

UNIVERSIDADE FEDERAL DE ALFENAS

MILENE GABRIELLE DE FARIA

**ESTUDO DE VIABILIDADE DO USO DO HOME ASSISTANT COMO
FERRAMENTA PARA AUTOMAÇÃO RESIDENCIAL**

Alfenas/MG

2023

MILENE GABRIELLE DE FARIA

**ESTUDO DE VIABILIDADE DO USO DO HOME ASSISTANT COMO
FERRAMENTA PARA AUTOMAÇÃO RESIDENCIAL**

Trabalho apresentado como parte dos requisitos para
obtenção do título de bacharel em Ciência da
Computação pela Universidade Federal de Alfenas.
Área de concentração: Internet das Coisas.

Orientador: Luiz Eduardo da Silva.

Alfenas/MG

2023

RESUMO

A Internet das Coisas (IoT) tem se destacado muito no mercado de Tecnologia da Informação, fato que pode ser observado através da popularização dos dispositivos inteligentes, como por exemplo, os aparelhos de assistente virtual. Alguns dos fatores que contribuem para o crescimento do ramo se dão pela disseminação do uso da Internet e o aumento do número de dispositivos conectados, o que cria um ambiente favorável para a evolução da IoT. A IoT se destaca por sua versatilidade, onde antes era empregada principalmente no meio comercial e industrial, passou a conquistar o cenário domiciliar, sendo utilizada como ferramenta para a automação residencial. Em relação ao âmbito domiciliar, existem novos aspectos a serem considerados, uma vez que as soluções inteligentes estarão direcionadas a um grupo de usuários com limitações e necessidades específicas. Para este presente trabalho, será considerado o custo e a facilidade de implementação ao propor uma solução inteligente simples e economicamente acessível e, para isto, será apresentado o *Home Assistant*, um aplicativo *open source* e gratuito, como plataforma de gerenciamento de dispositivos inteligentes, onde será utilizado um protótipo eletrônico para automatizar um serviço de envio de *e-mail* através da detecção de temperatura de um sensor. A composição deste trabalho se dá por um levantamento bibliográfico, uma apresentação do *Home Assistant*, a documentação dos processos de instalação do aplicativo e criação de uma solução IoT acessível. Os resultados evidenciam a versatilidade e a facilidade da utilização da ferramenta, onde se destaca principalmente a facilidade na criação de sistemas de automação, o que abrange a comunicação de diferentes tipos de dispositivos eletrônicos com a plataforma e a utilização de serviços disponibilizados para a construção de um sistema de automação.

Palavras-chave: Automação residencial. Internet das Coisas. Home Assistant.

ABSTRACT

The Internet of Things (IoT) has stood out a lot in the Information Technology market, a fact that can be observed through the popularization of smart devices, such as virtual assistant devices. Some of the factors that contribute to the growth of the sector are due to the spread of Internet use and the increase in the number of connected devices, which creates a favorable environment for the evolution of IoT. The IoT stands out for its versatility, where before it was used mainly in the commercial and industrial environment, it has now conquered the home scenario, being used as a tool for home automation. Regarding the home environment, there are new aspects to be considered, since the smart solutions will be directed to a group of users with specific limitations and needs. For this present work, the cost and ease of implementation will be considered when proposing a simple and economically accessible smart solution and, for this, the Home Assistant will be presented, an open source and free application, as a platform for managing smart devices, where an electronic prototype will be used to automate an e-mail sending service by detecting the temperature of a sensor. The composition of this work is based on a bibliographical survey, a presentation of the Home Assistant, the documentation of the application installation processes and the creation of an accessible IoT solution. The results show the versatility and facility of use of the tool, where the ease in creating automation systems stands out, which includes the communication of different types of electronic devices with the platform and the use of services available for building an automation system.

Palavras-chave: Home Automation, Internet of Things, Home Assistant.

LISTA DE FIGURAS

Figura 3.1 - ESP32 (SoC).....	14
Figura 3.2 - Dispositivos ESP32.....	14
Figura 4.1 - Pilha do Home Assistant.....	17
Figura 4.2 - Funcionamento do Home Assistant core.....	17
Figura 4.3 - Fluxo de informações de uma automação.....	18
Figura 4.4 - Métodos de instalação do Home Assistant.....	19
Figura 5.1 - Instalação da máquina virtual.....	21
Figura 5.2 - Definição do nome e sistema operacional da máquina virtual.....	22
Figura 5.3 - Sumário com as configurações da máquina.....	23
Figura 5.4 - Inicialização do Home Assistant.....	23
Figura 5.5 - Primeiro acesso ao Home Assistant.....	24
Figura 5.6 - Visão Geral do Home Assistant.....	25
Figura 5.7 - Reconhecimento do dispositivo pelo ESPHome.....	25
Figura 5.8 - Dispositivo reconhecido pelo Home Assistant.....	26
Figura 5.9 - Configuração do sensor.....	27
Figura 6.1 - ESP32 (Pinagem).....	28
Figura 6.2 - DHT22 (Pinagem).....	28
Figura 6.3 - Protótipo do sensor.....	29
Figura 6.4 - Detecção de Temperatura e Umidade pelo sensor.....	30
Figura 7.1 - Dispositivos, serviços e suas entidades.....	31
Figura 7.2 - Entidades do sensor.....	31
Figura 7.3 - Dashboard com exibição das entidades.....	32
Figura 7.4 - Gatilhos disponibilizados para o dispositivo.....	34
Figura 7.5 - Configuração do serviço.....	35
Figura 7.6 - Notificação enviada pela aplicação.....	35

LISTA DE ABREVIACOES E SIGLAS

IOT	Internet of Things
SoC	System on Chip
SRAM	Static Random Access Memory
ROM	Read-Only Memory
RF	Radio Frequency
DevKit	Developer Kit
HASS	Home Assistant
OS	Operating System
SBC	Single-board Computer
CPU	Central Process Unit
EFI	Extensible Firmware Interface
AP	Access Point
VCC	Common Collector Voltage
GND	Ground (Pino Terra)
SMTP	Simple Mail Transfer Protocol

SUMÁRIO

1 INTRODUÇÃO.....	8
1.1 OBJETIVO GERAL.....	7
1.2 OBJETIVOS ESPECÍFICOS.....	8
1.3 JUSTIFICATIVA.....	8
1.4 METODOLOGIA.....	9
2 REVISÃO BIBLIOGRÁFICA.....	9
2.1 AUTOMAÇÃO RESIDENCIAL.....	9
2.2 ELEMENTOS BÁSICOS DE UMA AUTOMAÇÃO RESIDENCIAL.....	10
2.3 TIPOS DE AUTOMAÇÃO RESIDENCIAL.....	11
2.4 INTERNET DAS COISAS (IOT).....	11
2.5 DISPOSITIVOS INTELIGENTES.....	12
3 COMPONENTES ELETRÔNICOS UTILIZADOS.....	12
3.1 ESP32.....	12
3.2 DHT22.....	14
4 HOME ASSISTANT.....	15
4.1 VISÃO GERAL.....	15
4.2 ARQUITETURA.....	16
4.3 MÉTODOS DE INSTALAÇÃO.....	18
5 INSTALAÇÃO.....	20
5.1 VIRTUALBOX.....	21
5.2 HOME ASSISTANT OS.....	21
5.3 ESPHOME.....	25
6 CONSTRUÇÃO DO PROTÓTIPO.....	28
7 AUTOMAÇÃO.....	31
7.1 ADICIONANDO AS ENTIDADES DO SENSOR.....	31
7.2 ADICIONANDO O SERVIÇO.....	32
7.3 CRIANDO A AUTOMAÇÃO.....	33
7.4 CONFIGURAÇÃO DE OUTROS SENSORES NO HOME ASSISTANT.....	35
8 CONCLUSÃO E TRABALHOS FUTUROS.....	37
REFERÊNCIAS.....	39

1 INTRODUÇÃO

Diante do cenário tecnológico atual, sobretudo considerando as influências da Indústria 4.0, o desenvolvimento de soluções voltadas para a área de automação seguem sendo uma forte tendência no mercado. A IoT (Internet of Things) faz parte deste grupo de tecnologias emergentes que têm modificado a interação do ser humano com o mundo digital, promovendo a interconexão e interoperabilidade de diferentes dispositivos em rede de forma que troquem informações e funcionem de maneira autônoma.

Segundo Morais (2018) o termo IoT “é usado para transmitir a ideia de que a internet pode estar presente em todas as coisas”. Em um momento como o atual, onde a internet tem se fundido cada vez mais no cotidiano das pessoas, o sucesso do ramo se torna premeditado. De acordo com Atzori, Iera e Morabito (2010), a demanda popular, combinada com os avanços tecnológicos, podem impulsionar a difusão generalizada da IoT e também contribuir para o desenvolvimento econômico de outros setores.

A IoT pode ser aplicada em diversos contextos. Quando consideramos o contexto doméstico, as soluções inteligentes podem ser usadas na monitoração e na automatização de processos rotineiros. Pinheiro (2004) afirma que a automação residencial tem como objetivo tornar a vida no lar “mais prática, segura e confortável, sem deixar de lado o aspecto pessoal e familiar necessário a um agradável ambiente”.

Apesar do aumento da procura de serviços especializados no país (LOGICALIS, 2022), o uso da IoT ainda está concentrado, em grande parte, no setor empresarial e industrial, não estando completamente difundida como uma tecnologia que também trás benefícios no uso domiciliar. Dentre os desafios relacionados a este aspecto, o custo dos dispositivos e a falta de conhecimento acerca da tecnologia podem ser alguns dos empecilhos.

Sendo assim, para esse presente trabalho será feito um estudo de viabilidade do uso de uma aplicação *open source* e gratuita – *Home Assistant* – para automação residencial, com o objetivo de demonstrar a construção de um sistema inteligente que seja simples e barato. Será apresentado o funcionamento do aplicativo, bem como, um manual de instalação e em seguida será demonstrado a integração de um dispositivo à plataforma. Será feita a construção de um protótipo de baixo custo utilizando o sensor DHT22 e um microcontrolador ESP32 e que, através de um serviço de *e-mail* do *Home Assistant*, enviará uma notificação de acordo com a leitura de temperatura feita pelo sensor.

Este trabalho está organizado da seguinte forma. Primeiro (i) será apresentado os objetivos, justificativa e metodologia deste trabalho, no capítulo (ii) será apresentado a

revisão bibliográfica dos principais conceitos ligados à automação residencial. Em seguida será listado os componentes eletrônicos utilizados. No capítulo (iv), será apresentado o *Home Assistant* e seu funcionamento, seguido pelos processos de instalação da plataforma (v). A construção do protótipo e integração deste com a aplicação é mostrado logo após (vi) e o desenvolvimento do sistema de automação é feito depois (vii). Por último, será apresentado as considerações finais e conclusão (viii).

1.1 OBJETIVO GERAL

O objetivo principal deste trabalho é realizar um estudo de viabilidade do uso do Home Assistant para a automação residencial, considerando os aspectos de simplicidade de implementação de um sistema inteligente e seu custo. Através deste trabalho, será disponibilizada uma documentação completa de instalação do *Home Assistant*, uma apresentação de seus recursos principais e o desenvolvimento de uma automação na plataforma.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar levantamento e estudo bibliográfico;
- Reunir e organizar um passo a passo referentes a instalação *Home Assistant*;
- Apresentar os principais recursos da plataforma;
- Produzir um protótipo de automação residencial de baixo custo;
- Integrar o dispositivo;
- Desenvolver uma automação utilizando um serviço do aplicativo;

1.3 JUSTIFICATIVA

As aplicações IoT podem ser muito úteis no contexto domiciliar. Elas podem ser utilizadas na implementação de sistemas de segurança, monitoração de ambientes e automatização de tarefas cotidianas. Essas aplicações podem aumentar a qualidade de vida trazendo mais comodidade ao dia-a-dia. Contudo, os produtos e serviços que fazem uso dessa tecnologia tendem a ter um alto custo no mercado, sendo inacessíveis para grande parte da população. Somando a isso, a falta de informação acerca do assunto também pode causar um certo afastamento de um público que não necessariamente pertence a área de tecnologia. Tendo isso em consideração, é importante que existam opções de soluções inteligentes que

sejam simples, eficientes e baratas, permitindo um maior acesso e usufruto desses sistemas por um maior número de pessoas.

O *Home Assistant* fornece um ambiente completo de controle e automação de dispositivos e o acesso ao aplicativo é gratuito. Ele possui uma *interface* amigável e intuitiva, podendo ser facilmente navegado por qualquer pessoa que já possui o hábito de utilizar aplicativos *online*. A plataforma é instalada já possuindo serviços ligados a automação, eliminando a necessidade de uma programação pesada para implementar uma solução, além disso, possui uma ampla gama de *add-ons* (módulos que adicionam funcionalidades extras a plataforma do *Home Assistant*) específicos para produtos do mercado, o que pode facilitar a construção de sistemas em que se queira utilizá-los.

Através deste trabalho, será possível construir e integrar um sensor de temperatura no *Home Assistant* e utilizar um serviço da plataforma para enviar mensagens de *e-mail* conforme os valores captados pelo sensor e através disso, dar os primeiros passos no processo de automação residencial.

1.4 METODOLOGIA

Este trabalho trata-se de uma pesquisa aplicada, tendo como objetivo apresentar uma proposta de solução inteligente de baixo custo para automação residencial.

Foi realizado um levantamento de documentações, fóruns, livros e artigos científicos que abrangem os tópicos abordados e que constituem a base do desenvolvimento dessa pesquisa.

O trabalho é explicativo, conforme é registrado em formato passo a passo, todos os processos realizados durante o desenvolvimento da solução.

Por fim, é apresentada uma conclusão dedutiva através dos resultados obtidos durante a elaboração da pesquisa.

2 REVISÃO BIBLIOGRÁFICA

2.1 AUTOMAÇÃO RESIDENCIAL

A automação surgiu como um produto da revolução industrial que tinha como um dos objetivos criar ferramentas capazes de executar tarefas repetitivas a fim de diminuir o esforço de um trabalho até então manual. Essa concepção e sua influência na indústria, foram responsáveis pelas mais brilhantes invenções tecnológicas, aumentando o nível de conforto, segurança e eficiência em processos que passaram a ser feitos de forma mecanizada.

Segundo Muratori e Dal Bó (2011), as primeiras incursões das tecnologias de automação datam ao final da década de 1970 e tiveram sua origem nos Estados Unidos. Nesse período, as soluções eram simples e resolviam situações pontuais, como ligar remotamente algum equipamento ou luzes.

Os primeiros módulos ligados à automação eram desenvolvidos, sobretudo, para construções prediais, usados, principalmente, em indústrias e ambientes corporativos. Porém, a incorporação de sensores e outros elementos tecnológicos, fez com que a automação pudesse ser implantada em cenários mais comuns como o domiciliar, dando início a formação de um novo ramo dentro da automação, a automação residencial.

A automação residencial também pode ser referenciada como domótica:

O termo domótica resulta da junção da palavra romana *domus*, que se refere à casa, com a palavra robótica, que, por sua vez, refere-se à realização de controle automatizado de algo por robôs, mas que pode ser simplificado pela automatização do ambiente em si (STEVAN JUNIOR, 2018, p. 16).

A domótica criou um espaço para uma maior variedade de produtos e serviços ligados à automação além de influenciar o mercado de dispositivos eletrônicos. Aspectos como *design* – dimensão, cabeamento, adaptação ao estilo do ambiente – e custo passaram a ser considerados, uma vez que o contexto domiciliar difere-se bastante do predial, onde os primeiros dispositivos para este fim foram usados. Segundo Bolzani (2004), a automação predial utiliza um usuário padrão para direcionar o planejamento e instalação dos sistemas, ao contrário da automação residencial que é orientada a um grupo de usuários que possuem necessidades específicas, o que faz com que as aplicações sejam muito mais customizáveis.

2.2 ELEMENTOS BÁSICOS DE UMA AUTOMAÇÃO RESIDENCIAL

Um sistema de automação residencial geralmente é composto por três elementos básicos: sensores, controladores e atuadores (STEVAN JUNIOR, 2018). Os sensores são capazes de medir e monitorar variáveis físicas do ambiente convertendo-as em um valor passível de manipulação por sistemas computacionais. (ACCARDI; DODONOV, 2012). Já os controladores são incumbidos de receber os dados dos sensores e decidir a ação a ser tomada. Funcionam como intermediários fazendo a comunicação dos dispositivos de entrada (sensores) e saída (atuadores). Para uma aplicação de monitoramento, por exemplo, o controlador é programado para receber os dados captados do sensor e transportá-los para um atuador que poderia ser um alarme. Os atuadores são dispositivos que modificam uma variável controlada recebendo um sinal proveniente do controlador e agindo sobre o sistema controlado (WENDLING, 2010), podendo ser em forma de algum sinal (sonoro, visual, etc.) ou alguma ação no espaço físico.

2.3 TIPOS DE AUTOMAÇÃO RESIDENCIAL

É possível agrupar os sistemas de automação residencial em três níveis de interação, que estão relacionados à autossuficiência dos seus componentes e à comunicação entre eles (TEZA, 2002):

- **Sistemas Autônomos:** sistemas que possuem módulos pré-programados que funcionam de maneira autônoma. Neste nível, uma residência pode contar com uma ou mais aplicações autônomas em que cada módulo executa sua função, porém sem comunicação entre eles.
- **Sistemas Integrados:** módulos ou subsistemas que operam sob um controlador central com comunicação entre si. Nesse sistema, os dispositivos podem trocar informações, porém mantêm-se limitados às suas funções específicas.
- **Residência Inteligente:** conciliação de diversos subsistemas que compõem uma automação residencial. As características desse nível de maturidade de automação, incluem a formalização de um sistema arquitetural com serviços customizáveis e integração a outros módulos com interoperabilidade entre eles. Esses sistemas “dependem de comunicação de mão-dupla e re-alimentação de status entre todos os subsistemas” (TEZA, 2002).

2.4 INTERNET DAS COISAS (IOT)

A IoT surgiu na segunda década do século XXI baseada na ideia de que qualquer coisa pode fornecer informações e ser acessada *online* em qualquer lugar. (STEVAN JUNIOR, 2018). Ela consegue proporcionar aos objetos do dia-a-dia (quaisquer que sejam), capacidade computacional e de comunicação (SANTOS, B. P. et al, 2016).

Para muitos autores, não é possível abstrair a IoT da Internet em si, sendo entendida como uma extensão da mesma, que pode ser percebida através do uso de componentes como sensores ou atuadores que irão interagir com o espaço físico. Na perspectiva de Miorandi et al. (2012), a Internet não se encaixa no conceito convencional de uma estrutura de rede que alcança terminais finais de usuários, mas sim uma estrutura que possui uma disposição de dispositivos inteligentes interconectados a ela. Essa interação é capaz de fornecer ambientes computadorizados envolventes de forma que se torna normal a difusão destes elementos com a vida real.

Relógios, televisores, geladeiras, impressoras, assistentes virtuais, são exemplos de eletrodomésticos que passaram a contar com recursos IoT que permitem a comunicação desses aparelhos com a rede doméstica. Para Atzori, Iera e Morabito (2010), a IoT tem o potencial de adicionar uma nova dimensão aos aparelhos, permitindo comunicação de qualquer mídia, a qualquer hora e em qualquer lugar.

2.5 DISPOSITIVOS INTELIGENTES

Os dispositivos inteligentes têm alcançado cada vez mais relevância no mercado de tecnologia da informação devido às suas inúmeras aplicabilidades e cenários de uso, onde se destacam principalmente, nas construções prediais e domésticas.

Os dispositivos inteligentes fazem parte do domínio da IoT. São aparelhos que podem (i) ser identificados, (ii) comunicar-se e (iii) interagir (MIORANDI, D. et al, 2012). Dessa forma, um dispositivo inteligente se conecta com a rede e/ou outros dispositivos para realizar diversas ações como troca de mensagens ou tomada de decisões (MORAIS, I. S. et al, 2018).

Para Bolzani (2004), qualquer dispositivo eletro-eletrônico que desenvolva uma tarefa básica, efetue troca de informações e possibilite comando remoto, pode ser considerado um dispositivo inteligente. É visto que muitos desses dispositivos possuem capacidade de adaptação e podem ser programados para automatizar tarefas. Em sistemas avançados todos esses sistemas podem ser interligados proporcionando integração, intercomunicação e conectividade (STEVAN JUNIOR, 2018).

3 COMPONENTES ELETRÔNICOS UTILIZADOS

3.1 ESP32

O ESP32 é um *chip* criado e produzido pela Espressif Systems¹, que também é responsável pela fabricação do ESP8266. O ESP8266 é considerado a primeira linha de *chip* de baixo custo criado especificamente para atender aplicações IoT. Isso se deve ao fato de ter sistema Wi-Fi embutido, evitando a necessidade de instalação de bibliotecas externas para conectar o *chip* com a rede de internet. De acordo com Pedro Minatel², *Developer Advocate* da Espressif, com o lançamento do ESP8266, a empresa conseguiu reduzir o preço de um *chip* contendo recurso Wi-Fi, no mercado consumidor, de US\$20 para US\$2.

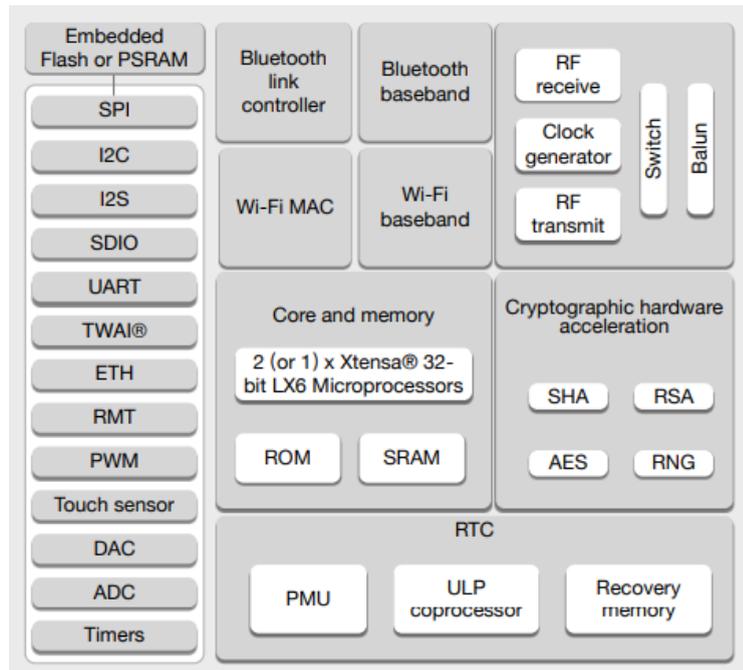
O ESP32 é a linha sucessória do ESP8266, lançado em 2016. Além de trazer o módulo Wi-Fi, também contempla um sistema para conexão *bluetooth*, sendo ainda mais vantajoso sua utilização. Ademais, a família ESP foi desenvolvida para ter baixo consumo de energia e capacidade de atuar em situações adversas, suportando uma variação de temperatura de -40 a 125°C (Espressif, 2022).

O SoC (System on Chip) do ESP32 reúne diversos elementos (FIGURA 3.1). Ele é equipado com processador tendo uma memória estática (SRAM) e uma memória não volátil (ROM), sistema para Wi-Fi e radiofrequência (RF) e memória embutida, além de possuir pinagem para memória externa. O *chip* possui também aceleradores de *hardware*, um novo recurso que também é trazido como novidade em relação ao ESP8266. De acordo com a fabricante, os aceleradores de *hardware* além de melhorar o desempenho do *chip*, também realizam a criptografia do código da memória *flash*, uma camada de segurança para evitar invasões feitas por terceiros (Espressif, 2022).

¹ Disponível em: <https://www.espressif.com/>

² Disponível em: <https://www.youtube.com/watch?v=yvbeDSDY73Y&list=WL&index=49&t=51s>

Figura 3.1 - ESP32 (SoC)



Fonte: Espressif (2022).

Além do SoC, o ESP32 também pode ser comercializado integrado a um módulo ou através de um *Dev Kit* (FIGURA 3.2).

Figura 3.2 - Dispositivos ESP32



Fonte: Adaptada de Mouser (2022).

O módulo reúne o *chip* do ESP32 com capacitores, resistores, antenas RF e outros periféricos. Já o microcontrolador ou *Dev Kit*, incorpora todos esses elementos a uma placa de

hardware, facilitando o manuseio e conexão via *jumpers*. Os microcontroladores são indicados para desenvolvedores que querem usufruir dos recursos do ESP32 mas que desejam focar na aplicação sem se preocupar com o baixo nível de implementação do *chip*.

3.2 DHT22

O DHT22 (ou AM2302), é um sensor de temperatura e umidade muito utilizado em projetos de microeletrônica devido ao seu baixo custo. Ele consegue medir temperaturas em uma faixa de -40° a $+80^{\circ}\text{C}$ e umidade entre 0 a 100%. O Quadro 3.1, exibe algumas das especificações do sensor:

Quadro 3.1 - Especificações Técnicas do DHT22

Modelo	DHT22
Faixa de Operação	Umidade: 0-100%RH; Temperatura: $-40\sim 80^{\circ}\text{C}$
Precisão	Umidade: $\pm 2\%$ RH (Max $\pm 5\%$ RH); Temperatura: $\pm 0.5^{\circ}\text{C}$
Voltagem	5V

Fonte: Adaptado de AllDatasheet (2022).

Além do preço acessível, o DHT22 é fácil de ser utilizado em projetos eletrônicos devido a simplicidade de sua pinagem. Ele possui apenas três pinos funcionais sendo um de alimentação (VCC), um pino terra ou GND, um pino para saída de dados.

4 HOME ASSISTANT

4.1 VISÃO GERAL

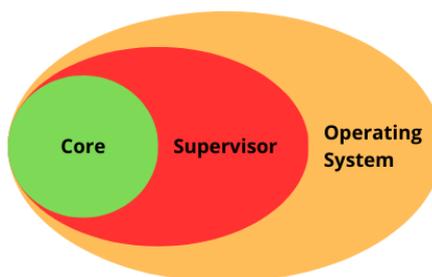
O *Home Assistant* (HASS), foi idealizado para fornecer um ambiente completo de integração de dispositivos para pessoas que têm interesse em automatizar suas casas. É uma plataforma *open source*, mantida por uma comunidade referida como "*makers*", responsáveis por desenvolver funcionalidades, disponibilizar *add-ons* e manter a documentação do aplicativo atualizada. Essa comunidade abrange profissionais e entusiastas que contribuem diariamente com soluções de problemas e instruções relacionadas ao uso da plataforma, criando uma vasta rede de compartilhamento de conhecimento.

Até o presente momento, o *Home Assistant* possui integração com mais de mil marcas de dispositivos inteligentes e suporta 2285 diferentes *add-ons* incluindo dispositivos de alarmes, câmeras, energia, luz, sensores, entre outros. O *Home Assistant* não é apenas um aplicativo, mas um sistema embarcado de controle e automação residencial onde é possível integrar, configurar e atualizar dispositivos por meio de uma *interface* fácil de usar (HOME ASSISTANT, 2022).

4.2 ARQUITETURA

O *Home Assistant* pode ser dividido em partes que compõem uma pilha (FIGURA 4.1). O *core* é a versão mais enxuta do HASS, o essencial para se utilizar a plataforma, pois é onde estão inseridas as funções necessárias para o funcionamento das automações. O *supervisor* é uma camada extra que acrescenta novas funcionalidades ao *core*. Além de cuidar dos *add-ons*, ele é responsável por realizar as tarefas de gerenciamento do aplicativo, sendo elas: a execução e atualização do *Home Assistant*, fornecimento de sistema de áudio unificado, realização e restauração de *backups*. Já a camada *Operating System* é a versão mais completa do *software*, que é gerenciada pelo supervisor e roda em um *container* Docker (SILVA, 2019). Possui uma *interface* gráfica navegável e intuitiva para uso, sendo a versão mais indicada para um maior aproveitamento dos recursos oferecidos pelo HASS.

Figura 4.1 - Pilha do Home Assistant

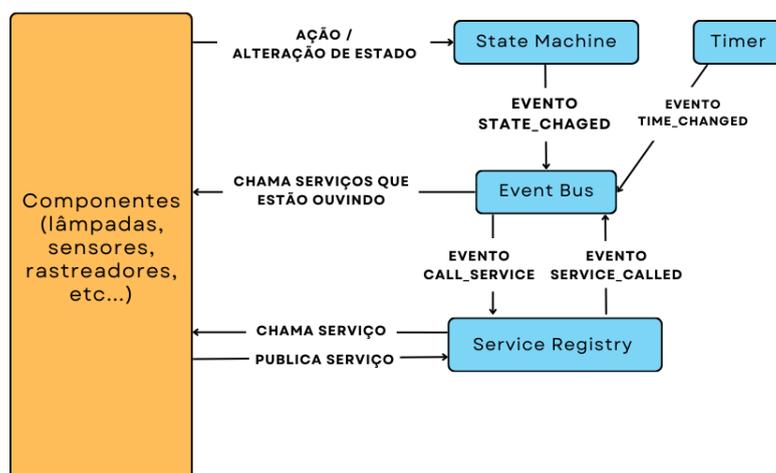


Fonte: Adaptada do Home Assistant (2023).

Tendo isso em vista, o *core* é onde ocorre o processamento de informação de uma automação feita no HASS. O fluxo deste processamento é baseado em eventos e o *core* conta com quatro elementos para conseguir gerenciá-los: *Event Bus*, *State Machine*, *Service Registry* e *Timer*. Ao *Event Bus* fica incumbido as tarefas de gerenciar os eventos, encaminhando-os quando ocorre uma chamada de serviço ou mudança de estado de algum componente. O *State Machine*, responsável pelo controle dos estados, aciona o *Event Bus* para que ele atue. O *Service Registry* mantém um histórico de chamadas de serviços e pode ser usado também para chamar serviços. O *Timer* atualiza o *Event Bus* a cada segundo.

Um componente eletrônico conectado ao *Home Assistant*, ao sofrer uma alteração em seu estado (sensores) ou após desempenhar uma ação (atuadores), poderá disparar um gatilho. Os gatilhos são configurados no HASS através de arquivos de configuração de seus respectivos dispositivos. Após o acionamento de um gatilho, o comportamento da automação seguirá o fluxo apresentado abaixo na Figura 4.2.

Figura 4.2 - Funcionamento do Home Assistant core

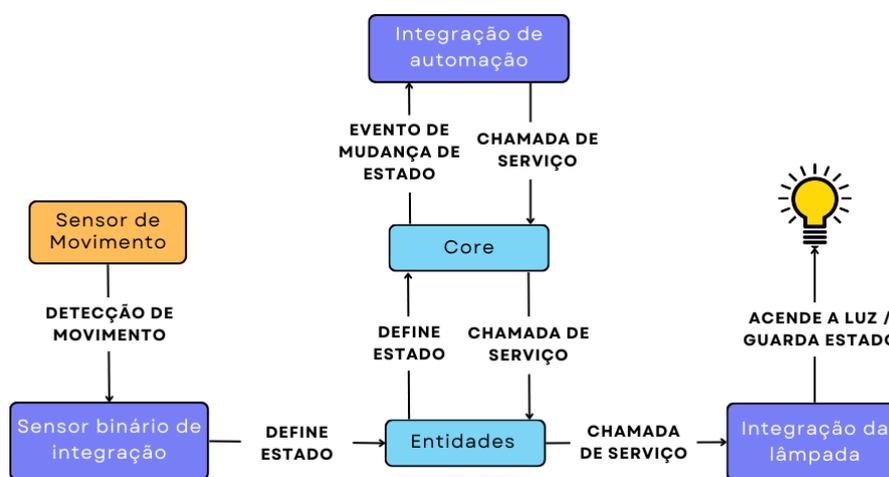


Fonte: Adaptada do Home Assistant (2023).

Através da arquitetura, o *Home Assistant* consegue padronizar o fluxo de informação para qualquer automação, isolando o funcionamento interno do aplicativo. Isso é possível juntamente com a utilização de outro recurso da plataforma: as entidades. As entidades são definições de domínios que implementarão serviços e eventos específicos.

Na Figura 4.3 é mostrado visualmente como acontece o fluxo de informação de integração de uma lâmpada, utilizando a entidade `light`, que será acendida após a detecção de um sensor de movimento.

Figura 4.3 - Fluxo de informações de uma automação



Fonte: Adaptada do Home Assistant (2023).

Neste exemplo (FIGURA 4.3), consideramos que está sendo utilizada a entidade `light` para definir os comportamentos da lâmpada. O que é feito é, a classe `light` é estendida e então temos à disposição seus atributos e métodos para implementar, abstraindo o funcionamento interno do serviço.

Essa arquitetura foi projetada visto que muitos componentes (lâmpada, temperatura, câmera, etc) apesar de serem usados em sistemas customizáveis, possuem atributos em comum. Com as entidades, o desenvolvedor consegue focar no sistema que deseja construir e a automação se torna muito mais simples e rápida.

4.3 MÉTODOS DE INSTALAÇÃO

Como mencionado na seção anterior, o *Home Assistant* é dividido em partições. Essas partes da pilha são versões diferentes do aplicativo, de forma que é possível instalá-las separadamente, alinhando as necessidades dos usuários de acordo com os recursos disponíveis em cada nível. Na Figura 4.4 são exibidos os métodos de instalação e os recursos que cada método abrange.

Figura 4.4 - Métodos de instalação do *Home Assistant*

	OS	Container	Core	Supervised
Automations	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Backups	✓	✓ ¹	✓ ¹	✓
Managed OS	✓	✗	✗	✗

Fonte: *Home Assistant* (2022).

Em relação aos recursos apresentados na Figura 4.4, o *Managed OS* é exclusivo do OS (*Operating System*). O *Managed OS* é um recurso implementado para garantir um maior nível de controle do sistema pelo usuário, dando a ele opções de configuração personalizadas para gerenciar e atualizar o *Home Assistant* de acordo com suas preferências, todavia, a instalação do OS é feita apenas em sistemas operacionais Linux ou em SBCs (Single-board Computers) e indicada para máquinas dedicadas exclusivamente para executar a aplicação, podendo ser feita em máquinas virtuais. Os *add-ons*, também são recursos notáveis do HASS. São criados pela comunidade e disponibilizados para *download* dentro da plataforma. Algumas de suas atuações são configurar os dispositivos eletrônicos de forma automática, fazer instalações e permitir a comunicação entre diferentes módulos.

Dentre os métodos de instalação, como já mencionado no capítulo anterior, o *core* representa a opção mais limitada do HASS que contempla apenas os recursos mínimos para o funcionamento do aplicativo. O método de instalação Container é a instalação do *core* através de um *container* Docker (FIGURA 4.4).

Basicamente, o *Home Assistant* pode ser instalado em qualquer computador Linux, computadores de arquitetura x64 que suportam virtualização ou em computadores de placa única, como por exemplo, o Raspberry Pi. Porém, cada tipo de máquina aceita determinados métodos de instalação. O Quadro 4.1 relaciona os sistemas operacionais e os métodos de instalação suportados.

Quadro 4.1 - Métodos de instalação suportados por diferentes computadores.

	Computadores de placa única	Windows	macOS	Linux
OS	✓	✓*	✓*	✓
Container	✓	×	×	✓
Supervised	×	×	×	✓
Core	✓	✓	✓	✓

Fonte: Adaptado do Home Assistant (2022).

Notas: O método OS pode ser realizado em máquinas Windows e macOS desde que se utilize uma máquina virtual Linux para isto.

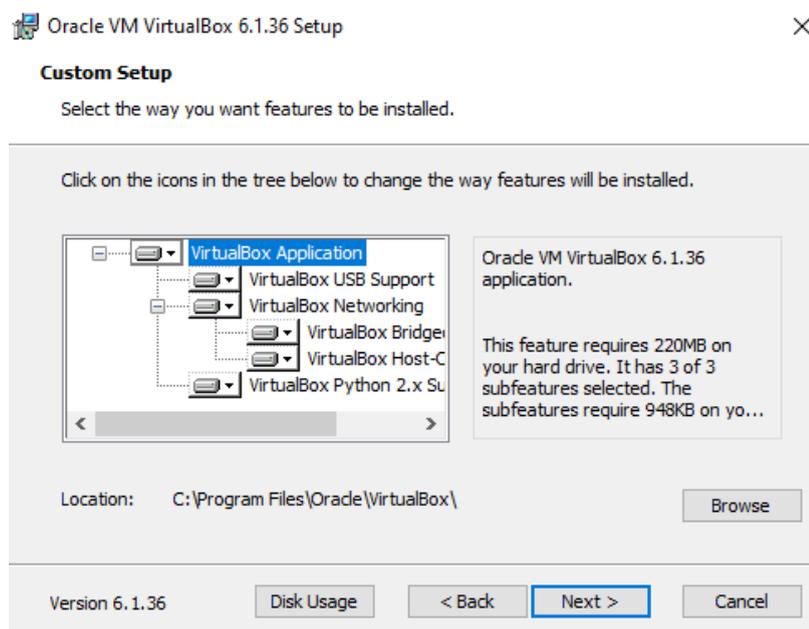
5 INSTALAÇÃO

Para este trabalho será utilizado o método de instalação OS, pois é com ele que têm-se acesso a todos os recursos disponíveis no HASS. Este método pode ser realizado em qualquer sistema operacional Linux, considerando a recomendação que seja feito em uma máquina exclusiva para a operação do HASS. Assim sendo, será utilizada uma máquina virtual específica para tal.

5.1 VIRTUALBOX

Será utilizada a versão 7.0 da VirtualBox para Windows, disponibilizada para *download* no site da mesma³. Após o término do *download*, o arquivo deverá ser executado para que se inicie o processo de instalação. No primeiro passo (FIGURA 5.1), instalar a opção “*VirtualBox Application*” que engloba os recursos necessários para a utilização da máquina virtual.

Figura 5.1 - Instalação da máquina virtual



Fonte: Autora (2023).

As configurações da etapa seguinte estão relacionadas às preferências de acesso à VirtualBox e podem ser definidas de acordo com o gosto pessoal do usuário, não tendo

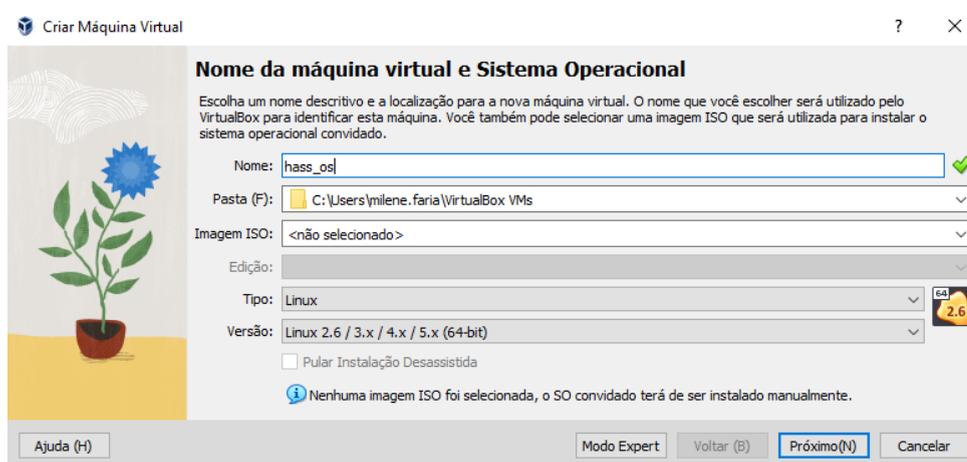
³ Disponível em: <https://www.virtualbox.org/wiki/Downloads>

impacto no funcionamento do aplicativo. Assim, basta avançar e aguardar a conclusão da instalação.

5.2 HOME ASSISTANT OS

Para iniciar a instalação do OS em uma máquina virtual, primeiro é necessário fazer o *download* da imagem vdi (formato utilizado na VirtualBox), disponível no site do *Home Assistant*⁴ no submenu “Windows”. Feito isso, basta abrir o aplicativo da VirtualBox, criar uma nova máquina virtual em “Novo”, e então preencher os campos requeridos. Na caixa de seleção de tipo, é necessário selecionar a opção Linux e versão “Linux 2.6 / 3.x / 4.x (64-bit)”.

Figura 5.2 - Definição do nome e sistema operacional da máquina virtual



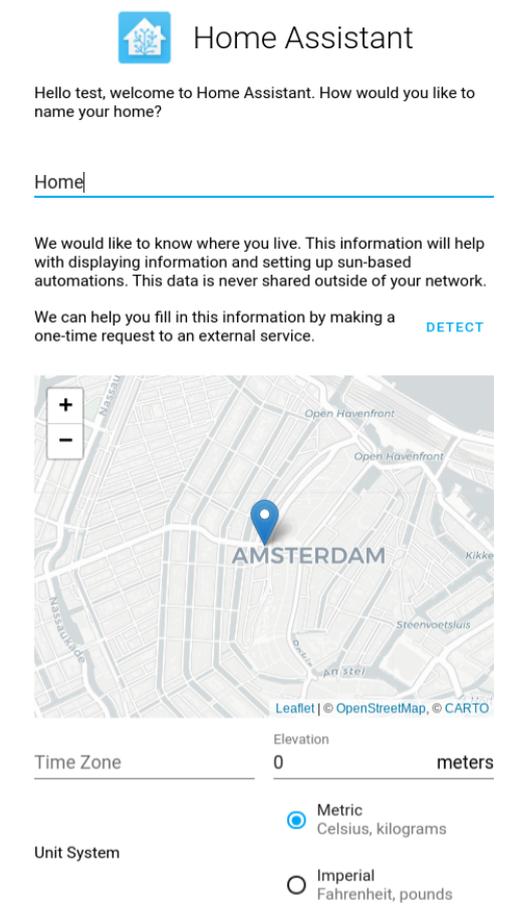
Fonte: Autora (2023).

Os próximos passos são referentes a alocação de memória e processamento que a máquina virtual poderá utilizar. É recomendado que sejam alocados, no mínimo, 2048 MB de memória principal e 2 CPUs. Em seguida, em “Disco Rígido Virtual” é dada a opção para “utilizar um disco rígido virtual existente”. Nesse passo, adiciona-se a imagem vdi baixada do site do *Home Assistant*. Na Figura 5.3 pode-se visualizar um pequeno sumário contendo as configurações feitas.

⁴ Disponível em: <https://www.home-assistant.io/>

Após acessar o endereço, é requisitado um cadastro de primeiro acesso de usuário. Após as definições de usuário e senha, é solicitada permissão de detecção da máquina pela rede, assim, o HASS consegue realizar a localização da residência, deixando essa informação salva para automações que necessitem desses dados posteriormente.

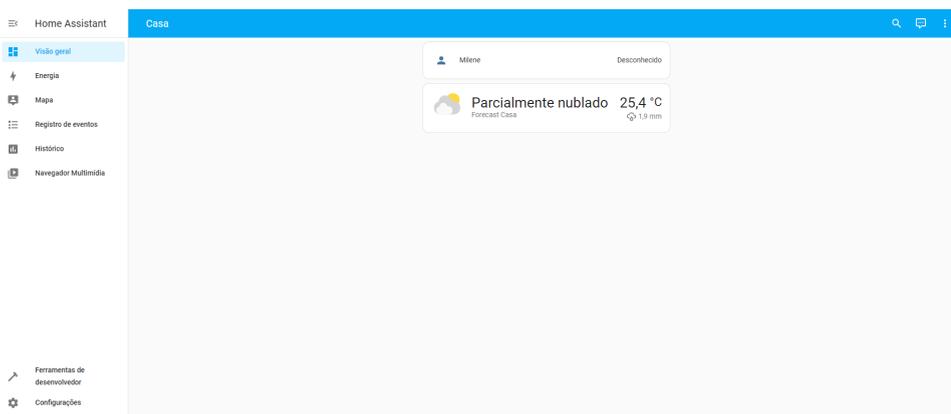
Figura 5.5 - Primeiro acesso ao Home Assistant



Fonte: Autora (2023).

A aplicação é aberta logo após, como pode ser visto na Figura 5.6. O menu “Visão Geral” (FIGURA 5.6) mostra todos os dispositivos conectados no *Home Assistant*. Os menus mais utilizados neste trabalho são os menus “Ferramentas de desenvolvedor” e “Configurações”. Em “Ferramentas de desenvolvedor” temos acesso a todos os arquivos de configuração de dispositivos e serviços. Já em “Configurações” podemos visualizar atualizações pendentes, adicionar novos dispositivos, automações e acessar o catálogo de *add-ons*.

Figura 5.6 - Visão Geral do Home Assistant



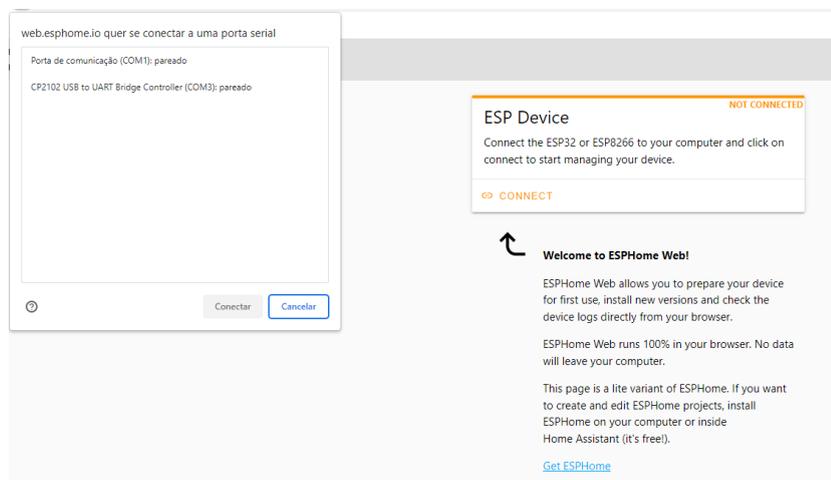
Fonte: Autora (2023).

5.3 ESPHOME

O ESPHome é uma aplicação *open source* desenvolvida para facilitar a integração de dispositivos ESP ao HASS. Ele será utilizado para configurar e instalar o *firmware* no ESP32 de forma automática.

O *download* do ESPHome é feito acessando a página de *add-ons* do HASS no menu “Ferramentas do Desenvolvedor”. Nesta página, basta procurar pelo ESPHome e instalá-lo. Após a instalação, ele pergunta se deseja-se acessar a *interface web* para que seja redirecionado para a tela inicial do ESPHome. Nesta tela, clicar em “adicionar um dispositivo”. Para dar prosseguimento nessa etapa, é necessário que o ESP32 já esteja conectado ao computador via cabo USB e reconhecido pelo sistema operacional do mesmo, pois o ESPHome fará uma busca pelo dispositivo através das portas seriais da máquina.

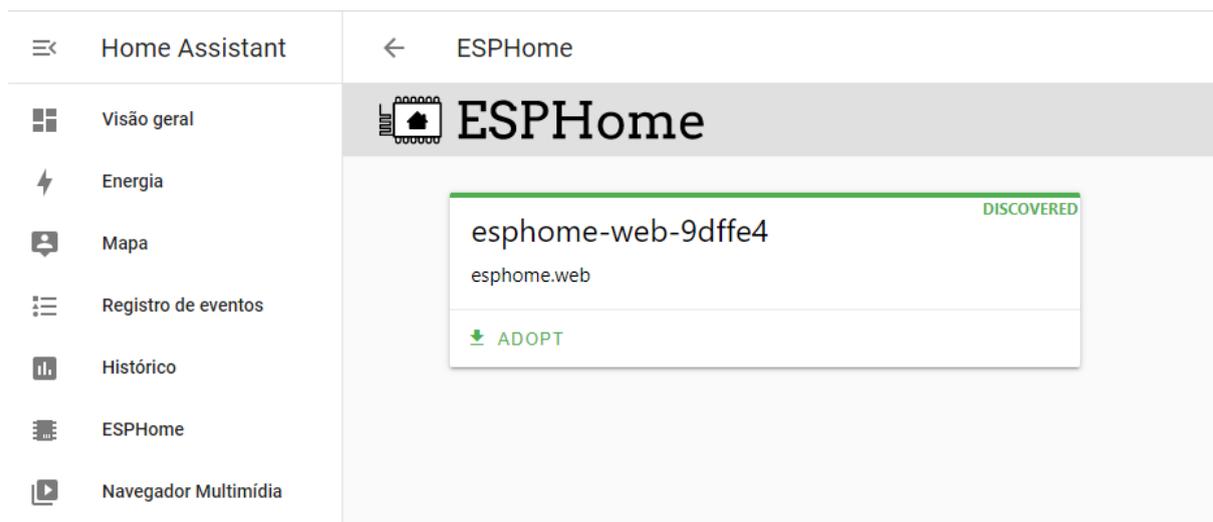
Figura 5.7 - Reconhecimento do dispositivo pelo ESPHome



Fonte: Autora (2023).

Após confirmar o ESP32 e sua porta, é necessário clicar em “preparar para primeiro uso” e depois em “instalar”. O ESPHome pedirá as credenciais para acessar o Wi-Fi, fará o *download* do *firmware* específico para o microcontrolador para dar início a instalação deste *firmware* no dispositivo. Concluído esse processo, o dispositivo passa para o status “*discovered*” (FIGURA 5.8).

Figura 5.8 - Dispositivo reconhecido pelo Home Assistant



Fonte: Autora (2023).

Após isso, vamos para a configuração clicando na opção “*adopt*” que irá gerar o arquivo de configuração para o microcontrolador. Criado o arquivo, basta acessá-lo e clicar em instalar. Como o ESP32 possui Wi-Fi e neste ponto já está com seu firmware instalado, basta escolher a opção “Wirelessly” e aguardar a conclusão da instalação.

Concluído o processo de *adopt*, o arquivo de configuração gerado para o dispositivo indicam algumas propriedades relevantes (FIGURA 5.9). Primeiro é apresentado as informações do dispositivo. O nome será alterado para `sensor-temperatura` para melhor identificação do sistema. A variável `logger` é utilizada para habilitar o registro detalhado de *log* em processamentos realizados no ESP32, podendo ser bastante útil para analisar erros inesperados. A `api` contém um chave criptografada que é gerada durante o processo *adopt*. A variável `ap` cria uma rede Wi-Fi caso o dispositivo não consiga se conectar à rede configurada, e a variável `captive_portal` é usada para habilitar a *interface* do `ap`.

Figura 5.9 - Configuração do sensor

```
× esphome-web-9dffe4.yaml
1 esphome:
2   name: sensor-temperatura
3
4 esp32:
5   board: esp32dev
6   framework:
7     type: arduino
8
9 # Enable logging
10 logger:
11
12 # Enable Home Assistant API
13 api:
14   encryption:
15     key: "mGJ9AyCk+EJ9zKCJp0NpJyLsNadZJ5zh1dTovN/7ivY="
16
17 ota:
18
19
20 wifi:
21   ssid: !secret wifi_ssid
22   password: !secret wifi_password
23
24 # Enable fallback hotspot (captive portal) in case wifi connection fails
25 ap:
26   ssid: "Esphome-Web-9Dffe4"
27   password: "pH39o7mopAuS"
28
29 captive_portal:
30
```

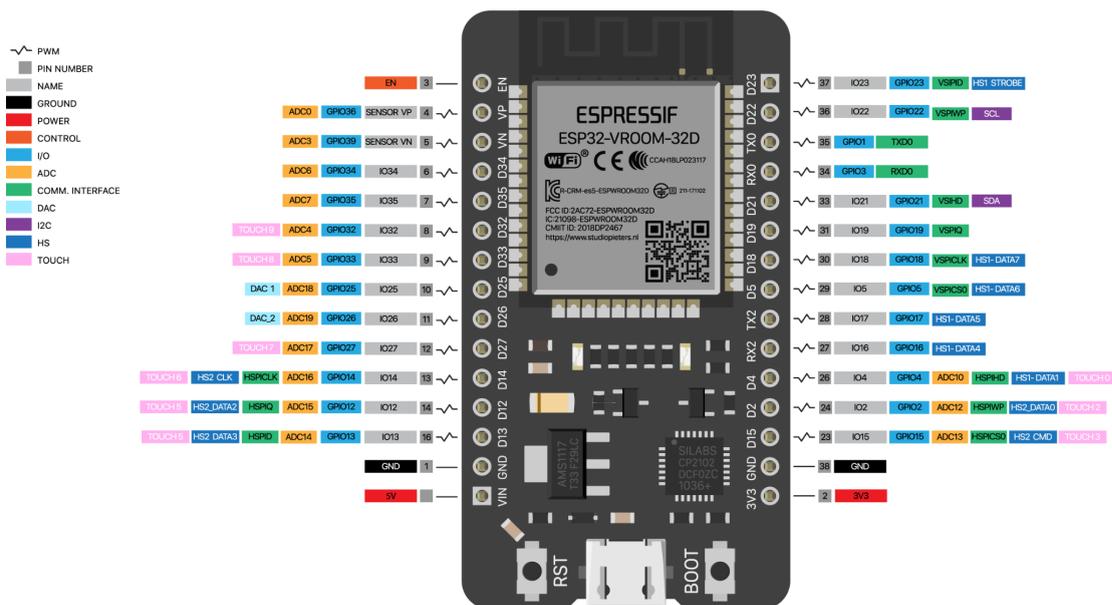
Fonte: Autora (2023).

6 CONSTRUÇÃO DO PROTÓTIPO

Essa seção dá início ao processo de construção do protótipo que utilizará um microcontrolador para ler dados enviados de um sensor de temperatura.

Antes de realizar as conexões entre o microcontrolador e o sensor, é importante consultar as pinagens do ESP32 (FIGURA 6.1) e DHT22 (FIGURA 6.2).

Figura 6.1 - ESP32 (Pinagem)



Fonte: Studio Pieters (2023).

Figura 6.2 - DHT22 (Pinagem)

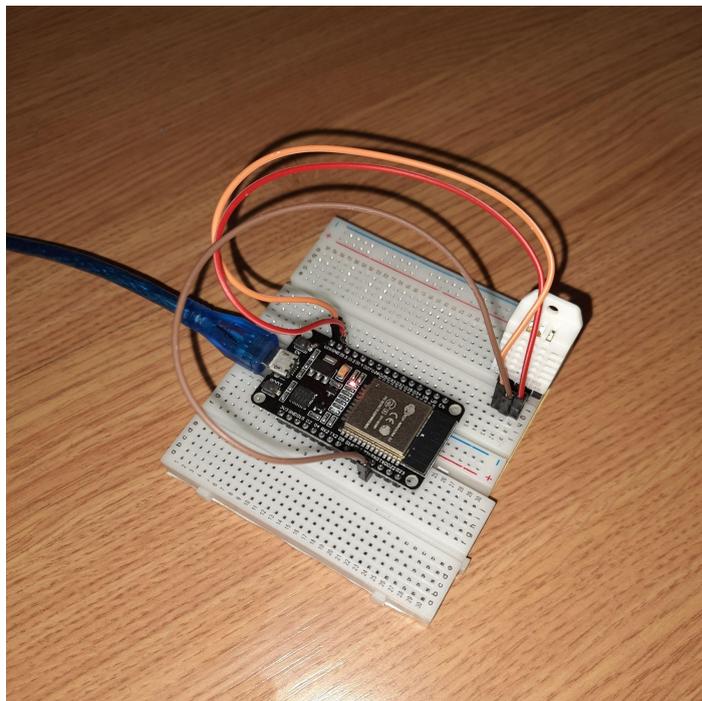


Fonte: Filipeflop (2022).

Considerando que a voltagem do DHT22 é de 5V, deverá ser conectado ao pino VCC do sensor o pino que fornece essa voltagem no ESP32, neste caso, o pino VIN. Também deverá ser conectado os pinos GNDs entre si. O pino referido como “out” no sensor será

conectado ao RX0 no ESP32 por se tratar de um pino que faz recepção de dados no microcontrolador. A conectividade entre eles pode ser feita diretamente pelos pinos com *jumpers* fêmea-fêmea ou através de uma *protoboard* com *jumpers* macho-macho.

Figura 6.3 - Protótipo do sensor



Fonte: Autora (2023).

Após isso, o sensor deve ser configurado em seu respectivo arquivo (ESPHOME, 2023) como mostra o Código 1.

```
1. sensor:  
2.   - platform: dht  
3.   pin: GPIO03  
4.   temperature:  
5.     name: "Temperatura"  
6.   humidity:  
7.     name: "Umidade"  
8.   update_interval: 60s
```

Código 1 - Configuração do sensor.
Fonte: Autora (2023).

Dentre o apresentado:

- `platform`: refere-se a marca do dispositivo
- `pin`: o pino que o microcontrolador recebe os dados. Nesse caso, utilizando a Figura 6.1 como referência, o RX0 da placa onde foi conectado à saída de dados do DHT22 é reconhecido como GPIO03
- `temperature`: para indicar as informações de temperatura do sensor
- `humidity`: para indicar as informações de umidade do sensor
- `name`: obrigatório para nomear as variáveis do sensor
- `update_interval`: variável opcional para determinar o intervalo de atualização de leitura do sensor

Após isso, basta salvar o arquivo para que a instalação seja feita. Após concluído, será exibido no *log* contendo a medição da temperatura e umidade do ambiente pelo sensor de acordo com o intervalo definido, conforme a Figura 6.4.

Figura 6.4 - Detecção de Temperatura e Umidade pelo sensor

```
[15:58:16][D][dht:048]: Got Temperature=28.5°C Humidity=57.7%
[15:58:16][D][sensor:127]: 'Temperatura': Sending state 28.50000 °C with 1 decimals of accuracy
[15:58:16][D][sensor:127]: 'Umidade': Sending state 57.70000 % with 0 decimals of accuracy
[15:59:16][D][dht:048]: Got Temperature=28.5°C Humidity=57.9%
[15:59:16][D][sensor:127]: 'Temperatura': Sending state 28.50000 °C with 1 decimals of accuracy
[15:59:16][D][sensor:127]: 'Umidade': Sending state 57.90000 % with 0 decimals of accuracy
[16:00:16][D][dht:048]: Got Temperature=28.6°C Humidity=57.8%
[16:00:16][D][sensor:127]: 'Temperatura': Sending state 28.60000 °C with 1 decimals of accuracy
[16:00:16][D][sensor:127]: 'Umidade': Sending state 57.80000 % with 0 decimals of accuracy
```

Fonte: Autora (2023).

7 AUTOMAÇÃO

Tendo o protótipo configurado no *Home Assistant*, neste capítulo será apresentado o processo de criação da automação que através da configuração do serviço de notificação do HASS, enviará um alerta de acordo com dados lidos do sensor.

7.1 ADICIONANDO AS ENTIDADES DO SENSOR

Para adicionar uma entidade basta ir em configurações, dispositivos e serviços. O sensor deverá ser exibido na lista de dispositivos reconhecidos pelo HASS, sendo necessário apenas adicioná-lo.

Figura 7.1 - Dispositivos, serviços e suas entidades



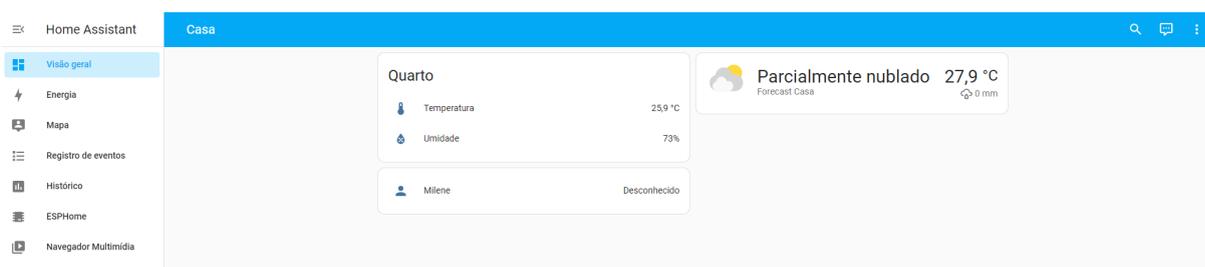
Fonte: Autora (2023).

Figura 7.2 - Entidades do sensor

Procurar entidades		Filtrando por integração "ESPHome: sensor-temperatura" LIMPAR				
<input type="checkbox"/>	↑ Nome	ID da entidade	Integração	Área	Desativado por	Esta...
<input type="checkbox"/>	sensor-temperatura Firmware	update_sensor_temperatura_firmware	ESPHome	Quarto	-	-
<input type="checkbox"/>	Temperatura	sensor.temperatura	ESPHome	Quarto	-	-
<input type="checkbox"/>	Umidade	sensor.umidade	ESPHome	Quarto	-	-

Fonte: Autora (2023).

Ao adicionar um dispositivo é dada a opção de nomear o seu ambiente para melhor gerenciamento dos sistemas de automação. Neste caso, será nomeado como “Quarto”. Para verificar se as entidades estão habilitadas, basta voltar na página inicial ou “Visão Geral”. Todas as entidades reconhecidas e configuradas no *Home Assistant* são exibidas na tela.

Figura 7.3 - Dashboard com exibição das entidades

Fonte: Autora (2023).

7.2 ADICIONANDO O SERVIÇO

Será utilizado o serviço SMTP do *Home Assistant* para enviar notificações via *e-mail*. Para facilitar o processo de configuração deste serviço, instalaremos o *add-on* File Editor, que pode ser utilizado para editar o arquivo de configuração do HASS que é onde é adicionado a configuração do SMTP.

Após a instalação do File Editor, basta adicionar as configurações do serviço de notificação que são disponibilizadas pela documentação do HASS (HOME ASSISTANT, 2023). O arquivo de configuração ficará da seguinte forma:

```

1. # Loads default set of integrations. Do not remove.
2. default_config:
3.
4. # Load frontend themes from the themes folder
5. frontend:
6.   themes: !include_dir_merge_named themes
7.
8. # Text to speech
9. tts:
10.   - platform: google_translate
11.
12.   automation: !include automations.yaml
13.   script: !include scripts.yaml
14.   scene: !include scenes.yaml
15.
16. # Example configuration.yaml entry for Google
    Mail.
17.   notify:

```

```
18.     - name: "Temperatura"
19.     platform: smtp
20.     server: "smtp.gmail.com"
21.     port: 587
22.     timeout: 15
23.     sender: "email@gmail.com"
24.     encryption: starttls
25.     username: username
26.     password: senha criptografada
27.     recipient:
28.         - "exemplo_1@gmail.com"
29.         - "exemplo_2@hotmail.com"
30.     sender_name: "Home Assistant"
```

Código 2 - Configuração do serviço SMTP.
Fonte: Autora (2023).

Dentre as variáveis do serviço de notificação, temos:

- name: variável opcional usada para identificar os serviços de notificação
- sender: endereço de *e-mail* que enviará a notificação
- password: senha criptografada pelo Gmail
- recipient: lista de endereços de *e-mail* que receberão a notificação
- sender_name: o nome que referencia quem enviou a notificação
- username: nome de usuário para o serviço
- password: senha criptografada disponibilizada pelo Gmail

As demais variáveis serão mantidas com seu valor padrão de acordo com o definido originalmente pela documentação. Após essa configuração é necessário reiniciar o *Home Assistant* que pode ser feito acessando o menu “Ferramentas de desenvolvedor”.

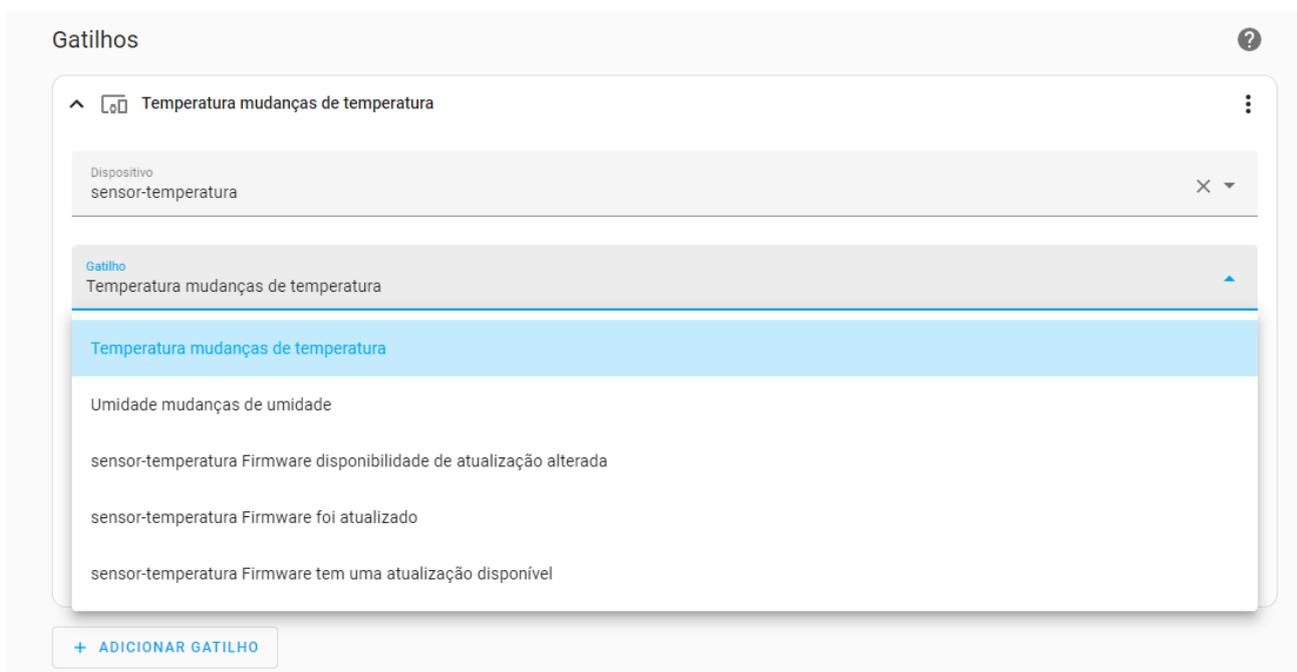
7.3 CRIANDO A AUTOMAÇÃO

Depois de termos as entidades e configurado o serviço, podemos prosseguir com a automação. Para isto, no menu “Configurações”, acessar “Automações & Cenas”, e adicionar uma automação.

A criação de um serviço de automação pode ser feito inteiramente através da *interface* gráfica do *Home Assistant*, onde é necessário selecionar um gatilho ou condição e definir um

serviço a ser executado. Neste caso, o gatilho será disparado de acordo com a leitura do sensor. Como o sensor se trata de um DHT22 que possui as entidades temperatura e umidade, ao selecionar o sensor, o HASS já sugere os respectivos gatilhos como pode-se observar na Figura 7.4.

Figura 7.4 - Gatilhos disponibilizados para o dispositivo



Fonte: Autora (2023).

Neste caso, o gatilho escolhido irá operar de acordo com a mudança de temperatura. Então, configura-se os valores que serão usados de parâmetro acima ou abaixo do captado pelo sensor. Em “ações”, seleciona-se o serviço configurado no capítulo anterior. Nele, podemos definir um título e o corpo da mensagem que será enviado para o *e-mail*. E então é apresentado os campos que preencherão o título e corpo do *e-mail* que será enviado como notificação (FIGURA 7.5). Na Figura 7.6 é mostrado o resultado da notificação enviada.

Figura 7.5 - Configuração do serviço

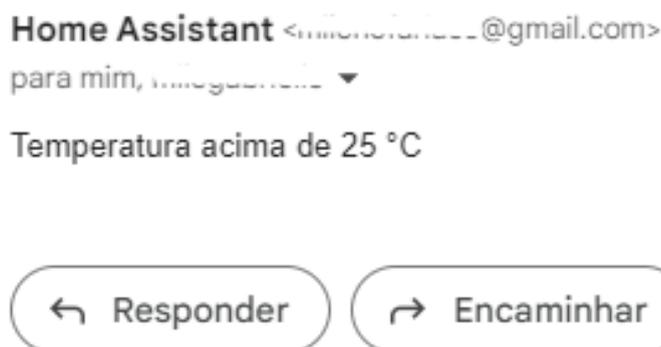
The screenshot shows the configuration page for a service named "Notificações: Send a notification with temperatura". The interface is titled "Ações" and includes a search bar and a list of services. The selected service is expanded, showing its description: "Sends a notification message using the temperatura service." Below this, there are several configuration fields:

- Message:** "Message body of the notification." with the value "Temperatura acima de 25 °C".
- Title:** "Title for your notification." with the value "Notificação do Home Assistant".
- Target:** "An array of targets to send the notification to. Optional depending on the platform." with a value of "1".
- Data:** "Extended information for notification. Optional depending on the platform." with a value of "1".

At the bottom of the configuration panel, there is a button labeled "+ ADICIONAR AÇÃO".

Fonte: Autora (2023).

Figura 7.6 - Notificação enviada pela aplicação



Fonte: Autora (2023).

7.4 CONFIGURAÇÃO DE OUTROS SENSORES NO HOME ASSISTANT

Através da utilização do *add-on* ESPHome, o HASS mantém o mesmo padrão de configuração para qualquer microcontrolador pertencente a família ESP enquanto que, a associação de outros módulos (sensores ou atuadores) são feitos baseando sempre na pinagem do microcontrolador utilizado e do módulo, de forma a estabelecer a comunicação entre eles.

Geralmente, o microcontrolador deverá fornecer alimentação para os módulos já que muitos sensores e atuadores não possuem outra alternativa de fonte de energia, esta também é uma das vantagens relacionadas à utilização de microcontroladores em sistemas com componentes eletrônicos. O fluxo de informação entre módulo e microcontrolador é particular de cada sistema, pois além de depender da comunicação entre os pinos, deve-se considerar também o objetivo que a aplicação deseja alcançar. Dessa forma, o desenvolvedor possui a liberdade de criar sua solução personalizada utilizando os módulos que desejar e adicionando as respectivas entidades no arquivo de configuração do dispositivo, como o demonstrado. As automações do HASS também seguem o padrão de utilização de um dispositivo responsável pelo acionamento de um gatilho e um serviço configurado na plataforma que atuará de acordo com o acionamento do gatilho.

Todos os tipos de dispositivos e as informações relacionadas às suas entidades podem ser encontradas na documentação do HASS, mas incluem módulos de diversos tipos e diversas marcas, como, câmeras, lâmpadas, alarmes, impressoras, assistentes virtuais, entre outros.

8 CONCLUSÃO E TRABALHOS FUTUROS

O *Home Assistant* cumpre o seu papel de fornecer um ambiente completo de automação de sistemas inteligentes dentro do contexto doméstico. A utilização da plataforma como ferramenta para a automação residencial conecta recursos IoT a usuários com diferentes níveis de conhecimento, por trazer diversas funcionalidades para facilitar a programação dos dispositivos sem exigir um conhecimento profundo em tecnologia. Isso é atingido através de uma arquitetura orientada a objetos que isola o funcionamento interno da automação, dando ao desenvolvedor mais tempo hábil para focar na customização de seu sistema. A plataforma é *open source* e possui uma comunidade consolidada onde os usuários interagem através de fóruns de discussões e colaboram no enriquecimento da documentação do produto.

Destaca-se também a disponibilização de um grande número de *add-ons*, também desenvolvidos pela comunidade e que também têm o objetivo de facilitar a integração das automações. Ademais, a disponibilização de uma *interface* gráfica que possibilita uma navegação amigável onde é possível desfrutar dos serviços disponibilizados de maneira visual e não necessariamente através de linhas de código.

Uma outra vantagem de se utilizar o *Home Assistant* é devido a sua versatilidade. O *software* é totalmente gratuito, e, por ter uma arquitetura segmentada, ele é disponibilizado em diferentes métodos de instalação que abrangem todos os sistemas operacionais existentes no mercado, ou seja, além de estar disponível para qualquer máquina, o desenvolvedor também pode adaptar o que necessita para sua aplicação de acordo com os recursos disponibilizados em cada método.

Quando discutimos a acessibilidade a recursos tecnológicos, além de considerarmos *softwares*, também existe um destaque no mercado de microeletrônica. Como visto, a família de microcontroladores ESP foi concebida com o foco em IoT considerando também o custo de aquisição feito pelo consumidor. Para este trabalho, foi gasto um valor total de R\$102,70 o que inclui o ESP32 (R\$54,90), protoboard e *jumpers* (R\$16,90) e o sensor DHT22 (R\$30,90).

Ainda que se desconsidere o protótipo utilizado, o *Home Assistant* reconhece qualquer dispositivo que utilize a internet. Considerando o ambiente residencial, isso abrange assistentes virtuais, impressoras, eletrodomésticos e outros equipamentos. Portanto, as automações feitas na plataforma podem utilizar produtos já presentes na residência, eliminando ou reduzindo a necessidade de aquisição de aparelhos microeletrônicos para a construção de um sistema inteligente.

Assim, através do desenvolvimento do trabalho na construção de um manual de instalação, configuração e utilização da ferramenta, conclui-se que é possível fazer um sistema de automação acessível e de fácil implementação, utilizando o *Home Assistant* como um aliado na programação de sistemas inteligentes para a automação residencial.

Cabe observar ainda que, a partir desse trabalho, foi apresentada e aprovada uma proposta para descaracterização e reuso de TVBox, em um Workshop nacional organizado pela Receita Federal. Neste Workshop, as instituições públicas foram convidadas a propor soluções de uso sustentável e com alcance social desses aparelhos usados para capturar sinal pirata de TVs por assinatura, apreendidos pela Receita Federal. Baseado nesse trabalho, foi apresentada a solução denominada “Gerenciamento de ambiente laboratorial com *Home Assistant*”, cujo objetivo é descaracterizar as TVBox para usá-las como um sistema para gerenciamento e monitoramento de ambiente laboratorial, usando o *Home Assistant* como serviço principal.

REFERÊNCIAS

- MORAIS, I. S.; GONÇALVES, P. F.; LEDUR, C. L.; CÓRDOVA JUNIOR, R. S.; SARAIVA, M. O.; FRIGERO, S. R. **Introdução a big data e internet das coisas (IOT)**. 1.ed. Porto Alegre: SAGAH, 2018.
- ATZORI, L.; IERA, A.; MORABITO, G. **The Internet of Things: A survey**. Computer Networks, v. 54, n. 15, p. 2787-2805, 2010. DOI: <<https://doi.org/10.1016/j.comnet.2010.05.010>>. Acesso em: 4 de nov. de 2022.
- PINHEIRO, J. M. S. **Falando de Automação Predial**. 2004. Disponível em: <http://www.projetoederedes.com.br/artigos/artigo_falando_de_automacao_predial.php>. Acesso em: 22 de out. de 2022.
- LOGICALIS. **IoT Snapshot**. Disponível em: <<https://imagine.la.logicalis.com/iot-snapshot-2022>>. Acesso em: 31 de jul. de 2022.
- MURATORI, J. R.; DAL BÓ, P. H. **Automação residencial: histórico, definições e conceitos**. O Setor Elétrico, São Paulo, ed. 62, mar. 2011. Disponível em: <https://www.osestoreletrico.com.br/wp-content/uploads/2011/04/Ed62_fasc_automacao_capI.pdf>. Acesso em: 22 de out. de 2022.
- STEVAN JUNIOR, S. L.; FARINELLI, F. A. **Domótica: automação residencial e casas inteligentes com Arduino e ESP8266**. 1.ed. São Paulo: Érica, 2018.
- BOLZANI, C. A. M. **Residências inteligentes**. 1.ed. São Paulo: Editora Livraria da Física, 2004.
- ACCARDI, A.; DODONOV, E. **Automação Residencial: Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos**. T.I.S. – Tecnologias, Infraestrutura e Software, São Carlos, v.1, n.2, nov. 2012. Disponível em: <<http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/17829/material/ARTIG002.pdf>>. Acesso em: 5 de nov. de 2022.
- WENDLING, M. **Sensores**. Colégio Técnico Industrial de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2010. Disponível em: <<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>. Acesso em: 10 de fev. de 2023.
- TEZA, V. R. **Alguns Aspectos Sobre A Automação Residencial - Domótica**. Orientador: Prof. Dr. João Bosco da Mota Alves. 2002. 106f. Dissertação (Mestrado) – Ciência da Computação, Concentração Sistemas de Computação, Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/83015/212312.pdf>> . Acesso em: 18 de nov. de 2022.
- SANTOS, B. P.; SILVA, L. A. M.; CELES, C. S. F. S.; BORGES NETO, J. B.; PERES, B. S.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. A. F. **Livro de Minicursos SBRC 2016**. Salvador: Sociedade Brasileira de Computação (SBC),

2016. Disponível em: <<http://sbrc2016.ufba.br/downloads/anais/MinicursosSBRC2016.pdf>>. Acesso em: 20 de abr. de 2023.

MIORANDI, D.; SICARI, S.; DE PELLEGRINI, F.; CHLAMTAC, I. **Internet of things: Vision, applications and research challenges**. Ad Hoc Networks, v. 10, n. 7, p. 1497-1516, 2012. DOI: <<https://doi.org/10.1016/j.adhoc.2012.02.016>>. Acesso em: 4 de nov. de 2022.

Espressif. **ESP32 Overview**. Disponível em: <<https://www.espressif.com/en/products/socs/esp32>>. Acesso em: 22 de nov. de 2022.

Espressif. **ESP32 Datasheet**. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 22 de nov. de 2022.

Espressif. **ESP8266 Technical Reference Manual**. Disponível em: <https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=14> Acesso em: 22 de nov. de 2022.

Mouser. **ESP32**. Disponível em: <<https://br.mouser.com/c/?q=esp32>>. Acesso em: 08 de dez. de 2022.

ALLDATASHEET. **DHT22 Datasheet (PDF) - List of Unclassified Manufacturers**. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>>. Acesso em: 22 de nov. de 2022.

Filipeflop. **Sensor de Umidade e Temperatura AM2302 DHT22**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/>>. Acesso em: 10 de abr. de 2022.

SILVA, L. J. N. **Utilizando o Home Assistant e o NodeMCU para um modelo genérico de automação moderna**. Orientador: Giovanny Fernando Lucero Palma. 2019. 47 f. TCC (Graduação) – Ciência da Computação, Centro de Ciências Exatas E Tecnologia, Departamento de Computação, Universidade Federal de Sergipe, São Cristóvão, 2019. Disponível em: <https://ri.ufs.br/bitstream/riufs/12305/2/Leonardo_Jose_Nascimento_Silva.pdf>. Acesso em: 12 de fev. de 2023.

DEVELOPERS, H. A. **Architecture Overview**. Disponível em: <https://developers.home-assistant.io/docs/en/architecture_index.html>. Acesso em: 10 de fev. de 2023.

ASSISTANT, H. **Installation**. Disponível em: <<https://www.home-assistant.io/installation/>>. Acesso em: 19 de dez. de 2022.

COMMUNITY, H. A. **HA OS vs Supervised**. Disponível em: <<https://community.home-assistant.io/t/ha-os-vs-supervised/395968>>. Acesso em: 10 de dez. de 2022.

ASSISTANT, H. **Home Assistant vs. Home Assistant Core**. Disponível em: <<https://www.home-assistant.io/faq/ha-vs-hassio/>>. Acesso em: 10 de fev. de 2023.

VirtualBox. **VirtualBox**. Disponível em: <<https://www.virtualbox.org/>>. Acesso em: 5 de ago. de 2022.

ESPHOME. **ESPHome**. Disponível em: <<https://esphome.io/>>. Acesso em: 10 de fev. de 2023.

STUDIO PIETERS. **ESP32 – PinOut**. Disponível em: <<https://www.studiopieters.nl/esp32-pinout/>>. Acesso em: 10 de fev. de 2023.

ASSISTANT, H. **SMTP**. Disponível em: <<https://www.home-assistant.io/integrations/smtp/>>. Acesso em: 23 de mar. de 2023.