

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Victor Oliveira de Andrade Melo

**Central de Automação Residencial Microprocessada de Baixo Custo Aplicada na Execução
de Eventos Pontuais e Pré-Programados.**

SÃO LUÍS

2019

Victor Oliveira de Andrade Melo

**Central de Automação Residencial Microprocessada de Baixo Custo Aplicada na Execução
de Eventos Pontuais e Pré-Programados.**

Monografia apresentada ao Curso de
Engenharia Elétrica da Universidade Federal
do Maranhão como requisito para obtenção do
grau de Bacharel em Engenharia Elétrica

Orientador: Prof. Luciano Buonocore.

São Luís
2019

Oliveira de Andrade Melo, Victor.

Central de Automação Residencial Microprocessada de Baixo Custo Aplicada na Execução de Eventos Pontuais e Pré-Programados / Victor Oliveira de Andrade Melo. - 2019. 87 p.

Orientador(a): Luciano Buonocore.

Monografia (Graduação) - Curso de Engenharia Elétrica, Universidade Federal do Maranhão, São Luís, 2019.

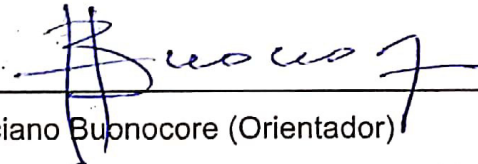
1. Automação. 2. Baixo. 3. Custo. 4. Perfil. 5. Residencial. I. Buonocore, Luciano. II. Título.

Victor Oliveira de Andrade Melo

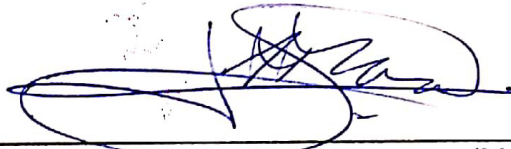
**Central de Automação Residencial Microprocessada de Baixo Custo Aplicada
na Execução de Eventos Pontuais e Pré-Programados.**

Aprovada em 18/07/2019

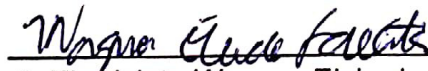
BANCA EXAMINADORA



Prof. Luciano Buonocore (Orientador)
Departamento de Engenharia de Eletricidade – UFMA



Prof. Angel Fernando Torrico Caceres (Membro)
Departamento de Engenharia de Eletricidade – UFMA



Engº. Eletricista Wagner Elvio de Loiola Costa (Membro)

AGRADECIMENTOS

Agradeço primeiramente à minha família, principalmente meu pai, minha mãe, minha avó e meus irmãos, por me fornecerem apoio e amor para continuar firme nessa jornada. Por toda a perseverança que me ensinaram a ter, serei eternamente grato.

Aos meus tios, por terem cuidado muito bem de mim, fornecendo apoio e abrigo. Ao meu primo Jorginho por ter me fornecido meu primeiro computador, que foi indispensável nessa caminhada.

Ao professor Luciano Buonocore, pela orientação, disposição, dedicação e principalmente pela empatia, pois parece nunca ter se esquecido que já foi um aluno. Me forneceu todo o apoio que eu precisava e só tenho a agradecer por ter conhecido um profissional tão competente e altruísta.

A todos os alunos do LRC, pois me fornecerem um ambiente amigável e produtivo, além de tempo e disposição. Principalmente ao Zarley por ter me fornecido muito apoio e motivação na conclusão desse trabalho.

Aos colegas de engenharia elétrica, principalmente à turma 2012.2, por participarem dessa jornada árdua, porém recompensante. Enfim, agradeço a todos aqueles que participarem direta ou indiretamente da minha formação acadêmica.

RESUMO

Inserido no contexto da quarta revolução industrial e com o aprimoramento de conceitos como a internet das coisas, robótica e inteligência artificial, a automação residencial se mostra um mercado promissor com perspectivas de grandes investimentos. Esse trabalho fez a implementação de um sistema de automação residencial, dando prosseguimento a uma pesquisa anterior. Este traz como avanços importantes a execuções de rotinas com ações reativas, baseadas em informações sensoriais e de perfis customizados pelo residente, onde horários de ligar e desligar equipamentos na residência são realizados, simulando a presença de pessoas. Ambas os tipos de automações implementados visam melhorar a segurança da casa, sendo que a dos perfis tenta minimizar a tentativa de roubos, quando não existem moradores na residência. Outra fundamental contribuição desta monografia foi a diminuição dos custos, onde o principal elemento removido foi o dispositivo Alexa, usado no trabalho anterior. Para realizar a ação de atuadores ou a leitura de sensores, bem como a execução das automações programadas (reativas e perfis customizados) foram utilizados os microcontroladores ESP01 e o NodeMCU, além da placa embarcada microprocessada Raspberry Pi 3B (em substituição ao modelo Zero W por questões de disponibilidade), todos com capacidade de comunicação Wi-Fi. Os *software* utilizados nesse trabalho foram o *Home Assistant*, o *Hass.io* e a IDE (Integrated Development Environment) do Arduino, todos *open source*, o que proporcionou o desenvolvimento de um modelo barato. Experimentos foram conduzidos para acionar e desligar individualmente os equipamentos pertencentes ao sistema de automação implementado, bem como na validação das automações reativas e de perfis customizados. Em todos esses testes as reações do sistema foram consistentes com o esperado, sendo os dados apresentados no texto do trabalho.

Palavras-chaves: Automação residencial; Microcontroladores; Raspberry Pi 3B; Perfis Customizados;

ABSTRACT

In the context of the fourth industrial revolution and with the improvement of concepts such as the internet of things, robotics and artificial intelligence, residential automation is currently a promising market with great investment prospects. This paper made the implementation of a residential automation system, proceeding from a previous research. It brings important implements as the execution of routines with reactive actions, based on sensory information and customized profiles by the resident, which moments to turn on and off equipment in the residence are performed, simulating the presence of people. Both types of implemented automations aim to improve the safety of the house, while the one with the profiles tries to minimize the robbery attempt, when there are no residents in the residence. Another fundamental contribution of this monography was the reduction of costs, where the main element removed was the Alexa device, used in the previous project. To perform the actuator action or the sensor reading, as well as the programmed automation (reactive and customized profiles), the ESP01 and NodeMCU microcontrollers, as well as the microprocessor board Raspberry Pi 3B (instead of the Zero W model due to availability issues), were used, all of them with Wi-Fi capability. The software used in this work was the Home Assistant, Hass.io and the Arduino IDE, all open source, which provided the development of a very cheap model. Experiments were conducted to individually turn on and off the devices implemented in the automation system, and to validate the reactive automations and customized profiles. In all of these tests, the reactions of the system were consistent with the expected, being the data presented in the text of the work.

Keywords: Residential automation; Microcontrollers; Raspberry Pi 3B; Customized Profiles;

LISTA DE FIGURAS

Figura 1 – Trator sendo utilizado na produção agrícola.....	19
Figura 2 – Lavadora e secadora de roupa Samsung Qdrive	20
Figura 3 – Robô Aspirador Xiaomi Roborock S50 2ª Geração	21
Figura 4 – Módulos X-10 para automação residencial.....	21
Figura 5 – Dispositivos presentes em uma casa inteligente	22
Figura 6 – Comparação dos principais mercados de automação residencial	23
Figura 7 – Estimativa de crescimento do mercado de automação residencial	24
Figura 8 – Placa ESP8266	24
Figura 9 – Raspberry Pi Zero W.....	25
Figura 10 – Arquitetura centralizada de uma central de automação	26
Figura 11 – Sensor magnético de abertura de porta e janela.....	27
Figura 12 – Módulo atuador para o controle de cargas elétricas.....	27
Figura 13 – Exemplo de central de automação baseada em arquitetura centralizada	28
Figura 14 – Ranking quantitativo de linguagens utilizadas em sistemas embarcados	29
Figura 15 – Exemplo de rede mesh	32
Figura 16 – Raspberry Pi 3 modelo B (RPi 3B).....	33
Figura 17 – Amazon Echo Dot	36
Figura 18 – Interface Arduíno IDE.....	37
Figura 19 – Exemplo de funcionamento do fluxo de dados no protocolo MQTT	38
Figura 20 – ESP8266 placa ESP01	39
Figura 21 – Microcontrolador ESP01 com módulo adaptador USB-UART.....	39
Figura 22 – ESP8266 NodeMCU: a) Foto e b) Pinagem	40
Figura 23 – Broadlink Rm Mini 3	41
Figura 24 – Sonoff instalado na residência	41
Figura 25 – Sensor DSB18B20+	42
Figura 26 – Sensor PIR HC-SR501	43
Figura 27 – Módulo LDR.....	44
Figura 28 – Sensor de metano MQ4.....	45
Figura 29 – Regulador de tensão LM2596	45
Figura 30 – XBee explorer USB adapter.....	46

Figura 31 – Esquema do sistema de automação completo	47
Figura 32 – Arquivo de configuração da rede Wi-Fi	49
Figura 33 – Acesso e atualização do sistema Hass.io	50
Figura 34 – Login no sistema do Home Assistant.....	50
Figura 35 - ESP8266 acoplado à placa do dispositivo Sonoff	51
Figura 36 – Códigos na Arduíno IDE para a programação do Sonoff	52
Figura 37 – Código do arquivo de configuração referente ao Sonoff	53
Figura 38 – Controle e estado da lâmpada na interface do Home Assistant	53
Figura 39 – Endereços de IP e MAC encontrados através do aplicativo Network Scanner	54
Figura 40 - Código para integração do Broadlink à Central de Automação	54
Figura 41 – Código do controle recebido pelo Home Assistant e inserido no arquivo de configuração	55
Figura 42– Interface de Controle do ar-condicionado no Home Assistant	56
Figura 43 – Sensores integrados à interface do Home Assistant.....	56
Figura 44 – Linhas de código para adição do DSB18B20+ ao Home Assistant.....	57
Figura 45 – Linhas de código na Arduíno IDE para aquisição de dados e transferência via MQTT do sensor de temperatura	57
Figura 46 – Linhas do código de automação para economia de energia.....	58
Figura 47 – ESP01 e DS18B20+ instalados na residência	59
Figura 48 - Linhas de código para adição do PIR HC – SR501 ao Home Assistant.....	59
Figura 49 - Linhas de código na Arduíno IDE para aquisição de dados e transferência via MQTT do sensor de movimento	60
Figura 50 – Automação do sensor de movimento ativada na interface de controle.....	61
Figura 51 – Linhas do código de automação para o alarme de segurança	61
Figura 52 – Linhas de código para adição do Módulo LDR ao Home Assistant.....	62
Figura 53 – Linhas de código na Arduíno IDE para aquisição de dados e transferência via MQTT do sensor de luz	62
Figura 54 – Linhas do código de automação para rotina de segurança em períodos de ausência..	63
Figura 55 – ESP01 e Módulo LDR em funcionamento	64
Figura 56 – Linhas de código na IDE do Arduíno para a aquisição de dados e transferência via MQTT do sensor de gás	64

Figura 57 – Linhas de código para adição do MQ4 ao Home Assistant	65
Figura 58 – Estado do Sensor na interface do Home Assistant.....	65
Figura 59 – Linhas do código de automação para rotina de níveis irregulares de gás de cozinha. 66	66
Figura 60 – NodeMCU e MQ4 em funcionamento	66
Figura 61 – Interface no Home Assistant para ativação das automações.....	67
Figura 62 – Código da automação com a função de definir um alarme	68
Figura 63 – Interface da entrada numérica e sliders.....	68
Figura 64 – Acesso a interface para dispositivos móveis na rede local	69
Figura 65 – Configuração dos dispositivos na interface para celular.....	70
Figura 66 – Interface para dispositivos móveis com design obsidian	71
Figura 67 – Interface presente no computador	72
Figura 68 – Leitura do sensor de movimento no Log de Eventos.....	73
Figura 69 – “SafetyAlarm” sendo acionada no log de eventos	74
Figura 70 – Temperatura inicial do sensor	74
Figura 71 – Temperatura máxima atingida pelo sensor durante os experimentos.	75
Figura 72 – Histórico de leitura do sensor.....	75
Figura 73 – Acionamento das automações “AirEnergySaver1” e “AirEnergySaver2”	76
Figura 74 – Valores percentuais do sensor.....	76
Figura 75 – Histórico de medição do sensor de gás	77
Figura 76 – Acionamento da automação “GasSafety”	77
Figura 77 – Log de eventos módulo LDR.	78
Figura 78 – Log de eventos automação “VacationOn”	79
Figura 79 – Acionamento remoto dos dispositivos controlados por infravermelho.....	79
Figura 80 – Acionamentos da lâmpada da sala de estar.	80
Figura 81 – Acionamento consecutivo do buzzer na seção log de eventos.....	81
Figura 82 – Temporização na customização de um perfil de 13 horas	82
Figura 83 – Perfil customizado acionado nos horários programados.....	83
Figura 84 – Destaque de dispositivos acionado na interface móvel.....	83
Figura 85 – Histórico de atividades do Home Assistant	84

LISTA DE TABELAS

Tabela 1 – Especificações técnicas do Raspberry Pi 3	35
Tabela 2 – Custo aproximado do sistema de automação residencial implementado.....	71

LISTA DE ABERVIATURAS E SIGLAS

AURESIDE Associação Brasileira de Automação Residencial

IoT *Internet of Things*

TEEE Tópicos Especiais de Engenharia Elétrica

PLC Power Line Carrier

TV Televisão

RPi3B Raspberry Pi 3B

USB Universal Serial Bus

IDE Integrated Development Environment

IP Internet Protocol

UART Universal Asynchronous Receiver Transmitter

MQTT Message Queuing Telemetry

TCP Transmission Control Protocol

MAC Media Access Control

LRC Laboratório de Robótica Móvel e Comunicação Sem Fio

SUMÁRIO

Listas de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

1. INTRODUÇÃO	15
1.1. Objetivo	16
1.2. Motivação	17
1.3 Metodologia.....	17
1.4. Organização do trabalho.....	18
2. FUNDAMENTAÇÃO TEÓRICA	19
2.1. Uma visão geral sobre a automação residencial.....	20
2.1.1. A evolução da automação residencial	20
2.1.2. O mercado da automação residencial	22
2.2. Sistemas embarcados e a automação residencial.....	23
2.2.1 Placas microcontroladas	24
2.2.2 Placas microprocessadas	25
2.3 Elementos de um sistema de automação residencial	26
2.3.1 Sensores.....	26
2.3.2 Transdutores	26
2.3.3 Atuadores	26
2.3.4 Central de automação	27
2.4 Linguagens de programação usuais em sistemas de automação residencial	29
2.4.1 Linguagem C/C++	30
2.4.2 Linguagem Python.....	30
2.5 Meios de comunicação entre elementos de um sistema de automação residencial.....	30
2.5.1 Cabeado	31
2.5.2 Bluetooth	31
2.5.3 Radiofrequência.....	31
2.5.4 Wi-Fi.....	31
2.6 Proposta desenvolvida neste trabalho.....	32

3. HARDWARE E SOFTWARE EMPREGADOS NO SISTEMA DE AUTOMAÇÃO RESIDENCIAL	33
3.1. Central de automação e interface física	33
3.1.1 Placa Raspberry Pi 3B	34
3.1.2 Amazon Echo Dot	34
3.2. Software e Protocolos empregados	35
3.2.1.Home Assistant e o Hass.io	35
3.2.2.Arduino IDE	36
3.2.3.MQTT broker	36
3.3 Atuadores	37
3.3.1 Placa ESP8266.....	38
3.3.2 Broadlink Rm Mini 3.....	40
3.3.3 Sonoff Basic	41
3.4 Sensores	42
3.4.1 Sensor de temperatura DS18B20+	42
3.4.2 Sensor de movimento PIR HC-SR501	43
3.4.3 Sensor de luz em módulo LDR	43
3.4.4 Sensor de Metano – MQ4.....	44
3.4.5 Outros componentes	45
3.5 – Sistema Proposto Completo	46
4. IMPLEMENTAÇÃO DA ESTAÇÃO SENSORIAL DE BAIXO CUSTO COM CAPACIDADE DE CONTROLE	48
4.1. Instalação, conexão e ajustes de configuração com a central de automação.....	48
4.2. Programação e integração do atuador Sonoff.....	50
4.2.1.Programação do ESP8266 (ESP01) presente na placa do Sonoff	51
4.3. Integração do Broadlink RM Mini 3 ao Hass.io.....	52
4.4. Programação dos Sensores da Central.....	55
4.4.1.Integração do DS18B20+ ao sistema da central de automação	56
4.4.2.Integração do PIR HC - SR501 ao sistema da central de automação.....	59
4.4.3.Integração do módulo LDR ao sistema da central de automação.....	60
4.4.4.Integração do sensor de metano MQ4 ao sistema da central de automação.....	63

4.5. Implementação adicional de automações	67
4.6. Implementação da interface para dispositivos móveis	69
4.7. Estimativa do custo do sistema básico implementado.....	69
5. EXPERIMENTOS E RESULTADOS.....	72
5.1. Leitura dos sensores	72
5.1.1.Sensor de movimento PIR HC – SR501.....	73
5.1.2 Sensor de temperatura DS18B20+	73
5.1.3.Sensor de monóxido de carbono – MQ4	76
5.1.3.Módulo LDR	78
5.2. Acionamento dos atuadores.....	78
5.2.1. Broadlink Rm Mini 3	78
5.2.2. Sonoff	80
5.2.3.Sirene	80
5.3. Execução de perfis customizados	80
5.4. Acionamento por meio da interface móvel	82
5.5. Histórico de atividades	82
6. CONCLUSÕES E TRABALHOS FUTUROS.....	85
REFERÊNCIAS BIBLIOGRÁFICAS	

1. INTRODUÇÃO

Atualmente vive-se a era da quarta revolução industrial, denominada de indústria 4.0, termo definido pelo alemão Klaus Schwab para definir o processo de profundas mudanças que estão ocorrendo atualmente¹. Diferente da terceira revolução que estabeleceu a eletrônica como verdadeira modernização da indústria, essa nova versão vai além, priorizando dispositivos inteligentes e a conectividade. A qualidade da tecnologia produzida é fundamental na aceitação do produto e, como exemplos, têm-se empresas como a Microsoft, Google, Facebook IBM e Amazon.

A revolução é percebida por todos, não sendo necessário estar em contato direto com os novos meios de produção para ser afetado. É comum experimentar novas formas de consumos, aquisições de produtos e até mesmo manter relacionamentos com pessoas. A IoT (*Internet of Things*), inteligência artificial e robótica são campos onde pode-se facilmente identificar a mudança na vida das pessoas. A automação chega com força não somente dentro das fábricas, mas também em residências (domótica) e tem-se como exemplo os *smartphones* que auxiliam as pessoas cada vez mais em tarefas do cotidiano¹.

Durante essa revolução um dos setores que mais cresceu é o da automação residencial e segundo a AURESIDE (Associação Brasileira de Automação Residencial e Predial), isso se deve a vários fatores, como o significativo crescimento do mercado de IoT, a redução no custo dos sistemas computacionais, a expansão de tecnologias pelos fabricantes e uma crescente importância no monitoramento remoto das residências².

A AURESIDE também cita em um de seu relatório que, globalmente estava previsto um crescimento no mercado de automação residencial de 32 bilhões de dólares em 2015, projetando para 78 bilhões de dólares em 2022. Já no Brasil essa empresa aponta que há um potencial de mercado para 1,8 milhões de residências. No entanto, estima-se que somente 300 mil dessas residências já possuem um sistema de automação, deixando margem para um crescimento animador no ramo da domótica³.

Um sistema de automação residencial visa otimizar tarefas do cotidiano, economizando tempo e energia para os residentes. Tarefas como acender lâmpadas, ligar ar-condicionado, aspirar

¹ Site: <https://www.startse.com/noticia/nova-economia/60414/industria-4-0-entenda-o-que-e-quarta-revolucao-industrial>

² Site: <http://www.aureside.org.br/noticias/previsoes-para-o-mercado-global-de-aut.-residencial>

³ Site: <http://www.aureside.org.br/noticias/automacao-residencial--demanda-na-construcao-civil>

a casa e lavar as plantas, já podem todas ser feitas por meio de sensores e acionadores, atuando de forma inteligente, não sendo necessária a interferência humana.

Esse trabalho de monografia aborda um sistema de automação de baixo custo que visa acionar remotamente vários dispositivos presentes em uma residência, além de customizar rotinas de automação pré-estabelecidas, fornecendo maior conforto e segurança ao residente. Os acionamentos foram feitos por meio de computadores e dispositivos móveis como *smartphones e tablets* e as automações foram configuradas: a) com base em dados de entradas fornecidos por sensores, denominadas nesse trabalho, de reativas; e b) pela execução de perfis programados por usuários (customizados) simulando a presença de pessoas quando a residência está vazia. Esse projeto dá continuidade ao trabalho de monografia desenvolvido por Almeida (2018) e propõe mudanças que visam a redução do custo e execução de automações (reativas e perfis customizados).

1.1. Objetivos

O objetivo geral desse trabalho é propor uma alternativa de custo reduzido ao sistema de automação *wireless* com a possibilidade de programação de rotinas reativas e customizadas para aumentar a segurança da residência, inclusive na ausência de pessoas.

Com relações aos objetivos específicos, têm-se:

- a) Viabilizar o acionamento de atuadores via *wireless* com o uso de sistemas embarcados de baixo custo; e
- b) Possibilitar um aumento de segurança da residência com a programação de rotinas automatizadas reativas (baseada em dados sensoriais) e de perfis customizados pelo residente para serem processados quando não houver pessoas na mesma;

1.2 Motivação

A motivação para o desenvolvimento desse trabalho está relacionada ao potencial ocioso ainda a ser desenvolvido em automação residencial no Brasil e na tendência mundial prevista. Dessa forma, a domótica é uma área de desenvolvimento bastante atrativa para engenheiros eletricitistas.

Outro aspecto relevante para mencionar como estímulo no desenvolvimento de trabalho em automação residencial é o avanço da tecnologia com o crescente aumento de inteligência dos

componentes eletrônicos envolvidos, das possibilidades de conectividades entre eles e da diminuição dos custos com o aumento das demandas dos produtos e das tecnologias empregadas.

Um terceiro ponto que merece destaque é o potencial criativo dos profissionais dessa área pelo que atualmente tem disponível e as promessas de ampliação de ofertas de produtos no mercado da domótica.

Finalmente, ainda existe a contribuição da implementação e validação desta proposta de trabalho na formação acadêmica, aprimorando os conhecimentos adquiridos na disciplina TEEE – Sistemas Embarcados, a qual é oferecida regularmente desde o segundo semestre de 2016.

1.3 Metodologia

A metodologia empregada no desenvolvimento deste trabalho de monografia teve como ponto de partida um trabalho anterior na área que foi apresentado como monografia ao Curso de Engenharia Elétrica (ALMEIDA, 2018). No trabalho citado, o projeto fez uso de sistemas de automação, comandado via voz ou por um dispositivo móvel (*smatphone, tablet, etc.*) para realizar a ativação de um dispositivo de controle residencial (lâmpada, ar-condicionado, TV, etc) ou ler alguma variável de um sensor (temperatura).

Partindo desse desenvolvimento, foi proposto e desenvolvido um sistema de automação residencial que eliminou a presença da central usada no trabalho anterior, a qual foi substituída por um sistema embarcado de baixo custo. Esse sistema, além da questão de menor preço, teve como propósito viabilizar a programação de automações programadas para serem executadas de forma reativa à leitura de sensores que indiquem irregularidades ou para a execução de perfis programados. Esse último tipo consiste em tarefas para serem executadas em horários definidos, simulando a presença de pessoas na residência, quando a mesma encontra-se vazia. O objetivo foi maximizar a segurança da residência, inclusive contra eventos anormais e da tentativa de assaltos.

Para validar o desenvolvimento do sistema implementado foram feitos testes tanto de acionamento intencional pela pessoa da residência, para a verificação de forma isolada dos sensores e atuadores incorporados ao sistema, bem como ensaios com as automações reativas e de perfis customizados, este últimos programados para ocorrerem em horários pré-definidos.

1.4 Organização do trabalho

O trabalho está organizado em seis capítulos, incluindo o atual da introdução do tema. No Capítulo 2 é feita a fundamentação teórica, onde abordam-se os temas ligados a automação residencial como surgimento, evolução, mercado, além de descrever os elementos utilizados na construção de um sistema como sensores, atuadores, central de comunicação e possíveis métodos de comunicação. O Capítulo 3 trata da apresentação dos elementos que foram escolhidos para constituir o *hardware* e *software* do sistema proposto. O Capítulo 4 é dedicado ao detalhamento do projeto e as implementações do *hardware* e *software* do sistema de automação residencial no que diz respeito aos procedimentos feitos passo-a-passo. No capítulo 5 são realizados os experimentos nos dispositivos controlados e na execução de automações reativas e de perfis customizados pelo dono da residência para a comprovação da sua eficiência. Finalmente, no Capítulo 6 apresentam-se as conclusões e os trabalhos futuros que podem ser realizados a partir do atual estado alcançados com o desenvolvimento desta monografia.

2. FUNDAMENTAÇÃO TEÓRICA

A automação consiste na capacidade de dispositivos ou sistemas de se autorregularem independentemente da ação humana. O processo de mecanização ocorreu anterior ao desenvolvimento da automação que, apesar de disporem da utilização de novas tecnologias não devem ser confundidos. A mecanização consiste na utilização de ferramentas tecnológicas de forma a reduzir o esforço humano e acelerar procedimentos como a utilização de máquinas e dispositivos na produção agrícola, conforme ilustrado na Figura 1.

A mecanização de trabalhos manuais ocorre desde os tempos pré-históricos ao passo que invenções como moinhos e roda d'água, além da utilização de animais já mostra a criatividade humana para poupar esforço dos trabalhadores. No entanto, a automação e o seu desenvolvimento não ocorreu até meados do século XVIII, onde a evolução do capitalismo e a revolução industrial vivenciaram a necessidade de substituição do trabalho artesanal por algo mais eficiente e de maior produtividade. Dessa forma, máquinas mais precisas, rápidas e capazes de atuar sem intervenção humana, começaram a ser implementadas. (BORGES *et al.*, 2000).

Assim, a automação surgiu como uma forma capitalista de aumentar produção e lucro. Entretanto, a sua utilidade expandiu de tal forma que, atualmente, se encontra presente em diversas áreas como transporte, comunicação, medicina e até mesmo no conforto diário das pessoas (BORGES *et al.*, 2000).

Figura 1 – Trator sendo utilizado na produção agrícola.



Fonte: <https://www.cpt.com.br/cursos-mecanizacao-agricultura/artigos/conheca-o-beneficio-dos-tratores-na-agricultura>

2.1 Uma visão geral sobre a automação residencial

Inserido em um roteiro contemporâneo cada vez mais capitalista e com alta demanda de consumo, o ser humano percebeu a necessidade de investir na automação, que anteriormente era limitada ao cenário industrial, mas passa a fazer parte do residencial. Pode-se definir a automação residencial como a utilização da tecnologia para automatizar tarefas habituais realizadas nos lares. E por meio da utilização de sensores, atuadores e interfaces é possível proporcionar maior praticidade, segurança, economia e conforto para o morador⁴. Hoje essa automação pode ser vista desde dispositivos mais rotineiros como as máquinas de lavar e secar, capazes de se comunicar com smartphones e dispor de funções inteligentes, a dispositivos menos habituais como *drones*. Na Figura 2 é possível observar uma dessas secadoras.

Figura 2 – Lavadora e secadora de roupa Samsung Qdrive



Fonte: <http://www.fxreview.com.br/2018/10/265-samsung.html>

Como exemplo também se pode citar os aspiradores robotizados capazes de mapear e explorar toda uma casa de forma autônoma, enquanto a limpa, conforme ilustrado na foto da Figura 3, onde sua programação é feita pelo *smartphone*.

2.1.1 A evolução da automação residencial

As primeiras tecnologias definidas como automação residencial foram desenvolvidas na década de 1970 nos Estados Unidos. A comunicação ocorria por meio da própria rede elétrica

⁴ Site: <https://www.automatichouse.com.br/automacao-residencial/o-que-e-automacao-residencial>

Figura 3 – Robô Aspirador Xiaomi Roborock S50 2ª Geração



Fonte: https://http2.mlstatic.com/robo-aspirador-xiaomi-robotrock-s50-2-geraco-nota-fiscal-D_NQ_NP_700928-MLB29218514817_012019-F.webp

baseada no conceito de PLC (*Power Line Carrier*) e dispunha de comandos bem simples como acender e apagar lâmpadas remotamente. O sistema era bem limitado e quase não permitia a integração entre dispositivos. Os primeiros dispositivos que surgiram e foram utilizados em rede doméstica foram os módulos X-10 (MURATORI, 2014), mostrado na Figura 4.

Figura 4 – Módulos X-10 para automação residencial



Fonte: <https://pt.wikipedia.org/wiki/X10>

Entretanto, o desenvolvimento de novas tecnologias como a quarta geração da internet e seus respectivos protocolos, permite hoje a implementação de sistemas integrados em larga escala e até o surgimento de novos conceitos como a Iot e casas inteligentes (MONDEVAIN *et al.*, 2000).

A ideia da tecnologia IoT visa transferir todos os dispositivos do nosso cotidiano do mundo *offline* para o mundo *online*, dando origem ao que pode se chamar de uma casa inteligente, onde todos os dispositivos estão integrados entre si, permitindo a comunicação entre os mesmos e a tomada de decisões inteligentes. Como pode ser visto no exemplo da Figura.5.

Figura 5 – Dispositivos presentes em uma casa inteligente



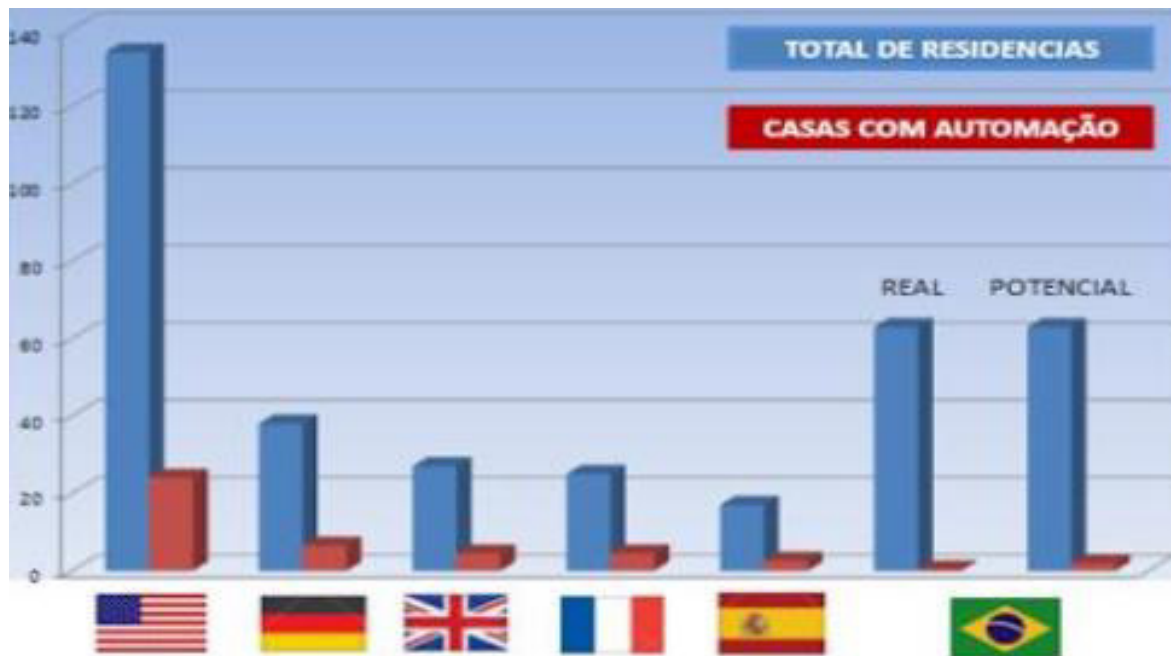
Fonte: <https://www.unaerp.br/documentos/2159-automacao-residencial-e-predial-e-a-internet-das-coisas/file>

2.1.2 O mercado da automação residencial

O desenvolvimento de tecnologias adequadas a automação residencial nas últimas décadas permitiu um aumento significativo dessa nova tendência ao redor do mundo, apesar de a inserção ao mercado não ter sido amplamente difundida em alguns países como o Brasil. Além disso, foi necessário superar algumas barreiras como a qualificação da mão de obra. Entretanto, como grande vantagem, o cenário da aplicação é promissor, pois o número de fabricantes teve um rápido crescimento nos últimos anos e o número de fornecedores mais que triplicou, conforme ilustra o gráfico mostrado na Figura 6⁵.

⁵ Site: https://www.aecweb.com.br/cont/a/os-desafios-do-mercado-da-automacao-residencial_8192

Figura 6 – Comparação dos principais mercados de automação residencial



Fonte: <http://www.smartautomacao.com.br/home/como-se-situa-o-brasil-no-mundo-da-automacao-residencial/>

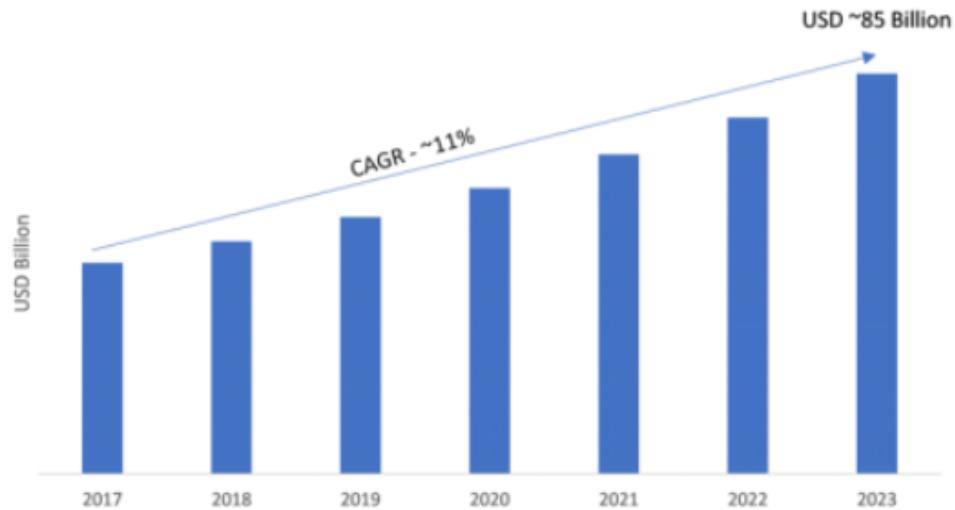
Diante da crescente popularidade do negócio e da aparente aceitação do público a essas novas tecnologias, que proporcionam cada vez mais conforto, agilidade, qualidade de vida e segurança na rotina de residentes. É previsto um crescimento de onze por cento no intervalo de 2017 a 2023, conforme ilustrado no gráfico da Figura 7⁶.

2.2 Sistemas embarcados e a automação residencial

Apesar de existirem várias definições para o termo sistemas embarcados ou sistemas embutidos, todas tendem a convergir para a mesma ideia. Um sistema baseado em uma placa microprocessada ou microcontrolada, sendo implementados em tamanhos reduzidos e que visam a execução de tarefas dedicadas e usualmente consomem uma quantidade de energia muito baixa. É possível encontrar essas placas presentes nas mais diversas aplicações como o controle de um forno micro-ondas ou até mesmo em sistemas bem mais complexos dispostos de várias camadas de hardware como aqueles encontrados em aeronaves comerciais (DINIZ, 2017).

⁶ Site: <https://www.reuters.com/brandfeatures/venture-capital/article?id=45598>

Figura 7 – Estimativa de crescimento do mercado de automação residencial



Fonte: <https://www.reuters.com/brandfeatures/venture-capital/article?id=45598>

2.2.1 Placas microcontroladas

As placas microcontroladas são dispositivos baseados em um *chip* microcontrolador, possuindo geralmente várias portas de entrada e saída, além de vários recursos incorporados ao *chip*. Em decorrência disso, são usualmente utilizados para aquisição de dados proveniente de sensores e para o acionamento de atuadores, como é o caso da placa ESP8266 (Figura 8).

Figura 8 – Placa ESP8266

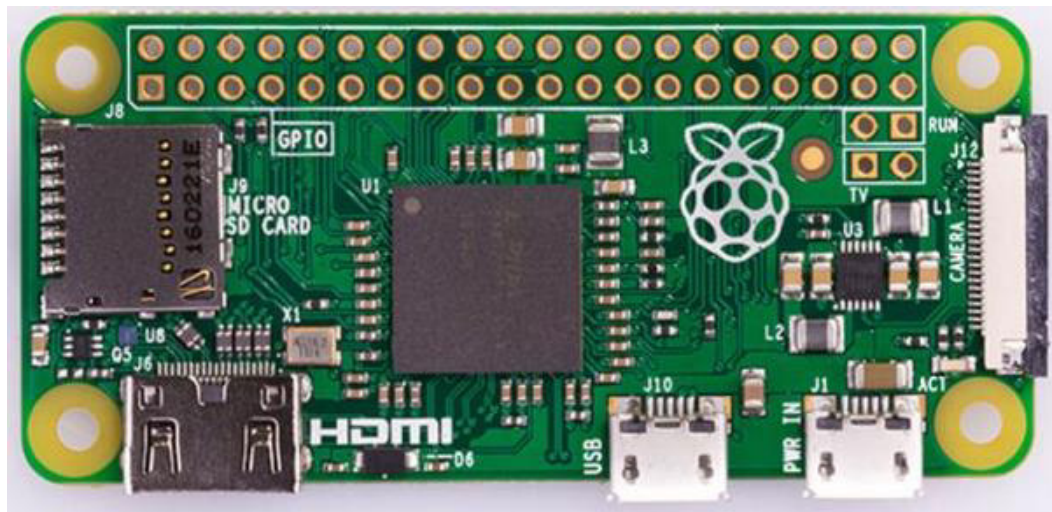


Fonte: <https://www.amazon.in/ESP8266-NodeMcu-WiFi-Development-Board/dp/B00UY8C3N0>

2.2.2 Placas microprocessadas

As placas microprocessadas são dispositivos baseados em *chip* microprocessador e, por isso, dispõem de uma capacidade maior de execução do que as microcontroladas. Em decorrência disso, são geralmente utilizadas como peças essenciais na integração de sistemas funcionando como gerenciadores de informações recebidas por outros dispositivos. Portanto, essas placas são indispensáveis em sistemas de automação, atuando como o “cérebro” do mesmo e são normalmente conhecidas como centrais de automação. Uma das placas embarcadas microprocessadas disponibilizadas no mercado com uma boa relação custo-benefício, ou seja, apresenta uma boa capacidade de processamento a um custo baixo é o Raspberry Pi Zero W de U\$ 10⁷ (Figura 9), atendendo as necessidades de implementação deste projeto e possuindo com principais características: a) baixo custo e boa capacidade de recursos e processamento; e b) comunicação Wi-Fi embutida.

Figura 9 – Raspberry Pi Zero W



Fonte: <https://tecnoblog.net/209672/raspberry-pi-zero-wifi/>

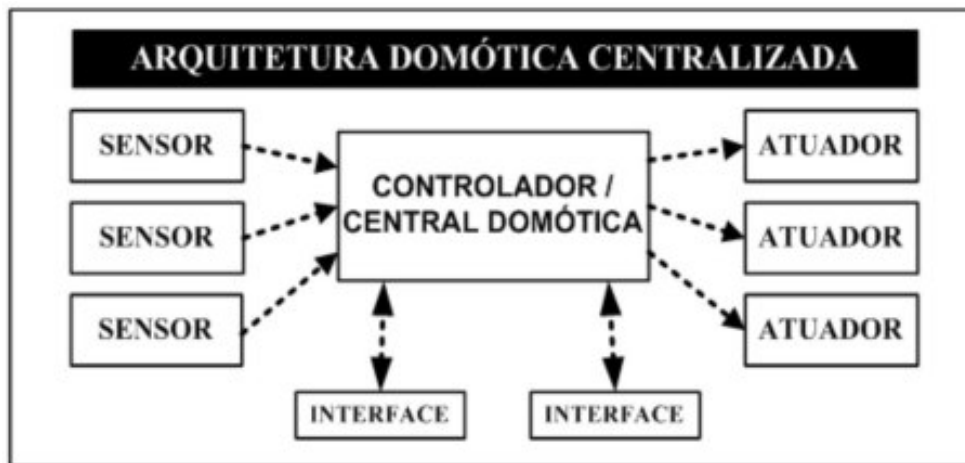
Devido apenas a questão de disponibilidade, foi utilizada a placa Raspberry Pi 3B de maior custo e maior capacidade de processamento em substituição a Zero W. Entretanto, a versão Zero W é totalmente aplicável em recursos e capacidade de implementação que a utilizada.

⁷ Site: <https://tecnoblog.net/209672/raspberry-pi-zero-wifi/>

2.3 Elementos de um sistema de automação residencial

Os elementos de um sistema de automação podem ser resumidamente definidos em três categorias: sensores, atuadores e a central de automação. Os sensores são responsáveis por coletar dados, a central por interpretar e processar esses dados e os atuadores por acionarem dispositivos, assim como a arquitetura ilustrada em diagrama de blocos na Figura 10.

Figura 10 – Arquitetura centralizada de uma central de automação



Fonte: <https://www.ebah.com.br/content/ABAAAAsK4AB/automacao-residencial-com-utilizacao-controlador-logico-programavel>

2.3.1 Sensores

Os sensores são elementos responsáveis por detectar de forma eficiente estímulos fornecidos pelo ambiente. Existem sensores que respondem a vestímulos diferentes, como pressão, temperatura, luz, etc. Por exemplo, eles são muito comuns em tarefas referentes à segurança ou controle de temperatura⁸. A Figura 11 mostra um sensor magnético utilizado para detectar abertura de porta e janela.

2.3.2 Transdutores

Os tradutores são dispositivos responsáveis por converter grandezas físicas como temperatura e pressão em uma elétrica que possa ser interpretada, como tensão ou corrente⁹.

⁸ Site: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>

⁹ Site: <https://www.pce-medidores.com.pt/sistemas/transdutores.htm>

Figura 11 – Sensor magnético de abertura de porta e janela



Fonte: <https://www.tudoforte.com.br/alarmes/sensores-magnetico/ecp-f105547-intruder-magnetico>

2.3.3 Atuadores

Os atuadores são dispositivos elétricos responsáveis pelo acionamento de outros equipamentos, como um relé que é responsável por ligar e desligar lâmpadas ou um motor responsável por movimentar cortinas. Esses são geralmente dispositivos que tendem a converter sinais elétricos em algum tipo de movimento ou outra ação qualquer¹⁰. Na Figura 12 é possível observar um módulo para o acionamento de cargas.

2.3.3 Central de automação

Figura 12 – Módulo atuador para o controle de cargas elétricas



Fonte: <http://www.msato.com.br/produtos/power-relay/detalhe>

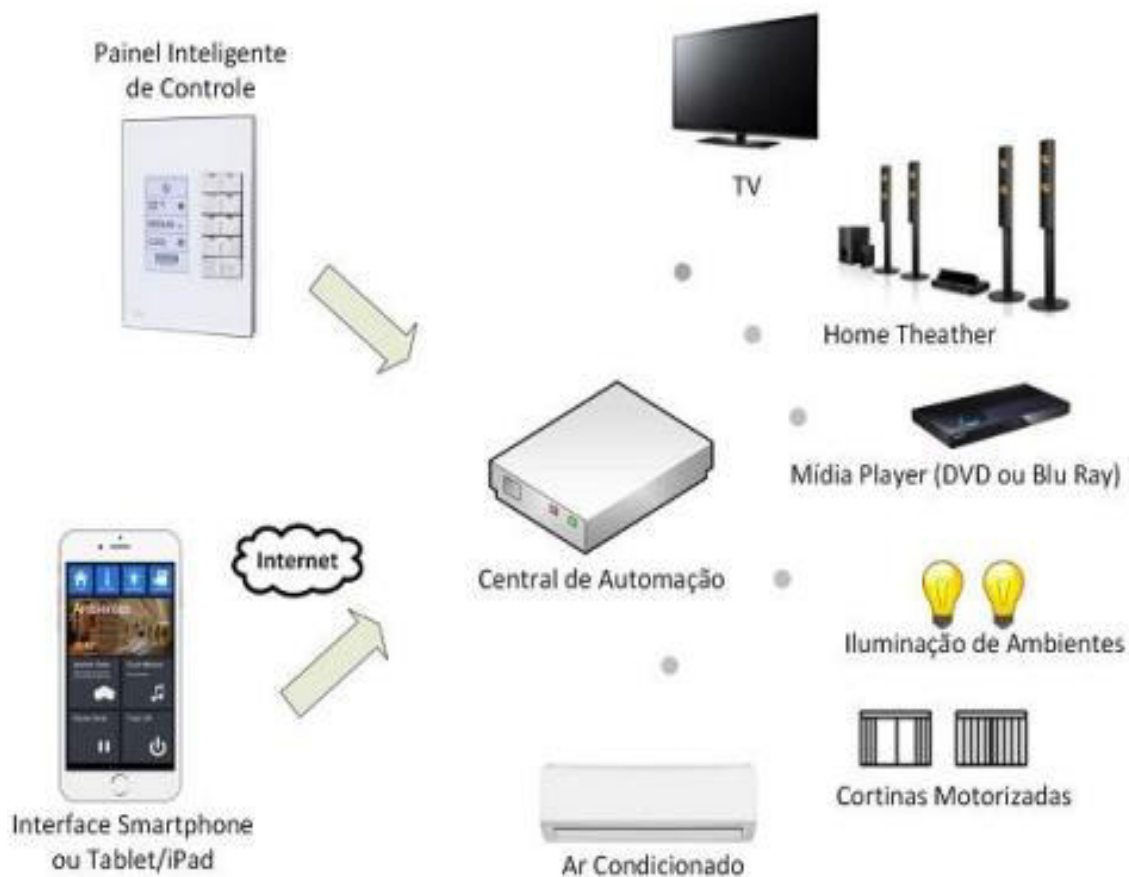
A central de automação funciona como o “cérebro” do sistema, recebendo e interpretando dados recebidos pelos sensores, pelas interfaces dos usuários e enviando comandos aos atuadores.

¹⁰Site: <https://www.mecanicaindustrial.com.br/374-para-que-servem-os-atuadores/>

A central funciona como o ponto de comunicação entre todos os dispositivos, podendo se comunicar pela rede por meio de um cabo ethernet ou de forma *wireless* (ALMEIDA, 2018). Além disso, é possível trabalhar com diferentes topologias, a de arquitetura centralizada, onde todos os dispositivos são concentrados em um único ponto da casa e os condutores para acionamento partem desse ponto. Ou a arquitetura distribuída onde os sensores, atuadores e controladores estão distribuídos pela residência e se conectam por alguma rede de comunicação¹¹.

Veja na Figura 13 um exemplo de central de automação funcionando no modelo de arquitetura centralizada.

Figura 13 – Exemplo de central de automação baseada em arquitetura centralizada



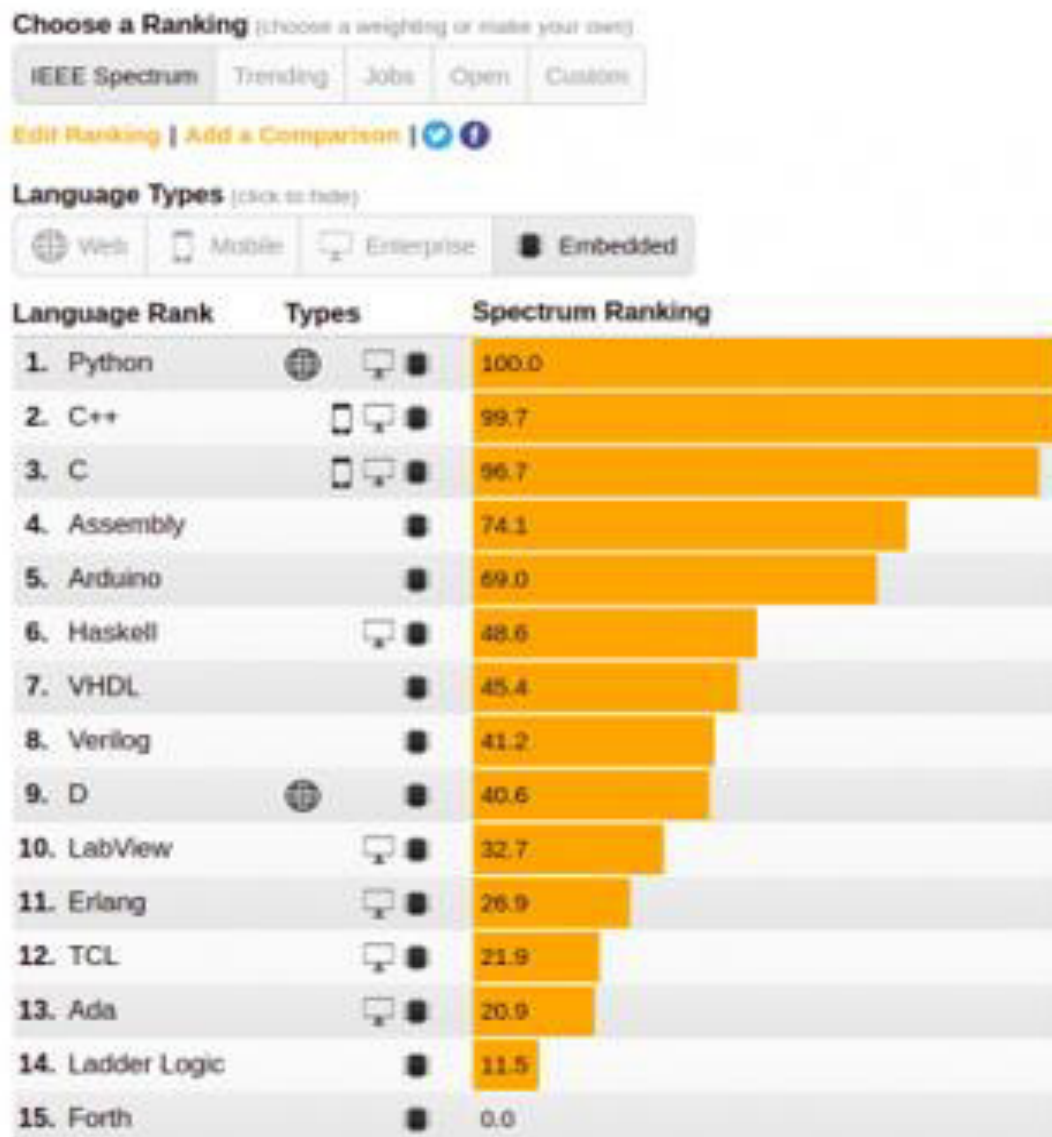
Fonte: <http://integrahaus.com.br/produtos/automacao-residencial/>

¹¹Site: <http://labdegaragem.com/m/discussion?id=6223006%3ATopic%3A285953>

2.4. Linguagens de programação usuais em sistemas de automação residencial

A linguagem utilizada é dependente dos equipamentos utilizados. Atualmente, a adesão de placas como ESP8266, Arduino e Raspberry Pi promoveu um crescimento significativo da utilização de linguagens como o C/C++ e, principalmente, o Python. No trabalho apresentado, a central de automação microprocessada utiliza um sistema baseado em Python ao passo que os microcontroladores que a interliga aos sensores e atuadores utilizam de linguagem C/C++. Na Figura 14 é possível ver um ranking quantitativo das linguagens utilizadas em sistemas embarcados (ALMEIDA, 2018).

Figura 14 – Ranking quantitativo de linguagens utilizadas em sistemas embarcados



2.4.1 Linguagem C/C++

A linguagem C é uma linguagem estruturada criada inicialmente com o propósito de ser usado no desenvolvimento de uma nova versão do sistema operacional Unix. Além disso, é extremamente portátil e pode ser usada em diferentes plataformas, por isso, acaba possibilitando a utilização em diferentes campos como desenvolvimento de *softwares*, jogos e programação de microcontroladores e microprocessadores em geral¹².

A linguagem C++, não é exatamente a mesma que a C, no entanto, foi baseada nesta, procurando manter o máximo de compatibilidade possível. A intenção foi de melhorar a linguagem trazendo alguns aspectos de linguagem de baixo para o alto nível, adicionando à linguagem C algumas características como a programação voltada para classes, recursos que a classifica como orientada a objeto¹³.

2.4.2 Linguagem Python

Python é uma linguagem interpretada de alto nível que também possui suporte à programação orientada a objeto. É uma linguagem ideal para iniciantes, pois possui uma sintaxe simples, atua em várias plataformas e atualmente é utilizada em vários campos, como desenvolvimento de jogos e outros sistemas inteligentes. Inicialmente foi idealizada para suprir as necessidades da linguagem ABC que trabalhava com o sistema operacional Amoeba. O nome tem origem em um gosto pessoal do principal desenvolvedor, Guido Van Rossum, que se inspirou na série de comédia britânica Monty Python¹⁴.

2.5 Meios de comunicação entre elementos de um sistema de automação residencial

Para efetuar as tarefas de automação de um sistema é necessário que os elementos deste se comuniquem. A central de automação necessita processar os dados detectados pelos sensores e enviar comandos de acionamentos para os atuadores. As formas de comunicação entre dispositivos do sistema podem ser divididas em três categorias: cabeada, sem fio (*wireless*) e híbrida (utiliza dos dois métodos anteriores). Optando-se pela categoria *wireless* existem vários métodos com diferentes características (ALMEIDA, 2018).

¹² Site: <http://linguagemc.com.br/o-que-e-linguagem-c/>

¹³ Site: <http://www.tiexpert.net/programacao/c/introducao-cpp.php>

¹⁴ Site: <http://mindbending.org/pt/a-historia-do-python>

2.5.1 Cabeado

Os sistemas cabeados normalmente são implementados quando fazem parte do projeto inicial da residência, pois necessitam de mais espaço e estrutura, além disso, pode custar cerca de cinco a dez por cento do preço da obra, o que é relativamente mais caro do que sistemas sem fio. Todavia, essa estrutura também possui suas vantagens, são mais robustas e confiáveis, possuem menor latência e não apresentam problemas relativos ao alcance do sinal¹⁵.

2.5.2 Bluetooth

Apesar de ser uma opção de comunicação sem fio com boa velocidade de transmissão, o sistema *bluetooth* não costuma ser aplicado em automação residencial, pois apresenta algumas restrições. Essa comunicação é considerada de curto alcance e possui limitação no número de dispositivos que podem comunicar entre si. Diante disso, é mais utilizada na comunicação de *smartphones* e seus *gadgets*¹⁶.

2.5.3 Radiofrequência

Esse estilo de comunicação é recentemente um dos mais utilizados em sistemas de automação, principalmente em áreas industriais, pois possui um consumo de energia pequeno, baixo custo e pode trabalhar num com redes *mesh*. Nessa rede cada um dos elementos do sistema opera como um repetidor de forma que existem inúmeros caminhos para que a mensagem chegue ao destinatário, aumentando a confiabilidade do sistema, assim como mostrado na Figura 15. Um dispositivo muito utilizado nessa aplicação é o *Xbee* (ALMEIDA, 2018).

2.5.4 Wi-Fi

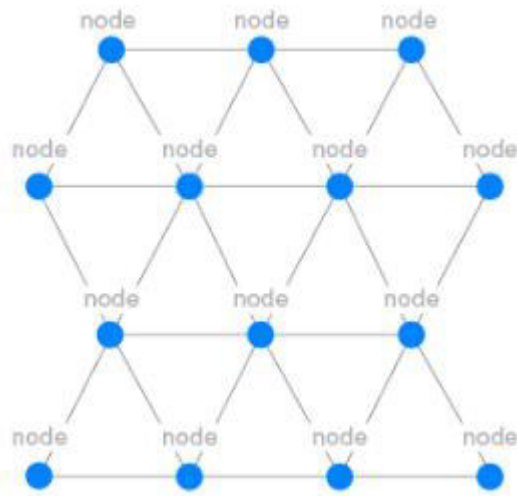
Esse estilo de comunicação tem se tornado cada vez mais comum dentro do ambiente da automação residencial. Pois, o desenvolvimento de novas tecnologias como a então em progresso quinta geração da internet móvel e conceitos como a internet das coisas visam cada vez mais converter o mundo *offline* em *online*¹⁷. Dessa forma o surgimento de eletrodomésticos aptos a se comunicar com a internet aumenta de forma impressionante. Atualmente, já existem disponíveis lâmpadas, máquinas de lavar, aspiradores de pó, televisão, geladeira, etc. A tendência é que essa

¹⁵ Site: <https://www.neocontrol.com.br/news/casa-automatizada/#automacao-cabeada>

¹⁶ Site: <https://www.devmedia.com.br/introducao-a-comunicacao-bluetooth-no-android/27636>

¹⁷ Site: <https://tecnoblog.net/263907/o-que-e-internet-das-coisas/>

Figura 15 – Exemplo de rede *mesh*



Fonte: <http://www.decom.ufop.br/imobilis/rede-mesh-bluetooth-uma-introducao/>

variedade continue a aumentar e a facilidade de se utilizar microprocessadores para a comunicação com esses dispositivos faz com que automação residencial caminhe a passos largos nessa direção.

2.6 Proposta desenvolvida neste trabalho

O presente trabalho pretende dar continuidade ao projeto desenvolvido inicialmente por Almeida (2018), onde o foco esteve na utilização do *Hass.io* para o controle remoto de dispositivos da residência e na utilização do *gadget* Alexa, explicado posteriormente, como interface de controle.

Dessa forma, o objetivo nesta monografia é disponibilizar uma variedade de sensores pela residência de forma que serão usados como dados de entrada para o acionamento de rotinas de automação pré-definidas, e não apenas por comando manual. Esse aspecto possibilita a criação de perfis de acionamentos customizados para finalidades como, por exemplo, acionamento de lâmpadas, TV, etc, em horários pré-definidos, de modo a simular a presença da pessoa na casa, enquanto seu dono estiver ausente. Além disso, propõe a utilização como central uma placa Raspberry Pi Zero W de baixo custo em substituição ao RPi3B, visando a redução de custo no sistema completo. E por fim, a criação de uma interface de controle para dispositivos móveis.

3. **HARDWARE E SOFTWARE EMPREGADOS NO SISTEMA DE AUTOMAÇÃO RESIDENCIAL**

No presente capítulo, as ferramentas utilizadas na implementação do sistema de automação são descritas, incluindo os componentes de hardware e software empregados. Os dispositivos foram divididos em categorias, onde cada uma tem sua devida importância para o funcionamento da automação como um todo.

3.1. **Central de automação e interface física**

A parte principal do sistema que corresponde a central de automação foi implementada em um sistema embarcado Raspberry Pi 3B (RPi3B), pois este permite a instalação de um *software*, desenvolvido por hobistas, ideal para essa aplicação. Esse *software* funciona como uma interface de controle remoto dos sensores e atuadores que são acionados pelo usuário, além de integrar diversos outros recursos. O dispositivo Alexa Amazon, por outro lado, pode funcionar como uma interface física, sendo acionada por comandos de voz.

Por uma questão de conveniência o modelo RPi 3B foi utilizado (Figura 16), pois esse se encontrava disponível no **LRC (Laboratório de Robótica Móvel e Comunicação sem Fio)**, evitando custos extras. No entanto, ele não é essencial para o funcionamento do sistema uma vez que possui uma capacidade de processamento considerada alta para a aplicação proposta. Esse

Figura 16 – Raspberry Pi 3 modelo B (RPi 3B)



sistema possui quatro núcleos, o que o torna ideal para aplicações multithreading. A aplicação proposta não necessita desse hardware ampliado, por isso, o modelo Zero que teve sua primeira versão lançada em 2015, a US\$ 5,00, com um processador single-core de 1 GHz e uma memória RAM de 1 GBytes é suficiente para o processamento do sistema de automação, o que torna o ideal para uso neste trabalho, considerando custos e consumo menores (FERREIRA, 2018).

3.1.1 Placa Raspberry Pi 3B

Devido as justificativas apresentadas, a placa RPi 3B foi utilizada como a central do sistema, pois é ideal para a execução das tarefas de automação, já que o *Home Assistant* e *Hass.io*, que serão descritos mais adiante, foram feitos de forma a funcionar idealmente nessa placa, bem como na que seria usada como original do projeto (Raspberry Pi Zero W).

A placa RPi 3B possui uma alta capacidade de processamento, com frequências de até 1,2 GHz, arquitetura 64 *bits* e quatro núcleos, além de incorporar um *chip* de comunicação Wi-Fi embutido. Apesar de possuir um número de 26 de pinos destinados a entrada e saída ou GPIO (*General Purpose Input Output*) de um total de 40 (outros são usados para alimentação +5V, +3,3V e GND), esse sistema embarcado é empregado neste trabalho somente como uma central de automação. Assim, os GPIOs não serão utilizados na leitura de sensores e acionamento de atuadores, pois a necessidade de atuar em diferentes regiões do ambiente com conexões físicas torna-se inviável. Dessa forma, a leitura e a escrita dos dispositivos envolvidos na automação residencial são feitas por microcontroladores, de baixo custo, acoplados a cada elemento que se comunica com a placa central RPi 3B. Na Tabela 1 é possível verificar as especificações da placa (RIBEIRO, 2017).

3.1.2 Amazon Echo Dot

A Amazon Echo Dot, popularmente conhecida como Alexa, é uma assistente virtual desenvolvida pela empresa Amazon com capacidade de controlar os dispositivos de uma residência através de comandos de voz. Ela se mostra capaz de realizar chamadas, reproduzir músicas, enviar comandos de acionamento, fazer chamadas a serviços de Uber, entre outras aplicações possíveis¹⁸.

¹⁸ Site: <https://www.techtudo.com.br/noticias/2018/05/tudo-sobre-amazon-echo-dot-veja-se-funciona-no-brasil-e-ficha-tecnica.ghtml>

Tabela 1 – Especificações técnicas do Raspberry Pi 3

Componente	Especificação
Processador	ARMv8 Quad-Core 64 bits (1,2 GHz)
Processador de vídeo	Broadcom VideoCore IV
Memória RAM	1 GB LPDDR2 (900 MHz)
Adaptador Wi-Fi	802.11n
Bluetooth	4.1 (BLE)
Quantidade de pinos	40
Interface para camera	CSI – Câmera Serial Interface
Interface para display	DSI – Display Serial Interface/ HDMI – HIGH DEFINITION MULTIMEDIA INTERFACE

Fonte: (RIBEIRO, 2017)

A Alexa pode ser acionada por meio do comando de voz do usuário e se comunica com outros dispositivos por meio de Wi-Fi e *bluetooth*. Apesar de ter sido utilizada no sistema original, desenvolvido por Almeida (2018), no presente trabalho abre-se mão desta central como uma forma de economizar gastos e proporcionar a programação de perfis customizados para atuarem em horários pré-definidos no RPi 3B. Dessa maneira, a interface proporcionada pela Amazon Echo Dot no projeto original é substituída neste trabalho de monografia por *smartphones*. A Figura 17 apresenta uma foto desse dispositivo.

3.2. Software e Protocolos empregados

Para o funcionamento do sistema como um todo, que inclui a central localizada no RPi 3B e os dispositivos envolvidos na automação residencial, foi necessário a integração do *hardware* com alguns *software* como a IDE do Arduíno e o *Home Assistant*, além de conectá-los (central e dispositivos) com o protocolo MQTT, ideal para aplicações IoT.

3.2.1. Home Assistant e o Hass.io

O *Home Assistant* é um programa escrito em Python, podendo rodar em vários sistemas e fornecendo a habilidade de monitorar e automatizar dispositivos. O *Hass.io* é um sistema operacional formado por uma combinação do *Home Assistant* e outras ferramentas, resultando numa solução *all-in-one* e possuindo uma interface de gerenciamento que pode ser acessada pelo

Figura 17 – Amazon Echo Dot



Fonte: <https://www.techtudo.com.br/noticias/2018/05/tudo-sobre-amazon-echo-dot-veja-se-funciona-no-brasil-e-ficha-tecnica.ghtml>

Home Assistant. Além, de funcionar em um Raspberry e outras plataformas sem a necessidade da instalação prévia de outro sistema operacional¹⁹.

3.2.2. Arduino IDE

O IDE do Arduino é um *software* de código aberto escrito em Java e integrado à placa Arduino, funciona em diferentes sistemas operacionais e pode ser usado para fazer o *upload* de códigos em C/C++ na placa²⁰. No atual trabalho, foi utilizado justamente para fazer o carregamento de códigos na placa ESP8266, a qual será comentada mais adiante. A interface da placa pode ser vista na Figura 18.

3.2.3. MQTT broker

O protocolo MQTT (*Message Queuing Telemetry Transport*), é um protocolo de comunicação entre máquinas, desenvolvido pela IBM no final da década de 90. Esse programa é leve e baseado no protocolo TCP/IP, tendo-se tornado a principal alternativa no desenvolvimento de aplicações envolvendo a internet das coisas²¹.

¹⁹ Site: <https://www.home-assistant.io/faq/ha-vs-hassio/>

²⁰ Site: <https://www.arduino.cc/en/Main/Software>

²¹ Site: <https://engprocess.com.br/mqtt-broker/>

Figura 18 – Interface Arduino IDE



```

piscar_led | Arduino 1.8.8
Arquivo Editar Sketch Ferramentas Ajuda

piscar_led

//ItKindaWorks - Creative Commons 2016
//github.com/ItKindaWorks
//
//Requires PubSubClient found here: https://github.com/knolleary/pubsubclient
//
//ESP8266 Simple MQTT light controller

#include <PubSubClient.h>
#include <ESP8266WiFi.h>

//EDIT THESE LINES TO MATCH YOUR SETUP
#define MQTT_SERVER "192.168.0.8"

```

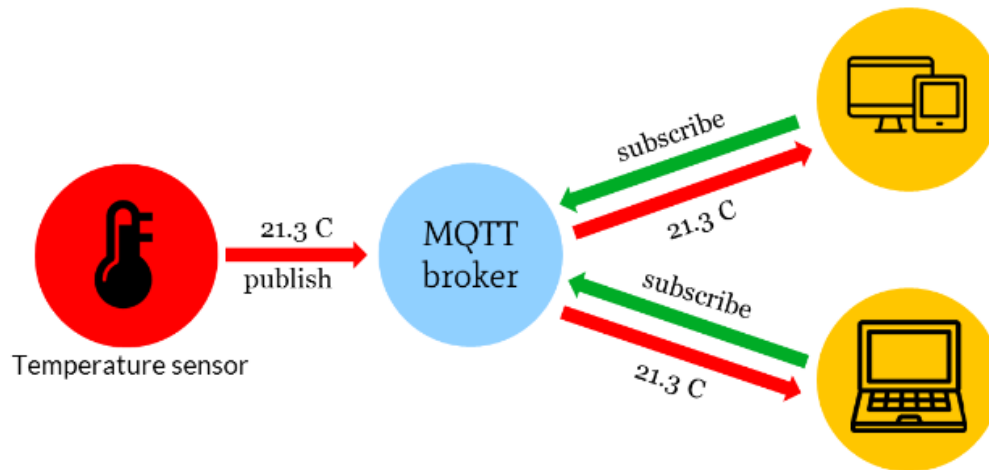
Fonte: Autor

O protocolo permite a criação de tópicos de forma que clientes podem escrever (*publish*) e ler (*subscribe*), fazendo inscrições nos mesmos. O *broker* funciona como uma forma de mediador, gerenciando as publicações e subscrições, assim como ilustrado na Figura 19, onde um termômetro pode receber operações de publicações (escrita de configurações) e fornecer a temperatura em subscrições (leitura do valor de temperatura) com um smartphone e um notebook mediado através do MQTT broker que mantém a inscrição do tópico relacionado a estes pedidos.

3.3 Atuadores

Nessa seção descrevem-se os atuadores utilizados para o acionamento dos dispositivos da residência, como ar condicionado, lâmpadas, sirenes, etc. Para isso foi necessário a utilização de interruptores, controles remotos inteligentes e microcontroladores para fazer a operação dos mesmos.

Figura 19 – Exemplo de funcionamento do fluxo de dados no protocolo MQTT



Fonte: http://istsos.org/en/trunk/doc/ws_mqtt.html

3.3.1 Placa ESP8266

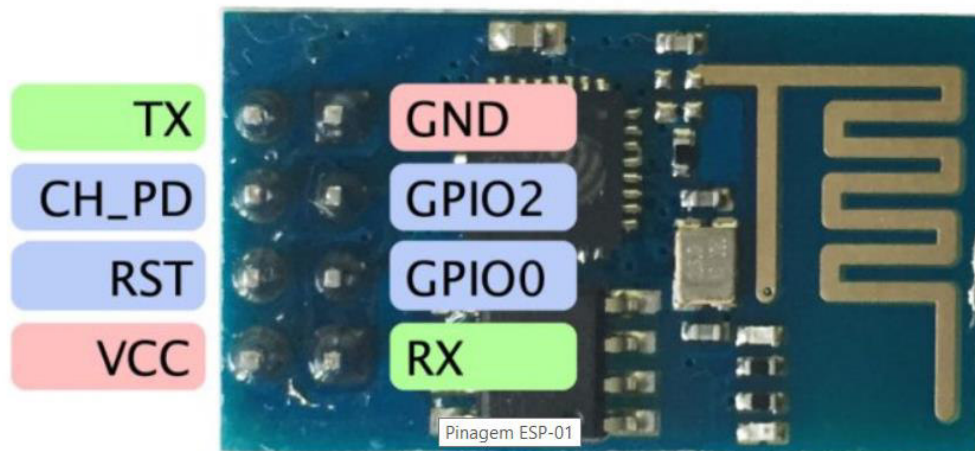
O *chip* ESP8266 foi desenvolvido pela empresa chinesa Espressif Systems, sendo criado com o intuito de solucionar o problema da comunicação Wi-Fi com microcontroladores e, atualmente, é largamente utilizado nas aplicações envolvendo IoT (KOLBAN, 2016). Para facilitar as mais diversas aplicações, os fabricantes desse *chip* de módulo Wi-Fi desenvolveram vários modelos de placas.

No presente trabalho, além da utilização do modelo ESP01 foi também utilizado o NodeMCU, comentado mais adiante. Ambas as placas funcionam com uma tensão de 3,3V, no entanto só a segunda possui regulador de tensão. A ESP01 é a versão mais comum desse *chip*, a primeira utilizada por entusiastas de IoT²², contendo o mínimo necessário de oito pinos, não possuindo pinos analógicos e somente dois digitais de propósito geral, além dos usados em comunicação UART (RX e TX). Essa placa pode ser vista na Figura 20.

Essa versão do ESP8266 não possui uma interface USB (*Universal Serial Bus*), o que requereu o uso de módulo que converta o bus USB em UART para a gravação de rotinas no *chip*, como mostrado na Figura 21.

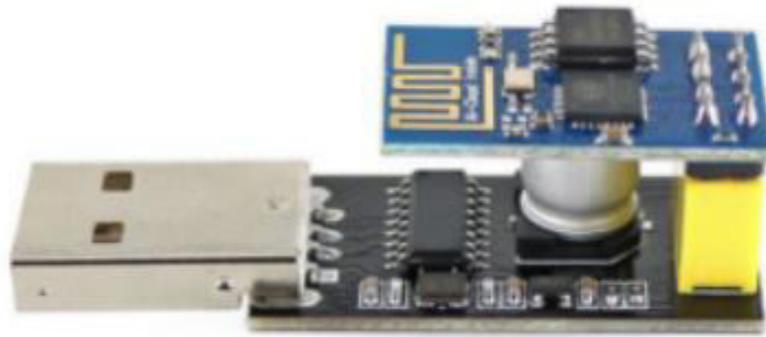
²² Site: <https://www.filipeflop.com/blog/guia-do-usuario-do-esp8266/>

Figura 20 – ESP8266 placa ESP01



Fonte: <https://www.filipeflop.com/blog/guia-do-usuario-do-esp8266/>

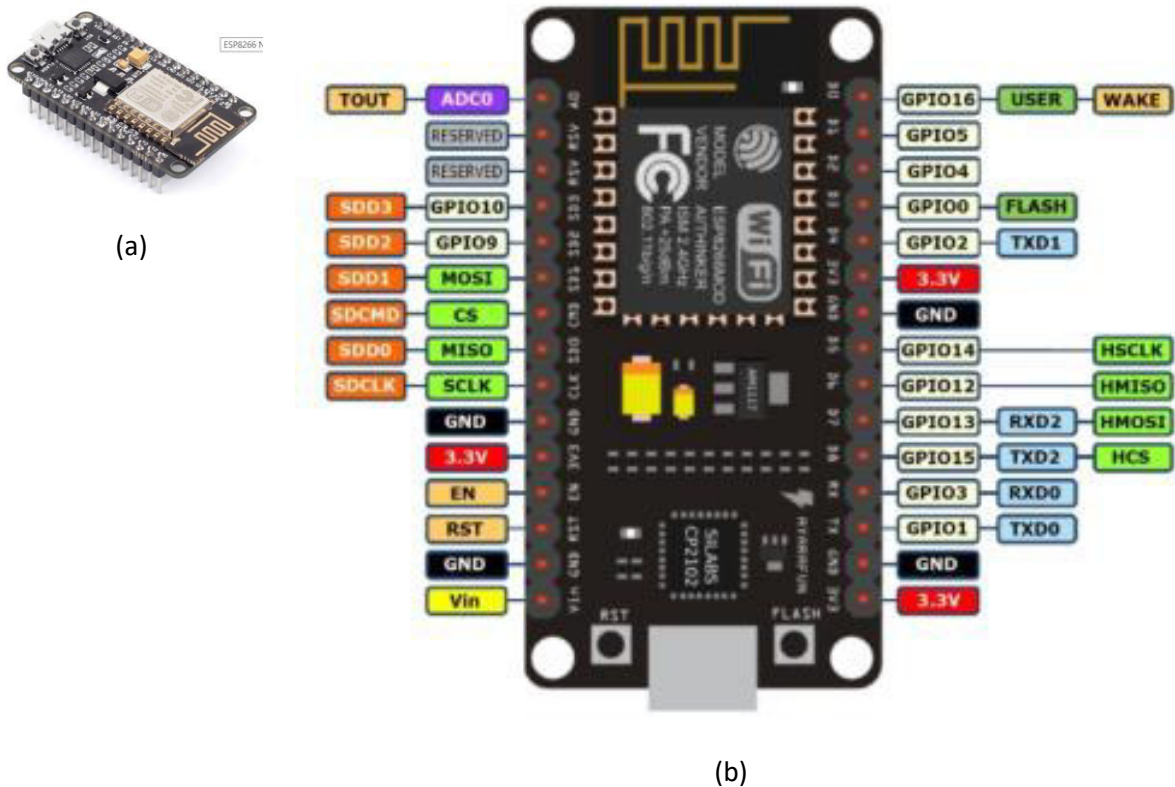
Figura 21 – Microcontrolador ESP01 com módulo adaptador USB-UART.



Fonte: <https://www.filipeflop.com/blog/guia-do-usuario-do-esp8266/>

A utilização da placa NodeMCU se mostrou ideal, pois é uma placa de desenvolvimento completa, possuindo conexão micro-USB e um número maior de pinos, permitindo a conexão de mais de 2 dispositivos simultaneamente e a leitura de sinais analógicos. A placa foi utilizada nesse projeto para a leitura de um sensor analógico e também permite a programação de forma bem mais eficiente uma vez que não precisa da utilização de módulos adicionais. Além disso, possui comunicação I2C e SPI, podendo ser utilizadas em trabalhos futuros. É interessante ressaltar que apesar de mais completa, as placas podem ser encontradas por aproximadamente o mesmo preço, cerca de vinte reais (valor médio em sites que comercializam o produto em 2019). Na Figura 22 essa placa pode ser vista.

Figura 22 – ESP8266 NodeMCU: a) Foto e b) Pinagem



Fonte: <https://www.filipeflop.com/blog/guia-do-usuario-do-esp8266/>

As placas ESP8266, foram utilizadas para a leitura de sensores e acionamento dos atuadores por meio da rede Wi-Fi. Estas estão presentes em *Sonoff* e são utilizadas para o acionamento de *relays*, cada um dos sensores também está acompanhado de uma delas. O sensor MQ4 (comentado adiante), exclusivamente, está conectado ao NodeMCU por ser analógico. As placas também são as responsáveis por se comunicar com a central e as rotinas das mesmas podem ser programadas por meio da IDE de desenvolvimento do Arduino.

3.3.2 Broadlink Rm Mini 3

Esse dispositivo foi desenvolvido para emitir sinais infravermelhos e funcionar como um controle remoto para interface com equipamentos que dispõem dessa tecnologia na residência, o qual pode ser visto na Figura 23. O aparelho possui comunicação Wi-Fi e se comunica diretamente

Figura 23 – *Broadlink Rm Mini 3*

Fonte: <https://cocoopower.com/products/broadlink-ir-control-hub-smart-home-wi-fi-enabled-infrared-universal-remote?variant=8153993576493>

com a central do sistema de automação, podendo controlar praticamente qualquer dispositivo que possua tecnologia infravermelho como TV, ar condicionado, *home theater*, etc²³.

3.3.3 *Sonoff Basic*

O *Sonoff* é um interruptor inteligente com comunicação Wi-Fi, o que permite a troca de dados com a central. É constituído basicamente por um ESP8266, um relé e um regulador de tensão AC-DC capaz de ser alimentado com tensões de entrada AC de 90V a 250V²⁴. Na atual aplicação é

Figura 24 – *Sonoff* instalado na residência

Fonte: ALMEIDA,2018

²³ Site: <http://www.ibroadlink.com/rmMini3/>

²⁴ Site: <http://blogmasterwalkershop.com.br/automacao/conhecendo-o-sonoff-rele-wifi-para-automacao-residencial/>

utilizado ao lado de um interruptor convencional para o controle de lâmpadas, como mostrado na Figura 24.

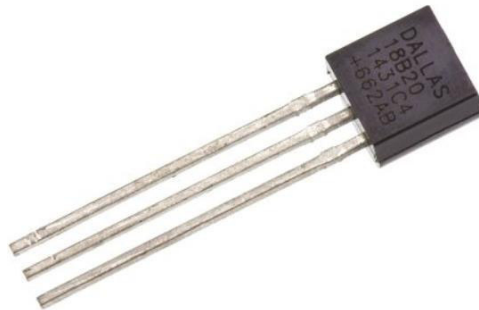
3.4 Sensores

Nessa seção são apresentados os sensores que neste projeto foram espalhados pela residência, visando utilizá-los como entrada de dados para automação de alguns processos. Cada um dos sensores pode vir a acionar uma rotina diferente, a depender dos dados lidos.

3.4.1 Sensor de temperatura DS18B20+

O DS18B20+ funciona como um termômetro digital de alta precisão, fornecendo medidas de temperatura em graus Celsius com nove *bits* de precisão. Possui um alcance de $-55\text{ }^{\circ}\text{C}$ a $+125\text{ }^{\circ}\text{C}$, com um desvio padrão de $\pm 0,5^{\circ}\text{C}$ entre as temperaturas de -10°C a $+85^{\circ}\text{C}$ ²⁵. Além disso, comunica usando o protocolo *One Wire*, o que é ideal para o funcionamento do sistema, levando em consideração as limitações do ESP01 que possui apenas 2 entradas digitais, uma delas requerida

Figura 25 – Sensor DSB18B20+



Fonte: <https://ae.rsdelivers.com/product/maxim/ds18b20/maxim-ds18b20-temperature-sensor-55-125-c-05c-1/5402805>

pelo protocolo usadas para ler a informação digital da temperatura. Esse sensor pode ser usado para acionar rotinas de automação destinadas à refrigeração de ambientes, como mostrado na Figura 25.

²⁵ Site: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

3.4.2 Sensor de movimento PIR HC-SR501

O sensor PIR HC-SR501 opera com uma tensão de 5V a 20V, detecta movimentos a uma distância de até 7 metros e possui uma abertura de detecção de 120 graus. Ele se comunica com o microcontrolador enviando um sinal de 3,3V através do seu pino de saída, quando detecta um movimento. Também possui dois potenciômetros que permitem o ajuste do *delay* associado ao tempo em que o pino de saída ficará em nível alto ao detectar algo (podendo variar de 5 segundos a dois minutos e meio) e também à sensibilidade com relação à distância máxima de detecção (podendo variar de 3 a 7 metros)²⁶.

No sistema proposto neste trabalho, o sensor tem a função de detectar movimentos na sala

Figura 26 – Sensor PIR HC-SR501



Fonte: <https://www.byteflop.com.br/sensor-de-presenca-pir-hc-sr501>

de estar, onde está localizada a porta principal da residência e enviar um sinal para acionamento de uma sirene, caso a respectiva rotina de automação esteja ligada. A foto deste sensor é mostrada na Figura 26.

3.4.3 Sensor de luz em módulo LDR

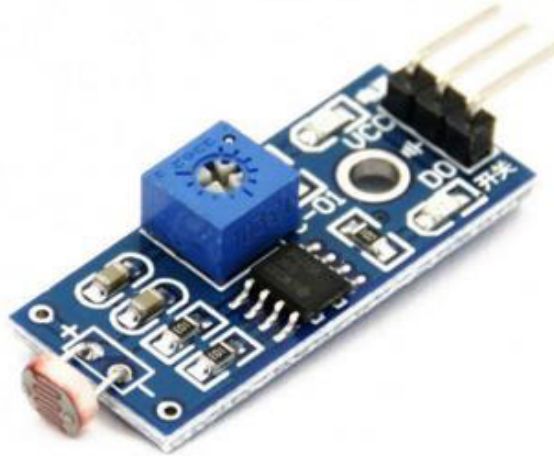
O LDR (*Light Dependent Resistor*) é um potenciômetro cuja resistência varia de acordo com a intensidade da luz. O seu módulo possui um resistor limitador de corrente embutido e,

²⁶ Site: <https://www.mpja.com/download/31227sc.pdf>

usualmente, pode funcionar ou com um pino digital, situação em que envia somente sinais de detecção de luz, ou com um pino analógico, quando informa a intensidade da luz detectada²⁷.

Neste sistema, funciona de forma a implementar uma rotina de segurança para o acionamento de lâmpadas em horários pré-definidos em um perfil programado para executar, quando da ausência

Figura 27 – Módulo LDR



Fonte: <https://www.eletrogate.com/modulo-sensor-de-luminosidade-ldr>

de moradores na residência. O módulo pode ser visto na Figura 27.

3.4.4 Sensor de Metano – MQ4

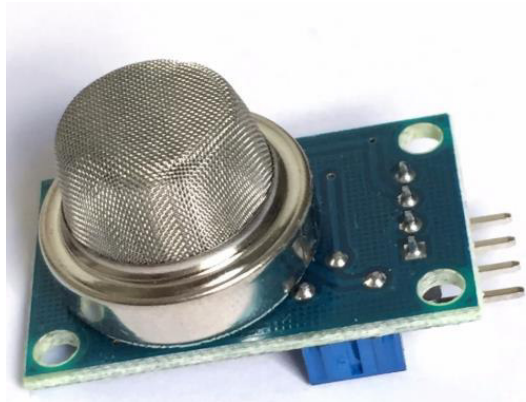
O MQ4 é um sensor de metano e gás natural de alta sensibilidade, além de possuir uma baixa sensibilidade ao álcool e à fumaça. Funciona com uma tensão de entrada de 5V, transmitindo sinais por meio de uma saída analógica. O ESP01 não possui um pino analógico, portanto o NodeMCU realiza as leituras desse sensor²⁸.

Neste projeto, o sensor (Figura 28) se encontra localizado na cozinha e avisa os residentes por meio de uma sirene (simulada por um *buzzer*) caso algum vazamento ocorra, de modo a que se possa tomar a providência necessária para sanar o vazamento, como a chave de entrada do gás.

²⁷ Site: <https://www.filipeflop.com/produto/sensor-de-luz-ldr/>

²⁸ Site: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-4.pdf>

Figura 28 – Sensor de metano MQ4



Fonte: <https://produto.mercadolivre.com.br/MLB-923102206-modulo-sensor-mq-4-mq4-gas-metano-butano-glp-fumaca-arduino- JM?quantity=1>

3.4.5 Outros componentes

Além dos componentes já citados, também foram usados alguns dispositivos auxiliares para o funcionamento do sistema de automação proposto, como fontes de 9V (alimentação de alguns

Figura 29 – Regulador de tensão LM2596

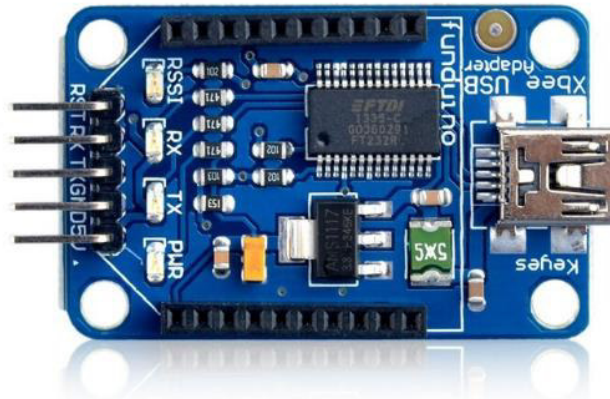


Fonte: <https://www.filipeflop.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>

sensores e seus respectivos ESP01), *protoboards*, *jumpers* e reguladores de tensão de 5V, este último o LM2596 mostrado na Figura 29. Os reguladores são necessários uma vez que o ESP01 não possui esse tipo de circuito embutido.

Para a gravação das rotinas no ESP01, o XBee *explorer* USB *adapter* (Figura 30) foi utilizado, funcionando de forma similar ao citado na Figura 21.

Figura 30 – XBee explorer USB adapter

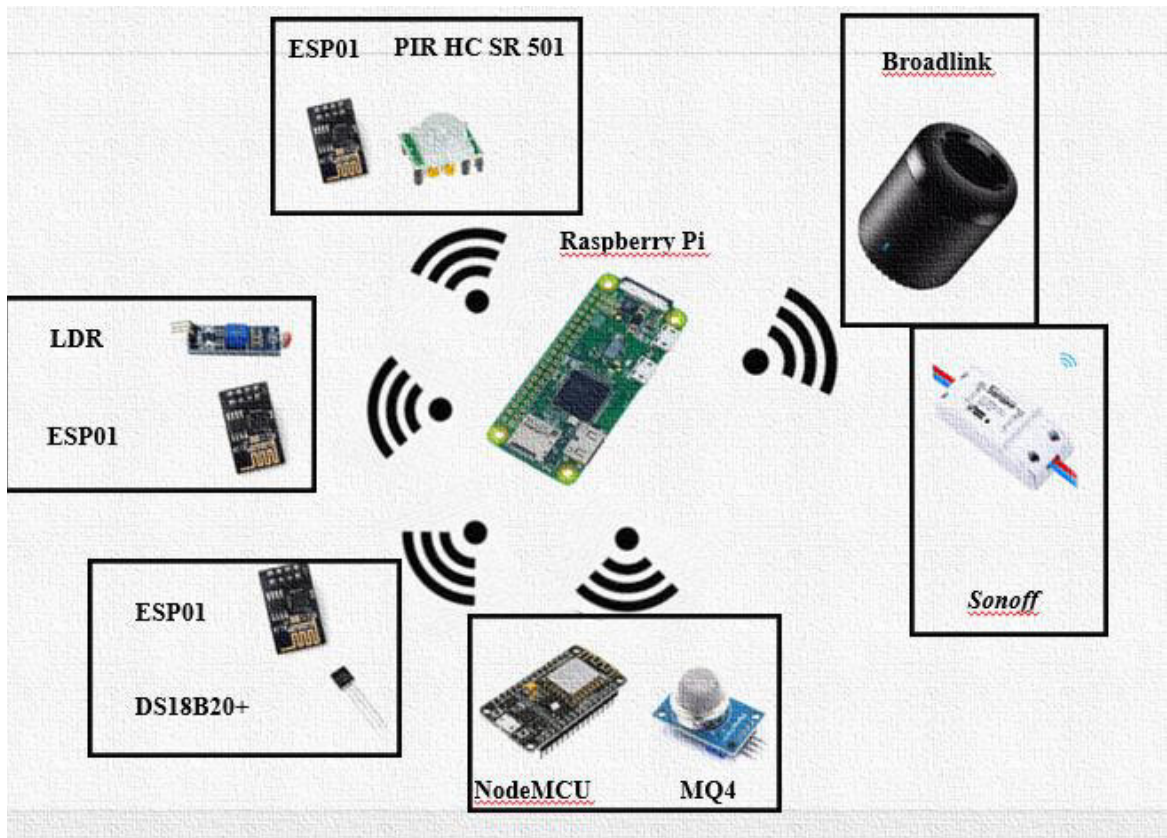


Fonte: <https://www.filipeflop.com/produto/xbee-explorer-usb-adapter/>

3.5 – Sistema Proposto Completo

A ideia primordial do sistema de automação é utilizar dados fornecidos pelos sensores de forma a automatizar dispositivos presentes na residência. Para isso é necessário posicionar os sensores em conjunto com os seus respectivos microcontroladores: a) ESP01 para o módulo LDRC, PIR HC-SR501, DS18B20+; e b) MQ4 conectado ao NodeMCU. Dessa forma, os estímulos do ambiente serão captados e transmitidos para a central, no caso o Raspberry Pi, por meio de comunicação *wireless*. A central que está integrada ao *Home Assistant*, uma vez em posse de dados, está apta a tomar decisões que automatizam o ambiente como o acionamento de dispositivos como o *Sonoff* e o *Broadlink Rm Mini 3* que conseqüentemente controlam aparelhos como lâmpadas, ar-condicionado, televisão e sirene. Uma ilustração do funcionamento do sistema pode ser visto na Figura 31. Além disso, o sistema também funciona com uma interface própria para dispositivos móveis a qual é descrita no Capítulo 4.

Figura 31 – Esquema do sistema de automação completo



Fonte: Autor

4. IMPLEMENTAÇÃO DA ESTAÇÃO SENSORIAL DE BAIXO CUSTO COM CAPACIDADE DE CONTROLE

Nesse capítulo descreve-se a integração do sistema e as automações implementadas com os *hardware* e *software* citados no Capítulo 3. Os sensores foram espalhados numa residência, em São Luís, em locais convenientes para detecção de dados de forma a realizar as automações reativas pretendidas. No que se refere às automações programadas e perfis customizados, como fazem uso dos recursos comentados neste capítulo, só será apresentada como experimento no Capítulo 5. A parte inicial do projeto é similar à apresentada por Almeida (2018), uma vez que é necessário executar os mesmos passos, como a instalação do *Hass.io* e a configuração do sistema.

4.1. Instalação, conexão e ajustes de configuração com a central de automação.

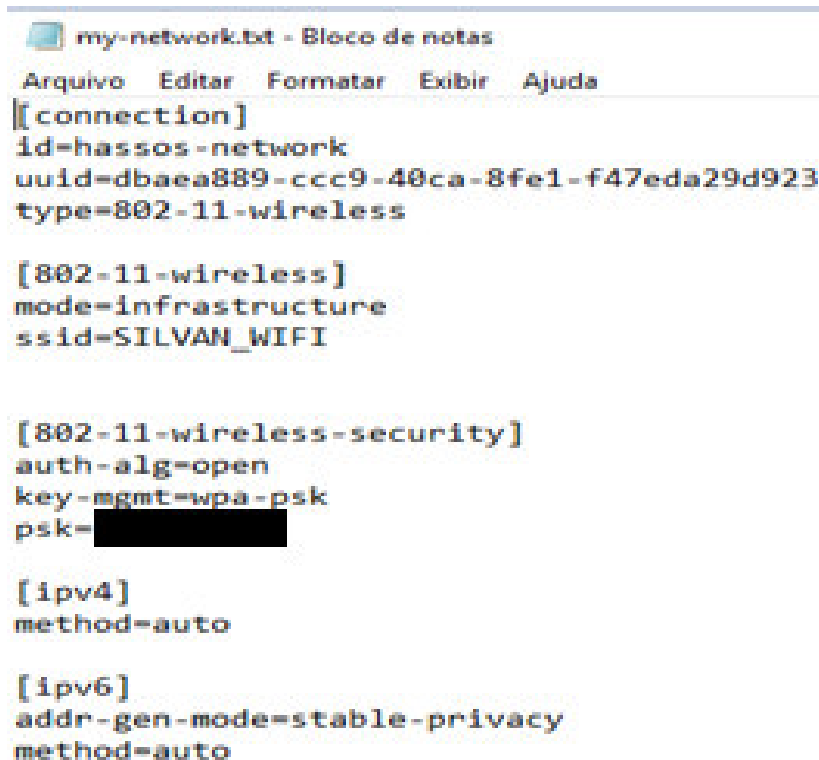
O Raspberry Pi 3B (RPi 3B) e o *Hass.io* foram os *hardware* e *software*, respectivamente, utilizados para a implementação da central de automação, devido ao baixo custo da placa e a gratuidade e versatilidade do *software*. É importante ressaltar o que foi dito no Capítulo 3 sobre o uso do Raspberry Pi 3B em substituição a versão Zero W, por motivos da disponibilidade, já que a versão Zero W, de aproximadamente 35% do custo da versão 3B, tem recurso computacional para operar como o sistema da central.

O primeiro passo a ser tomado foi a instalação do *Hass.io*. A imagem do sistema foi baixada diretamente do site oficial e foi carregada em um cartão de memória, o qual os desenvolvedores aconselham ser de pelo menos 32 GB. Entretanto, o cartão de 16 GB foi suficiente para os propósitos da aplicação, tendo sido o de tamanho utilizado neste projeto.

Após a instalação do sistema foi necessário garantir que o mesmo estivesse apto a se conectar com a internet e realizar as atualizações necessárias, existindo algumas possibilidades para isso. No projeto optou-se pela utilização de um *pen drive* conectado à placa RPi 3B, que opera como uma pasta de configuração do sistema, sendo necessário seguir e executar alguns passos. Primeiramente, formatou-se e renomeou o *pen drive* para “**CONFIG**”, e em sua raiz foi criada uma pasta com o nome “**network**” onde se salvou o arquivo de configuração de rede Wi-Fi “**my-network.txt**”, como código mostrado na Figura 32. Nesse arquivo foi inserido o nome da rede a ser conectada no tópico “**ssid**” e a senha no tópico “**psk**”.

Uma vez que o *pen drive* esteja configurado pode ser conectado ao Raspberry, o sistema pode ser inicializado. O próximo passo foi esperar, pois a última atualização do sistema foi

Figura 32 – Arquivo de configuração da rede Wi-Fi



```

my-network.txt - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
[[connection]
id=hassos-network
uuid=dbaea889-ccc9-40ca-8fe1-f47eda29d923
type=802-11-wireless

[802-11-wireless]
mode=infrastructure
ssid=SILVAN_WIFI

[802-11-wireless-security]
auth-alg=open
key-mgmt=wpa-psk
psk=██████████

[ipv4]
method=auto

[ipv6]
addr-gen-mode=stable-privacy
method=auto

```

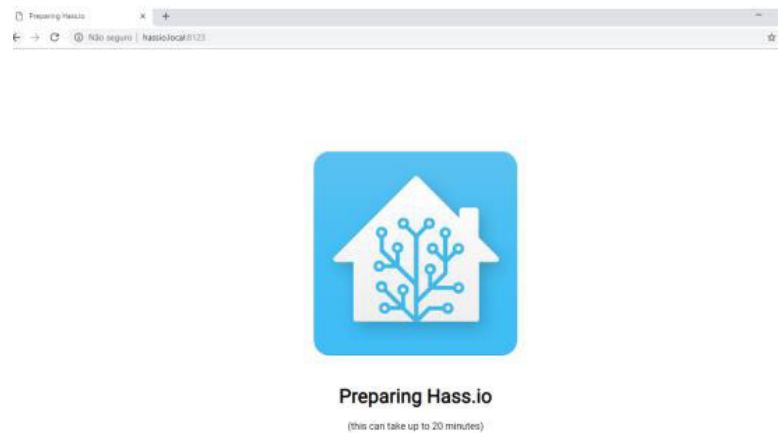
Fonte: Autor

automaticamente instalada e normalmente leva cerca de 20 minutos. O processo de instalação pode ser acessado pela rede local por meio do link <https://hassio.local:8123>, assim como ilustrado na Figura 33.

Logo após a atualização foi necessário finalizar algumas configurações, como estabelecer um usuário e senha para serem utilizadas durante o *login*, como apresentado na da Figura 34.

Além disso, a utilização de *add-on*, que são *software* suportados pelo sistema, podem fornecer uma experiência mais agradável na utilização do *Hass.io*. Apesar de muitos *add-on* serem adaptáveis ao sistema, nesse trabalho somente os três citados a seguir foram instalados:

- a) *SSH Server*: Permite o acesso ao sistema por meio do protocolo SSH.
- b) *Configurator*: Permite abrir arquivos do sistema *Hass.io* por meio de editores de texto, facilitando a programação.
- c) *Mosquitto Broker*: Um gerenciador que permite ao sistema trocar mensagens por meio do protocolo MQTT.

Figura 33 – Acesso e atualização do sistema *Hass.io*

Fonte: Autor

Figura 34 – Login no sistema do *Home Assistant*The Home Assistant login page features the Home Assistant logo (a blue house with a white circuit pattern) and the text 'Home Assistant'. Below this, a message asks: 'Você está pronto para despertar sua casa, recuperar sua privacidade e se juntar a uma comunidade mundial de consertadores?' followed by 'Vamos começar criando uma conta de usuário.' The form includes three input fields: 'Nome' (with 'Obrigatório' below it), 'Usuário', and 'Senha'. A 'CRIAR CONTA' button is positioned at the bottom center.

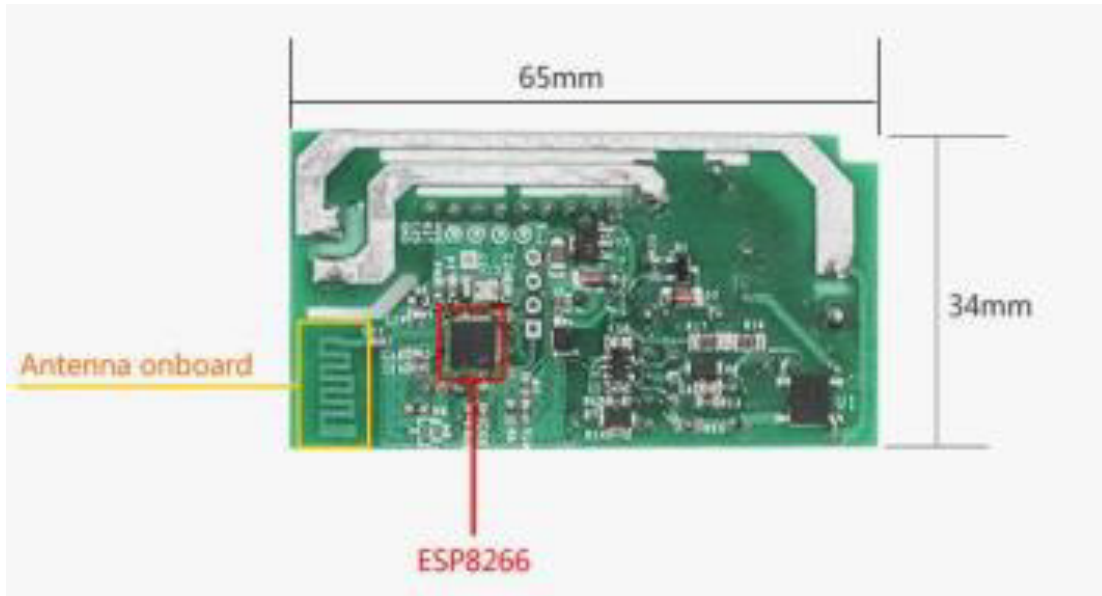
Fonte: Autor

4.2. Programação e integração do atuador *Sonoff*

O primeiro dispositivo estabelecido foi o *Sonoff*, pois já haviam sido programados no projeto de Almeida (2018), não havendo a necessidade de refazer tal procedimento. No entanto, o processo

de integração do dispositivo é descrito nos tópicos a seguir e para isso foi necessário acessar o *chip* ESP8266 presente na placa, conforme mostrado na Figura 35.

Figura 35 - ESP8266 acoplado à placa do dispositivo *Sonoff*



Fonte: <http://wannabits.com/2016/12/26/conectar-configurar-conmutador-inteligente-wifi-sonoff>

4.2.1. Programação do ESP8266 (ESP01) presente na placa do *Sonoff*

O *Xbee explorer USB adapter* foi utilizado para realizar a transferência do código em linguagem C/C++ para o ESP01. O código em questão visa estabelecer uma comunicação com o *MQTT broker* integrado à central de automação, conforme mostrado na Figura 36.

Após adicionar essas linhas, foi necessário adicionar outras com a finalidade de configuração do dispositivo relativo à central de automação por meio do programa apresentado na Figura 37. O acesso ao arquivo foi feito por meio do *add-on Configurator*. As linhas adicionadas no código da Figura 37 permitem a comunicação, via protocolo MQTT, além da inserção de comandos de acionamento na interface do *Home Assistant* (ALMEIDA, 2018).

O *Sonoff* em questão foi adicionado para o controle da iluminação da sala de estar e pode ser acionado por meio dos botões virtuais inseridos na interface, como mostrado na Figura 38, a qual também mostra o estado de acionamento desse interruptor.

Figura 36 – Códigos na Arduíno IDE para a programação do *Sonoff*

```

sonoff - my_user_config.h | Arduino 1.8.7
jivo Editar Sketch Ferramentas Ajuda

sonoff  _shangela09  core_esp8266_btmac  core_esp8266_wiring_digital  core_esp8266_wiring_pwm  i18n.h  my_user_config.h  settings

#define MQTT_BUTTON_RETAIN    0           // [ButtonRetain] Button may send retain flag (0 = off, 1 = on)
#define MQTT_POWER_RETAIN     0           // [PowerRetain] Power status message may send retain flag (0 = off, 1 = on)
#define MQTT_SWITCH_RETAIN    0           // [SwitchRetain] Switch may send retain flag (0 = off, 1 = on)

#define MQTT_STATUS_OFF       "OFF"       // [StateText1] Command or Status result when turned off (needs to be a string like
#define MQTT_STATUS_ON        "ON"        // [StateText2] Command or Status result when turned on (needs to be a string like
#define MQTT_CMD_TOGGLE       "TOGGLE"    // [StateText3] Command to send when toggling (needs to be a string like "1" or "T
#define MQTT_CMD_HOLD         "HOLD"      // [StateText4] Command to send when button is kept down for over KEY_HOLD_TIME

-- MQTT topics -----
/* Example "tasmota/bedroom/#topic/#prefix/" up to 80 characters
#define MQTT_FULLTOPIC        "#prefix/#topic/" // [FullTopic] Subscribe and Publish full topic name - Legacy topic

#prefix# token options
#define SUB_PREFIX            "cmd"        // [Prefix1] Sonoff devices subscribe to #prefix/#topic# being SUB_PREFIX/MQTT_TO
#define PUB_PREFIX            "stat"       // [Prefix2] Sonoff devices publish to #prefix/#topic# being PUB_PREFIX/MQTT_TOPI
#define PUB_PREFIX2           "tele"      // [Prefix3] Sonoff devices publish telemetry data to #prefix/#topic# being PUB_P
// May be named the same as PUB_PREFIX

#topic# token options (also ButtonTopic and SwitchTopic)
#define MQTT_TOPIC            PROJECT      // [Topic] (unique) MQTT device topic, set to 'PROJECT "_"$04X"' for unique topic i
#define MQTT_GRPTOPIC         "sonoffs"   // [GroupTopic] MQTT Group topic
#define MQTT_BUTTON_TOPIC     "0"         // [ButtonTopic] MQTT button topic, "0" = same as MQTT_TOPIC, set to 'PROJECT "_BT
#define MQTT_SWITCH_TOPIC     "0"         // [SwitchTopic] MQTT button topic, "0" = same as MQTT_TOPIC, set to 'PROJECT "_SW
#define MQTT_CLIENT_ID        "DVS_$06X" // [MqttClient] Also fall back topic using Chip Id = last 6 characters of MAC addr

-- MQTT - Telemetry -----
#define TELE_PERIOD           300          // [TelePeriod] Telemetry (0 = disable, 10 = 3600 seconds)

```

Fonte: ALMEIDA, 2018

4.3. Integração do *Broadlink* RM Mini 3 ao *Hass.io*

O *Broadlink* é outro dispositivo que não foi necessário ser reprogramado, pois o processo também foi realizado por Almeida (2018) e aproveitado nesse projeto, sendo os passos usados descritos a seguir.

Ao ligar o *Broadlink* RM Mini 3 o primeiro passo foi conectar o dispositivo à rede Wi-Fi. Quando ligado pela primeira vez, o dispositivo cria uma rede própria que serve para definir o nome e a senha ao qual o dispositivo deve se conectar na conexão Wi-Fi.

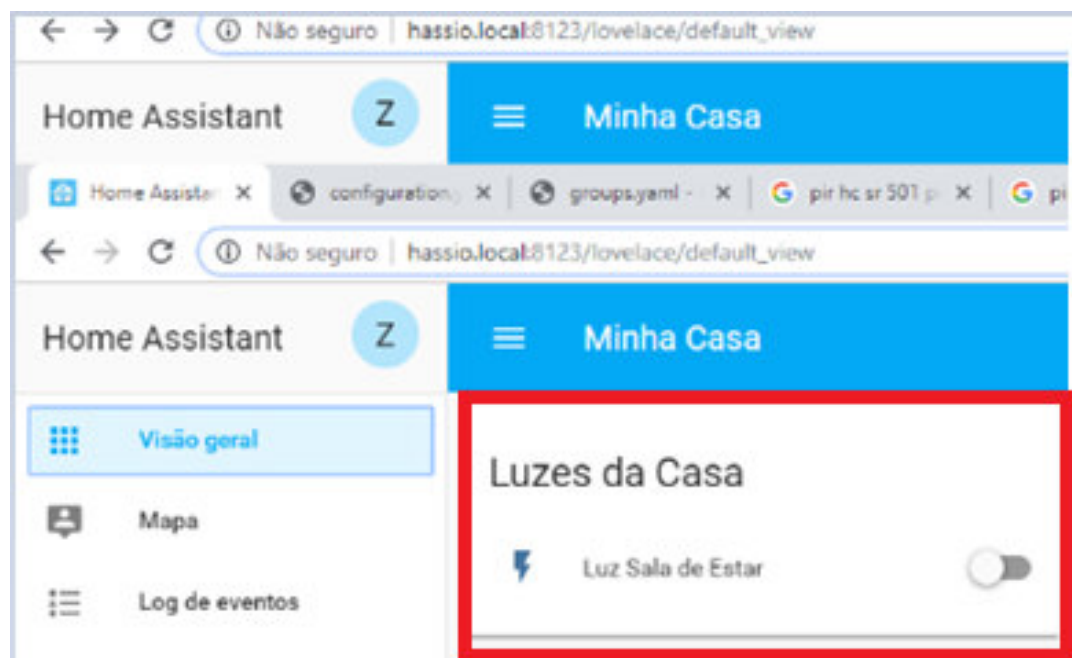
Figura 37 – Código do arquivo de configuração referente ao *Sonoff*

```
#MQTT~
mqtt:~
  ...broker: 192.168.0.7~
  ...username: zarley~
  ...password: [REDACTED]~
~
switch:~
  ...platform: mqtt~
  ...name: "Sonoff Switch-01"~
  ...command_topic: "cmd/sonoff01/power"~
  ...state_topic: "stat/sonoff01/POWER"~
  ...qos: 1~
  ...payload_on: "ON"~
  ...payload_off: "OFF"~
  ...retain: true~
```

Fonte: (ALMEIDA, 2018)

Além disso, para integrar o *Broadlink* com o *Hass.io* foi necessário descobrir o endereço de IP e o MAC (*Media Access Control*) do dispositivo, o que foi feito por meio do programa *Network Scanner*, como pode ser visto na Figura 39 (ALMEIDA, 2018).

Figura 38 – Controle e estado da lâmpada na interface do *Home Assistant*



Fonte: Autor

Figura 39 – Endereços de IP e MAC encontrados através do aplicativo *Network Scanner*

IP Address	MAC Address	Response Time	Host ...
192.168.0.1	74-DA-DA-EC-51...	6 ms	
192.168.0.2	34-EA-34-40-8E-62	74 ms	
192.168.0.4	2C-3A-E8-0F-30...	34 ms	
192.168.0.7	B8-27-EB-7E-31-0F	7 ms	HASSIO
192.168.0.8	F4-B7-E2-D5-79-...	0 ms	DESK...
192.168.0.12	18-FE-34-F2-5F-09	71 ms	
192.168.0.3	FC-A6-67-44-97...	120 ms	
192.168.0.5	38-2D-D1-97-8D...	3120 ms	
192.168.0.9	CC-61-E5-EA-07...	3121 ms	

Fonte: (ALMEIDA, 2018)

Essas informações foram adicionadas à central por meio do *add-on Configurator* como mostrado no código da Figura 40.

Figura 40 - Código para integração do *Broadlink* à Central de Automação

```

platform
  switch:
    - platform: mqtt
      name: "Sonoff Switch 01"
      command_topic: "cmd/sonoff01/power"
      state_topic: "stat/sonoff01/POWER"
      qos: 1
      payload_on: "ON"
      payload_off: "OFF"
      retain: true
    - platform: broadlink
      # type: rm_mini
      timeout: 15
      host: 192.168.0.2
      mac: "34:EA:34:40:8E:62"
  switch.ac_16)
    ac_power:
      friendly_name: "AC Power"
      command_on: '3gCSAAAB3pUU0BMVEXQTORM4FDgTFRMUFBQTFBQUExUTFE
      command_off: '3gCSAAABK3QVnHUTFRiEHU3FRIVExUSFjYVnXU2FBQVE
    ac 16:
  
```

Fonte: (ALMEIDA, 2018)

Em seguida foram gravados no dispositivo os códigos dos aparelhos a serem controlados. O *Broadlink* recebe o código enviado pelo controle, o qual aparece na interface do *Home Assistant* convertido para o formato *Base64* (usado para transmitir dados binários em formato de texto). Esse código é inserido no *Configurator* para a inserção gráfica do aparelho na interface do *Home Assistant*, conforme mostrado na Figura 41 (ALMEIDA, 2018).

Figura 41 – Código do controle recebido pelo *Home Assistant* e inserido no arquivo de configuração



Fonte: ALMEIDA, 2018

Um dos exemplos de aparelhos adicionados ao sistema é o ar-condicionado, cujo controle pode ser feito por meio da interface do *Home Assistant*, conforme mostrado na Figura 42. Há dois controles na imagem, o de ligar-desligar e o que regula a sua temperatura programada.

4.4. Programação dos Sensores da Central

A programação dos sensores foi um ponto de grande diferença em relação ao apresentado por Almeida (2018), pois automações foram adicionadas em função dos dados recebidos por cada um dos sensores. Na Figura 43 é possível ver todos os sensores já integrados à interface do *Home Assistant*.

Figura 42– Interface de Controle do ar-condicionado no *Home Assistant*



Fonte: Autor

Figura 43 – Sensores integrados à interface do *Home Assistant*



Fonte: Autor

4.4.1. Integração do DS18B20+ ao sistema da central de automação

Primeiramente, foi necessário adicionar linhas de código para a adição desse sensor à central de automação, de forma similar ao ocorrido com o *Sonoff*, conforme pode ser observado na Figura 44. Novamente, esse procedimento é feito por meio do *Configurator*.

A seguir foi necessário fazer com que o DS18B20+ se comunique com a central. Isso foi feito por meio do ESP 01 e a plataforma IDE do Arduíno com o auxílio do *Xbee explorer USB adapter*, como pode ser visto na Figura 45. Pode-se perceber no programa, no procedimento “**loop**” que o valor de temperatura lido pelo sensor (“**temperature**”) é publicado com a função

Figura 44 – Linhas de código para adição do DSB18B20+ ao *Home Assistant*

```

# Sensors
sensor:
  # Weather prediction
  - platform: yr
  ...
  - platform: mqtt
  - state_topic: "/house/temp1"
  - name: "SensorTemperatura"
  ...
  - platform: mqtt
  - state_topic: "/house/gas1"
  - name: "SensorGas"
  ...

```

Fonte: Autor

“**cliente.publish(tempTopic, temperature)**” no tópico “**tempTopic**” do MQTT broker. Passos necessários para estabelecer o processo de comunicação via MQTT.

Figura 45 – Linhas de código na Arduíno IDE para aquisição de dados e transferência via MQTT do sensor de temperatura

```

MQTT_TEMP-SENSOR

void loop() {

  //get the new value
  float currentTempFloat = binarySensor(D0);

  //convert the temp float to a string and publish to the temp topic
  char temperature[10];
  dtostrf(currentTempFloat,4,1,temperature);
  client.publish(tempTopic, temperature);
  Serial.println(temperature);

  //reconnect if connection is lost
  if (!client.connected() && WiFi.status() == 3) {reconnect();}
  //maintain MQTT connection
  client.loop();
  //MUST delay to allow ESP8266 WIFI functions to run
  delay(5000);
}

```

Fonte: Autor

O primeiro passo que faz com que o projeto se diferencie significativamente do apresentado por Almeida (2018) foi implementado a seguir. Enquanto o projeto citado se preocupou com a integração de dispositivos e a utilização da Alexa como interface de controle, esse visa a utilização de automações reativas pré-programadas para manter a comodidade do residente, neste caso com relação à temperatura do ambiente. Esse processo foi necessário uma vez que a versatilidade da Alexa foi retirada.

Usando o *Configurator* foi criado um *script*, que define ações a serem tomadas, e pode ser acionado na aba de automações, a qual define as condições necessárias para que esse *script* seja executado. O código implementado (Figura 46) funciona como uma rotina de economia de energia e por meio das medidas adquiridas pela central, o ar-condicionado é desligado assim que chegue a 16°C e volta a ser ligado quando alcança 23°C.

Figura 46 – Linhas do código de automação para economia de energia

```

...service: switch.turn_on-
- alias: AirEnergySaver1-
- hide_entity: false-
- trigger:-
...platform: template-
...value_template: '{{states.sensor.sensortemperatura.state > 34}}'-
- action:-
...service: script.airon-
- id: 18c315baf32844a1901ea6d4e00c7138-
- alias: AirEnergySaver2-
- hide_entity: false-
- trigger:-
...platform: template-
...value_template: '{{states.sensor.sensortemperatura.state < 33}}'-
- action:-
...service: script.airoff-
- id: 3c6d015a18b644b48dc6c29dfd39b6f5-
- alias: AlarmOn-

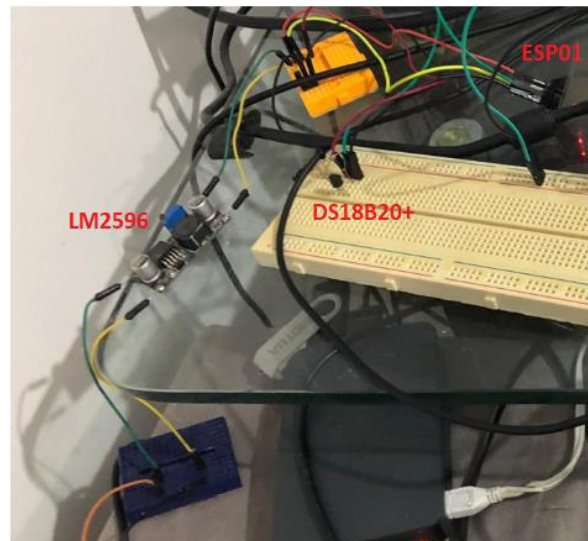
```

Fonte: Autor

A automação apresentada na seção “**alias: ‘AirEnergySaver1’**”, da Figura 46, representa as condições necessárias para que o ar-condicionado seja ligado, ao passo que a seção “**alias: ‘AirEnergySaver2’**” representa as condições necessárias para que o mesmo seja desligado.

Na Figura 47 é possível observar as conexões do protótipo testado com o sensor DS18B20+ e ESP01 em funcionamento.

Figura 47 – ESP01 e DS18B20+ instalados na residência



Fonte: Autor

4.4.2. Integração do PIR HC - SR501 ao sistema da central de automação

Os passos a serem seguidos para todos os sensores são similares, primeiro foi necessário adicionar o sensor à central de automação por meio do *Configurator*, assim como mostrado na Figura 48 para o PIR HC – SR501.

O código apresentado na Figura 49 é referente a aquisição de dados por esse sensor e ao estabelecimento da comunicação MQTT processados pelo ESP01. Através do nível lógico da

Figura 48 - Linhas de código para adição do PIR HC – SR501 ao *Home Assistant*

```

..... 0
..... {% - endif - %}
..... {{ states.sensor.alarm clock minute.state }}
-- platform: mqtt
-- name: "Esp8266 PIR 01"
-- state_topic: "/house/PIRsensor1"
-- platform: mqtt
-- name: "Modulo LDR"
-- state_topic: "/house/LDR"

```

Fonte: Autor

Figura 49 - Linhas de código na Arduino IDE para aquisição de dados e transferência via MQTT do sensor de movimento

```

float valor_digital2 = digitalRead(D0);
char digital2[10];
dtostrf(valor_digital2, 4, 1, digital2);
client.publish(digitalTopic2, digital2);
Serial.println(digital2);

//reconnect if connection is lost
if (!client.connected() && WiFi.status() == 3) {reconnect();}
//maintain MQTT connection
client.loop();
//MUST delay to allow ESP8266 WIFI functions to run
  
```

Leitura

Transferência

Fonte: Autor

variável “**digital12**” e do comando de publicação “**cliente.publish(digitalTopic2, digital12)**” é possível informar a presença ou a ausência de pessoa no ambiente monitorado.

O *script* de automação foi criado nesse caso para funcionar como um alarme de segurança. A automação fica visível na interface de controle do *Home Assistant*, como pode ser observado na Figura 50 e, uma vez que ela esteja ativada, o sinal recebido pelo sensor acionará uma sirene que no projeto foi simulado por um *buzzer* (buzina).

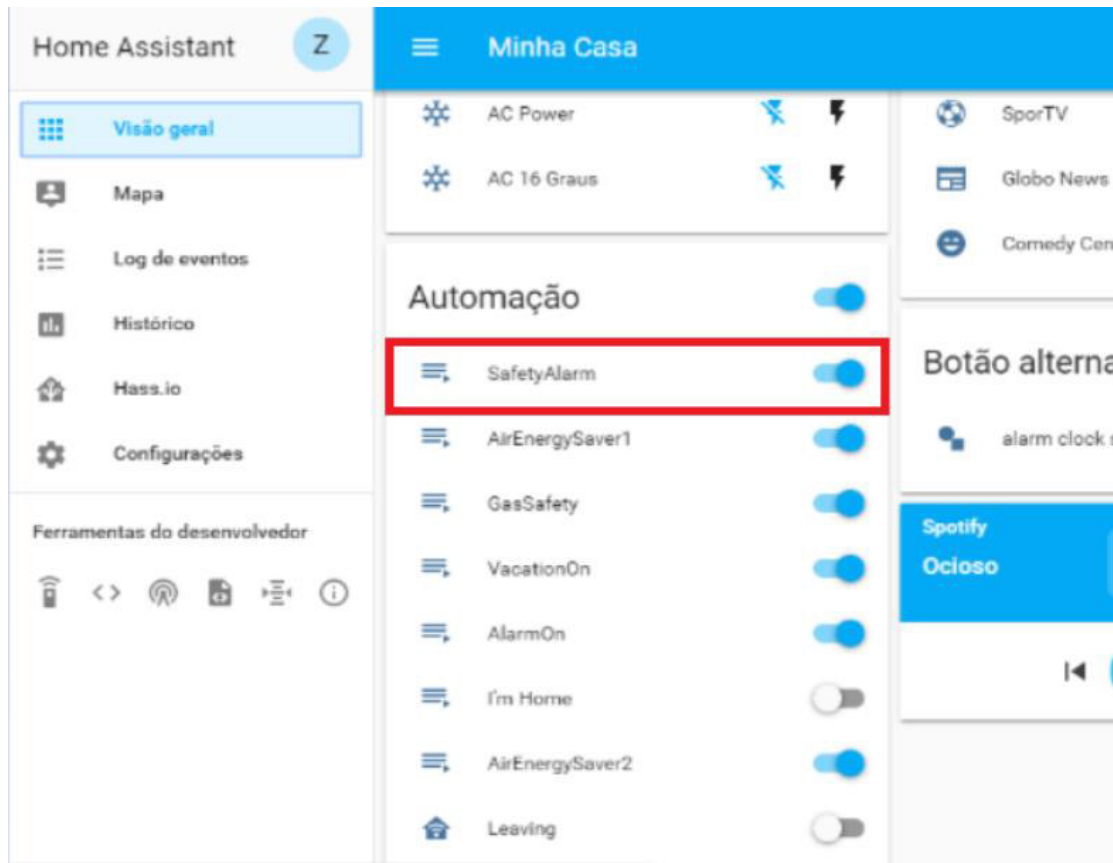
O código da automação pode ser visto na Figura 51, representada pela seção “**alias: ‘SafetyAlarm’**”, onde a região destacada em azul refere-se às condições necessárias para a chamada do *script* e a vermelha corresponde a ação em si com a ativação da buzina.

4.4.3. Integração do módulo LDR ao sistema da central de automação

O código referente à adição do sensor à central de automação por meio do *Configurator* pode ser visto na Figura 52, no retângulo destacado em vermelho.

Já o código apresentado na Figura 53 é referente à aquisição de dados pelo LDR e ao estabelecimento da comunicação MQTT processados no ESP01. O LDR pode funcionar como um sensor digital, e por isso o ESP01 é utilizado, uma vez que só duas condições são necessárias neste projeto (escuro ou não escuro).

Figura 50 – Automação do sensor de movimento ativada na interface de controle



Fonte: Autor

Figura 51 – Linhas do código de automação para o alarme de segurança

```

actions:
  - service: script.buzzeron
    alias: 'SafetyAlarm'
    hide_entity: False
    trigger:
      platform: template
      value_template: '{{states.sensor.esp8266_pir_01.state == 0}}
    action:
      - service: script.buzzeron
        alias: 'VacationOn'
        hide_entity: False
  - data:
    - entity_id: switch.ac_power
    buzzerOn:
      - sequence:
        - service: homeassistant.turn_on
          data:
            entity_id: switch.buzzer
      lightOn:
        - sequence:

```

Fonte: Autor

O *script* de automação desse sensor foi criado para funcionar como uma rotina de segurança para longos períodos de ausência da residência. Entretanto, ele ainda não se trata de automação para execução dos perfis customizados, que fazem uso de atuadores descritos aqui e que terá um experimento no próximo capítulo. Assim, em momentos como as férias, por exemplo, se a automação estiver ativada, a ausência de iluminação resultará no estado escuro e fará com que alguns dispositivos possam ser ligados, em tempos previamente programados em perfis customizados, como TV e iluminação da sala. O intuito é passar a impressão de que há pessoas na residência, minimizando a motivação de pessoas mal-intencionadas de invadir o ambiente. O código dessa automação reativa pode ser visto em destaque na Figura 54 na seção “**alias: ‘VacationOn’**”.

Figura 54 – Linhas do código de automação para rotina de segurança em períodos de ausência.

```

...platform: template~
...value_template: '{{states.binary_sensor.esp8266_pir_01.state == 0}}'~
...action: ~
...service: script.buzzeron~
alias: 'VacationOn'~
hide_entity: false~
trigger: ~
...platform: template~
...value_template: '{{states.sensor.moduloldr.state == 0}}'~
...action: ~
...service: script.lighton~

```

Fonte: Autor

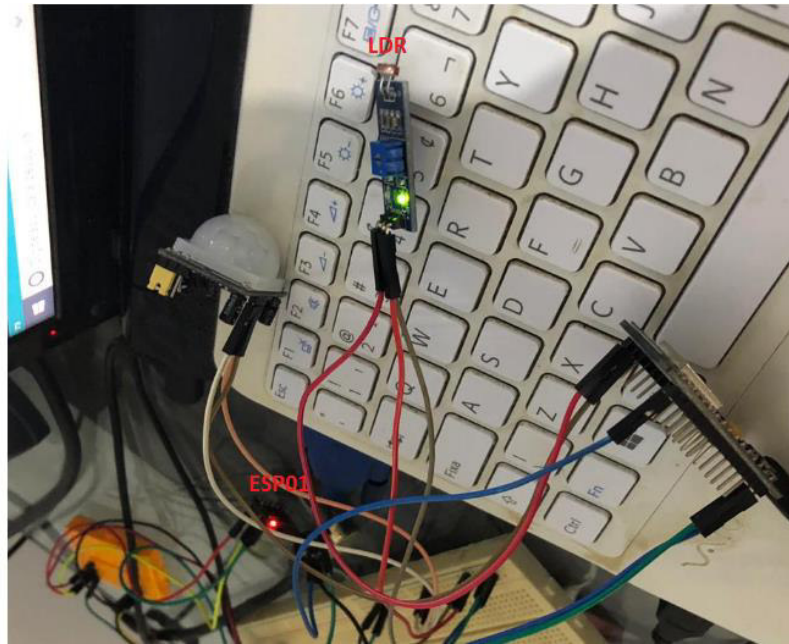
O módulo LDR funcionando juntamente com o ESP01 pode ser visto nas conexões mostradas Figura 55.

4.4.4. Integração do sensor de metano MQ4 ao sistema da central de automação

O MQ4 precisa ser lido por meio de um pino analógico, portanto, foi necessário utilizar o NodeMCU para essa finalidade. Uma vez que o NodeMCU pode fazer a troca de informações diretamente com o computador não foi necessário a utilização do Xbee *explorer* USB *adapter* para fazer a carga do programa da IDE do Arduino. Esse código é apresentado na Figura 56, sendo referente à aquisição de dados pelo MQ4 (comando “**float valor_analogico = analogRead(A0)**”)

e ao envio de dados via comunicação MQTT processados no NodeMCU (“**cliente.publish(analogTopic, analog)**”).

Figura 55 – ESP01 e Módulo LDR em funcionamento



Fonte: Autor

Figura 56 – Linhas de código na IDE do Arduino para a aquisição de dados e transferência via MQTT do sensor de gás

```

testeLDR_Gas | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda

testeLDR_Gas

void loop() {

  // Send the command to update temperatures
  //sensors.requestTemperatures();

  //get the new temperature
  //float currentTempFloat = sensors.getTempCByIndex(0);
  float valor_analogico = analogRead(A0);
  valor_analogico = map(valor_analogico, 0, 1024, 0, 100);
  //conver the temp float to a string and publish to the temp topic
  char analog[10];

  dtostrf(valor_analogico,4,1,analog);
  cliente.publish(analogTopic, analog);
  Serial.println(analog);
}

```

Fonte: Autor

A seção de código referente à adição do sensor à central de automação por meio do *Configurator*, destacada no retângulo preto, pode ser visto na Figura 57.

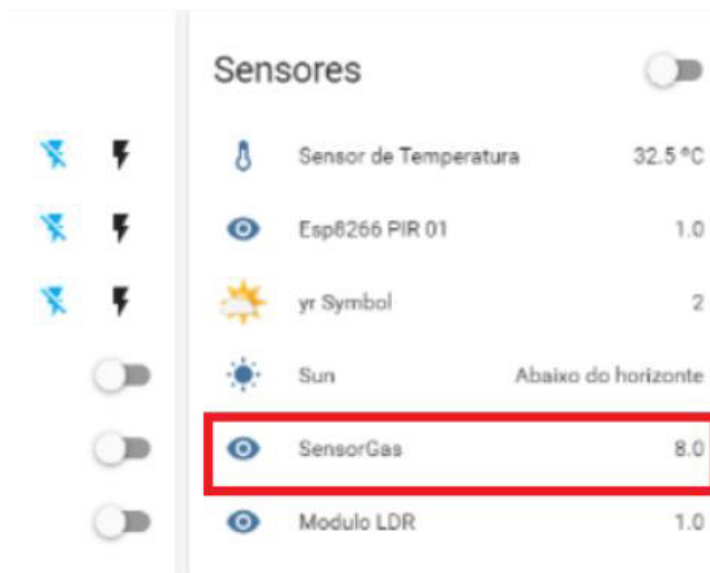
Figura 57 – Linhas de código para adição do MQ4 ao *Home Assistant*

```
# Sensors
sensor:
  # Weather prediction
  -- platform: yr
  --
  -- platform: mqtt
  -- state_topic: "/house/temp1"
  -- name: "SensorTemperatura"
  --
  -- platform: mqtt
  -- state_topic: "/house/gas1"
  -- name: "SensorGas"
```

Fonte: Autor

O estado do sensor na interface do *Home Assistant* pode ser visto na Figura 58, destacado no retângulo, onde é apresentado o valor 8.0 que indica uma leitura percentual do sensor. Inicialmente, a leitura deste era feita entre 0 e 1024 referente aos seus 10 bits de resolução, no entanto, por meio da função “**map (valor_analógico, 0, 1024, 0, 100)**”, apresentada na Figura 56, esses valores foram mapeados entre 0 e 100.

Figura 58 – Estado do Sensor na interface do *Home Assistant*



Fonte: Autor

O *script* de automação desse sensor visa funcionar como um mecanismo de segurança, alertando o residente sobre níveis irregulares de gás liquefeito de petróleo (gás de cozinha). Quando um nível irregular programado é atingido, a automação aciona o *script* que avisa o morador por meio de uma sirene (simulado pelo *buzzer*). O código da automação (seção “**alias: ‘GasSafety’**”)

Figura 59 – Linhas do código de automação para rotina de níveis irregulares de gás de cozinha.

```

... service: switch.turn_off~
... #Victor~
- alias: 'GasSafety'~
- hide_entity: False~
- trigger:~
  - platform: template~
  - value_template: '{{states.sensor.gassensor.state > 40}}'~
- action:~
  - service: script.buzzeron~
- alias: SafetyAlarm~
- hide_entity: False~
- trigger:~

```

Fonte: Autor

pode ser visto na Figura 59.

Na Figura 60 é possível visualizar o NodeMCU conectado ao sensor MQ4 e em funcionamento.

Figura 60 – NodeMCU e MQ4 em funcionamento

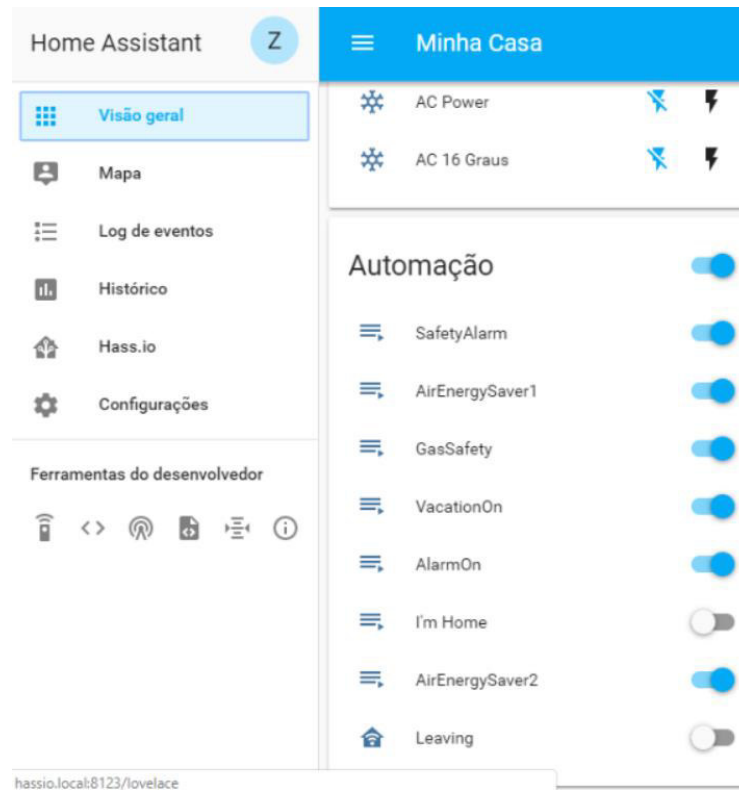


Fonte: Autor

4.5. Implementação adicional de automações

Além das automações diretamente relacionadas aos sensores, mais algumas foram implementadas para complementar o sistema. A interface gráfica que ativa todas as automações residências deste projeto pode ser vista na Figura 61.

Figura 61 – Interface no *Home Assistant* para ativação das automações.



Fonte: Autor

O processo de criação das automações adicionais é similar à dos sensores, um *script* descrevendo o funcionamento da mesma é adicionado com o uso do *Configurator* e as condições de acionamento são definidas na aba de automações. Como exemplo, uma dessas automações visa funcionar com um alarme, desligando aparelhos como o ar-condicionado e ligando luzes. O código da mesma pode ser visto na Figura 62.

Convém lembrar que os perfis que podem ser criados pelo usuário do sistema visam simular a presença de pessoas na residência, quando na verdade a mesma encontra-se vazia. Dessa forma, a definição dos horários para que os aparelhos sejam desligados e ligados é definido pelos *sliders* apresentados na aba “**Entrada Numérica**” da Figura 63.

Figura 62 – Código da automação com a função de definir um alarme

```

id: 3c6d015a18b644b48dc6c29dfd39b6f5
alias: AlarmOn
initial_state: true
hide_entity: false
trigger:
- platform: template
- value_template: '{{ states.sensor.time.state == states.sensor.alarm_clock_time_long.state }}'
condition:
- condition: state
- entity_id: input_boolean.alarm_clock_status
- state: 'on'
action:
- data:
- entity_id: switch.sonoff_switch_01
- service: switch.turn_off

```

Fonte: Autor

Figura 63 – Interface da entrada numérica e *sliders*

Fonte: Autor

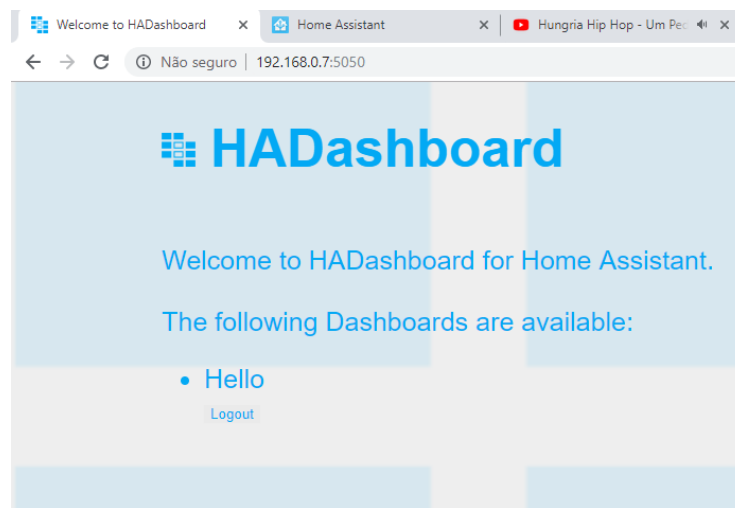
O primeiro *slider* “**alarm clock hour**” varia de 0 a 24 e define a hora que os aparelhos devem ser acionados ou desligados, o segundo *slider* varia de 0 a 60 e define os minutos.

Uma função adicional dessa automação é complementar a automação do LDR, pois de forma bem simples visa simplesmente desligar os aparelhos da residência em um horário pré-definido, escolhido no projeto como meia noite. Dessa forma, a sensação de que há alguém na residência ligando e desligando aparelhos fica completa. O código de tal automação pode ser visto na Figura 62, a única modificação necessária foi que nesse caso, esta somente desligue aparelhos.

4.6. Implementação da interface para dispositivos móveis

A criação de uma interface amigável para dispositivos móveis foi feita mediante a adição de um *add-on* ao *Home Assistant*, o *AppDaemon*, o qual permite organizar de forma bem espaçada os dispositivos previamente integrados ao *Hass.io*, correspondendo de forma eficiente às telas de nossos celulares ou *tablets*. O acesso pode ser feito por meio do link “<http://hassio.local:5050>” na rede local, conforme mostrado na Figura 64.

Figura 64 – Acesso a interface para dispositivos móveis na rede local



Fonte: Autor

A configuração dessa interface e o *layout* dos dispositivos pode ser feita por meio da aba “</config/appdaemon/appdaemon.yaml>” no *Configurator*, sendo o código escrito conforme o mostrado na Figura 65. É possível definir títulos e ícones para diferentes entidades, como automações ou atuadores, e em seguida organizar a disposição dessas entidades na aba *layout*.

No *AppDaemon* é possível a utilização de *designs* previamente padronizados para a interface. Um dos *designs* que se mostrou bem amigável e foi escolhido para o projeto é o *obsidian*, que pode ser visto na Figura 66 e pode ser utilizado por meio da adição de alguns comandos ao link “<http://hassio.local:5050/Hello? skin=obsidian>”.

4.7. Estimativa do custo do sistema básico implementado

Mesmo quando comparado com o projeto desenvolvido por Almeida (2018), o valor do sistema pode ser reduzido a quase metade do preço. No entanto, vale ressaltar que os valores de frete não estão inclusos e o modelo de Raspberry utilizado na conta foi o Zero W. O sistema

desenvolvido anteriormente possuiu um custo de R\$ 748,57. Os preços dos novos componentes utilizados podem ser vistos na Tabela 2.

Figura 65 – Configuração dos dispositivos na interface para celular

```

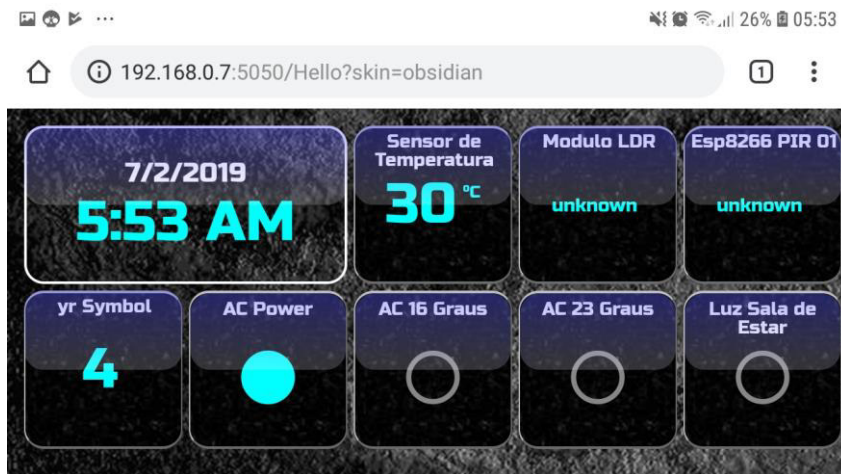
-- title: EnergyS1
-- icon_on: mdi-alpha-l-box
-- icon_off: mdi-alpha-d-box
irenergysaver2:
-- entity: automation.airenergysaver2
-- widget_type: switch
-- title: EnergyS2
-- icon_on: mdi-alpha-l-box
-- icon_off: mdi-alpha-d-box
afetyalarm:
-- entity: automation.safetyalarm
-- widget_type: switch
-- title: SafeAlarm
-- icon_on: mdi-alpha-l-box
-- icon_off: mdi-alpha-d-box
_m_home:
-- entity: automation.i_m_home
-- widget_type: switch
-- title: ImHome
-- icon_on: mdi-alpha-l-box
-- icon_off: mdi-alpha-d-box
---
ayout:
--- clock.clock (2x1), sensor.sensorgas, sensor.sensortemperatura, binary_sensor.modulo_ldr, t
--- alarmon (2x1), airenergysaver1, airenergysaver2, safetyalarm, i_m_home
--- input_number.alarm_clock_hour, input_number.alarm_clock_minute, switch.ac_power, switch.ac_
--- switch.tv.power, switch.sky.power, switch.tv.mute, switch.soonoff, switch.01

```

Fonte: Autor

Para executar as mesmas funções realizadas no projeto, o custo de um sistema de automação residencial no mercado brasileiro normalmente gira em torno de R\$ 3000,00 a R\$ 5000,00, dependendo dos equipamentos utilizados. Esse valor mostra a versatilidade do projeto, principalmente para entusiastas que visam automatizar sua residência por conta própria.

Figura 66 – Interface para dispositivos móveis com *design obsidian*



Fonte: Autor

Tabela 2 – Custo aproximado do sistema de automação residencial implementado

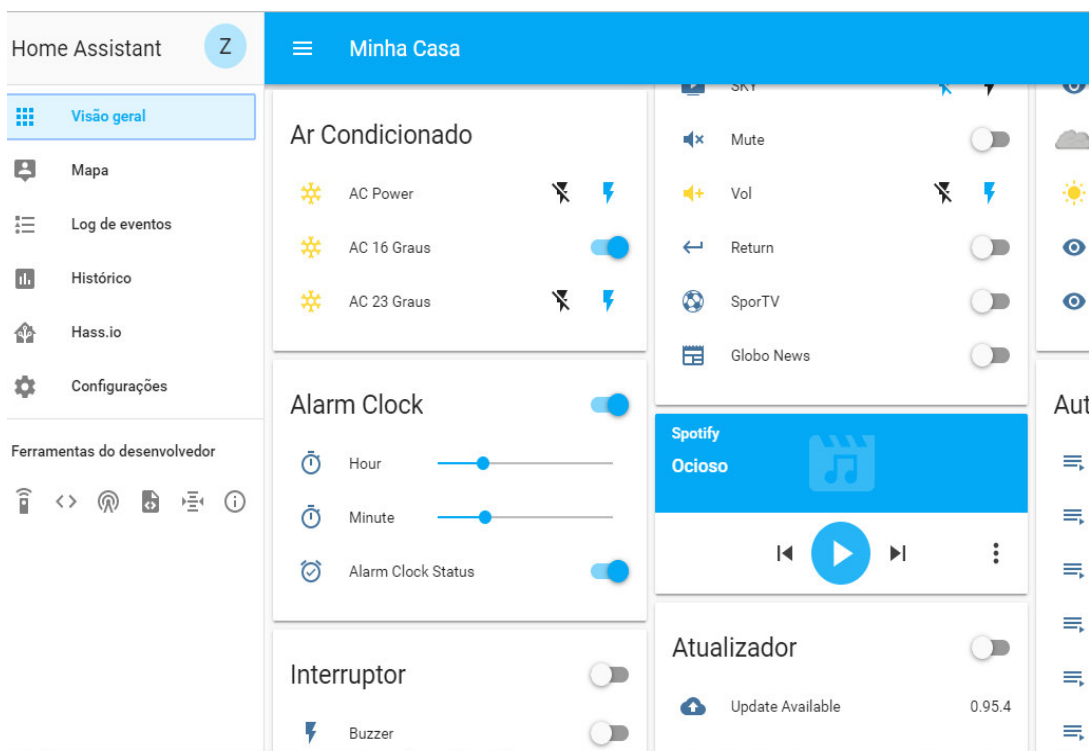
Valores dos Componentes do Sistema de Automação Residencial	
Item	Valor
Raspberry Pi Zero W	R\$ 115,00
Broadlink Rm Mini 3	R\$ 87,37
Sonoff Basic	R\$ 37,00
Sensor de Movimento PIR HC – SR501	R\$ 14,00
Sensor de Gás – MQ4	R\$ 15,00
Sensor de Temperatura DS18B20+	R\$ 11,50
Módulo LDR	R\$ 06,90
ESP8266 – ESP01x3	R\$ 80,70
NodeMCU	R\$ 31,90
Valor Total	R\$ 399,37

Fonte: Autor

5. Experimentos e resultados

Para comprovar a eficiência do sistema, testes foram realizados com todos os dispositivos sensores e atuadores conectados aos mesmos, tanto de forma intencional pelo residente quanto pela execução das duas formas de automação (reativa e perfis customizados). Os experimentos foram realizados utilizando o computador e o celular, verificados por meio da interface conforme mostrada na Figura 67. Os tópicos subsequentes do Capítulo 5 mostram os resultados desses experimentos que comprovam a robustez e eficiência do sistema implementado.

Figura 67 – Interface presente no computador



Fonte: Autor

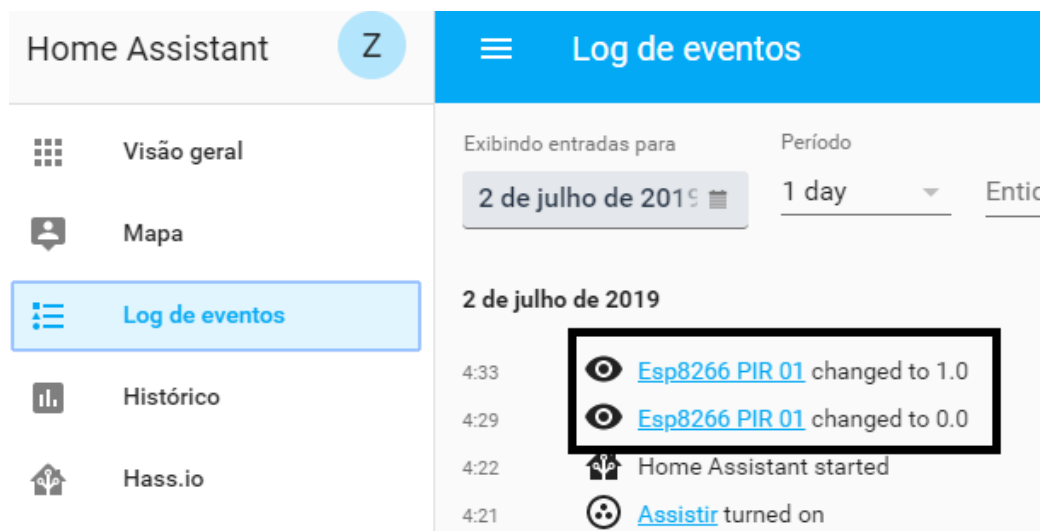
5.1. Leitura dos sensores

O estado dos sensores aparece disponível na interface do *Home Assistant* e para o teste dos mesmos foi necessário gerar estímulos no ambiente e verificar a resposta na interface. Além disso, a leitura dos sensores também permite testar o gatilho das automações reativas e de perfis customizados.

5.1.1. Sensor de movimento PIR HC – SR501

Os estímulos para o sensor de movimento foram realizados com a própria mão, na medida em que esse dispositivo capta movimento e envia à central em um intervalo de 3 a 5 segundos. Se não há movimento nesse intervalo ele também envia essa informação. Para a visualização desses valores a seção *log* de eventos na interface do *Home Assistant* foi utilizada, assim como pode ser visto na Figura 68, onde a mudança de estados está destacada no retângulo.

Figura 68 – Leitura do sensor de movimento no Log de Eventos



Fonte: Autor

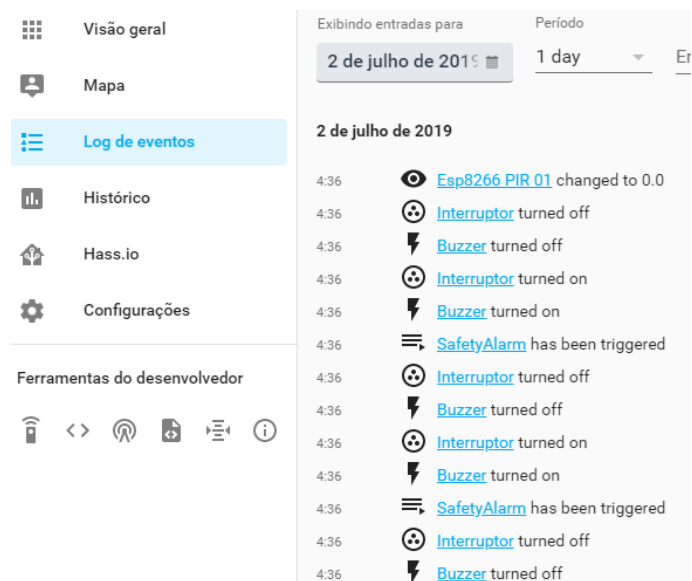
Além disso, ainda utilizando o *log* de eventos foi possível testar a resposta da automação “**SafetyAlarm**”, que visa acionar o *buzzer*, de acordo com a leitura do sensor. Essa programação que ao detectar movimento aciona a buzina está prevista em ambos os tipos de automações. Na Figura 69 é possível observar o gatilho da automação sendo disparado.

5.1.2 Sensor de temperatura DS18B20+

Os estímulos para o sensor de temperatura foram gerados por meio de um isqueiro que, quando acionada a sua chama, permitiu elevar em níveis significativos e em curto período a temperatura do sensor. Na Figura 70 é possível observar a temperatura inicial do sensor.

Já na Figura 71 é possível verificar a temperatura máxima atingida pelo sensor e registrada na interface durante os experimentos.

Figura 69 – “SafetyAlarm” sendo acionada no log de eventos



Fonte: Autor

Figura 70 – Temperatura inicial do sensor



Fonte: Autor

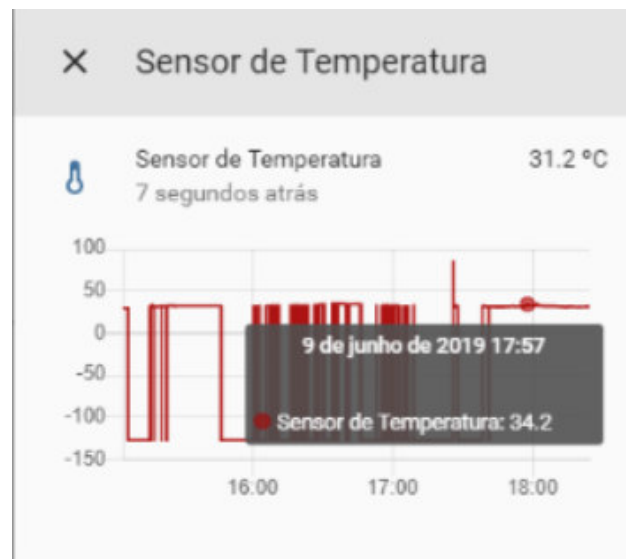
A interface do *Home Assistant* também permite visualizar o histórico de leituras desse sensor. Durante alguns momentos o sensor mostrou instabilidade e sua leitura caiu para -127°C (valor inconsistente), porém ao voltar à estabilidade permaneceu por volta de 31°C , assim como pode ser visto na Figura 72.

Figura 71 – Temperatura máxima atingida pelo sensor durante os experimentos.



Fonte: Autor

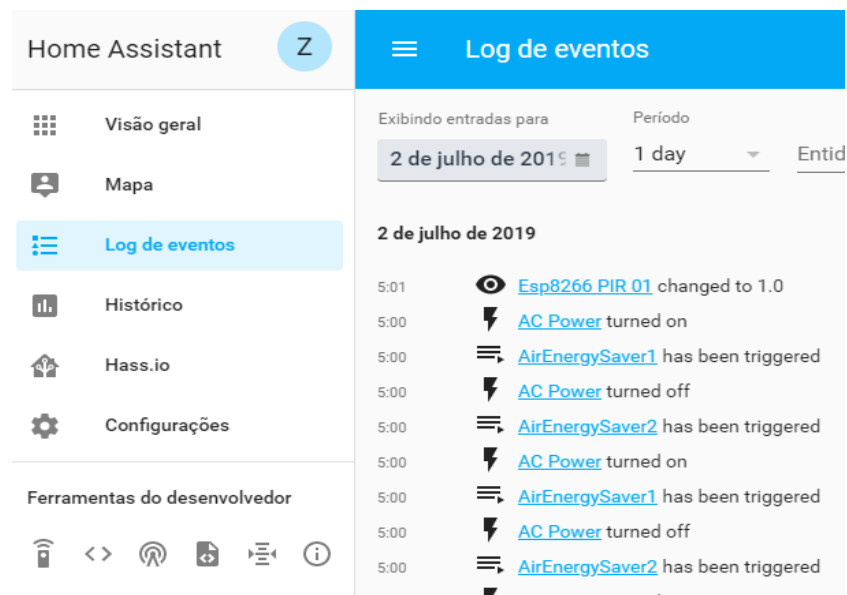
Figura 72 – Histórico de leitura do sensor.



Fonte: Autor

De acordo com a leitura do sensor, as automações denominadas de “**AirEnergySaver1**” e “**AirEnergySaver2**”, que visam ligar e desligar o ar-condicionado, respectivamente, também puderam ser testadas. Para facilitar os experimentos, temperaturas de 32 a 34 graus puderam ser atingidas e medidas pelo sensor com a ajuda da chama de um isqueiro. O acionamento das automações consideradas reativas pode ser visto na interface mostrada na Figura 73.

Figura 73 – Acionamento das automações “AirEnergySaver1” e “AirEnergySaver2”



Fonte: Autor

5.1.3 Sensor de monóxido de carbono – MQ4

Os estímulos realizados para o sensor de gás também foram feitos com o auxílio do vazamento de gás (e não chama) de um isqueiro. O sensor digital retorna valores de 10 bits, 0 a 1024. No entanto, esses números foram mapeados percentualmente e atingem valores de 0 a 100%. Na Figura 74 é possível ver a interface com informações do sensor.

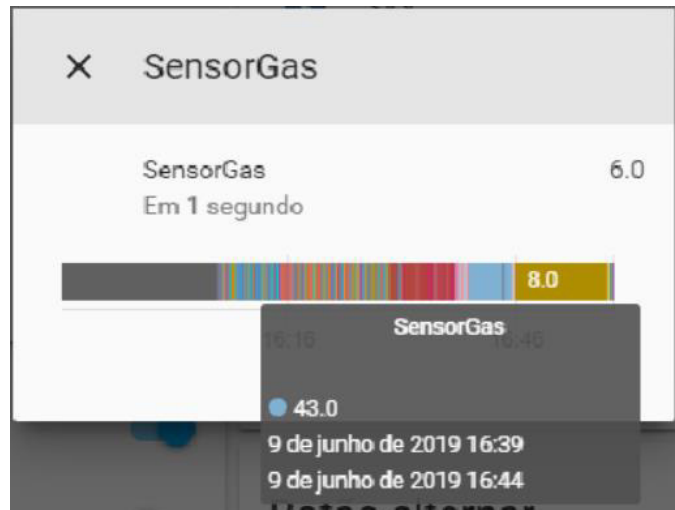
Figura 74 – Valores percentuais do sensor



Fonte: Autor

Também é possível observar o histórico de medição desse sensor, assim como mostrado na Figura 75.

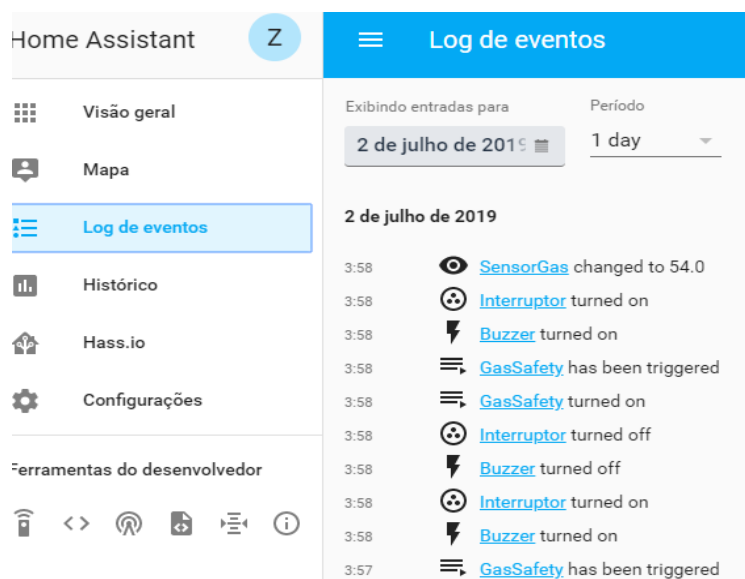
Figura 75 – Histórico de medição do sensor de gás



Fonte: Autor

Para valores percentuais acima de 40, a automação reativa “**GasSafety**” também é acionada ativando o *buzzer*, como pode ser visto no *log* de eventos na Figura 76.

Figura 76 – Acionamento da automação “**GasSafety**”



Fonte: Autor

5.1.4 Módulo LDR

De forma similar ao sensor de presença, o de luminosidade funcionou com somente dois estados, 0 ou 1, indicando claro ou escuro, respectivamente. Os testes foram realizados escurecendo o ambiente com as próprias mãos e ajustando a chave de sensibilidade do módulo. Os valores analisados podem ser vistos na seção *log* de eventos mostrada na Figura 77.

Figura 77 – Log de eventos módulo LDR.



Fonte: Autor

A automação de perfil “**VacationOn**” também foi ativada de acordo com as leituras do sensor, ativando as luzes da residência, assim como pode ser visto no *log* de eventos da Figura 78.

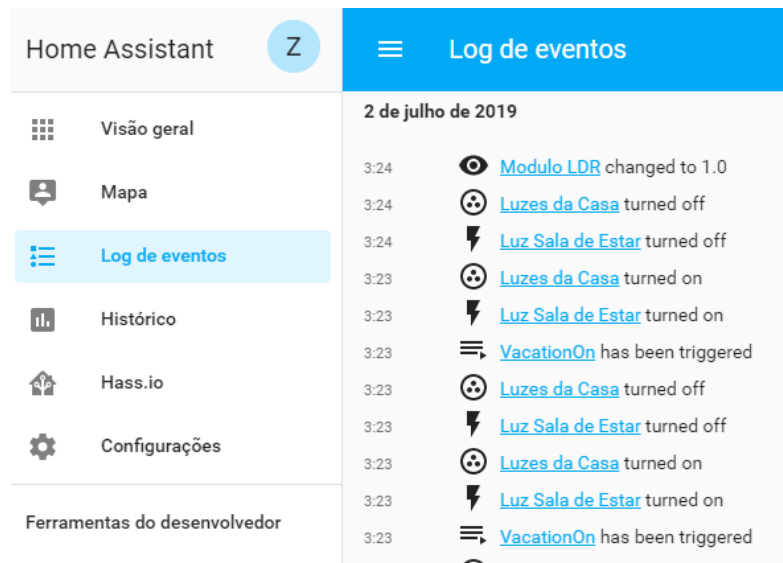
5.2 Acionamento dos atuadores

Por meio dos dados monitorados pela interface do *Home Assistant*, também foi possível verificar a eficiência dos atuadores que foram acionados de forma consecutiva com o intuito de ligar ou desligar dispositivos e registrar qualquer falha. Esses dados puderam ser observados na seção *log* de eventos.

5.2.2 Broadlink Rm Mini 3

Os dispositivos controlados por meio de sinais infravermelhos como ar-condicionado, televisão e o receptor de TV a cabo foram acionados remotamente mais de dez vezes cada sem o registro de falhas, como pode ser visto na Figura 79. Infelizmente por problemas de espaço para

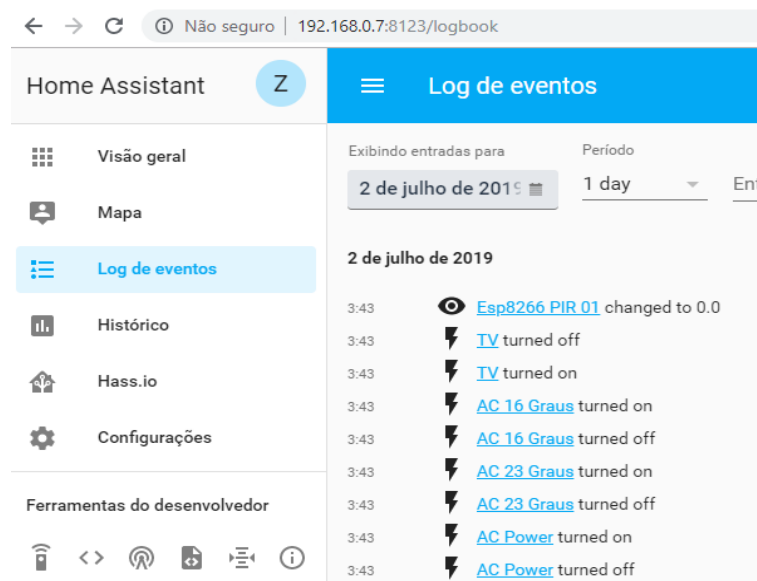
Figura 78 – Log de eventos automação “VacationOn”



Fonte: Autor

apresentação de todos os eventos, apenas oito foram apresentados. Além disso, o dispositivo também pode ser testado durante automações como a mostrada na Figura 73.

Figura 79 – Acionamento remoto dos dispositivos controlados por infravermelho

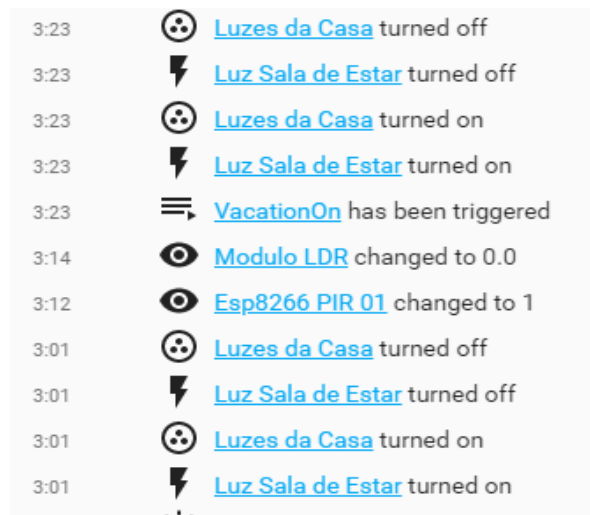


Fonte: Autor

5.2.3 Sonoff

O *Sonoff* também foi testado executando o acionamento remoto da lâmpada da sala de estar, onde esse atuador está instalado. Os acionamentos foram realizados mais de dez vezes sem o registro de falhas e alguns desses eventos podem ser vistos na Figura 80. Além disso, o dispositivo também pode ser testado na automação “**VacationOn**” mostrada na Figura 72.

Figura 80 – Acionamentos da lâmpada da sala de estar.



Fonte: Autor

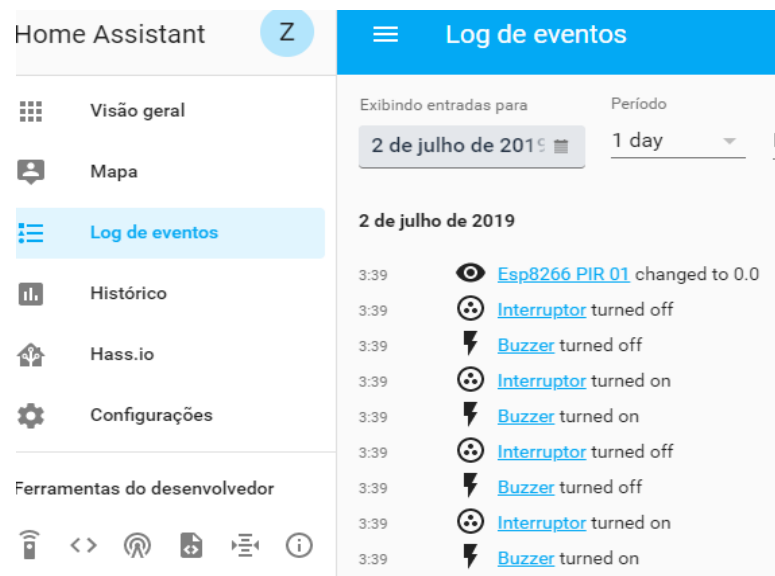
5.2.4 Sirene

O *buzzer* foi conectado a um dos pinos digitais ociosos do NodeMCU e acionado remotamente pela central. Os acionamentos foram feitos mais de dez vezes, sem falhas, e alguns deles podem ser vistos na Figura 81. Além disso, o dispositivo também foi testado nas automações reativas “**SafetyAlarm**” e “**GasSafety**” vistos na Figura 69 e Figura 76, respectivamente.

5.3 Execução de perfis customizados

Os perfis customizados são automações que visam simular a presença de pessoas na casa durante o período em que a residência encontra-se desocupada. Sua finalidade é a de minimizar a tentativa de invasão de pessoas estranhas, acionando e desligando dispositivos em horários pré-definidos, como lâmpadas, TV ou outros equipamentos. No caso desse projeto, devido ao limite

Figura 81 – Acionamento consecutivo do *buzzer* na seção *log* de eventos.



Fonte: Autor

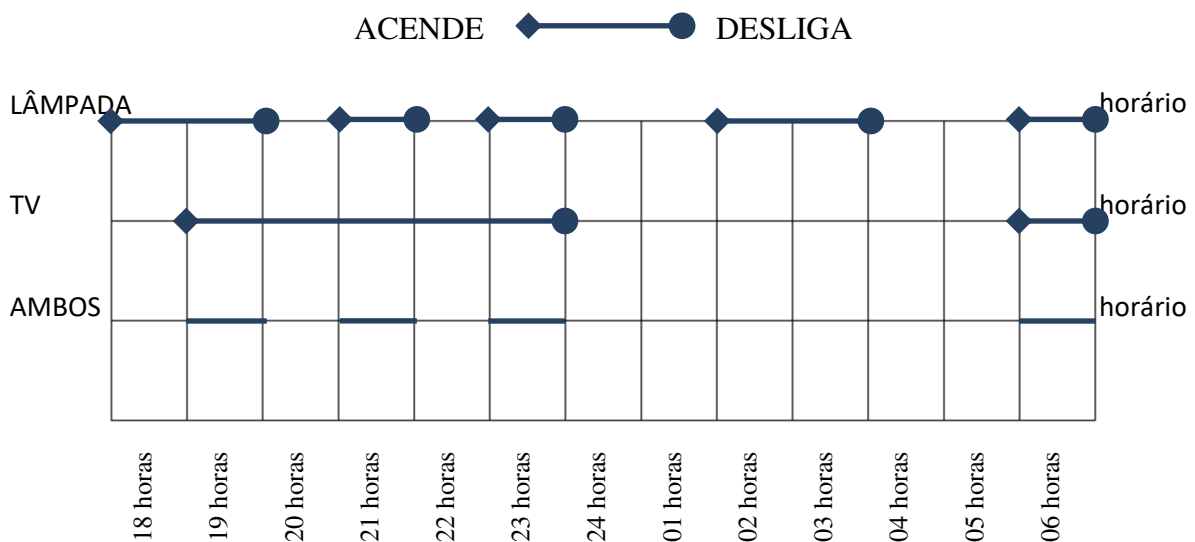
de dispositivos atuadores que podem ter um perfil com essa finalidade, será usado o interruptor e a TV.

Apenas a título de exemplo, é feito um perfil durante uma noite em que o usuário deste sistema ficará ausente na residência, supondo das 18 h às 7 horas do dia seguinte, ou seja, 13 horas. Um possível perfil customizado seria acionar e desligar a lâmpada e a TV nas seguintes condições: apenas a lâmpada ou apenas a TV. Evidentemente ambas poderão estar ligados ou desligados em período do perfil programado.

Para melhor compreender o perfil customizado neste exemplo é feito um gráfico de tempo no intervalo citado, onde os tempos de acionar e desligar os equipamentos podem ser mais bem observados, conforme ilustrado na Figura 82. Podem ser observadas três linhas do tempo no horário definido no perfil, onde são mostrados os acionamentos e desligamentos apenas da lâmpada, apenas da TV e quando ambos estão em operação (não marcadas ambas as ações mencionadas).

Para fins de experimentação para validação da execução do perfil, foi adotada uma escala no tempo, já que aguardar 13 horas para realizar os testes é impraticável. Na escala usada na diminuição do tempo total do experimento, cada hora foi considerada como 1 minuto e, dessa maneira, esse experimento teve a duração de 13 minutos. Em diversos ensaios realizados o perfil foi executado de acordo com o programado, conforme pode ser observado parcialmente na Figura

Figura 82 – Temporização na customização de um perfil de 13 horas



Fonte: Autor

83 nos tempos indicados na interface. Perceba que o horário escolhido para iniciar o experimento da execução do perfil foi 01:01h e os demais eventos de ligar e desligar a lâmpada e a TV ocorrem dentro do especificado, com aplicação da escala. Por exemplo, o primeiro acionamento e desligamento da lâmpada ocorre as 01:01h e 01:03h, enquanto o da TV são 01:02 e 01:07, respectivamente.

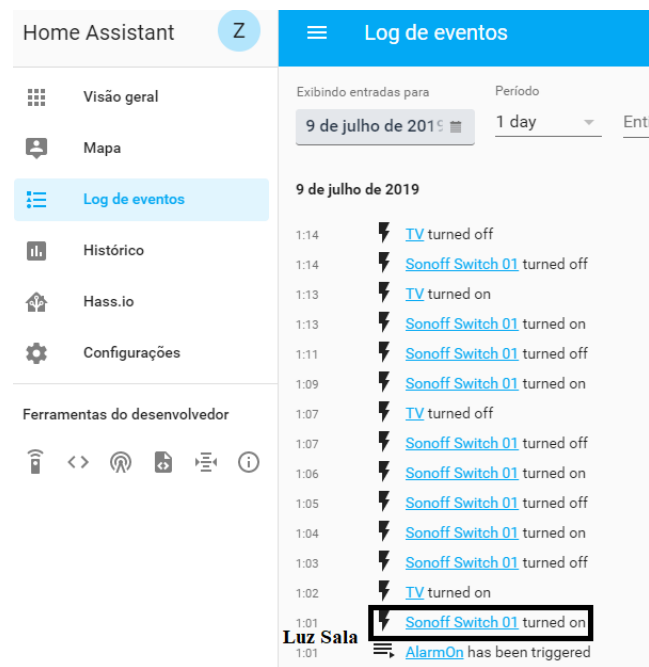
5.4 Acionamento por meio da interface móvel

A interface móvel foi testada por meio de acionamentos sucessivos, de forma similar às automações, no entanto, o *log* de eventos só pode ser acessado por meio da interface original, não sendo proveitoso mostrar a sua imagem. A interface móvel não mostrou nenhuma falha, funcionando de forma similar a interface primária do *Home Assistant*. Os dispositivos acionados na interface móvel ficam destacados em azul assim como mostrado na Figura 84.

5.5 Histórico de atividades

A interface do *Home Assistant* também fornece acesso ao histórico de atividades, permitindo acesso a todos os acionamentos feitos pelo usuário de acordo com a data e horário dos

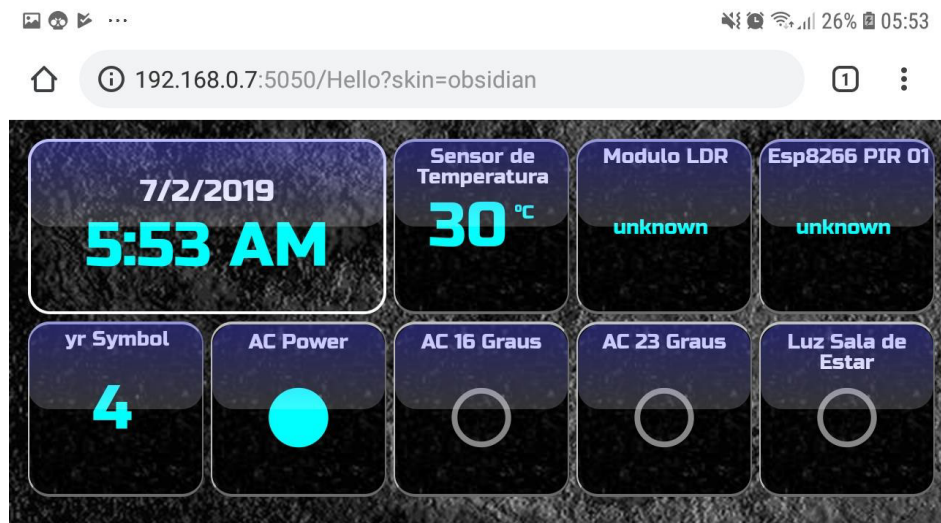
Figura 83 – Perfil customizado acionado nos horários programados



Fonte: Autor

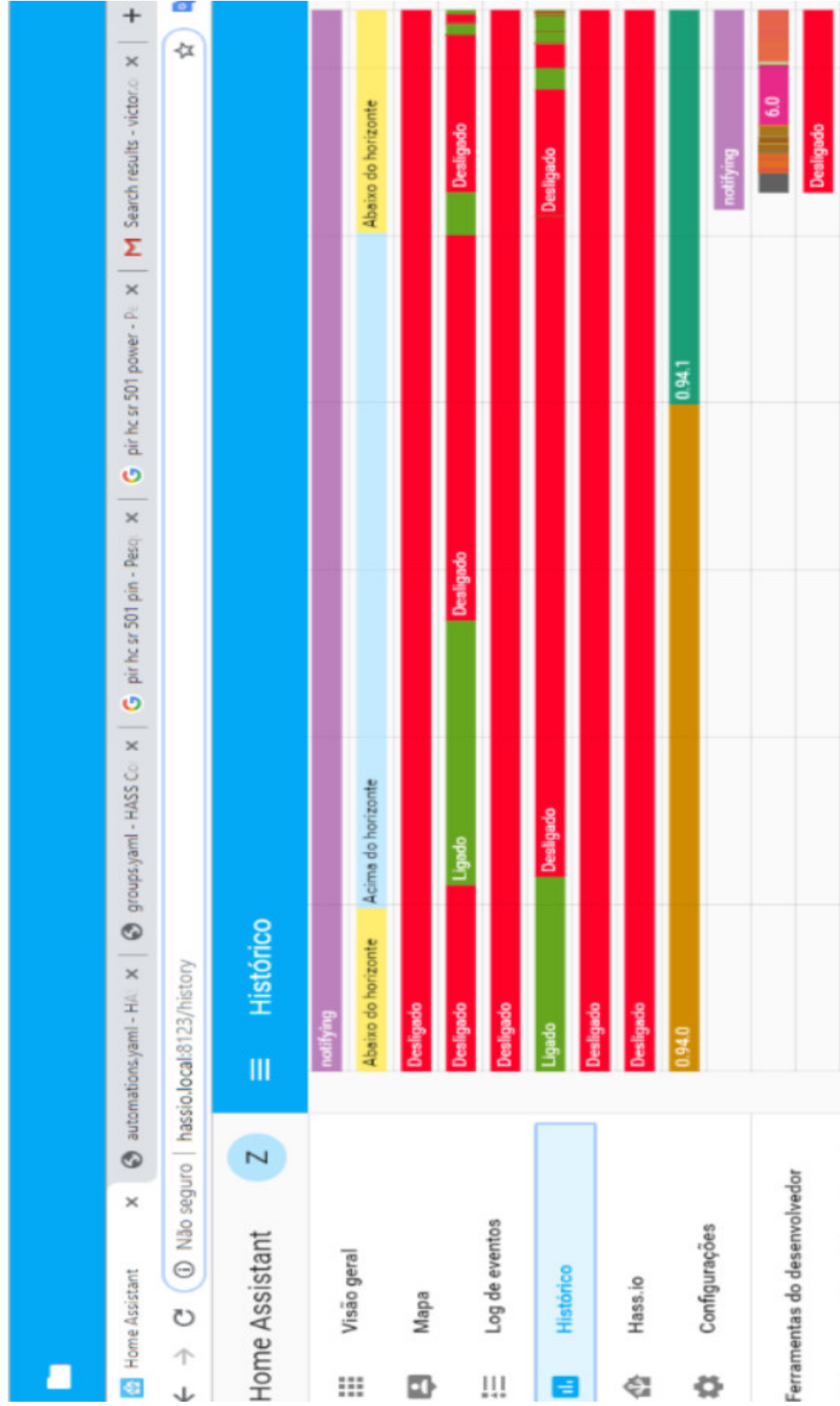
acontecimentos. Além de manter um rastreamento das leituras efetuadas pelos sensores, constitui mais uma ferramenta de gestão do sistema de automação, assim como mostrado na Figura 85.

Figura 84 – Destaque de dispositivos acionado na interface móvel.



Fonte: Autor

Figura 85 – Histórico de atividades do Home Assistant



Fonte: Autor

6. CONCLUSÕES E TRABALHOS FUTUROS

Essa proposta de trabalho deu continuidade a outra monografia que tinha o foco no controle por comandos de voz via a interface Alexa Amazon, tendo desenvolvido um sistema de automação residencial com comunicação *wireless*. Neste trabalho também foi realizada os acionamento remotos de dispositivos presentes na residência, como televisores, lâmpadas e ar-condicionado. Porém, processos de automação foram acrescentados, sendo essa a principal contribuição deste trabalho. Para prover o sistema de automação com a capacidade de realizar automações programadas, foi necessário substituir a interface Alexa pelo Raspberry Pi 3B, este último usado em função da sua disponibilidade, porém podendo ser substituído pela versão Zero W de mais baixo custo.

Os elementos do sistema (atuadores, sensores e central de automação) foram instalados numa residência de São Luís juntamente com o *software Home Assistant*, permitindo os testes de todos os elementos e automações programadas, como apresentados nos experimentos, os quais mostraram a eficiência e confiabilidade do sistema implementado. Para os casos específicos das validações das automações, foram consideradas as duas categorias:

- a) Atuação reativa baseada em informações de sensores (não humana) em eventos programados como, por exemplo, acionamento de sirene na detecção de vazamento de gás; e
- b) Execução de perfis customizados pré-programados para ocorrer em horários específicos na ausência de pessoas na casa, simulando a presença de residentes pelos acionamentos definidos nesses perfis como, por exemplo, lâmpadas e TV.

O presente trabalho mostra com a tecnologia hoje disponibilizada em sistemas embarcados, além da oferta de uma grande variedade de dispositivos atuadores e sensores, viabiliza que um sistema de automação residencial eficiente e de baixo custo é plenamente possível de ser implementado, sendo que sua complexidade dependente do grau de automação residencial desejado pelo morador da casa.

Como trabalhos futuros que podem ser conduzidos de forma imediata, a partir do atual estágio alcançado com a implementação desta monografia, pode-se sugerir:

- a) A ampliação do sistema através da inserção de novos sensores e atuadores para controle e sensoriamento de diferentes recursos ou ambientes de uma residência;

- b) A integração de câmeras IP ao sistema de automação, possibilitando a capacidade de visualização e da gravação de eventos, podendo eles ser inclusive utilizados para a ação pré-programada nos perfis do sistema de automação; e
- c) Habilitar o acesso do sistema via redes externas, possibilitando o monitoramento e acompanhamento da residência mesmo fora da rede local.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, Z. **Sistema de Automação Residencial de Baixo Custo em Rede Wi-Fi**. Monografia do Curso de Graduação em Engenharia Elétrica, Universidade Federal do Maranhão, São Luís, 2018, 70 p.
- BORGES, F. et al. **Automação**. São Paulo: [s.n.] 2000.
- DINIZ, A. **Acionamento e Monitoramento de Sistemas Embarcados Via Web**. Monografia do Curso de Graduação em Engenharia Elétrica, Universidade Federal do Maranhão, São Luís, 2017, 118 p.
- FERREIRA, A. **Implementação de Uma Estrutura Cliente-Servidor para Controle e Monitoramento de Dispositivos Usando o Raspberry Pi 3B**. Monografia do Curso de Graduação em Engenharia Elétrica, Universidade Federal do Maranhão, São Luís, 2018, 71 p.
- MONDEVAIN, D.; SILVA, D. S.; SILVA, V. P. B. **Automação Residencial e Predial e a Internet das Coisas**. São Paulo: [s.n.], 2016
- MURATORI, J. R.; DAL BÓ, P. H. **Automação Residencial Conceitos e Aplicações**. 2ª Ed. Belo Horizonte: Educere, 2014.
- RIBEIRO, D. V. ROSELI: **Robô Seguidor de Linha para Mapeamento de Ambientes Internos**. Monografia do Curso de Graduação em Engenharia Elétrica, Universidade Federal do Maranhão, 2017, 94 p.