



UNIVERSIDADE DO ESTADO DO AMAZONAS-UEA
ESCOLA SUPERIOR DE TECNOLOGIA-EST
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

JOÃO PEDRO COELHO DOS SANTOS

**SISTEMA DE CONTROLE DOMÓTICO POR VOZ PARA PESSOAS COM
DEFICIÊNCIAS MOTORAS.**

Manaus

2019

JOÃO PEDRO COELHO DOS SANTOS

**SISTEMA DE CONTROLE DOMÓTICO POR VOZ PARA PESSOAS COM
DEFICIÊNCIAS MOTORAS.**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada a banca avaliadora do Curso de Engenharia de Controle e Automação da Escola Superior de tecnologia da Universidade Estadual do Amazonas, como pré-requisito para a obtenção do título de bacharel de Engenharia de Controle e Automação.

Orientador: Moisés Pereira Bastos, Me.

MANAUS

2019

JOÃO PEDRO COELHO DOS SANTOS

**SISTEMA DE CONTROLE DOMÓTICO POR VOZ PARA PESSOAS COM
DEFICIÊNCIAS MOTORAS.**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada a banca avaliadora do Curso de Engenharia de Controle e Automação da Escola Superior de tecnologia da Universidade Estadual do Amazonas, como pré-requisito para a obtenção do título de bacharel de Engenharia de Controle e Automação.

Orientador: Moisés Pereira Bastos, Me.

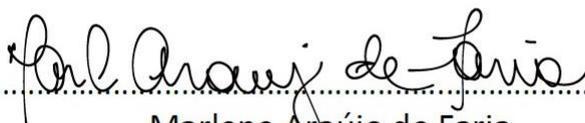
Aprovado em 25 de Novembro de 2019



Moisés Pereira Bastos



Cleto Cavalcante De Souza Leal



Marlene Araújo de Faria

Manaus

2019

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo, por ter aberto essa porta e me dado a oportunidade de estudar na Universidade do Estado do Amazonas, mesmo não tendo muita confiança que conseguiria adentrar em uma universidade no ano de 2013, por ter me conduzido até esta etapa tão importante em minha vida, por ter me abençoado com saúde, força e muita paciência, a jornada não foi fácil mas com certeza foi muito produtiva. A Ele toda a glória!

A minha família por todo incentivo moral, espiritual e não menos importante financeiramente. Ressaltando meus pais que em momentos que eu me encontrava desmotivado me incentivaram a continuar, aos meus tios Edivaldo da Silva Alves e Ana Célia dos Santos Alves, que no momento em que mudei para Manaus me acolheram em sua casa e me deram um lar e aos meus primos Thiago Alves e Edivaldo Junior que durante os primeiros anos da faculdade dividimos momentos especiais e difíceis juntos.

Ao meu orientador, Me. Moisés Pereira Bastos, que mesmo em momentos de desespero por não tê-lo perto, sempre passou a segurança que tudo daria certo e mesmo com todas as adversidades seria possível concluir essa monografia, por sua paciência e orientação, por sempre transmitir o máximo de conhecimento em suas aulas e por estar sempre disponível a orientar seus alunos.

A Escola Superior de Tecnologia EST/UEA e todo o seu corpo docente, por me proporcionarem todo o aprendizado necessário para a minha formação.

Agradeço também aos meus amigos, principalmente ao Thiago dos Santos Alves, Larissa Cardoso, Márcia de Souza que me ajudaram aconselhando, ajudando no desenvolvimento desta monografia, e aos demais por me acompanharem nesses anos de estudo, por todo apoio, sobretudo nos momentos que mais precisei.

“Porque a sua ira dura só um momento; no seu favor está a vida. O choro pode durar uma noite, mas a alegria vem pela manhã.”

Salmos 30:5

RESUMO

Processos de evolução e adaptação são temas presentes no cotidiano na vida do ser humano, inúmeras vezes a humanidade teve ou tem que se adaptar as suas limitações, seja ela a dificuldade de se locomover em distancias grandes em o menor tempo possível, seja conversar com alguém que se encontre distante, sempre procurou-se superar os obstáculos e trazer maior comodidade a vida de todos. Neste trabalho foi desenvolvido um protótipo de sistema de controle domótico por voz para acionamento e controle de eletrodomésticos e iluminação para pessoas com deficiências motoras. O protótipo para fim de testes foi implementado, visando verificar a sua funcionalidade, foi utilizado o protocolo MQTT na comunicação entre os clientes publicadores e os clientes assinantes. Para a elaboração do protótipo foi utilizado o módulo de reconhecimento de voz, que ao receber o comando o mesmo fosse efetuado através do módulo de infravermelho ou o modulo relé. Após a realização dos testes, foi observado a resposta do módulo criado, que em sua avaliação quantitativa teve uma resposta satisfatória aos comandos ordenados pelo usuário que gravou os comandos, mas deixou a desejar quando outros usuários ordenaram o comando.

Palavras -chaves: MQTT. Módulo de Reconhecimento de Voz. Módulo Infravermelho. Módulo Relé.

ABSTRACT

Evolution and adaptation processes are themes present in the daily life of human beings, many times humanity has had or has to adapt to its limitations, be it the difficulty of moving over long distances in the shortest possible time, or talking to someone that is far away, always tried to overcome obstacles and bring greater comfort to everyone's life. In this work we developed a prototype home voice control system for the activation and control of appliances and lighting for people with motor disabilities. The prototype for end of tests was implemented, aiming to verify its functionality, the MQTT protocol was used in the communication between the publishing clients and the subscribing clients. For the elaboration of the prototype was used the speech recognition module, which upon receiving the command was made through the infrared module or the relay module. After performing the tests, the response of the created module was observed, which in its quantitative evaluation had a satisfactory response to the commands ordered by the user who recorded the commands, but left it to be desired when other users ordered the command.

Keywords: MQTT. Voice Recognition Module. Infrared module. Relay module.

LISTA DE ILUSTRAÇÕES

Figura 1- Casa Inteligente.....	18
Figura 2- Diagrama de Blocos de um Microcontrolador.....	19
Figura 3- Microcontrolador	20
Figura 4- Wemos D1 Mini	21
Figura 5- Arduino Uno	22
Figura 6- IDE Arduino	23
Figura 7- Sensor Ultrasônico.....	24
Figura 8- Módulo de Voz V3	25
Figura 9- Divisão dos Grupos.....	25
Figura 10- Estrutura MQTT	27
Figura 11- Servidor ClodMQTT.....	28
Figura 12- Espectro Eletromagnético	28
Figura 13- Relé.....	29
Figura 14- Protótipo pronto aberto	30
Figura 15- Diagrama de Caso de Uso.....	31
Figura 16- Diagrama da Arquitetura do Projeto	33
Figura 17- Módulo Central de reconhecimento de voz	34
Figura 18- Tela de gravação das assinaturas	35
Figura 19- Comandos treinados e sua funções	35
Figura 20- Arduino Nano Pinagem	36
Figura 21- Trecho do programa Nano: mudar de grupo.....	36
Figura 22- Tabela com Binário e o Decimal enviado pelo Nano	37
Figura 23- Trecho do programa Wemos: Publicar liga_desliga_lamp.....	37
Figura 24- Regulado de Tensão 5V	38
Figura 25- Optoacoplador PC817.....	38

Figura 26- Buzzer	39
Figura 27- Capacitor Eletrolítico.....	39
Figura 28- Diagrama da estrutura MQTT do projeto	40
Figura 29- Tópicos do projeto	40
Figura 30- Mensagem no servidor.....	41
Figura 31- Módulo emissor e receptor de Infravermelho com Wemos D1 Mini.....	41
Figura 32- Trecho do Programa Wemos infravermelho: recebe liga_desliga_ar.....	42
Figura 33- ESP-01 Pinagem	42
Figura 34- Módulo relé com ESP-01.....	43
Figura 35- Mensagem recebida pelo módulo relé	43
Figura 36- Teste do usuário 1	44
Figura 37- Teste do usuário 2.....	44
Figura 38- Figura 38- Teste dos usuários 3 e 4	45
Figura 39- Gastos usando ESP8266 NodeMcu	46
Figura 40- Gastos usando ESP8266 Wemos D1 Mini	46

LISTA DE ABREVEATURAS E SIGLAS

<i>IBGE</i>	<i>Instituto Brasileiro de Geografia e Estatística</i>
<i>CFTV</i>	<i>Circuito fechado de televisão</i>
<i>ULA</i>	<i>Unidade Lógica Aritmética</i>
<i>CPU</i>	<i>Central Processor Unit</i>
<i>PROM</i>	<i>Programmable Read Only Memory</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>WI-FI</i>	<i>Wireless Fidelity</i>
<i>GPIO</i>	<i>General Purpose Input/Output</i>
<i>I2C</i>	<i>Inter-Integrated Circui</i>
<i>UART</i>	<i>Universal Asynchronous Receiver/Transmitter</i>
<i>PWM</i>	<i>Pulse-Width Modulation</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>IOT</i>	<i>Internet Of Things</i>
<i>MQTT</i>	<i>Message Queue Telemetry Transport</i>
<i>QoS</i>	<i>Qualidade de Serviços</i>
<i>TCP/IP</i>	<i>Transmission Control Protocol/Internet Protocol</i>
<i>RXD</i>	<i>Recepção de dados</i>
<i>TXD</i>	<i>Transmissão de dados</i>
<i>VCC</i>	<i>Voltagem Corrente Contínua</i>
<i>RST</i>	<i>Reset</i>
<i>LED</i>	<i>light-emitting diode</i>
<i>GND</i>	<i>Graduated Neutral Density ou Terra</i>

SUMÁRIO

SUMÁRIO.....	11
1. INTRODUÇÃO.....	13
1.1 TEMA.....	14
1.2 PROBLEMA DA PESQUISA.....	14
1.3 HIPÓTESE	14
1.4 OBJETIVO	14
1.4.1 OBJETIVO GERAL	14
1.4.2 OBJETIVOS ESPECÍFICOS.....	15
1.5 JUSTIFICATIVA	15
1.6 METODOLOGIA.....	15
2. REFERENCIAL TEÓRICO	17
2.1 AUTOMAÇÃO RESIDENCIAL	17
2.2 MICROCONTROLADORES	18
2.2.1 ESP8266.....	20
2.2.2 ARDUINO	21
2.3 INSTRUMENTAÇÃO INDUSTRIAL	24
2.3.1 SENSORES	24
2.4 INTERNET DAS COISAS (IOT)	26
2.5 IR- INFRAVERMELHO (INFRARED)	28
2.6 RELÉ	29
2.7 TRABALHOS RELACIONADOS	29
2.7.1 SISTEMA DE CONTROLE DE DISPOSITIVOS PARA AUXILIAR PORTADORES DE DEFICIÊNCIA FÍSICA E/OU VISUAL.....	30

2.7.2	AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM ARDUINO MEGA E ETHERNET SHIELD	30
2.7.3	SISTEMA DE COMANDOS E IDENTIFICAÇÃO POR VOZ	31
3.	METODOLOGIA.....	33
3.1	MÓDULO DO COMANDO CENTRAL.....	33
3.1.1	MÓDULO DE VOZ.....	34
3.1.2	ARDUINO NANO	36
3.1.3	ESP8266 Wemos D1 Mini	37
3.1.4	REGULADOR DE TENSÃO, OPTOACOPLADORES, RESISTORES, BUZZER, CAPACITOR e LED's	38
3.2	NUVEM.....	40
3.3	MÓDULO INFRAVERMELHO.....	41
3.4	MÓDULO RELÉ.....	42
4.	RESULTADOS OBTIDOS	44
4.1	TESTES DO PROTÓTIPO	44
4.2	CUSTO DO PROJETO	45
5.	CONCLUSÃO	48
5.1	DIFICULDADES ENCONTRADAS	49
5.2	TRABALHOS FUTUROS	49
	REFERÊNCIAS BIBLIOGRÁFICAS	50
	APÊNDICE A- CÓDIGO DECIFRADOR DE IR	53
	APÊNDICE B- CÓDIGO ARDUINO NANO.....	54
	APÊNDICE C- CÓDIGO WEMOS D1 MINI.....	61
	APÊNDICE D - CÓDIGO MÓDULO RELÉ	68
	APÊNDICE E - CÓDIGO MÓDULO RELÉ	73

1. INTRODUÇÃO

A tecnologia evolui a cada dia, proporcionando mais conforto, economia e segurança para os seus usuários. Nos dias atuais a maioria das casas não possuem algum tipo de ambiente automatizado ou sistemas integrados para proporcionar maior conforto, economia ou segurança. Destaca-se ainda a ausência desse tipo de tecnologia nas casas de pessoas com algum tipo de deficiência, dificultando simples tarefas como ligar um ar-condicionado. Os sistemas estão cada vez mais acessíveis e as pessoas, no mínimo, já ouviram falar a respeito, seja por meio da mídia ou por algum amigo que já possui um sistema instalado em sua residência.

Os Censos Demográficos são a principal fonte de informação sobre a situação de vida da população em cada um dos municípios e localidades do País. As demais pesquisas domiciliares são, por exemplo, levantamentos por amostragem, que não são representativas para todos esses níveis geográficos. Um levantamento em 2010 feito pelo IBGE (Instituto Brasileiro de Geografia e Estatística), constatou que em torno de 13 milhões de pessoas no Brasil possuem algum tipo de deficiência motora, sendo que ao todo são 45,6 milhões de pessoas que possuem algum tipo de deficiência, isso representa em torno de 24% da população total do país (CASA ADAPTADA,2018).

Com os dados apresentados acima observamos que as pessoas que possuem algum tipo de deficiência necessitam de um olhar mais cuidadoso, visto que representam quase um quarto da população brasileira, se faz necessário o uso da tecnologia para facilitar a vida dessas pessoas. Não apenas elas serão beneficiadas visto que todos podem usufruir desse tipo de tecnologia, que será abordada neste presente trabalho de pesquisa e desenvolvimento.

Com o surgimento dos microcontroladores em meados da década de 70 para década de 80, tornou-se possível deixar nosso cotidiano mais cômodo a partir da ideia de automação residencial difundida através do termo domótica, que referencia a utilização de processos automatizados em casas, apartamentos e escritórios. Segundo Muratori (2013, p.70) com a integralização de sistemas é possível automatizar uma casa, tendo em vista que as tecnologias que hoje existem permitem a comunicação de vários equipamentos entre si.

Face ao exposto, o presente projeto de pesquisa será realizado com o objetivo de desenvolver e construir um protótipo para auxiliar pessoas com deficiência através da captação da voz e dos gestos, com a finalidade de acionar e controlar uma televisão, um condicionador

de ar e a iluminação, trazendo um custo mais baixo, comandos na língua portuguesa e compatibilidade com as tecnologias existentes.

1.1 TEMA

Sistema de controle doméstico por voz para pessoas com deficiências motoras.

1.2 PROBLEMA DA PESQUISA

Devido ao mercado não disponibilizar sistemas que facilitem a interação das pessoas com o ambiente doméstico por um preço acessível, comandos na língua portuguesa e compatibilidade com tecnologias anteriores (Infravermelho). Tal que as pessoas com deficiências motoras possam utilizar para facilitar a utilização dos eletrodomésticos e a iluminação de suas residências.

1.3 HIPÓTESE

É possível desenvolver um protótipo de sistema de controle doméstico usando o kit de prototipagem ESP8266 e o kit de prototipagem Arduino, sensor de voz e relés como atuadores, tecnologias que possuem um preço baixo em comparação aos produtos que estão disponíveis no mercado para realizar as funções que facilite a interação entre uma pessoa com deficiência com os eletrodomésticos e a iluminação em um determinado ambiente doméstico.

1.4 OBJETIVO

1.4.1 OBJETIVO GERAL

Desenvolvimento de um protótipo de sistema de controle doméstico por voz para acionamento e controle de eletrodomésticos e iluminação para pessoas com deficiências motoras.

1.4.2 OBJETIVOS ESPECÍFICOS

- Realizar levantamento bibliográfico referente aos assuntos abordados;
- Desenvolver o módulo central de controle;
- Desenvolver módulo infravermelho e relé;
- Implementar o IOT com armazenamento em nuvem utilizando o protocolo MQTT;
- Realizar testes.

1.5 JUSTIFICATIVA

O presente projeto se justifica pela aplicação junto à sociedade se justifica pela fabricação de um sistema mais acessível financeiramente, e que permitirá aos deficientes uma maior liberdade e autonomia na utilização dos eletrodomésticos e a iluminação no ambiente em que o sistema estiver implantado. Tendo em vista que os dispositivos oferecidos no mercado como Google Home e Alexa, não possuem compatibilidade com algumas tecnologias antigas e os comandos em sua maioria são em inglês, apenas o Alexa com alguns comandos em português.

Para a sociedade acadêmica o aprofundamento dos conhecimentos adquiridos no curso de engenharia de controle e automação, tais como: Linguagem e programação I e II, eletrônica analógica, eletrônica digital, circuitos elétricos I e II e instrumentação industrial.

1.6 METODOLOGIA

O presente trabalho será uma Pesquisa Aplicada, cujo objetivo será a realização de uma pesquisa exploratória sobre o conteúdo bibliográfico, bem como de laboratório adquiridos sobre o assunto. Os procedimentos técnicos a serem utilizados serão os de pesquisa bibliográfica e experimental. O método de abordagem a ser utilizado, será o hipotético-dedutivo e a elaboração seguirá o método de procedimento monográfico. Para coleta de dados será utilizada documentação indireta, com auxílio de documentos primários e secundários, e a análise e interpretação de seus dados será qualitativa.

Inicialmente, serão realizadas pesquisas bibliográficas na área de automação residencial, plataforma ESP8266, linguagem de programação, circuitos elétricos, eletrônica digital e analógica, sensores especificamente de voz, com o intuito de coletar informações acerca de como será executada a comunicação entre o software e hardware no projeto. A pesquisa em

laboratório será feita, para coleta de dados e a obtenção de respostas do andamento dos projetos, depurar possíveis erros e problemas que eventualmente ocorram durante a montagem do protótipo.

Após a realização da seleção do material para a construção do protótipo, esses materiais serão um kit de prototipagem ESP8266 Wemos D1 Mini, sensores de voz (voice recognition v3), relés, módulos ESP8266-01, emissores e receptores de infravermelho. Será realizado o desenvolvimento de um circuito auxiliar que irá simular o protótipo, tendo em vista que isso facilitará a montagem de um protótipo com a menor quantidade de erros possíveis no hardware e no software, será construída um recipiente a partir de uma impressora 3D para comportar o hardware do projeto. Será testado tanto o circuito auxiliar como o protótipo finalizado em um cômodo da casa do aluno João Pedro, o autor em questão.

Será elaborado um algoritmo na linguagem processing, que é aceita em Arduino, com a lógica necessária para a execução dos comandos para o acionamento e controle da iluminação e dos eletrodomésticos em teste. Utilizando assim as seguintes ferramentas o software do Arduino e das ferramentas físicas alicates de corte, chaves entre outros.

Após testes de validação do código de programação, será realizado a execução do mesmo, com toda estrutura montada a fim de verificar como se comporta o programa com todas as funções sendo executadas e o desempenho do hardware.

2. REFERENCIAL TEÓRICO

2.1 AUTOMAÇÃO RESIDENCIAL

Definição de automação industrial segundo Muratori (2013): “[...] É o conjunto de serviços proporcionados por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação [...] e conforto de uma habitação [...]”.

Também conhecida como domótica na Europa, que se origina da palavra domitique, junção da palavra domus (casa) e imotique (automático) (ADAMI,[200?]).

Domótica é a automação e o controle aplicados à residência. Esta automatização e controle se realizam mediante o uso de equipamentos que dispõem de capacidade para se comunicar interativamente entre eles e com capacidade de seguir as instruções de um programa previamente estabelecido pelo usuário da residência e com possibilidade de alterações conforme interesses. Em consequência, a domótica, permite maior qualidade de vida, reduz o trabalho doméstico, aumenta o bem-estar e a segurança, racionaliza o consumo de energia e, além disso, sua evolução permite oferecer continuamente novas aplicações. (MURATORI E DAL BÓ, p.70).

A partir dessas definições percebe-se que o fator que define uma instalação de automação residencial é a interação dos sistemas tecnológicos aliado a capacidade de executar funções e comandos mediante instruções programáveis. Essa interação deve ser capaz de abranger todos os sistemas tecnológicos da residência, por exemplo:

- Instalação elétrica: Iluminação, gestão de energia e outros;
- Sistema de segurança: Monitoramento por CFTV, alarme de incêndio, alarme de invasão e outros;
- Sistemas de multimídia: Áudio e vídeo;
- Sistemas de comunicação: Rede doméstica, televisores e outros;
- Utilidade: Climatização, bombas, irrigação e outros.

Na figura 1 observamos um esquema de como seria uma casa inteligente com tudo interligado.

Figura 1- Casa Inteligente



Fonte: CASA EFICIENTE (2016)

Esse conceito de casa inteligente, como mostrada na figura 1, fica cada vez mais comum devido aos avanços das tecnologias, disseminando mais essa ideia levando conforto e qualidade de vida aos seus usuários, hoje pode se controlar uma casa com a utilização de um smartfone e sistemas integrados.

2.2 MICROCONTROLADORES

Um microcontrolador é, em última análise, um computador em um único chip e esse chip contém um processador (Unidade Lógica Aritmética-ULA), memória, periféricos de entrada e saída, temporizadores, dispositivos de comunicação serial, dentre outros. (PENIDO,2013, p.15).

Segundo Penido (2013, p.15) os microcontroladores surgiram naturalmente com a evolução dos circuitos digitais devido ao aumento da complexidade dos mesmos. Estes dispositivos revolucionaram a eletrônica pela capacidade de processamento em um tamanho reduzido e pelas facilidades que eles dão para a realização de vários projetos de grande aceite na sociedade.

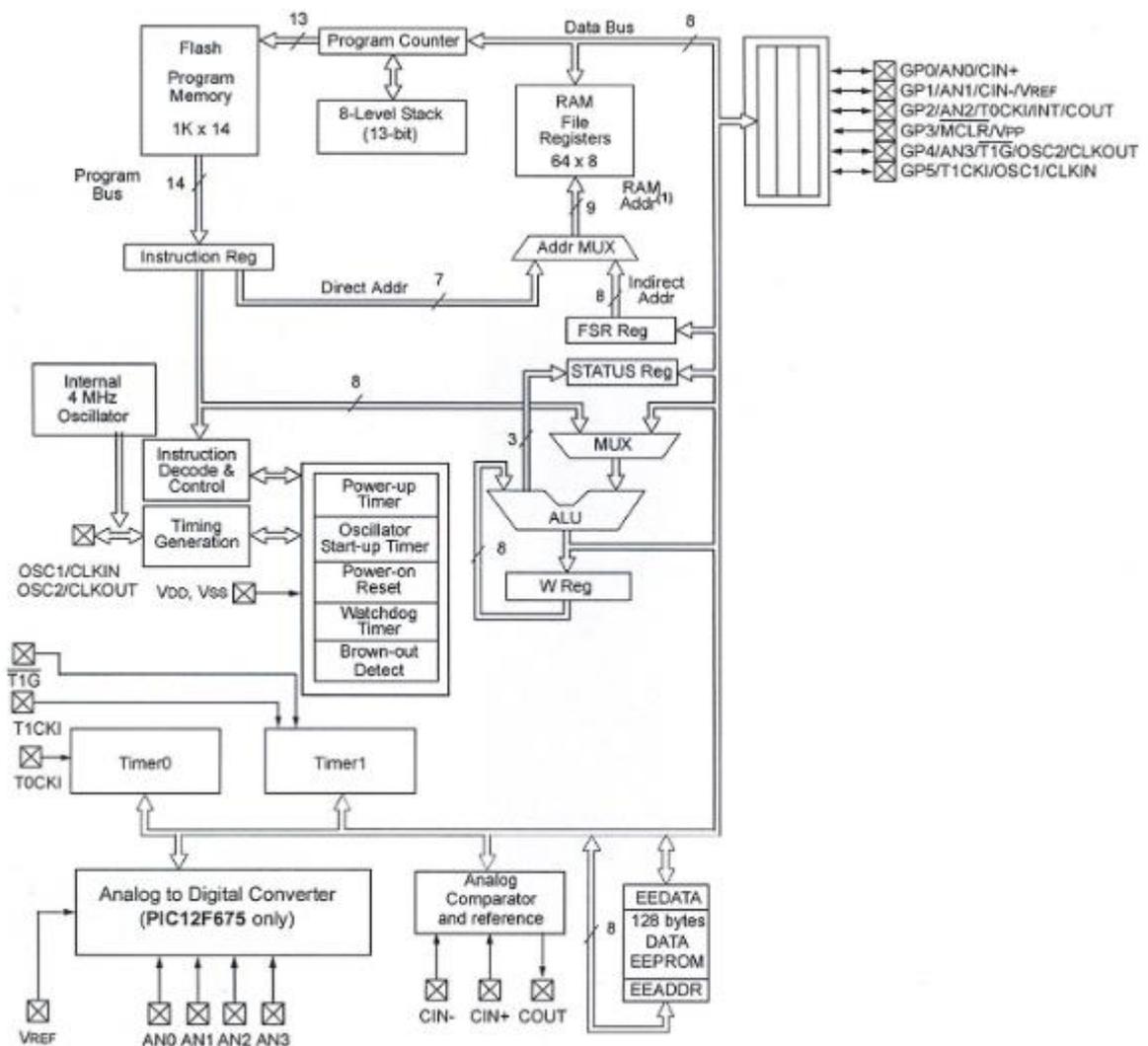
Segundo Penido (2013, p.16) um microcontrolador possui internamente os seguintes dispositivos:

- Uma CPU (Central Processor Unit ou Unidade de Processamento central), cuja a finalidade é interpretar as instruções de programa;
- Uma memória PROM (Programmable Read Only Memory ou Memória Programável Somente de Leitura) na qual são gravadas as instruções do programa;

- Uma memória RAM (Random Access Memory ou Memória de Acesso Aleatório) utilizada para memorizar as variáveis utilizadas pelo programa;
- Um conjunto de linhas de I/O para controlar dispositivos externos ou receber impulsos de sensores, interruptores, etc.
- Um conjunto de dispositivos auxiliares ao funcionamento, ou seja, gerador de clock, contadores, UASART para comunicação, etc.

A figura 2 apresenta o diagrama de blocos de um microcontrolador, mostrando os principais elementos básicos descritos anteriormente.

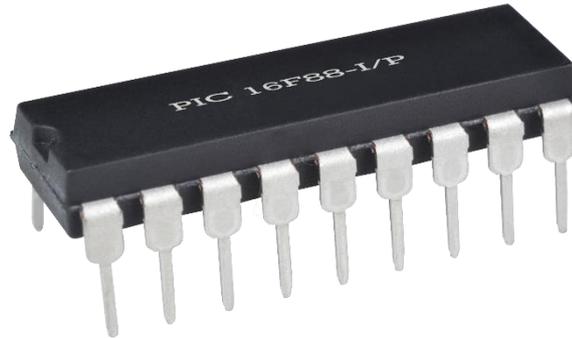
Figura 2- Diagrama de Blocos de um microcontrolador



Fonte: Penido e Trindade (2013, p.17)

A figura 3 ilustra o microcontrolador PIC 16F88-I/P e seus respectivos pinos.

Figura 3- Microcontrolador PIC16F88-I/P



Fonte: COMPIC (2017)

2.2.1 ESP8266

Segundo Curvello (2015) hoje em dia o lema é conectividade e mobilidade. Estamos vendo coisas cada vez mais móveis e conectadas, principalmente no que tange à internet. E dentre os inúmeros módulos que surgiram recentemente para explorar a onda da Internet das Coisas (IoT), o que mais se destaca é o ESP8266, da empresa Espressif.

O ESP8266 pode receber entrada de uma variedade de sensores e pode atuar como controlador de iluminação, motores e outros atuadores; sua programação é feita em linguagem *processing* (linguagem do Arduino) e LUA (Linguagem desenvolvida para o ESP8266), mas no presente trabalho será usado *processing* uma próxima a linguagem C.

Algumas características segundo Curvello (2015) do ESP8266:

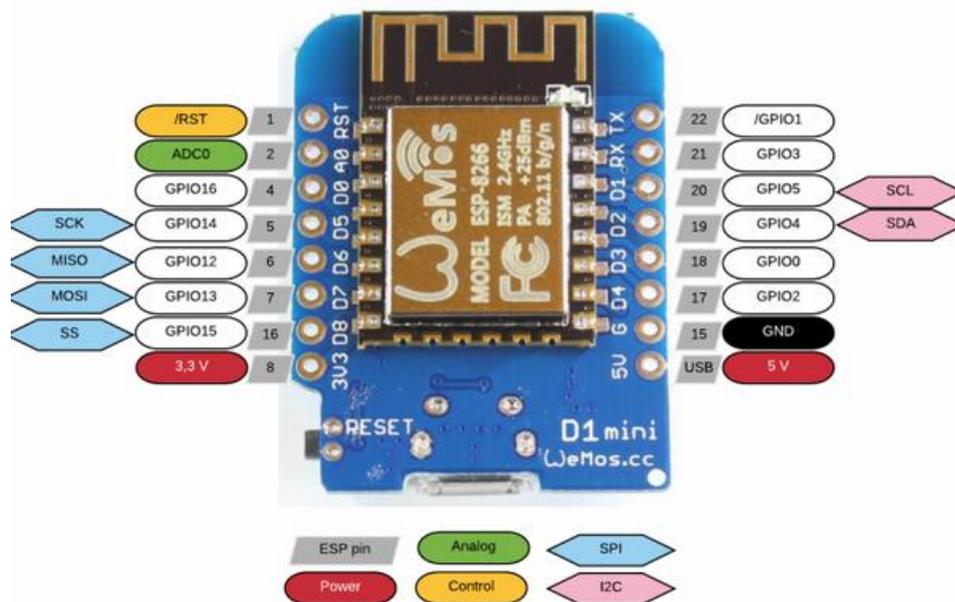
- É um System-On-Chip com Wi-Fi embutido;
- Tem conectores GPIO, barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura;
- CPU que opera em 80MHz, com possibilidade de operar em 160MHz;
- Arquitetura RISC de 32 bits;
- 32KBytes de RAM para instruções;
- 96KBytes de RAM para dados;
- 64KBytes de ROM para boot;
- Possui uma memória Flash SPI Winbond W25Q40BVNIG de 512KBytes;
- O núcleo é baseado no IP Diamand Standard LX3 da Tensilica;

- Fabricado pela Espressif;
- Existem módulos de diferentes tamanhos e fabricantes.

Os módulos ESP8266 são fornecidos numa ampla variedade de modelos, com diferenças perceptíveis principalmente no que tange à quantidade de IOs disponíveis para acesso externo, e no tamanho do módulo. Mas a placa que usaremos será o Wemos D1 Mini.

A placa do ESP8266 Wemos D1 Mini é um pequeno circuito microcontrolador, onde são colocados todos os componentes necessários para que funcione e se comunique com o computador, como mostra a figura 4:

Figura 4- Wemos D1 Mini



Fonte: Escapequotes (2016)

Conforme a figura 4 a placa do Wemos D1 Mini possui vários conectores que servem para interface com o mundo externo, a seguir será mostrada uma breve explicação de como pode ser utilizado cada pino.

O software para a programação do ESP8266 Wemos D1 Mini é a IDE do Arduino (Integrated Development Environment ou Ambiente Integrado de Desenvolvimento) que permite a criação de sketches (programas) para a placa Wemos D1 Mini. A linguagem é programada de acordo com a linguagem processing.

2.2.2 ARDUINO

O Arduino é considerado uma plataforma de código aberto (open-source) de hardware e software para prototipagem eletrônica, essa plataforma é destinada a designers, técnicos, engenheiros e pessoas interessadas em criar projetos ou ambientes interativos segundo a apostila de introdução à Arduino da FBS ELETRÔNICA.

Arduino pode receber entrada de uma variedade de sensores e pode atuar como controlador de iluminação, motores e outros atuadores, sua programação é feita em uma linguagem própria bem próxima da linguagem C diferenciando apenas em algumas coisas na escrita. O Arduino é diferente de outras plataformas presentes no mercado devido aos seguintes fatores:

- É um ambiente multiplataforma, podendo executar nos sistemas operacionais existentes;
- Tem por base um ambiente de fácil utilização em processing;
- Hardware de baixo custo;
- Grande comunidade ativa de usuários, facilitando a disseminação dos projetos.
- Ambiente educacional, ideal para iniciantes que desejam aprender rápido no próprio site Arduino.

A placa do Arduino é um pequeno circuito microcontrolador, onde é colocado todos os componentes necessários para que funcione e se comunique com o computador, como mostra a figura 5.

Figura 5- Arduino UNO



Fonte: FBS ELETRÔNICA ([entre 2012 e 2014], p.8)

Conforme a figura 5 a placa do Arduino uno possui vários conectores que servem para interface com o mundo externo, a seguir será mostrada uma breve explicação de como pode ser utilizado cada pino.

- 14 pinos de entrada e saída digital (pinos 0-13):
 - Esses pinos podem ser utilizados como entradas ou saídas digitais de acordo com a necessidade do projeto.
- 6 pinos de entradas analógicas (pinos A0-A5):
 - Esses pinos são dedicados a receber valores analógicos, por exemplo, a tensão de um sensor.
- 6 pinos de saída analógicas (pinos 3, 5, 6, 9, 10 e 11):
 - São pinos que podem ser programados para serem utilizados como saídas analógicas, utilizando modulação PWM.

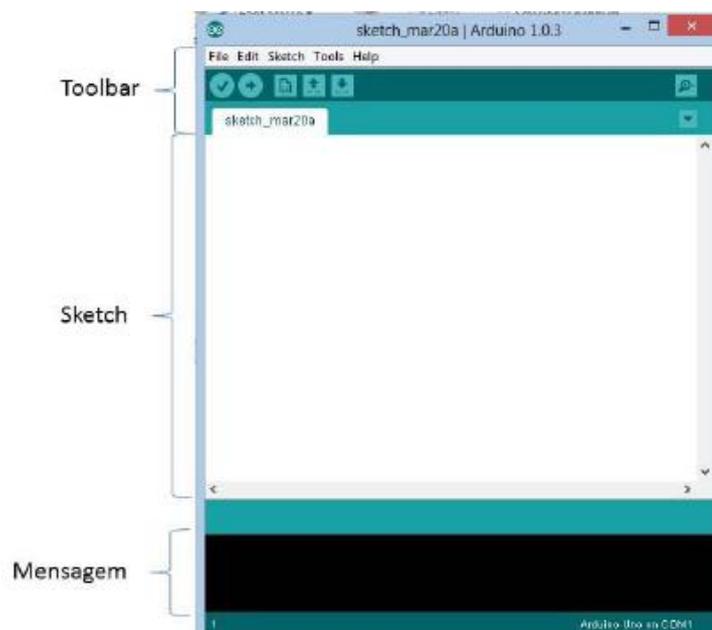
O software para a programação do Arduino é uma IDE (Integrated Development Environment ou Ambiente Integrado de Desenvolvimento), figura 6, que permite a criação de sketches (programas) para a placa Arduino. A linguagem é programa de acordo com a linguagem processing e quando pressionado o botão de upload da IDE, o código é traduzido para a linguagem C e é transmitida para o compilador avr-cc.

O ciclo de programação do Arduino pode ser dividido da seguinte forma:

- i. Conexão da placa a uma porta USB do computador;
- ii. Desenvolvimento de um sketch com comando para a placa;
- iii. Upload do sketch para a placa, utilizando a comunicação USB;
- iv. Aguardar a reinicialização, após ocorrerá a execução do sketch criado.

Na figura 11 observa-se a divisão da IDE do Arduino.

Figura 6- IDE Arduino



Fonte: FBS ELETRÔNICA ([entre 2012 e 2014], p.11)

2.3 INSTRUMENTAÇÃO INDUSTRIAL

Segundo Abelha (2014) a instrumentação industrial é a ciência que estuda, desenvolve e aplica instrumentos para medir, transmitir, indicar, registrar e controlar variáveis de processos industriais. A conceituação de Instrumentação Industrial é bem simples, mas vamos simplificar um pouco mais: A Instrumentação Industrial é um ramo da engenharia responsável por monitorar e controlar processos produtivos por meio de instrumentos.

2.3.1 SENSORES

Sensores são instrumentos industriais utilizados para aferir medidas, dependendo da característica do sensor ele pode ser capaz de medir temperatura, vazão, outros. Exemplo temos o sensor ultrassônico representado na figura 7.

Segundo Patsko (2016) literalmente, podemos definir a palavra sensor como “aquilo que sente”. Na eletrônica, um sensor é conhecido como qualquer componente ou circuito eletrônico que permita a análise de uma determinada condição do ambiente, podendo ela ser algo simples como temperatura ou luminosidade; uma medida um pouco mais complexa como a rotação de um motor ou a distância de um carro até algum obstáculo próximo ou até mesmo eventos distantes do nosso cotidiano, como a detecção de partículas subatômicas e radiações cósmicas.

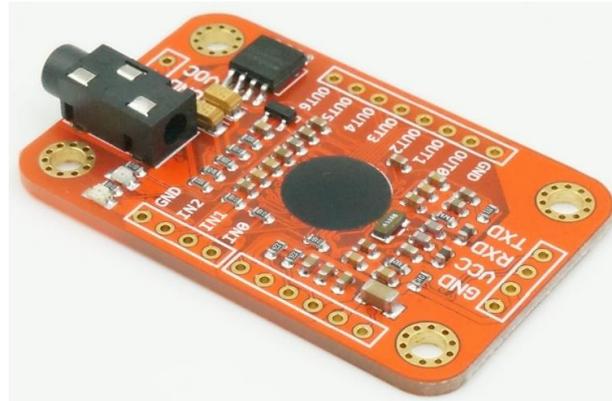
Figura 7- Sensor Ultrassônico



Fonte: FILIPFLOP (2014)

Módulo de reconhecimento de voz que será utilizado no trabalho será *voice recognizer module v3*, módulo compatível com a IDE do Arduino, o mesmo será usado para a gravação dos comandos e o reconhecimento das ordens emitidas pelo usuário, abaixo temos a figura 8 do módulo:

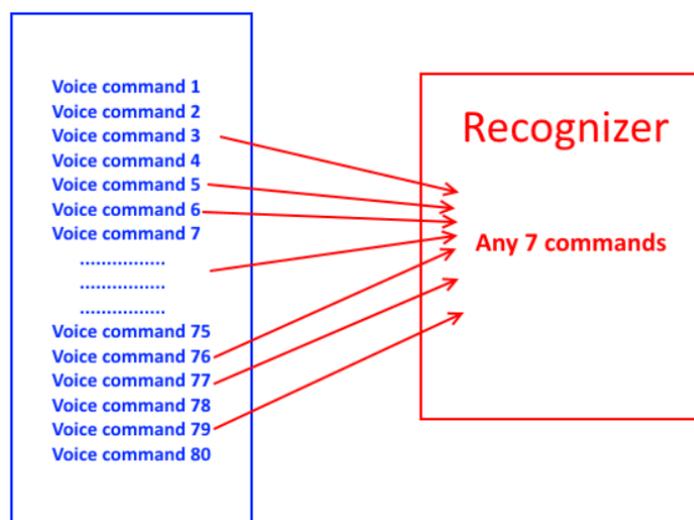
Figura 8- Módulo de Voz V3



Fonte: Elechouse (2014, p.1)

O módulo de reconhecimento de voz tem a capacidade de armazenar 80 comandos, porém apenas 7 podem ser executados ao mesmo tempo, o que nos leva a dividir esses 80 comandos em grupos de 7. Como ilustrado na figura 9.

Figura 9- Divisão dos Grupos



Fonte: Elechouse (2014, p.4)

Alguns parâmetros do módulo de voz:

- Tensão: 4.5-5v;

- Corrente:< 40mA;
- Suporte de 80 comandos de voz;
- 99% Precisão de reconhecimento de voz; e
- Biblioteca para Arduino.

2.4 INTERNET DAS COISAS (IOT)

A internet das coisas, ou IOT (*Internet Of Things*), é a relação entre dispositivos que compartilham informação, enviando ou coletando, na internet.

Segundo Hewlett Packard Enterprise, O termo "Internet das Coisas", a "IoT," descreve o grande e cada vez maior conjunto de dispositivos digitais — agora na casa dos bilhões — que operam entre redes de escala potencialmente global. Ao contrário da Internet normal (das pessoas), a IoT é composta apenas por sensores e outros dispositivos inteligentes. Entre seus usos estão a captação de dados operacionais de sensores remotos em plataformas de petróleo, a coleta de dados climáticos e o controle de termostatos inteligentes.

Evans (2011) diz que, a IoT (Internet of Things, Internet das coisas), algumas vezes referida como a Internet dos objetos, mudará tudo, inclusive nós mesmos. Isso pode parecer uma declaração arrojada, mas considere o impacto que a Internet já teve na educação, na comunicação, nos negócios, na ciência, no governo e na humanidade. Claramente, a Internet é uma das criações mais importantes e poderosas de toda a história humana.

Protocolo MQTT (Message Queue Telemetry Transport ou Transporte de Telemetria da Fila de Mensagens), é um protocolo de comunicação para mensagens de sensores e dispositivos que são otimizados para redes TCP/IP.

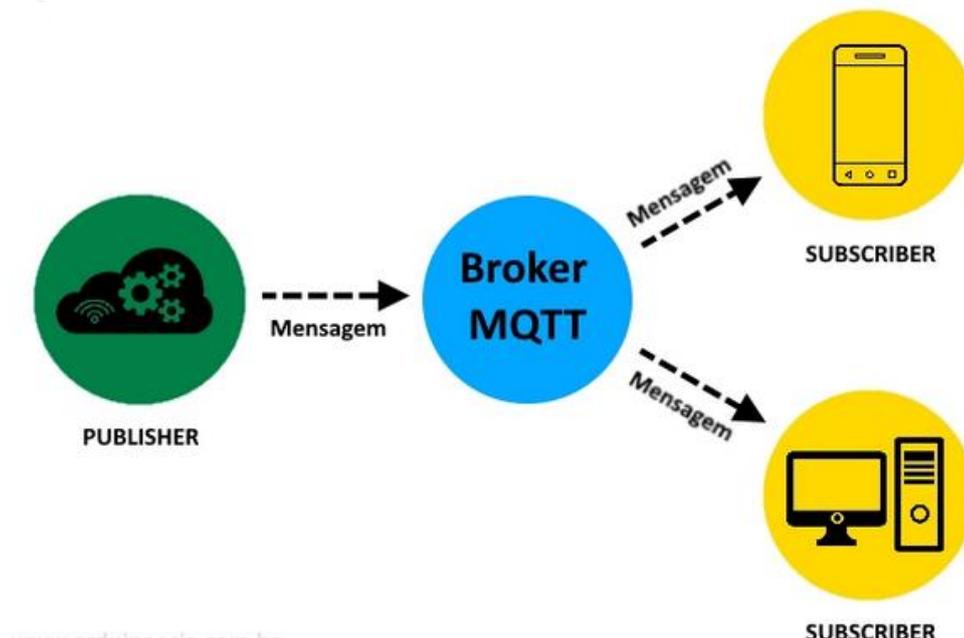
As características do MQTT segundo Bratinu (2018), estão listadas abaixo:

- Agnóstico em termo de dados: O MQTT pode transportar todos os tipos de dados, desde dados de sensor até imagens ou atualizações via rede sem fio.
- Leve e eficiente em termos de largura de banda: o menor quadro tem apenas 2 bytes.
- Oferece QoS: três níveis de Qualidade de Serviços(QoS).
- Funciona sobre a pilha TCP/IP.
- Desenvolvimento simples: existem clientes para todos os sistemas operacionais e linguagens de programação.

- Broker central: pode conectar diferentes dispositivos sem se preocupar com a compatibilidade.
- Percepção de sessão: oferece uma abordagem baseada em identidade para assinaturas.
- Tópicos de assinaturas flexíveis.

Na figura 10, está exposto a estrutura do protocolo MQTT, com seus clientes e suas respectivas funções.

Figura10 - Estrutura MQTT



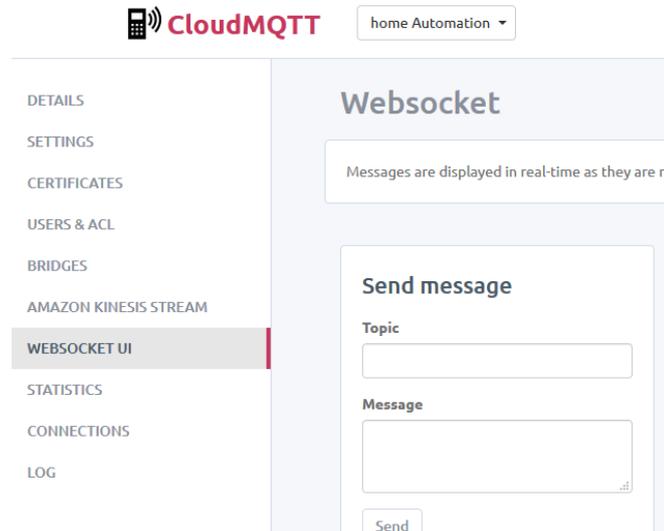
Fonte: Arduinoecia(2019)

Terminologia básica do protocolo MQTT:

- Broker: responsável por receber e enviar as mensagens;
- Cliente: é o responsável por publicar (Publish) ou receber (subscriber) uma mensagem;
- Publicar: ação de enviar uma mensagem;
- Assinar: ação de mostrar interesse por alguma mensagem.

Como servidor, está sendo utilizado o CloudMqtt, um servidor gratuito, ilustrado na figura 11:

Figura 11- Servidor ClodMQTT

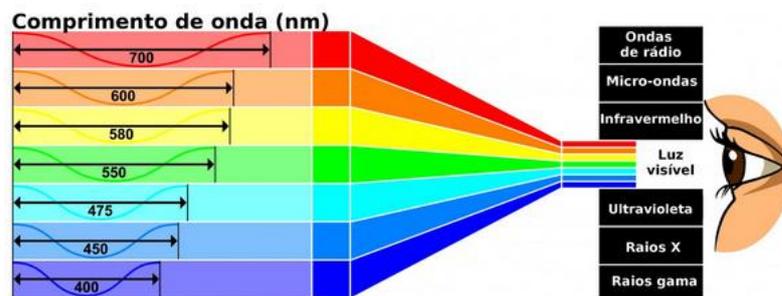


Fonte: Autor

2.5 IR- INFRAVERMELHO (INFRARED)

Partindo da base de conhecimento que temos a respeito de ondas eletromagnéticas, conseguimos separá-las em dois tipos, ondas visíveis e ondas não visíveis, o infravermelho se encaixa no tipo de ondas não visíveis, como mostra a figura 12 abaixo.

Figura 12- Espectro Eletromagnético



O infravermelho não é visível ao olho humano por ser uma onda de comprimento menor que a luz vermelha

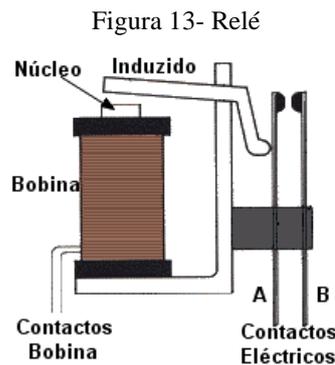
Fonte: Brasil Escola (2019)

Alguns dispositivos utilizados pelo homem funcionam por meio da emissão de raios infravermelhos, por exemplo, o controle remoto de um televisor possui um LED que emite essa

radiação e o televisor possui um receptor que detecta e executa o comando enviado para o mesmo.

2.6 RELÉ

Um relé, figura 13, é basicamente um interruptor mecânico que abre ou fecha circuitos de cargas elevadas, é operado eletricamente por aplicação de uma tensão baixa na bobina de comando.



Fonte: Eletrônica PT (2019)

O módulo relé de 5v é um dispositivo comutador eletromecânico, ilustrado na figura 36, ele funciona da seguinte maneira, quando a corrente circula pela bobina é criado um campo magnético que atrai os contatos, e quando atraídos fecham o circuito normalmente aberto e abrem o normalmente fechado, o utilizado em questão é acionado em nível lógico baixo.

Ele é utilizado pois os kits de prototipagem trabalham com correntes e tensões baixas que impossibilitam o acionamento de cargas que demandam mais corrente ou mesmo tensão.

2.7 TRABALHOS RELACIONADOS

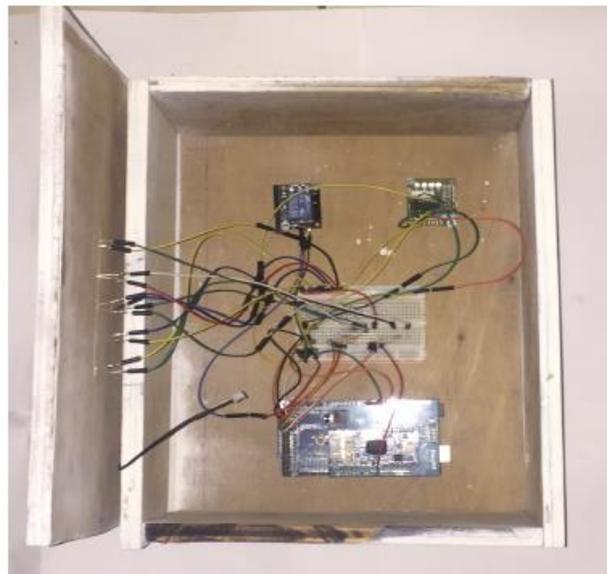
No capítulo será apresentado alguns trabalhos científicos que se assemelham com o tema proposto nessa monografia, resumindo cada um deles e citando suas principais semelhanças.

2.7.1 SISTEMA DE CONTROLE DE DISPOSITIVOS PARA AUXILIAR PORTADORES DE DEFICIÊNCIA FÍSICA E/OU VISUAL

Na monografia de Cardoso (2017), o autor propõe que o desenvolvimento do projeto é focado na Tecnologia Assistiva, área inovadora que utiliza a tecnologia atual para ajudar pessoas com diversas deficiências, sendo empregado em vários trabalhos por interagir com a sociedade. O projeto visa auxiliar os portadores de deficiência física e/ou visual nas atividades diárias ligadas aos dispositivos da residência, modificando o meio de controle físico por voz, deste modo, facilita a vida dos usuários, reduzindo a diferença social existente entre os portadores de deficiência física e/ou visual e as pessoas sem deficiência neste quesito.

Este é o projeto base para o desenvolvimento do presente projeto apresentado, procurou-se atualizar os métodos usados pelo autor para obter resultados melhores. A figura 14 mostra o protótipo desenvolvido.

Figura 14- Protótipo pronto aberto



Fonte: Cardoso (2017, p.53)

2.7.2 AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM ARDUINO MEGA E ETHERNET SHIELD

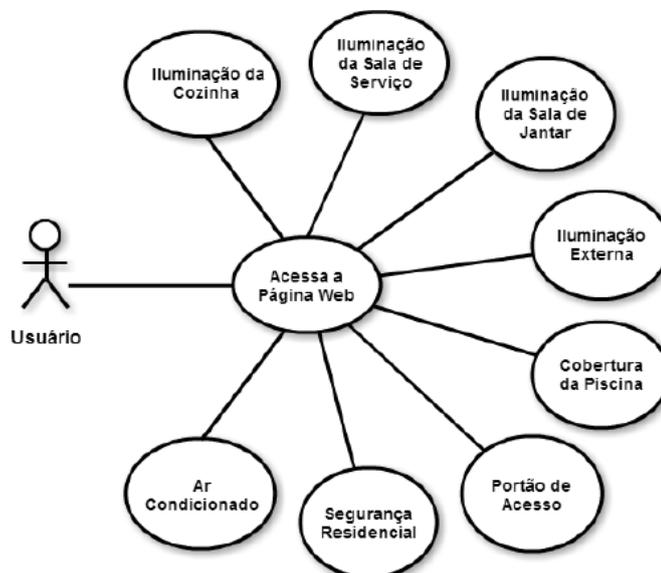
Neste artigo de Lima, Nobre e Alencar (2015), os autores propõem uma solução domótica (automação residencial) de baixo custo para gerir procedimentos via Web, com

utilização do Arduino Mega e Ethernet Shield. O intuito também é mostrar como está o mercado no Brasil, bem como os custos envolvidos para automatizar uma residência.

A domótica tem uma grande importância para a acessibilidade, uma vez que facilita a interação de pessoas que tenham algum tipo de deficiência física com os mecanismos do dia-a-dia, de forma que se ajuste às suas necessidades e se mostrando cada vez mais presente em nossas vidas.

Como percebido o artigo em questão se assemelha tentando realizar a automação residencial buscando um baixo custo. A figura 15 mostra o diagrama do caso de uso.

Figura 15- Diagrama de Caso de Uso



Fonte: Lima, Nobre e Alencar (2015, p.12)

2.7.3 SISTEMA DE COMANDOS E IDENTIFICAÇÃO POR VOZ

Na monografia de Landim (2017), o autor propõe que o projeto visa a implementação de um sistema capaz de compreender comandos de voz e executar as tarefas correspondentes. O processo de reconhecimento de voz foi implementado com base em um módulo que utiliza o conversor de fala para texto chamado Wit.ai. Para a conversão de texto-para-fala, foi utilizado o sintetizador de voz Flite. O sistema foi desenvolvido em Raspberry Pi e permite a execução das seguintes tarefas: acionamento de lâmpadas, gravação de áudios, captura de imagens, escrita e leitura de textos, buscas na internet e identificação de voz. Todos os comandos foram desenvolvidos na linguagem de programação Python. Uma página na web foi desenvolvida para permitir o acesso aos dados armazenados pelo sistema. A identificação da voz do usuário

permite que acessos indevidos ao sistema sejam reportados. Nota-se que o número de amostras treinadas resulta em melhorias na identificação da voz. Os mecanismos de reconhecimento de fala são limitados, o Wit.ai necessita de conexão à internet e seu processamento é inferior quando comparado a outros serviços pagos.

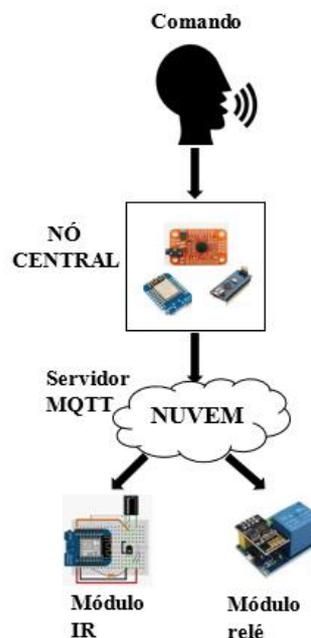
Apesar do sistema de reconhecimento seja diferente a intenção é a mesma, através do reconhecimento da fala executar comandos no ambiente.

3. METODOLOGIA

No tocante ao desenvolvimento deste trabalho foi criado um protótipo que faça reconhecimento do comando a partir da fala do usuário e as execute; e ainda que esse protótipo indique que recebeu o comando através da indicação de led's e um buzzer. No mais as informações serão tratadas em forma de binário e lidas pelo Wemos D1 mini, que por sua vez irá enviar os comandos para o servidor MQTT (Message Queue Telemetry Transport ou Transporte de Telemetria da Fila de Mensagens), que envia para os outros módulos o comando que foi enviado pelo usuário.

O sistema irá receber o comando por voz, o modulo de voz por sua vez enviará o comando entendido para o Arduino nano, que irá enviar a mensagem para o ESP8266 Wemos d1 mini que disponibilizará o comando no servidor MQTT, onde um dos outros módulos irá coletar e executar o comando proposto pelo usuário. Conforme o esquema ilustrado na figura 16.

Figura 16- Diagrama da Arquitetura do Projeto



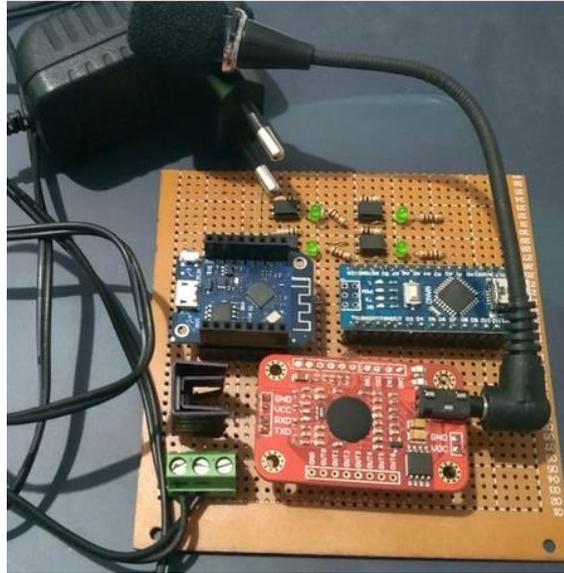
Fonte: Autor

3.1 MÓDULO DO COMANDO CENTRAL

Na figura 17 está disposto o módulo central de reconhecimento de voz, que foi feito para conseguir entender os comandos de voz e executá-los. Tal sistema contém um módulo de

reconhecimento de voz, um Arduino Nano, um ESP8266 Wemos d1 mini e outros componentes eletrônicos (resistores, capacitor, optoacopladores, buzzer e regulador de tensão).

Figura 17- Módulo Central de reconhecimento de voz



Fonte: Autor

Na figura 17 observa-se o módulo principal, responsável por captar os comandos através do módulo de reconhecimento de voz, após reconhecido o sinal é tratado no Arduino nano e enviado para o Wemos responsável por enviar os comandos para a nuvem no servidor CloudMQTT, que enviará aos outros módulos os comandos ordenados. O Módulo é alimentado por uma fonte de tensão de 12 V.

3.1.1 MÓDULO DE VOZ

Como antes já mencionado na revisão teórica do trabalho, o modulo de reconhecimento de voz V3, possui algumas características que devem ser levadas em conta, a primeira seria a sua capacidade de executar apenas 7 comandos ao mesmo tempo, sendo necessária a implementação de lógicas para fazer a utilização dos outros comandos gravados, segunda ele precisa ser alimentado e comunicado com 5v.

Sua função é captar os comandos enviados pelo usuário através da fala, captada pelo microfone de eletreto que acompanha o módulo, fazer a comparação entre os comandos pré-gravados e enviar a informação do que foi entendido para os próximos componentes.

A figura 18 mostra a tela de gravação dos comandos para que o módulo possa fazer as comparações.

Figura 18- Tela de gravação das assinaturas



Fonte: Eletrogate (2014, p.9)

Para gravar os comandos, precisa utilizar o comando sigtrain, local de armazenamento do comando e o nome da assinatura, o comando só é treinado quando aparece em seu fim trained, a figura 19 mostra os comandos que foram treinados e suas respectivas funções.

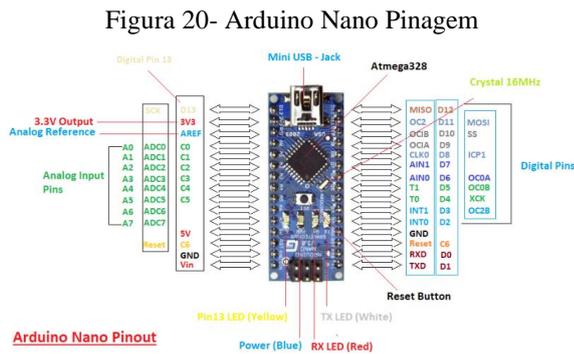
Figura 19- Comandos treinados e suas funções

Comandos treinados	Funções
mudar	muda entres os grupos
central	ligar e desligar a central
quente	aumentar a temperatura do ar condicionado
frio	diminuir a temperatura do ar condicionado
clima	função mode do ar condicionado
luz	liga e desliga a iluminação
tv	liga e desliga a televisão
alto	aumenta o volume da televisão
baixo	diminui o volume da televisão
mais	aumenta o canal da Televisão
menos	diminui o canal da Televisão
record	muda para o canal da record na televisão

Fonte: Autor

3.1.2 ARDUINO NANO

O Arduino Nano com próprio nome já expõe, é uma versão menor do Arduino e possui algumas características que o leva ele a ser muito usado, tais como: ser compacto, possuir uma entrada mini usb tipo a e ter muitas portas digitais e analógicas, como ilustrado na figura 20.



Fonte: theengineeringprojects (2018)

Sua função é receber do módulo de voz o comando que foi enviado e assim transmitir a informação via código binário para o próximo componente (Wemos), conforme a figura 21, que ao identificar o comando **mudar**, o sistema faz uma alternância entre o grupo 1 (grupo da lâmpada e do ar condicionado) e o grupo 2 (grupo da televisão).

Figura 21- Trecho do programa Nano: mudar de grupo

```
switch (group) {
  case 0:
    myVR.clear();
    record[0] = switchgroup;
    record[1] = ar_liga_desliga;
    record[2] = ar_temperatura_maior;
    record[3] = ar_temperatura_menor;
    record[4] = ar_mode;
    record[5] = lamp_liga_desliga;
    record[6] = tv_liga_desliga;
    if (myVR.load(record, 7) >= 0) {
      printRecord(record, 7);
      Serial.println(F("carregado o grupo do
ar_e_lamp"));
    }
    break;
  case 1:
    myVR.clear();
    record[0] = switchgroup;

    record[1] = tv_liga_desliga;
    record[2] = tv_volume_mais;
    record[3] = tv_volume_menos ;
    record[4] = tv_canal_mais;
    record[5] = tv_canal_menos;
    record[6] = tv_canal_4;
    if (myVR.load(record, 7) >= 0) {
      Serial.println(F("carregado o grupo da
tevelisão"));
    }
    break;
}
```

Fonte: Autor

Caso o comando seja reconhecido ocorre uma troca do grupo1 para o grupo 2 e uma gravação dos comandos que estão pré-carregados no grupo da televisão.

Na figura 22, fica disponibilizado os comandos em forma de binário que depois são traduzidos para decimal e utilizado pelo próximo componente (Wemos).

Figura 21- Tabela com Binário e o Decimal enviado pelo Nano

Comandos treinados	LED	LED1	LED2	LED3	BINÁRIO	DECIMAL
mudar	1	1	1	1	1111	15
central	1	0	0	0	1000	8
quente	0	1	0	0	0100	4
frio	0	0	1	0	0010	2
clima	0	0	0	1	0001	1
luz	1	1	0	0	1100	12
tv	1	0	0	1	1001	9
alto	1	0	1	0	1010	10
baixo	0	1	0	1	0101	5
mais	0	1	1	0	0110	6
menos	0	0	1	1	0011	3
record	1	1	0	1	1101	13

Fonte: Autor

3.1.3 ESP8266 Wemos D1 Mini

O Wemos que já foi citado na revisão teórica do projeto, figura 4, tem como principal função decodificar o binário que chega e colocar o mesmo em decimal, e assim executar o envio de informações para a Nuvem. Conforme a figura 23, o Wemos envia uma mensagem em um determinado tópico, e fica em nosso sistema como publicador (*publisher*).

Figura 23- Trecho do programa Wemos: Publicar liga_desliga_lamp

```
//-----Liga e deliga a lampada-----
if (binario == 12 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/lamp", "liga_desliga_lamp");
    cont = 0;
    binario = 0;
    delay(600);
  }
}
//-----
```

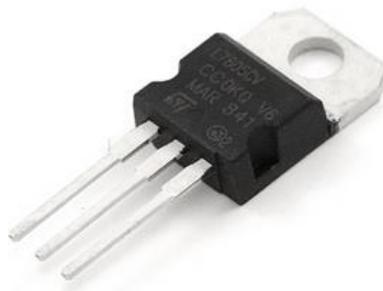
Fonte: Autor

Ainda no trecho de código da figura 23 se o binário enviado pelo Arduino Nano for igual a 12 em decimal, o Wemos irá publicar no tópico “home/lamp”, a seguinte mensagem “liga_desliga_lamp”.

3.1.4 REGULADOR DE TENSÃO, OPTOACOPLOADORES, RESISTORES, BUZZER, CAPACITOR e LED's

O regulador de tensão, figura 24, tem um papel importante no circuito, pois o mesmo será responsável por deixar a tensão das fontes ou da bateria próximo de 5V, sendo a alimentação ideal para os kit's de prototipagem eletrônica, evitando assim qualquer tipo de sobretensão e mal funcionamento dos kit's.

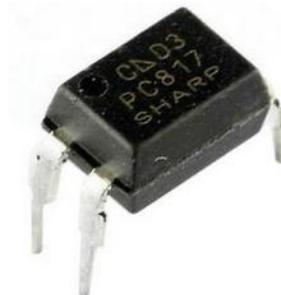
Figura 24- Regulador de Tensão 5V



Fonte: AutoCore Robótica (2019)

Os optoacopladores, figura 25, são de maneira geral utilizados para fazer a comunicação ou combinação de elementos que possuem níveis de sinais diferentes, no caso específico o Arduino nano e o Wemos D1 Mini, o Arduino possui saída de 5v e o Wemos entrada de 3,3v, e para que eles se comuniquem foi utilizado optoacopladores para fazer essa intermediação.

Figura 25- Optoacoplador PC817



Fonte: Baú da Eletrônica (2019)

Os resistores estão sendo usados como divisores de tensão para os LED's e optoacopladores, o circuito possui resistores de 1Kohm e 10Kohm.

O Buzzer como identificador de comando, toda vez que um comando é reconhecido ele é acionado, mostrando que o comando foi entendido, o buzzer está representado pela figura 26.

Figura 26- Buzzer



Fonte: Baú da Eletrônica (2019)

Os capacitores estão sendo utilizados como filtro para as tensões DC's, na figura 27 temos um capacitor eletrolítico.

Figura 27- Capacitor Eletrolitico



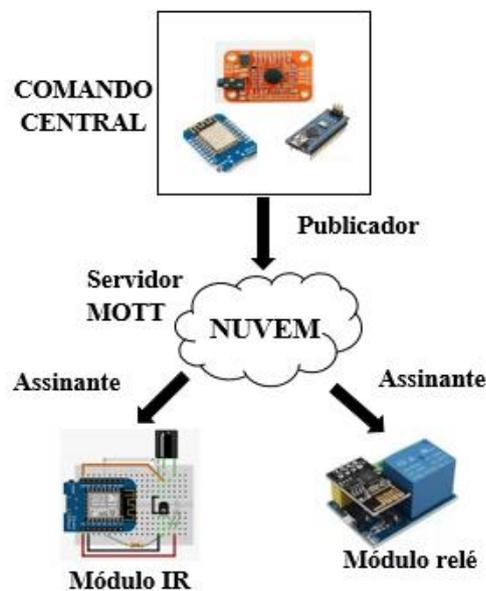
Fonte: Mercado Livre (2019)

Os LED's por sua vez são responsáveis por mostrar que houve comunicação, sendo ela entre o Arduino e o Wemos, conexão com a rede, ou indicando que o comando de infravermelho está sendo enviado ou até mesmo enviando comando de infravermelho.

3.2 NUVEM

A nuvem é o intermediário entre o comando central (publicador) e os demais módulos (assinantes), tendo a função de armazenar as publicações do módulo central e deixar disponível para os assinantes. O CloudMQTT de forma gratuita, cede ao usuário um servidor MQTT para armazenar os dados bastando apenas ser configurado. Na figura 28 temos o diagrama de envio MQTT, que revela o publicador e o assinante.

Figura 28- Diagrama da estrutura MQTT do projeto



Fonte: Autor

Para configurar o CloudMQTT para receber os comandos, foi criado alguns tópicos: home/lamp, home/ar e home/tv, como mostra a figura 29.

Figura 29- Tópicos do projeto

Type	Pattern	Read/Write	
topic	joaoPedro - home/ar	true/true	Delete
topic	joaoPedro - home/tv	true/true	Delete
topic	joaoPedro - home/lamp	true/true	Delete

Fonte: Autor

E ao chegar mensagem muda para a mensagem e o t3pico do qual foram destinadas, conforme a figura 30.

Figura 30- Mensagem no servidor

Received messages

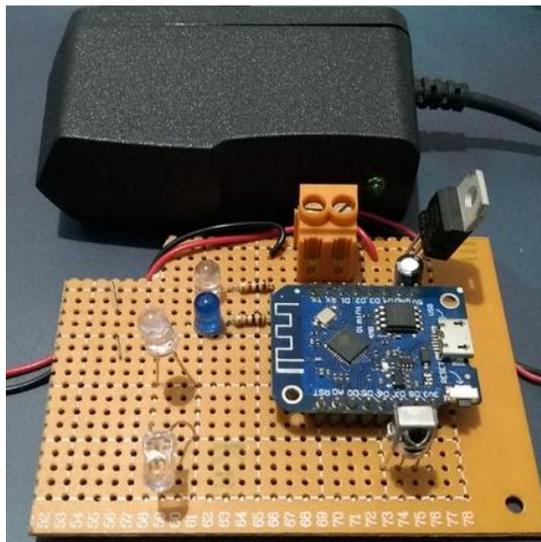
Topic	Message
home/lamp	liga_desliga_lamp

Fonte: Autor

3.3 M3DULO INFRAVERMELHO

Este m3dulo 3 respons3vel por receber as informa33es da nuvem e enviar o c3digo correspondente a mensagem recebida. A figura 31 mostra como este est3 montado.

Figura 31- M3dulo emissor e receptor de Infravermelho com Wemos D1 Mini



Fonte: Autor

Na figura 32 observa-se o m3dulo de comunica33o de infravermelho que ao receber os comandos do servidor CloudMQTT, envia as strings para executar o comando solicitado.

Figura 32- Trecho do Programa Wemos infravermelho: recebe liga_desliga_ar

```
//-----identificação do comando ar condicionado-----
if (String(topic).equals("home/ar")) {
  // se o tópico no servidor CloudMQTT for home/ar
  //-----Liga e Desliga arcondicionado-----
  if ((char)payloadstr.equals("liga_desliga_ar")) {
    //-----se a mensagem do tópico for liga_desli_ar, é enviado a string para ligar o ar-----
    irsend.sendRaw(liga_ar, sizeof(liga_ar) / sizeof(liga_ar[0]), frequencia); // PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado: Liga / Desliga_AR");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
  }
}
```

Fonte: Autor

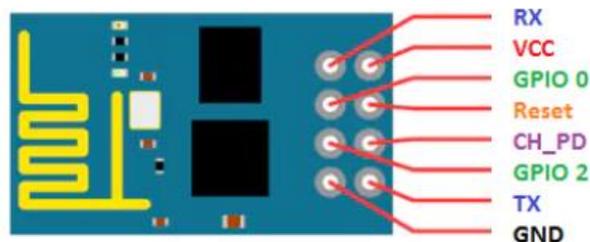
Ao enviar a string, é acionado um LED para confirma o envio da string.

Os componentes presentes neste módulo são semelhantes ao anterior, resistores, LED's e o componente que diferencia é o receptor de infravermelho, que foi colocado, caso queira adicionar mais strings ao programa. O módulo é alimentado por uma fonte de tensão de 12V.

3.4 MÓDULO RELÉ

O ESP-01 é o modulo Wi-fi de menor custo comparado a família, porém ele tem uma enorme desvantagem, sua quantidade de entradas e saídas são poucas apenas 4, sendo que duas são compartilhados para comunicação serial, restando apenas duas manipuláveis, ilustrado na figura 33.

Figura 33- ESP-01 Pinagem



Fonte: theengineeringprojects (2018)

O módulo relé ilustrado na figura 34 é responsável por pegar a mensagem que está no tópico home/lamp, e acionar o relé ligando então a lâmpada. O módulo é alimentado por uma fonte de tensão de 5v.

Figura34- Módulo relé com ESP-01



Fonte: Autor

Ao identificar o comando enviado no seu devido tópico e a sua informação correta ele aciona o relé e acende a iluminação em um controle ON-OFF, como ilustrado na figura 35.

Figura 35- Mensagem recebida pelo módulo relé

```
//-----identificação do comando da lampada---
if (String(topic).equals("home/lamp")) {
  // se o tópico for home/lamp
  //-----Liga e Desliga lampada-----
  if ((char)payloadstr.equals("liga_desliga_lamp")) {
    digitalWrite(0, digitalRead(0) ^ 1);
    Serial.println(digitalRead(0));
    Serial.println("Lamp_on_or_off");
  }
}
```

Fonte: Autor

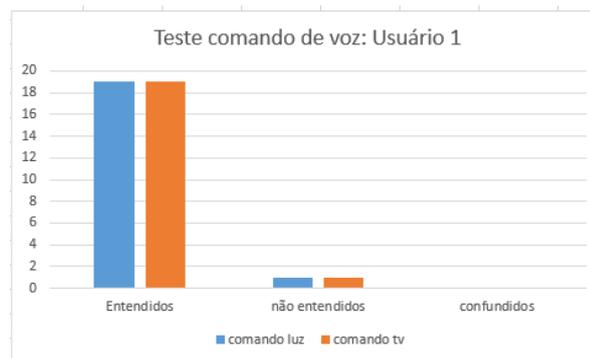
4. RESULTADOS OBTIDOS

4.1 TESTES DO PROTÓTIPO

Os testes realizados nos informarão como está a identificação dos comandos pré-gravados ao observar suas respostas após 20 amostras de dois comandos diferentes com vários usuários e por meio disso medir a precisão do módulo.

A figura 36 mostra o teste feito para confirmar a funcionalidade do protótipo. Foram escolhidos dois comandos para fazer o teste, o comando “luz” que liga e desliga a iluminação e o comando “tv” que liga e desliga a televisão. Cada um foi repetido 20 vezes por quatro usuários diferentes, porém apenas um deles gravou os comandos. As figuras 36 ,37 e 38 mostram os resultados obtidos.

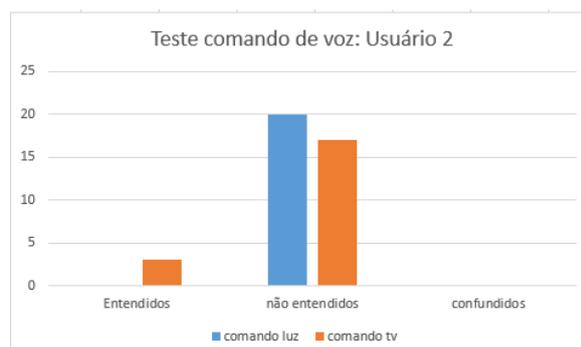
Figura 36- Teste do usuário 1



Fonte: Autor

Na figura 36 temos o resultado do usuário 1, do sexo masculino, que gravou os comandos, com uma assertiva de 19 comandos entendidos, 1 comando não entendido e 0 comando confundido, dos dois comandos em questão. Tendo uma taxa de acerto de 95%.

Figura 37- Teste do usuário 2

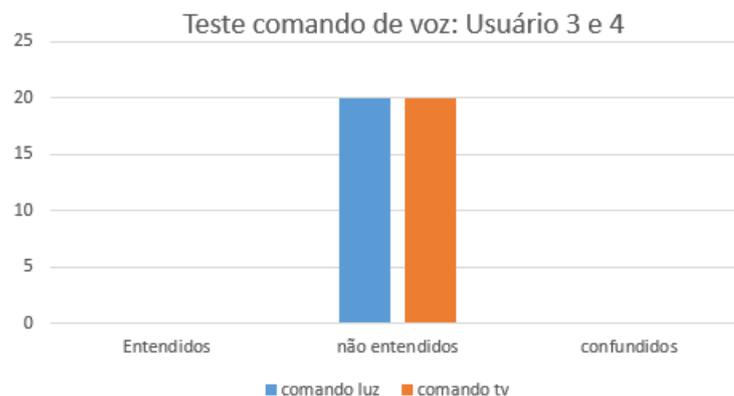


Fonte: Autor

Na figura 37 temos resultado do usuário 2, do sexo masculino, que não gravou os comandos, tendo uma assertiva de 3 comando entendidos, 17 comandos não entendidos e 0 comandos confundidos para o comando “tv”; e teve uma assertiva de 0 comando entendidos, 20 comandos não entendido e 0 comandos confundidos para o comando “luz”. Tendo uma taxa de acerto de 15% e 0% para comandos “tv” e “luz” respectivamente.

Os outros dois usuários são do sexo feminino e ambos não tiveram nenhum comando entendido, como mostra a figura 38.

Figura 38- Teste dos usuários 3 e 4



Fonte: Autor

Com a observação da figura 38 fica subentendido que a mudança de timbre interfere na identificação do comando, com isso a pessoa que for usar deve ser a mesma que grava os comandos.

4.2 CUSTO DO PROJETO

O projeto tem um custo aproximado equivalente aos componentes utilizados. Para diminuir os custos foram utilizados componentes que estavam em placas antigas. O projeto tenta ser o mais prático e fácil para que qualquer pessoa que possua o mínimo conhecimento de eletrônica possa reproduzir a partir da leitura desse trabalho, o cálculo foi feito com base no dólar do dia pesquisado R\$ 4,20, visto que sai mais barato importar os componentes, pois Manaus não dispõe de todos os componentes necessários para a realização do projeto, apenas

os mais complexos foram comprados fora, os menores foram encontrados em eletrônicas na cidade referida.

As figuras 39 e 40 mostram os custos dos componentes utilizados, suas quantidades e o gasto médio encontrado no mercado. A figura 39 utilizando o ESP8266 NodeMcu e a figura 30 com o ESP8266 Wemos D1 Mini, ficando a critério do usuário a escolha de qual usar, uma vez que os dois desempenham a mesma função.

Figura 39- Gastos usando ESP8266 NodeMcu

Quant.	Componente	Valor Unit.	Valor Total
1	MÓDULO DE VOZ	R\$ 71,67	R\$ 71,67
2	ESP8266 NODEMCU	R\$ 7,57	R\$ 15,14
1	MODULO RELÉ	R\$ 10,57	R\$ 10,57
1	ARDUINO NANO	R\$ 8,56	R\$ 8,56
3	REGULADOR DE TENSÃO	R\$ 1,50	R\$ 4,50
15	RESISTORES	R\$ 1,00	R\$ 15,00
3	BATERIAS 9V	R\$ 5,00	R\$ 15,00
9	LEDS	R\$ 1,50	R\$ 13,50
2	KIT INFRAVERMELHO	R\$ 5,00	R\$ 10,00
4	OPTOACOPLADORES	R\$ 2,50	R\$ 10,00
Sub-Total		R\$ 173,94	
Frete		R\$ 55,00	
TOTAL		R\$ 228,94	

Fonte: Autor

Figura 40- Gastos usando ESP8266 Wemos D1 Mini

Quant.	Componente	Valor Unit.	Valor Total
1	MÓDULO DE VOZ	R\$ 71,67	R\$ 71,67
2	ESP8266 WEMOS	R\$ 8,38	R\$ 16,76
1	MODULO RELÉ	R\$ 10,57	R\$ 10,57
1	ARDUINO NANO	R\$ 8,56	R\$ 8,56
3	REGULADOR DE TENSÃO	R\$ 1,50	R\$ 4,50
15	RESISTORES	R\$ 1,00	R\$ 15,00
3	BATERIAS 9V	R\$ 5,00	R\$ 15,00
9	LEDS	R\$ 1,50	R\$ 13,50
2	KIT INFRAVERMELHO	R\$ 5,00	R\$ 10,00
4	OPTOACOPLADORES	R\$ 2,50	R\$ 10,00
Sub-Total		R\$ 175,56	
Frete		R\$ 56,17	
TOTAL		R\$ 231,73	

Fonte: Autor

Com relação ao custo podemos comparar os preços dos dispositivos no mercado que entregam funcionalidades parecidas com a do protótipo desenvolvido, tais como:.

- Google home: R\$375,00-R\$430,00
- Alexa: R\$349,00

Observando os preços, foi possível desenvolver um protótipo com baixo custo, em português e compatível com as tecnologias existentes, visto que o dispositivo Google home tem comandos apenas em inglês.

Claro que comparar as tecnologias com o protótipo deixaria muito a desejar, porém pode-se controlar dispositivos por meio da tecnologia sem fio infravermelho, contudo o Google Home e a Alexa necessitam que os outros eletrônicos estejam conectados na rede e possuem apenas algumas compatibilidades, dando ainda mais crédito ao protótipo desenvolvido nesse projeto.

5. CONCLUSÃO

Face ao exposto, esta pesquisa se propôs, como objetivo geral, a elaboração de um protótipo de um sistema domótico controlado por voz para acionamento de um ar condicionado, uma televisão e a iluminação de um ambiente, para pessoas com deficiências motoras. Com tudo observamos a importância da sociedade acadêmica está preocupada com o desenvolvimento de tecnologias assistivas, visando melhorar a qualidade de vida das pessoas com deficiência; no mais o uso destas tecnologias instiga os alunos a terem contato com novos equipamentos e novas soluções que o mundo digital nos oferecem.

Mesmo que o protótipo ainda necessite de melhorias, com componentes mais adequados para a comunicação, devido à falta de compatibilidade entre os módulos utilizados para o desenvolvimento do mesmo, percebe-se que ele se comporta bem, apesar de possuir alguns atrasos e não conseguir compreender algumas vezes o comando ordenado.

Com a utilização do kit de prototipagem eletrônica, no caso o ESP8266 Wemos D1 Mini, percebe-se um novo caminho a ser descoberto na área de domótica, abrangendo não só pessoas com deficiências físicas, mas trazendo comodidade aos usuários dessa tecnologia. Cabe aos estudantes desbravar essas novas possibilidades e tornar essa tecnologia cada vez mais acessível a todos.

Para o acionamento dos eletrodomésticos envolvidos e a iluminação foram desenvolvidos vários programas cada um com sua determinada função, foram utilizados de programas já prontos cedidos pelos módulos para a execução dos mesmos. Para que se compreenda o comando, fez-se necessários gravá-los no módulo de reconhecimentos de voz e dividi-los em grupos, depois a criação do programa que ao reconhecer o comando envia-se as informações para que as execuções fossem feitas. Foi utilizado o protocolo MQTT e armazenada os comandos enviados na nuvem e depois distribuídos para os respectivos módulos de execução. Com isso foi possível construir um sistema sem fio, permitindo ao usuário mover os módulos junto com seus eletrodomésticos sem precisar aumentar cabos, o concedendo mais liberdade.

Mesmo com os problemas de comunicação e entendimento o protótipo se demonstrou funcional e de fácil manipulação, sendo possível alterar seu código e aumentar o número de comando e conseqüentemente o número de eletrodomésticos conectados.

5.1 DIFICULDADES ENCONTRADAS

Devido à falta de compatibilidade entre a comunicação RX,TX do módulo de reconhecimento de voz e o ESP8266 Wemos D1 mini, perdeu-se muito tempo até achar essa solução viável, que torna meio redundante o tratamento da comando, desperdiçando um pouco a capacidade de processamento do Wemos D1 mini, são dificuldades que encontramos no caminho e devemos contorna-las. A segunda dificuldade encontrada tem mais haver com a logística devido os componentes não estarem disponíveis na cidade de Manaus foi necessário pedir equipamentos de fora, alguns de São Paulo, outros da China o que deixou a montagem do protótipo cada vez mais próxima da apresentação, para contornar essa dificuldade foi preciso pedir a maioria dos componentes emprestados para concluir o projeto. Terceiro e menos esperado foi a dificuldade de coletar as strings do ar condicionado do laboratório de microprocessados onde seriam desenvolvidos os testes, mas devido esse problema foi necessário mudar o local para realizar os testes do protótipo em ambiente real.

5.2 TRABALHOS FUTUROS

Nesta seção são apresentadas propostas de trabalhos futuros, visando a melhoria da funcionalidade e expansão das aplicações do protótipo, integrando ainda mais eletrônicos através do IOT.

Faz-se necessário descobrir um meio mais simples de fazer a comunicação entre o modulo de reconhecimento de voz e o ESP8266 Wemos D1 Mini, para tirar o Arduino Nano como intermediário nessa comunicação e deixa-la direta.

Implementação de novos equipamentos expandindo de um cômodo para controle da casa, deixando a mesma inteligente, integrando também o uso dos smartphones para os acionamentos.

Outra proposta no campo de tecnologia assistiva é adicionar outros tipos de deficiências ao projeto, afim dele ser mais abrangente, utilizando outros tipos de sensores, para pessoas com deficiência motoras e deficiências na fala.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABELHA, Daniel. **O que é instrumentação industrial?**.2014. Disponível em: <<http://24volts.com.br/instrumentacao/o-que-e-instrumentacao-industrial/>>, acessado 25 de set. 2018.
- ADAMI, Anna. **Domótica**. Disponível em: <<https://www.infoescola.com/tecnologia/domotica/>>, acessado 24 de set. 2018
- ARDUINO E CIA. **Enviando mensagens MQTT com módulo ESP32**. Disponível em: <<https://www.arduinoecia.com.br/enviando-mensagens-mqtt-modulo-esp32-wifi/>>, acessado em 15 de novembro de 2019.
- AUTOCORE ROBÓTICA. **Regulador de tensão 5v**. Disponível em: <<https://www.autocorerobotica.com.br/7905-ci-regulador-de-tensao-negativo-5v-1a>>, acessado em 17 de novembro de 2019.
- BAÚ DA ELETRÔNICA. **Optoacoplador**. Disponível em: <https://www.baudaeletronica.com.br/optoacoplador-pc817.html?gclid=CjwKCAiA_MPuBRB5EiwAHTTvmMeVBdIgyTAA1J9Q-OhaMIK43IPhbIJegSG2PsxhlVyIf1VI-NZvilhoC8XEQAvD_BwE>, acessado em 17 de novembro de 2019.
- BAÚ DA ELETRÔNICA. **Buzzer**. Disponível em: <https://www.baudaeletronica.com.br/buzzer-5v.html?gclid=CjwKCAiA_MPuBRB5EiwAHTTvmMSk-fRflQhdfNJrf5KNVhKrh0J9S11VrO7OHmtd4I2q36aCWn4UBVxoCDVUQAvD_BwE>, acessado em 17 de novembro de 2019.
- BATRINU, Catalin. **Projetos de Automação Residencial com ESP8266**. São Paulo: Novatec,2018.
- BRASIL ESCOLA. **O que é Infravermelho?**. Disponível em :<<https://brasile scola.uol.com.br/o-que-e/fisica/o-que-e-infravermelho.html>>, acessado em 22 de setembro de 2019.
- Eletronica PT.Relé. Disponível em: <<https://www.electronica-pt.com/rele>>, acessado em 15 de novembro de 2019.
- CARDOSO, David. **Sistema de controle de dispositivos para auxiliar portadores de deficiências físicas**. Manaus,2018.
- CASA EFICIENTE. **Eficiência, controle e segurança**.2018. Disponível em: <<https://casaeficiente.com/2018/01/16/eficiencia-controlo-e-seguranca-os-eixos-que-comandam-a-domotica/>>, acessado 24 de set. 2018.
- CASA ADAPTADA. **Rio de Janeiro é a capital do Sudeste com maior índice de pessoas com deficiência e mobilidade reduzida**. Disponível em: <<https://casadaptada.com.br/2018/07/rio-de-janeiro-e-a-capital-do-sudeste-com-maior-indice-de-pessoas-com-deficiencia-e-mobilidade-reduzida/>>, acessado em 15 de novembro de 2019.

COMPIC. **Microcontrolador PIC 16F88A**. Disponível em : <<https://compic.es/pic-micro/210-microcontrolador-pic-16f88a.html>>, acessado 24 de set. 2018.

CURTO CIRCUITO. **Módulo Relé**. Disponível em: <<https://www.curtocircuito.com.br/modulo-rele-wifi-esp8266-sem-esp01.html/>>, acessado em 17 de novembro de 2019.

CURVELLO, André. **Apresentando o módulo ESP8266**.2015. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>, acessado em 19 de novembro de 2019.

ESCAPEQUOTES. **Wemos D1 mini pins and diagram**. Disponível em: <<https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>>, acessado em 15 de novembro de 2019.

ELECHOUSE. **Voice recognition v3**.2014. Disponível em: <https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf>, acessado em 19 de novembro de 2019.

EVANS, Dave. **A Internet das Coisas Como a próxima evolução da Internet está mudando tudo**. Cisco Internet Business Solutions Group (IBSG)Cisco IBSG © 2011 Cisco e/ou suas afiliadas. Todos os direitos reservados. 2011

FLIPFLOP. **Sensor de Distância Ultrassônico HC-SR04**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>>, acessado 26 set. 2018.

FBS ELETRONICA. **Apostila Arduino**. Disponível em <www.fbseletronica.com.br>, acessado 24 de set. 2018.

HEWLETT PACKARD ENTERPRISE. **O que é internet das coisas**. Disponível em: <<https://www.hpe.com/br/pt/what-is/internet-of-things.html>>, acessado em 22 de setembro de 2019.

Landim, Tiago. **Sistema de comandos e identificação de voz**. São Carlos, 2017.

LIMA, E.M.; NOBRE, A.Y.; ALENCAR, R.A.. **Automação de baixo custo com Arduino mega e ethernet shield**. Ceará, 2015.

MURATORI, José Roberto; DAL BÓ, Paulo Henrique. **Automação Residencial**. Disponível em: <http://www.instalacoeseletricas.com/download/Automacao_residencial1.pdf>, acessado 24 de set. 2018.

MERCADO LIVRE. **Capacitor Eletrolítico**. Disponível em: <https://produto.mercadolivre.com.br/MLB-1021132031-80-capacitores-eletrolitico-47uf-x-450v-105-c-original-_JM?matt_tool=79246729&matt_word&gclid=CjwKCAiA_MPuBRB5EiwAHTTvMSBJVN80pGlkCUqQO2kRe1PTDSDpmKOxf6I2HJdb1tLjEl7myIqL6hoCXJUQAvD_BwE&quantity=1&variation=28317472133>, acessado em 17 de novembro de 2019.

PASTSKO, Luís. **Tutoriais aplicações, funcionamento e utilização de sensores**. 2006.

Disponível em:

<http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>, acessado 25 set. 2018.

PENIDO, Édilus; TRINDADE, Ronaldo. **Microcontroladores**.2013. Disponível em:

<<https://www2.ifmg.edu.br/ceadop3/apostilas/microcontroladores>>, acessado 25 de set. 2018.

THEENGGPROJETS. **Introducion to Arduino Nano**. Disponível em:

<<https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>>, acessado em 15 de novembro de 2019.

THEENGGPROJETS. **Introducion to ESP8266**. Disponível em:

<<https://www.theengineeringprojects.com/2018/08/introduction-to-esp8266.html>>, acessado em 15 de novembro de 2019.

APÊNDICE A- CÓDIGO DECIFRADOR DE IR

```

#include <IRremoteESP8266.h>

int RECV_PIN = 9;
  IRrecv irrecv(RECV_PIN);
  decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}
void dump(decode_results *results) {
  int count = results->rawlen;
  Serial.print("Raw (");
  Serial.print(count, DEC);
  Serial.print("): ");

  for (int i = 1; i < count; i++) {
    if (i & 1) {
      Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
    }
    else {
      Serial.write('-');
      Serial.print((unsigned long) results->rawbuf[i]*USECPERTICK, DEC);
    }
    Serial.print(" ");
  }
  Serial.println();
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    dump(&results);
    irrecv.resume();
  }
}

```

APÊNDICE B- CÓDIGO ARDUINO NANO

```

#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"

VR myVR(2, 3); // 2:RX 3:TX, você pode mudar para seus pinos de preferência.

uint8_t record[7]; // Gravações salvas
uint8_t buf[64];

int group = 0; //modificador de grupo
int lamp = 6;

//usando os leds para envio de comando binários, para comunicar com o Wemos mini
int led = A0;
int led1 = A1;
int led2 = A2;
int led3 = A3;

// definição das gravações e sua respectivas variáveis, divididos em seus grupos.
#define switchgroup      (0)

#define ar_liga_desliga   (1)
#define ar_temperaura_maior (2)
#define ar_temperatura_menor (3)
#define ar_mode           (4)

#define lamp_liga_desliga (5)

#define tv_liga_desliga   (6)
#define tv_volume_mais    (7)
#define tv_volume_menos   (8)
#define tv_canal_mais     (9)
#define tv_canal_menos    (10)
#define tv_canal_4        (11)
#define buzzer 6

void setup()
{
  // inicialização
  myVR.begin(9600);

  Serial.begin(115200);
  Serial.println("voice recognition");
  // set dos leds e buzzer
  pinMode(led, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

```

```

pinMode(led3, OUTPUT);
pinMode(buzzer, OUTPUT);
digitalWrite(buzzer, LOW);

if (myVR.clear() == 0) {
  Serial.println("Gravações limpas.");
} else {
  Serial.println("não encontramos o VoiceRecognitionModule.");
  Serial.println("por favor check a conexão e reinicie o arduino.");
  while (1);
}
//como o modulo só le 7 comandos por vez setamos o primeiro grupo
record[0] = switchgroup;
record[1] = ar_liga_desliga;
record[2] = ar_temperaura_maior;
record[3] = ar_temperatura_menor;
record[4] = ar_mode;
record[5] = lamp_liga_desliga;
group = 0;
if (myVR.load(record, 6) >= 0) {
  Serial.println(F("loaded"));
}
}

```

```

void loop()
{
  int ret;
  ret = myVR.recognize(buf, 50);
  if (ret > 0) {
    switch (buf[1]) {
      //mudança de grupo caso seja identificado o comando
      case switchgroup:
        digitalWrite(led, (digitalRead(A0) ^ 1));
        digitalWrite(led1, (digitalRead(A1) ^ 1));
        digitalWrite(led2, (digitalRead(A2) ^ 1));
        digitalWrite(led3, (digitalRead(A3) ^ 1));
        digitalWrite(led, HIGH);
        digitalWrite(led1, HIGH);
        digitalWrite(led2, HIGH);
        digitalWrite(led3, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(500);
        digitalWrite(buzzer, LOW);
        digitalWrite(led, LOW);
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(led3, LOW);
        Serial.println("Mudando de grupo");
        if (group == 0) {

```

```

    group = 1;
    Serial.println("Grupo da televisao acionado");
  }
  else {
    group = 0;
    Serial.println("Grupo do ar acionado");
  }
  switch (group) {
    case 0:
      myVR.clear();
      record[0] = switchgroup;
      record[1] = ar_liga_desliga;
      record[2] = ar_temperatura_maior;
      record[3] = ar_temperatura_menor;
      record[4] = ar_mode;
      record[5] = lamp_liga_desliga;
      if (myVR.load(record, 7) >= 0) {
        printRecord(record, 7);
        Serial.println(F("carregado o grupo do ar_e_lamp"));
      }
      break;
    case 1:
      myVR.clear();
      record[0] = switchgroup;
      record[1] = tv_liga_desliga;
      record[2] = tv_volume_mais;
      record[3] = tv_volume_menos ;
      record[4] = tv_canal_mais;
      record[5] = tv_canal_menos;
      record[6] = tv_canal_4;
      if (myVR.load(record, 6) >= 0) {
        Serial.println(F("carregado o grupo da tevelis o"));
      }
      break;
  }
}

```

```
break;
```

```

//grupor ar condicionado
case ar_liga_desliga:
  //liga e desliga ar
  digitalWrite(led, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(500);
  digitalWrite(buzzer, LOW);
  digitalWrite(led, LOW);
  break;

```

```
case ar_temperatura_maior:
```

```
//aumentar temp do ar
digitalWrite(led1, HIGH);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
digitalWrite(led1, LOW);
break;

case ar_temperatura_menor:
//diminuir temp do ar
digitalWrite(led2, HIGH);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
digitalWrite(led2, LOW);
break;

case ar_mode:
//mode do ar
digitalWrite(led3, HIGH);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
digitalWrite(led3, LOW);
break;

case lamp_liga_desliga:
//liga e desliga lampada
digitalWrite(led, HIGH);
digitalWrite(led1, HIGH);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
digitalWrite(led, LOW);
digitalWrite(led1, LOW);
break;

//grupo televisão

case tv_liga_desliga:
//liga e desliga tv
digitalWrite(led, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
digitalWrite(led, LOW);
digitalWrite(led2, LOW);
break;
```

```
case tv_volume_mais:
    //aumenta volume
    digitalWrite(led, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(led, LOW);
    digitalWrite(led3, LOW);
    break;

case tv_volume_menos:
    //diminui volume
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    break;

case tv_canal_mais:
    //aumenta canal
    digitalWrite(led1, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(led1, LOW);
    digitalWrite(led3, LOW);
    break;

case tv_canal_menos:
    //diminui canal
    digitalWrite(led3, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led2, LOW);
    break;

case tv_canal_4:
    //muda par canal 4
    digitalWrite(led, HIGH);
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(buzzer, HIGH);
```

```

    delay(500);
    digitalWrite(buzzer, LOW);
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led, LOW);
    break;

default:
    Serial.println("Não definido");
    break;
}
printVR(buf);
}
}

void printVR(uint8_t *buf)
{
    Serial.println("VR Index\tGroup\tRecordNum\tSignature");

    Serial.print(buf[2], DEC);
    Serial.print("\t\t");

    if (buf[0] == 0xFF) {
        Serial.print("NONE");
    }
    else if (buf[0] & 0x80) {
        Serial.print("UG ");
        Serial.print(buf[0] & (~0x80), DEC);
    }
    else {
        Serial.print("SG ");
        Serial.print(buf[0], DEC);
    }
    Serial.print("\t");

    Serial.print(buf[1], DEC);
    Serial.print("\t\t");
    if (buf[3] > 0) {
        printSignature(buf + 4, buf[3]);
    }
    else {
        Serial.print("NONE");
    }
    // Serial.println("\r\n");
    Serial.println();
}

void printSignature(uint8_t *buf, int len)
{

```

```
int i;
for (i = 0; i < len; i++) {
    if (buf[i] > 0x19 && buf[i] < 0x7F) {
        Serial.write(buf[i]);
    }
    else {
        Serial.print("[");
        Serial.print(buf[i], HEX);
        Serial.print("]");
    }
}

void printRecord(uint8_t *buf, uint8_t len)
{
    Serial.print(F("Record: "));
    for (int i = 0; i < len; i++) {
        Serial.print(buf[i], DEC);
        Serial.print(", ");
    }
}
```

APÊNDICE C- CÓDIGO WEMOS D1 MINI

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ArduinoOTA.h>
#include <ESP8266mDNS.h>
#include <ESP8266HTTPUpdateServer.h>
#include <ESP8266WebServer.h>
#include <SPI.h>
#include <Ethernet.h>
#define lamp 2;
#define statusWifi D5
#define in1 D4
#define in2 D0
#define in3 D8
#define in4 D7
int cont = 0;
int bin1;
int bin2;
int bin3;
int bin4;
int binario = 0;

// Mude os valores de acordo com sua Rede de internet

const char* ssid = "Thiago&Nayane"; //Nome da rede
const char* password = "32233242"; //Senha da rede
//Servidor MQTT
const char* mqtt_port = "14613";
const char* Server_mqtt = "farmer.cloudmqtt.com";
const char* User = "joaoPedro";
const char* Pass = "21dejunho";
int port = 14613;
const char* mdnsName = "Home-Automation";
//-----OTA-----
const char *OTAName = "Home-Automation-JP"; // A name and a password for the OTA
service

//-----
ESP8266WebServer httpServer(80);
ESP8266HTTPUpdateServer httpUpdater;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
char Temp[50];
void envia();
void setup_wifi();

```

```

void startOTA();
void startMDNS();
void lcd_standart();
void lcd_OTA();
void callback(char* topic, byte* payload, unsigned int length);
void reconnect();

//-----
void setup() {

    pinMode(in1, INPUT);
    pinMode(in2, INPUT);
    pinMode(in3, INPUT);
    pinMode(in4, INPUT);
    pinMode(statusWifi, OUTPUT);
    digitalWrite(statusWifi, LOW);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);

    Serial.begin(115200);
    setup_wifi();
    client.setServer(Server_mqtt, 14613);
    client.setCallback(callback);
    startOTA();
    startMDNS();
    delay(1000);
}
//-----
void loop() {

    envia();

    ArduinoOTA.handle();
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    httpServer.handleClient();

}
//-----
void envia() {
    bin1 = digitalRead(in1);
    bin2 = digitalRead(in2);
    bin3 = digitalRead(in3);
    bin4 = digitalRead(in4);

```

```

bin1 = bin1 << 3;
bin2 = bin2 << 2;
bin3 = bin3 << 1;
bin4 = bin4 << 0;
binario = bin1 + bin2 + bin3 + bin4;
Serial.println(binario, BIN);

//-----Liga e Desliga arcondicionado-----
if (binario == 8 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/ar", "liga_desliga_ar");
    cont = 0;
    binario = 0;
    delay(600);
  }
}
//-----Aumenta a temperatura do ar-----
if (binario == 4 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/ar", "temp_+");
    cont = 0;
    binario = 0;
    delay(600);
  }
}
//-----muda a opção do ar-----
if (binario == 1 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/ar", "mode");
    cont = 0;
    binario = 0;
    delay(600);
  }
}
//-----Diminui a temperatura do ar-----
if (binario == 2 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/ar", "temp_-");
    cont = 0;
    binario = 0;
    delay(600);
  }
}
//-----Liga e deliga a lampada-----
if (binario == 12 && cont == 0) {
  cont = 1;

```

```

if (cont == 1) {
  client.publish("home/lamp", "liga_desliga_lamp");
  cont = 0;
  binario = 0;
  delay(600);

}
}
//-----Liga e desliga a Televisão-----
if (binario == 10 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/tv", "liga_desliga_tv");
    cont = 0;
    binario = 0;
    delay(600);

  }
}
//-----Aumenta volume da Televisão-----
if (binario == 9 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/tv", "vol_+");
    cont = 0;
    binario = 0;
    delay(600);

  }
}
//-----Diminui o volume da Televisão-----
if (binario == 6 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/tv", "vol_-");
    cont = 0;
    binario = 0;
    delay(600);

  }
}
//-----Aumenta o canal da Televisão-----
if (binario == 5 && cont == 0) {
  cont = 1;
  if (cont == 1) {
    client.publish("home/tv", "canal_+");
    cont = 0;
    binario = 0;
    delay(600);

```

```

}
}
//-----Diminui o canal da televisão-----
  if (binario == 3 && cont == 0) {
    cont = 1;
    if (cont == 1) {
      client.publish("home/tv", "canal_-");
      cont = 0;
      binario = 0;
      delay(600);

    }
  }
//-----canal 4 da televisão-----
  if (binario == 14 && cont == 0) {
    cont = 1;
    if (cont == 1) {
      client.publish("home/tv", "canal_4");
      cont = 0;
      binario = 0;
      delay(600);

    }
  }

}
//-----
void startMDNS() { // Start the mDNS responder
MDNS.begin(mdnsName); // start the multicast domain name server
Serial.print("mDNS responder started: http://");
Serial.print(mdnsName);
Serial.println(".local");

MDNS.begin("Manager");

  httpUpdater.setup(&httpServer);
  httpServer.begin();

  MDNS.addService("http", "tcp", 80);
  Serial.printf("HTTPUpdateServer ready! Open http://Manager.local/update in your
browser\n");
}
//-----
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String payloadstr= "";
  for (int i = 0; i < length; i++) {
    payloadstr +=(char)payload[i];

```

```

}
Serial.println();
//-----
void startOTA() {
// Start the OTA service
//ArduinoOTA.setPort(82661);
ArduinoOTA.setHostname(OTAName);
//ArduinoOTA.setPassword((const char*)"12101991");
lcd_OTA();
ArduinoOTA.onStart([]() {
Serial.println("Start");

});
ArduinoOTA.onEnd([]() {
Serial.println("\r\nEnd");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
});
ArduinoOTA.onError([](ota_error_t error) {
Serial.printf("Error[%u]: ", error);
if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
else if (error == OTA_END_ERROR) Serial.println("End Failed");
});
ArduinoOTA.begin();
Serial.println("OTA ready\r\n");
}
//-----
void reconnect() {
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("MQTT connection...");
// Create a random client ID
String clientId = "Thiago";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str(),User,Pass)){

Serial.println("connected");
digitalWrite(statusWifi,HIGH);

// ... and subscriptions-----
client.subscribe("home/ar");
client.subscribe("home/tv");
client.subscribe("home/lamp");

} else {

```

```

Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
//digitalWrite(D1,LOW);
delay(5000);

}
}
}
//-----
void setup_wifi(){

delay(10);
// We start by connecting to a WiFi network
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.waitForConnectResult() != WL_CONNECTED) {
  Serial.println("Conexao falhou! Reiniciando...");
  delay(5000);
  ESP.restart();
}
randomSeed(micros());

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

}

```

APÊNDICE D - CÓDIGO MÓDULO RELÉ

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include<LiquidCrystal_I2C.h>
#include <ArduinoOTA.h>
#include <ESP8266mDNS.h>
#include <ESP8266HTTPUpdateServer.h>
#include <ESP8266WebServer.h>
#include <SPI.h>
#include <Ethernet.h>

int cont = 0;
int bin1;
int bin2;
int bin3;
int bin4;
int binario=0;

const char* ssid = "Thiago&Nayane";
const char* password = "32233242";
const char* mqtt_port = "14613";
const char* Server_mqtt = "farmer.cloudmqtt.com";
const char* User = "joaoPedro";
const char* Pass = "21dejunho";
int port = 14613;
const char* mdnsName = "Home-Automation";
//-----OTA-----
const char *OTAName = "Home-Automation-JP-lamp"; // A name and a password for the OTA
service

//-----
//-----
//-----
ESP8266WebServer httpServer(80);
ESP8266HTTPUpdateServer httpUpdater;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
char Temp[50];
void envia();
void setup_wifi();
void startOTA();
void startMDNS();

```

```

void callback(char* topic, byte* payload, unsigned int length);
void reconnect();
//-----
void setup() {

  pinMode(0, OUTPUT);
  digitalWrite(0, HIGH);

  Serial.begin(115200);
  setup_wifi();
  client.setServer(Server_mqtt, 14613);
  client.setCallback(callback);
  startOTA();
  startMDNS();
  delay(1000);

}
//-----
void loop() {

  ArduinoOTA.handle();
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  httpServer.handleClient();

}
//-----
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String payloadstr = "";
  for (int i = 0; i < length; i++) {
    payloadstr += (char)payload[i];
  }
  Serial.println();
  //-----identificação do comando da lampada-----
  if (String(topic).equals("home/lamp")) {
    // se o tópico for home/lamp
    //-----Liga e Desliga lampada-----
    if ((char)payloadstr.equals("liga_desliga_lamp")) {
      digitalWrite(0, digitalRead(0) ^ 1);
    }
  }
}

```

```

    Serial.println(digitalRead(0));
    Serial.println("Lamp_on_or_off");
}

}
}
//-----
void startOTA() {
// Start the OTA service
//ArduinoOTA.setPort(82661);
ArduinoOTA.setHostname(OTAName);
//ArduinoOTA.setPassword((const char*)"12101991");
lcd_OTA();
ArduinoOTA.onStart([]() {
Serial.println("Start");

});
ArduinoOTA.onEnd([]() {
Serial.println("\r\nEnd");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
});
ArduinoOTA.onError([](ota_error_t error) {
Serial.printf("Error[%u]: ", error);
if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
else if (error == OTA_END_ERROR) Serial.println("End Failed");
});
ArduinoOTA.begin();
Serial.println("OTA ready\r\n");
}
//-----
void reconnect() {
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("MQTT connection...");
// Create a random client ID
String clientId = "Thiago";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str(),User,Pass)){

Serial.println("connected");
digitalWrite(statusWifi,HIGH);

// ... and subscriptions-----

```

```

    client.subscribe("home/ar");
    client.subscribe("home/tv");
    client.subscribe("home/lamp");

    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        // Wait 5 seconds before retrying
        //digitalWrite(D1,LOW);
        delay(5000);

    }
}
}
}
//-----
void setup_wifi(){

    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.waitForConnectResult() != WL_CONNECTED) {
        Serial.println("Conexao falhou! Reiniciando...");
        delay(5000);
        ESP.restart();
    }
    randomSeed(micros());

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

}

//-----
void startMDNS() { // Start the mDNS responder
MDNS.begin(mdnsName); // start the multicast domain name server
Serial.print("mDNS responder started: http://");
Serial.print(mdnsName);
Serial.println(".local");
}

```

```
MDNS.begin("Manager");

httpUpdater.setup(&httpServer);
httpServer.begin();

MDNS.addService("http", "tcp", 80);
Serial.printf("HTTPUpdateServer ready! Open http://Manager.local/update in your
browser\n");
}
```

APÊNDICE E - CÓDIGO MÓDULO RELÉ

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <LiquidCrystal_I2C.h>
#include <ArduinoOTA.h>
#include <ESP8266mDNS.h>
#include <ESP8266HTTPUpdateServer.h>
#include <ESP8266WebServer.h>
#include <SPI.h>
#include <Ethernet.h>
#include <IRremoteESP8266.h>
```

```
IRsend irsend(14); //FUNÇÃO RESPONSÁVEL PELO MÉTODO DE ENVIO DO SINAL IR
/ UTILIZA O GPIO14(D5)
```

```
int frequencia = 38; //FREQUÊNCIA DO SINAL IR(32KHz)
int cont = 0;
int bin1;
int bin2;
int bin3;
int bin4;
int binario = 0;
```

```
const char* ssid = "Thiago&Nayane";
const char* password = "32233242";
const char* mqtt_port = "14613";
const char* Server_mqtt = "farmer.cloudmqtt.com";
const char* User = "joaoPedro";
const char* Pass = "21dejunho";
int port = 14613;
const char* mdnsName = "Home-Automation";
//-----OTA-----
const char *OTAName = "Home-Automation-JP-IR"; // A name and a password for the OTA
service
```

```
//----Botoes do controle-----
```

```
unsigned int liga_tv[] = {3500, 1700, 450, 400, 500, 1250, 450, 400, 500, 400, 450, 400, 450,
450, 450, 400, 450, 400, 450, 400, 500, 400, 450, 400, 450, 450, 450, 400, 450, 1250, 500, 400,
450, 450, 450, 400, 450, 400, 450, 400, 500, 400, 450, 400, 450, 450, 450, 400, 450, 1250, 500,
400, 450, 400, 450, 450, 450, 400, 450, 400, 450, 400, 500, 400, 450, 400, 500, 1250, 450, 400,
500, 1250, 450, 1300, 450, 1300, 450, 1300, 400, 400, 500, 400, 450, 1250, 500, 400, 450,
1300, 450, 1250, 500, 1250, 450, 1300, 450, 400, 450, 1300, 450};
```

```
unsigned int volume_maior[] = {3450, 1700, 450, 450, 400, 1350, 400, 450, 450, 400, 450, 450,
400, 450, 450, 450, 400, 450, 400, 450, 450, 400, 450, 450, 400, 450, 450, 400, 450, 1300, 450,
400, 450, 450, 450, 400, 450, 400, 450, 450, 450, 450, 400, 450, 450, 400, 450, 450, 400, 450, 1300,
```



```
500, 550, 500, 550, 500, 550, 1600, 550, 500, 550, 1600, 550, 500, 550, 500, 600, 500, 550,
1550, 600, 1550, 550, 500, 550, 1600, 550, 500, 550, 1600, 550, 1550, 600, 1550, 550};
```

```
unsigned int mode_[] = {4350, 4350, 550, 1600, 500, 550, 550, 1600, 500, 1600, 550, 550, 500,
550, 550, 1550, 550, 550, 550, 500, 550, 1600, 500, 550, 550, 550, 500, 1600, 550, 1550, 550,
550, 550, 1550, 550, 1600, 550, 500, 550, 1600, 550, 1550, 550, 1600, 550, 1600, 550, 1550,
550, 1600, 550, 500, 550, 1600, 550, 500, 550, 550, 500, 550, 550, 500, 550, 550, 500, 550,
550, 500, 550, 500, 550, 1600, 550, 500, 550, 550, 500, 550, 550, 500, 550, 500, 550, 1600,
550, 1550, 600, 500, 550, 1600, 550, 1550, 550, 1600, 550, 1600, 500, 1600, 550};
```

```
//-----
```

```
ESP8266WebServer httpServer(80);
ESP8266HTTPUpdateServer httpUpdater;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
char Temp[50];
void envia();
void setup_wifi();
void startOTA();
void startMDNS();
void callback(char* topic, byte* payload, unsigned int length);
void reconnect();
```

```
//-----
```

```
void setup() {
  //-----IRREMOTE-----
  irsend.begin(); //INICIALIZA A FUNÇÃO
```

```
//-----
```

```
pinMode(D3, OUTPUT);
digitalWrite(D3, LOW);
pinMode(D4, OUTPUT);
digitalWrite(D4, LOW);
Serial.begin(115200);
setup_wifi();
client.setServer(Server_mqtt, 14613);
client.setCallback(callback);
startOTA();
startMDNS();
delay(1000);
```

```
}
```

```
//-----
```

```
void loop() {
```

```

ArduinoOTA.handle();
if (!client.connected()) {
    reconnect();
}
client.loop();

httpServer.handleClient();

}
//-----
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    String payloadstr = "";
    for (int i = 0; i < length; i++) {
        payloadstr += (char)payload[i];
    }
    Serial.println();

    //-----identificação do comando ar condicionado-----
    if (String(topic).equals("home/ar")) {
        // se o tópico for home/ar
        //-----Liga e Desliga arcondicionado-----
        if ((char)payloadstr.equals("liga_desliga_ar")) {
            irsend.sendRaw(liga_ar, sizeof(liga_ar) / sizeof(liga_ar[0]), frequencia); //
PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
            Serial.println("Comando enviado: Liga / Desliga_AR");
            delay(50);
            digitalWrite(D3, HIGH);
            delay(500);
            digitalWrite(D3, LOW);
        }
        //-----Aumenta a temperatura do ar-----
        if ((char)payloadstr.equals("temp+")) {
            irsend.sendRaw(temp_maior, sizeof(temp_maior) / sizeof(temp_maior[0]), frequencia); //
PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
            Serial.println("Comando enviado: temp+");
            delay(50);
            digitalWrite(D3, HIGH);
            delay(500);
            digitalWrite(D3, LOW);
        }
        //-----muda a opção do ar-----
        if ((char)payloadstr.equals("mode")) {
            irsend.sendRaw(mode_, sizeof(mode_) / sizeof(mode_[0]), frequencia); // PARÂMETROS
NECESSÁRIOS PARA ENVIO DO SINAL IR
            Serial.println("Comando enviado: mode");
            delay(50);
            digitalWrite(D3, HIGH);

```

```

    delay(500);
    digitalWrite(D3, LOW);
  }
  //-----Diminui a temperatura do ar-----
  if ((char)payloadstr.equals("temp_-")) {
    irsend.sendRaw(temp_menor, sizeof(temp_menor) / sizeof(temp_menor[0]), frequencia); //
PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado:temp-");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
  }
}

//-----Liga e desliga a Televisão-----

if (String(topic).equals("home/tv")) {
  // se o tópico for home/tv
  //-----Liga e Desliga arcondicionado-----
  if ((char)payloadstr.equals("liga_desliga_tv")) {
    irsend.sendRaw(liga_tv, sizeof(liga_tv) / sizeof(liga_tv[0]), frequencia); //
PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado: Liga / Desliga_TV");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
  }

  //-----Aumenta volume da Televisão-----
  if ((char)payloadstr.equals("vol_+")) {
    irsend.sendRaw(volume_maior, sizeof(volume_maior) / sizeof(volume_maior[0]),
frequencia); // PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado: vol+");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
  }
  //-----Diminui o volume da Televisão-----
  if ((char)payloadstr.equals("vol_-")) {
    irsend.sendRaw(volume_menor, sizeof(volume_menor) / sizeof(volume_menor[0]),
frequencia); // PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado:vol-");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
  }
}

```

```

//-----Aumenta o canal da Televisão-----
if ((char)payloadstr.equals("canal_+")) {
    irsend.sendRaw(canal_maior, sizeof(canal_maior) / sizeof(canal_maior[0]), frequencia); //
    PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado: canal+");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
}
//-----Diminui o canal da televisão-----
if ((char)payloadstr.equals("canal_-")) {
    irsend.sendRaw(canal_menor, sizeof(canal_menor) / sizeof(canal_menor[0]), frequencia);
// PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado:canal-");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
}

//----- canal 4 da televisão-----
if ((char)payloadstr.equals("canal_4")) {
    irsend.sendRaw(canal_quatro, sizeof(canal_quatro) / sizeof(canal_quatro[0]), frequencia);
// PARÂMETROS NECESSÁRIOS PARA ENVIO DO SINAL IR
    Serial.println("Comando enviado:canal4");
    delay(50);
    digitalWrite(D3, HIGH);
    delay(500);
    digitalWrite(D3, LOW);
}

}

}

//-----
void startMDNS() {
    MDNS.begin(mdnsName);
    Serial.print("mDNS responder started: http://");
    Serial.print(mdnsName);
    Serial.println(".local");

    MDNS.begin("Manager");

    httpUpdater.setup(&httpServer);
    httpServer.begin();

    MDNS.addService("http", "tcp", 80);
    Serial.printf("HTTPUpdateServer ready! Open http://Manager.local/update in your
browser\n");
}

```

```

}
//-----
void startOTA() { // Start the OTA service
  //ArduinoOTA.setPort(82661);
  ArduinoOTA.setHostname(OTAName);
  //ArduinoOTA.setPassword((const char*)"12101991");
  lcd_OTA();
  ArduinoOTA.onStart([]() {
    Serial.println("Start");

  });
  ArduinoOTA.onEnd([]() {
    Serial.println("\r\nEnd");
  });
  ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
  });
  ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });
  ArduinoOTA.begin();
  Serial.println("OTA ready\r\n");
}
//-----
void reconnect() {

  while (!client.connected()) {

    Serial.print("MQTT connection...");

    String clientId = "Thiago";
    clientId += String(random(0xffff), HEX);

    if (client.connect(clientId.c_str(), User, Pass)) {

      Serial.println("connected");
      digitalWrite(D4, HIGH);

      // ... and subscriptions-----
      client.subscribe("home/ar");
      client.subscribe("home/tv");
      client.subscribe("home/lamp");

    } else {
      Serial.print("failed, rc=");

```

```
Serial.print(client.state());
Serial.println(" try again in 5 seconds");

digitalWrite(D4, LOW);
delay(5000);

}
}
}
//-----
void setup_wifi() {

delay(10);

Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
}
randomSeed(micros());

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

}
```