

**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES  
PRÓ-REITORIA DE ENSINO, PESQUISA E PÓS-GRADUAÇÃO  
CÂMPUS DE ERECHIM  
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO  
CURSO DE ENGENHARIA ELÉTRICA**

**ANDERSON ALEX ALVES DA COSTA**

**DESENVOLVIMENTO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL  
UTILIZANDO MICROCONTROLADOR**

**TRABALHO DE CONCLUSÃO DE CURSO**

**ERECHIM - RS  
2020**

**ANDERSON ALEX ALVES DA COSTA**

**DESENVOLVIMENTO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL  
UTILIZANDO MICROCONTROLADOR**

**Trabalho de Conclusão de curso apresentado ao Curso de Engenharia Elétrica como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica, Departamento de Engenharias e Ciência da Computação da Universidade Regional Integrada do Alto Uruguai e das Missões – Câmpus de Erechim.**

**Orientador:** Prof. Adilson Luís Stankiewicz

**ERECHIM - RS  
2020**

**ANDERSON ALEX ALVES DA COSTA**

**DESENVOLVIMENTO DE SISTEMA DE AUTOMAÇÃO RESIDENCIAL  
UTILIZANDO MICROCONTROLADOR**

**Trabalho de Conclusão de curso apresentado ao Curso de Engenharia Elétrica como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica, Departamento de Engenharias e Ciência da Computação da Universidade Regional Integrada do Alto Uruguai e das Missões – Câmpus de Erechim.**

Erechim, 09 de dezembro de 2020.

**BANCA EXAMINADORA**

---

Prof. Me. Adilson Luís Stankiewicz (Orientador)  
URI Erechim

---

Prof. Me. Claodomir Martinazzo (Examinador)  
URI Erechim

---

Prof.<sup>a</sup> Ma. Camila Sampaio dos Reis (Examinadora)  
URI Erechim

Dedico este trabalho a minha família, a qual nunca mediu esforços para que eu alcançasse todos os meus objetivos.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por me permitir viver tantos momentos especiais e ter me dado força para enfrentar os momentos difíceis, tornando meu sonho possível.

A minha família, minha base e apoio, pela ajuda e incentivo prestados para que eu alcançasse o sonho de ser bacharel em engenharia elétrica.

Agradeço a minha esposa, por estar sempre ao meu lado me incentivando e acreditando no meu potencial.

Nesta oportunidade, agradeço também ao meu orientador Prof. Me. Adilson Luís Stankiewicz, pela colaboração e empenho para a realização desse trabalho.

A todos os professores que contribuíram para a minha formação educacional, por todo conhecimento passado e pela dedicação incansável em formar profissionais exemplares.

Por fim, agradeço a todos os amigos que conheci durante todo o curso, que de certa forma, acompanharam esta longa e difícil caminhada.

*“Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças”*

**(Charles Darwin)**

## RESUMO

Desde o início a automação residencial é considerada um grande avanço da tecnologia, provendo o que em tempos passados era visto como um tema futurista em filmes e séries de televisão. Essa tecnologia que vem ganhando espaço no mercado nos últimos tempos, tem como objetivo proporcionar para as pessoas, praticidade, comodidade, segurança e economia. Visando por conforto e segurança, este projeto apresenta o estudo e implementação de um sistema de Automação Residencial de baixo custo, que tem como finalidade o monitoramento e controle dos dispositivos automatizados na residência, tais como controle de iluminação, sistema de alarme e monitoramento do nível do reservatório de água. Este trabalho também tem o objetivo de demonstrar a viabilidade de soluções de baixo custo, popularizando o acesso à automação residencial, para tal finalidade foi utilizado o microcontrolador ESP 32 como central de automação, gerenciado por meio de um aplicativo desenvolvido para *smartphone*.

**Palavras-chave:** Domótica, Microcontroladores, Automação Residencial, ESP 32.

## ABSTRACT

From the beginning, home automation has been considered a major advance in technology, providing what in the past was seen as a futuristic theme in films and television series. This technology that has been gaining space in the market recently, aims to provide people with practicality, convenience, security and economy. Aiming for comfort and safety, this project presents the study and implementation of a low cost Home Automation system, which aims to monitor and control the automated devices in the residence, such as lighting control, alarm system and level monitoring the water tank. This work also aims to demonstrate the feasibility of low-cost solutions, popularizing access to home automation for this purpose, the ESP 32 microcontroller was used as an automation center, managed through an application developed for a smartphone.

**Keywords:** Home automation, Microcontrollers, Home Automation, ESP 32.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Previsões de crescimento da AR(em bilhões de dólares) .....	19
Figura 2 – Arquitetura centralizada. ....	21
Figura 3 – Arquitetura descentralizada.....	22
Figura 4 - Blocos básicos de um microcomputador .....	24
Figura 5 - Arquitetura Von Neumann .....	26
Figura 6 - Arquitetura Harvard.....	26
Figura 7 - Arduino Mega 2560 .....	27
Figura 8 - Ethernet Shild .....	28
Figura 9 - ESP 32 .....	29
Figura 10 - Sensor de Nível (Tipo Boia Horizontal).....	31
Figura 11 - Módulo relé simples .....	32
Figura 12 - <i>datasheet</i> do NodeMCU-32 .....	33
Figura 13 – Esquemático do sistema de monitoramento de nível.....	35
Figura 14 – Sensor de temperatura DS18B20 .....	36
Figura 15 - Módulo RFID-RC522 .....	37
Figura 16 - Sensor de Presença PIR - HC-SR50 .....	37
Figura 17 – Instalação da Biblioteca ESP32 .....	38
Figura 18 – Definição dos Pinos a serem utilizados.....	39
Figura 19 – Definição dos pinos de entrada e saída de dados .....	39
Figura 20 – Programação para realizar a conexão Wi-Fi .....	40
Figura 21 – Tela de desenvolvimento do APP .....	41
Figura 22 – Lista de telas do Aplicativo.....	42
Figura 23 – Esquema de ligação proposto para a central de automação interna.....	43
Figura 24 – Montagem do Protótipo.....	44
Figura 25 – Conexão do ESP 32 a rede local .....	45
Figura 26 – Dados para conexão a Rede local do ESP 32.....	45
Figura 27– Teste do Controle de Iluminação .....	46
Figura 28 – Reservatório Vazio.....	47
Figura 29 – Nível do Reservatório de 40% a 60% .....	47
Figura 30 – Nível do Reservatório em 100% .....	47
Figura 31 – Alarme Desligado.....	48
Figura 32 – Alarme Ligado .....	48

Figura 33 – Alarme Disparado .....	48
Figura 34 – Interface de Verificação da Temperatura.....	49
Figura 35 – Acesso Liberado.....	49
Figura 36 – Acesso Negado.....	49
Quadro 1 - Resultado da pesquisa .....	19
Quadro 2 - Especificações Arduino MEGA2560.....	28
Quadro 3 – Especificações ESP 32 .....	30
Quadro 4 – Cômodos e Dispositivos automatizados.....	35

## **LISTA DE TABELAS**

Tabela 1 – Tabela Verdade do acionamento da Motobomba .....	36
Tabela 2 – Componentes do protótipo de bancada.....	44
Tabela 3 - Demonstração de valores e quantidade dos componentes utilizados. ....	50

## **APÊNDICES**

APÊNDICE A – Planta Baixa da residência a ser instalado o sistema futuramente.

APÊNDICE B – Código Fonte do Sistema de Automação Interno da Residência.

APÊNDICE C – Código Fonte do Sistema de Controle de Acesso.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
<b>1.1 Objetivo.....</b>	<b>14</b>
1.1.1 Objetivos Específico.....	15
<b>1.2 Organização do Trabalho.....</b>	<b>15</b>
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>16</b>
<b>2.1 Automação Residencial.....</b>	<b>16</b>
2.1.1 Breve História da Automação Residencial.....	17
2.1.2 Atual Mercado da Automação Residencial.....	18
2.1.3 Vantagens da Automação Residencial.....	20
2.1.4 Arquitetura dos sistemas de Automação.....	20
<b>2.2 Sistemas Embarcados .....</b>	<b>22</b>
2.2.1 Microprocessador.....	22
2.2.2 Microcontroladores.....	23
2.2.2.1 Arquitetura Von Neumann e Harvard.....	26
2.2.3 ARDUINO.....	27
2.2.4 ESP 32.....	29
<b>2.3 Sensores e Atuadores .....</b>	<b>30</b>
2.3.1 Sensores.....	30
2.3.1.1 Sensor de Nível.....	31
2.3.1.2 Sensor de Presença.....	31
2.3.2 Atuadores.....	32
2.3.2.1 Módulo Relé.....	32
<b>3 METODOLOGIA.....</b>	<b>33</b>
<b>3.1 Definição do Microcontrolador.....</b>	<b>33</b>
<b>3.2 Levantamento de Dados e Variáveis a Serem Controladas .....</b>	<b>34</b>
3.2.1 Iluminação.....	34
3.2.2 Monitoramento de Nível da Caixa D'água.....	35
3.2.3 Monitoramento de Temperatura.....	36
3.2.4 Sistema de Segurança.....	37
<b>3.3 Desenvolvimento do Projeto.....</b>	<b>38</b>
3.3.1 Programação do ESP32.....	38
3.3.2 Desenvolvimento do Aplicativo.....	41

3.3.3 Testes de bancada .....	42
<b>4 RESULTADOS E DISCUSSÕES.....</b>	<b>45</b>
<b>4.1 Conexão do Sistema .....</b>	<b>45</b>
<b>4.2 Validação do Sistema de Automação Interno da Residência .....</b>	<b>46</b>
<b>4.3 Validação do Controle de Acesso .....</b>	<b>49</b>
<b>4.4 Recursos Financeiros .....</b>	<b>50</b>
<b>4.5 Dificuldades Encontradas.....</b>	<b>50</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>51</b>
<b>5.1 Perspectivas Futuras .....</b>	<b>52</b>
<b>REFERÊNCIAS .....</b>	<b>53</b>

## 1 INTRODUÇÃO

A Automação Residencial ou Domótica é uma tecnologia que possibilita tornar atividades simples, como controlar a iluminação, portas e equipamentos eletrônicos de uma residência facilmente via aplicativo, promovendo uma mudança radical no ambiente em que a mesma está instalada, pois a informação fornecida por qualquer sensor, seja ele termostato, detector de movimento, sensor de luminosidade, entre outros, é colocada à disposição do morador por meio de uma rede interna da residência, podendo ser monitorada a qualquer momento.

Para Quinderé (2009), nas indústrias a automação já é uma realidade a muitos anos, devido a necessidade de simplificar ou realizar tarefas inadequadas ao ser humano, ou simplesmente para automatizar atividades como forma de redução de custo.

De fato, uma das principais características da casa automatizada é a capacidade de comunicação entre os equipamentos, compartilhando informações dos sensores e atuadores que compõem o sistema. Entre os atrativos desta tecnologia, o grande destaque é o conforto, pois por meio de uma interface de comando é possível controlar todos os dispositivos do sistema, deste modo é possível por exemplo: acender ou apagar a luz de qualquer cômodo sem a necessidade de se deslocar até o mesmo (BOLZANI, 2004).

Molina et al. (2019) argumentam que um projeto básico de automação residencial realizado por uma empresa especializada, tem um custo inicial de aproximadamente 3 mil reais, o que torna a implementação do sistema pouco acessível, pois segundo uma pesquisa realizado em 2017 pelo IBGE, o salário médio do Brasileiro é de R\$ 2.112,00.

Diante do exposto, este trabalho propõe o projeto de um sistema de automação residencial de baixo custo, para o acionamento e monitoramento dos principais dispositivos de uma residência, tais como, iluminação, sistema de alarme e controle de acesso do portão principal. Ademais, a implementação de um sistema de monitoramento da temperatura ambiente e monitoramento do nível do reservatório d'água. Para tal finalidade, para desenvolver o projeto foi utilizado um Microcontroladores ESP 32, controlado e monitorado via wi-fi, por meio de um aplicativo desenvolvido para *smartphone*.

### 1.1 Objetivo

Desenvolver um sistema de automação residencial utilizando microcontrolador como alternativa de baixo custo.

### 1.1.1 Objetivos Específico

- Realizar pesquisas bibliográficas referentes a automação residencial;
- Analisar a viabilidade dos principais Microcontroladores disponíveis no mercado para a finalidade proposta;
- Pesquisar e analisar os componentes a serem utilizados no projeto;
- Programação do microcontrolador;
- Desenvolvimento de uma interface para o gerenciamento do sistema;
- Instalação de um protótipo para teste da funcionalidade do sistema;
- Análise dos resultados.

## 1.2 Organização do Trabalho

O capítulo 1 apresenta a introdução do tema abordado, enfatizando a contextualização, aplicações, os objetivos traçados e a estrutura no qual o trabalho foi organizado.

O Capítulo 2 destina-se a apresentações das definições relacionadas a processos de automação num todo, apresentando os níveis de automação, tipos de arquiteturas existentes, vantagens proporcionadas pela automação residencial, um breve embasamento sobre microcontroladores e os diferentes tipos de sensores e atuadores responsáveis pelo acionamento dos equipamentos.

No Capítulo 3 apresentam-se os sistemas a serem automatizados na residência, bem como, os métodos propostos para a implementação de ambos. Ademais o capítulo apresenta em partes a depuração do código fonte do microcontrolador, a plataforma utilizada para o desenvolvimento do aplicativo, além de apresentar os procedimentos metodológicos utilizados na implementação prática do sistema de automação.

O capítulo 4 dispõe os resultados dos testes realizados em bancada do sistema de automação proposto, bem como, as sugestões para trabalhos futuros. Por fim, o capítulo 5 apresenta as considerações finais do trabalho.

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados conceitos importantes que foram fundamentais para o desenvolvimento do projeto, obtidos por meio do estudo de trabalhos já publicados na área, visando obter o embasamento teórico sobre a automação residencial, bem como os principais Microcontroladores e componentes que podem ser integrados neste sistema proposto.

### 2.1 Automação Residencial

A automação residencial, também conhecida por Domótica, termo originado da junção das palavras *Domus* (casa em latim) e a palavra robótica (controle automatizado de algo), é a tecnologia que integra os dispositivos eletrônicos e eletromecânicos de uma residência, possibilitando o controle de forma automática de tarefas que antes dependiam unicamente dos moradores, incluindo iluminação, sistema de som, climatização, controles de acesso, dentre outros dispositivos. (BOLZANI, 2004).

“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 possibilidades de alterações conforme seus 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.” (PEREIRA, 2019).

Segundo Roveri (2012), esta integração de atividades e instruções deve abranger todos os sistemas tecnológicos de uma residência, dentre eles os principais são:

- a) Controle da iluminação - com o controle automatizado da iluminação utilizando um smartphone ou um *tablet*, é possível apagar as luzes de qualquer cômodo caso tenha esquecido alguma acesa, ou simplesmente acendê-las quando estiver chegando em casa;
- b) Sistema de som - este sistema possibilita controlar quais cômodos da casa serão ligados os autofalantes, além de controlar o volume em cada um dos ambientes;
- c) Sistema de irrigação de jardim - Com a utilização de temporizadores ou sensores de umidade do solo, é possível automatizar a irrigação do jardim de acordo com as condições ideais para as plantas;

- d) Controle automatizado de cortinas e janelas - por meio deste sistema é possível fechar de forma automática as janelas caso tenha esquecido alguma aberta, ou simplesmente instalar sensores de luz para abrir as cortinas automaticamente aproveitando ao máximo a luz natural;
- e) Sistema de segurança – Com este sistema é possível automatizar a porta da residência utilizando biometria, sistemas de senha ou controle de acesso, assim como o sistema de alarmes e câmeras, podem ser monitorados remotamente enviando alertas ao proprietário de qualquer violação na segurança.

### 2.1.1 Breve História da Automação Residencial

Segundo Campos (2014), o marco inicial da automação residencial foi no final da década de 60, quando foram criados os primeiros CLPs (controladores lógicos programáveis), possibilitados pela evolução da microeletrônica muitas empresas de automação, que até então trabalhavam com automação industrial, mudaram seu foco para a automações em residências. (CAMPOS, 2014).

“Cronologicamente, o desenvolvimento dos sistemas de automação residencial surge depois de seus similares nas áreas industrial e comercial. Por óbvios motivos econômicos e de escala de produção, os fabricantes e os prestadores de serviços, num primeiro momento, se voltam a aqueles segmentos que lhes propiciam maior rapidez no retorno de seus investimentos.” (TEZA, 2002).

Ainda, segundo Campos (2014), a automação residencial começou a evoluir rapidamente no final da década de 70 com o lançamento do primeiro módulo de automação inteligente nos Estados Unidos, o “protocolo X-10”, o qual de maneira simples, por meio do método *Power Line Carrier* (PLC), permitia automatizar uma residência sem ter que alterar a infraestrutura da rede elétrica da mesma.

“Nos anos 80, aparecem os primeiros termos como *Smart House*, *Intelligent House* ou *Domótica*, visando à tendência a economia de energia. Surgem os sistemas de automação de segurança, iluminação e intrusão, mostrando coordenação entre componentes do mesmo sistema [...]. Nos anos 1990, a tendência foi a de integrar os diferentes sistemas (segurança, condicionamento ambiental, etc.) que compõem uma residência, com o objetivo de melhorar a produtividade e o conforto das pessoas. Conhecida como “Casa Sistema”, onde todos os elementos começam a se inter-relacionar tanto internamente como com o exterior.” (BUNEMER, 2014, p.12).

Na década de 90 esta tecnologia se popularizava lentamente nos lares Americanos, porém no Brasil, surgem comercialmente os primeiros módulos para automação somente no início do século XX, por se tratar de um produto importado, poucas empresas dispunham de tal

tecnologia, aliado a isto e a falta de mão de obra qualificada para instalação e manutenção, esta tecnologia chegou inicialmente aos lares brasileiros de alto padrão. (GOMES, 2016).

Atualmente com o avanço da tecnologia, novas soluções e métodos são criados constantemente, possibilitando o acesso a uma tecnologia que inicialmente era algo específico de uma classe economicamente alta, tornando a automação residencial um assunto cada vez mais presente no cotidiano das pessoas. (GOMES, 2016).

### 2.1.2 Atual Mercado da Automação Residencial

Para Jaramillo (2019), o mercado da automação residencial e predial está crescendo fortemente, impulsionado pelos dispositivos móveis como *smartphones* e *tablets* acompanhados da proliferação de aplicativos. Especialistas apostam neste mercado tanto no Brasil quanto no mundo, pois segundo uma pesquisa publicada no site da AURISIDE (Associação Brasileira de Automação Residencial e Predial), a projeção de crescimento do mercado de automação residencial no mundo é de 11,5% ao ano.

“A tendência chave que afetará predominantemente o mercado de automação residencial nos próximos anos é que a maioria das pessoas está preparada para ter acesso a uma enorme quantidade de dispositivos de automação residencial em quase todos os aspectos do seu estilo de vida.” (AURISIDE, 2018).

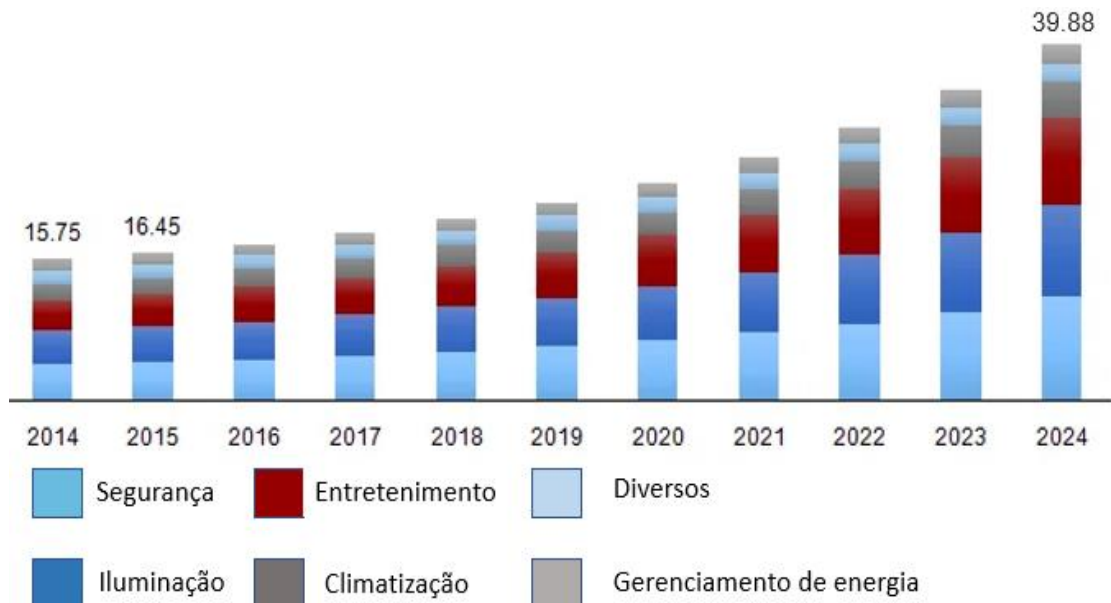
Segundo AURISIDE (2018), devido aos custos de instalação da automação caírem cerca de 50% nos últimos anos, o uso destas tecnologias começa a ganhar espaço em residências e apartamentos de classe média.

“Dois fatores contribuem para a procura dessa tecnologia e para o crescimento desse mercado – o primeiro é o ambiente residencial que permanece praticamente inexplorado para a implementação de sistemas de redes e de controle - o outro é o apelo ao novo que aflora desse tema. Para hobistas e entusiastas a automação residencial (AR) é sinônima de equipamentos de iluminação automatizados e home-theaters incrementados. Para os que enxergam além, a AR representa novas descobertas, desafios e oportunidades” (BOLZANI 2007).

Estima-se que o mercado global da automação deve chegar a US \$ 39,88 bilhões até 2024, este crescimento é um impacto do avanço da tecnologia que envolve a automação, aliado à sua constante evolução e redução de custos, tornando tal tecnologia uma opção mais atrativa a futuros clientes. (AURISIDE, 2019).

A Figura 1 apresenta em formato de gráfico as previsões de crescimento do mercado da automação residencial e predial até 2024.

Figura 1 - Previsões de crescimento da AR(em bilhões de dólares)



Fonte: UpMyTech.com, 2016.

Segundo Muratori (2018), atualmente possuem poucos estudos de mercado quando o assunto é automação residencial, principalmente com o foco voltado para o consumidor final visando suas preferências de automação.

O Quadro 1 apresenta os resultados de uma pesquisa realizada pela AURESIDE junto a um grupo de aproximadamente 200 empresas de projeto e integração no Brasil, com o intuito de obter os principais sistemas de automação solicitados pelos futuros clientes.

Quadro 1 - Resultado da pesquisa

Sistema	Percentual (%)
Controle de iluminação	83,8
Home Theater e som ambiente	63,1
Sistema de Câmeras	62,2
Redes sem fios	48,6
Cortinas Morizadas	36,9
Fechadura Biométrica	29,7
Aspiração central	8,1
Irrigação automatizada	7,7

Fonte: Revista Lumiere Electric edição 240, adaptado pelo Autor, 2020

### 2.1.3 Vantagens da Automação Residencial

A expansão da automação residencial está relacionada com as inúmeras vantagens que a mesma possibilita, além da sua constante modernização destaca-se a acessibilidade, conforto e eficiência energética, proporcionando um conceito cada vez mais moderno de moradia. Esses pontos proporcionam uma maior aceitação entre os clientes divulgando o mercado da Domótica (PRUDENTE, 2011).

Neste contexto Prudente (2011), cita as principais vantagens da Domótica em relação a tecnologia convencional:

- a) Maior conforto – Por meio da Domótica o ambiente torna-se mais acolhedor e agradável, pois é possível controlar os parâmetros que favorecem uma boa qualidade de vida;
- b) Maior segurança – Essa Domótica possibilita aumentar o nível de segurança, pois permite controlar o acesso a residência, detectando tentativas de furto e até mesmo casos de incêndio ou vazamento de gás;
- c) Maior versatilidade – É possível alterar a configuração das instalações e acompanhar suas funções através de um software;
- d) Maior economia na gestão da instalação – Por meio desta tecnologia é possível controlar o consumo (iluminação, aquecimento, ar condicionado e etc.), permitindo uma economia no custo de energia da residência.

### 2.1.4 Arquitetura dos sistemas de Automação

Para prover tais benefícios e uma experiência ideal de acordo com as necessidades, hábitos e gostos de cada usuário, a Domótica integra diversos equipamentos trabalhando em conjunto, variando de simples sensores até complexas centrais de automação. Para o desenvolvimento de tal tecnologia, o sistema de automação conta com um dispositivo controlador interligado a sensores, atuadores e interface de comunicação. (ACCARDI; DODONOV, 2012).

- a) Controlador – É responsável por monitorar e controlar os dispositivos automatizados (atuadores e sensores), bem como realizar a comunicação com as interfaces, ou seja, coordenam todos os aparelhos e equipamentos da residência que fazem parte da automação;

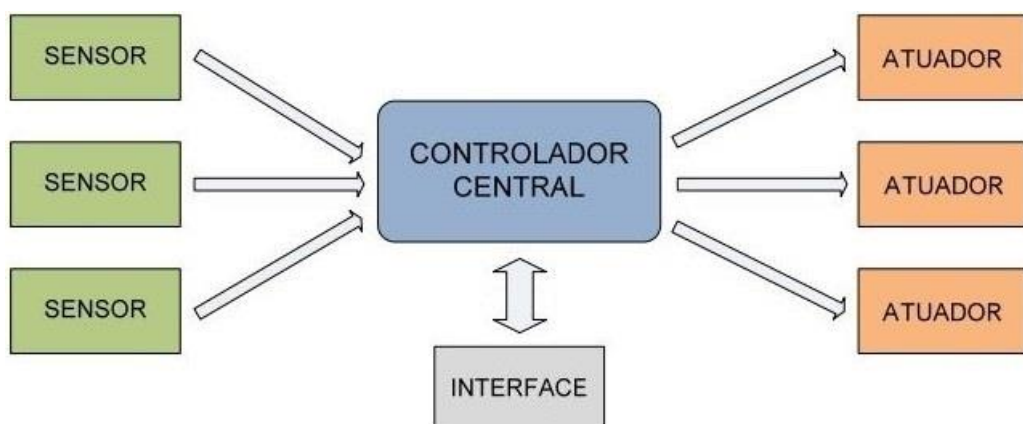
- b) Sensores – Detectam informações físicas do ambiente como: temperatura, luminosidade e umidade, convertendo as mesmas em um valor matemático que é transmitido para o controlador;
- c) Atuadores - São módulos eletromecânicos ligados entre a rede elétrica e os controladores responsáveis por acionar os equipamentos elétricos após receber o comando do controlador;
- d) Interfaces de Comunicação - As interfaces, tais como: celulares, tablets, controle remoto e interruptor, são os dispositivos que permitem ao usuário enviar comandos e interagir com o sistema de automação.

Segundo Ferreira (2008), tanto os sistemas de automação residencial, como os prediais voltados para hotéis e centros comerciais, visam uma melhoria na interação e comunicação destes dispositivos para com o usuário, porém para atender a tais requisitos, a complexidade e a arquitetura de cada sistema varia com base no local onde se encontra tal inteligência.

Neste contexto Accardi e Dodonov (2012) citam: “A arquitetura da automação residencial está relacionada com a forma com que seus elementos básicos se comunicam. As arquiteturas mais utilizadas são a centralizada e a descentralizada.”

No sistema com arquitetura centralizada, ilustrado na Figura 2, todos os dispositivos do sistema estão conectados a um controlador central, que de acordo com sua programação é responsável por receber e tratar as informações dos sensores e enviar os comandos para os atuadores. (FERREIRA, 2008).

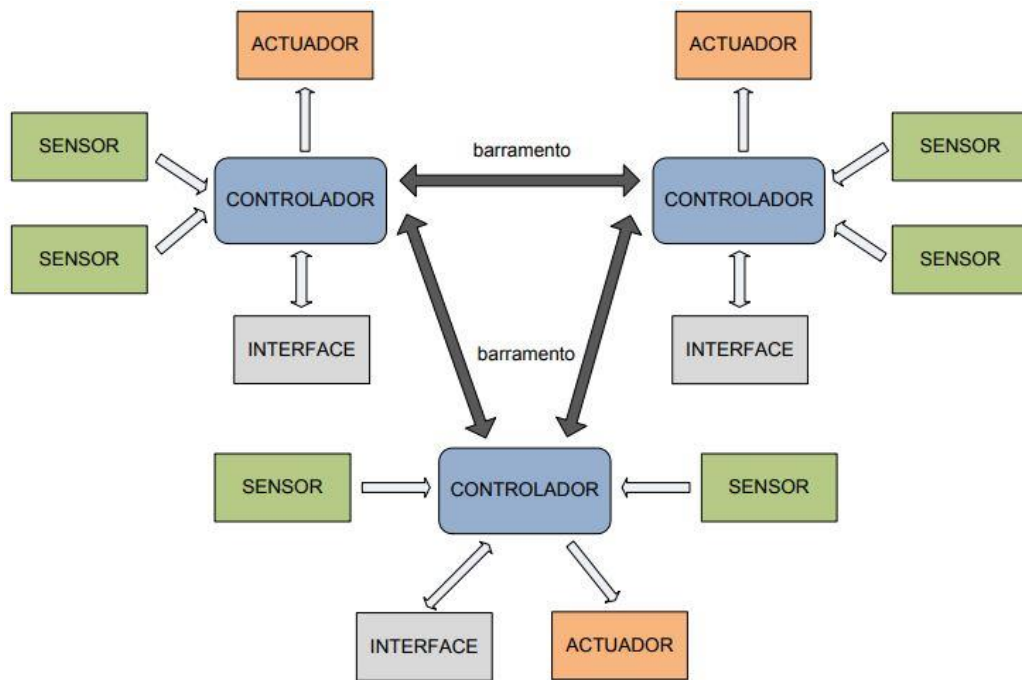
Figura 2 – Arquitetura centralizada.



Fonte: Interface homem-máquina para Domótica baseado em tecnologias Web, Ferreira, 2008.

Em sistemas com arquitetura descentralizada como mostra a Figura 3, podem existir vários controladores distribuídos interligados entre si, ambos compartilhando informações obtidas dos sensores e atuadores por meio de barramentos de dados e informações. (FERREIRA 2008)

Figura 3 – Arquitetura descentralizada



Fonte: Interface homem-máquina para Domótica baseado em tecnologias Web, Ferreira, 2008.

## 2.2 Sistemas Embarcados

A Domótica teve um grande avanço com a chegada dos microprocessadores e a criação de sensores e atuadores, pois a implementação de tais equipamentos possibilitaram a interação com outros dispositivos que necessitavam de menos manutenção, ocasionando uma redução de espaço utilizado pelo sistema de automação juntamente com uma melhor qualidade de controle. (BOLZANI, 2004).

### 2.2.1 Microprocessador

Um microprocessador ou CPU (Unidade central de processamento) é um dispositivo lógico programável construído sob a tecnologia VLSI (*Very-large-scale integration*), a qual combina um grande número de transistores em um único chip de silício. O Microprocessador é

considerado o componente mais importante em um computador ou dispositivo com sistema embarcado, sua função é processar e executar as tomadas de decisões responsáveis por controlar o sistema, bem como processamento de dados e informações.

Segundo Lenz (2019), o microprocessador deve ser dinâmico o suficiente para processar os dados recebidos e executá-los de maneira aleatória de acordo com as sequências de instruções definidas pelo programador.

“Todo microprocessador possui uma capacidade de processamento estabelecida pelo fabricante de acordo com a quantidade de operações (instruções) que o microprocessador é capaz de realizar. Uma instrução representa uma única operação que o microprocessador pode executar por vez. Essas operações podem ser classificadas como sendo de transferência de dados de um lugar para outro.” (GIMENEZ 2010).

Antes da invenção do microprocessador, era necessário construir um circuito de controle específico para cada sistema, eventuais modificações no circuito tornavam-se inviáveis, pois os valores eram muito elevados devido à alta complexidade destes sistemas. Sua utilização possibilitou controlar máquinas a partir de um programa específico, o qual pode ser facilmente modificado para atender futuras necessidades. (PEREIRA, 2007).

O primeiro microprocessador surgiu no início da década de 70, quando a *Nippon Calculating Machine* solicitou a Intel o desenvolvimento de 12 chips personalizados para sua nova calculadora *Busicom 141-PF.*, porém seus engenheiros sugeriram e projetaram um conjunto de 4 chips, conhecidos como MCS-4. Esta família era composta de uma unidade de processamento central (CPU) o Intel i4004, dois chips destinados para memória (ROM e RAM), e outro chip para entradas e saídas de dados e informações. (INTEL, 2011).

Com base na arquitetura de um sistema composto por microprocessador e periféricos, foi projetado um dispositivo que integrava fisicamente em uma única placa o microprocessador, memórias RAM e ROM e unidade de entrada/saída de dados dando origem aos Microcontroladores. (MARTINS, 2005).

### 2.2.2 Microcontroladores

Os microcontroladores são basicamente circuitos integrados utilizados em aplicações de sistemas digitais, seus recursos internos são capazes de executar programas que possibilitam a leitura e escrita digital na forma de valores de tensão em suas portas, o que permite a comunicação com diversos periféricos externos permitindo a expansão de suas capacidades, tendo assim inúmeras aplicações. Atualmente no mercado existe um grande número de

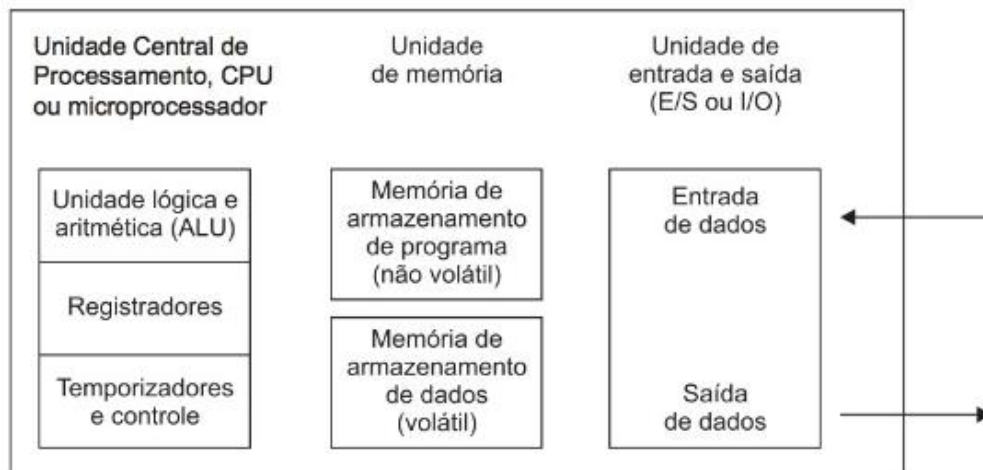
fabricantes com diversas especificações e características, devido a isto torna-se fundamental conhecer seus princípios de funcionamento para aplicações específicas. (FIOZERA, 2017).

“Em nosso cotidiano utilizamos muitos equipamentos que possuem internamente um microcontrolador, como eletroeletrônicos e eletrodomésticos. O timer de um aparelho micro-ondas, o controle remoto de um aparelho televisor ou ar condicionado, um relógio digital, o controlador de voo de um Drone, uma impressora 3D e muitos outros dispositivos podem ser construídos fazendo uso de Microcontroladores.” (FIOZERA, 2017).

“Os Microcontroladores, ao contrário de seus irmãos microprocessadores, são dispositivos mais simples, com memórias RAM e ROM internas, oscilador interno de *clock*, I/O interno, entre outros, sendo por isso chamados de computadores em um único chip.” (PEREIRA, 2007).

Segundo Gimenez (2010), os microcontroladores são dispositivos eletrônicos que possuem uma grande velocidade de processamento de dados e informações, criados com o objetivo de repetir e executar automaticamente tarefas do ser humano. Basicamente sua estrutura é formada por três blocos básicos, ilustrados na Figura 4. Unidade central de processamento (CPU), unidade de memória e dispositivos de entrada e saída de dados.

Figura 4 - Blocos básicos de um microcomputador



Fonte: Microcontroladores 8051 - Conceitos, Operação, Fluxogramas e Programação.

A Unidade Central de processamento também conhecida como microprocessador é considerada o cérebro do microcontrolador, constituída por milhões de transistores incorpora em um único circuito: registradores, somadores, comparadores, temporizadores e etc. Sua estrutura se divide em três componentes principais: (GIMENEZ, 2010).

- a) Unidade Lógica Aritmética – também conhecida como Arithmetic Logic Unit (ALU) é responsável por realizar as operações lógicas (E, OU, XOR, Inversão) e aritméticas do CPU, controlar a entrada e saída de dados do sistema e definir a ordem e o tempo das instruções por meio de cálculos matemáticos complexos);
- b) Registradores – Basicamente é a memória interna que o CPU utiliza para armazenamento temporário para processar dados, os microcontroladores atuais possuem entre 16 e 32 bits, quanto maior for a quantidade de bits processados em paralelo por vez, maior será sua capacidade de processamento;
- c) Unidade de Controle - A Unidade de Controle ao receber uma instrução é responsável por decodificar e enviar os sinais de controle para onde for necessário, basicamente sua função é ler um código em binário e interpretar a operação relativa a esse código, controlando o fluxo de dados do microprocessador para as unidades de memória e para as unidades de entrada e saída.

A unidade de memória tem como função o armazenamento interno das instruções e dados a serem manipuladas pelo microcontroladores, basicamente essas memórias se dividem em dois tipos: (MIYADAIRA, 2013).

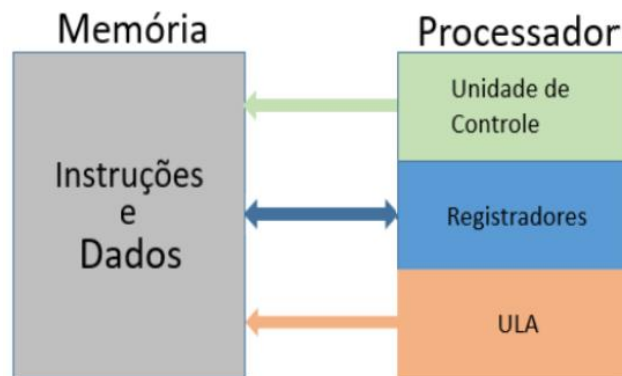
- a) Memória Volátil - As memórias voláteis ou também conhecidas como memórias RAM do inglês (*Random Access Memory*), são responsáveis pela leitura e escrita dos dados do sistema a ser controlado, essas informações são gerenciadas pelo programa para controlar as entradas e saídas de dados, neste tipo de memória os dados são perdidos sempre que o dispositivo for desenergizado, deste modo as variáveis devem ser carregadas sempre que o sistema for iniciado.
- b) Memória Não Volátil – As memórias não voláteis ou memórias ROM do inglês (*Read Only Memory*), são responsáveis pelo armazenamento do programa, neste tipo de memória as informações não são perdidas quando há falta de energia, para gravar uma memória deste tipo são necessários equipamentos específicos.

Os dispositivos de entrada e saída de dados (E/S ou I/O, do inglês *Input/Output*), são responsáveis por realizar a comunicação do sistema com o meio externo, transferindo informações do microcontroladores para os periféricos e assim vice-versa. (GIMENEZ, 2010).

### 2.2.2.1 Arquitetura Von Neumann e Harvard

Na arquitetura de Von Neumann representada na Figura 5, possui um único barramento comum para dados e instruções do sistema, ou seja, o programa e os dados são armazenados na mesma memória, deste modo é impossível que o processador acesse uma instrução e um dado ao mesmo tempo ocasionando lentidão no sistema. (OLIVEIRA, 2018).

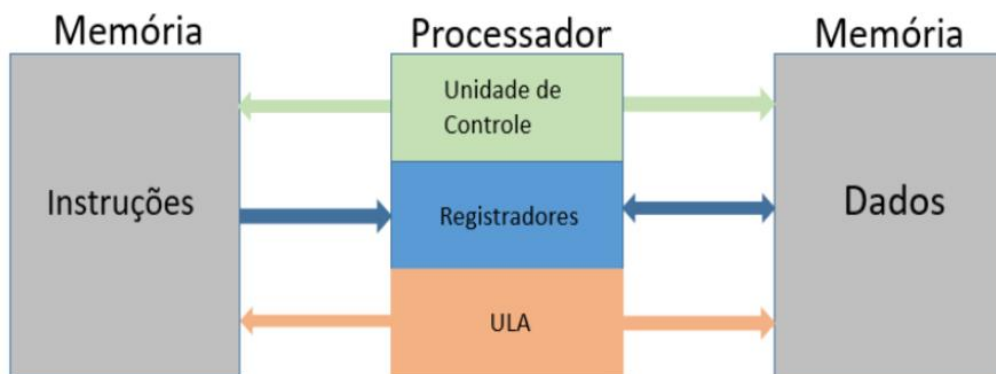
Figura 5 - Arquitetura Von Neumann



Fonte: © 2018 Diego Macêdo, adaptado pelo Autor, 2020.

A arquitetura Harvard representada na Figura 6 possui as memórias de dados e instruções ligadas ao processador por meio de barramentos separados, visando assim um melhor desempenho do microcontrolador, deste modo enquanto a CPU processa uma informação, outra nova informação está sendo buscada de forma sucessiva, aumentando consideravelmente a performance do sistema em termos de velocidade. (OLIVEIRA, 2018).

Figura 6 - Arquitetura Harvard



Fonte: © 2018 Diego Macêdo, adaptado pelo Autor, 2020.

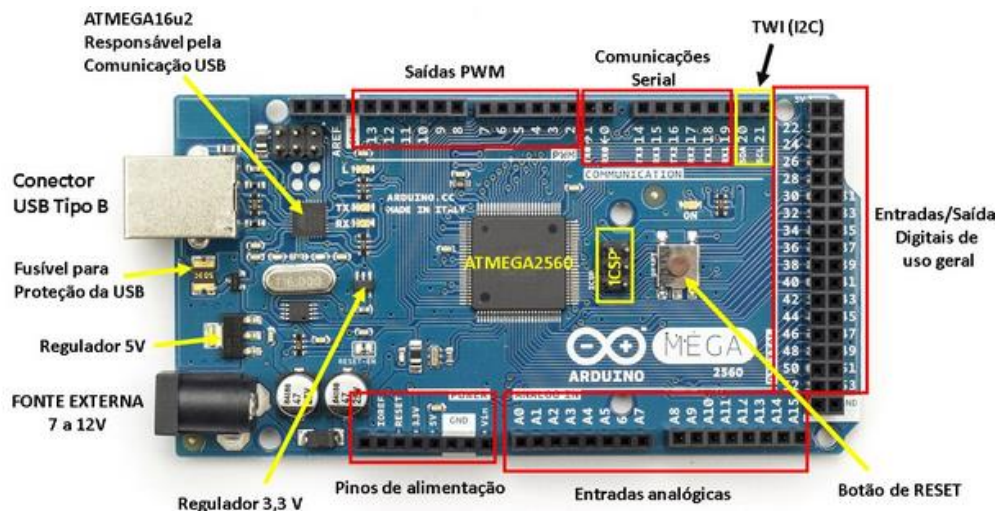
### 2.2.3 ARDUINO

O Arduino é uma plataforma de desenvolvimento totalmente de código aberto, destinada para todos que possuem interesse em projetar sistemas eletrônicos de forma fácil. Sua estrutura e programação permitem de forma simples monitorar e controlar componentes eletrônicos como sensores e atuadores, além de possibilitar adicionar *Shields* para ampliar suas funcionalidades de acordo com o projeto.

O primeiro microcontrolador Arduino foi introduzido no Mercado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Originalmente iniciado no ano de 2000 o objetivo da pesquisa era elaborar um dispositivo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas que não possuíam experiência anterior em eletrônica ou programação de microcontroladores. Atualmente o Arduino é considerado um dos principais dispositivos de hardware e código livre do mundo, pois qualquer um pode montar, modificar, melhorar e personalizar o mesmo. (BEGHINI, 2013 apud DAL AGNOL 2018).

Atualmente no mercado existem diversos modelos de Arduino, com diferentes configurações, número de portas de entrada e saída de dados, tipos de conexões e velocidade de processamento são um dos critérios de escolha do modelo ideal para o protótipo. Dentre os principais modelos em relação custo/benefício destaca-se o Arduino Mega2560 (Figura 7). (THOMSEN, 2014).

Figura 7 - Arduino Mega 2560



Fonte: © Arduino, adaptado pelo Autor, 2020.

O Quadro 2 apresenta as principais especificações técnicas do microcontrolador Arduino MEGA2560, de acordo com o site do fabricante.

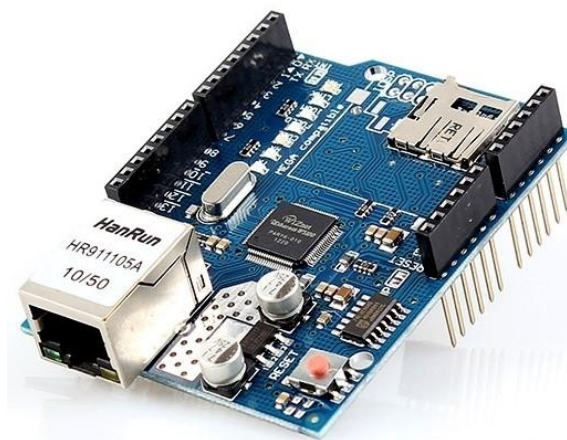
Quadro 2 - Especificações Arduino MEGA2560

Nome comercial	Arduino MEGA2560
Alimentação	5 V
Entrada	7 ~ 12 V
Consumo	Média de 70 Ma
Temperatura	-40 ~ +85° C
Processador	ATmega2560
Frequência de Operação	0 ~ 16 MHz
Memória FLASH	256 KB
Memória RAM	8 KB
Memória ROM	4 KB
Pinos de I/O	54 pinos com 14 PWM
WiFi	Somente com Shield
Bluetooth	Não Possui
Preço Médio	R\$ 99,99

Fonte: © Arduino, adaptado pelo Autor, 2020.

Muitos projetos desenvolvidos com Arduino necessitam de acesso à internet para serem controlados remotamente, para suprir esta necessidade foi desenvolvido o *Ethernet Shield* (Figura 8). Este dispositivo permite ao Arduino se conectar a uma rede local ou a internet através de um cabo de rede RJ45, há também um slot para cartões micro-SD, usado para armazenar arquivos que vão ser utilizados na rede (ARDUINO, 2020)

Figura 8 - Ethernet Shield



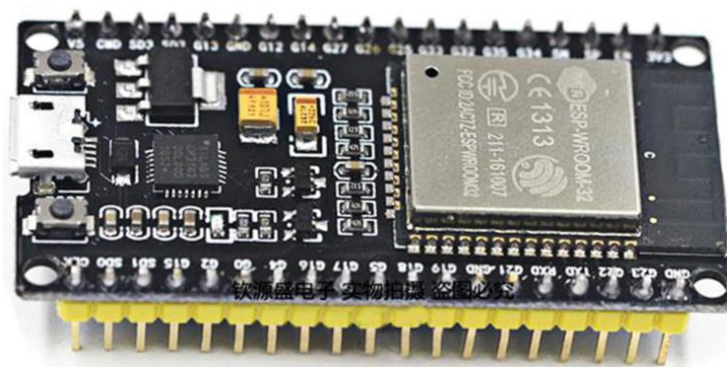
Fonte: © Arduino, adaptado pelo Autor, 2020.

Para desenvolver programas e gravá-los no Arduino é necessário utilizar o IDE (Ambiente de Desenvolvimento Integrado), disponível gratuitamente para download no site oficial do fabricante. Esta plataforma possibilita escrever códigos de programação de maneira rápida e eficiente, pois a mesma conta com diversas bibliotecas disponíveis para download a fim de simplificar a programação. Após o código estar pronto, basta conectar o Arduino no computador, escolher o modelo a ser utilizado, a porta na qual o mesmo está conectado e carregar o código. (THOMSEN, 2014).

#### 2.2.4 ESP 32

O microcontrolador ESP 32 apresentado na Figura 9, tem sido cada vez mais utilizado para fins acadêmicos e industriais, especialmente com aplicações que necessitam de um acompanhamento em tempo real, aliada a uma combinação de baixo custo e alto desempenho quando comparados a outros Microcontroladores disponíveis no mercado. (ALLAFI; IQBAL, 2017 apud DAL AGNOL 2018).

Figura 9 - ESP 32



Fonte: ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2019.

Lançado em setembro de 2016 pela *Espressif* Sistemas, o ESP 32 demonstra ser mais versátil do que seu antecessor o ESP8266 desenvolvido em 2015, pois além do clássico módulo de comunicação Wi-Fi o modelo atual possui um sistema com processador Dual Core, Bluetooth 4.2 e múltiplos sensores embutidos em seu chip, o que facilita muito em projetos com aplicações diversas. (ALMEIDA, 2018).

O quadro 3 apresenta as principais especificações técnicas do Microcontrolador Esp32, de acordo com o site do fabricante.

Quadro 3 – Especificações ESP 32

Nome comercial	ESP 32
Alimentação	2,2 ~ 3,3 V
Entrada	5 ~ 9 V
Consumo	Média de 80 mA
Temperatura	-40 ~ +85° C
Processador	Xtensa® Dual-Core 32-bit LX6
Frequência de Operação	80 ~ 240 MHz
Memória FLASH	4 MB
Memória RAM	520 KB
Memória ROM	448 KB
Pinos de I/O	34 pinos com 16 PWM
WiFi	2,4 GHz até 150 Mbps
Bluetooth	Bluetooth Low Energy v4.2
Preço Médio	R\$ 52,00

Fonte: ESPRESSIF SYSTEMS (SHANGHAI), adaptado pelo Autor, 2020.

Assim como seu concorrente Arduino, o ESP32 é um chip de código aberto e sua fabricante *Espressif* possui o ESP-IDF (*Espressif IoT Development Framework*), um conjunto de programas, compiladores, bibliotecas dentre outros arquivos que é preciso para o desenvolvimento de uma aplicação prática. Esse framework pode ser integrado em plataformas de programação como o Eclipse que é um ambiente versátil e poderoso, muito utilizado por fabricantes de sistema embarcados. Outra forma de programar o ESP32 é por meio da IDE do Arduino, dessa forma é possível trabalhar em um ambiente mais simples, bastante conhecido por quem já programa em C / C++. (CURVELLO, 2018)

## 2.3 Sensores e Atuadores

Em uma residência existem diferentes processos que podem ser automatizados, para isto o sistema de automação conta seus respectivos dispositivos auxiliares (sensores e atuadores). Os sensores transformam variáveis físicas como temperatura e umidade em sinais elétricos, e os atuadores são dispositivos eletromecânicos que alteram suas características conforme os impulsos elétricos recebidos pelo controlador. (BOLZANI, 2004).

### 2.3.1 Sensores

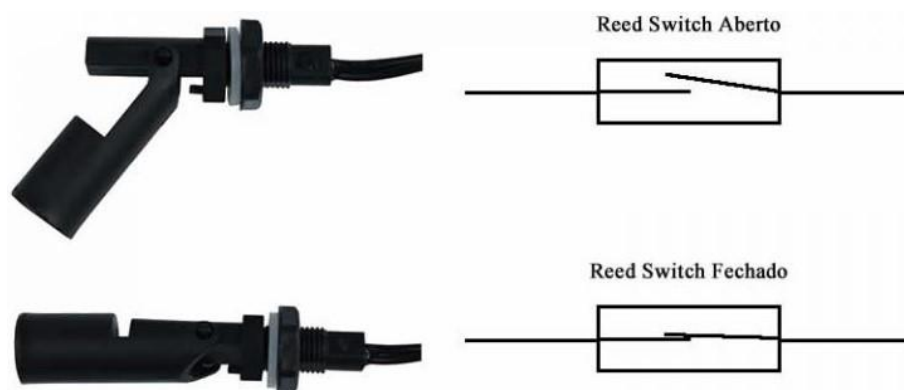
Basicamente os sensores são dispositivos destinados a detectar grandezas físicas do ambiente em que estão instalados, convertendo essas informações obtidas em sinais elétricos enviados ao controlador do sistema por meio de barramentos de comunicação. Alguns exemplos

das variáveis físicas que podem ser medidas são: níveis de reservatórios, velocidade, temperatura e presença de pessoas ou/e objetos (SILVA, 2009).

### 2.3.1.1 Sensor de Nível

O Sensor de nível (Tipo Boia Horizontal) é um componente desenvolvido com a finalidade de detectar o nível de líquido em um recipiente, este sensor funciona como uma chave magnética variando seu contato para NA (Normalmente aberto) ou NF (Normalmente fechado), conforme ilustração na Figura 10. Na prática a partir do seu estado (NA/NF) uma informação é enviada para a plataforma microcontrolada responsável por executar as instruções definidas na programação, podendo por exemplo, acionar bombas d'água para preencher o volume de um determinado reservatório ou simplesmente acionar dispositivos sinalizadores para aviso (USINAINFO, 2020).

Figura 10 - Sensor de Nível (Tipo Boia Horizontal)



Fonte: ROBOCORE®, 2020.

### 2.3.1.2 Sensor de Presença

O sensor de presença é um dispositivo muito utilizado em garagens, escadas, corredores, banheiros e outros locais de acesso comum em residências e condomínios, os sensores mais utilizados para tal finalidade são os sensores de infravermelho, esses sensores detectam a presença de pessoas ou objetos por meio de fonte de calor, deste modo, quando há movimentação o sensor infravermelho realiza a detecção de presença através da distinção dos valores de temperatura, acionando o dispositivo no momento em que o objeto se alinha com o mesmo. (MORAIS, 2017).

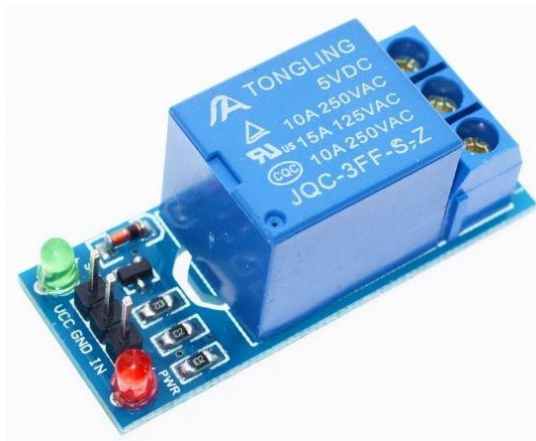
### 2.3.2 Atuadores

Os atuadores são componentes que recebem um estímulo que pode ser elétrico, hidráulico ou pneumático através de um microcontrolador que têm a capacidade de converter em movimentos mecânicos, sonoros ou luminosos (DUTRA; ROMANO, 2016)

#### 2.3.2.1 Módulo Relé

Os Microcontroladores tais como Arduino e Esp32 mencionados anteriormente, funcionam com tensões e corrente relativamente baixas, devido a isto suas portas de saídas fornecem uma tensão de no máximo 5 V com uma corrente na casa de mA, devido a isto, para acionar dispositivos com valores de corrente e tensões maiores, se faz necessário o uso de algum dispositivo capaz de suportar tal carga, o módulo relé demonstrado na Figura 11 é uma forma de controlar lâmpadas, motores e outras cargas com potências diferentes do circuito de comando.

Figura 11 - Módulo relé simples



Fonte: ROBOCORE®, 2020.

Atualmente existem vários modelos de módulos relés com diversas quantidades de canais, indo até 16 relés em uma só placa, alguns modelos contam com Leds indicadores que mostram o estado atual do relé (ligado/desligado) em cada canal. Sua tenção de operação varia de 3,3 ~ 5 V DC, com uma corrente de operação de 15 ~ 20 mA, e uma capacidade de saída de 15 A /125 V e 10 A / 220 V.

### 3 METODOLOGIA

Neste capítulo, serão apresentados os métodos utilizados para o desenvolvimento do projeto com base no microcontrolador escolhido. Inicialmente foi realizado o levantamento do local onde o sistema será instalado futuramente, assim como, os sistemas a serem automatizados, os dispositivos necessários e os métodos de implementação para cada sistema.

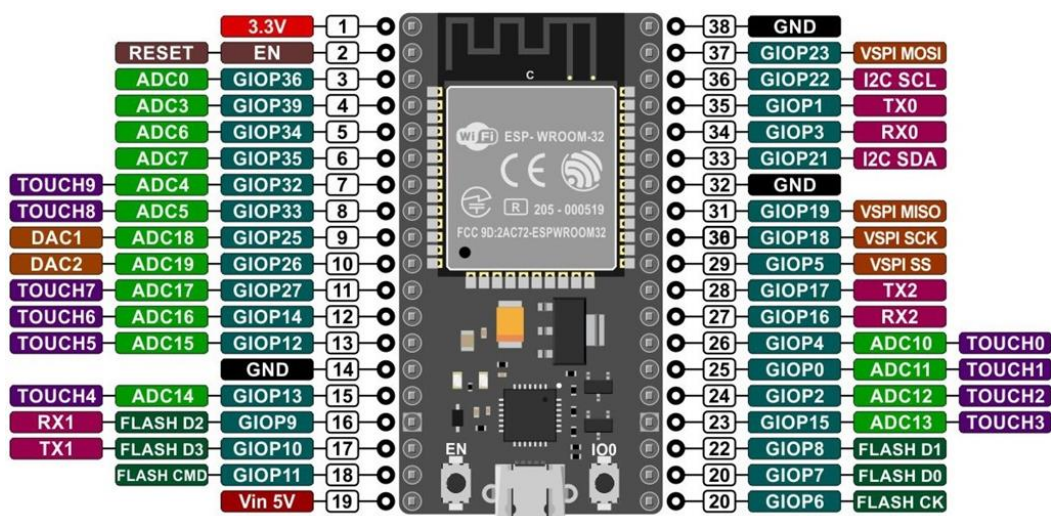
Em um segundo momento será apresentado a plataforma utilizada para programar o microcontrolador, juntamente com algumas definições importantes para a depuração do código. Posteriormente será apresentada a plataforma definida para o desenvolvimento do aplicativo e os métodos utilizados para efetuar a montagem do projeto.

#### 3.1 Definição do Microcontrolador

Para a definição do microcontrolador tomou-se como base o referencial teórico, analisando os recursos oferecidos por cada um dos Hardwares e optou-se pela utilização do ESP 32. Os principais critérios levados em consideração foram o fato do ESP32 possuir 2 núcleos de processamento, sua frequência de clock que pode chegar a 240MHz e a conectividade sem fio como Wi-Fi e Bluetooth que já vem integrados no chip, enquanto o Arduino Mega requer módulos externos para conexão Wi-Fi e Bluetooth.

A Figura 12 apresenta o *datasheet* do NodeMCU-32, modelo escolhido para o desenvolvimento do trabalho.

Figura 12 - *datasheet* do NodeMCU-32



Fonte: ROBOCORE®, adaptado pelo Autor, 2020.

## 3.2 Levantamento de Dados e Variáveis a Serem Controladas

Com base no referencial teórico é importante ressaltar que um dos principais objetivos deste trabalho foi desenvolver um sistema de automação com um custo reduzido quando comparado aos demais sistemas disponíveis no mercado da automação, dado que, atualmente é possível automatizar quase todos os dispositivos eletrônicos existentes em uma residência, entretanto sua aplicação e o nível da automação é limitado as preferencias e orçamentos de cada cliente.

Deste modo para o desenvolvimento do projeto tomou-se como base uma residência pequena (Anexo A), com 5 cômodos internos, os quais estão dispostos em: sala de estar, cozinha, banheiro, lavanderia e um quarto, no espaço externo a casa possui uma garagem também utilizada como salão de festas.

A aplicação do sistema de automação proposto para esta residência tem como base as principais vantagens da automação citadas no capítulo 2.1.3: maior conforto, segurança e eficiência energética. Neste contexto os dispositivos propostos a serem automatizados no projeto são:

- Controle de Iluminação;
- Monitoramento do nível do reservatório de água com o acionamento automático da bomba;
- Monitoramento de temperatura;
- Sistema de Segurança;

### 3.2.1 Iluminação

Um dos principais objetivos do projeto é o controle e monitoramento do sistema de iluminação da residência por meio de um aplicativo desenvolvido para *smartphone*. Para tal finalidade basta selecionar o comando (Liga ou Desliga) que o aplicativo envia um sinal para o microcontrolador ESP 32, que sequencialmente atua sobre o relé diretamente conectado com a lâmpada.

O Quadro 4 destaca os cômodos da residência automatizados, bem como os respectivos elementos controlados.

Quadro 4 – Cômodos e Dispositivos automatizados

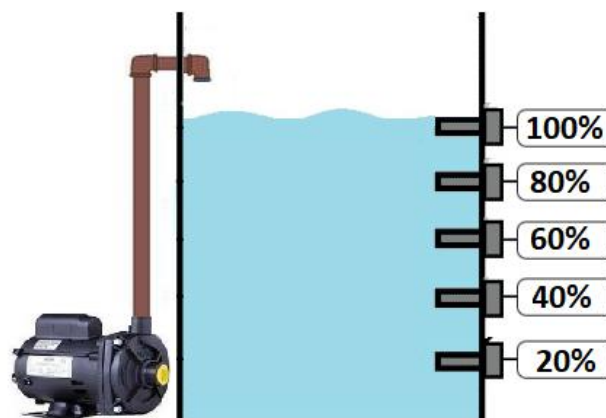
<b>Cômodo</b>	<b>Objeto de controle</b>
Sala de estar	1 Lâmpada central 1 Fita Led (Sanca de Gesso)
Banheiro	1 Lâmpada Led Central
Quarto 1	1 Lâmpada Led Central 2 Spot Led (Escrivaninha)
Cozinha	1 Lâmpada Led Central
Lavanderia	1 Lâmpada Led Central
Garagem	1 Lâmpada Led Central

Fonte: Elaborado pelo Autor, 2020.

### 3.2.2 Monitoramento de Nível da Caixa D'água

O processo de monitoramento do nível da caixa d'água é composto por 5 sensores de nível (Tipo Boia Horizontal), instalados dentro do reservatório responsável pelo abastecimento de água da residência cada um dos sensores indica os respectivos níveis: 20, 40, 60, 80 e 100 por cento, ilustrados na Figura 13. Tomando como base a eficiência energética nas aplicações do sistema de controle de nível em residências e edifícios, o sistema possui acionamento automático da bomba d'água responsável por encher o reservatório.

Figura 13 – Esquemático do sistema de monitoramento de nível



Fonte: Elaborado pelo Autor, 2020.

O acionamento automático da bomba responsável por encher o reservatório de água, é dado por meio da medição das variáveis de nível, de tal modo que a bomba é acionada sempre

que o nível de água estiver abaixo de 60 por cento e permanecerá acionada até que o sensor que indica 100 por cento do nível seja acionado.

A Tabela 1, apresenta a Tabela Verdade da lógica utilizada para desenvolver a programação do monitoramento de nível do reservatório e o acionamento do relé responsável por Ligar e Desligar a motobomba.

Tabela 1 – Tabela Verdade do Acionamento da Motobomba

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Nível (%)	Comando
0	0	0	0	0	0	
1	0	0	0	0	20	
1	1	0	0	0	40	Ligar Bomba
1	1	1	0	0	60	
1	1	1	1	0	80	
1	1	1	1	1	100	Desligar Bomba

Fonte: Elaborado pelo Autor, 2020.

### 3.2.3 Monitoramento de Temperatura

Para o monitoramento de temperatura foi utilizado o sensor DS18B20 a prova d'água (Figura 14), um dos mais usados em projetos que envolvem medição de temperaturas em ambientes úmidos, pois o sensor é revestido por um material à prova d'água, com sua ponta encapsulada em aço inoxidável. Além disto o sensor pode efetuar leituras com precisão de até  $\pm 0,5$  °C, com uma faixa de medição de temperatura de -55 a +125 °C.

Figura 14 – Sensor de temperatura DS18B20

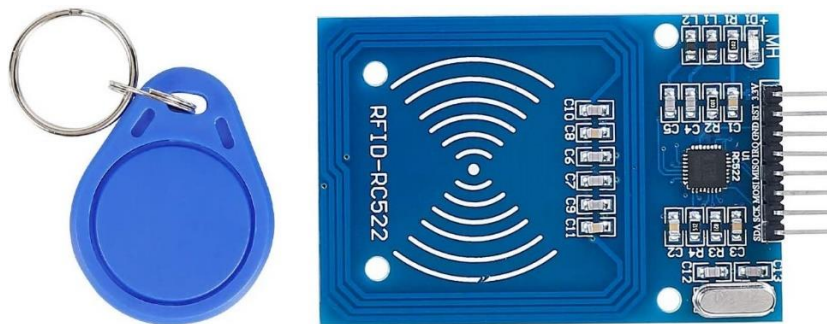


Fonte: ROBOCORE®, 2020.

### 3.2.4 Sistema de Segurança

Para a segurança externa da residência o sistema dispõe de um controle de acesso responsável pela abertura do portão principal, para tal finalidade foi utilizado um módulo RFID-RC522 demonstrado na Figura 15, que identifica automaticamente *tags* e cartão de acesso. Para liberar a fechadura eletrônica do portão as *tags* de acesso são previamente configuradas e suas informações salvas no código fonte do microcontrolador, deste modo, após a identificação de uma *tag* ou cartão cadastrados um relé é acionado liberando a fechadura eletrônica do portão.

Figura 15 - Módulo RFID-RC522



Fonte: ROBOCORE®, 2020.

No interior do ambiente foi implementado um sistema de monitoramento utilizando dois sensores de presença PIR - HC-SR50 (Figura 16), acionados através da detecção de calor incomum ao local escolhido, este sensor é capaz de cobrir uma área de até 7 metros com um ângulo de abertura de aproximadamente 120°.

Figura 16 - Sensor de Presença PIR - HC-SR50



Fonte: ROBOCORE®, 2020.

### 3.3 Desenvolvimento do Projeto

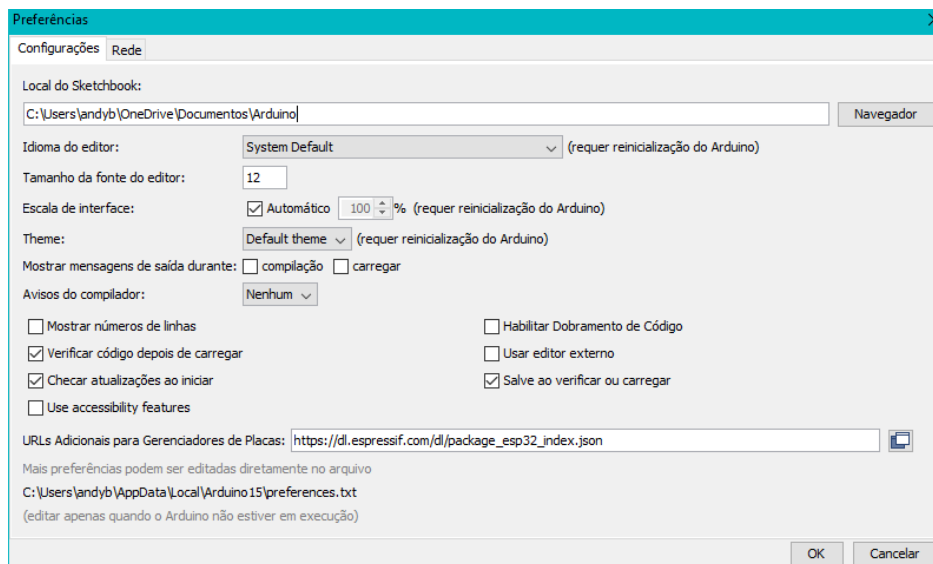
A implementação do projeto pode ser descrita em três etapas: depuração do código do Microcontrolador; Desenvolvimento do aplicativo e Testes de bancada. Tais etapas caracterizam, respectivamente, as subseções a seguir.

#### 3.3.1 Programação do ESP32

A programação do ESP 32 foi realizada por meio do programa IDE do Arduino, porém para realizar a gravação do código no ESP 32 é necessário fazer algumas alterações nas “preferências” do software, deste modo, foi preciso realizar o download de uma biblioteca adicional a “esp32” específica para o seu correto funcionamento.

Para adicionar esta biblioteca, basta executar a IDE do Arduino, clicar em **Arquivo > Preferências**, inserir o URL ([https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)) indicado na figura 17, e clicar em ok.

Figura 17 – Instalação da biblioteca ESP 32



Fonte: Elaborada pelo Autor, 2020.

Após realizar estas alterações, partiu-se para a depuração do código fonte responsável por gerenciar o sistema, diante disso, visando simplificar o desenvolvimento da programação foram definidas algumas bibliotecas a serem utilizadas tornando o código mais simples e organizado. Posteriormente com base no *datasheet* do NodeMCU-32 (Figura 13), foram definidos os pinos responsáveis por gerenciar cada dispositivo do sistema (Figura 18).

Figura 18 – Definição dos pinos a serem utilizados

```

Projeto_TCC
#include <WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>
// Dados para autenticação do WiFi
const char* ssid = "TCCAnderson";
const char* password = "tcc2020";
// Declaração dos pinos para cada função
int rele_1 = 23; // Luz Central Sala
int rele_2 = 22; // Luz Fita Led Sala
int rele_3 = 21; // Luz Principal Quarto
int rele_4 = 19; // Luz Escrivania Quarto
int rele_5 = 18; // Luz Cozinha
int rele_6 = 17; // Luz Lavanderia
int rele_7 = 16; // Luz Banheiro
int rele_8 = 15; // Luz Garagem
int alarme = 0;
const int oneWireBus = 33; // Sensor de Temperatura
int sensor_presenca_1 = 12; // Sensor de Presença 1
int sensor_presenca_2 = 14; // Sensor de Presença 2
int rele_sirene = 2; // Sirene do alarme
int sensor_nivel_1 = 26; // Sensor de nivel 1
int sensor_nivel_25 = 25; // Sensor de nivel 2
int sensor_nivel_50 = 35; // Sensor de nivel 3
int sensor_nivel_75 = 34; // Sensor de nivel 4
int sensor_nivel_100 = 39; // Sensor de nivel 5
int rele_bomba_agua = 32;

```

Fonte: Elaborado pelo Autor, 2020.

A Figura 19 apresenta os pinos configurados como entrada ou saída de dados do sistema, ambos definidos dentro do comando “void setup”.

Figura 19 – Definição dos pinos de entrada e saída de dados

```

void setup() {
  delay(1000);
  pinMode(rele_1, OUTPUT); //Define como Saida
  pinMode(rele_2, OUTPUT); //Define como Saida
  pinMode(rele_3, OUTPUT); //Define como Saida
  pinMode(rele_4, OUTPUT); //Define como Saida
  pinMode(rele_5, OUTPUT); //Define como Saida
  pinMode(rele_6, OUTPUT); //Define como Saida
  pinMode(rele_7, OUTPUT); //Define como Saida
  pinMode(rele_8, OUTPUT); //Define como Saida
  pinMode(rele_sirene, OUTPUT); //Define como Saida
  pinMode(sensor_nivel_1, INPUT); //Define como Entrada
  pinMode(sensor_nivel_50, INPUT); //Define como Entrada
  pinMode(sensor_nivel_75, INPUT); //Define como Entrada
  pinMode(sensor_nivel_100, INPUT); //Define como Entrada
  pinMode(rele_bomba_agua, OUTPUT); //Define como Saida
}

```

Fonte: Elaborado pelo Autor, 2020.

Para inicializar o servidor no microcontrolador, foi utilizada a biblioteca "Wifi.h", esta biblioteca permite conectar o ESP 32 a rede Wi-Fi utilizando o comando `WiFi.begin(ssid, pwd)`, no qual a identificação da rede (ssid) e a senha da rede (pwd) são constantes definidas no código do sistema.

Na Figura 20 é apresentando as informações citadas para conectar o ESP 32 a rede, bem como, a parte da programação responsável por criar um servidor no IP Address escolhido, que é utilizado para realizar a comunicação entre o aplicativo e o microcontrolador.

Figura 20 – Programação para realizar a conexão Wi-Fi

```
// Conectando à rede WiFi
Serial.println();
Serial.println();
Serial.print("Conectando com: ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi conectado com sucesso");

//Configurações do IP fixo, conforme a rede escolhida
IPAddress ip(192, 168, 0, 150);
IPAddress gateway(192, 168, 0, 254);
IPAddress subnet(255, 255, 255, 0);
Serial.print("Configurando IP fixo para : ");
Serial.println(ip);

//Envia a configuração
WiFi.config(ip, gateway, subnet);

//Inicializa o server criado na porta 80
server.begin();

//Mostra no monitor serial o IP que o ESP possui
Serial.print("Server em: ");
Serial.println(WiFi.localIP());
```

Fonte: Elaborado pelo Autor, 2020.

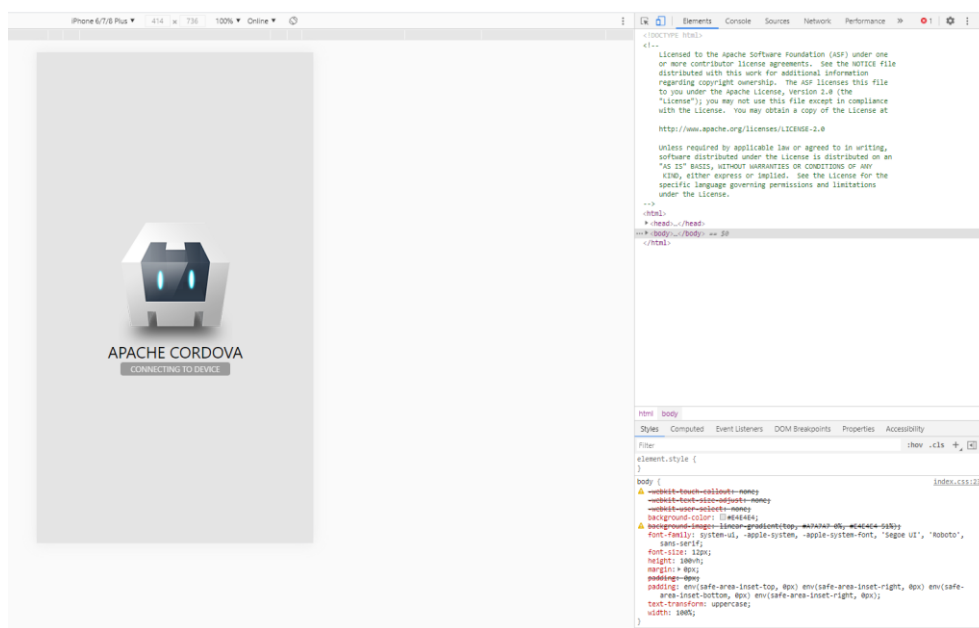
Dentro da função “void loop”, (Anexo B e Anexo C), foi desenvolvida a lógica de programação do sistema de iluminação, segurança, temperatura e o monitoramento de nível de água, pois como o nome sugere, esta função repete-se consecutivamente enquanto o microcontrolador estiver ligado, permitindo assim ao programa detectar as mudanças de estados dos sensores e atuadores.

### 3.3.2 Desenvolvimento do Aplicativo

Para desenvolver o aplicativo responsável por controlar o sistema, foram avaliadas duas opções de plataformas, a primeira foi o *MIT App Inventor*, um ambiente de programação para *smartphones* e *tablets* que utilizam o sistema operacional Android, e a segunda, o Apache Cordova que permite escrever aplicativos utilizando HTML, JavaScript, e CSS capaz de gerar versões nativas para plataformas como iOS e Android.

Entretanto devido ao fato do *MIT App Inventor* ser uma plataforma exclusiva do sistema Android, optou-se pela utilização do Apache Cordova (Figura 21), que permitir criar aplicações com base no Webview, ou seja, funciona como uma interface de navegador capaz de executar HTML, javascript, e CSS dentro da própria plataforma, porém toda a interface do navegador como barra de endereço e demais ícones são descartados, sobrando apenas um renderizador de HTML, o que torna o ambiente de desenvolvimento uma multiplataforma, podendo executá-lo em diversos sistemas, tais como: iOS e Android.

Figura 21 – Tela de desenvolvimento do APP



Fonte: Elaborado pelo Autor, 2020.

Visando uma interface mais intuitiva e de melhor interação para com os moradores, o controle do sistema de iluminação, nível de água, temperatura e alarme foram divididos em abas diferentes no aplicativo, as quais podem ser selecionadas em uma lista conforme ilustrado na Figura 22.

Para controlar o sistema utilizando o aplicativo é necessário conectar o *smartphone* e o microcontrolador a mesma rede Wi-Fi, para tal finalidade, uma quinta aba foi adicionado ao APP: a aba “Rede”.

Figura 22 – Lista de telas do aplicativo



Fonte - Elaborado pelo Autor, 2020.

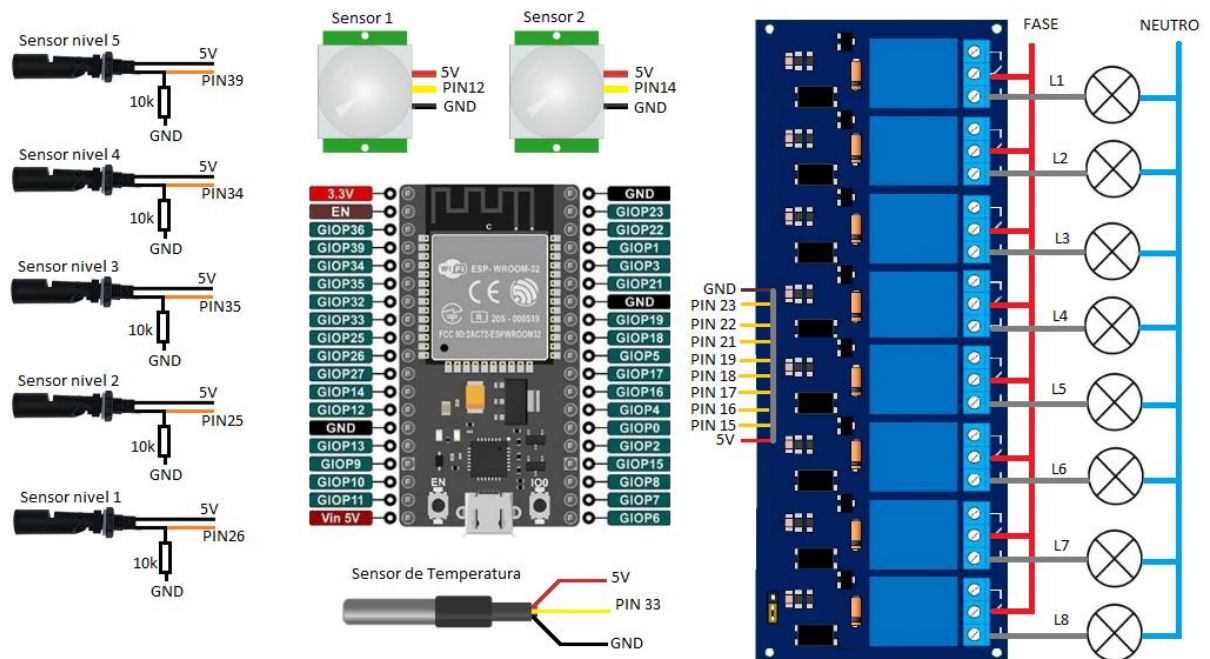
### 3.3.3 Testes de bancada

Após efetuar a depuração do código fonte do microcontrolador e concluir o desenvolvimento do aplicativo, partiu-se para os testes de bancada. Para tanto, o sistema de automação interno da residência e o controle de acesso do portão principal, foram realizados de formas distintas.

Deste modo, motivado pela necessidade de estruturar um código fonte capaz de conectar o microcontrolador a rede local mantendo uma comunicação estável entre o dispositivo e o aplicativo, inicialmente foram realizados os testes de conexão Wi-Fi, tomado como base as configurações de rede já existentes na residência.

Para evitar problemas de interferências ao instalar o sistema de automação interno da residência, a validação do protótipo consistiu em montá-lo em uma maquete, deste modo, a fim de mitigar possíveis erros de ligações e montagem, foi realizado um projeto prévio da conexão física de cada dispositivo utilizado no sistema, englobando em uma única central de automação o controle de iluminação, monitoramento do nível do reservatório, sensor de temperatura e alarme (Figura 23)

Figura 23 – Esquema de ligação proposto para a central de automação interna



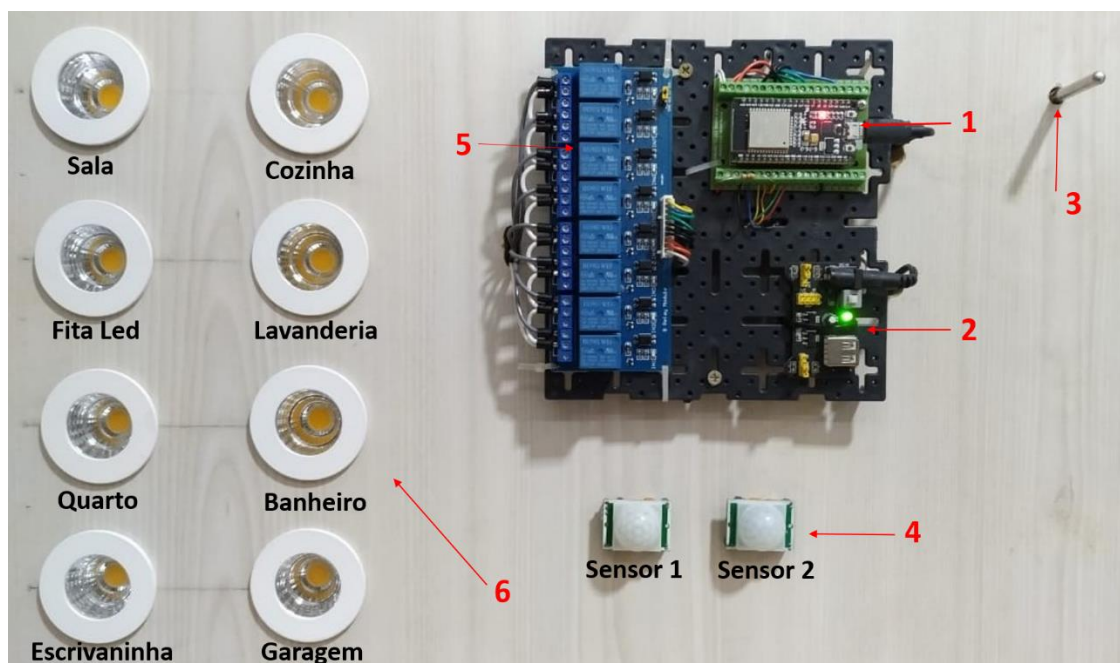
Fonte - Elaborado pelo Autor, 2020.

Para validar o sistema de controle de acesso foram adquiridas 4 tags/etiquetas, destas apenas duas foram validas para liberar o acesso a residência, pois como citado anteriormente no item 3.2.4 o módulo RFID conta com um sistema eletrônico que verifica as tags aproximadas e libera somente aquelas as quais estiverem previamente autorizadas no código raiz do programa.

Frente a isto, para monitorar se o acesso foi permitido ou negado foi utilizado um Display 16×2, deste modo, o programa exibe na tela mensagens referentes as tags utilizadas, neste caso, o display apresenta a mensagem de “Acesso Liberado” para as tags cadastradas, e “Acesso Negado” para as tags não cadastradas.

A Figura 24 exibe o protótipo montado em bancada com a indicação dos principais componentes do circuito, os quais são especificados na Tabela 2

Figura 24 – Montagem do Protótipo



Fonte - Elaborado pelo Autor, 2020.

Tabela 2 – Componentes do protótipo de bancada

Componente	Descrição
1	Microcontrolador ESP 32
2	Fonte Ajustável 3.3v 5v 1A
3	Sensor de Temperatura
4	Sensores de Presença
5	Módulo Relé de 8 canais 5V
6	Lâmpadas LED

Fonte - Elaborado pelo Autor, 2020.

## 4 RESULTADOS E DISCUSSÕES

O capítulo a seguir apresenta os principais resultados obtidos por intermédio da metodologia utilizada, expressa no capítulo 3.

Ademais, verifica-se a conexão entre o microcontrolador e o aplicativo por intermédio do servidor local criado. Em seguida, faz-se a análise dos testes de bancada do sistema a fim de validar a eficácia do projeto, apontando as soluções para os problemas encontrados e quais as atividades futuras que podem ser desenvolvidas a partir deste trabalho.

### 4.1 Conexão do Sistema

Na Figura 25 valida-se os testes de conexão do sistema apresentada na seção 3.3.1, que tem o objetivo de configurar o ESP 32 para se conectar à rede local, criando um servidor no endereço IPAddress definido. Após a criação do servidor, para se conectar ao mesmo, e controlar o sistema, na tela “Rede” do aplicativo, é inserido o IPAddress, e a porta de conexão utilizada, que neste trabalho foi definido como a 80 (Figura 26).

Figura 25 – Conexão do ESP 32 a rede local

```
Conectando com: TCCAnderson
.
WiFi conectado com sucesso
Configurando IP fixo para : 192.168.0.150
Server em: 192.168.0.150
```

Fonte - Elaborado pelo Autor, 2020.

Figura 26 – Dados para conexão a Rede local do ESP 32



The screenshot shows a mobile application interface with a blue header bar containing a menu icon and the text 'Automação Residencial'. Below the header, the text 'Insira a rede:' is displayed. There are two input fields: the first is labeled 'IP' and contains the value '192.168.0.150'; the second is labeled 'Porta' and contains the value '80'. A blue button labeled 'Salvar' is positioned below the input fields. The top status bar shows the time as 21:11 and various system icons.

Fonte - Elaborado pelo Autor, 2020.

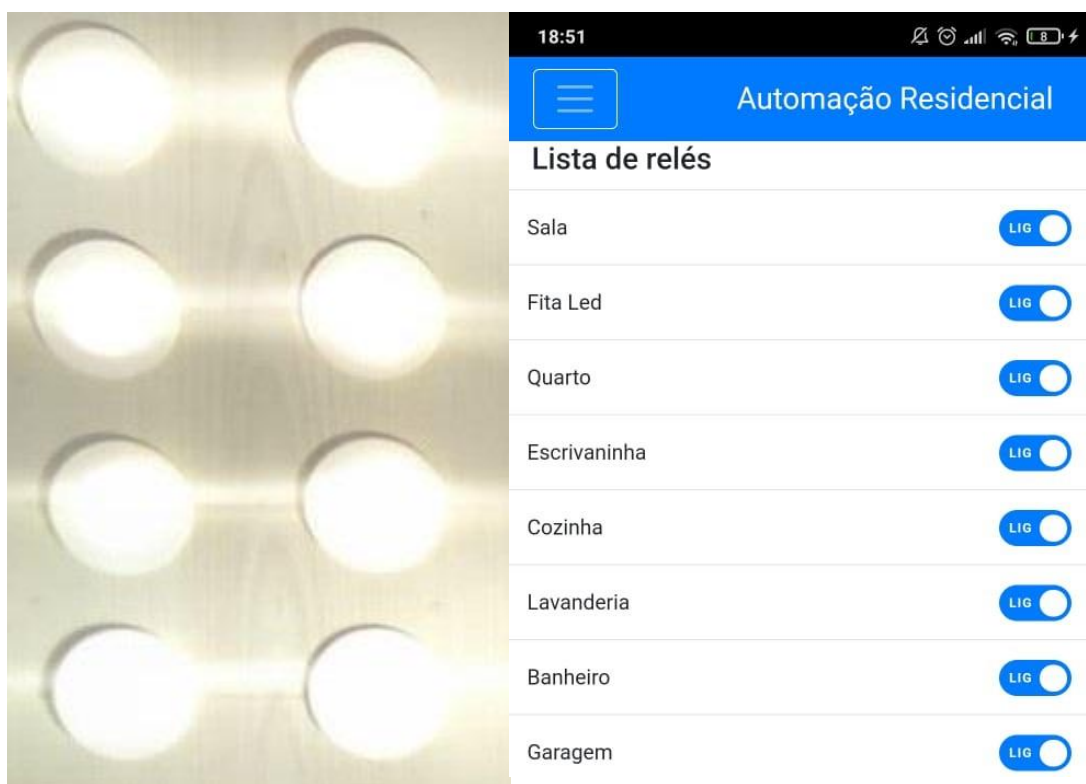
## 4.2 Validação do Sistema de Automação Interno da Residência

Nos testes de bancada realizados para o sistema de iluminação, as lâmpadas da maquete foram alimentadas com uma tensão de 220V, deste modo foi possível extrair resultados mais próximos de como o sistema se comportara após a implementação na residência.

A fim de identificar qualquer possível falha no sistema de iluminação, o acionamento das lâmpadas foi efetuado de diversas maneiras, sempre partindo de um cômodo aleatório para os demais.

A seguir na Figura 27, do lado esquerdo da imagem é possível observar o sistema de iluminação em teste, com as 8 lâmpadas ligadas, posteriormente do lado direito, observa-se a interface de interação do aplicativo, responsável pelo controle do sistema de iluminação.

Figura 27– Teste do controle de iluminação



Fonte - Elaborado pelo Autor, 2020.

Por fim, após realizar diversos testes, foi comprovando que o sistema funcionou como o esperado sem apresentar nenhuma eventual falha.

Para comprovar o funcionamento do sistema de monitoramento do nível do reservatório, os testes iniciais foram efetuados de maneira manual em uma protoboard, pois devido ao fato do sistema realizar automaticamente o acionamento e desligamento da bomba de água responsável por encher o reservatório, toda e qualquer possível falha deve ser previamente evitada.

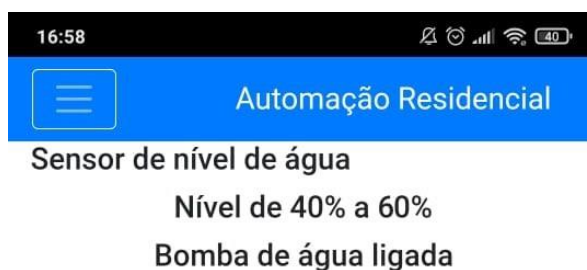
Na sequência de imagens a seguir é possível observar as informações exibidas no aplicativo com o reservatório inicialmente vazio (Figura 28), posteriormente com o nível do reservatório entre 40 a 60% (Figura 29), e por fim, na Figura 30 o reservatório com 100% da sua capacidade de armazenamento.

Figura 28 – Reservatório Vazio



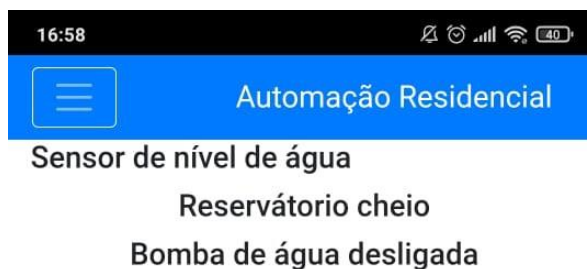
Fonte - Elaborado pelo Autor, 2020.

Figura 29 – Nível do Reservatório de 40% a 60%



Fonte - Elaborado pelo Autor, 2020.

Figura 30 – Nível do Reservatório em 100%



Fonte - Elaborado pelo Autor, 2020.

Conforme citado no item 3.2.4, para o sistema de segurança foi utilizando 2 sensores de presença modelo PIR-HC-SR50. Após efetuar alguns testes, verificou-se um atraso de aproximadamente 3 segundos na resposta de ambos os sensores, apesar disso, o sistema funcionou como o esperado, de tal modo que, ao ativar o alarme, quando algum dos sensores detecta um movimento o alarme dispara e uma sirena é acionada permanecendo ligada até o alarme ser desligado.

A seguir na Figura 31 é possível observar a interface do aplicativo quando o alarme está desativado, e posteriormente na Figura 32 a interface com o alarme ativado.

Figura 31 – Alarme Desligado



Fonte - Elaborado pelo Autor, 2020.

Figura 32 – Alarme Ligado



Fonte - Elaborado pelo Autor, 2020.

Verifica-se por meio da Figura 33, que o sensor 1 detectou movimento disparando o alarme, desta forma, a sirene é ativada e permanece ligada até o alarme ser desligado.

Figura 33 – Alarme Disparado



Fonte - Elaborado pelo Autor, 2020.

Na Figura 34, observa-se a interface desenvolvida no aplicativo para monitoramento da temperatura, o sensor DS18B20 demonstrou ser uma ótima opção, atendendo perfeitamente os objetivos esperados.

Figura 34 – Interface de Verificação da Temperatura



Fonte - Elaborado pelo Autor, 2020.

### 4.3 Validação do Controle de Acesso

Conforme citado no item 3.3.3, na Figura 35 observa-se a resposta do sistema ao aproximar um tag previamente autorizada no código fonte do microcontrolador, para liberar o acesso. Posteriormente na Figura 36 é apresentado a resposta do sistema ao aproximar do módulo RFID, as tags não cadastradas.

Na Figura 35 também é possível observar que o rele está com o led verde ligado, indicando a abertura da fechadura eletrônica.

Figura 35 – Acesso Liberado



Fonte - Elaborado pelo Autor, 2020.

Figura 36 – Acesso Negado



Fonte - Elaborado pelo Autor, 2020.

#### 4.4 Recursos Financeiros

Na Tabela 3 estão dispostos todos os componentes utilizados para a elaboração do protótipo, bem como seus custos e quantidades.

Tabela 3 - Demonstração de valores e quantidade dos componentes utilizados.

<b>Componente</b>	<b>Quantidade</b>	<b>Valor Unitário R\$</b>	<b>Preço Total R\$</b>
Microcontrolador ESP32	2	56,00	112,00
Módulo Relé de 8 canais 5V	1	34,80	34,80
Módulo Relé 1 canal 5V	2	7,90	15,80
Sensor de nível	5	21,90	109,50
Resistor 10 k	15	0,15	2,25
Resistor 4,7 k	2	0,10	0,20
Sensor de presença e movimento (PIR)	2	10,90	21,80
Sensor de temperatura DS18B20	1	19,50	19,50
Leitor Rfid Rc522 13,56 Mhz	1	19,90	19,90
Tags de acesso	4	1,99	7,96
LED	4	1,00	4,00
Display Lcd 16x2	1	18,99	18,99
Alarme Buzzer	1	1,70	1,70
Fonte Ajustável 3.3v 5v 1A	1	14,90	14,90
Protoboard 400 Pontos	1	13,00	13,00
Fios e conectores	1	20,90	20,90
		<b>Custo final:</b>	<b>417,20</b>

Fonte - Elaborado pelo Autor, 2020.

Levando em consideração o custo total dos componentes utilizados para o desenvolvimento do sistema de automação, torna-se visível a vantagem econômica da elaboração deste projeto. Entretanto o valor da instalação do sistema pode variar de acordo com o profissional contrato.

#### 4.5 Dificuldades Encontradas

De fato, uma das maiores dificuldades encontradas foi realizar o desenvolvimento do aplicativo, pois apesar da interface amigável do Corcova, é preciso um conhecimento em aplicações com HTML, CSS e JavaScript. Deste modo, se fez necessário o estudo de conceitos importantes sobre linguagens e métodos de programação do mesmo.

## 5 CONSIDERAÇÕES FINAIS

Mesmo sendo uma área pouco explorada, a automação residencial está se desenvolvendo em ritmo acelerado devido as várias oportunidades e inovação que a mesma oferece. Porém seu maior desafio é atender de forma personalizada cada usuário, de acordo com seu estilo de vida e preferências, já que determinado individuo dá maior importância a segurança, enquanto outro prefere o conforto. Deste modo o sistema de automação instalado deve ser dinâmico o suficiente para atender ambos, pois se o mesmo não atender aos critérios mínimos de cada morador, em questão de meses ele vai ser desligado e deixado de lado.

Deste modo, o objetivo deste trabalho foi projetar e implementar um Sistema de automação Residencial de baixo custo quando comparado aos demais sistemas disponíveis atualmente no mercado, tornando o ambiente residencial mais acessível, englobando aspectos referentes a segurança, comodidade e eficiência energética.

Na primeira etapa do projeto, com base no estudo bibliográfico realizado, tomou-se conhecimento de conceitos importantes os quais foram fundamentais para o entendimento e desenvolvimento do projeto, pois mesmo sendo uma área pouco explorada, a automação residencial está se desenvolvendo em ritmo acelerado devido as diversas oportunidades e inovações que a mesma oferece.

Desta maneira, para validar a metodologia e atender os objetivos propostos no projeto, foi imprescindível a utilização de um microcontrolador confiável com uma grande flexibilidade de aplicações, tais características encontradas no ESP 32 que se apresentou como uma alternativa economicamente viável.

Na implementação pratica do projeto, ao integrar *software* e *hardware*, verificou-se uma grande facilidade de controlar o sistema, pois o ESP 32 mostrou-se bastante eficiente e rápido ao executar os comandos recebidos por meio da interface de gerenciamento do sistema, bem como, uma resposta imediata do aplicativo ao atualizar as informações recebidas dos sensores de nível e temperatura.

Diante do exposto, ao analisarmos os resultados, podemos concluir que todos os objetivos do trabalho foram alcançados. Resultando em um protótipo de custo reduzido, para o controle e supervisão dos principais acionamentos elétricos definidos, com plenas condições de ser implementado posteriormente na residência.

## 5.1 Perspectivas Futuras

Durante o desenvolvimento do projeto, foram identificadas as seguintes oportunidades de melhorias para trabalhos futuros.

- Controle de novos dispositivos, tais como, televisores, sistema de som e sistema de climatização;
- Implementação de um sensor de temperatura externo;
- Sistema de alarme para vazamento de gás inflamável, como o gás butano;
- Adicionar um sistema de reconhecimento de voz, para acionamento dos dispositivos;
- Por fim realizar a implantação de um *Login* de acesso ao sistema, visando uma maior segurança.

## REFERÊNCIAS

ACCARDI, A; DODONOV, E. **Automação Residencial: Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos.** São Carlos. Disponível em: <<http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/17829/material/ARTIG002.pdf>>. Acesso em: 11 abr. de 2020.

ALLAFI. I.; Iqbal. T. **Design and Implementation of a Low Cost Web Server Using ESP32 for Real-Time Photovoltaic System Monitoring,** IEEE Electrical Power and Energy Conference (EPEC), 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8286184>>. Acesso em: 17 abr. de 2020.

ALMEIDA, Gabriel. **Configurando o ambiente de desenvolvimento do ESP32 no Windows.** Mar,2018. Disponível em:<<https://www.embarcados.com.br/ambiente-esp32-no-windows/>>. Acesso em: 25 abr. de 2020.

ARDUÍNO. **ARDUINO MEGA 2560 REV3.** 2019.. Disponível em:<<https://store.arduino.cc/usa/mega-2560-r3>>. Acesso em: 25 abr. de 2020.

AURESIDE, **Associação Brasileira de Automação Residencial.** Disponível em: <[www.aureside.org.br](http://www.aureside.org.br)>. Acesso em: 2 mai. de 2020.

BEGHINI, Lucas B, **Automação residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android.** Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-04022014-152853/?&lang=br>>. Acesso em: 02 mai. de 2020.

BOLZANI, C. A. M. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes.** 2010 Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos, 2010.

BOLZANI, C. A. M. **Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes: Uma Introdução aos Sistemas Domóticos.** 2004. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos, 2004.

BOLZANI, C. A. M. **Desmistificando a Domótica.** Disponível em: <[http://www.bolzani.com.br/artigos/art01\\_07.pdf](http://www.bolzani.com.br/artigos/art01_07.pdf)>. Acesso em: 16 mai. de 2020.

BOLZANI, C. A. M. **Domótica, a nova ciência do século XXI.** Revista Fonte, p105, dez. 2013. Disponível em: <<http://www.bolzani.com.br/artigos/revistafonte.pdf>>. Acesso em: 16 mai. de 2020.

BOLZANI, C. A. M. **Residências Inteligentes.** São Paulo: Editora e Livraria da Física, 2004. Acesso em: 16 mai. de 2020.

BUNEMER, Ricardo, Dissertação de mestrado **Domótica Assistiva Utilizando Sistemas Integrados de Supervisão e Controle.** CAMPINAS 2014.

CAMPOS, Roberto. **Automação Residencial Utilizando Arduino E Aplicação Web**. Junho 2014, BRASÍLIA – DF.

CURVELLO, André. **ESP32 – Um grande aliado para o Maker IoT**. Fev, 2018. Disponível em: <<https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>>. Acesso em: 23 mai. de 2020.

DAL AGNOL, Cleiton. **Comparação entre microcontroladores e aplicação do fpga no controle do conversor boost**. LAGES 2018.

DUTRA, M. S.; ROMANO, V. F. **Introdução à robótica industrial**. 1. ed. São Paulo: Blucher, 2016.

ESPRESSIF. **ESP-WROOM-32 Series: Datasheet. 3. ed. Xangai**: Espressif System, 2019. Disponível em:

<[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)>.

Acesso em: jun. de 2020. Acesso em: 12 jul. de 2020.

FERREIRA, João. **Interface homem-máquina para domótica baseado em tecnologias Web** junho 2008.

GIMENEZ, Pinillos, S. **Microcontroladores 8051 - Conceitos, Operação, Fluxogramas e Programação**. 1. ed. São Paulo: Editora Érica, 2015.

GIMENEZ, Pinillos, S. **Microcontroladores 8051 - Teoria e Prática**. 1. ed. São Paulo: Editora Érica, 2010.

GOMES, Andrew B; SILVA, Guilherme A. C; GELACKI, Raphae. **Automação residencial utilizando uma plataforma de baixo custo**. In: **Trabalho de Conclusão de Curso (Tecnologia em Automação Industrial)**. [S.l.]: Universidade Tecnológica Federal do Paraná, 2016.

LENZ, Lucian, M. **Microprocessadores**. Porto Alegre: SAGAH, 2019.

MARTINS, N. A. **Sistemas Microcontrolados: uma abordagem com o microcontrolador PIC16F84**. São Paulo: Novatec, 2005. Disponível em: <[www.martinsfontespaulista.com.br/anexos/produtos/capitulos/203289.pdf](http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/203289.pdf)>. Acesso: 4 jun, 2020.

MIYADAIRA, Alberto N. **Microcontroladores PIC18 - Aprenda e Programe em Linguagem C**. 4 ed. São Paulo; Editora Érica, 2013.

MOLINA, et al. **Proposta de um protótipo de automação residencial de baixo custo controlado por dispositivos móveis**. Instituto federal de alagoas – IFAL, 2019.

MORAIS, Henrique S. **Automação Residencial - Sistema Integrado de Uma Residência Inteligente Utilizando Controlador Lógico Programável**. João Monlevade, MG 2017.

MURATORI, J. R. **Os desafios do mercado da Automação Residencial**. Disponível em: <[http://www.aecweb.com.br/cont/a/os-desafios-do-mercado-da-residencial\\_8192](http://www.aecweb.com.br/cont/a/os-desafios-do-mercado-da-residencial_8192)>. Acesso em: 13 jun. de 2020.

MURATORI, J. R.; DAL BÓ, P. H. **Automação Residencial, Conceitos e Aplicações**. São Paulo: Educere, 2013.

OLIVEIRA, N, M. **Paralelo entre as arquiteturas von Neumann e Harvard**. Recife, 2018

PEREIRA, Fábio. **Microcontroladores PIC - Técnicas Avançadas**. 6 ed. São Paulo: Editora Érica, 2007.

PEREIRA, Marcus. **Casa robótica já não é mais tão futurístico assim**. 2019. Disponível em: <<https://jornaldebrasil.com.br/nahorah/casa-robotica-ja-nao-e-mais-tao-futuristico-assim/>>. Acesso em: 18 jun. de 2020.

PRUDENTE, Francesco. **Automação Predial e Residencial - Uma Introdução**. Rio de Janeiro: LTC, 2017.

QUINDERÉ, Patrick. **Casa Inteligente – Um Protótipo de Sistema de Automação Residencial de Baixo Custo**. Fortaleza, 2009.

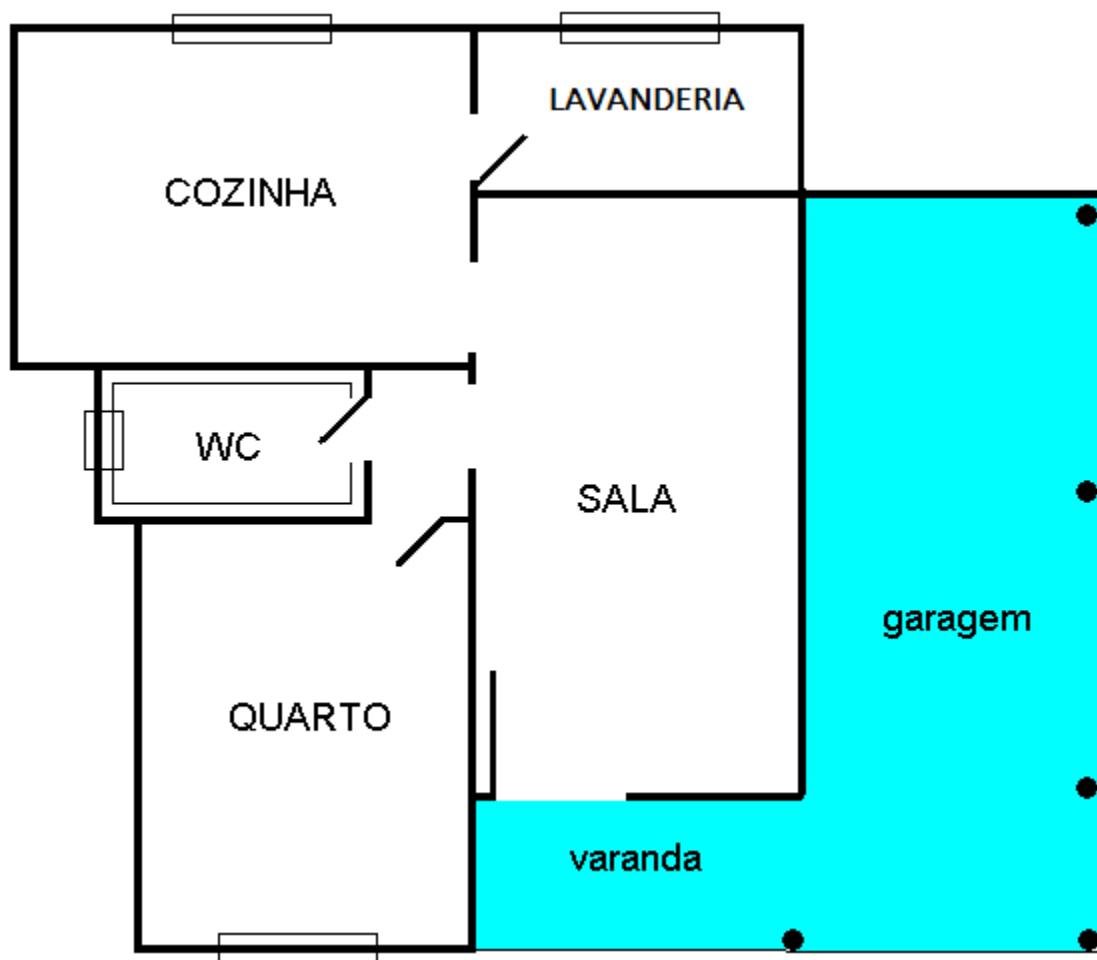
ROVERI, Michael. **Curso superior de tecnologia em redes de computadores**. Santa Bárbara d'Oeste 2012

SILVA, D. S. **Desenvolvimento e Implantação de um Sistema de Supervisão e Controle Residencial**. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte, Natal-RN, 2009

TEZA, V. R. **Alguns aspectos sobre a automação residencial**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Florianópolis, 2002.

THOMSEN, Adilson. **O que é Arduino**. Set, 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 26 abr. de 2020.

USINAINFO. **Sensor de Nível de Água com Boia Horizontal**. Disponível em: <<https://www.usinainfo.com.br/sensor-de-nivel-arduino/sensor-de-nivel-de-agua-com-boia-horizontal-2580.html>>. Acesso em: 4 jul. de 2020.

**APÊNDICE A – Planta Baixa da residência a ser instalado o sistema futuramente.**

## APÊNDICE B – Código Fonte do Sistema de Automação Interno da Residência.

```

#include <WiFi.h> //INCLUSÃO DA BIBLIOTECA
#include <OneWire.h> //INCLUSÃO DA BIBLIOTECA
#include <DallasTemperature.h> //INCLUSÃO DA BIBLIOTECA
// Dados para autenticação do WiFi
const char* ssid = "TCCAnderson";
const char* password = "tcc2020";
// Declaração dos pinos para cada função
int rele_1 = 23; // Luz Central Sala
int rele_2 = 22; // Luz Fita Led Sala
int rele_3 = 21; // Luz Principal Quarto
int rele_4 = 19; // Luz Escrivania Quarto
int rele_5 = 18; // Luz Cozinha
int rele_6 = 17; // Luz Lavanderia
int rele_7 = 16; // Luz Banheiro
int rele_8 = 15; // Luz Garagem
int alarme = 0;
const int oneWireBus = 33; // Sensor de Temperatura
int sensor_presenca_1 = 12; // Sensor de Presença 1
int sensor_presenca_2 = 14; // Sensor de Presença 2
int rele_sirene = 2; // Sirene do alarme
int sensor_nivel_1 = 26; // Sensor de nivel 1
int sensor_nivel_25 = 25; // Sensor de nivel 2
int sensor_nivel_50 = 35; // Sensor de nivel 3
int sensor_nivel_75 = 34; // Sensor de nivel 4
int sensor_nivel_100 = 39; // Sensor de nivel 5
int rele_bomba_agua = 32;
int nivel_1 = 1;
int nivel_25 = 1;
int nivel_50 = 1;
int nivel_75 = 1;
int nivel_100 = 1;
int nivel_agua = 0;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);
boolean rele_1_ligado = HIGH;
boolean rele_2_ligado = HIGH;
boolean rele_3_ligado = HIGH;
boolean rele_4_ligado = HIGH;
boolean rele_5_ligado = HIGH;
boolean rele_6_ligado = HIGH;
boolean rele_7_ligado = HIGH;
boolean rele_8_ligado = HIGH;
WiFiServer server (80);

```

```

void setup() {
  delay(1000);
  pinMode(rele_1, OUTPUT);
  pinMode(rele_2, OUTPUT);
  pinMode(rele_3, OUTPUT);
  pinMode(rele_4, OUTPUT);
  pinMode(rele_5, OUTPUT);
  pinMode(rele_6, OUTPUT);
  pinMode(rele_7, OUTPUT);
  pinMode(rele_8, OUTPUT);
  pinMode(rele_sirene, OUTPUT);
  pinMode(sensor_nivel_1, INPUT);
  pinMode(sensor_nivel_25, INPUT);
  pinMode(sensor_nivel_50, INPUT);
  pinMode(sensor_nivel_75, INPUT);
  pinMode(sensor_nivel_100, INPUT);
  pinMode(rele_bomba_agua, OUTPUT);
  digitalWrite(rele_1, HIGH);
  digitalWrite(rele_2, HIGH);
  digitalWrite(rele_3, HIGH);
  digitalWrite(rele_4, HIGH);
  digitalWrite(rele_5, HIGH);
  digitalWrite(rele_6, HIGH);
  digitalWrite(rele_7, HIGH);
  digitalWrite(rele_8, HIGH);
  digitalWrite(rele_sirene, HIGH);
  digitalWrite(rele_bomba_agua, HIGH);
  pinMode(sensor_presenca_1, INPUT);
  pinMode(sensor_presenca_2, INPUT);
  Serial.begin(9600); // Vamos usar para debug
  sensors.begin();

  // Conectando à rede WiFi
  Serial.println();
  Serial.println();
  Serial.print("Conectando com: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado com sucesso");

  //Configurações do IP fixo, conforme a rede escolhida
  IPAddress ip(192, 168, 0, 150);
  IPAddress gateway (192, 168, 0, 254);
  IPAddress subnet(255, 255, 255, 0);
  Serial.print("Configurando IP fixo para: ");
  Serial.println(ip);

  //Envia a configuração
  WiFi.config(ip, gateway, subnet);

```

```

//Inicializa o server criado na porta 80
server.begin();

//Mostra no monitor serial o IP que o ESP possui
Serial.print("Server em: ");
Serial.println(WiFi.localIP());

}
void loop() {

  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);
  float temperatureF = sensors.getTempFByIndex(0);
  Serial.print(temperatureC);
  Serial.println("°C");
  Serial.print(temperatureF);
  Serial.println("°F");

  int sinal_sensor_presenca_1 = digitalRead(sensor_presenca_1);
  int sinal_sensor_presenca_2 = digitalRead(sensor_presenca_2);

  Serial.print(sinal_sensor_presenca_1);
  Serial.println("sinal_sensor_presenca_1");
  Serial.print(sinal_sensor_presenca_2);
  Serial.println("testesinal_sensor_presenca_2");

  if(alarme == 0){
    Serial.println("alarme desligado"); }
  else{
    Serial.println("alarme ligado"); }

  int nivel_1 = digitalRead(sensor_nivel_1);
  int nivel_25 = digitalRead(sensor_nivel_25);
  int nivel_50 = digitalRead(sensor_nivel_50);
  int nivel_75 = digitalRead(sensor_nivel_75);
  int nivel_100 = digitalRead(sensor_nivel_100);

  if ((nivel_1 == 1) && (nivel_25 == 1) && (nivel_50 == 1) && (nivel_75 == 1) && (nivel_100 ==
1)) {
    Serial.println("Reservatório Cheio");
    nivel_agua = 100;}

  if ((nivel_1 == 1) && (nivel_25 == 1) && (nivel_50 == 1) && (nivel_75 == 1) && (nivel_100 == 0))
  {
    Serial.println("Nível de 80 a 100%");
    nivel_agua = 75;}

  if ((nivel_1 == 1) && (nivel_25 == 1) && (nivel_50 == 1) && (nivel_75 == 0) && (nivel_100 == 0))
  {
    Serial.println("Nível de 60 a 80%");
    nivel_agua = 50;}

  if ((nivel_1 == 1) && (nivel_25 == 1) && (nivel_50 == 0) && (nivel_75 == 0) && (nivel_100 == 0))
  {
    Serial.println("Nível de 40 a 60%");

```

```

        nivel_agua = 25; }

if ((nivel_1 == 1) && (nivel_25 == 0) && (nivel_50 == 0) && (nivel_75 == 0) && (nivel_100 == 0))
{
    Serial.println("Nivel Critico");
    nivel_agua = 1; }

if ((nivel_1 == 0) && (nivel_25 == 0) && (nivel_50 == 0) && (nivel_75 == 0) && (nivel_100 == 0))
{
    Serial.println("Reservatório Vazio");
    nivel_agua = 0;
}

// Aguardando requisições do morador
WiFiClient client = server.available();
if(client)
{
    boolean continua = true;
    String linha = "";

    while(client.connected()){
        if(client.available()){
            char c = client.read();
            linha.concat(c);

            if(c == '\n' && continua)
            {
                client.println("HTTP/1.1 200 OK");
                client.println("Content-Type: text/html");
                client.println();
                client.println();

                int iniciofrente = linha.indexOf("?");

                if(iniciofrente>-1){
                    iniciofrente = iniciofrente+6;
                    int fimfrente = iniciofrente+3;
                    String acao = linha.substring(iniciofrente,fimfrente);
                    Serial.println("acao uai ");
                    Serial.print(acao);
                    if (acao == "001") {
                        rele_1_ligado = !rele_1_ligado;
                        digitalWrite(rele_1,rele_1_ligado); }

                    else if (acao == "002") {
                        rele_2_ligado = !rele_2_ligado;
                        digitalWrite(rele_2,rele_2_ligado); }

                    else if (acao == "003") {
                        rele_3_ligado = !rele_3_ligado;
                        digitalWrite(rele_3,rele_3_ligado); }

                    else if (acao == "004") {
                        rele_4_ligado = !rele_4_ligado;
                        digitalWrite(rele_4,rele_4_ligado); }

```

```

else if (acao == "005") {
    rele_5_ligado = !rele_5_ligado;
    digitalWrite(rele_5,rele_5_ligado); }

else if (acao == "006") {
    rele_6_ligado = !rele_6_ligado;
    digitalWrite(rele_6,rele_6_ligado); }

else if (acao == "007") {
    rele_7_ligado = !rele_7_ligado;
    digitalWrite(rele_7,rele_7_ligado); }

else if (acao == "008") {
    rele_8_ligado = !rele_8_ligado;
    digitalWrite(rele_8,rele_8_ligado); }

else if (acao == "010") {
    Serial.println("liga_alarme");
    alarma = 1; }

else if (acao == "011") {
    Serial.println("desliga_alarme");
    alarma = 0; }

else if (acao == "012") {
    Serial.println("desliga_alarme");
    alarma = 0; }

if ((alarma == 1)&&((sinal_sensor_presenca_1 == 1)||((sinal_sensor_presenca_2 == 1))) {
    Serial.println("alarma disparando");
    alarma = 2; }

if(alarma != 2){
    digitalWrite(rele_sirene, LOW); }

if(alarma == 2){
    digitalWrite(rele_sirene, HIGH); }

if(nivel_agua == 100){
    digitalWrite(rele_bomba_agua, LOW); }

if(nivel_agua < 50){
    digitalWrite(rele_bomba_agua, HIGH); }

else { }
client.print("dados({ rele_1: ");
client.print(digitalRead(rele_1));
client.print(",");
client.print(" rele_2 : ");
client.print(digitalRead(rele_2));
client.print(",");
client.print(" rele_3 : ");
client.print(digitalRead(rele_3));
client.print(",");

```

```

client.print(" rele_4 : ");
client.print(digitalRead(rele_4));
client.print(",");
client.print(" rele_5 : ");
client.print(digitalRead(rele_5));
client.print(",");
client.print(" rele_6 : ");
client.print(digitalRead(rele_6));
client.print(",");
client.print(" rele_7 : ");
client.print(digitalRead(rele_7));
client.print(",");
client.print(" rele_8 : ");
client.print(digitalRead(rele_8));
client.print(",");
client.print(" rele_sirene : ");
client.print(digitalRead(rele_sirene));
client.print(",");
client.print(" temp : ");
client.print((temperatureC));
client.print(",");
client.print(" sensor_presenca_1 : ");
client.print((sinal_sensor_presenca_1));
client.print(",");
client.print(" sensor_presenca_2 : ");
client.print((sinal_sensor_presenca_2));
client.print(",");
client.print(" status_alarme : ");
client.print((alarme));
client.print(",");
client.print(" nivel_agua : ");
client.print((nivel_agua));
client.print(",");
client.print(" rele_bomba_agua : ");
client.print(digitalRead(rele_bomba_agua));
client.print("}"));
}
break;
}
if(c == '\n') {continua = true; }
else if (c != '\r') {continua = false; }

} }

delay(1);
client.stop();}}

```

## APÊNDICE C – Código Fonte do Sistema de Controle de Acesso.

```

#include <Wire.h> //INCLUSÃO DA BIBLIOTECA
#include <SPI.h> //INCLUSÃO DE BIBLIOTECA
#include <MFRC522.h> //INCLUSÃO DE BIBLIOTECA
#include <LiquidCrystal_I2C.h>;

#define SS_PIN 10 //PINO SDA
#define RST_PIN 9 //PINO DE RESET

LiquidCrystal_I2C lcd(0x27, 16, 2);

MFRC522 rfid(SS_PIN, RST_PIN); //PASSAGEM DE PARÂMETROS REFERENTE AOS PINOS

const int pinoLedVerde = 5; //PINO DIGITAL REFERENTE AO LED VERDE
const int pinoLedVermelho = 6; //PINO DIGITAL REFERENTE AO LED VERMELHO
const int pinoRele = 2; //PINO REFERENTE AO RELE

void setup()
{
  lcd.init();
  lcd.backlight();

  Wire.begin(); //INICIALIZA A BIBLIOTECA WIRE
  SPI.begin(); //INICIALIZA O BARRAMENTO SPI
  rfid.PCD_Init(); //INICIALIZA MFRC522

  pinMode(pinoLedVerde, OUTPUT); //DEFINE O PINO COMO SAÍDA
  pinMode(pinoLedVermelho, OUTPUT); //DEFINE O PINO COMO SAÍDA
  pinMode(pinoRele, OUTPUT); //DEFINE O PINO COMO SAÍDA

  digitalWrite(pinoLedVerde, LOW); //LED INICIA DESLIGADO
  digitalWrite(pinoLedVermelho, LOW); //LED INICIA DESLIGADO
  digitalWrite(pinoRele, LOW); //RELE INICIA DESLIGADO
  Serial.begin(9600);

  // Mensagem inicial no lcd
  lcd.setCursor(1, 0);
  lcd.print("APROXIME A TAG");
}

void loop() {
  leituraRfid(); //CHAMA A FUNÇÃO RESPONSÁVEL PELA VALIDAÇÃO DA TAG RFID
}

//FUNÇÃO DE VALIDAÇÃO DA TAG RFID
void leituraRfid(){
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) //VERIFICA SE O CARTÃO
  PRESENTE NO LEITOR É DIFERENTE DO ÚLTIMO CARTÃO LIDO. CASO NÃO SEJA, FAZ
  return; //RETORNA PARA LER NOVAMENTE

  /***INICIO BLOCO DE CÓDIGO RESPONSÁVEL POR GERAR A TAG RFID LIDA***/
  String strID = "";
  for (byte i = 0; i < 4; i++)

```

```

{
  strID +=
  (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
  String(rfid.uid.uidByte[i], HEX) +
  (i!=3 ? ":" : "");
}
strID.toUpperCase();

/**FIM DO BLOCO DE CÓDIGO RESPONSÁVEL POR GERAR A TAG RFID LIDA***/

if (strID.indexOf("39:29:F2:B9") &&("B9:6A:00:C2") >= 0)

//SE O ENDEREÇO DA TAG LIDA FOR IGUAL AO ENDEREÇO INFORMADO, FAZ
{
  digitalWrite(pinoLedVerde, HIGH); //LIGA O LED VERDE
  digitalWrite(pinoRele, HIGH); //LIBERA O PORTÃO
  Serial.println("Acesso liberado !");
  lcd.setCursor(1, 0);
  lcd.print("BEM VINDO !!! ");
  lcd.setCursor(0, 1);
  lcd.print("ACESSO LIBERADO ");

  delay(3000); //INTERVALO DE 4 SEGUNDOS

  digitalWrite(pinoLedVerde, LOW); //DESLIGA O LED VERDE
  digitalWrite(pinoRele, LOW); //DESLIGA O RELE DO PORTÃO
  lcd.setCursor(1, 0);
  lcd.print("APROXIME A TAG");
  lcd.setCursor(0, 1);
  lcd.print("          "); }

else //SENÃO, FAZ (CASO A TAG LIDA NÃO SEJÁ VÁLIDA)
{
  digitalWrite(pinoLedVermelho, HIGH); //LIGA O LED VERMELHO
  Serial.println("Acesso Negado !");
  lcd.setCursor(1, 0);
  lcd.print("ERRO !!! ");
  lcd.setCursor(1, 1);
  lcd.print("ACESSO NEGADO");

  delay(3000); ///INTERVALO DE 6 SEGUNDOS

  digitalWrite(pinoLedVermelho, LOW); //DESLIGA O LED VERDE
  lcd.setCursor(1, 0);
  lcd.print("APROXIME A TAG");
  lcd.setCursor(0, 1);
  lcd.print("          ");
}

rfid.PICC_HaltA(); //PARADA DA LEITURA DO CARTÃO
rfid.PCD_StopCrypto1(); //PARADA DA CRIPTOGRAFIA NO PCD
}

```