

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
ESCOLA POLITÉCNICA E DE ARTES  
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



Automatização de um ventilador com Alexa

Goiânia  
2023

JOÃO VICTOR BANCZEK RODRIGUES

Automatização de um ventilador com Alexa

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Marcelo Antônio Adad de Araújo, M.E. E.

Goiânia

2023

JOÃO VICTOR BANCZEK RODRIGUES

Automatização de um ventilador com Alexa

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, em \_\_\_/\_\_\_/\_\_\_\_\_.

---

Prof. Ma. Ludmilla Reis Pinheiro dos Santos  
Coordenadora de Trabalho de Conclusão de  
Curso

Banca examinadora:

---

Orientador: Prof. Marcelo Antonio Adad de  
Araújo, M.E.E.

---

Prof. Carlos Alexandre Ferreira de Lima, M.E.E.

---

Prof. Nilson Cardoso Amaral, Dr.

Goiânia

2023

## **AGRADECIMENTOS**

A Deus primeiramente por me dar saúde e permitir estar presente no exato momento, à minha família que está do meu lado confiando em mim em orações e me incentivando.

Expresso minha gratidão ao meu orientador e professor, Marcelo Adad Antonio de Araújo, que me auxiliou desde o início até a conclusão deste projeto, dedicando-se especialmente aos sábados e, ocasionalmente, às terças-feiras.

Estendo meus agradecimentos aos meus colegas, Higor Ferreira, que contribuiu significativamente para o desenvolvimento do código, e Ismael Carlos, que foi imprescindível na montagem do circuito eletrônico.

"A vida só pode ser comprendida, olhando-se para trás; mas só pode ser vivida, olhando-se para frente."

Søren Kierkegaard

## RESUMO

O trabalho em questão expõe uma pesquisa sobre a automação e acionamento de um ventilador, mediante o uso de um microcontrolador ESP32, cuja gestão pode ser operada por meio do aplicativo Amazon Alexa. Tal dispositivo encontra-se vinculado à rede mundial de computadores, por meio da comunicação sem fio. Para realizar a conexão entre o microcontrolador e o ventilador, três saídas são usadas, para controlar a velocidade de funcionamento do referido equipamento. O programa da Amazon Alexa, por sua vez, exercerá o controle remoto do ESP32, através da conexão sem fio. Este estudo objetiva contribuir para o aumento do número de dispositivos pertencentes à chamada Internet das Coisas (IoT), no contexto atual do mercado.

**Palavras-chave:** ventilador, Amazon, Alexa, internet das coisas, acionamento

## **ABSTRACT**

The present work presents research on the automation and control of a fan, using an ESP32 microcontroller, whose management can be operated through the Amazon Alexa application. This device is linked to the worldwide web through wireless communication. To establish the connection between the microcontroller and the fan, three outputs will be employed to control the operating speed of the equipment. The Amazon Alexa program, in turn, will remotely control the ESP32 through wireless communication. This study aims to contribute to the increase in the number of devices belonging to the Internet of Things (IoT) in the current market context.

**Keywords:** fan, Amazon, Alexa, internet of things, microcontroller.

## LISTA DE FIGURAS

Figura 1 - Esquema de um Motor monofásico.....	16
Figura 2 - Gráfico da velocidade resultante do motor. (a) Motor de indução monofásico com base nas ondas constantes, progressivas e retrógrada, de fluxo (b) Motor de indução monofásico levando em consideração as alterações nas ondas .....	17
Figura 3 - Partes expandidas de um Motor monofásico .....	17
Figura 4 - Motor monofásico de indução de polo sombreado série TL58.....	18
Figura 6 – Conexão de um motor de partida com capacitor.....	19
Figura 7 - Esquema motor monofásico universal .....	20
Figura 8 – Principais casos de uso do Internet das Coisas.....	21
Figura 9 – Protocolo MQTT .....	21
Figura 10 – Especificação ESP32 .....	22
Figura 11 – Pinagem Esp32.....	23
Figura 12 – Consumo ESP32 em carga total .....	24
Figura 13 – Consumo ESP32 em Hibernação.....	24
Figura 14 - Alexa Echo 4 Geração .....	25
Figura 15 – Análise de uma ordem .....	26
Figura 16 – Serviço Alexa .....	26
Figura 17 – Computação em Nuvem.....	27
Figura 18 – AWS (Amazon Web Services) .....	28
Figura 19 – Tipos de serviço na computação em nuvem .....	28
Figura 20 - Componentes do Proteus .....	30
Figura 21 - Circuito genérico do trabalho .....	30
Figura 22 - Esquemático modulo Relê de 4 canais com Fotoaclopador .....	31
Figura 23 - Dispositivos com sistemas embarcados .....	32
Figura 24 - Interface Web Sinric Pro .....	33
Figura 25 - Criação do Dispositivo .....	34
Figura 26 - Seção Dispositivos.....	35
Figura 27 – Seção Credenciais .....	35
Figura 28 - Seção mais Alexa .....	36
Figura 29 - Skill e Jogos.....	36
Figura 30 - Pesquisa Skills.....	37
Figura 31 - Parte definição código.....	37
Figura 32 - Configuração para ligar o Ventilador.....	38

Figura 33 - Definições da velocidade .....	39
Figura 34 - Página Modo Estação .....	39
Figura 35 - Configuração ESP32 Leds .....	40
Figura 36 - Configuração código Wifi .....	40
Figura 37 - Setup programa .....	41
Figura 38 - Diagrama de bloco projeto .....	41

## LISTA DE ABREVIATURA

M.E.E	Mestre em Engenharia Elétrica
Ma	Mestra
PUC	Pontifícia Universidade Católica
Prof	Professor
Dr	Doutor
Phd	<i>Philosophy Doctor</i>

## LISTA DE SIGLAS

IoT	<i>Internet of Things</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
TCP	<i>Transmission Control Protocol</i>
IP	<i>Internet Protocol</i>
IDE	<i>Integrated Development Environment</i>
AWS	<i>Amazon web service</i>
Web	<i>World Wide Web</i>
AVS	<i>Amazon voice service</i>
IaaS	infraestrutura como serviço
PaaS	Plataforma como serviço
SaaS	Software como serviço
PCB	Placa de circuito impresso
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos

## SUMÁRIO

1	Introdução.....	13
1.1	Objetivos.....	13
1.1.1	Objetivo Geral .....	14
1.1.2	Objetivo Específico.....	14
1.2	Justificativa .....	14
1.3	Métodos .....	14
1.5	Resultados Esperados.....	15
2	Fundamentação Teórica.....	16
2.1	Motor Monofásico .....	16
2.2	ESP32.....	20
2.3	Alexa .....	24
2.4	Computação em Nuvem .....	27
2.5	<i>WebSocket</i> .....	29
2.6	Proteus .....	29
2.7	Sistemas embarcados .....	32
2.8	Sinric Pro .....	32
3	Proposta de solução.....	34
4	Considerações finais.....	42
5	Referências .....	44
	APÊNDICES.....	48
	ANEXO.....	53

## **1 Introdução**

No cenário atual do desenvolvimento tecnológico um conceito que vem crescendo exponencialmente no mercado de eletrônicos, usado em automações residenciais, industriais, ambientes hospitalares e segurança e Internet das coisas (IoT). Trata-se de uma tecnologia que utiliza um protocolo onde é realizado um efeito de malha em cada um dos dispositivos que são conectados em uma rede sem fio. Essas malhas são interconectadas entre si possibilitando a comunicação uma com a outra através desta rede sem fio. (Oliveira, 2021)

Tendo em vista que há um aumento na quantidade de dispositivos IoT, uma grande dificuldade é o custo, juntamente com o nível de consumo de energia. Atualmente pode-se encontrar dispositivos de baixo consumo, que fazem a comunicação dos dispositivos IoT a rede sem fio e se dialogam com a rede, estes são os microcontroladores, sendo ESP32, Arduino, NodeMcu, os principais no mercado. (Oliveira, 2021)

Com o aumento do uso de smartphones e assistentes virtuais, como Alexa, Google Assistente e Siri, os dispositivos IoT, se tornam acessíveis. Isso ocorre porque, com um smartphone e assistentes virtuais, é possível controlar funções como ligar, desligar, mudar a cor do dispositivo e verificar o consumo de energia, seja por meio de um aplicativo ou utilizando comandos de voz. (Tadeu, 2016)

O primeiro capítulo apresenta os objetivos e métodos que são utilizados para realização do projeto, a justificativa, a lista de atividades e os resultados esperados.

No segundo capítulo é apresentado uma breve exposição do assunto a ser tratado no trabalho destacando conteúdos importantes para entendimento do trabalho.

### **1.1 Objetivos**

Neste tópico são apresentados os objetivos de propósito geral e específico do projeto.

### 1.1.1 Objetivo Geral

Esse trabalho tem como objetivo demonstrar o funcionamento dos circuitos e programas necessários para automatização de um ventilador, utilizando a integração com o aplicativo Alexa da Amazon.

### 1.1.2 Objetivos Específicos

- Elaborar um circuito para conexão do microcontrolador e os relês.
- Programar o uso do microcontrolador para controle de velocidade e energia do ventilador.
- Utilizar a conexão de rede wireless (sem fio) para ligação do microcontrolador com dispositivos conectados na rede IoT.
- Criar rotinas para automatização do uso do ventilador.

## 1.2 Justificativa

Conforme uma pesquisa realizada pela *fortune business insights*, a estimativa do aumento do mercado de dispositivos que realizam a automação residencial, tem como previsão sair dos atuais US\$ 72,30 bilhões em 2021 para US\$ 163,24 bilhões em 2028, tendo em vista um aumento percentual de 12,3 % ao ano conforme a taxa de crescimento anual composta. (Agência New Voice, 2022)

Tendo em vista esse aumento o trabalho apresentado tem como justificativa, a criação de dispositivos para automatizar um ventilador para uma casa conectada, através de um microcontrolador pois no presente momento possui poucas opções no mercado para estes dispositivos que são de grande importância para deixar o ambiente da casa arejado.

## 1.3 Métodos

Para fins de elaboração deste trabalho, foi necessário realizar pesquisas com o propósito de expandir as competências na elaboração dos componentes e circuitos necessários, que são desenvolvidos com auxílio do programa Protheus, neste software foi executado o cálculo dos resistores e capacitores utilizados juntamente

com o projeto e simulação do circuito, na elaboração deste, são utilizados artigos científicos, trabalhos de conclusão de cursos, livros, publicações online e manuais de fabricante.

O trabalho de conclusão de curso será descrito e realizado a partir da montagem dos componentes eletrônicos adquiridos, como microcontrolador, relês, resistores, capacitores em uma placa de circuito impresso, estes componentes serão organizados e colocados em uma case de régua que estará interligada no ventilador. Para automação do projeto foi utilizado o aplicativo nativo da Amazon denominado Alexa, para controlar o ventilador, tendo em vista as opções de ligar e desligar o dispositivo, e controlar as 3 velocidades que são definidas na programação.

### **1.5 Resultados Esperados**

Espera-se que no final deste trabalho seja possível o uso deste dispositivo nas residências e que seja de baixo custo de produção.

Espera-se que a sociedade utilize este trabalho para automatizar a área residencial tendo em vista a eficiência energética, sendo transparente à utilização de energia do usuário com o dispositivo e que seja possível futuramente melhorá-lo.

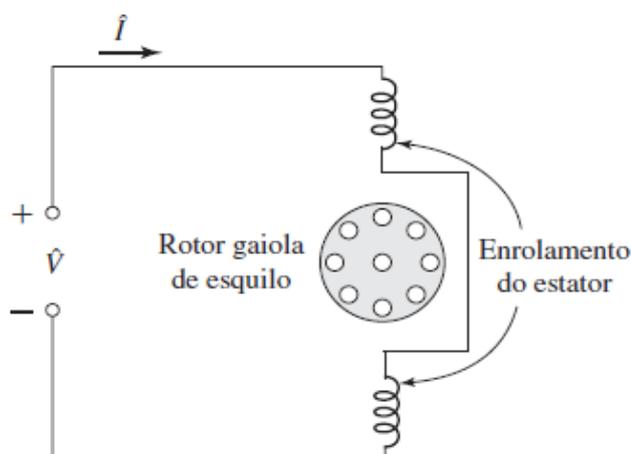
## 2 Fundamentação Teórica

Para deslocamentos de ar utilizados em circulação, na atualidade os dispositivos mais simples a serem utilizados seriam os ventiladores, o conjunto utilizado para gerar o movimento normalmente trata-se de dois tipos de máquinas elétricas encontradas no comércio para esses citados dispositivos. Essas máquinas são baseadas em motores monofásicos em motores *brushless*. Como mais de 90 % dos motores encontrados no comércio para dispositivos do tipo ventilador são motores monofásicos então descreve-se o motor monofásico.

### 2.1 Motor Monofásico

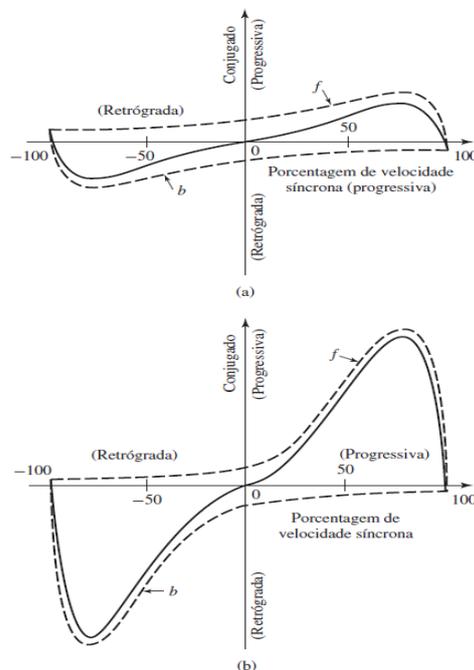
O motor monofásico consiste em um motor semelhante ao bifásico com enrolamentos assimétricos conforme demonstra a figura 1. Esse motor, portanto, consegue produzir campos girantes eletromagnéticos iguais ao se rotacionar para frente ou para trás o rotor em relação ao estator pois possui simetria, para que o motor consiga rotacionar em uma direção é necessário usar uma força auxiliar, promovida por um enrolamento direcional, adaptado entre o enrolamento normal do campo girante do próprio motor. Pode-se observar a força realizada no movimento na figura 1. (Fitzgerald, 2003).

Figura 1 - Esquema de um Motor monofásico de indução



Fonte: (Fitzgerald, 2003).

Figura 2 - Gráfico da velocidade resultante do motor. (a) Motor de indução monofásico com base nas ondas constantes, progressivas e retrógrada, de fluxo (b) Motor de indução monofásico levando em consideração as alterações nas ondas



Fonte: (Fitzgerald, 2003).

Em um motor monofásico o estator consiste em um cilindro com pequenas aberturas, denominadas ranhuras, que permitem acomodar os fios da bobina em um núcleo de aço silício, que por sua vez, ao passar uma corrente alternada nos fios, é gerado um campo magnético rotativo, para um melhor entendimento observar Figura 3. (Bertulucci, 2016)

Figura 3 - Partes expandidas de um Motor monofásico



Fonte: (Bertulucci, 2016)

O Rotor é feito com uma haste, denominada eixo, e devidamente colocadas barras condutivas em volta do seu núcleo. Essas barras geram um campo magnético e elétrico quando correntes são induzidas nos enrolamentos do estator. Os enrolamentos são um agrupamento de fios de cobre envolvido em um núcleo ferromagnético que produzem um campo magnético. Isso faz com que o rotor gire, devido ao campo magnético girante e crie torque (Bertulucci, 2016)

Quando uma tensão alternada é aplicada aos enrolamentos do estator, um campo rotativo é formado. Isso faz com que a corrente flua através dos fios condutores de corrente do rotor, que faz o ponto de partida do motor. (Fitzgerald, 2003)

Existem vários tipos de motores monofásicos sendo; motor de polo sombreado, motor de fase dividida, motor de capacitor dividido permanente e motor de capacitor de dois valores. Os motores de polos sombreados e motores de fase dividida não usam capacitores e os motores de capacitor dividido permanentemente e motor de capacitor de dois valores utilizam capacitores, sendo os mais baratos para aquisição são motores de polo sombreado e motores de fase dividida.

O motor de indução de polo sombreado é um motor de indução monofásico que usa um anel de cobre para dar partida automaticamente. Outro nome para o anel de cobre é o anel de sombra, que atua como um enrolamento do motor do anel secundário. Ele só gira em uma direção específica, é impossível reverter o momento. O motor tem perdas por indução de potência muito altas e tem um baixo fator de potência, estes motores são usados em coifas, bombas de óleo, fornos de micro-ondas, e refrigeradores como pode se observar na figura 4. (WentLon)

Figura 4 - Motor monofásico de indução de polo sombreado série TL58

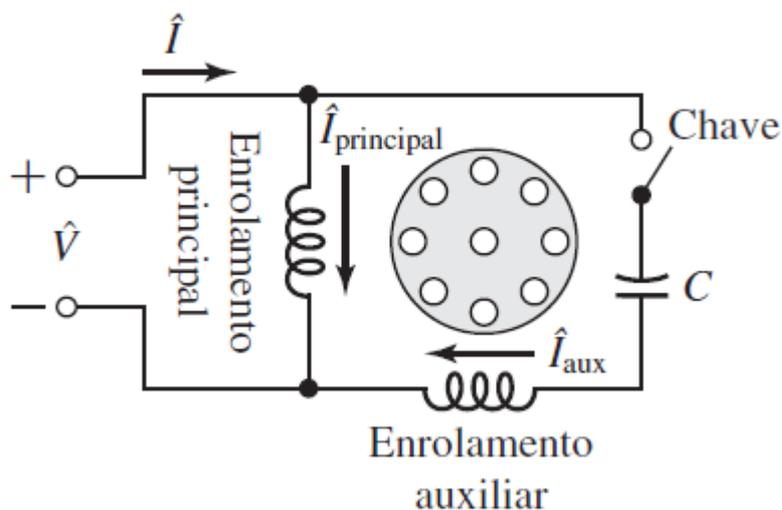


Fonte: (WentLon)

O motor de capacitor de dois valores, possui em seu circuito um enrolamento principal que fica interligado a uma rede de fornecimento de energia, um outro enrolamento auxiliar fica conectado, usualmente a 90 graus do enrolamento principal, mediante a conexão com dois ou mais capacitores, um dos capacitores possui a função de partida do circuito, subseqüente tendo a função de condição de regime. (ABNT, 2016)

Um motor que utiliza capacitor de partida para ter êxito é necessário possuir o capacitor em série com o enrolamento auxiliar direcionador do sentido do movimento, observar Figura 6. (Fitzgerald, 2003)

Figura 5 – Conexão de um motor de partida com capacitor

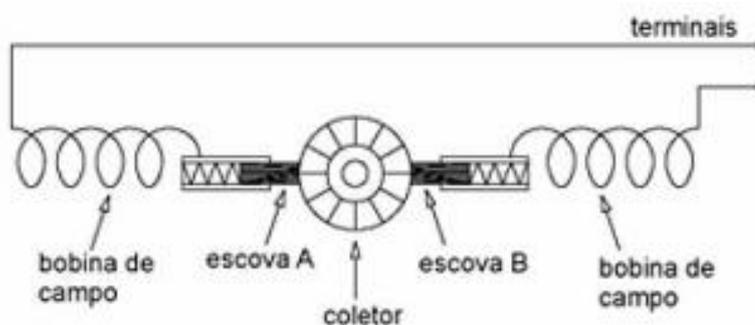


Fonte: (Fitzgerald, 2003)

Se os campos de fluxo direto e reverso no entreferro permanecerem os mesmos à medida que o rotor gira, cada campo componente produzirá características de torque-velocidade semelhantes a um motor polifásico com impedância de dispersão do estator desprezível (Fitzgerald, 2003). O conjugado é definido pela quantidade de esforço necessário para se realizar um movimento para girar o eixo. (Fitzgerald, 2003)

Neste trabalho será utilizado um motor monofásico do tipo universal, estes motores são ligados em série e possuem estrutura laminada para diminuir as perdas por correntes parasitas, este motor também é composto por um enrolamento de campo e um enrolamento de armadura ou enrolamento induzido. O enrolamento de campo precisa ser executado em um núcleo formado pelo encapsulamento de uma placa de silício para desviar as correntes parasitas existentes, pode-se observar na figura 7 o esquema do motor monofásico do tipo universal. (Pereira, 2018)

Figura 6 - Esquema motor monofásico universal



Fonte: (Pereira, 2018)

## 2.2 ESP32

Internet das coisas ou em inglês *Internet of Things* (IoT) é composto por protocolos e tecnologias que se comunicam e permitem que diferentes componentes se dialoguem em uma rede de troca de dados e informações composta por coisas e pessoas por meio de canais de comunicação eletrônicos com ou sem fio. (Santos; Sales, 2016)

Um exemplo de IoT é a conexão de dispositivos sendo notebooks, smartphones, televisores, lâmpadas, que por sua vez conseguem se comunicar com a rede de internet, como se pode observar na figura 8. (Junior; Moreno, 2016)

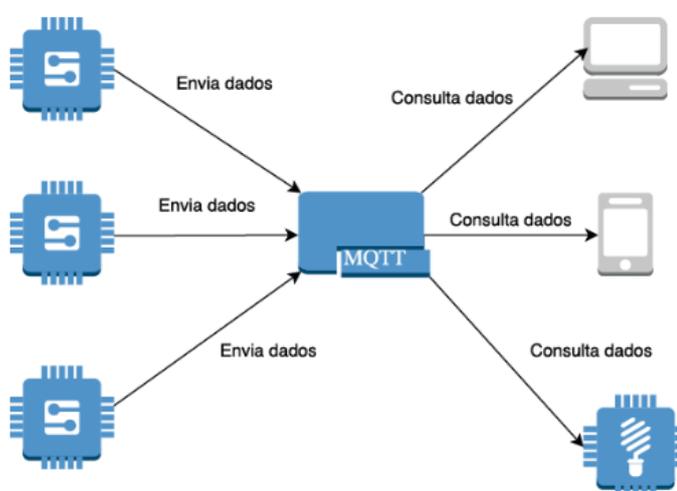
Figura 7 – Principais casos de uso do Internet das Coisas



Fonte Adaptada: (Tibco, 2022)

Um protocolo que é usado em IoT é o MQTT que utiliza o conceito de broker, sendo um servidor para receber e enviar os dados, e transmite esses dados quando solicitado através do TCP/IP. Este *broker* consegue definir se os dados dos dispositivos serão apenas para leitura ou mudar seu estado, desligado, ligado, como pode-se observar na figura 9. (Oliveira, 2021)

Figura 8 – Protocolo MQTT



Fonte: (Oliveira, 2021)

O ESP32 é um microcontrolador de baixo consumo, normalmente usado em projetos IoT (Internet das coisas), que possui já integrado um módulo *wireless* e um

modulo *Bluetooth* em seu corpo. Suas especificações podem ser observadas na figura 10. (Oliveira,2017)

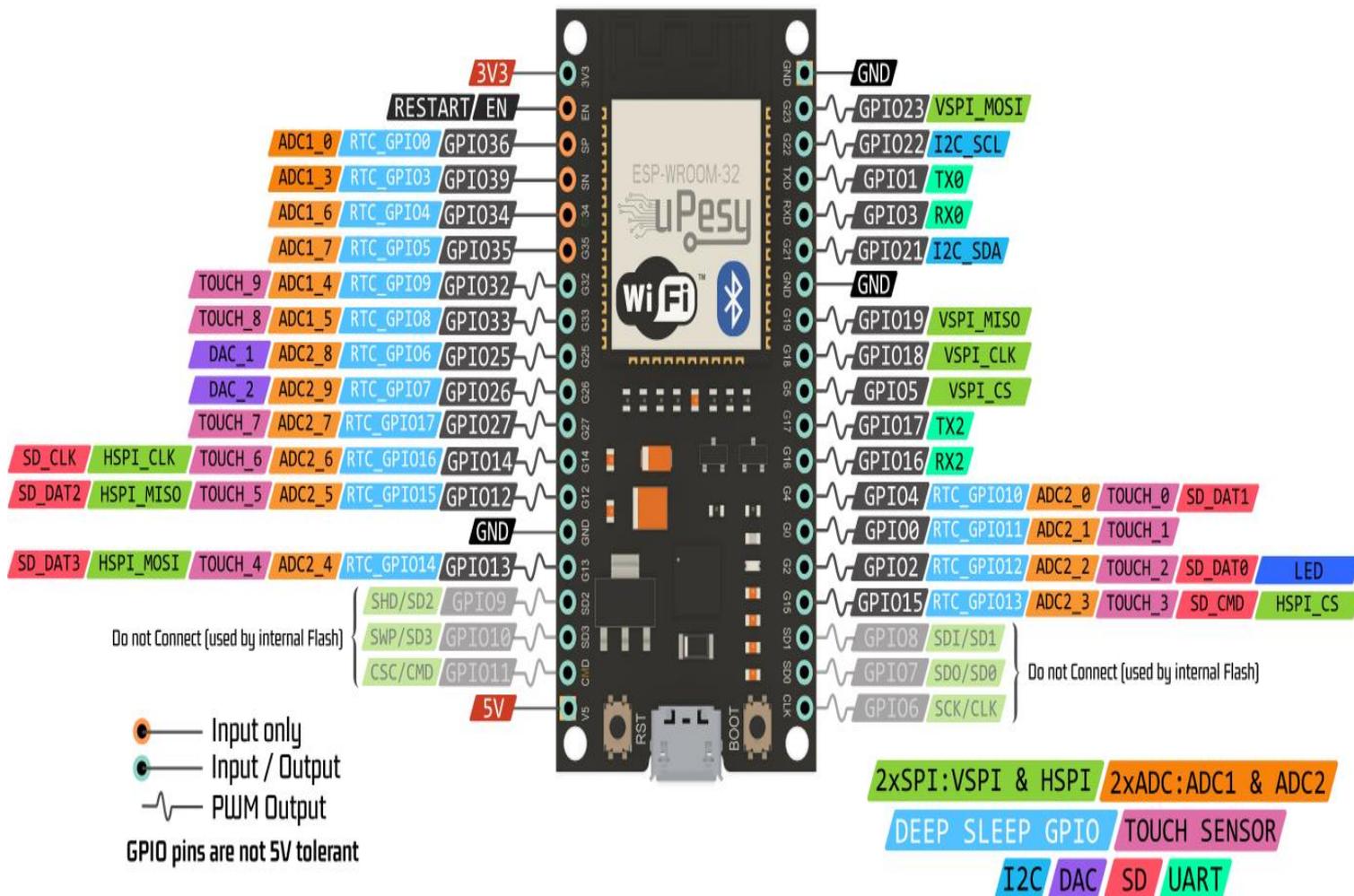
Figura 9 – Especificação ESP32

	ESP32
Cores	2
Arquitetura	32 bits
Clock	160MHz
WiFi	Sim
Bluetooth	Sim
RAM	512KB
FLASH	16Mb
GPIO	36
Interfaces	SPI / I2C / UART / I2S / CAN
ADC	18
DAC	2

Fonte: (Oliveira)

Este microcontrolador possui 36 pinos, como demonstra a figura 11, e pode ser utilizado em aplicações como automação residencial, industrial e agrícola, dispositivos de saúde, reconhecimento de voz, imagem e câmeras de monitoramento. (ALLDATASHEET)

Figura 10 – Pinagem Esp32

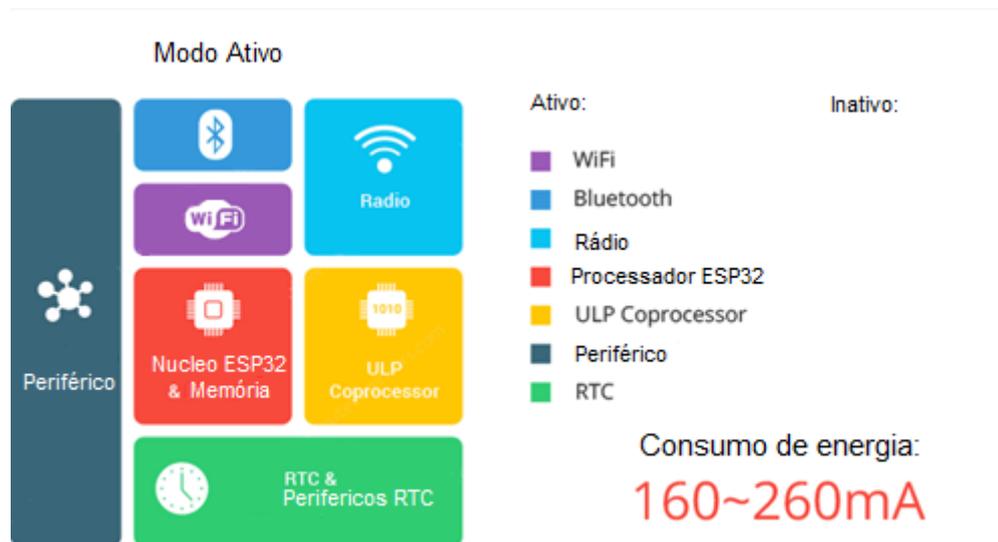


Fonte:(ALLDATASHEET, 2022)

O Microcontrolador ESP32 pode ser utilizado em diversos projetos, utilizando sensores sendo capaz de transferir os dados adquiridos nos sensores para internet ou para nuvem. (Oliveira, 2021)

É compatível com a IDE Arduino. Esse microcontrolador possui processamento suficiente para executar projetos que necessitam da utilização de áudio, vídeo e possuindo a vantagem energética pois é classificado como dispositivo *de baixo consumo*, consumindo alguns *miliwatts* em seu funcionamento e em modo de repouso consome na ordem de *microwatts* conforme ilustra a figura 12 e 13. (Oliveira, 2021)

Figura 11 – Consumo ESP32 em carga total



Fonte Adaptada: (Last Minute Engineers, 2022)

Figura 12 – Consumo ESP32 em Hibernação



Fonte adaptada: (Last Minute Engineers, 2022)

## 2.3 Alexa

A Alexa foi divulgada em 2014 ao público, e consiste em uma assistente virtual criada pela empresa da Amazon. Juntamente com ela foi lançado também a caixa de som Echo que possui a Alexa integrada em seu funcionamento. Pode-se ver sua

aparência na figura 14. Ela possui uma conexão sem fio que se conecta a internet. Auxilia na execução de tarefas simples como ativar alarmes, falar a previsão do tempo, reproduzir músicas utilizando aplicativos de streaming de vídeos, e controlar aparelhos inteligentes podendo ligar, desligar, ou alterar seu funcionamento alterando dados dos dispositivos. (Tecnoblog, 2021)

Figura 13 - Alexa Echo 4 Geração



Fonte: (Canaltech, 2022)

Esta assistente virtual pode também solicitar uma compra no site da Amazon e entregar no endereço da conta a que ela está conectada, fazer pedidos em alguns aplicativos de terceiros como Pizza Hut, Starbucks, estando aberta também para novas funcionalidades que são chamadas de *skills* que podem ser criadas por terceiros. (Tecnoblog, 2021)

No processo de reconhecimento de voz é usado o serviço AVS (*Alexa voice service*) onde é feito o processamento natural da linguagem, gravando a fala através do dispositivo Alexa echo dot e enviando os dados para um servidor da Amazon que transforma esse áudio em formato de texto. Para transcrever este áudio é usada uma inteligência artificial com *deep learning* na qual é utilizado um algoritmo para identificar as correlações dos conjuntos de padrões para analisar as frases dadas, onde a frase é separada em 3 partes: uma identifica qual será a palavra para fazer a Alexa acordar, podendo utilizar a palavra “Alexa” ou “Amazon”, tendo escolha no aplicativo da

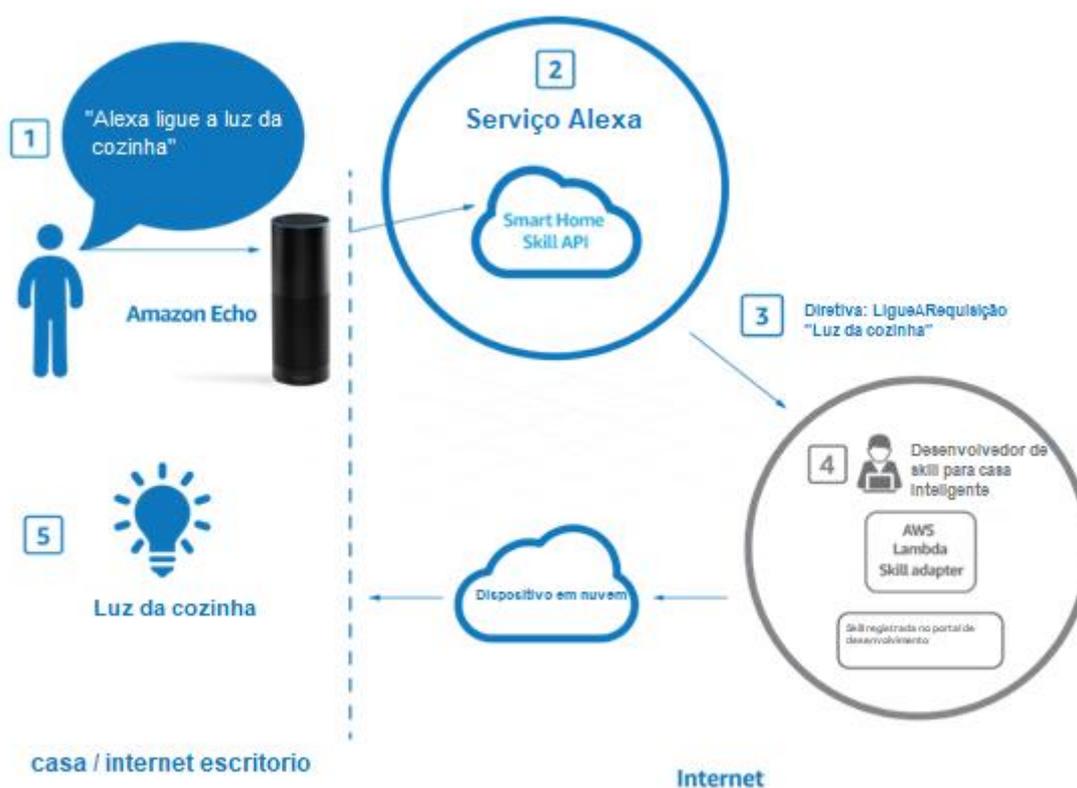
Amazon Alexa; uma palavra para invocação da habilidade tendo como exemplo uma ação ou comando para ela executar; o último tópico a ser analisado é o que foi exposto para Alexa executar. Pode-se observar na figura 15 e 16 seu funcionamento. (Golfalonieri, 2018)

Figura 14 – Análise de uma ordem



Fonte adaptada: (Golfalonieri, 2018)

Figura 15 – Serviço Alexa

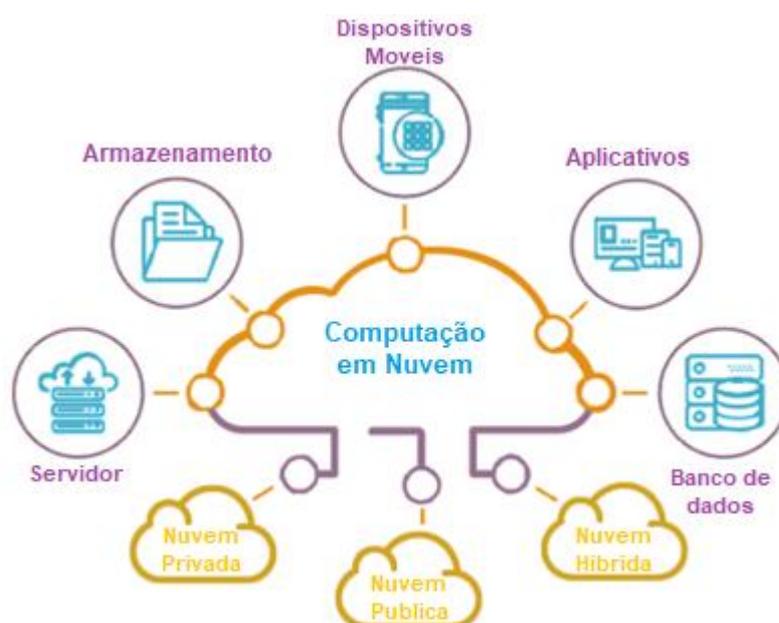


Fonte adaptada: (Golfalonieri, 2018)

## 2.4 Computação em Nuvem

A computação em nuvem surge através de servidores externos conectados na internet, que são utilizados basicamente para armazenamento de dados, ou processamento dos mesmos, por isso é usado a terminologia “Nuvem”. Esses dados originam-se de dispositivos como computadores, celulares, notebooks que mandam os dados para serem processados nesses servidores externos, como se pode observar na figura 17. (Oliveira, 2021)

Figura 16 – Computação em Nuvem



Fonte Adaptada: (Baird, 2019)

Os benefícios provenientes da utilização de serviços em computação em nuvem, demonstram inúmeros proveitos na agilidade, na implantação dos recursos gráficos, de armazenamento de dados, gerenciamento e análise de dados, podendo também ajustar qual será a quantidade da utilização desses recursos podendo diminuir ou aumentar de acordo com a evolução do programa, podendo diminuir o custo do projeto, que ao usar esse serviço se pagará apenas pelos recursos solicitados. (Amazon, 2022)

O serviço de nuvem que será usado neste trabalho é o AWS (*Amazon Web Services*) que é disponibilizado pela Amazon, neste serviço de modelo *IaaS* (infraestrutura como serviço) tem-se bancos de dados, serviços de processamentos

de dados, biblioteca de inteligência artificial e aprendizado de máquina, que podem ser utilizados pelo cliente para desenvolver seus aplicativos com estes recursos conforme observado na figura 18. (Oliveira, 2021)

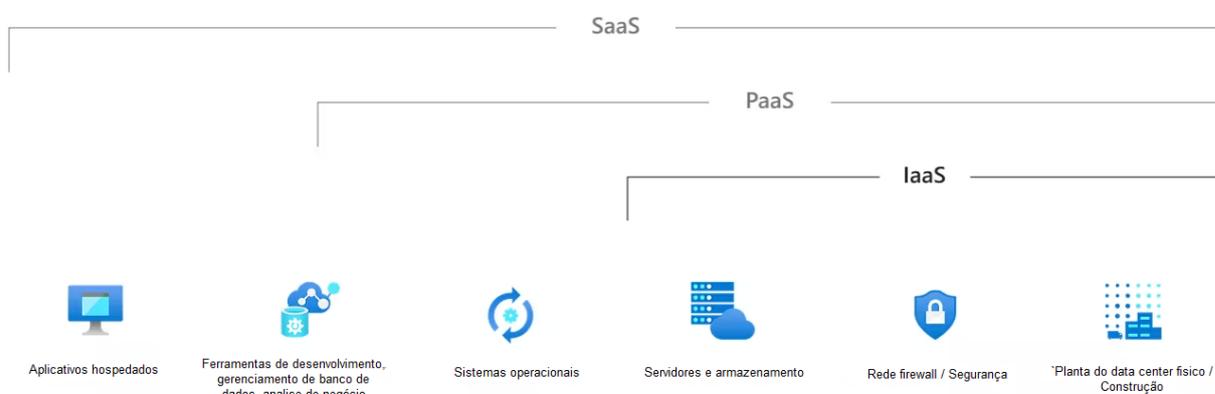
Figura 17 – AWS (Amazon Web Services)



Fonte Adaptada: (SpiceWorks)

O tipo IaaS da computação em nuvem, oferece recursos como: armazenamento, redes de segurança e servidores físicos, sendo que não há a necessidade de maior dispêndio financeiro se houver um grau de computação de recursos elevados, pois cada recurso é separado por módulo e é feita a cobrança por módulos conforme demonstra a figura 19. (Azure, 2022)

Figura 18 – Tipos de serviço na computação em nuvem



Fonte adaptada: (Azure, 2022)

Esse tipo oferece vantagens como Backup dos arquivos, recuperação se houver falhas, aplicativos para gerenciamento dos recursos, segurança aprimorada, fortalecimento a segurança na aplicação e gerenciamento dos recursos financeiros. (Azure, 2022)

## **2.5 WebSocket**

O protocolo *WebSocket* é um protocolo de comunicação bidirecional que permite a comunicação em tempo real entre o navegador web e o servidor. Ele é baseado em uma conexão TCP única e mantida aberta, permitindo que ambas as partes enviem dados a qualquer momento sem precisar estabelecer uma nova conexão. O protocolo *WebSocket* é amplamente utilizado em aplicativos web que exigem atualizações em tempo real, como jogos online, bate-papo ao vivo, feeds de notícias, entre outros. (Taylor, 2022)

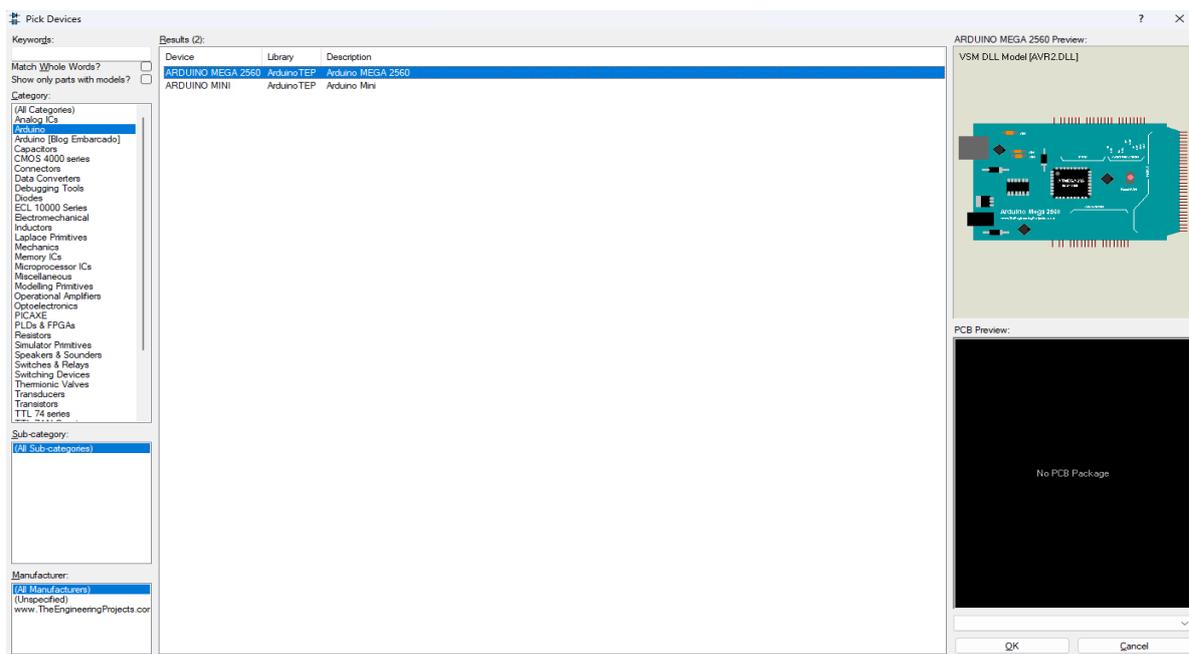
## **2.6 Proteus**

O Proteus é um software desenvolvido pela empresa Labcenter Electronics, elaborado para aplicações analógicas e digitais. Este software permite criar esquemas de circuitos, simulações de circuitos de microcontroladores e circuitos elétricos. (Labcenter, 2022)

Um dos principais recursos do Proteus é a capacidade de criar templates de placas PCB (placa de circuito impresso). Isso permite que os engenheiros projetem e visualizem a disposição física dos componentes eletrônicos em uma placa de circuito. O software também oferece ferramentas para a roteamento automático de trilhas, otimizando o layout da placa para eficiência e confiabilidade. (Labcenter, 2022)

Além disso, o Proteus possui ferramentas robustas para análise de circuitos. Ele permite que os usuários analisem o comportamento do circuito projetado usando multímetros virtuais, osciloscópios e analisadores de espectro. Isso permite que os engenheiros testem e validem seus designs de circuitos antes da fabricação física, economizando tempo e recursos, pode-se observar a ampla biblioteca na figura 20. (Labcenter, 2022)

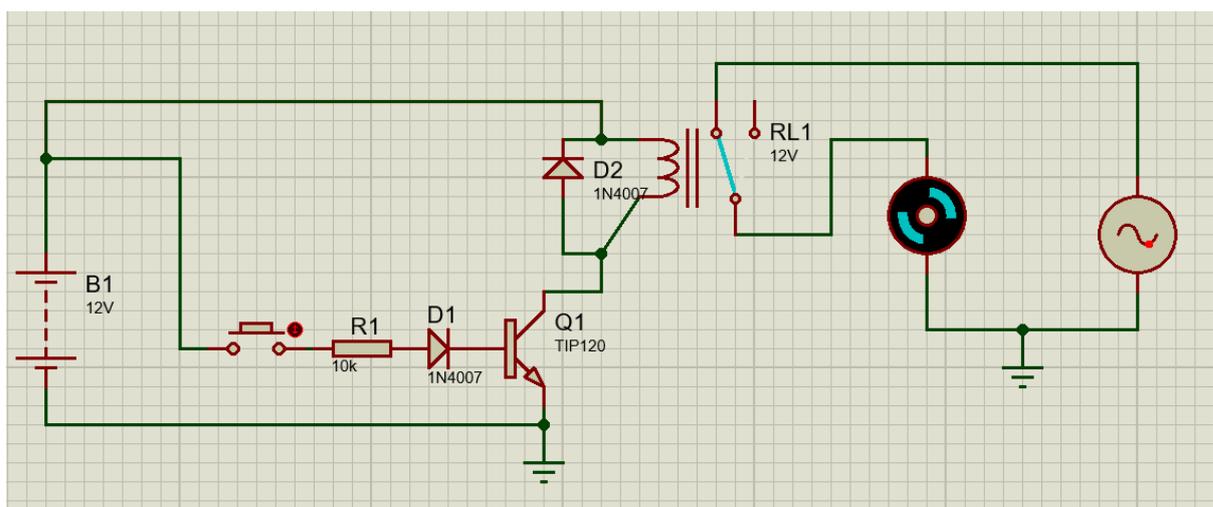
Figura 19 - Componentes do Proteus



Fonte: (Labcenter, 2022)

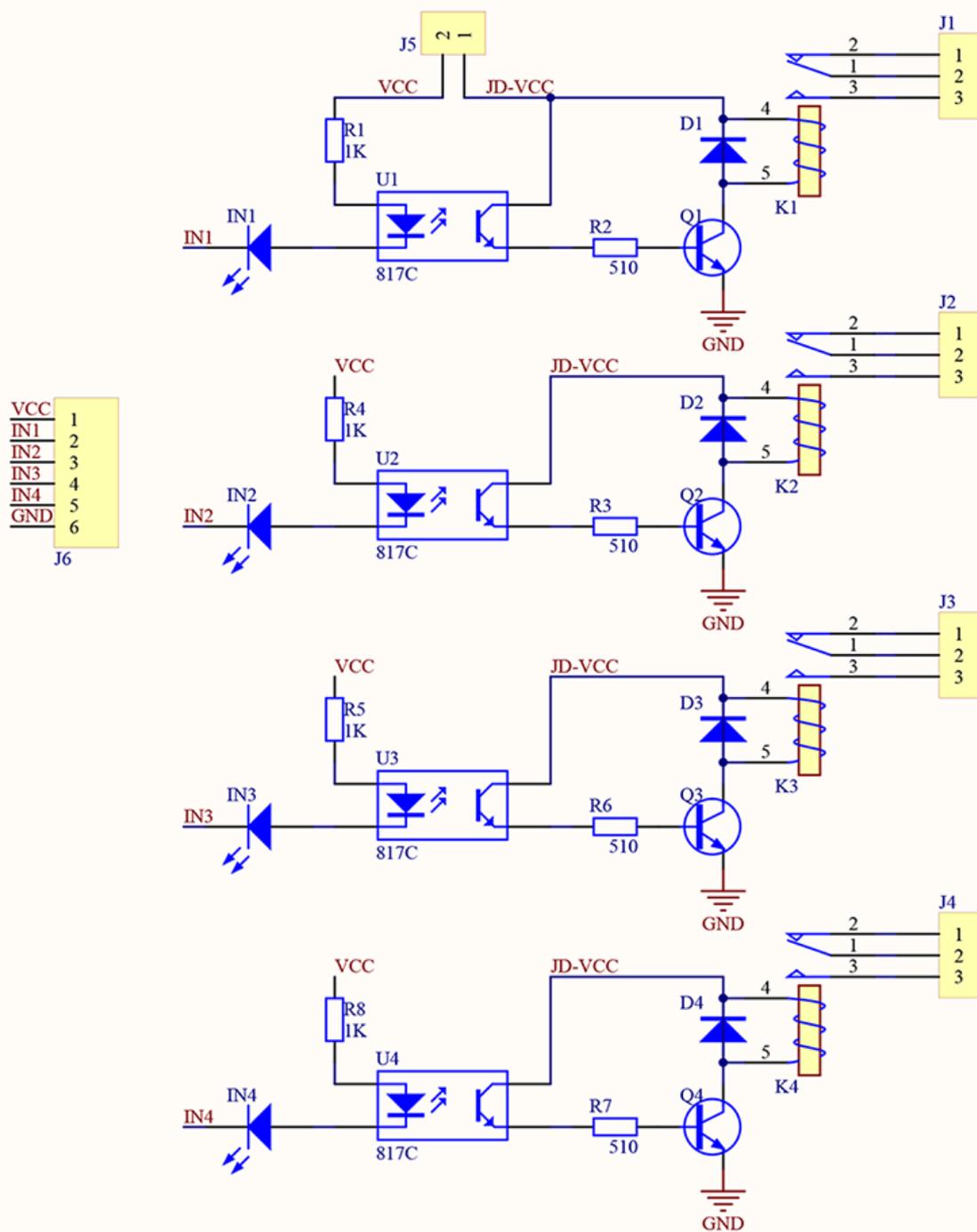
Foram utilizados, alguns componentes, como. Transistor TIP120, um motor AC monofásico, 3 relês, resistores, diodos, cada entrada deste circuito vai ser conectada no pino analógico do microcontrolador ESP32, a figura 21 e 22 demonstra os o circuito de conexão dos relês ao motor do ventilador.

Figura 20 - Circuito genérico do trabalho



Fonte:(elaborado pelo autor, 2022)

Figura 21 - Esquemático modulo Relê de 4 canais com Fotoaclopador



Fonte: (Handson Technology, 2023)

## 2.7 Sistemas embarcados

Esses sistemas possuem um poder de computação independente pois são computadores de propósito específico, ou seja, sempre executam a mesma ação, podendo ser repetida infinitas vezes. Tem-se como exemplo destes dispositivos: televisores, ar-condicionado, câmara fotográficas podendo ser observados na figura 23, possuindo sistemas operacionais próprios para estes dispositivos, pois apresentam processamento limitado relacionado à tarefa. (wikilivros, 2020)

Figura 22 - Dispositivos com sistemas embarcados



Fonte:(Kofuji, Sergio; Rehder, Gustavo; Seabra, Antonio, 2017)

Estes sistemas operacionais são importantes pois possuem semáforos que definem as prioridades das tarefas que serão executadas com auxílio de rotinas que utilizam o tempo do relógio deste sistema e escalonadores. (*Brazilian RTOS blog*, 2013)

## 2.8 Sinric Pro

A plataforma Sinric Pro consiste em uma solução tecnológica de automação residencial que oferece aos seus usuários uma ampla gama de recursos para gerenciar e monitorar dispositivos inteligentes em suas casas. Esta plataforma utiliza o *WebSocket* para funcionamento. Por meio de um sistema completo integrado, que combina assistentes virtuais, aplicativos móveis e comandos de voz, os usuários

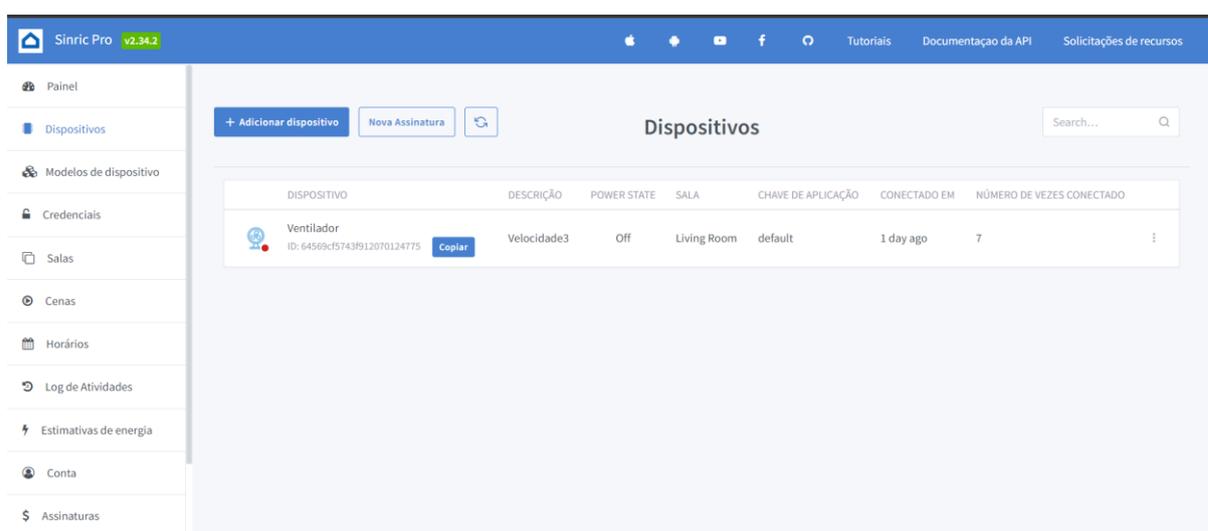
podem controlar desde a temperatura e iluminação até os aparelhos domésticos e sistemas de segurança. (SinricPro, 2023)

No âmbito da sua funcionalidade, a plataforma disponibiliza ferramentas de personalização, como a criação de cenas, que podem ser programadas de forma customizada, proporcionando uma experiência única de utilização. Ademais, a Sinric Pro também oferece monitoramento de energia, possibilitando aos usuários controlar os seus gastos e, assim, obter economias significativas em suas contas de serviços públicos. (SinricPro, 2023)

Esta Plataforma oferece uma *skill* que permite a integração da plataforma com a Amazon Alexa e o Google Assistant. Com a *skill*, os usuários podem controlar e monitorar seus dispositivos inteligentes usando comandos de voz. (SinricPro, 2023)

Ao habilitar a *skill* da Sinric Pro em seus assistentes virtuais, os usuários podem controlar todos os dispositivos conectados à plataforma com facilidade. Por exemplo, é possível ajustar a temperatura do ar-condicionado, acender ou apagar as luzes, abrir e fechar as cortinas, e até mesmo ligar ou desligar aparelhos eletrônicos, tudo por meio de comandos de voz. Pode-se observar na figura 24 a interface da plataforma. (SinricPro, 2023)

Figura 23 - Interface Web Sinric Pro



Fonte: (elaborado pelo autor,2023)

### 3 Implementação da solução

Neste capítulo, são apresentadas as tecnologias e ferramentas empregadas no projeto com base no referencial teórico do trabalho.

A fim de usufruir dos recursos disponibilizados pela plataforma Sinric Pro, é imprescindível registrar-se previamente no site oficial <https://sinric.pro/pt-index>. Uma vez cadastrado, é possível proceder à criação de um dispositivo específico para o controle de um ventilador, conforme exemplificado na Figura 25.

Figura 24 - Criação do Dispositivo

Panel

- Dispositivos
- Modelos de dispositivo
- Credenciais
- Salas
- Cenas
- Horários
- Log de Atividades
- Estimativas de energia
- Conta
- Assinaturas
- Novidades

### Novo dispositivo

1 Informação do dispositivo 2 Notificações 3 Temporizadores 4 Outra

Nome do dispositivo  
Ventilador

Descrição  
Description

Tipo do dispositivo  
Fan

Chave de aplicação  
default

Sala  
Living Room

\* Recomendamos a utilização de diferentes chaves de aplicação para diferentes módulos de hardware

Próxima

Como conectar meu dispositivo? Enunciados para Alexa e Google Home

Fonte: (Elaborado pelo autor, 2023)

Na seção "Dispositivos e Credenciais" da plataforma Sinric Pro, é essencial armazenar cuidadosamente o ID do dispositivo criado, bem como a chave e a senha do aplicativo, conforme demonstrado respectivamente na Figura 26 e Figura 27. Essas informações são de extrema importância para estabelecer a comunicação correta entre o dispositivo e a plataforma, garantindo um funcionamento adequado e seguro. É necessário certificar-se de registrar e manter esses dados em um local seguro, pois eles serão necessários para a configuração adequada da integração entre o dispositivo e a plataforma Sinric Pro.

Figura 25 - Seção Dispositivos

DISPOSITIVO	DESCRIÇÃO	POWER STATE	SALA	CHAVE DE APLICAÇÃO	CONECTADO EM	NÚMERO DE VEZES CONECTADO
Ventilador ID: 64569cf5743f912070124775	Ventilador	Off	Living Room	default	5 days ago	7

Fonte: (Elaborado pelo autor, 2023)

Figura 26 – Seção Credenciais

Por favor copie sua chave de API e salve em um lugar seguro, não compartilhe com ninguém. O vazamento de suas credenciais pode resultar no uso indevido de suas credenciais, bem como cobranças não esperadas.

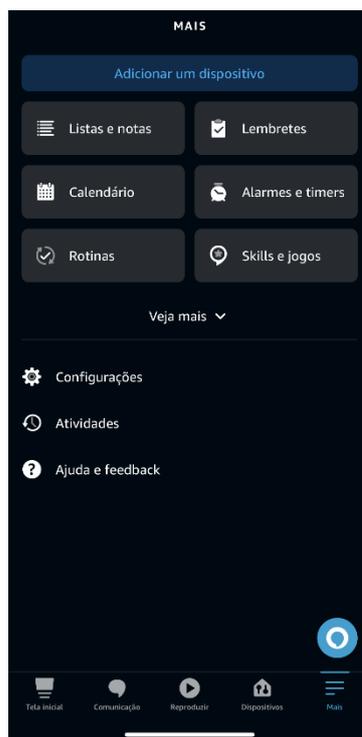
Suas chaves e senhas dos Apps

NOME	CHAVE DO APP	SENHA DO APP	CRIAD
default	b563c127-ef54-4ffe-bXXXXXXXXXXXXXXXXX	7c53ee25-51ea-4d33-bc36-00b232053345-084d01d1-27a2-428c-aXXXXXXXXXXXXXXXXX	2023-

Fonte: (Elaborado pelo Autor, 2023)

Após fazer as configurações no site, é necessário adicionar a skill do Sinric Pro no aplicativo Alexa. Para isso, dentro do aplicativo, é necessário entrar na seção "mais" e clicar em "skills e jogos". Em seguida, a pessoa deve realizar uma pesquisa pelo nome "Sinric Pro" na barra de busca. Ao encontrar a skill desejada, é necessário ativá-la e fazer o login na conta previamente criada na plataforma web do Sinric Pro, conforme mencionado anteriormente. Para auxiliar no processo, é possível observar o passo a passo nas figuras 28, 29 e 30 do aplicativo.

Figura 27 - Seção mais Alexa



Fonte: (Elaborado pelo autor, 2023)

Figura 28 - Skill e Jogos



Fonte: (Elaborado pelo autor, 2023)

Figura 29 - Pesquisa Skills



Fonte: (Elaborado pelo autor, 2023)

No início do código, realiza-se a inclusão das bibliotecas necessárias para a funcionalidade de conexão por rede sem fio (WiFi) e as bibliotecas da plataforma Sinric Pro. Após essas inclusões, é feita a seleção do pino específico do ESP32 destinado à saída, que controlará o LED do WiFi. Quando ocorre a conexão, é utilizado uma diretiva para armazenar as informações pertinentes ao ID do Ventilador, à chave do aplicativo e à senha do aplicativo, conforme ilustrado na Figura 31, a qual é apresentada abaixo.

Figura 30 - Parte definição código

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiManager.h>

#include "SinricPro.h"
#include "SinricProFanUS.h"

#define wifiled 2 // Pino LED wifi
//-----PINOS DE OUTPUT-----
#define RELAYPIN_1 25
#define RELAYPIN_2 26
#define RELAYPIN_3 27
//-----Wifi--CONFIG----- *CASO UTILIZE A FUNCAO WifiSetup*
#define WIFI_SSID "INTELBRAS"
#define WIFI_PASS "12121980"
//-----CONFIGURACOES DO SINRIC PRO-----
#define APP_KEY "b563c127-ef54-4ffe-b7e5-35c98ce9bb2b" // chave do apap
#define APP_SECRET "7c53ee25-51ea-4d33-bc36-00b232053345-084d01d1-27a2-428c-a46d-137e28141a37" // Codigo secreto do app
#define FAN_ID "64569cf5743f912070124775" // ID do Fan
#define BAUD_RATE 9600
//
```

Fonte: (Elaborado pelo autor, 2023)

Posteriormente às definições iniciais, uma estrutura é criada com o objetivo de armazenar o estado atual do ventilador e sua respectiva velocidade. Na função "onpowerstate", é realizada uma verificação, na plataforma, para averiguar se o dispositivo está ligado ou desligado. Caso a intenção seja de ligar o dispositivo, o código desativa os relés conectados aos pinos 26 e 27, e ativa o relé conectado ao pino 25, uma vez que o ventilador sempre iniciará na primeira velocidade ao ser ligado. Caso a intenção seja de desligar o ventilador, todos os relés são desativados, como ilustrado na Figura 32 apresentada abaixo.

Figura 31 - Configuração para ligar o Ventilador

```
//-----
// estrutura para armazenar os valores do Fan
// Velocidade de 1 a 3
struct {
    bool powerState = false;
    int fanSpeed = 1;
} device_state;

bool onPowerState(const String &deviceId, bool &state) {
    Serial.printf("O fan %s\r\n", state?"Ligou":"Desligou");
    device_state.powerState = state;
    if(device_state.powerState == false) // Verifica se o dispositivo esta desligado ele desativa os relés
    {
        digitalWrite(RELAYPIN_1,HIGH);
        digitalWrite(RELAYPIN_2,HIGH);
        digitalWrite(RELAYPIN_3,HIGH);
    }
    if(device_state.powerState == true) // Caso o dispositivo ligue ele sempre começa na velocidade 1
    {
        digitalWrite(RELAYPIN_1,LOW);
        digitalWrite(RELAYPIN_2,HIGH);
        digitalWrite(RELAYPIN_3,HIGH);
    }

    return true;
}
```

Fonte: (Elaborado pelo autor, 2023)

As funções "onRangeValue" e "onAdjustRangeValue" assumem uma função primordial no contexto de atribuir a velocidade ao ventilador. No cenário em que a velocidade 1 é selecionada, observa-se que o relé associado ao pino 25 é acionado, ao passo que os relés vinculados aos pinos 26 e 27 são desativados. É fundamental destacar que, para desligar o relé, requer-se que ele seja colocado em um estado lógico alto, ao passo que um estado lógico baixo é necessário para ativá-lo. Essa mesma abordagem é adotada para as demais velocidades: o relé correspondente à velocidade selecionada é ativado, ao mesmo tempo em que os demais relés são desativados. A Figura 33, disposta abaixo, ilustra de forma esquemática esse procedimento, permitindo uma melhor compreensão do seu funcionamento.

Figura 32 - Definições da velocidade

```

// A velocidade pode ser definida de 1 a 3
bool onRangeValue(const String &deviceId, int& rangeValue) {
  device_state.fanSpeed = rangeValue;
  Serial.printf("Velocidade do fan mudada para %d\r\n", device_state.fanSpeed);
  if(device_state.fanSpeed == 1) // Se a velocidade for a 1 ele desativa os outros reles e deixa ativado somente o primeiro
  {
    digitalWrite(RELAYPIN_1,LOW);
    digitalWrite(RELAYPIN_2,HIGH);
    digitalWrite(RELAYPIN_3,HIGH);
  }
  if(device_state.fanSpeed == 2) // Se a velocidade for a 2 ele desativa os outros reles e deixa ativado somente o segundo
  {
    digitalWrite(RELAYPIN_2,LOW);
    digitalWrite(RELAYPIN_1,HIGH);
    digitalWrite(RELAYPIN_3,HIGH);
  }
  if(device_state.fanSpeed == 3) //Se a velocidade for a 3 ele desativa os outros reles e deixa ativado somente o terceiro
  {
    digitalWrite(RELAYPIN_2,HIGH);
    digitalWrite(RELAYPIN_1,HIGH);
    digitalWrite(RELAYPIN_3,LOW);
  }
  return true;
}

// Fan rangeValueDelta is from -3..+3
bool onAdjustRangeValue(const String &deviceId, int& rangeValueDelta) {
  device_state.fanSpeed += rangeValueDelta;
  Serial.printf("A velocidade do ventilador mudou aproximadamente %i to %d\r\n", rangeValueDelta, device_state.fanSpeed);

  rangeValueDelta = device_state.fanSpeed; // retorna o valor absoluto da velocidade
  return true;
}

```

Fonte: (Elaborado pelo autor, 2023)

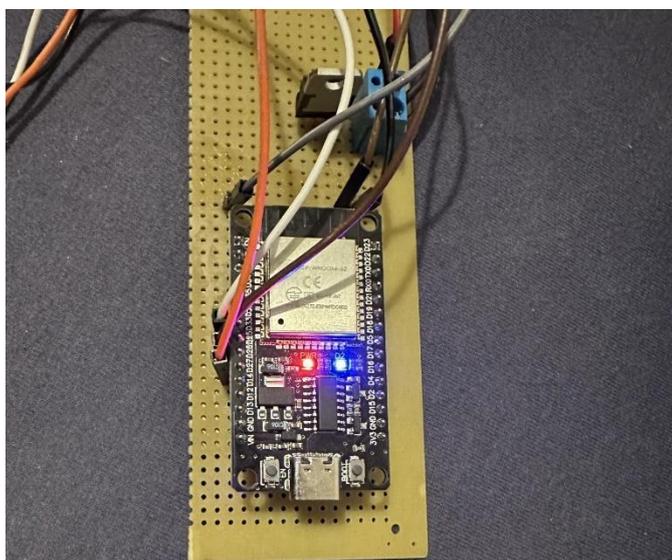
No intuito de estabelecer a conectividade por meio do protocolo WiFi, o dispositivo ESP32 é configurado no modo estação, viabilizando, assim, a administração da conexão com uma rede de acesso pontual previamente designada como "VentiladorESP". Após a conclusão bem-sucedida dessa conexão, um indicador luminoso de tonalidade azul é acionado, exercendo a função de comunicar visualmente a ativação da conexão WiFi, conforme ilustrado na Figura 34, 35, 36 apresentada abaixo.

Figura 33 - Página Modo Estação



Fonte: (Elaborada pelo autor, 2023)

Figura 34 - Configuração ESP32 Leds



Fonte: (Elaborada pelo autor, 2023)

Figura 35 - Configuração código Wifi

```
// Criando uma rede Wireless para conexão
void setupWifiSTA() {
  WiFi.mode(WIFI_STA);

  digitalWrite(wifiLed, LOW);
  bool res;
  WiFiManager wm;

  res = wm.autoConnect("VentiladorESP", "12345678");
  Serial.printf("\r\n[WiFi]: Conectando");
  Serial.printf("\r\n[WiFi]: Conectado à rede %s\r\n[WiFi]: IP-Address é %s\r\n", WiFi.SSID().c_str(), WiFi.localIP().toString().c_str());
  digitalWrite(wifiLed, HIGH);
}
```

Fonte: (Elaborado pelo autor, 2023)

Após a implementação das funções de atribuição de velocidade, procede-se com a definição dos valores necessários para a plataforma Sinric Pro. Na função principal, o pino responsável pelo LED do WiFi é configurado como saída, bem como os pinos relacionados aos relês. Em seguida, é estabelecido um nível lógico alto para os relês, preparando-os para o estado inicial.

Em seguida, são chamadas as funções responsáveis pela configuração do WiFi e da plataforma Sinric Pro, viabilizando, assim, a correta configuração e estabelecimento das conexões essenciais. Essa etapa crucial pode ser visualizada na Figura 37, que oferece uma representação esquemática dessa sequência de operações, segue na figura 38 o diagrama de bloco correspondente ao circuito feito.

Figura 36 - Setup programa

```

// funcao de configuracao do sinric pro
void setupSinricPro() {
  SinricProFanUS &myFan = SinricPro[FAN_ID];

  myFan.onPowerState(onPowerState);
  myFan.onRangeValue(onRangeValue);
  myFan.onAdjustRangeValue(onAdjustRangeValue);

  // setup SinricPro
  SinricPro.onConnected([](){ Serial.printf("Conectado ao SinricPro\r\n"); });
  SinricPro.onDisconnected([](){ Serial.printf("Desconectado do SinricPro\r\n"); });
  SinricPro.begin(APP_KEY, APP_SECRET);
}

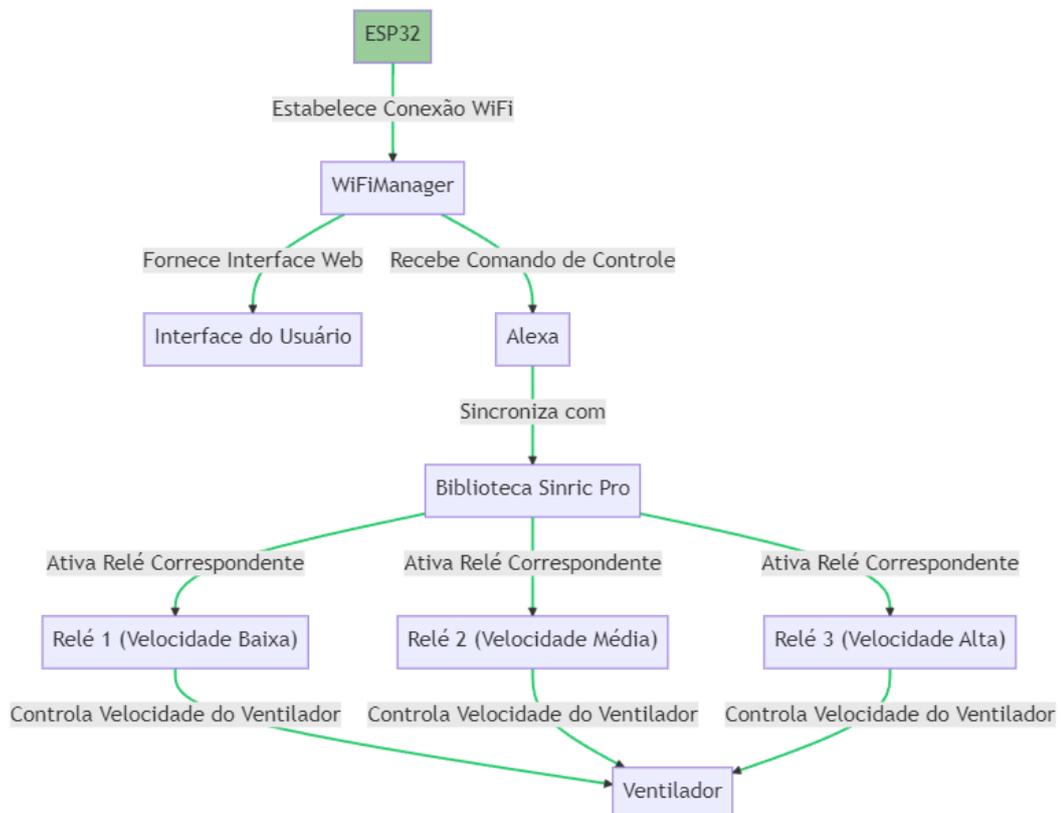
// Funcao de definicao principal do programa
void setup() {
  // setando os pinos como saida
  pinMode(wifiLed, OUTPUT);
  pinMode(RELAYPIN_1,OUTPUT);
  pinMode(RELAYPIN_2,OUTPUT);
  pinMode(RELAYPIN_3,OUTPUT);
  // setando os reles para HIGH (Desligado)
  digitalWrite(RELAYPIN_1,HIGH);
  digitalWrite(RELAYPIN_2,HIGH);
  digitalWrite(RELAYPIN_3,HIGH);
  Serial.begin(BAUD_RATE);
  Serial.printf("\r\n\r\n");
  setupWiFiSTA();
  setupSinricPro();
}

void loop() {
  SinricPro.handle();
}

```

Fonte: (Elaborado pelo autor, 2023)

Figura 37 - Diagrama de bloco projeto



Fonte: (Elaborado pelo autor, 2023)

## 4 Considerações finais

O presente estudo tem como objetivo a integração de um ventilador com a assistente virtual Alexa, a fim de permitir que os usuários controlem as funções básicas, tais como ligar, desligar e alterar a velocidade do aparelho por meio de comandos de voz. Para alcançar esse objetivo, é utilizado um circuito padrão para acionamento de relês, que conectados a um microcontrolador ESP32 e, à internet por meio de uma rede sem fio. A configuração do ventilador será realizada por meio do aplicativo da Amazon Alexa, proporcionando uma experiência fácil e intuitiva aos usuários.

Após a realização de estudos e análises, optou-se pelo uso do microcontrolador ESP32 devido à sua compatibilidade com as funções requeridas, além do baixo custo de aquisição. Uma outra funcionalidade para utilização desse microcontrolador no presente projeto é a utilização de mais dispositivos que podem ser controlados com esse único microcontrolador, o que se pretende implementar num futuro próximo.

A proposta deste trabalho visa ampliar o número de dispositivos disponíveis para automação residencial, oferecendo uma solução inovadora e acessível para os usuários que desejam controlar seus ventiladores de forma remota. É importante ressaltar que, apesar de existirem diversos dispositivos para automação residencial no mercado, a quantidade de opções para controle de ventiladores ainda é limitada, tornando a proposta apresentada neste estudo uma alternativa promissora para atender às necessidades dos consumidores.

Para aplicações futuras deste trabalho, sugere-se a implementação do protocolo *matter*, que é um protocolo padrão para a internet das coisas (IoT) e que permite a interoperabilidade entre dispositivos de diferentes fabricantes e plataformas. Dessa forma, a integração do ventilador com outras assistentes virtuais, como o Google *Assistant* ou a Siri, seria possível, proporcionando maior flexibilidade aos usuários.

Além disso, outra sugestão para aplicação futura deste trabalho seria a criação de um aplicativo próprio para controle do ventilador por meio de conexão Bluetooth, permitindo que os usuários possam controlar o dispositivo mesmo em situações em que não há acesso à internet, como em áreas remotas ou em casos de falhas na conexão Wi-Fi. Com essa solução, seria possível usar o aplicativo para configurar e

controlar o ventilador de forma rápida e fácil, sem a necessidade de conexão com a internet.

Em resumo, a implementação do protocolo *matter* e a criação de um aplicativo próprio para controle pelo Bluetooth são sugestões interessantes para aprimorar e expandir a funcionalidade do sistema proposto neste trabalho, oferecendo aos usuários mais opções e maior flexibilidade para controlar o ventilador de acordo com suas necessidades e preferências, o código fonte está localizado no apêndice A.

O termo de autorização para publicação se encontra no Anexo.

## 5 Referências

*Fitzgerald, Arthur; Kingsley Charles. Máquinas Elétricas, 7th Edition, Porto Alegre, 2014.*

*Fitzgerald, Arthur; Kingsley Charles. Electric Machinery 6th Edition, New York, 2003.*  
Bertulucci, Cristiano, **Motor elétrico Ca: Quais os tipos e como especificar**, 2016  
Disponível em: <<https://www.citisystems.com.br/motor-eletrico/> acesso as: 17:56, 07/09/2022

Universidade Federal do Rio Grande do Norte (UFRN).2011, Disponível em: <<https://materialpublic.imd.ufrn.br/curso/disciplina/1/58/3/19> acesso as:22:00, 15/09/2022

Silveira, Alexandre. **Motores de Indução Monofásicos**, 2020, Disponível em: <<https://recipp.ipp.pt/handle/10400.22/17897> >.

WentLon. **Motor monofásico de indução de polo sombreado**, Disponível em: <<http://wt-micromotor.com/1-shaded-pole-motor.html> acesso as:22:00, 15/09/2022

ABNT NBR, **Máquinas elétricas girantes Parte 2: Motores de indução monofásicos—Requisitos**, 2016, Disponível em: <<http://www2.uesb.br/biblioteca/wp-content/uploads/2022/03/NBR-17094-M%C3%81QUINAS-EL%C3%89TRICAS-GIRANTES-PARTE-2-MOTORES-DE-INDU%C3%87%C3%83O-MONOF%C3%81SICOS-REQUESITOS.pdf> >

Oliveira, Euler. **Conhecendo o NodeMCU-32S ESP32**, 2017, Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32> >

ALLDATASHEET. **Esp 32 Datasheet**, Disponível em: <<https://www.alldatasheet.com/datasheet-pdf/pdf/1243003/ESPRESSIF/ESP32.html>

Oracle. **O que é IOT?** Disponível em: < <https://www.oracle.com/br/internet-of-things/what-is-iot/> >

Tibco. **O que é internet das coisas?**, Disponível em: < <https://www.tibco.com/pt-br/reference-center/what-is-the-internet-of-things-IoT>>

Santos, Carlos; Sales Jefferson. **O Desafio da Privacidade na Internet das Coisas.** Disponível em: <<https://periodicos.ufpe.br/revistas/gestaoorg/article/download/22115/18481>>. Acesso em: 9 de maio. 2016.

Junior, Airton; Moreno, Edward. **Segurança em Infraestrutura para Internet das Coisas.** Disponível em: <<https://periodicos.ufpe.br/revistas/gestaoorg/article/download/22122/18487>>. Acesso em: 9 maio. 2016.

Oliveira, Sergio. **Internet das coisas com ESP8266 Arduino e Raspberry PI**, 2ª edição, 2021.

Gogoni, Ronaldo, **O que é Alexa**, Disponível em:< <https://tecnoblog.net/responde/o-que-e-a-Alexa-ou-melhor-quem-e/> >.

Cipoli, Pedro. **Amazon Alexa echo dot 4ª geração**, 2021, disponível em: < <https://canaltech.com.br/produto/Amazon/echo-dot-4a-geracao/> >.

ESFÉRICA TECNOLOGIA. **Computação em nuvem**, 2020, Disponível em: < <https://www.esferica.com.br/wp-content/uploads/2020/12/ebook-Cloud-Computing-v1.0-Capa.pdf> >.

Baird Colin. **A primer on cloud computing**, 2019, Disponível em: <[https://medium.com/@colinbaird\\_51123/a-primer-on-cloud-computing-9a34e90303c8](https://medium.com/@colinbaird_51123/a-primer-on-cloud-computing-9a34e90303c8) >.

Hi tecnologia. **o que é o protocolo mqtt**, 2022, disponível em: < <https://materiais.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/> >

Last Minute Engineers. **Insight Into ESP32 Sleep Modes & Their Power Consumption**, 2022, Disponível em: < <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/#:~:text=The%20chip%20consumes%20around%200.15,up%20the%20CPU%20whenever%20necessary> >.

Amazon. **what-is-cloud-computing**, 2022, Disponível em: < <https://aws.amazon.com/pt/what-is-cloud-computing/> >

Azure. **What is IaaS**, 2022, Disponível em: < <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-iaas/#overview> >

UFRJ. **Aplicações ZigBee**, 2022, Disponível em: < [https://www.gta.ufrj.br/grad/10\\_1/zigbee/aplicacoes.html](https://www.gta.ufrj.br/grad/10_1/zigbee/aplicacoes.html) >

Samarth, **6LOWPAN- AN IP BASED WIRELESS PROTOCOL**, 2020, Disponível em: < <https://psiborg.in/6lowpan-an-ip-based-wireless-protocol/> >

Qubino. **Z-Wave: 9 Features That Make It Awesome**, 2022, Disponível em: < <https://qubino.com/z-wave-9-features-that-make-it-awesome/> >

Golfalonieri. **How Amazon Alexa works? Your guide to Natural Language Processing (AI)**, 2018, Disponível em: < <https://towardsdatascience.com/how-Amazon-Alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3> >

VRTECH. **O que é Z-Wave?**, 2021, disponível em: < <https://sindcontvr.org.br/o-que-e-z-wave/> >

Agência New Voice. **Automação residencial vai atingir US\$ 163 bi até 2028**, disponível em: < <https://newvoice.ai/2022/01/21/automacao-residencial-vai-atingir-us-163-bi-ate-2028/> >

Wikilivros. **Sistemas operacionais / Sistemas embarcados**, 2020, disponível em < [https://pt.wikibooks.org/wiki/Sistemas\\_operacionais/Sistemas\\_embarcados](https://pt.wikibooks.org/wiki/Sistemas_operacionais/Sistemas_embarcados) >

Brazilian RTOS blog, **BRTOS reference manual**, 2013, disponível em < <https://brtosblog.wordpress.com/> >

Kofuji, Sergio; Rehder, Gustavo; Seabra, Antonio. **Arquitetura de Sistemas Embarcados**, 2017, disponível em < [https://edisciplinas.usp.br/pluginfile.php/2855105/mod\\_resource/content/1/PSI3441-A01.pdf](https://edisciplinas.usp.br/pluginfile.php/2855105/mod_resource/content/1/PSI3441-A01.pdf) >

Tadeu, Erivelto. **Use o smartphone como centro de controle da Internet das Coisas**, 2016, disponível em < <https://itforum.com.br/noticias/use-o-smartphone-como-centro-de-controle-da-internet-das-coisas/> >

Pereira, Ivaneide. **Estudo sobre harmônicos gerado pelo motor universal**, 2018, disponível em < [https://repositorio.ufersa.edu.br/bitstream/prefix/3607/2/IvaneideCP\\_MONO.pdf](https://repositorio.ufersa.edu.br/bitstream/prefix/3607/2/IvaneideCP_MONO.pdf) >

Bibliography

FAHAD. **Sinric Pro ESP32 for Home Automation and Sensor Monitoring**. Disponível em: <<https://www.electronicclinic.com/sinric-pro-esp32-for-home-automation-and-sensor-monitoring/>>. Acesso em: 01 maio 2023.

**SinricPro IOT cloud platform documentation**. Disponível em: <<https://help.sinric.pro/>>. Acesso em: 01 maio.2023.

Handson Technology. **4 Channel 5V Optical Isolated Relay Module**. Disponível em: <<https://www.handsontec.com/dataspecs/4Ch-relay.pdf>>. Acesso em: 16 maio. 2023

TAYLOR, A. **O que são WebSockets e Como Criá-los?**, 2022, Disponível em: <<https://appmaster.io/pt/blog/o-que-sao-websockets-e-como-cria-los>>. Acesso em: 21 maio. 2023.

## APÊNDICES

### APÊNDICE A – Código Fonte

Arquivo – Fan\_tcc.ino

```
#ifndef ENABLE_DEBUG
    #define DEBUG_ESP_PORT Serial
    #define NODEBUG_WEBSOCKETS
    #define NDEBUG
#endif

#include <Arduino.h>
#include <WiFi.h>
#include <WiFiManager.h>

#include "SinricPro.h"
#include "SinricProFanUS.h"

#define wifiLed 2 // Pino LED wifi
//-----PINOS DE OUTPUT-----
#define RELAYPIN_1 25 //
#define RELAYPIN_2 26
#define RELAYPIN_3 27
//-----Wifi--CONFIG----- *CASO UTILIZE A
FUNCAO WifiSetup*
#define WIFI_SSID "INTELBRAS"
#define WIFI_PASS "12121980"
//-----CONFIGURACOES DO SINRIC PRO-----
-----
#define APP_KEY "b563c127-ef54-4ffe-b7e5-35c98ce9bb2b" // chave do
apap
#define APP_SECRET "7c53ee25-51ea-4d33-bc36-00b232053345-084d01d1-
27a2-428c-a46d-137e28141a37" // Codigo secreto do app
#define FAN_ID "64569cf5743f912070124775" // ID do Fan
#define BAUD_RATE 9600
//-----
```

```

// estrutura para armazenar os valores do Fan
// Velocidade de 1 a 3
struct {
    bool powerState = false;
    int fanSpeed = 1;
} device_state;

bool onPowerState(const String &deviceId, bool &state) {
    Serial.printf("O fan %s\r\n", state?"Ligou":"Desligou");
    device_state.powerState = state;
    if(device_state.powerState == false) // Verifica se o dispositivo esta desligado ele
desativa os relês
    {
        digitalWrite(RELAYPIN_1,HIGH);
        digitalWrite(RELAYPIN_2,HIGH);
        digitalWrite(RELAYPIN_3,HIGH);
    }
    if(device_state.powerState == true) // Caso o dispositivo ligue ele sempre começa na
velocidade 1
    {
        digitalWrite(RELAYPIN_1,LOW);
        digitalWrite(RELAYPIN_2,HIGH);
        digitalWrite(RELAYPIN_3,HIGH);
    }

    return true;
}

// A velocidade pode ser definida de 1 a 3
bool onRangeValue(const String &deviceId, int& rangeValue) {
    device_state.fanSpeed = rangeValue;
    Serial.printf("Velocidade do fan mudada para %d\r\n", device_state.fanSpeed);
    if(device_state.fanSpeed == 1) // Se a velocidade for a 1 ele desativa os outros relês
e deixa ativado somente o primeiro

```

```

{
  digitalWrite(RELAYPIN_1,LOW);
  digitalWrite(RELAYPIN_2,HIGH);
  digitalWrite(RELAYPIN_3,HIGH);
}
if(device_state.fanSpeed == 2) // Se a velocidade for a 2 ele desativa os outros relês
e deixa ativado somente o segundo
{
  digitalWrite(RELAYPIN_2,LOW);
  digitalWrite(RELAYPIN_1,HIGH);
  digitalWrite(RELAYPIN_3,HIGH);
}
if(device_state.fanSpeed == 3) //Se a velocidade for a 3 ele desativa os outros relês
e deixa ativado somente o terceiro
{
  digitalWrite(RELAYPIN_2,HIGH);
  digitalWrite(RELAYPIN_1,HIGH);
  digitalWrite(RELAYPIN_3,LOW);
}
return true;
}

// Fan rangeValueDelta is from -3..+3
bool onAdjustRangeValue(const String &deviceId, int& rangeValueDelta) {
  device_state.fanSpeed += rangeValueDelta;
  Serial.printf("A velocidade do ventilador mudou aproximadamente %i to %d\r\n",
rangeValueDelta, device_state.fanSpeed);

  rangeValueDelta = device_state.fanSpeed; // retorna o valor absoluto da velocidade
  return true;
}

// Criando uma rede Wireless para conexão
void setupWiFiSTA() {
  WiFi.mode(WIFI_STA);

```

```

digitalWrite(wifiLed, LOW);
bool res;
WiFiManager wm;
wm.resetSettings();
res = wm.autoConnect("VentiladorESP","12345678");
Serial.printf("\r\n[Wifi]: Conectando");
    Serial.printf("\r\n[Wifi]: Conectado à rede %s\r\n[Wifi]: IP-Address é %s\r\n",
WiFi.SSID().c_str(), WiFi.localIP().toString().c_str());
    digitalWrite(wifiLed, HIGH);
}
// Setando wifi
void setupWiFi() {
    Serial.printf("\r\n[Wifi]: Conectando");
    WiFi.begin(WIFI_SSID, WIFI_PASS);

    while (WiFi.status() != WL_CONNECTED) {
        Serial.printf(".");
        delay(250);
    }
    digitalWrite(wifiLed, HIGH);
        Serial.printf("conectado!\r\n[Wifi]:      IP-Address      é      %s\r\n",
WiFi.localIP().toString().c_str());
}
// funcao de configuracao do Sinric Pro
void setupSinricPro() {
    SinricProFanUS &myFan = SinricPro[FAN_ID];

    myFan.onPowerState(onPowerState);
    myFan.onRangeValue(onRangeValue);
    myFan.onAdjustRangeValue(onAdjustRangeValue);

    // setup SinricPro

```

```
SinricPro.onConnected([]){ Serial.printf("Conectado ao SinricPro\r\n"); });
SinricPro.onDisconnected([]){ Serial.printf("Desconectado do SinricPro\r\n"); });
SinricPro.begin(APP_KEY, APP_SECRET);
}

// Funcao de definicao principal do programa
void setup() {
  // setando os pinos como saida
  pinMode(wifiLed, OUTPUT);
  pinMode(RELAYPIN_1,OUTPUT);
  pinMode(RELAYPIN_2,OUTPUT);
  pinMode(RELAYPIN_3,OUTPUT);
  // setando os relês para HIGH (Desligado)
  digitalWrite(RELAYPIN_1,HIGH);
  digitalWrite(RELAYPIN_2,HIGH);
  digitalWrite(RELAYPIN_3,HIGH);
  Serial.begin(BAUD_RATE);
  Serial.printf("\r\n\r\n");
  setupWiFiSTA();
  setupSinricPro();
}

void loop() {
  SinricPro.handle();
}
```

## ANEXO

## ANEXO A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADEMICA



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
GABINETE DO REITOR

Av. Universitária, 1069 • Setor Universitário  
Caixa Postal 86 • CEP 74605-010  
Goiânia • Goiás • Brasil  
Fone: (62) 3946.1000  
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

## RESOLUÇÃO nº 038/2020 – CEPE

## ANEXO I

## APÊNDICE ao TCC

## Termo de autorização de publicação de produção acadêmica

O(A) estudante Spão Victor Banezyk Rodrigues  
do Curso de Engenharia de computação, matrícula 20181003300562,  
telefone: 62 991862866 e-mail JK081200@GMAIL.COM,  
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Automatização de um ventilador com Alexa,  
gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 28 de fevereiro de 2023.

Assinatura do autor: Spão Banezyk

Nome completo do autor: Spão Victor Banezyk Rodrigues

Assinatura do professor-orientador: [Assinatura]

Nome completo do professor-orientador: Marcelo Antonio Adad de Araújo