escolhida a *SpeechRecognition*, que é de fácil manuseio e permite a escolha entre diversas APIs de reconhecimento de fala, conforme explanado na seção 2.10. Para a interface do hardware do transmissor infravermelho, foi feito uso da versão Python da biblioteca LIRC, que permite o envio de comandos de múltiplos controles remotos. No caso do acionamento da lâmpada, foi utilizada a biblioteca RPi-GPIO, que permite a manipulação dos pinos digitais de entrada e saída usados no controle de periféricos.

O reconhecimento de fala é feito por meio de instâncias das classes *Recognizer* e *Microphone*. A instância da classe *Microphone* é responsável por captar a entrada de áudio via microfone, com o argumento *device_index*, conforme mostrado na linha 11 da Figura 4.5, indicando qual dispositivo deve ser usado como microfone, dentre os possíveis; e a instância da classe *Recognizer* é utilizada para converter o áudio em texto. Este passo é realizado na linha 22 da Figura 4.5, com o uso do método *recognize_google*, em associação com *adjust_for_ambient_noise* para tratar os ruídos do ambiente.

```

10  r = sr.Recognizer()
11  mic = sr.Microphone(device_index=1)
17      try:
18          print("Aguardando comando")
19          with mic as source:
20              r.adjust_for_ambient_noise(source)
21              audio = r.listen(source)
22              word = r.recognize_google(audio, language='pt-BR').lower()
23              print("Voce disse: " + word)

```

Figura 4.5: Trecho do script referente ao reconhecimento de fala. Fonte: Autoria própria.

Após a transcrição da fala para o texto, o resultado é comparado em diversos blocos *if-else*, em que é necessário haver uma correspondência exata entre o texto transcrito e o comando esperado pelo sistema. Caso seja compreendido, é executado o comando e o dispositivo é acionado. Caso contrário, é informada a mensagem “Não compreendido”.

Quando o comando reconhecido está relacionado aos dispositivos controlados por sinais infravermelhos, é feito uso do método *send_once*, da classe *Client* presente na biblioteca LIRC.

A linha 28 da Figura 4.6 mostra o uso do método para controlar a televisão. Observa-se que são especificados o arquivo de configuração “tv” no primeiro argumento do método `send_once`, e o botão desejado no segundo argumento. Neste caso, o botão escolhido foi `KEY_POWER`, responsável por ligar e desligar o aparelho. Para controlar o ar-condicionado, o argumento do arquivo de configuração passa a ser “aircon” (com seus códigos mapeado em *raw code*), enquanto o argumento referente aos botões se torna “power_on” ou “power_off”, dependendo da ação solicitada.

```

12 client = lirc.Client()
25         if word == "ligar televisão":
26             print("Reconhecido: ligar tv")
27             time.sleep(1)
28             client.send_once("tv", "KEY_POWER")

```

Figura 4.6: Trecho do script para controle da televisão. Fonte: Autoria própria.

Já para o acionamento da lâmpada, é feito uso da biblioteca `RPi.GPIO`. Para uso da biblioteca, é necessário realizar configurações prévias, como definição da função dos pinos, os desligamentos de avisos, e o modo de identificação dos pinos. Quando se verifica que o comando é referente ao acionamento da iluminação, é acionado o pino 12, utilizando a função `output` da biblioteca `GPIO`, conforme mostram as linhas 44 e 48 da Figura 4.7. No caso do relé a ser chaveado, que possui lógica invertida de acionamento, é enviado o nível lógico baixo para que o contato seja fechado, e nível lógico alto para que seja aberto.

```

6 GPIO.setmode(GPIO.BOARD)
7 GPIO.setwarnings(False)
8 GPIO.setup(12, GPIO.OUT, initial=GPIO.HIGH)
41         elif word == "acender lâmpada":
42             print("Reconhecido: ligar luz")
43             time.sleep(1)
44             GPIO.output(12, 0)
45         elif word == "apagar lâmpada":
46             print("Reconhecido: desligar luz")
47             time.sleep(1)
48             GPIO.output(12, 1)

```

Figura 4.7: Trecho do script para acionamento da lâmpada. Fonte: Autoria própria.

4.5 Ambiente de testes

A obtenção dos resultados foi feita através da montagem do sistema com auxílio de uma placa *protoboard*, em que foi posicionado o circuito com o transmissor de infravermelho, e a lâmpada, associada a um soquete, foi conectada ao módulo relé, com sua alimentação da rede, 127 VAC. Para realizar a transcrição de fala para texto, o Raspberry Pi foi conectado à rede *Wi-Fi* da residência. Para que os dispositivos fossem acionados pelo sistema, o *protoboard* foi posicionado previamente, com o LED apontado para cada aparelho, da mesma forma em que se utiliza um controle remoto convencional. O processo teve início com a execução do script via comando de terminal. A Figura 5.1 ilustra o protótipo do sistema montado para coleta dos dados.

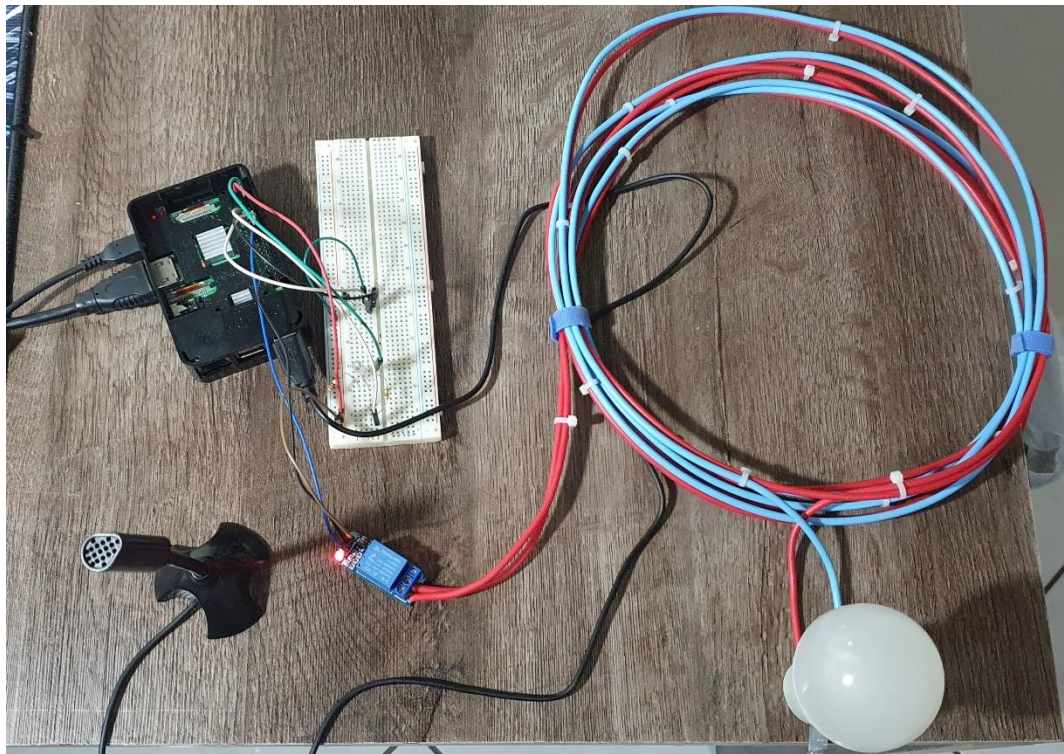


Figura 4.8: Ambiente de testes do sistema. Fonte: Autoria própria.

5 RESULTADOS

Para verificar a eficiência do reconhecimento de fala, cada comando foi pronunciado cinco vezes. Com o objetivo de avaliar a interferência causada pelo afastamento entre o usuário

e o microfone, foram realizados os testes em três distâncias diferentes: 10 cm, 60 cm e 1 m. A Tabela 5.1 apresenta os resultados obtidos.

Tabela 5.1: Quantidade de acertos dos comandos com base nas distâncias entre usuário e microfone.
Fonte: Autoria própria.

Comandos \ Distâncias	10 cm	60 cm	100 cm
Ligar televisão	5	4	4
Desligar televisão	5	3	3
Ligar ar-condicionado	5	5	4
Desligar ar-condicionado	5	4	3
Acender lâmpada	5	5	4
Apagar lâmpada	5	5	4

É possível identificar, através dos resultados da Tabela 5.1, uma redução na quantidade de acertos quando a distância entre usuário e microfone aumenta. A uma distância pequena, de 10 cm, o sistema apresenta um bom índice de acertos, com 5 comandos reconhecidos em 5 tentados, para todos os comandos. A queda no aproveitamento observada na distância de 60 cm foi leve, apresentando apenas um caso com 3 acertos. Já na distância de 100 cm, não houve casos em que um comando foi compreendido 5 vezes, sendo que nos comandos “Desligar televisão” e “Desligar ar-condicionado” o comando foi entendido somente 3 vezes.

As Figuras 5.2, 5.3 e 5.4 ilustram o percentual de acertos dos comandos, para cada distância.

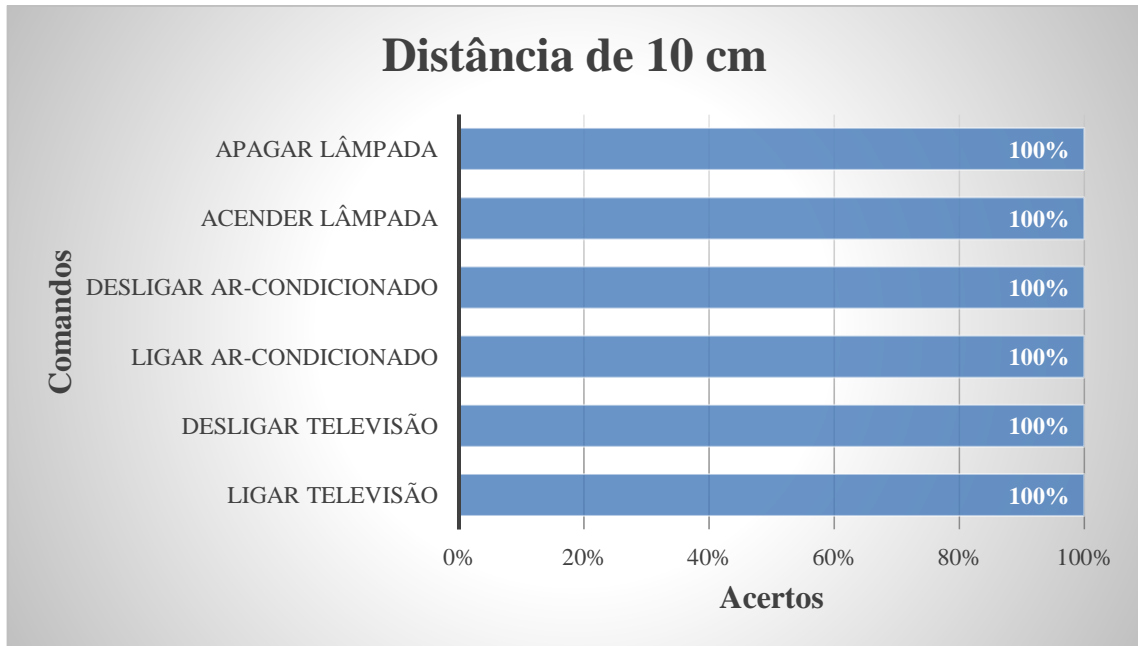


Figura 5.1: Porcentagem de acertos dos comandos com a distância de 10 cm. Fonte: Autoria própria.

É possível observar, a partir do gráfico da Figura 5.2, que o aproveitamento de acertos a uma distância de 10 cm é alto, com todos os comandos sendo compreendidos nas 5 vezes em que foram pronunciados, indicando o melhor caso de uso do sistema.

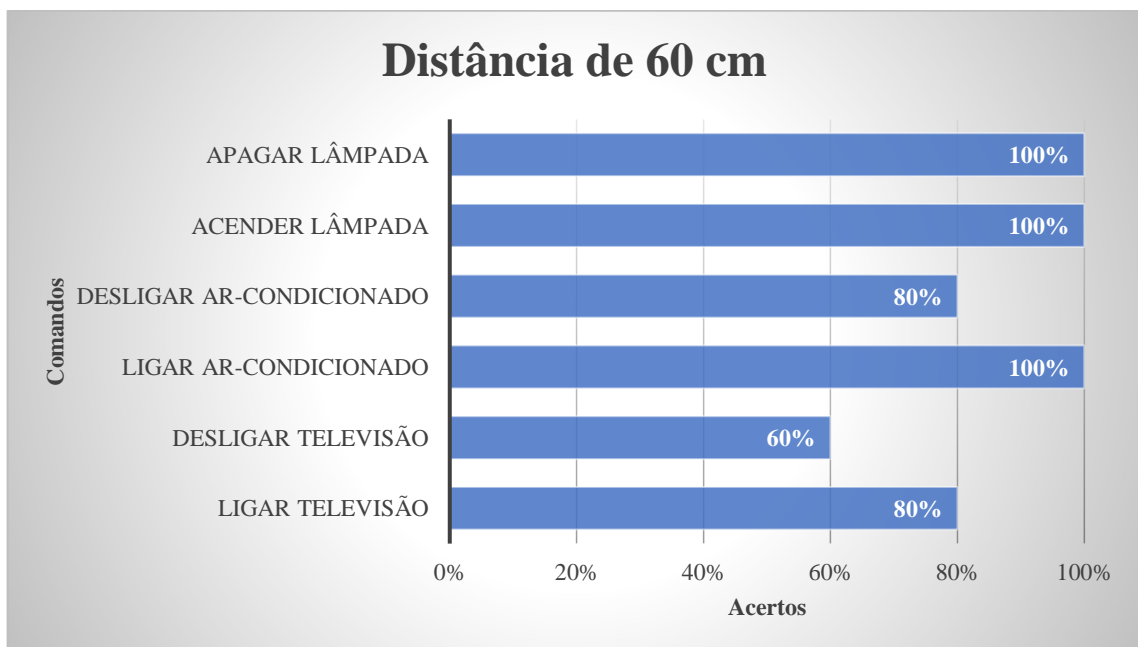


Figura 5.2: Porcentagem de acertos dos comandos com a distância de 60 cm. Fonte: Autoria própria.

O gráfico da Figura 5.3 mostra que, ao aumentar a distância, houve uma diminuição no número de acertos, se comparado com o resultado da distância de 10 cm, em 3 dos 6 comandos, sendo que nos comandos “Desligar ar-condicionado” e “Ligar televisão”, o percentual de

acertos caiu para 80%, enquanto no comando “Desligar televisão”, o índice passou a ser 60%. Os demais comandos mantiveram-se com aproveitamento máximo.

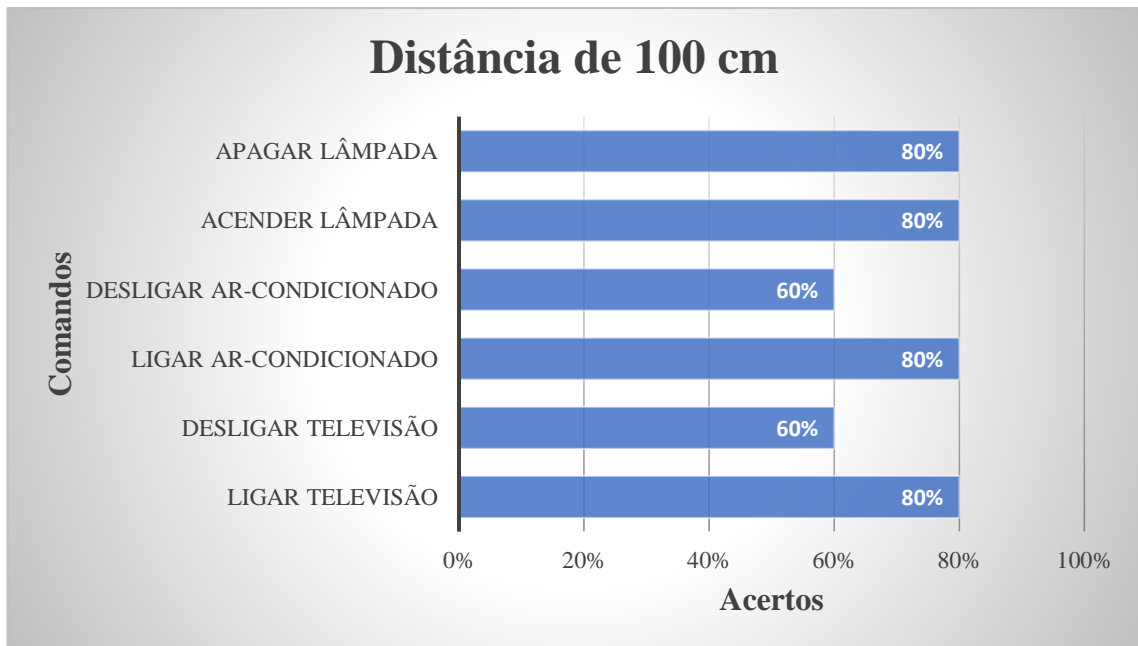


Figura 5.3: Porcentagem de acertos dos comandos com a distância de 100 cm. Fonte: Autoria própria.

Já no resultado obtido para a distância de 100 cm, ilustrado na Figura 5.4, observa-se que nenhum comando atingiu a porcentagem máxima, apresentando uma queda maior, se comparado às distâncias de 10 cm e 60 cm. Os comandos “Apagar lâmpada”, “Acender lâmpada”, “Ligar ar-condicionado” e “Ligar televisão” apresentaram um índice de acertos de 80%, deixando de identificar somente uma vez em 5 tentadas, e os comandos “Desligar ar-condicionado” e “Desligar televisão” tiveram seu aproveitamento em 60%.

A Figura 5.5 mostra a tendência da média de eficiência de acordo com o aumento da distância.

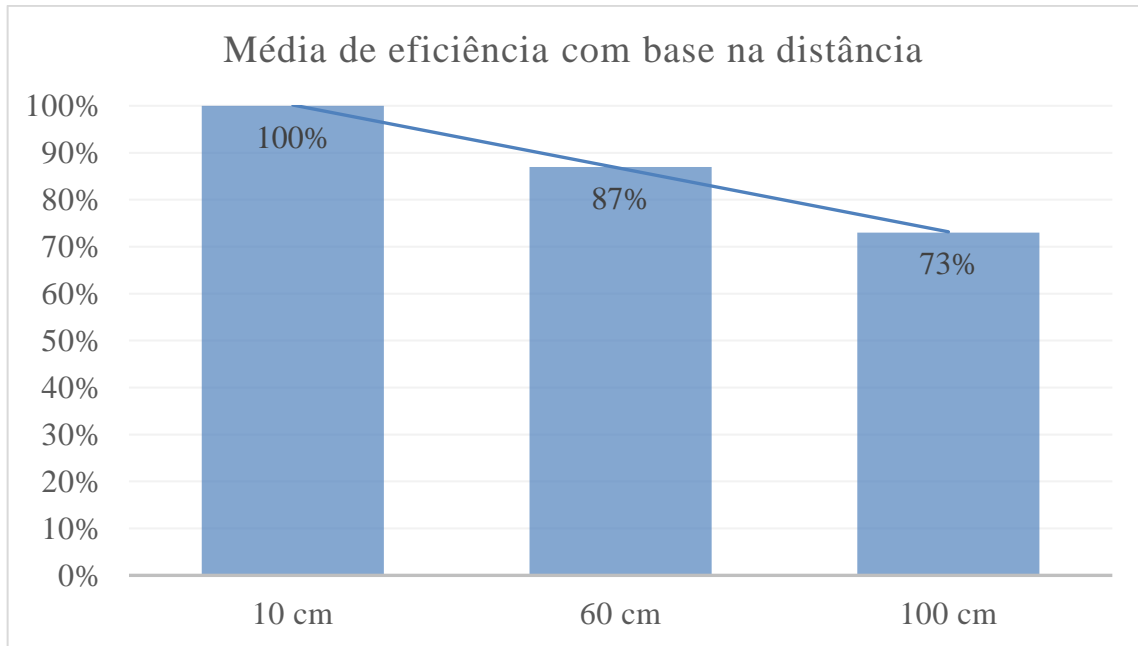


Figura 5.4: Tendência da média de eficiência em função da distância. Fonte: Autoria própria.

É possível visualizar, a partir do resultado apresentado na Figura 5.5, a diminuição da média geral de acertos referentes à compreensão dos comandos. A linha de tendência traçada indica uma redução linear na porcentagem média de acertos, conforme o aumento da distância entre o usuário e o microfone.

5.1 Discussão

Diante dos resultados obtidos, é observável que o sistema possui boa eficiência na compreensão dos comandos quando os mesmos são pronunciados a uma distância pequena do usuário, com uma média de 100% de acertos. Ao aumentar a distância, percebe-se uma queda no rendimento, chegando a uma média de 87% de acertos. No último caso, com a maior distância dos testes, a eficiência tem uma queda mais acentuada, com média por volta de 73% de acertos. Estes resultados apontam que o sistema tem seu melhor funcionamento a curtas distâncias, em que há maior compreensão dos comandos pronunciados. Há a possibilidade de utilizá-lo a distâncias maiores, porém com relativa dificuldade, visto que alguns comandos não têm a mesma eficiência de reconhecimento nestas condições.

Dentre os fatores que contribuem para a queda da eficiência, pode-se mencionar o sotaque da pessoa que pronuncia as palavras. No caso de pessoas naturais da região Norte do

Brasil, pronunciar palavras que contêm o som da letra “s” em seu meio ou fim pode produzir ruídos – ou chiados. A transcrição de palavras como “desligar” pode ter seu resultado diferente do esperado, pois o sistema de reconhecimento de fala remove o que foi compreendido como ruído. Desta forma, ao invés de receber na saída a palavra “desligar”, obtém-se o resultado “ligar”. Como visto na Tabela 5.1, comandos que contêm a palavra “desligar” são menos reconhecidos do que outros comandos que não apresentam palavras com o som de “s”.

Outro fator que tem influência no resultado obtido após a conversão de fala em texto é a qualidade da captação de áudio do microfone utilizado. Microfones com maior sensibilidade e melhor captação de áudio têm o preço elevado, tornando-se inviável a sua utilização no projeto. Além disso, nem todos são compatíveis com a plataforma de desenvolvimento escolhida. Portanto, o uso de microfones mais simples pode proporcionar áudios com baixo volume, comprometendo a eficiência da transcrição de fala em texto, acarretando em resultados abaixo do esperado em distâncias maiores.

6 CONCLUSÕES

O uso de tecnologias de automação residencial para promover a acessibilidade a pessoas com mobilidade reduzida tem papel fundamental no objetivo de promover ambientes de igualdade entre todos os indivíduos na sociedade. A proposta de utilizar uma interface controlada por voz se mostrou de grande praticidade, visto que a fala é um elemento comum entre essas pessoas, e permite a realização das atividades com maior rapidez.

O desenvolvimento do projeto não apresentou um alto grau de complexidade, possibilitando a sua utilização em ambientes domésticos sem maiores dificuldades. Isto foi possível devido à facilidade proporcionada pelas plataformas escolhidas e seus periféricos, que são de fácil configuração. O software de gerenciamento do sistema também é de fácil manipulação, permitindo que outros dispositivos sejam controlados pelo sistema, além dos mencionados ao decorrer do trabalho.

O trabalho atingiu seu objetivo de controlar dispositivos utilizando a fala, sendo possível comandar uma televisão, um ar-condicionado e uma lâmpada. Apesar da queda de eficiência no reconhecimento de fala, o sistema apresentou um índice aceitável de acertos dos comandos, atingindo a média de 87%. Este problema pode ser corrigido com a substituição do microfone atual por um de maior qualidade, a depender do orçamento disponível, visto que o problema reside no hardware, e não na ferramenta do *Google* de reconhecimento de fala.

A proposta de utilizar ferramentas de automação para a melhoria da qualidade de vida de PcD ou pessoas com mobilidade reduzida deve ser expandida, de forma que mais pesquisadores engajem na busca pela igualdade social entre todos os indivíduos, aplicando seus conhecimentos adquiridos pelo bem da sociedade.

6.1 Trabalhos futuros

Além da substituição do microfone para obtenção de melhores resultados, é possível vislumbrar para o futuro a adição de mecanismos que proporcionem melhorias na interação entre PcDs – ou pessoas com mobilidade reduzida – e as tecnologias utilizadas para promover a acessibilidade. Neste contexto, são apresentadas sugestões para a continuidade do desenvolvimento do trabalho.

- a. Proporcionar um alcance maior a públicos falantes de outros idiomas, por meio da possibilidade apresentada pela Google Speech API de configurar o idioma de reconhecimento;
- b. Aumentar a gama de dispositivos a serem controlados pelo sistema, fazendo uso de novos módulos periféricos para controle destes dispositivos, como *bluetooth*, que poderia fazer interface com aparelhos domésticos como a máquina de lavar roupas;
- c. Incluir mecanismos de reconhecimento de voz, através da implementação de softwares de inteligência artificial e aprendizado de máquina, que fariam o reconhecimento pela identificação do padrão de voz do usuário de acordo com o uso do sistema.

REFERÊNCIAS

- [1] KOC, T.C.; TEKER, S. Industrial revolutions and its effects on quality of life. **PressAcademia**, v. 9, 2019, p. 304-311. Disponível em: <<https://www.readcube.com/articles/10.17261%2Fpressacademia.2019.1109>>. Acesso em: 07 abr. 2022.
- [2] GUNGE, V. S.; YALAGI, P. S. Smart Home Automation: A Literature Review. **International Journal of Computer Applications**, n. 1, p. 6-10, 2016. Disponível em: <<https://www.ijcaonline.org/proceedings/rtdm2016/number1/24679-2568>>. Acesso em: 7 abr. 2022.
- [3] MTSHALI, P.; KHUBISA, F. A smart home appliance control system for physically disabled people. 2019 **Conference on Information Communications Technology and Society (ICTAS)**, p 1-5, 2019. Disponível em: <10.1109/ICTAS.2019.8703637>. Acesso em: 7 abr. 2022.
- [4] STEFANOV, D. H.; BIEN, Z.; BANG, W. The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives. **IEEE Transactions On Neural Systems And Rehabilitation Engineering**, v. 12, n. 2, p. 228-250, 2004. Disponível em: <<https://ieeexplore.ieee.org/document/1304864>>. Acesso em: 07 abr. 2022.
- [5] VAJPAI, J.; BORA, A. Industrial applications of automatic speech recognition systems. **Ijera**, v. 6, n. 3, p 88-95, 2016. Disponível em: <https://www.ijera.com/papers/Vol6_issue3/Part%20-%201/O6301088095.pdf>. Acesso em: 25 jun. 2021.
- [6] IBGE. **Pessoas com deficiência**. [S. l.], 2010. Disponível em: <<https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/20551-pessoas-com-deficiencia.html>>. Acesso em: 2 jul. 2021.
- [7] MPPR. **Conceitos de deficiência**. [S. l.], 2016. Disponível em: <<https://pcd.mppr.mp.br/pagina-41.html#>>. Acesso em: 2 jul. 2021.
- [8] UNIFESP. **Deficiência Física /Mobilidade Reduzida**. [S. l.], 25 set. 2020. Disponível em: <https://acessibilidade.unifesp.br/recursos/deficiencia-fisica-mobilidade-reduzida#:~:text=Pessoa%20com%20mobilidade%20reduzida%3A%20aquela,coordena%C3%A7%C3%A3o%20motora%20ou%20da%20percep%C3%A7%C3%A3o>. Acesso em: 7 abr. 2022.
- [9] AMOS, D. **The Ultimate Guide To Speech Recognition With Python**. Real Python, 2016. Disponível em: <<https://realpython.com/python-speech-recognition/>>. Acesso em: 7 jul. 2021.
- [10] BRASIL. **Decreto nº 3956, de 8 de outubro de 2001**. Promulga a Convenção Interamericana para a Eliminação de Todas as Formas de Discriminação contra as Pessoas Portadoras de Deficiência. 2001. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/2001/d3956.htm>. Acesso em: 7 abr. 2022.

- [11] BRASIL. **Decreto nº 5296, de 2 de dezembro de 2004**. Regulamenta as Leis nos 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências. [S. l.], 2 dez. 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 10 mar. 2022.
- [12] MACHADO, M. H.; LIMA, J. P. Avaliação multicritério da acessibilidade de pessoas com mobilidade reduzida: um estudo na região central de Irajubá (MG). **Revista Brasileira de Gestão Urbana**, v. 7, n. 3, p. 368-382, 2015. Disponível em: <<https://doi.org/10.1590/2175-3369.007.003.AO08>>. Acesso em: 5 fev. 2022.
- [13] UNIVERSIDADE FEDERAL DO CEARÁ. **Conceito de Acessibilidade**. [S. l.], 2010. Disponível em: <<https://www.ufc.br/acessibilidade/conceito-de-acessibilidade>>. Acesso em: 7 abr. 2022.
- [14] V SIMPÓSIO DE PÓS-GRADUAÇÃO, 2018, Uberaba. **Educação Inclusiva e Acessibilidade**. 2018. Disponível em: <<https://iftm.edu.br/simpos/2018/anais/668-%20Pronto%20ANAIS.pdf>>. Acesso em: 9 abr. 2022.
- [15] Núcleo de Informação e Coordenação do Ponto BR. **Acessibilidade e Tecnologias: um panorama sobre acesso e uso de Tecnologias de Informação e Comunicação por pessoas com deficiências no Brasil e na América Latina**. Cadernos NIC.br Estudos Setoriais. São Paulo, 2020. Disponível em: <<https://cetic.br/publicacao/acessibilidade-e-tecnologias-um-panorama-sobre-acesso-e-uso-de-tecnologias-de-informacao-e-comunicacao-por-pessoas-com-deficiencia-no-brasil-e-na-america-latina/>>. Acesso em: 9 abr. 2022.
- [16] SOUSA, N. V. **Automação Residencial por Comandos de Voz para Pessoas com Mobilidade Reduzida**. 2018. 101 f. Dissertação (Mestrado em Engenharia Elétrica) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB, João Pessoa, 2018.
- [17] CHAVAN, R.; SABLE, G. S. An overview of speech recognition using HMM. **International Journal of Computer Science and Mobile Computing**, v. 2, n. 6, p. 233-238, 2013. Disponível em: <<https://www.ijcsmc.com/docs/papers/June2013/V2I6201369.pdf>>. Acesso em: 9 abr. 2022.
- [18] PIMENTEL, V. C. A. **Plataforma embarcada de reconhecimento automático da fala para o auxílio de pessoas com mobilidade reduzida**. 2016. 239 f. Dissertação (Mestrado em Engenharia Elétrica) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB, João Pessoa, 2016.
- [19] RASPBERRY Pi OS, 2021. Disponível em: <<https://www.raspberrypi.com/documentation/computers/os.html>>. Acesso em: 10 abr. 2022.
- [20] FILIPEFLOP. **LED Emissor Infravermelho IR 5mm**. [S. l.], 2021. Disponível em: <<https://www.filipeflop.com/produto/led-emissor-infravermelho-ir-5mm/>>. Acesso em: 13 abr. 2022.

[21] SMARTKITS. **Receptor Infravermelho VS1838B**. [S. l.], 2022. Disponível em: <<https://www.smartkits.com.br/receptor-infravermelho-vs1838b>>. Acesso em: 12 abr. 2022.

[22] APLUS TOPPER. **Advantages and disadvantages of python**: python language advantages, disadvantages and its applications. 7 jan. 2022. Disponível em: <https://www.aplustopper.com/advantages-and-disadvantages-of-python/>. Acesso em: 25 fev. 2022.

[23] ALURA. **POO: o que é programação orientada a objetos?** [S. l.], 23 out. 2019. Disponível em: <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>. Acesso em: 7 abr. 2022.

[24] AMOS, D. The ultimate guide to speech recognition with python. [S. l.]: Real Python, 7 jan. 2022. Disponível em: <https://realpython.com/python-speech-recognition/#how-speech-recognition-works-an-overview>. Acesso em: 25 fev. 2022.

[25] PYTHON PACKAGE INDEX. **Library for performing speech recognition, with support for several engines and APIs, online and offline**. [S. l.], 5 dez. 2017. Disponível em: <<https://pypi.org/project/SpeechRecognition/>>. Acesso em: 2 jul. 2021.

[26] LINUX INFRARED REMOTE CONTROL. **What is LIRC?** 2017. Disponível em: <https://www.lirc.org/>. Acesso em: 16 mar. 2022.

[27] LIRC Python Package. Disponível em: <<https://lirc.readthedocs.io/en/latest/#>>. Acesso em: 16 mar. 2022.

[28] YAYLA, A. *et al.* Development of a remote laboratory for an electronic circuit design and analysis course with increased accessibility by using speech recognition technology. **Comput. Appl. Eng. Educ.**, p. 1-14, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22340>>. Acesso em: 2 jul. 2021.

[29] International Conference On Electrical, Computer And Communication Engineering (ECCE), 2019, Chatigão. **Voice Recognition based Intelligent Wheelchair and GPS Tracking System**, 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8679163>>. Acesso em: 10 abr. 2022.

[30] ARDUINO E CIA. **Ligando uma lâmpada com módulo relé Arduino**. [S. l.], 2013. Disponível em: <<https://www.arduinoecia.com.br/ligando-uma-lampada-com-modulo-rele-arduino/>>. Acesso em: 11 abr. 2022.

APÊNDICES

APÊNDICE A – Arquivo de configuração dos controles remotos da biblioteca LIRC

```
begin remote
```

```
name aircon
flags RAW_CODES
eps          30
aeps        100
```

```
gap          8988
```

```
begin raw_codes
```

```
name power_on
  548  17941  2982  8978  497  496
  477   1512   503   492  479  515
  480   516   507   488  478  541
  478   517   460   534  478 1514
  477   520   476   518  476 1515
  453   542   474   520  476 1515
  501  1491   501  1488  503 1489
  480  1513   500   518  477  518
  453   541   478   520  475  521
  473   520   475   519  450  545
  479   518   473   520  475  519
  503   493   502   493  502  492
  501   495   478   516  478  518
  500   496   503   515  454  540
  477   518   478   500  493  522
  477   518   475   520  449  545
  449   546   474   521  477  519
  500   494   501   493  478  517
  501  1489   503  1490  500 1514
  482  1509   455  3019 2955 9026
  478  1512   476   520  450  545
  473   522   474   520  478  518
  501   493   502   494  500  493
  502  1489   479   520  498  495
  481  1530   478   518  479 1514
  475  1517   474  1519  447 1541
  475  1517   503  1487  480  517
  501   493   502   494  477  516
  501   494   500   519  476  519
  454   541   480   515  478  521
  474   520   458   537  449  544
  450   551   467   524  473  521
```

480	514	501	494	502	492
502	494	478	517	500	497
476	542	475	497	507	509
477	517	478	518	477	520
476	520	449	545	450	545
450	545	474	522	474	519
478	518	477	517	477	3020
2958	8997	478	1513	491	506
499	496	498	519	454	539
455	540	455	544	452	541
476	519	475	1517	474	521
473	521	475	1515	502	1489
480	516	502	1488	478	517
479	1516	475	1535	456	1538
453	1538	475	1528	463	1517
474	1515	481	1511	479	517
477	516	479	517	500	1491
478	1515	474	1535	456	539
455	542	476	518	477	518
475	521	451	1540	475	520
477	1515	479	517	499	1490
480	1511	480	520	498	1491
477	1535	477	518	455	542
453	543	475	519	476	519
461	536	472	522	474	1514
503	1490	500	1490	480	1512
502					

name power_off

544	17923	2960	8995	478	516
479	1516	476	539	455	540
456	542	453	543	452	544
477	518	474	520	475	1515
476	519	478	518	478	1511
480	1512	479	516	479	1514
478	1516	474	1537	455	1539
452	1538	477	519	475	524
470	520	475	521	476	518
479	515	480	517	477	521
474	517	476	516	479	518
499	500	472	541	453	540
455	540	478	519	453	542
475	520	475	520	475	519
475	521	474	520	478	517
478	517	478	516	479	516
479	515	480	516	479	517
478	518	499	518	454	544
451	539	455	543	452	1537
477	1514	479	3018	2959	8998

```

476 1515 480 516 479 516
502 492 479 516 479 517
479 517 477 544 451 540
476 1513 456 542 454 542
450 1540 476 519 476 1517
474 1515 480 1511 479 1510
481 1511 479 1515 476 542
452 541 455 539 454 543
454 542 452 541 476 519
476 519 475 519 476 522
476 515 480 516 481 515
480 514 482 513 479 516
479 517 478 518 477 541
453 541 454 540 456 540
454 542 453 543 474 518
477 520 475 520 474 521
475 519 479 517 478 515
479 516 479 516 477 516
479 516 478 519 477 3020
2979 8999 455 1535 457 541
454 541 451 544 476 518
476 519 476 519 490 505
477 517 480 1510 504 491
480 515 479 1512 479 519
476 1515 476 1534 457 540
454 1538 476 1515 477 1514
480 1511 479 1510 505 1488
480 1511 481 1512 480 516
477 540 477 516 456 1536
455 1536 478 1513 478 519
479 516 475 518 479 516
480 516 479 1512 480 516
479 1511 480 517 505 1507
457 1535 456 542 453 1537
477 1514 476 520 477 519
475 517 478 516 481 515
480 514 480 516 479 516
479 516 479 1513 482 1531
478

```

```
end raw_codes
```

```
end remote
```

```
begin remote
```

```

name tv
bits 32
flags SPACE_ENC|CONST_LENGTH

```

```
eps          30
aeps        100

header      9018 4567
one         504 1742
zero        504  622
ptrail      506
repeat      9027 2296
gap         108645
toggle_bit_mask 0x0
frequency   38000
```

```
begin codes
```

```
KEY_POWER
```

```
end codes
```

```
0x20DF10EF 0xEAD43F00
```

```
end remote
```

APÊNDICE B – Código Python

```

1 import speech_recognition as sr
2 import RPi.GPIO as GPIO
3 import lirc
4 import time
5
6 GPIO.setmode(GPIO.BOARD)
7 GPIO.setwarnings(False)
8 GPIO.setup(12, GPIO.OUT, initial=GPIO.HIGH)
9
10 r = sr.Recognizer()
11 mic = sr.Microphone(device_index=1)
12 client = lirc.Client()
13
14 try:
15     while True:
16         time.sleep(0.5)
17         try:
18             print("Aguardando comando")
19             with mic as source:
20                 r.adjust_for_ambient_noise(source)
21                 audio = r.listen(source)
22                 word = r.recognize_google(audio, language='pt-BR').lower()
23                 print("Voce disse: " + word)
24
25                 if word == "ligar televisão":
26                     print("Reconhecido: ligar tv")
27                     time.sleep(1)
28                     client.send_once("tv", "KEY_POWER")
29                 elif word == "desligar televisão":
30                     print("Reconhecido: desligar")
31                     time.sleep(1)
32                     client.send_once("tv", "KEY_POWER")
33                 elif word == "ligar ar-condicionado" or word == "ligar ar
condicionado":
34                     print("Reconhecido: ligar ar")
35                     time.sleep(1)
36                     client.send_once("aircon", "power_on")
37                 elif word == "desligar ar-condicionado" or word == "desligar ar
condicionado":
38                     print("Reconhecido: desligar ar")
39                     time.sleep(1)
40                     client.send_once("aircon", "power_off")
41                 elif word == "acender lâmpada":
42                     print("Reconhecido: ligar luz")
43                     time.sleep(1)
44                     GPIO.output(12, 0)

```

```
45         elif word == "apagar lâmpada":
46             print("Reconhecido: desligar luz")
47             time.sleep(1)
48             GPIO.output(12, 1)
49         else:
50             print("Nao compreendido")
51     except sr.UnknownValueError:
52         print("Could not identify command.")
53
54 except KeyboardInterrupt:
55     print("\nKeyboard exit")
56     GPIO.cleanup()
```

