

**FUNDAÇÃO OSWALDO ARANHA  
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
TRABALHO DE CONCLUSÃO DE CURSO**

**ARTHUR SOARES TEIXEIRA  
CÍNTIA DE OLIVEIRA CELESTINO ELIAS  
MARIANA DE OLIVEIRA FERREIRA**

**INTERNET DAS COISAS – UMA ABORDAGEM DIRECIONADA A  
EDIFÍCIOS INTELIGENTES**

**VOLTA REDONDA  
2018**

**FUNDAÇÃO OSWALDO ARANHA  
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
TRABALHO DE CONCLUSÃO DE CURSO**

**INTERNET DAS COISAS – UMA ABORDAGEM DIRECIONADA A  
EDIFÍCIOS INTELIGENTES**

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Elétrica do UniFOA como requisito à  
obtenção do título de Engenheiros  
Eletricistas.

Alunos:  
Arthur Soares Teixeira  
Cíntia de Oliveira Celestino Elias  
Mariana de Oliveira Ferreira

Orientador:  
Prof. M. Sc. Cláudio Márcio de Freitas  
da Silva

Coorientador:  
Prof. Esp. Aloano Régio de Almeida  
Pereira



Fundação Oswaldo Aranha



## FOLHA DE APROVAÇÃO

**Curso:** Engenharia Elétrica

**Acadêmica: Matrícula:** Arthur Soares Teixeira 201410095; Cintia de Oliveira Celestino Elias 201410098; Mariana de Oliveira Ferreira 201410108

**Título do TCC:** Internet das coisas – Uma abordagem direcionada a edifícios inteligentes

Apresentado publicamente perante a Banca Avaliadora, como parte dos requisitos para conclusão do Curso de Engenharia Elétrica

Aprovada em .....10..... de .....setembro..... de .....2018.....

**Banca Avaliadora:**

Professor Orientador  
Cláudio Márcio de Freitas da Silva, Mestre, UniFOA

Professor Avaliador  
Mauricio Ferreira Haddad, Mestre, UniFOA

Professor Avaliador  
Aloano Regio de Almeida Pereira, Especialista, UniFOA

Centro Universitário UniFOA - Volta Redonda  
R. São João, 100 - Vila Rica - RJ - 27260-000  
Tel: (24) 3341-1111

Centro Universitário UniFOA - Volta Redonda  
Rua 24 - 311 - Volta Redonda  
27260-000 - Volta Redonda - RJ  
Tel: (24) 3341-1111

Centro UniFOA  
R. Lúcia Leopoldina - 40 - Volta Redonda  
27260-000 - Volta Redonda - RJ  
Tel: (24) 3341-1111

Centro UniFOA  
R. Nova América - 271 - Volta Redonda  
27260-000 - Volta Redonda - RJ  
Tel: (24) 3341-1111

Centro Universitário UniFOA - Volta Redonda  
R. São João, 100 - Vila Rica - RJ  
27260-000 - Volta Redonda - RJ  
Tel: (24) 3341-1111

Centro Universitário UniFOA - Volta Redonda  
R. São João, 100 - Vila Rica - RJ  
27260-000 - Volta Redonda - RJ  
Tel: (24) 3341-1111

*"If you can dream it, you can do it."*

Walt Disney

## **AGRADECIMENTOS**

Acreditamos que sozinhos não é possível alcançar sonhos nem os melhores objetivos. Por este motivo, gostaríamos de agradecer a todos que nos acompanharam durante esta caminhada.

Agradecemos primeiramente a Deus, que, com sua misericórdia e bondade, nos deu saúde e forças para que esta conquista pudesse ser alcançada.

Aos nossos pais, que foram peças fundamentais para a nossa educação e que nos deram todo o suporte necessário; aos nossos familiares e amigos, que sempre estiveram por perto quando precisamos de apoio.

Aos mestres, por toda a dedicação, sabedoria e ensinamentos ministrados durante o curso. Em especial, ao nosso orientador Claudio Marcio de Freitas da Silva e nosso coorientador Aloano Régio de Almeida Pereira, pela atenção, paciência, instruções e direcionamentos que tornaram possível a realização deste trabalho.

À instituição UniFOA e a todos os profissionais que contribuem diretamente ou indiretamente para a manutenção e o exercício da formação educacional de seus alunos.

## RESUMO

O crescente uso da internet e seu acesso a partir de diferentes dispositivos eletrônicos vêm transformando o mercado atual e proporcionando o desenvolvimento de novas tecnologias. Neste contexto, surge a Internet das Coisas (IoT), que promete gerar uma revolução tecnológica no cotidiano das pessoas, buscando trazer eficiência e praticidade. Paralelamente, existem os Edifícios Inteligentes, que buscam aplicar a IoT para correlacionar seus serviços, otimizar os processos e oferecer um ambiente prático e seguro aos seus usuários. Pensando nisso, o presente trabalho propõe realizar um estudo referente a estes conceitos e elaborar um protótipo a fim de demonstrar na prática a sua real aplicação. Este protótipo tem como objetivo principal acionar cargas, monitorar variáveis e detectar anomalias presentes em um ambiente. Todas essas funções poderão ser executadas ou visualizadas via *dashboard*, utilizando para isso qualquer dispositivo que contenha conexão com a internet e o *login* de acesso para autorização. Seu controle poderá ser feito em tempo real e remotamente. Além disso, algumas informações poderão ser notificadas via e-mail para o cliente cadastrado, em casos de incidentes no sistema. Para a elaboração do protótipo foi necessário utilizar o microcontrolador ESP32, serviços na nuvem, plataforma Node-Red e protocolo MQTT. Ao longo do texto, serão demonstrados os resultados obtidos, as perspectivas de melhorias futuras para o projeto e a sua utilidade para os diferentes tipos de edifícios existentes.

**Palavras-chave:** Internet das coisas. Edifícios inteligentes. ESP32. Node-RED.

## SUMÁRIO

RESUMO .....	6
1 INTRODUÇÃO .....	14
1.1 Justificativa .....	15
1.2 Objetivos .....	17
1.2.1 Objetivo geral .....	17
1.2.2 Objetivos específicos .....	17
1.3 Metodologia .....	18
2 EDIFÍCIOS INTELIGENTES .....	19
2.1 História e Conceitos .....	19
2.2 Integração em Edifícios Inteligentes .....	21
2.3 Serviços em Edifícios Inteligentes .....	22
2.4 Exemplos de Edifícios Inteligentes .....	23
2.4.1 Eldorado <i>Business Tower</i> – São Paulo, Brasil .....	24
2.4.2 Ventura <i>Corporate Towers</i> – Rio de Janeiro, Brasil .....	25
3 INTERNET DAS COISAS OU INTERNET OF THINGS (IoT) .....	28
3.1 Histórico .....	28
3.2 Definição .....	29
3.3 Blocos Básicos da Construção IoT .....	30
3.4 Aplicação .....	31
3.5 Atuação da IoT em Edifícios Inteligentes .....	33
3.6 Desafios para a Implantação da IoT .....	34
4 CONSUMO DE ENERGIA .....	36
4.1 Tensão Elétrica .....	38
4.2 Corrente Elétrica .....	40
5 PLATAFORMA, PROTOCOLO E SERVIÇOS UTILIZADOS .....	43
5.1 IBM Cloud .....	43
5.2 IBM Watson .....	44
5.3 MQTT ou <i>Message Queuing Telemetry Transport</i> .....	46
5.4 Node-RED .....	48
5.4.1 Estrutura de programação do Node-RED .....	49
5.4.2 Tipos de nós existentes no Node-RED .....	50

5.5	<i>Dashboard</i> .....	50
5.6	Arduino .....	51
5.6.1	Estrutura de programação do Arduino (IDE) .....	52
6	COMPONENTES DO SISTEMA .....	55
6.1	Node MCU-32S ESP32 .....	55
6.2	Sensores .....	57
6.2.1	Temperatura e umidade .....	58
6.2.2	Sensor de gás .....	59
6.2.3	Sensor de fogo .....	62
6.2.4	Sensor de tensão .....	64
6.2.5	Sensor de corrente .....	66
6.3	Atuador .....	68
6.3.1	Módulo relé .....	69
7	ETAPAS DE DESENVOLVIMENTO DO PROJETO .....	72
7.1	Montagem do Circuito Físico .....	73
7.1.1	Testes nos sensores .....	73
7.1.1.1	Sensor de tensão .....	73
7.1.1.2	Sensor de corrente .....	75
7.1.1.3	Sensor de temperatura e umidade .....	79
7.1.1.4	Sensor de gás .....	81
7.1.1.5	Sensor de fogo .....	83
7.1.2	Disposição dos componentes .....	86
7.1.3	Conexão dos dispositivos ao microcontrolador .....	87
7.2	Programação na IDE do Arduino .....	88
7.3	Integração do Servidor .....	91
7.3.1	Serviço de Internet das Coisas no IBM Cloud .....	91
7.3.2	Cadastrando o ESP32 na plataforma de IoT do IBM Cloud .....	93
7.4	Programação no Node-Red .....	96
7.5	Protótipo Finalizado .....	99
7.6	Análise dos Resultados .....	100
8	APLICABILIDADE DO PROTÓTIPO .....	102
8.1	Monitoramento da Temperatura e Umidade .....	102
8.2	Monitoramento da Concentração de Gás .....	103

8.3	Monitoramento da Presença de Fogo .....	103
8.4	Consumo de Energia Elétrica.....	104
9	CONCLUSÃO.....	106
10	REFERENCIAS BIBLIOGRÁFICAS .....	108
	APÊNDICE A .....	114
	ANEXO I – ORÇAMENTO DO PROTÓTIPO .....	121
	ANEXO II – DATASHEET DO ESP32.....	122
	ANEXO III – DATASHEET DO SENSOR DE TEMPERATURA E UMIDADE .....	126
	ANEXO IV – DATASHEET DO SENSOR DE GÁS .....	128
	ANEXO V – DATASHEET DO SENSOR DE CORRENTE .....	129
	ANEXO VI – DATASHEET DO SENSOR DE TENSÃO .....	130

## LISTA DE FIGURAS

Figura 1 – Áreas de concentração para a Internet das Coisas. ....	16
Figura 2 – Eldorado Business Tower. ....	24
Figura 3 – Ventura Corporate Towers. ....	26
Figura 4 – Relação da população mundial com a quantidade de dispositivos conectados. ....	29
Figura 5 – Blocos básicos da construção IoT. ....	30
Figura 6 – Algumas atuações da IoT. ....	32
Figura 7 – Imagem ilustrativa sobre IoT. ....	34
Figura 8 – Estrutura do consumo brasileiro de eletricidade na rede, por classe (%). 37	
Figura 9 – Modelo de ENCE (Etiqueta Nacional de Conservação de Energia) - PBE Edifica. ....	37
Figura 10 – Ciclo de uma onda alternada com indicação de seus parâmetros. ....	39
Figura 11 – Esquema de circuito elétrico fechado. ....	41
Figura 12 – Comparativo de ondas de corrente e tensão. ....	42
Figura 13 – Console do IBM Cloud. ....	44
Figura 14 – Exemplo de aplicação do protocolo MQTT. ....	47
Figura 15 – Modelo de <i>publish/subscribe</i> para sensores IoT. ....	48
Figura 16 – Estrutura do Node-RED. ....	49
Figura 17 – Nós existentes no Node-RED. ....	50
Figura 18 – Exemplo de dashboard. ....	50
Figura 19 – Ambiente de desenvolvimento integrado do Arduino. ....	52
Figura 20 – Node MCU-32S ESP32. ....	55
Figura 21 – Pinagem do Node MCU-32S ESP32. ....	56
Figura 22 – Sensor DHT11. ....	58
Figura 23 – Pinos do sensor DHT11. ....	59
Figura 24 – Esquema interno de um sensor de gás MQ. ....	60
Figura 25 – Encapsulamento de um sensor de gás MQ. ....	60
Figura 26 – Família de sensores de gás MQ. ....	61
Figura 27 – Sensor MQ-2 - parte inferior. ....	62
Figura 28 – Sensor Chama Fogo 760 a 1100 nm. ....	63
Figura 29 – Sensor Chama Fogo 760 a 1100 nm - parte inferior. ....	63
Figura 30 – Sensor de tensão AC 0 a 250Vac ZMPT101B. ....	64

Figura 31 – Sensor de tensão AC 0 a 250Vac ZMPT101B (pinagem).....	65
Figura 32 – Transformadores de corrente.....	66
Figura 33 – Sensor de corrente SCT-013-020.....	67
Figura 34 – Esquema para exemplificar o SCT-013.....	68
Figura 35 – Esquema de ligação interna de um relé.....	69
Figura 36 – Módulo relé de 2 canais.....	70
Figura 37 – Pinagem do módulo relé de 2 canais.....	71
Figura 38 – Diagrama de blocos das etapas do projeto.....	72
Figura 39 – Teste do sensor de tensão.....	74
Figura 40 – Comparação dos valores medidos pelo sensor de tensão e multímetro. .....	75
Figura 41 – Circuito auxiliar para a utilização do sensor de corrente.....	76
Figura 42 – Ligação do sensor de corrente com o circuito final.....	76
Figura 43 – Sobreposição de formas de onda.....	77
Figura 44 – Forma de conexão correta entre o sensor de corrente e o condutor. ....	77
Figura 45 – Teste do sensor de corrente.....	78
Figura 46 – Comparação dos valores medidos pelo sensor de corrente e alicate amperímetro.....	79
Figura 47 – Teste do sensor de temperatura e umidade - Circuito montado na <i>protoboard</i> . ....	80
Figura 48 – Comparação dos valores de temperatura.....	80
Figura 49 – Teste do sensor de gás - Circuito montado na <i>protoboard</i> .....	81
Figura 50 – Teste no sensor de gás - LED vermelho acesso: Concentração acima de 3.000 ppm de gás/fumaça detectado. ....	82
Figura 51 – Leitura das saídas do sensor de gás.....	83
Figura 52 – Teste do sensor de fogo - Circuito montado na <i>protoboard</i> .....	84
Figura 53 – Teste no sensor de gás - LED vermelho acesso: Fogo detectado.....	85
Figura 54 – Leitura da saída do sensor de fogo.....	85
Figura 55 – Placa 1 - Lâmpada, interruptor e tomada.....	86
Figura 56 – Placa 2 - Sensores de temperatura e umidade, fogo e gás.....	87
Figura 57 – Caixa com o circuito devidamente montado nas placas de fenolite.....	88
Figura 58 – Tela inicial da IBM Cloud.....	91
Figura 59 – Criando uma conta no IBMid.....	92
Figura 60 – Criando uma conta no IBM Cloud.....	93

Figura 61 – Cadastrando os dispositivos na plataforma IoT.....	93
Figura 62 – Preenchimento dos dados para identificação de dispositivo.....	94
Figura 63 – Chave token do dispositivo cadastrado. ....	95
Figura 64 – Chave API criada. ....	95
Figura 65 – <i>Flow</i> criado no Node-RED.....	97
Figura 66 – Protótipo finalizado.....	99
Figura 67 – <i>Dashboard</i> criada.....	100
Figura 68 – Cálculo da tensão, corrente e potência de uma carga (ferro elétrico)..	101

## LISTA DE TABELAS

Tabela 1 .....	66
----------------	----

## LISTA DE SIGLAS

A – Ampère  
API – Application Programming Interface  
FP – Fator de Potência  
IBM – International Business Machines  
IoT – Internet of Things  
IP – Internet Protocol  
LED – Light Emitting Diode  
M2M – Machine to Machine  
MQTT – Message Queuing Telemetry Transport  
NA – Normal Aberto  
NF – Normal Fechado  
PBE – Programa Brasileiro de Etiquetagem  
ppm – Partes por milhão  
SI – Sistema Internacional de Unidades  
TCP – Transmission Control Protocol  
V – Volt  
W – Watt  
Web – World Wide Web  
Wi-Fi – Wireless Fidelity

## 1 INTRODUÇÃO

A internet tornou-se parte fundamental da sociedade. Na concepção de SATO (2015), a rápida popularização de tecnologias como a internet e seu acesso a partir de dispositivos móveis pode ser apontada como a parte mais visível das transformações do cenário dos meios de comunicação. Nesse contexto, pode-se afirmar que há cada vez mais espaço e necessidade do mercado para manifestação de ideias, surgimento de conceitos e teorias, bem como investimento em novas tecnologias. Dentre elas, pode-se citar a Internet das Coisas, também conhecida como IoT (*Internet of Things*), que está promovendo uma revolução tecnológica no mundo atual.

De forma simples, a Internet das Coisas permite que objetos conectados através da rede possam se comunicar e compartilhar informações para facilitar ou criar diversas ações. Segundo Rafael Steinhauser, vice-presidente sênior da Qualcomm para a América Latina, quando interligamos os objetos que nos rodeiam, tudo muda, tudo fica mais inteligente, eficiente e controlável. (STEINHAUSER, 2015).

Vale ressaltar que, para que haja a interconexão dos dispositivos, a comunicação é a parte central da internet das coisas. Tecnologias de rede habilitam determinados dispositivos a comunicar-se com outros dispositivos, aplicações e serviços que estão sendo processados na nuvem. A internet depende de protocolos padronizados para garantir que a comunicação entre dispositivos diferentes ocorra com segurança e confiabilidade. Estes especificam as regras e formatos utilizados para estabilizar e gerenciar as redes, assim como a transmissão de informação entre as redes.

O termo Internet das Coisas consiste na ideia de que tudo pode ser conectado à internet. Em uma escala mais ampla, há aplicações para diferentes tipos de dispositivos utilizados no cotidiano das pessoas, que podem ser tanto eletrodomésticos ou bens de consumo, como também destinadas a diferentes tipos de setores, sejam eles na área de saúde, meio ambiente, segurança, construção civil etc. Tendo em vista que o setor de edificações está em constante desenvolvimento, o tema deste trabalho é voltado para a aplicabilidade da IoT em edifícios, tanto comerciais, como industriais ou residências.

A escolha deste foco deve-se ao fato de que o ramo da construção civil está crescendo cada vez mais e vem tornando-se tema de extrema importância nos meios científicos e acadêmicos em virtude de representar uma maior demanda em consumo de energia. Estima-se que, nos próximos anos, o mundo deverá construir 230 bilhões de metros quadrados nos próximos 40 anos.

Além disso, sabe-se também que o futuro das cidades está na sustentabilidade. Isso significa que as edificações terão que evoluir para melhorar a qualidade de vida da população e sua relação com o meio ambiente. Para isso, é de extrema importância investir em uma vasta gama de fatores, como sistemas inteligentes e interconectados de gestão, eficiência energética, mobilidade urbana, gestão de resíduos, acessibilidade etc.

Pensando na união destes fatores, o presente trabalho propõe realizar um estudo sobre a internet das coisas aplicada em edifícios inteligentes, bem como demonstrar através da elaboração de um protótipo as possíveis variáveis que se possa controlar em um ambiente e as tecnologias utilizadas para a realização do mesmo.

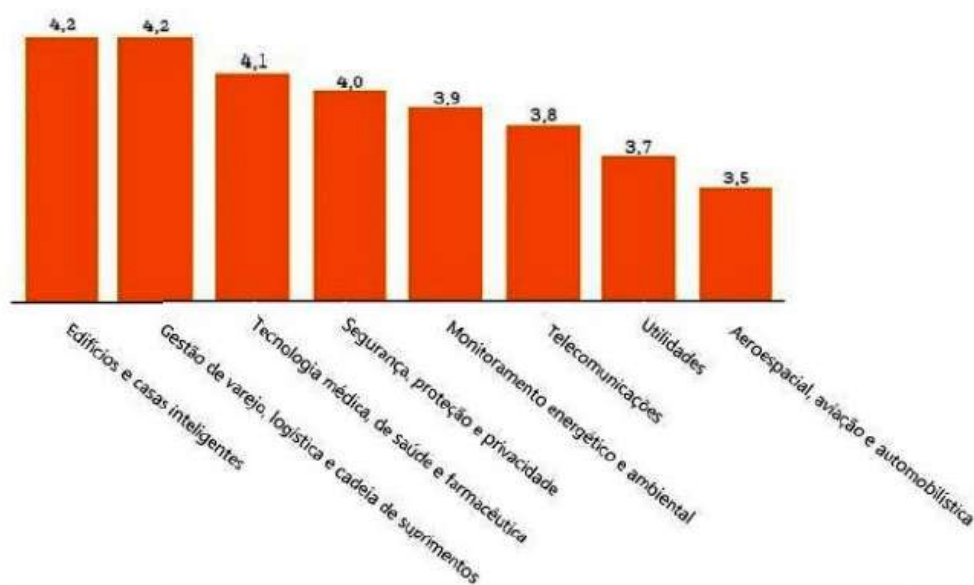
### **1.1 Justificativa**

O interesse em IoT por parte das empresas, organizações e governos tem crescido rapidamente nos últimos anos. MANYIKA et al. (2013) diz que:

O instituto McKinsey, em seu relatório de 2013 (*Disruptive Technologies*), identificou doze tecnologias que, até 2025, terão um impacto econômico massivo. A Internet das Coisas destacou-se como a tecnologia que terá o maior impacto, alcançando 37 trilhões de dólares em valor econômico.

Segundo WU et al. (2011), não é possível representar de maneira exaustiva todas as alternativas de soluções atribuídas à Internet das Coisas. Isto porque o processo de compreensão do seu potencial ainda encontra-se em fase inicial, tendo em vista a variedade de aplicações que podem ser endereçadas em diversos segmentos do mercado.

A Figura 1 contém um exemplo de concentração proporcional por área de aplicações para a Internet das Coisas, no qual “1” = insignificante e “5” = muito importante.



**Figura 1** – Áreas de concentração para a Internet das Coisas.  
(Fonte: LACERDA, 2015)

Através do gráfico, é possível constatar que a Internet das Coisas tem sido de grande importância para o conceito de edifícios inteligentes (*Smart Building*), tornando possível que os equipamentos e recursos funcionem como peças integradas, estando ligados uns aos outros para serem capazes de comunicar-se máquina a máquina.

Há de se considerar também que, segundo o Relatório Status Global 2017, os edifícios são considerados os maiores consumidores de energia em escala mundial, visto que representam cerca de 30% do consumo final mundial de energia e 28% da parcela de emissões de gás carbônico para a atmosfera. Acredita-se que esses números irão aumentar e, em consequência dessa expectativa, é necessário investir em medidas preventivas de eficiência energética, sustentabilidade e tecnologia.

Em busca de um melhor aproveitamento do espaço de forma eficiente e racional, os novos modelos de construção visam trazer inovações tecnológicas e automação predial. Por meio da aplicação da Internet das Coisas e outras tendências que estão surgindo a todo o momento no mercado, será possível coletar dados de forma aprimorada que irão proporcionar melhores decisões de negócios, aumento na produtividade e eficiência dos funcionários desses locais, além de um melhor aproveitamento de toda a infraestrutura, minimizando com isso os custos.

A IoT permite que sistemas operacionais transmitam informações mais precisas e úteis para fornecer as melhores experiências para os inquilinos e está promovendo transformações que devem impactar diversos segmentos. A chegada de soluções com a utilização de alta tecnologia visando o ganho de produtividade, segurança e prevenção de riscos é coisa do presente. De acordo com um relatório realizado pela consultoria americana MarketsandMarkets em 2017, os investimentos no mercado de edifícios inteligentes devem crescer a uma taxa anual composta de 33,7%, chegando a U\$31,74 bilhões até 2022.

Em virtude do que já foi exposto, o presente trabalho visa elaborar um estudo teórico sobre os conceitos de Internet das Coisas e seu funcionamento aplicado a edifícios inteligentes, bem como colocar na prática um modelo (protótipo) a fim de demonstrar algumas possibilidades existentes na utilizações de sensores, com o auxílio do *hardware* Node MCU-32S ESP32 e da plataforma Node-RED integrados a uma interface *web*.

Como se trata de uma solução de baixo custo e de fácil instalação, a tecnologia proposta é uma alternativa a empresas ou instituições que desejam manter-se atualizadas do ponto de vista tecnológico, assim como minimizar os desperdícios com energia elétrica.

## **1.2 Objetivos**

### **1.2.1 Objetivo geral**

O objetivo geral do presente trabalho acadêmico é realizar um estudo sobre Internet das Coisas e sua aplicação para os edifícios inteligentes, avaliando questões relacionadas à otimização de processos manuais, praticidade, facilidade e conforto aos usuários dos edifícios, além da economia de energia elétrica.

### **1.2.2 Objetivos específicos**

- Elaborar uma revisão bibliográfica sobre o tema abordado (Internet das Coisas e edifícios inteligentes);

- Elaborar um modelo (protótipo) a fim de demonstrar possíveis aplicações envolvendo o conceito de Internet das Coisas para os edifícios inteligentes, mediante o auxílio de sensores (temperatura e umidade, gás, fogo, corrente e tensão), atuadores (relés), *hardware* Node MCU-32S ESP32, *software* Arduino IDE e serviços da IBM Watson com a plataforma Node-RED. Todo o processo será monitorado e/ou controlado através de uma interface *web*, cujo acesso é feito através de qualquer dispositivo conectado à internet que possua o *link* da *dashboard* com seu respectivo *login* e senha;

- Verificar a medição do consumo de energia de uma carga qualquer;

- Realizar uma análise sobre os resultados obtidos após a conclusão do protótipo.

### **1.3 Metodologia**

- Revisão bibliográfica de trabalhos: leitura de artigos, livros e pesquisas acadêmicas relacionadas ao desenvolvimento de tecnologias IoT e suas principais aplicações em edifícios;

- Desenvolvimento do protótipo: a partir da pesquisa bibliográfica e com base na tecnologia selecionada, serão estruturadas etapas para o desenvolvimento do protótipo, desde seu conceito até sua implementação;

- Realização de testes do protótipo para verificar seu funcionamento;

- Análise de resultados do protótipo para identificar os pontos positivos e necessidades de melhoria, bem como diretrizes para eventuais ajustes.

## 2 EDIFÍCIOS INTELIGENTES

### 2.1 História e Conceitos

O surgimento das edificações está intimamente relacionado com a existência da civilização humana, onde as habitações eram consideradas formas de sobrevivência do homem. Convém destacar que construir moradias mais seguras e confortáveis sempre foi uma necessidade e também um símbolo de status e poder, visto que as edificações expressam a engenhosidade humana.

Com os avanços da sociedade, os edifícios passaram a se tornar grandes centros de negócio, onde grupos de pessoas se reuniam e trocavam informações, ao mesmo tempo em que buscavam estabelecer entre elas relações de colaborações formais e informais. Para completar o que foi mencionado acima, Pádua (2006, p. 24) descreve da seguinte forma: “Os edifícios tornaram-se o centro das atividades de negócios e de prestação de serviços, constituindo-se na base da vida urbana pós-moderna”.

Paralelamente a isso, na década posterior a Segunda Guerra Mundial, novas tecnologias e sistemas técnicos vieram a ser incorporados nos edifícios em decorrência do desenvolvimento das áreas da informática, da automação, do surgimento dos microprocessadores, dos meios de comunicação e dos sistemas de informação.

Entretanto, o surgimento da crise do petróleo em 1970 proporcionou uma mudança no cenário dessas edificações, onde era necessário que novas ideias existissem com o intuito de proporcionar um melhor aproveitamento do espaço de forma eficiente e racional. Ou seja, os novos modelos de construção ao mesmo tempo em que buscavam trazer inovações na área de automação predial e conceitos atuais para a indústria de construção civil, precisavam também criar prédios mais eficientes, econômicos e sustentáveis minimizando assim a complexidade, o custo e o luxo que essas edificações representavam naquela época.

Surge, então, em meados da década de 80, numa instituição conhecida como *Intelligent Building Institute* (IBI) de Washington, nos Estados Unidos, o termo Edifícios Inteligentes ou *Smart Building*. Este contém atualmente diversos conceitos

difundidos de forma descentralizada que procuram definir e explicar o funcionamento dessas novas metodologias de edificações.

Segundo o *Smart Building Institute* (2015, s.p.) é definido que:

Um edifício concebido para ser inteligente deve aumentar sua performance e facilitar as operações de manutenção durante seu ciclo de vida. O objetivo primário dessa construção é minimizar os custos de longo prazo sobre o seu ciclo de vida para os proprietários, os ocupantes e o meio ambiente. Em um prédio de alta performance todos os componentes do edifício são integrados de forma a trabalharem juntos. Isso melhora a performance operacional, aumenta o conforto e satisfação dos ocupantes e provém aos usuários da construção com sistemas, tecnologias e ferramentas para administrar e minimizar o consumo de energia.

Como foram mencionados acima, os edifícios inteligentes, ao contrário dos edifícios tradicionais, possuem integração entre todos os seus serviços, ou seja, as informações fluem entre os sistemas de segurança, o controle de acesso, o setor de climatização, o sistema digital de áudio e vídeo entre outros. Além disso, existe também uma central de processamento que tem por função realizar as operações diárias de manutenção e utilização de seus ocupantes.

É importante enfatizar que o monitoramento desses edifícios é feito de forma local ou remotamente. As operações e requisições de informações sobre o funcionamento do sistema poderão ser feitas em tempo real, emitindo assim alertas aos funcionários do edifício sobre eventuais falhas que surgirem, a fim de controlar e administrar os incidentes bem como sanar os erros.

Há de se questionar a respeito da forma como é adotado o termo “inteligente” para os edifícios. Segundo (ALVES e MOTA, 2003, p. 9) denomina-se inteligência algumas características como: segurança, economia, conforto, ecologia e integração. É necessário não apenas automatizar e instalar novas tecnologias em um sistema, mas também pensar na utilidade, ou seja, na necessidade de seus ocupantes bem como na eficiência em termos econômicos e sustentáveis. Dessa forma, nota-se que a “inteligência” de um edifício não deve ser avaliada tomando como base apenas a quantidade de sistemas automatizados existentes, uma vez que o nível de automação não necessariamente está relacionado com o conceito de inteligência.

## 2.2 Integração em Edifícios Inteligentes

Antigamente, os sistemas de um prédio eram pensados de forma individual de acordo com a necessidade de cada área. Isso quer dizer que não havia a preocupação sobre como cada um destes sistemas poderia impactar nos demais.

O conceito de edifícios inteligentes traz a filosofia de um prédio como um todo, ou seja, todos os sistemas são interligados e se comunicam a fim de obter um melhor desempenho e otimização dos processos, podendo ser denominados como sistemas integrados.

A noção de integração é extremamente importante, pois oferece inúmeras vantagens e possibilidades, tais como:

- ✓ Maior eficiência no emprego de recursos existentes, bem como seu melhor aproveitamento;
- ✓ Aquisição de novas funções possibilitadas pela interação e cooperação entre sistemas/aplicações;
- ✓ Ações mais rápidas e coordenadas;
- ✓ Possibilidade de correlacionar informações, de forma a processá-las e aperfeiçoar as tomadas de decisão;
- ✓ Possibilidade de acesso aos vários sistemas através de um mesmo local, tornando a utilização mais fácil, resumida, flexível e eficaz;
- ✓ Aumento na capacidade produtiva através da simplificação de tarefas mais elaboradas envolvendo diferentes sistemas;
- ✓ Soluções mais eficazes, com melhor relação funcionalidade/custo.

As vantagens da integração são inegáveis, porém é importante mencionar também seus aspectos negativos:

- ✓ Em caso de transtornos operacionais relacionados com a interação entre os sistemas, há uma maior dificuldade para identificar a origem do problema, o que, conseqüentemente, dificulta a tomada de decisão para solucioná-lo;
- ✓ Poderão existir bloqueios legislativos à integração (por exemplo,

existem países que obrigam a que os sistemas de detecção de incêndio sejam independentes e isolados).

### **2.3 Serviços em Edifícios Inteligentes**

Os serviços em edifícios inteligentes podem ser entendidos como conjuntos de funções que, devido a características comuns, são agrupados numa entidade individualizada. Estas características podem ser: mesma natureza, mesma inter-relação e/ou dependência, partilha ou intervenção sobre dada informação, associação a um mesmo tipo de equipamento físico etc.

Vale salientar que as funções exercidas por um serviço não estão necessariamente associadas a um dispositivo físico (envolvendo interações com sensores e atuadores). Um serviço pode ser formado apenas por funções virtuais – de natureza software, como um determinado programa ou uma base de dados.

A possibilidade de serviços para edifícios inteligentes é vasta. Abaixo são mencionados os principais serviços existentes.

- ✓ Apoio à portaria;
- ✓ Aquecimento, ventilação e ar condicionado;
- ✓ Comunicações e distribuição de áudio e vídeo;
- ✓ Controle de acessos;
- ✓ Controle de estacionamento de veículos;
- ✓ Controle de irrigação;
- ✓ Detecção de situações de emergência;
- ✓ Diagnóstico de falhas e manutenção de sistema;
- ✓ Elevadores;
- ✓ Gestão de cabeamento;
- ✓ Gestão de presença;
- ✓ Gestão e administração de sistema;
- ✓ Gestão energética;
- ✓ Iluminação;
- ✓ Informação;
- ✓ Inventariação e gestão patrimonial;
- ✓ Localização de pessoas e equipamentos;

- ✓ Manutenção do edifício;
- ✓ Vigilância e detecção de invasão.

Todos os serviços mencionados acima possuem as seguintes competências:

- ✓ Formatação e gestão do serviço: Permite ajustes de acordo com a aplicação, como, por exemplo, a possibilidade de definir quais são os dispositivos atrelados ao serviço, bem como suas respectivas identificações, localizações e seus tipos;
- ✓ Monitoração e teste dos equipamentos a fim de registrar tempos de funcionamento e detectar episódios de falhas (informação importante para determinar ações de manutenção);
- ✓ Gestão e controle de acesso dos usuários;
- ✓ Interação com os usuários, propiciando consulta de informações e também a realização de configurações, parametrização e gestão do serviço;
- ✓ Registro de informações importantes para a gestão do processo, seja para monitoração, análise de dados ou transferência para outras aplicações.

## **2.4 Exemplos de Edifícios Inteligentes**

Em tempos mais conscientes, diversos empreendedores e profissionais da construção apostam nos edifícios inteligentes. Criar ou transformar um prédio em um edifício inteligente é vantajoso tanto para o proprietário quanto para os usuários ou organizações. Dessa forma, é possível desenvolver construções comprometidas com a diminuição do desperdício de energia e com a otimização dos seus sistemas e recursos.

Existem instituições que certificam edifícios construídos segundo rigorosos padrões de qualidade e eficiência. Uma delas é a USGB (*United States Green building Council*), que emite o certificado *Leadership in Energy and Environmental Design* (LEED). Criada em 1998 nos Estados Unidos, a certificação vem sendo aplicada em vários países e é reconhecida como um indicador de qualidade.

Pode-se mencionar dois exemplos de edifícios inteligentes no Brasil: o primeiro implantado em São Paulo e o segundo construído no Rio de Janeiro. Ambos são considerados empreendimentos comerciais assim como são reconhecidos e possuem o certificado LEED.

Porém, é importante deixar claro que a referência do termo edifício que está sendo abordado neste trabalho diz respeito a uma variedade de aplicações, desde construções de uso residencial, industrial, hospitalares, escritórios, museus até mesmo os prédios comerciais mencionados abaixo.

#### **2.4.1 Eldorado *Business Tower* – São Paulo, Brasil**

É um dos principais edifícios inteligentes existentes no Brasil. Projetado e arquitetado em 2007, na Avenida das Nações Unidas, região da Marginal Pinheiros da capital de São Paulo, o Eldorado *Business Tower* fica localizado em uma das áreas mais bem movimentadas da cidade. Isso porque esse edifício de escritórios fica próximo ao *Shopping Eldorado* e também perto das avenidas comerciais Faria Lima e Rebouças. Esse prédio (Figura 2) possui uma área construída de 128.645 m<sup>2</sup>, sendo composto por 32 pavimentos, 4 subsolos (1.805 vagas), edifício garagem com 7 pavimentos, centro de convenções e heliponto.



**Figura 2** – Eldorado Business Tower.  
(Fonte: GALERIA DA ARQUITETURA, 2018)

Tornou-se uma das principais referências de prédios inteligentes no Brasil em virtude de ser o primeiro edifício da América Latina a receber o certificado LEED da categoria *Platinum*, o mais alto nível de certificação *Green building* pelo USGB. Esse conselho tem por característica avaliar e reconhecer soluções e tecnologias sustentáveis adotadas em projetos com o intuito de reduzir os impactos ambientais causados por uma edificação.

Algumas características que tornam esse empreendimento reconhecido e que o levou a ganhar a certificação na categoria mais alta são por possuir economia no consumo de energia, ou seja, possuem sistemas de ar condicionado, iluminação e elevadores eficientes do ponto de vista energético além de possuírem um sistema próprio capaz de gerar energia; aproveitamento racional da água das chuvas e da água eliminada pelo sistema de refrigeração sendo com isso reaproveitado em banheiros, irrigação ou espelho d'água localizada na entrada do prédio; a fachada é composta de vidro verde trazendo economia, conforto térmico e aproveitamento da luz natural entre outras tecnologias inteligentes e sustentáveis implantadas no edifício.

Para Anderson Benite, diretor da Área de Sustentabilidade do CTE – Centro de Tecnologia de Edificações, empresa responsável pela consultoria em *Green building* para o edifício, “a certificação *Platinum* foi uma grande conquista para o empreendimento e um salto gigantesco para a evolução da construção sustentável no Brasil, demonstrando que este novo paradigma para a construção torna-se cada vez mais uma realidade”.

#### **2.4.2 Ventura *Corporate Towers* – Rio de Janeiro, Brasil**

O segundo exemplo a ser abordado é o empreendimento comercial conhecido como *Ventura Corporate Towers* situado na Avenida República do Chile, próximo a Catedral Metropolitana do Estado do Rio de Janeiro (Figura 3).



**Figura 3** – Ventura Corporate Towers.  
(Fonte: GALERIA DA ARQUITETURA, 2018)

É o primeiro edifício implantado na cidade do Rio de Janeiro a ganhar o certificado LEED da categoria Gold, nível abaixo do exemplo anterior, porém de grande relevância para o Estado, por ser considerado o pioneiro no reconhecimento.

Parecido com o edifício Eldorado *Business Tower*, este modelo de prédio também conta com o uso inteligente e racionalizado de água e energia; controle do descarte de entulho e reciclagem; uso de fachadas que contenham vidros especiais que favoreçam a iluminação natural e a ventilação do ambiente entre outras características. O Ventura Corporate Towers é composto também por duas torres gêmeas que possuem 140 metros de altura contendo 5 subsolos, 36 andares e heliporto.

Segundo Rick Fedrizzi, Presidente, CEO e Fundador do USGBC:

"A cada novo edifício certificado *green building*, damos um passo em direção à visão do USGBC de se criar um ambiente sustentável

nesta geração. Como membro mais novo da família de empreendimentos LEED®, o Ventura Corporate Tower vem contribuir para o fortalecimento do movimento *green building*".

Conclui-se, então, que edifícios inteligentes vão além de construções capazes de economizar energia e contribuir com metas de sustentabilidade. Eles devem aproveitar ao máximo os recursos existentes através de tecnologias disponíveis. Uma das tecnologias mais recentes e abordadas neste trabalho diz respeito à Internet das Coisas.

Utilizando novas tecnologias e gerenciando bem os dados coletados pelos sensores, é possível alcançar uma automação capaz de monitorar o desempenho, detectar anomalias, diagnosticar possíveis causas, fazer ajustes e correções automaticamente. Tudo isso com o objetivo de aproveitar ao máximo os recursos. Resumindo, o sistema é capaz de conectar as tecnologias e estas, por sua vez, às pessoas. Assim, é possível alcançar sistemas inteligentes e eficientes de iluminação, ar condicionado, uso de elevadores, aproveitamento do vento, aproveitamento de água, entre outros.

### 3 INTERNET DAS COISAS OU INTERNET OF THINGS (IoT)

#### 3.1 Histórico

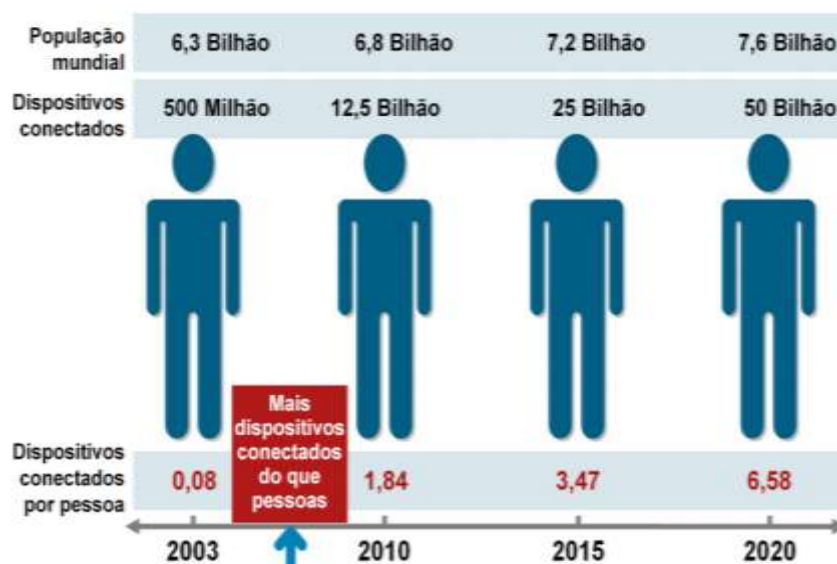
O surgimento da Internet das Coisas está atrelado com os avanços existentes em algumas áreas como, por exemplo, a microeletrônica, a informática, o sensoriamento e os sistemas embarcados. Por volta de 1991, a ideia de conectar objetos começou a ser discutida em virtude da disseminação da conexão TCP/IP e da internet. Nessa mesma época, Bill Joy, cofundador da Sun Microsystems, começou a abordar sobre a conexão entre diferentes dispositivos (D2D – *Device to Device*), cuja intervenção humana não era necessária.

Entretanto foi, aproximadamente, em 1999 que o termo “Internet das Coisas” foi de fato proposto pelo cofundador e diretor executivo do Auto-ID Center conhecido por Kevin Ashton, durante uma apresentação para os executivos da Procter & Gamble, onde ele mencionava a importância de se etiquetar eletronicamente os produtos com base na tecnologia RFID (do inglês “*Radio Frequency Identification*”), cujo intuito era melhorar a logística da cadeia de produção da empresa. É importante frisar que Kevin Ashton acreditava que os objetos do mundo físico poderiam conectar-se a internet, criando assim um mundo mais inteligente.

A partir de 2005, este conceito foi mais difundido, ganhando a atenção dos governantes que viam nesta nova tecnologia a oportunidade de avanço no que diz respeito à privacidade e segurança de dados. Nesse ano, a *International Telecommunications Union* (ITU) publicou um relatório com o conceito de Internet das Coisas, cuja ideia era de que a Internet das Coisas poderia conectar qualquer objeto, por meio de tecnologias, como RFID, sensores, rede de sensores sem fio, sistemas embarcados e nanotecnologia, além de transpor alguns desafios importantes como padronização, privacidade, espectro de frequência e questões sociais e éticas (FREITAS DIAS, 2016).

Entre 2008 e 2009 houve um ponto importante na história, onde dados divulgados pela Cisco IBSG – *Internet Business Solutions* apontavam que existia mais objetos ou coisas conectados a internet do que pessoas.

Atualmente, além dos tradicionais computadores, existe uma gama de dispositivos conectados a internet, como por exemplo, TVs, laptops, eletrodomésticos, automóveis, smartphones, entre outros. Nesse novo contexto, é possível perceber que os números de aparelhos conectados estão em constante crescimento e, conforme Figura 4 abaixo, há previsões indicando mais de 50 bilhões de dispositivos conectados até 2020.



**Figura 4** – Relação da população mundial com a quantidade de dispositivos conectados.  
(Fonte: CISCO IBSG, 2011)

### 3.2 Definição

Em vista da crescente popularidade da Internet das Coisas, não existe um conceito único para o seu significado. Segundo (RAIWANI, 2013; KRANENBURG, 2011) o termo Internet das Coisas é um pouco confuso, não está bem definido nas pesquisas científicas, conferências realizadas no mercado e na academia, e está sujeito ao debate filosófico.

Ainda assim, é importante ressaltar algumas definições com o intuito de esclarecer melhor o conceito de IoT. Uma das primeiras abordagens sobre o termo foi descrito em 2005 pela *International Telecommunication Union* (ITU) sendo considerada como: “uma conexão de todos os objetos e dispositivos do cotidiano a todos os tipos de redes: intranets, redes *peer-to-peer* e a Internet global que se conhece”.

Pode-se considerar que a Internet das Coisas permite que pessoas e coisas se conectem a qualquer hora, em qualquer lugar, com qualquer coisa, utilizando qualquer caminho, sendo este rede ou serviço.

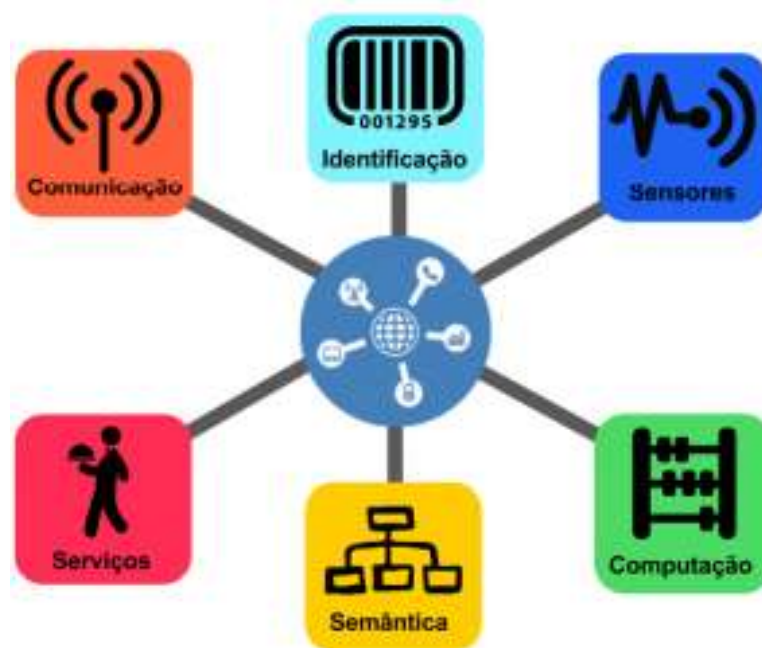
Outro conceito definido por Atzori et al (2011, p. 2787) diz que:

A ideia básica desse conceito é a presença generalizada à nossa volta de uma variedade de coisas ou objetos – como tags de identificação por radiofrequência (RFID), sensores, atuadores, telefones celulares, etc. – que, por meio de esquemas de endereçamento exclusivos, são capazes de interagir uns com os outros e cooperar com outros objetos para alcançar objetivos comuns.

Por fim, vale ressaltar que a IoT é uma rede banda larga que utiliza protocolos de comunicação padronizados e tem seu ponto de encontro na internet. O principal conceito da IoT é a utilização de sensores conectados com a internet, que tem por função coletar determinados tipos de informações e enviar para um local de destino afim de serem analisados e armazenados. Em seguida, serão tomadas algumas decisões com o intuito de alterar o ambiente no qual esses sensores estão inseridos.

### 3.3 Blocos Básicos da Construção IoT

Os blocos básicos da construção da IoT estão relacionados na Figura 5:



**Figura 5** – Blocos básicos da construção IoT.  
(Fonte: SANTOS et al., 2016)

- ✓ Identificação: É necessário fazer a identificação dos objetos com o intuito de conectá-los a internet. Algumas tecnologias podem ser úteis para a identificação das coisas, como exemplo, podemos destacar o RFID, o endereçamento IP entre outros;
- ✓ Sensores/Atuadores: Tem por função coletar as informações sobre o ambiente em que esses objetos se encontram para em seguida serem encaminhados e armazenados em data *warehouse*, *clouds* ou centros de armazenamento. Em relação aos atuadores, eles são responsáveis em modificar o ambiente ou reagir de acordo com os dados lidos;
- ✓ Comunicação: Está relacionado com as diversas técnicas usadas para conectar objetos inteligentes. Existem algumas tecnologias usadas e são elas: WiFi, Bluetooth entre outros;
- ✓ Computação: São considerados as unidades de processamento existentes, que podem ser os microcontroladores e processadores;
- ✓ Serviços: A IoT pode fornecer diversas classes de serviços, podendo ser de identificação, de agregação de dados, de colaboração e inteligência, entre outros;
- ✓ Semântica: Está relacionado com a descoberta de conhecimento e o uso eficiente dos recursos existentes na IoT, a partir dos dados gerados, com o objetivo de prover determinado serviço.

### **3.4 Aplicação**

Existem diversas aplicações relacionadas à Internet das Coisas em virtude da variedade de objetos que permeiam a vida diária das pessoas, empresas e sociedades como um todo e que são capazes de se conectarem a rede. A Figura 6 ilustra algumas atuações da IoT.



**Figura 6** – Algumas atuações da IoT.  
(Fonte: MANCINI, 2017)

- ✓ Bens de consumo: Equipamentos ou produtos inteligentes utilizados pelos consumidores e são eles: *Smart TV*, geladeiras, fogões, carros entre outros;
- ✓ *eHealth*: Está relacionado com a saúde das pessoas, ou seja, otimizar o monitoramento cardíaco, glicose entre outros;
- ✓ Transporte Inteligente: É possível fazer o controle sobre as condições de tráfego, o monitoramento sobre vias congestionadas ou com acidentes entre outros;
- ✓ Distribuição de Energia: Poderão ser feitas medições remotas de relógios nas residências ou edifícios, subestações inteligentes entre outros;
- ✓ Casas Inteligentes: Poderão ser feitas medições de consumo de energia, controle inteligente sobre equipamentos ou sobre a segurança das residências;
- ✓ Distribuição e logística: *Smart e-commerce*, rastreabilidade e um melhor gerenciamento na distribuição;
- ✓ Segurança Pública: Possibilita a utilização de equipamentos audiovisuais e microfones de alta definição e sensibilidade, de novas câmeras com tecnologia OCR (que conseguem reconhecer caracteres de placas de veículos) e tecnologia de reconhecimento facial, além de diversos tipos de sensores;

- ✓ Indústria e Manufatura: Aprimoramento sobre a indústria buscando economia e eficiência através da utilização de sensores e atuadores com capacidade para controlar os equipamentos de forma a prever manutenções e reparos;
- ✓ Gestão da agricultura e recursos naturais: Gerenciamento da qualidade do produto, monitoramento ambiental sobre as etapas de produção e cultivo;
- ✓ *Smart Cities*: É uma das principais aplicações da IoT e que tem por objetivo fazer o monitoramento sobre as diversas áreas ou equipamentos existentes nas ruas, como por exemplo, iluminação pública, segurança, trânsito, lixeiras entre outros.

### **3.5 Atuação da IoT em Edifícios Inteligentes**

A Internet das Coisas pode ser aplicada em edifícios inteligentes através da conexão dos seus sistemas de automação predial, facilitando a gestão do prédio, uma vez que todos os seus serviços (iluminação, ventilação, ar condicionado etc) são integrados, podendo inclusive conversar entre si e tomar decisões de forma autônoma (dentro de um escopo de atividades pré-programadas) relativas a cada um e em função das informações coletadas em outros sistemas.

Além disso, os resultados obtidos dos sistemas são considerados inteligentes e integrados e podem inclusive contar com um serviço auxiliar de manutenção preventiva, que através de históricos, estatísticas e dados atuais dos sensores, permitem melhorar o desempenho do edifício.

É importante destacar que os usuários cadastrados nas instalações dos edifícios podem, por exemplo, controlar a iluminação e climatização do ambiente, através do auxílio de um *smartphone* e de acordo com as suas necessidades diárias. É possível também que, quando for detectada a ausência de pessoas em determinado cômodo, o sistema desligue ou reduza a atividade dos equipamentos, minimizando assim o consumo de energia. Ou seja, a IoT pode permitir através dos mecanismos de automação, diminuir a demanda de energia elétrica nos períodos de maior consumo, o que gera enorme vantagem para as concessionárias, visto que

não se torna necessário uma expansão da rede instalada (ALLTOMATIC, 2015, s.p.).



**Figura 7** – Imagem ilustrativa sobre IoT.  
(Fonte: COMPUTER WORLD, 2018)

### **3.6 Desafios para a Implantação da IoT**

Atualmente existe um grande avanço na área da Internet das Coisas, proporcionado pela diminuição de custos e tamanhos dos sensores, chips e microprocessadores bem como pelo surgimento do *Big Data* e do protocolo IPV6. Porém, vale ressaltar que é necessário ultrapassar algumas barreiras para que de fato seja possível vislumbrar todas as “coisas”, ou seja, todos os objetos conectados, sendo elas:

- ✓ **Eficiência Energética:** É necessário que exista cada vez mais um aprimoramento sobre as tecnologias e a durabilidade das baterias dos dispositivos, diminuindo assim a necessidade de recarga e os impactos sobre o consumo de energia. Como alternativa, novas possibilidades são essenciais, como por exemplo, investir em baterias com capacidade de recarga através de fontes ambientais, seja pela luz solar, vibrações, fluxos de ar entre outras;
- ✓ **Padrões:** É importante continuar o progresso das padronizações da arquitetura e dos protocolos de comunicação para garantir a interoperabilidade do sistema e a convergência entre todos os dispositivos conectados da rede. A IEEE é apenas uma das organizações responsáveis pela normatização;

- ✓ **Segurança e Privacidade:** Em virtude do grande volume de dados que trafegam na rede, inclusive aqueles pessoais, faz-se necessário aumentar e aprimorar a confiabilidade e a criptografia da rede, evitando assim que os dados sejam vazados ou que os sistemas sejam invadidos. Os governos devem rever ou criar leis que regem a internet para evitar ou sanar qualquer dano às pessoas e corporações.

Apesar de existir alguns empecilhos para o desenvolvimento pleno da IoT, estes não são considerados intransponíveis, sendo apenas uma questão de tempo para que novos projetos sejam realizados.

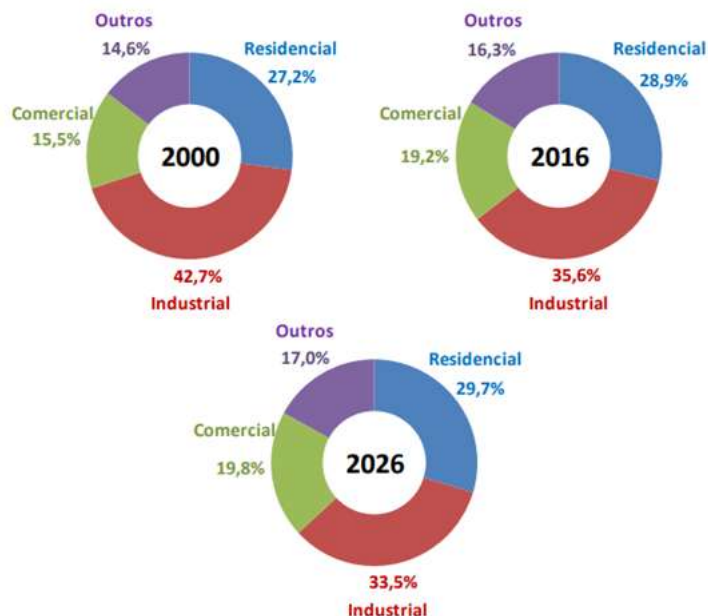
#### 4 CONSUMO DE ENERGIA

O crescente consumo de energia elétrica está diretamente relacionado com o uso cada vez mais frequente dos aparelhos eletrodomésticos e bens de consumo usufruídos pela população, tanto no setor residencial, como em setores indústrias, comerciais, entre outros, que demandam constantemente o uso da eletricidade. Este cenário requer que novos desafios sejam adotados no que tange a geração, transmissão e distribuição de energia, sendo necessário tomar medidas preventivas de eficiência energética, com o intuito de minimizar os gastos.

Sendo o foco do presente trabalho o setor de edificações, é importante investir em soluções que sejam econômicas do ponto de vista energético além de ser sustentável, é claro. Uma boa alternativa é fazer o monitoramento do consumo dos equipamentos e serviços utilizados. A medição de energia possibilita a obtenção de informações referente ao funcionamento de uma edificação e o conhecimento gerado através de análises tem potencial para aperfeiçoar o desempenho do edifício. Além disso, tem a oportunidade também de auxiliar na identificação de corte de custos detectando ineficiências no sistema.

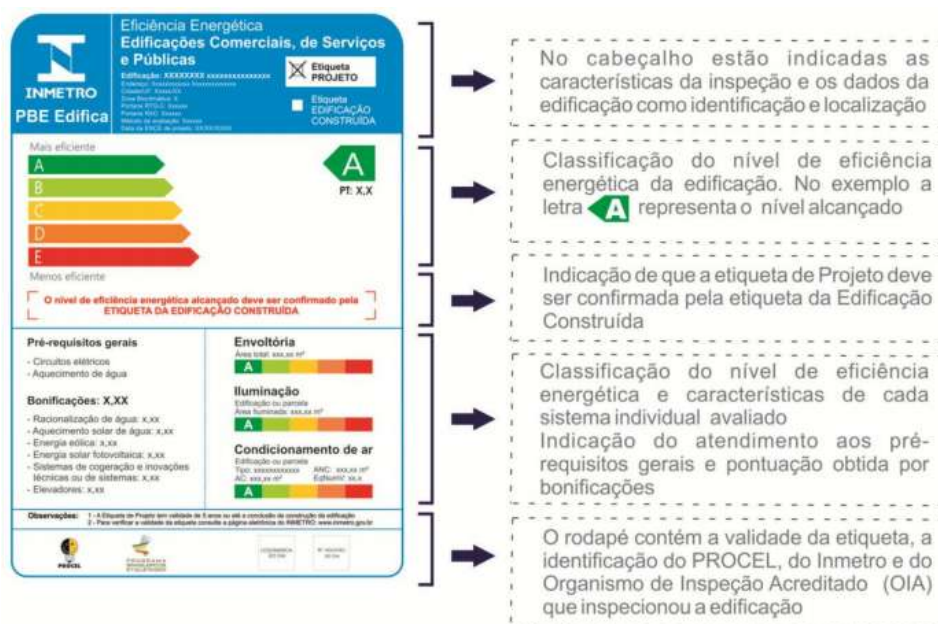
De acordo com a declaração do professor Roberto Lamberts, do Laboratório de Eficiência Energética em Edifícios da Universidade Federal de Santa Catarina (UFSC), mesmo os edifícios concebidos para atender a certificações de sustentabilidade, o monitoramento do consumo é necessário para garantir que a operação esteja de acordo com o projetado. Em acordo, Lamberts acredita também que “não adianta fazer apenas um projeto eficiente se a operação desperdiça energia, seja por falhas de manutenção nos equipamentos ou por falta de engajamento dos usuários”.

A Figura 8 representa dados sobre o consumo de energia elétrica nos diferentes tipos de edifícios existentes nos próximos anos. Pode-se observar que o consumo de eletricidade tende a aumentar principalmente nos setores comerciais e residenciais ao contrario dos edifícios industriais que tende a diminuir. Porém esses valores são considerados relevantes do ponto de vista energético e por isso é necessário tomar medidas conscientes de eficiência energética.



**Figura 8** – Estrutura do consumo brasileiro de eletricidade na rede, por classe (%).  
(Fonte: EPE, 2017)

Uma importante adoção que vem sendo realizada pelo governo diz respeito ao selo Procel de Edificações emitidos pela Eletrobrás e avaliada pelo Inmetro, chamada etiqueta de nacional de conservação de energia (ENCE) PBE Edifica. O objetivo é identificar prédios que apresentem as melhores classificações de eficiência energética em uma dada categoria, incentivando o mercado consumidor a adquirir e utilizar imóveis mais eficientes. O modelo da etiqueta está representado na Figura 9.



**Figura 9** – Modelo de ENCE (Etiqueta Nacional de Conservação de Energia) - PBE Edifica.  
Fonte: PBE EDIFICA, 2014

A classificação é de acordo com os serviços, no qual são avaliados: envoltória, iluminação e condicionamento de ar para edifícios comerciais, de serviços e públicos; e envoltória e sistema de aquecimento de água nas Unidades Habitacionais.

A seguir, serão abordadas algumas grandezas elétricas fundamentais para a realização do monitoramento de consumo de uma carga instalada.

#### 4.1 Tensão Elétrica

Tensão é uma grandeza elétrica que representa a diferença entre os potenciais de dois pontos distintos de um condutor. De modo geral, é a energia necessária para movimentar uma carga elétrica. A unidade de medida no Sistema Internacional de Unidades (SI) é o Volt (V).

Quando a tensão existente entre dois pontos distintos corresponde a um Volt, significa que cada carga, de um Coulomb, que se movimenta nesses pontos, transmite um Joule em energia. Portanto, essa grandeza é expressa pela seguinte Equação 4.1:

$$v = \frac{dw}{dq} \quad (4.1)$$

Onde:

- $v$  = tensão, dada em Volts;
- $w$  = energia, dada em Joules;
- $q$  = carga, dada em Coulombs.

A tensão elétrica pode ser caracterizada sobre duas formas: contínua (sem variação ao longo do tempo) e alternada (com variação ao longo do tempo). O foco deste trabalho é a tensão elétrica da rede, ou seja, a tensão alternada disponibilizada para o consumo em residências, indústrias, escritórios, comércios etc. A mesma possui uma forma de onda senoidal.

A Equação 4.2 representa a tensão senoidal. É importante mencionar que ela pode ser expressa tanto com a função seno quanto pela função cosseno. Esta por sua vez será a utilizada no presente trabalho.

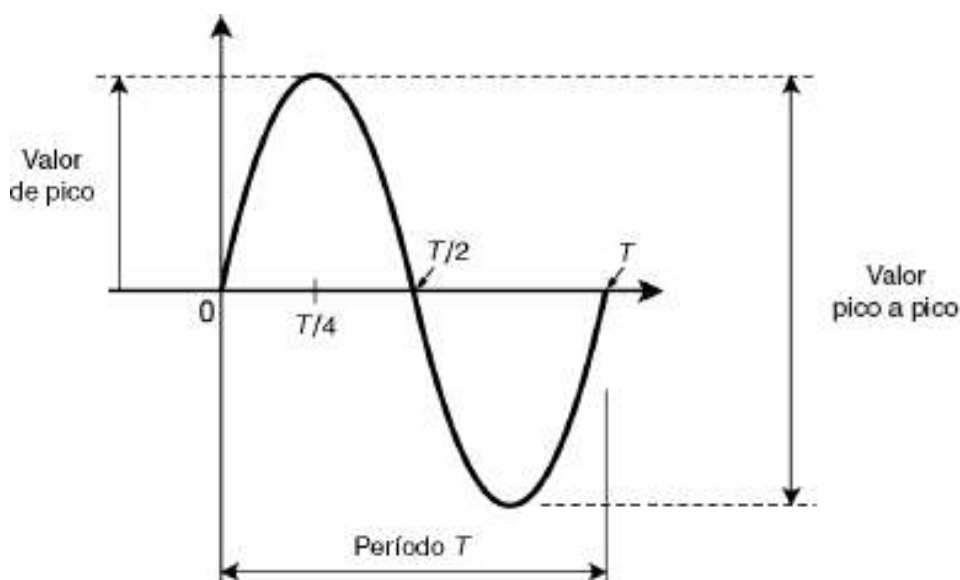
$$v(t) = V_m \times \cos (\omega t + \theta_v) \quad (4.2)$$

Onde:

- $v$  = tensão, dada em Volts;
- $V_m$  = amplitude máxima da tensão senoidal, dada em Volts;
- $\omega$  = frequência angular, dada em radianos por segundo;
- $t$  = tempo, dado em segundos;
- $\theta_v$  = o ângulo de fase da tensão, dado em graus.

Por repetir-se em intervalos regulares, a tensão elétrica é considerada uma onda senoidal periódica. Possui dois semiciclos, sendo um positivo (intensidade aumenta do zero até seu valor máximo e posteriormente vai sendo reduzida até retornar ao zero) e outro negativo (intensidade diminui do zero até seu valor mínimo e posteriormente vai sendo aumentado até retornar ao zero).

A Figura 10 ilustra o comportamento de um ciclo completo, ou seja, o conjunto de valores (positivos e negativos) que compõem a onda. Os parâmetros em destaque na figura são: amplitude máxima que a onda alcança ( $V_m$ — representado por “Valor de pico”), diferença entre o valor máximo e mínimo da amplitude (representado por “Valor pico a pico”) e o período, intervalo de tempo necessário para que a onda percorra todos os seus valores possíveis durante um ciclo (representado por “Período T”).



**Figura 10** – Ciclo de uma onda alternada com indicação de seus parâmetros.  
(Fonte: SILVA FILHO, 2007)

Através do período, pode-se extrair a frequência da onda, que corresponde ao número de oscilações ou ciclos por intervalo de tempo. A frequência adotada no Brasil é 60 Hertz (Hz). Segue abaixo a fórmula que lhe representa (Equação 4.3):

$$f = \frac{1}{T} \quad (4.3)$$

Onde:

- $f$  = frequência, dada em Hertz;
- $T$  = período, dado em segundos.

A frequência pode ser representada de forma linear ou angular. A forma linear é a mais comum, mostrada acima, enquanto a angular, ômega ( $\omega$ ), indica uma medida da velocidade de rotação e está representada na Equação 4.4.

$$\omega = 2\pi \times f \quad (4.4)$$

Onde:

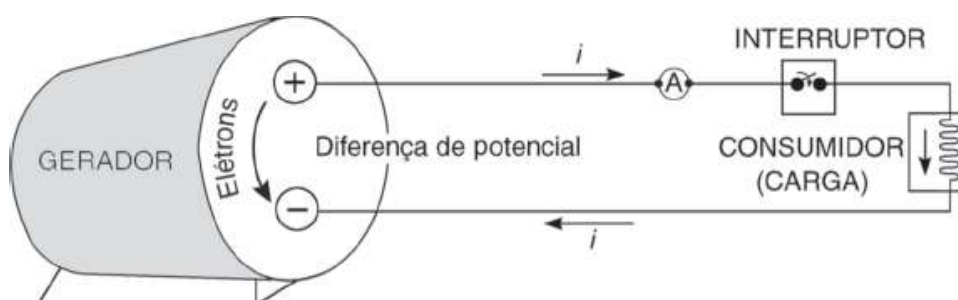
- $\omega$  = frequência angular, dada em radianos por segundo;
- $f$  = frequência, dada em Hertz;
- $\pi$  = constante circular, e o seu valor adimensional corresponde a 3,14159.

## 4.2 Corrente Elétrica

A corrente elétrica é uma grandeza elétrica que representa o fluxo de cargas que atravessa a superfície de um condutor em um determinado intervalo de tempo. Esse movimento de cargas se dá através dos elétrons livres que se encontram dentro do condutor e que estão sob o efeito da diferença de potencial (d.d.p.) entre os seus terminais. Sua unidade de medida no SI é o Ampère (A).

Para descrever melhor o que foi mencionado acima, a Figura 11 representa um esquema contemplando gerador, condutor, amperímetro e um interruptor em série com uma carga. Quando os terminais desse gerador estiverem ligados a um circuito elétrico fechado, uma corrente elétrica percorrerá o condutor, movimentando os elétrons livres no interior do mesmo devido à ocorrência de uma diferença de potencial entre as suas extremidades. Por fim, a carga poderá ser

acionada ou não, de acordo com a presença da corrente elétrica e também pelo estado (ligado/desligado) da chave do interruptor.



**Figura 11** – Esquema de circuito elétrico fechado.  
(Fonte: CREDER, 2016)

Segue abaixo Equação 4.5 que representa a intensidade da corrente elétrica:

$$i = \frac{dq}{dt} \quad (4.5)$$

Onde:

- $i$  = corrente elétrica, dada em Ampères;
- $q$  = carga elétrica, dada em Coulombs;
- $t$  = tempo, dado em segundos.

Assim como a tensão elétrica da rede é representada por uma forma de onda senoidal periódica, o mesmo pode ser considerado para a corrente elétrica, uma vez que ambas são consideradas alternadas. Portanto, a fórmula que representa a corrente elétrica alternada é a Equação 4.6 abaixo:

$$i(t) = I_m \times \cos(\omega t + \theta_i) \quad (4.6)$$

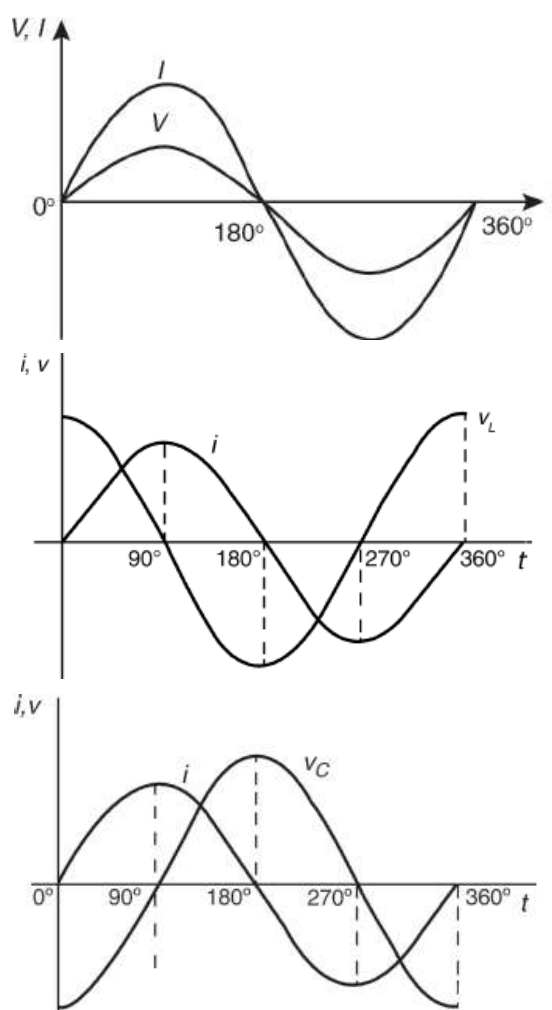
Onde:

- $i$  = a corrente elétrica, dada em Ampères;
- $I_m$  = amplitude máxima da corrente senoidal, dada em Ampères;
- $\omega$  = frequência angular, dada em radianos por segundo;
- $t$  = tempo, dado em segundos;
- $\theta_i$  = ângulo de fase da corrente, dado em graus.

Em sistemas de corrente alternada, é comum realizar comparação entre duas formas de onda utilizando um único gráfico. Quando ambas atingem valores máximos e mínimos ao mesmo tempo, elas estão em fase, caso contrário, estão

defasadas. Esta defasagem é medida com a unidade correspondente ao eixo das abscissas, geralmente graus. É importante salientar que, para que a medição ocorra de forma correta, é essencial que as senóides possuam a mesma frequência.

As ondas senoidais de tensão e corrente podem apresentar determinados comportamentos de acordo com a carga instalada. Neste contexto, é possível destacar três principais tipos de circuito: puramente resistivo, puramente indutivo e puramente capacitivo, conforme ilustrado na Figura 12 (a), (b) e (c), respectivamente. No primeiro caso, a tensão e corrente estão em fase; no segundo caso, como o indutor adianta a tensão, há um avanço de  $90^\circ$  em relação à corrente; no terceiro caso, ocorre o contrário: o capacitor atrasa a tensão e por isto a mesma está atrasada  $90^\circ$  em relação a corrente.



**Figura 12** – Comparativo de ondas de corrente e tensão.

- a) Circuito puramente resistivo, tensão e corrente em fase;
  - b) Circuito puramente indutivo, tensão avançada de  $90^\circ$  sobre a corrente;
  - c) Circuito puramente capacitivo, tensão atrasada de  $90^\circ$  sobre a corrente.
- (Fonte: CREDER, 2016)

## 5 PLATAFORMA, PROTOCOLO E SERVIÇOS UTILIZADOS

Para compreender como o projeto foi desenvolvido, é necessário ter o conhecimento da plataforma, do protocolo e dos serviços utilizados. A seguir serão introduzidos alguns conceitos gerais a fim de auxiliar o entendimento dos capítulos seguintes.

### 5.1 IBM Cloud

O IBM Cloud, também conhecido como IBM Bluemix, é uma plataforma de computação em nuvem de código aberto da empresa IBM (*International Business Machines*). Sua característica se baseia em um *Cloud Foundry*, ou seja, uma plataforma como serviço (PaaS) que permite criar, implementar e gerenciar rapidamente seus aplicativos na nuvem.

A plataforma como serviço provém a hospedagem e a implementação de *hardware* e *software* para criar diversas aplicações por meio da internet, substituindo a velha tradição de aplicação, onde era necessário fazer a instalação de programas nas máquinas dos usuários, muitos dos quais eram restritos ao sistema operacional que possuíam e cuja função demandava maiores custos e tempo para poder desenvolver os seus aplicativos.

Para enfatizar o que foi mencionado acima, REYES (2014) descreve a importância do IBM Cloud da seguinte forma:

Para os desenvolvedores, o IBM Cloud otimiza consequentemente o tempo gasto na criação do aplicativo da nuvem. Com poucos cliques, é possível proporcionar instâncias dos seus aplicativos com os serviços necessários para oferecer suporte. Esta gestão eficiente se traduz em incontáveis horas de instalação, configuração e solução de problemas que agora podem ser usadas em um tempo de rápida inovação e reação às intermináveis mudanças nos requisitos.

O console do IBM Cloud, conforme Figura 13, é um painel que representa uma interface para o usuário acessar todos os recursos disponíveis bem como efetuar *login*, acessar o menu de catálogo, documentação, suporte entre outros, ou seja, é considerado uma poderosa ferramenta que possui diversos serviços ou plataformas importantes para o desenvolvimento de aplicações na nuvem.



**Figura 13** – Console do IBM Cloud.  
(Fonte: Autores, 2018)

Existem diversos tipos de serviços que a IBM Cloud oferece como, por exemplo, serviços cognitivos, de Internet das Coisas, de inteligência artificial bem como suporte a diferentes tipos de linguagem de programação que são elas JavaScript, PHP, Python entre outros.

Os serviços disponíveis são tanto de nível básico quanto empresarial. Devido às suas tecnologias de código aberto implícitas, o IBM Cloud oferece flexibilidade para integrar o marco de desenvolvimento e os serviços que se adaptem às suas necessidades (REYES, 2014). Vale ressaltar que existem outras empresas que oferecem serviços de nuvem similares ao IBM Cloud, como o *Amazon Web Services* ou o *Microsoft Azure*.

## 5.2 IBM Watson

Watson é um software que foi criado pela própria IBM em 2011 e que ganhou visibilidade neste mesmo ano após vencer um programa de perguntas e respostas transmitido pela TV americana chamada *Jeopardy!*. Os concorrentes eram dois humanos campeões de edições anteriores.

A partir de então, o Watson foi cada vez mais desenvolvido e aprimorado para chegar ao ponto em que se encontra hoje. Atualmente, pode ser considerado como uma plataforma tecnológica que tem o objetivo de criar conhecimento de dados e melhorar os atributos humanos através do trabalho conjunto entre o homem e máquina.

Isso é possível através da junção de três conceitos básicos, sendo eles: inteligência artificial, *machine learning* e computação cognitiva. Enquanto a inteligência artificial simula a aptidão humana de raciocinar, perceber, resolver

problemas e tomar decisões, o *machine learning* torna possível a capacidade de aprendizado da máquina através de grandes volumes de dados. Aliado a isso, a computação cognitiva torna possível os computadores pensarem quase como seres humanos, sendo capazes de processar tanto dados estruturados como dados não estruturados (como vídeos e e-mails).

D'EGMONT (2016) diz que:

Na prática, são diversas API's disponíveis de Watson e, com elas, você pode criar seus próprios programas, sistemas, aplicações (como quiser chamar) cognitivas. Ou seja, toda a inteligência por trás do Watson está disponível para que desenvolvedores criem as mais incríveis aplicações em cima dela.

Para melhor entendimento deste conceito, deve-se entender também o que é API. Do inglês "*Application Programming Interface*", API é basicamente uma interface que conecta aplicações e pode ser utilizada em diversos ramos, por empresas de diferentes tamanhos e segmentos de mercado. As API's podem realizar integração entre sistemas de linguagens diferentes, possibilitando a conexão entre tecnologias heterogêneas.

Para utilizar o Watson é necessário ter certo conhecimento de programação e uma boa quantidade de dados – pois o Watson precisa ser "treinado" e esse treinamento se dá com o envio de dados para o sistema.

A partir do momento que esta etapa é concluída, o Watson é capaz de realizar uma determinada tarefa. As possibilidades são infinitas e com base no retorno de suas interações, o Watson se torna cada vez mais "inteligente".

Uma das aplicações desta tecnologia no Brasil se deu através de uma parceria do Banco Bradesco com a IBM Watson visando melhorar atendimento ao cliente. O principal problema verificado era a dúvida sobre produtos e serviços do banco. Quando qualquer atendente possuía alguma dúvida, era necessário entrar em contato com uma central e entrar numa fila de espera para ser atendido. Para resolver esta questão, a IBM e o Bradesco montaram uma equipe para treinar o Watson utilizando o modelo de perguntas e respostas. Este foi o primeiro projeto em português do sistema Watson e a complexidade era alta. Foi então que nasceu a BIA, Bradesco Inteligência Artificial.

Segundo SEGURA, 2018:

Com apenas 5 meses de treinamento, a BIA já conseguia entender 100% das perguntas escritas e 83% das perguntas faladas. E depois de 10 meses, o sistema já respondia 96% de todas as perguntas corretamente.

A BIA já está treinada em 62 produtos e responde a uma média de 283 mil perguntas por mês, com uma taxa de precisão de 95%, sendo que apenas 5% das perguntas recebidas exigem chamadas adicionais em busca de mais assistência. Em alguns casos, o tempo de resposta foi reduzido de 10 minutos para apenas alguns segundos.

As aplicações para o Watson são inúmeras e compreendem todos os ramos imagináveis. No presente projeto o Watson será utilizado juntamente com o IBM Cloud, através dos serviços de Internet das Coisas e da plataforma Node-RED.

### **5.3 MQTT ou *Message Queuing Telemetry Transport***

A IBM possui uma plataforma de IoT, que será utilizada no presente trabalho, baseada no protocolo MQTT. Por este motivo, a avaliação de outros protocolos não se faz necessário.

MQTT é um protocolo de comunicação que surgiu no final da década de 90 e que foi desenvolvido por duas empresas conhecidas como IBM e Eurotech com o intuito de realizar a troca de dados e informações entre diferentes dispositivos, ou seja, de máquina para máquina (M2M - *Machine to Machine*).

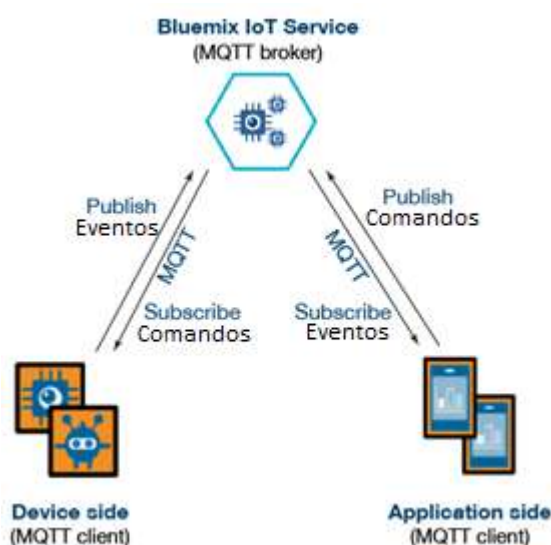
Além disso, por ser considerado um protocolo de rede leve, flexível, de fácil processamento, que requer uma baixa largura de banda e latência, também tem a característica de trazer a confiabilidade e segurança para as aplicações. Por este motivo, este protocolo é muito utilizado em Internet das Coisas, razão pela qual foi escolhido para o desenvolvimento deste presente trabalho.

Entretanto, vale salientar que para os dispositivos da Internet das Coisas se comunicarem, a conexão com a rede é um requisito. Paralelamente, o protocolo principal da internet é o TCP/IP, sendo assim o MQTT foi desenvolvido com base na pilha TCP/IP e atualmente tornou-se o padrão para comunicações de IoT.

O princípio de funcionamento do protocolo MQTT parte da utilização de duas entidades da rede e são eles: o servidor e os clientes. O primeiro, também

conhecido como *broker* MQTT, é responsável por receber, armazenar e repassar as informações necessárias. E o segundo, que é o cliente, é qualquer coisa que possa interagir com o *broker* e publicar/receber mensagens (*publish/subscribe*). Além disso, é importante ressaltar que esta comunicação só é possível devido à existência de uma chave (*token*) única utilizada para identificar a informação e um *payload*, que é a informação que será enviada ou recebida.

A Figura 14 abaixo mostra um exemplo no qual o *broker* é um servidor da IBM – Bluemix IoT e os clientes são dispositivos ou aplicativos. Dispositivos incluem máquinas que publicam informações que elas detectam e aplicativos são os programas que consomem as informações recebidas desses dispositivos (TANG, 2015).



**Figura 14** – Exemplo de aplicação do protocolo MQTT.  
(Fonte: Adaptado de TANG, 2015)

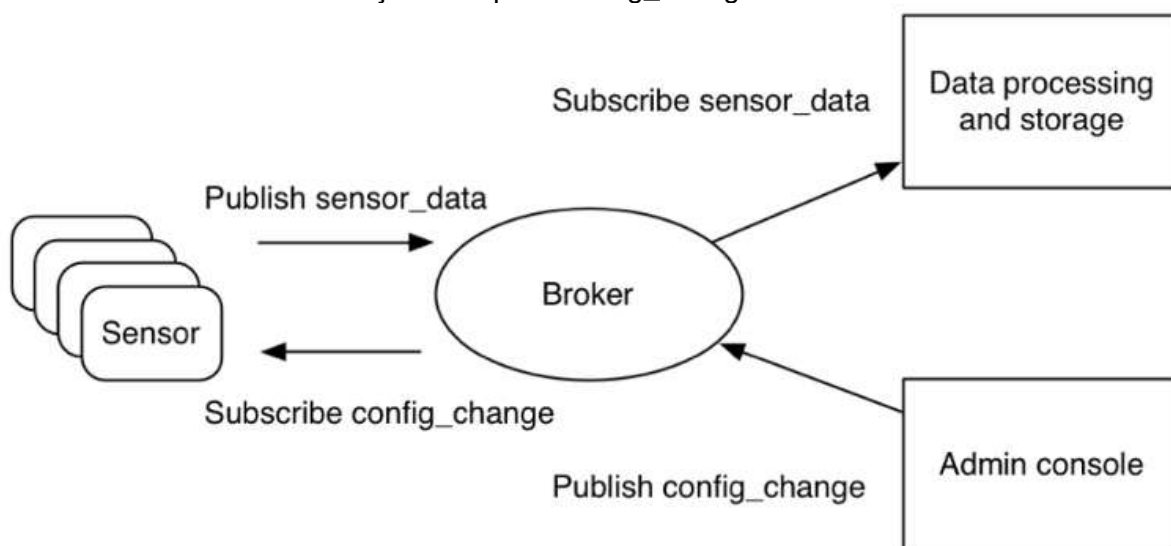
Para enfatizar o que foi exposto acima, a comunicação é feita da seguinte forma:

- I. O cliente se conecta ao *broker* MQTT, ou seja, ele envia uma mensagem de “*connect*” para estabelecer uma conexão do cliente com o servidor;
- II. O cliente publica/*publish* as mensagens em um tópico, enviando a mensagem e o tópico ao *broker*. O mesmo pode assinar qualquer “tópico” de mensagem no *broker*;
- III. Em seguida, o *broker* encaminha a mensagem a todos os clientes que assinam/*subscribe* esse tópico.

Por fim, vale ressaltar que as mensagens MQTT são organizadas por tópicos “*topics*”, assegurando assim uma segurança e flexibilidade para especificar quais clientes tem acesso a determinados tipos de mensagens na aplicação.

YUAN (2017) cita como exemplo o modelo *publish/subscribe* para sensores IoT explicado abaixo e ilustrado na Figura 15:

Os sensores publicarão suas leituras no tópico "sensor\_data" e assinarão o tópico "config\_change". Os aplicativos de processamento de dados que salvam os dados do sensor em um banco de dados de backend assinarão o tópico "sensor\_data". Um aplicativo de console administrativo poderia receber comandos do administrador do sistema para ajustar as configurações dos sensores, como a sensibilidade e a frequência de amostragem, e publicar essas mudanças no tópico "config\_change".



**Figura 15** – Modelo de *publish/subscribe* para sensores IoT.  
(Fonte: YUAN, 2017)

#### 5.4 Node-RED

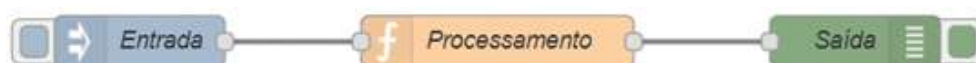
O Node-RED é uma ferramenta de programação desenvolvida em 2013, também pela empresa IBM, que conecta dispositivos de hardware, APIs e serviços *online*. É voltada principalmente para o desenvolvimento de aplicações de IoT e possui bastante versatilidade, sendo compatível com vários microcontroladores, como, por exemplo, Arduino, ESP32 e Raspberry Pi.

Utiliza a programação baseada em fluxo (FBP) que “é uma maneira de descrever o comportamento de uma aplicação como uma rede de caixas pretas, ou ‘nós’, como são chamados no Node-RED” (NODERED.ORG, 2017). Este modelo possui uma ótima representação visual, o que o torna mais acessível aos usuários,



### 5.4.2 Tipos de nós existentes no Node-RED

Existem três tipos de nós no Node-Red: nós de entrada, de saída e de processamento. Os dados são recebidos através dos nós de entradas, compilados e manipulados através dos nós de processamento e enviados através dos nós de saída, de acordo com a programação preestabelecida pelo usuário. Quando são conectados, forma-se uma rede de nós também conhecida como fluxo, conforme ilustrado na Figura 17. A maioria das aplicações utiliza estes três tipos de nós, porém há casos em somente dois deles (entrada e saída) são necessários



**Figura 17** – Nós existentes no Node-RED.  
(Fonte: Autores, 2018)

### 5.5 Dashboard

*Dashboard* pode ser entendido como um painel no qual as informações mais importantes para determinada aplicação podem ser facilmente visualizadas, possibilitando melhor monitoramento e análise dos dados. Estas informações geralmente são ajustadas em uma única tela a fim de concentrar a atenção do usuário num único ponto, trazendo maior agilidade ao processo.

No Node-RED, existe uma biblioteca chamada *node-red-dashboard* que proporciona a criação de uma interface web de fácil acesso com possibilidade de interação do usuário.

A Figura 18 representa uma *dashboard* implementada em uma aplicação com Arduino utilizando Node-RED.



**Figura 18** – Exemplo de dashboard.  
(Fonte: FLIPIPEFLOP, 2017)

A *dashboard* criada para o presente trabalho tem o objetivo de realizar uma integração entre os sistemas *software* e *hardware* através de uma interface, que irá conter gráficos indicando o que está sendo verificado e qual é o seu valor em tempo real, bem como botões que serão utilizados para comando.

## 5.6 Arduino

O Arduino é uma plataforma de *hardware* e *software* de código aberto que foi desenvolvido em 2005 por um grupo de cinco pesquisadores, sendo eles: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis.

Por todos esses anos, o Arduino tem sido o cérebro de milhares de projetos, de objetos do dia a dia à instrumentos científicos complexos. Por possuir código aberto, seus usuários podem montar placas de forma independente e adaptá-las as suas necessidades particulares. O *software* é, também, de código aberto e pode ser executado em diversas máquinas como Windows, Macintosh OSX e Linux, se tornando assim uma multiplataforma.

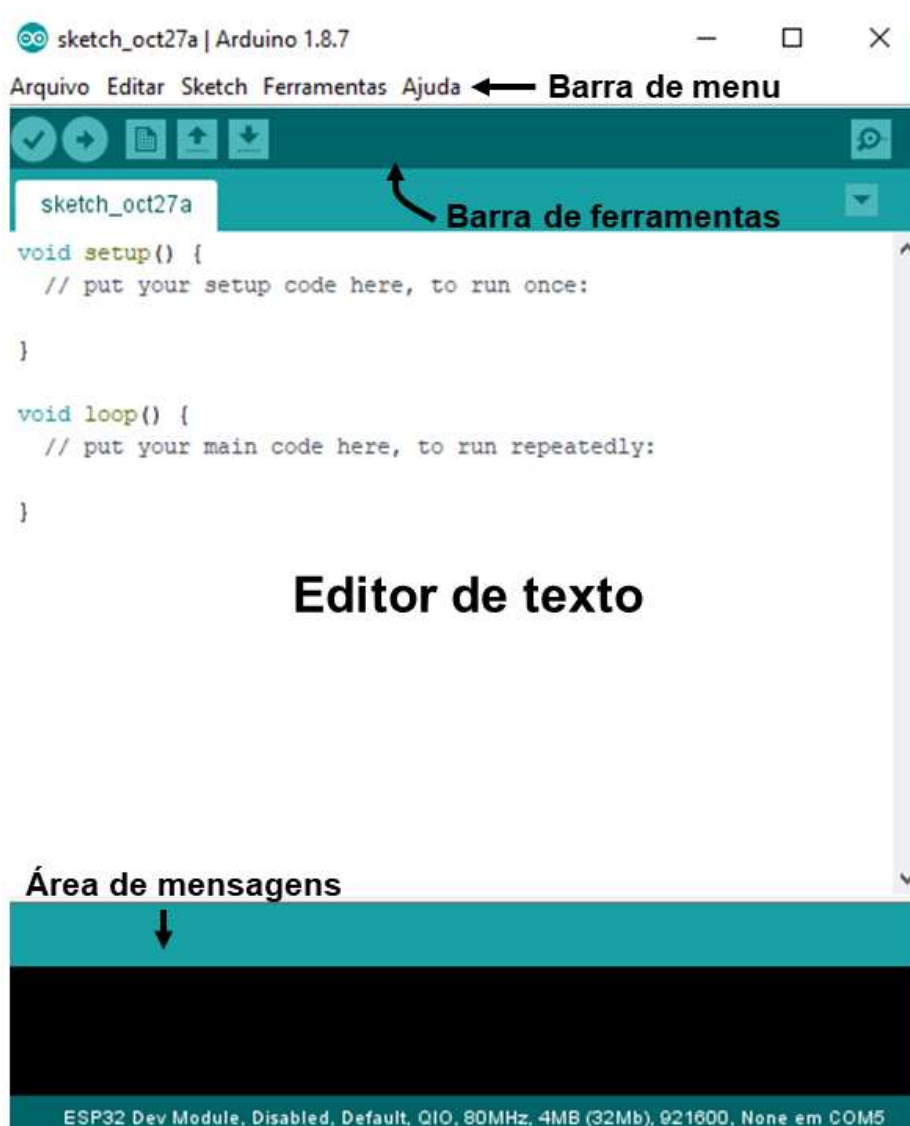
Neste trabalho o foco será apenas utilizar o *software* Arduino IDE, pois o *hardware* utilizado no projeto será o ESP32. Convém ressaltar então que toda programação desenvolvida na placa ESP32 será realizada nessa multiplataforma Arduino.

O *software* do Arduino IDE foi escrito em Java e seu desenvolvimento e linguagem foram desenvolvidos com base nos projetos *Processing* e *Wiring*. É considerado também um compilador de linguagens de programação GCC, possuindo então a capacidade de programar em C/C++.

A escolha deste *software* para desenvolver os códigos de programação que serão compilados no ESP32 se dá devido aos seguintes motivos: simplicidade da ferramenta; popularidade do programa; capacidade de se comunicar com os diversos dispositivos de automação e IoT que existem no mercado; e a variedade de bibliotecas, livros e artigos. Vale enfatizar também que a linguagem de programação é mais simples de ser entendida, a compilação é rápida e o programa gerado costuma ser leve.

### 5.6.1 Estrutura de programação do Arduino (IDE)

O ambiente de desenvolvimento integrado do Arduino, ou *software* Arduino (IDE), está ilustrado na Figura 19 e consiste basicamente em: uma barra de menus com diversos tipos de funções; uma barra de ferramentas com os principais botões de execução; um editor de texto para escrever o código de programação do programa, também conhecido como sketches ou esboços; e uma área de mensagens que serve para detectar possíveis erros ou exibir outras informações para o usuário do projeto.



**Figura 19** – Ambiente de desenvolvimento integrado do Arduino.  
(Fonte: Autores, 2018)

No momento em que o usuário for programar, ou seja, quando for escrever o seu código de programação no editor de texto do *software*, existem duas opções importantes a serem observadas:

- **setup()** – Inserida no início do programa, no qual é usado para definir as configurações de tudo o que você irá utilizar na sua lógica de programação;
- **loop()** – Onde toda a lógica e o controle da sua programação deve ficar escrito. Esta função executa de forma contínua, sem parar, fazendo com que a lógica de programação fique executando do início até o fim infinitas vezes.

Abaixo são ilustrados os botões existentes na barra de ferramentas mencionada anteriormente, com suas respectivas funções:



Verificar: Faz uma verificação no código do seu programa em busca de erros para depois compilá-los;



*Upload*: Compila o código do programa e faz o *upload* para o quadro configurado (porta/placa);



Novo: Cria um novo esboço, programa;



Abrir: Apresenta um menu de todos os esboços disponíveis;



Salvar: Grava seu programa;



Serial Monitor: Abre a janela do monitor serial e inicia a troca de dados com a porta/placa configurada.

Existem também alguns comandos adicionais que são encontrados na barra de menu com as seguintes abas: Arquivo, Editar, Esboço, Ferramentas e Ajuda. Elas possuem algumas funções simples como abrir, salvar e fechar um programa, mas também possuem funções específicas de selecionar a porta ou a placa que você está utilizando, incluir bibliotecas, acessar exemplos e compilar os programas.

Vale ressaltar que este *software* Arduino é compatível não só com o hardware Arduino, mas com diversos microcontroladores. Para realizar a

comunicação entre eles, basta apenas conecta-los através de um cabo USB ou fazer a comunicação via Bluetooth ou Wi-Fi.

## 6 COMPONENTES DO SISTEMA

Dentre os conhecimentos necessários para a compreensão do projeto, outro ponto importante é referente aos componentes utilizados. Este capítulo se dedicará a demonstrar os principais dispositivos empregados, suas especificações técnicas, seu funcionamento e a razão por terem sido escolhidos.

### 6.1 Node MCU-32S ESP32

Os ESP32 é um microprocessador de 32 bits com 4 MB de memória *flash* criado pela *Espressif Systems*, uma empresa chinesa focada no desenvolvimento de soluções WiFi de ponta, Bluetooth e IoT. Seus produtos mais populares são os chips ESP8266 e ESP32, seu sucessor.

Estes dois dispositivos destacaram-se no mercado IoT por possuírem placa WiFi integrada, além de serem de baixo custo e fácil acesso. O ESP32 é uma versão superior ao ESP8266. Seus principais diferenciais são: possuir conexão Bluetooth e maior quantidade de GPIO's.

O chamado NodeMCU é uma placa de desenvolvimento para o chip ESP8266 que se encontra soldada no módulo. Ou seja, toda a execução da programação, gerenciamento de memórias e conectividade WiFi são realizadas pelo ESP8266. Analogamente, foi criado o NodeMCU-32S, que é a plataforma de prototipagem própria para o ESP32, conforme ilustrado na Figura 20.

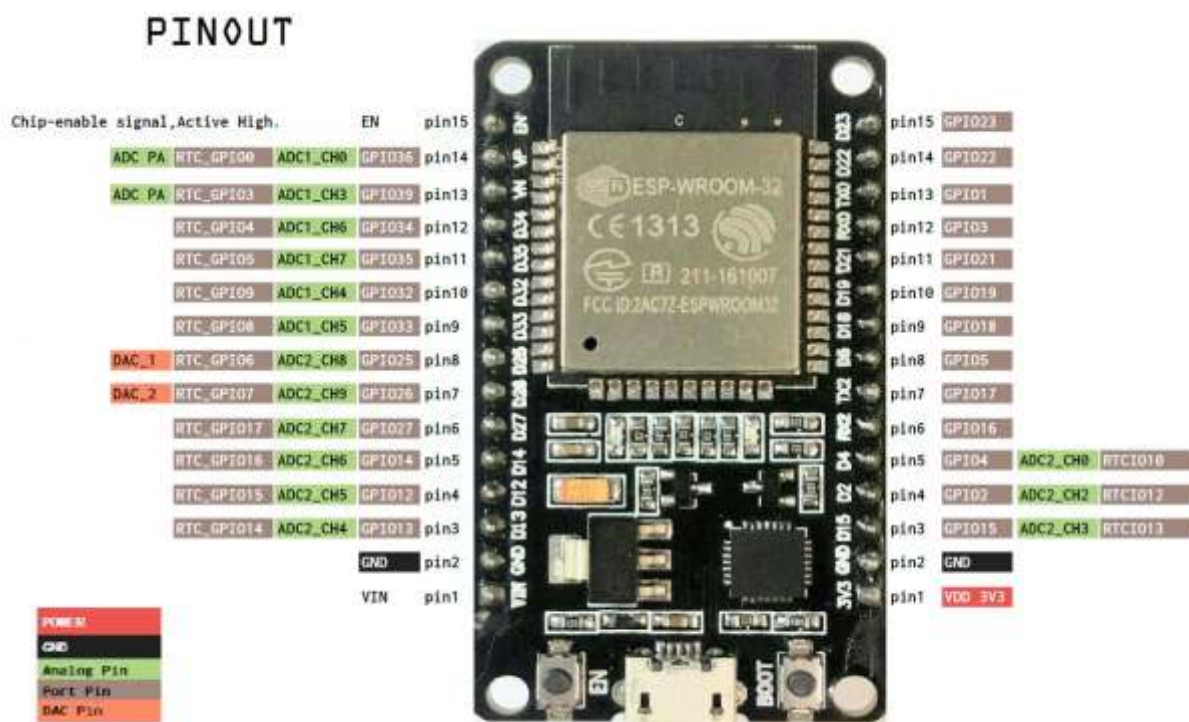


**Figura 20** – Node MCU-32S ESP32.  
(Fonte: ROBOCORE, 2018)

Esta plataforma é composta, além do chip, por circuitos de entrada e saída, uma interface USB serial e regulador de tensão 3,3 V. Sua arquitetura permite que ele possa ser programado de forma independente, sem a necessidade de utilizar microcontroladores como o Arduino, por exemplo. Pode ser facilmente conectada a um computador com auxílio de um cabo USB.

Seu funcionamento ocorre da seguinte forma: O ESP32 NodeMCU-32S recolhe dados do meio externo, processa internamente e realiza ações em resposta. A tensão de trabalho do ESP32 varia de 2,2 a 3,6 V, e quando operado por bateria, pode variar entre 2,8V e 3,6 V. O consumo pode chegar até 225 mA de pico nas transmissões, o que torna as pilhas alcalinas uma boa alternativa como fonte de alimentação.

Na Figura 21 é mostrado a pinagem do Node MCU-32S ESP32.



**Figura 21** – Pinagem do Node MCU-32S ESP32.  
(Fonte: Adaptado de LELONG, 2018)

A placa acima possui 25 GPIO's, que são portas programáveis de entrada e saída de dados digitais, utilizadas para realizar uma interconexão entre os periféricos e os microcontroladores/microprocessadores. Além destas, existem também 15 portas chamadas ADC\_CH (*analog-to-digital converter*), que têm como objetivo principal converter valores analógicos de tensão em valores digitais. Dessa

forma, pode-se inserir uma grandeza analógica e a mesma será convertida para um valor digital. O contrário ocorre nas 2 portas existentes chamadas DCA\_CH (*digital-to-analog converter*).

O NodeMCU-32S ESP32 foi escolhido para o desenvolvimento do projeto por diversas razões. Uma delas é a capacidade de atuar sozinho, sem a necessidade de inserir complementos para exercer suas funções, conforme já explicado anteriormente. Além disso, uma das premissas do projeto é a conectividade com a internet e por isso a conexão WiFi é imprescindível. Possuir WiFi integrado é uma grande vantagem, uma vez a aquisição de módulos separados para realizar esta função geraria um alto custo, além de aumentar a quantidade de equipamentos necessários. Outras vantagens são: baixo consumo de energia, alto desempenho de potência, amplificador de baixo ruído, robustez, versatilidade e confiabilidade. Por fim, é importante citar que este dispositivo pode ser programado em mais de 10 linguagens diferentes, entre elas linguagens populares como C++, Lua e Python.

Vale destacar que nos próximos capítulos o módulo NodeMCU-32S ESP32 será referenciado como ESP32 afim de fazer uma menção mais clara, simples e de fácil entendimento ao leitor.

## **6.2 Sensores**

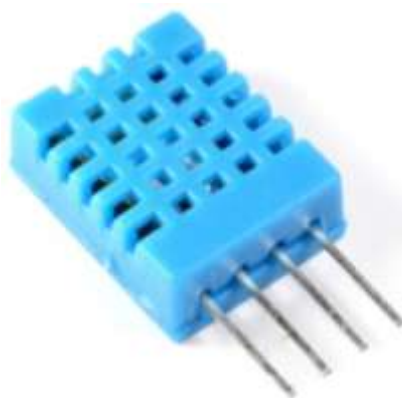
O sensor é um dos principais dispositivos da IoT, pois ele é uma parte integrante de algum tipo de sistema de aquisição de dados. Este recebe um estímulo e responde com um sinal elétrico. Por conseguinte, o principal propósito de um sensor é responder a uma propriedade física de entrada e convertê-la em um sinal elétrico compatível com os circuitos eletrônicos que estão na sua saída. O sinal de saída de um sensor pode ter: variação de tensão, variação de corrente, variação de resistência, modificação de uma das características de uma onda (amplitude, frequência ou fase), ou até mesmo ser um sinal digital.

A seguir serão mencionados os sensores utilizados no projeto, bem como as suas principais características e funcionamento, sendo estes: sensor de temperatura e umidade, sensor de gás, sensor de fogo, sensor de tensão e sensor de corrente.

### 6.2.1 Temperatura e umidade

Diversos segmentos da indústria e comércio em geral realizam o controle de determinadas variáveis a fim de manter a qualidade de seus serviços e produtos. Um dos principais monitoramentos realizados nesses setores diz respeito às condições climáticas do ambiente, ou seja, os valores de temperatura e umidade. Essas variáveis podem ser muito essenciais em processos industriais onde existe a manipulação de alimentos, manufaturas, cultivo de plantas e animais. Pode-se verificar, porém, que a utilização não se restringe apenas ao uso industrial. Atualmente, há uma grande preocupação em promover o conforto térmico das pessoas em suas residências ou locais de trabalho a fim de garantir o bem estar e o desenvolvimento pleno das suas atividades.

Para realizar este controle, o mercado oferece diferentes tipos de sensores, desde aqueles que fazem apenas o controle da temperatura ou da umidade separadamente, até os que possuem as duas funções integradas, como é o caso do sensor **DHT11** ilustrado na Figura 22.

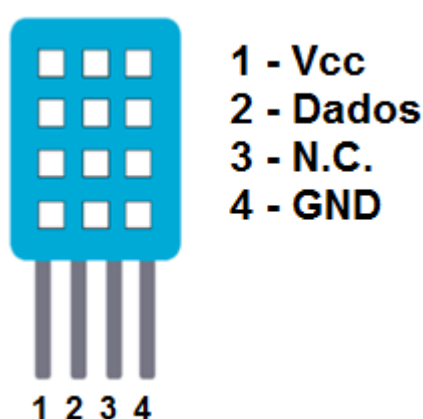


**Figura 22** – Sensor DHT11.  
(Fonte: FILIPEFLOP, 2018)

Este foi o sensor escolhido para o protótipo. Um dos principais pontos que levaram a esta escolha foi sua concepção simples e seu baixo custo. Além disso, levou-se em consideração também o fato de que este sensor é muito utilizado em aplicações utilizando plataformas microcontroladas, como o ESP32. Internamente, o mesmo é composto por um elemento termistor do tipo NTC para medir a temperatura e um elemento do tipo HR202 para medir a umidade. Pode medir temperaturas entre 0 a 50°C e umidade entre 20 a 90% UR, com precisão de  $\pm 2,0^\circ\text{C}$  e  $\pm 5,0\%$  UR, respectivamente. Deve ser alimentado com tensões entre 3 a 5

Vcc. Sua corrente varia de 200 uA a 500 mA (em *stand by* de 100 uA a 150 uA) e seu tempo de resposta é de 2 segundos.

Conforme ilustrado na Figura 23, este sensor possui 4 pinos, sendo dois deles utilizados para a alimentação (Vcc e GND), um para dados e o outro considerado como não conectado. Em virtude disso, o mesmo faz a leitura e o envio dos valores de temperatura e umidade para o microcontrolador através de um único pino. Vale ressaltar que estes dados são enviados em formato digital.



**Figura 23** – Pinos do sensor DHT11.  
(Fonte: FILIPEFLOP, 2018)

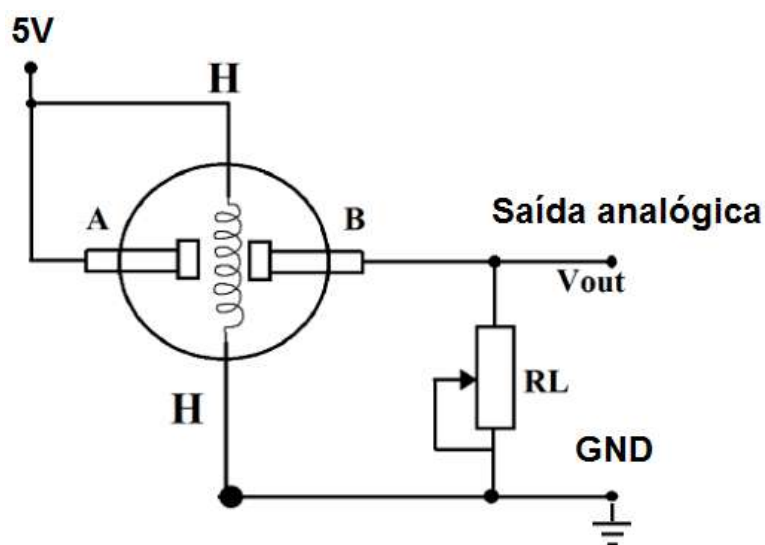
### 6.2.2 Sensor de gás

A aplicação de sensores de gás e fumaça em sistemas inteligentes é cada vez mais comum e vem tornando-se um tema de grande importância na atualidade. Na área residencial, a principal utilização está ligada à prevenção de incêndio (devido ao gás de cozinha), na qual é possível inclusive evitar a inalação do gás GLP por parte do morador. Na área industrial, a aplicação varia de acordo com cada sistema. Para alguns processos específicos, monitorar a concentração de gás ou fumaça pode ser crucial, pois a ocorrência de vazamento pode gerar grandes problemas, tanto para a planta como para seus funcionários e cidadãos que residem nas proximidades.

O mercado oferece diversos tipos de sensor nesta área, dentre os quais é possível citar: sensor catalítico, infravermelho, open-path, semicondutor e eletroquímico. A escolha do mesmo deve ser realizada de acordo com a necessidade do usuário. Para o presente projeto, optou-se por utilizar sensores do

tipo catalítico da família MQ. Estes são utilizados em pequenos projetos por possuírem baixo custo e terem a capacidade de detectar gás combustível.

Através da Figura 24 é possível entender a concepção interna destes sensores. Basicamente, há uma resistência elétrica ligada nos pinos H's (alimentação do sistema -5 V e GNV) e um elemento sensor localizado de A para B. A resistência varia de acordo com a concentração de gás poluente. Quanto maior a concentração, menor a resistência e maior tensão em cima do resistor RL. Ou seja, a tensão de saída é proporcional à quantidade de gases poluentes.

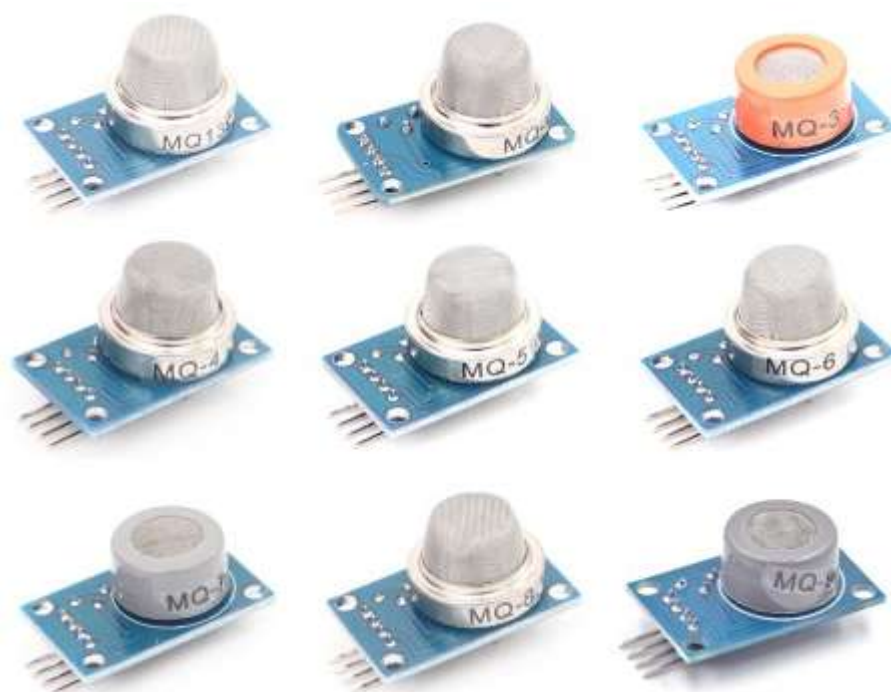


**Figura 24** – Esquema interno de um sensor de gás MQ.  
(Fonte: VIDA DE SILICIO, 2017)

Os sensores da família MQ possuem a aparência mostrada na Figura 25. Porém, para facilitar sua utilização, os mesmos são vendidos já integrados a uma placa, vide Figura 26. Esta possui os seguintes componentes: comparador LM393, LED indicador para tensão, LED indicador para saída digital e potenciômetro.



**Figura 25** – Encapsulamento de um sensor de gás MQ.  
(Fonte: VIDA DE SILICIO, 2017)



**Figura 26** – Família de sensores de gás MQ.  
(Fonte: VIDA DE SILÍCIO, 2017)

A Tabela 1 mostra as respectivas aplicações para cada sensor da linha MQ. Com base nestas informações, determinou-se que o sensor a ser utilizado no projeto seria o **MQ-2**, devido a sua aplicabilidade e custo-benefício (é o sensor mais barato e atende a necessidade dos autores).

**Tabela 1 - Sensores a gás da família MQ**

Tipo	Sensível para
MQ-2	Detecção de gases inflamáveis: GLP, metano, propano, butano, hidrogênio, álcool, gás natural, outros inflamáveis e fumaça.
MQ-3	Detecção de álcool, etanol e fumaça.
MQ-4	Detecção de metano, propano e butano.
MQ-5	Detecção de GLP e gás natural.
MQ-6	Detecção de gás GLP, propano, isobutano e gás natural liquefeito.
MQ-7	Detecção do gás monóxido de carbono.
MQ-8	Detecção do gás hidrogênio.
MQ-9	Detecção de monóxido de carbono e gases inflamáveis.
MQ-131	Detecção de ozônio.
MQ-135	Detecção de gás amônia, óxido nítrico, álcool, benzeno, dióxido de carbono e fumaça.
MQ-136	Detecção de gás sulfídrico H <sub>2</sub> S.
MQ-137	Detecção de gás amônia.
MQ-138	Detecção de n-hexano, benzeno, NH <sub>3</sub> , álcool, fumaça, CO, etc.

(Fonte: VIDA DE SILÍCIO, 2017)

A tensão de operação do MQ-2 é 5 Vcc e pode ser aplicado em ambientes com temperaturas entre -20 a 50 °C. Possui uma saída analógica (A0) e uma saída digital (D0). Seu nível de detecção é de 300 a 10.000 ppm (partes por milhão) e pode ser ajustado pelo potenciômetro localizado na placa. Quando a concentração de gás é superior ao nível ajustado pelo potenciômetro, a saída digital D0 fica em estado baixo. Caso contrário, o estado será alto. Além disso, para obter melhor resolução, é possível realizar também a medição através da saída analógica A0, conectando-a a um conversor AD, como presente em alguns microcontroladores como o ESP32. Os detalhes construtivos mencionados acima podem ser observados na Figura 27.

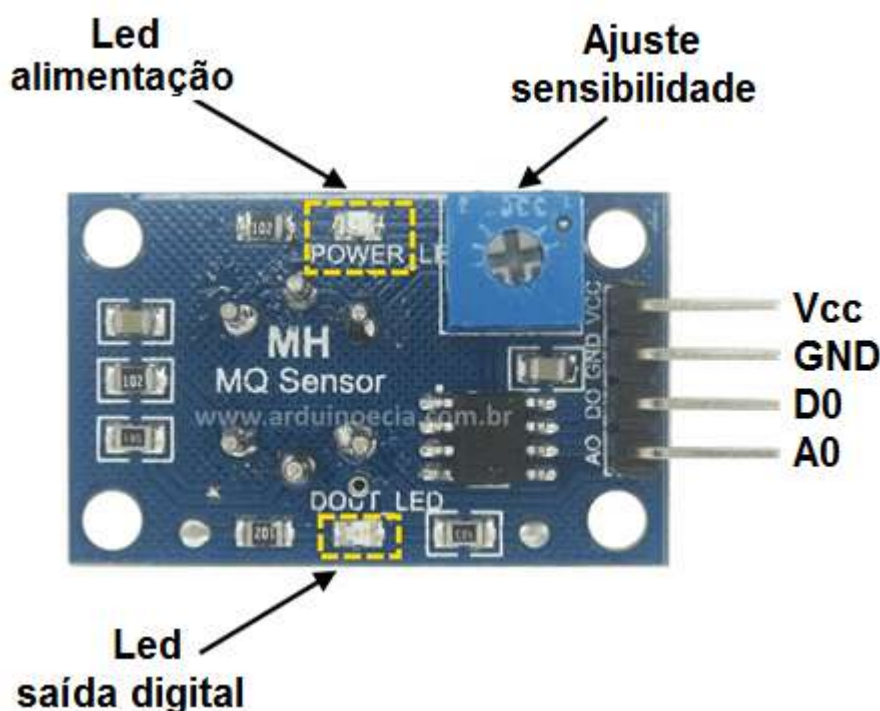


Figura 27 – Sensor MQ-2 - parte inferior.  
(Fonte: ARDUINO E CIA, 2015)

### 6.2.3 Sensor de fogo

O sensor de fogo, assim como o de gás, também tem grande relevância no conceito de edifícios inteligentes. Sua utilização tem como ponto principal garantir a segurança das pessoas e preservar a infraestrutura da construção, evitando maiores desastres na ocorrência de um possível incêndio.

Para o protótipo, utilizou-se o sensor de **Chama Fogo 760 a 1.100 nm**, um módulo eletrônico desenvolvido com a finalidade de detectar a presença de

fogo/chama/calor em um ambiente, vide Figura 28. A partir dessa identificação, o microcontrolador que estiver ligado ao módulo será notificado e, de acordo com a programação preestabelecida, uma ação é tomada. Como exemplo, pode-se citar o envio de notificações, o acionamento de um alarme ou de um sistema inibidor de incêndio.



**Figura 28** – Sensor Chama Fogo 760 a 1100 nm.  
(Fonte: ELETROGATE, 2018)

Este sensor é capaz de detectar o espectro de chama/fogo/calor que possua uma onda de largura de 760nm até 1.100 nm e que esteja a uma distância de até 80 cm. Além disso, possui um ângulo de detecção que pode chegar até 60°.

A Figura 29 mostra os componentes do módulo. É composto de 3 pinos, sendo eles: dois pinos de alimentação (Vcc e GND) e um pino correspondente à saída digital (D0). Possui também dois LED's, sendo um para a indicação de alimentação e o outro para a saída digital; um sensor IR para a detecção; e um potenciômetro responsável pelo o ajuste de sensibilidade do sensor.



**Figura 29** – Sensor Chama Fogo 760 a 1100 nm - parte inferior.  
(Fonte: Adaptado de FILIPEFLOP, 2015)

A leitura da saída deste sensor é realizada apenas na forma digital, cujo sinal é representado pelo nível alto (1) ou nível baixo (0). Quando o sensor IR detectar um valor de calor acima do pré-ajustado, a saída digital D0 fica em estado baixo. Caso contrário, o estado será alto.

É interessante salientar que este sensor não é capaz de identificar as chamas que são geradas por alguns tipos de gases, como, por exemplo, o gás natural e o GLP, sendo necessário utilizar um sensor de gás em conjunto com o sensor chama fogo.

Há de se considerar que este sensor é de fácil instalação, possui um baixo custo, é eficiente e compatível com diversos tipos de microcontroladores, como Arduino e ESP32, sendo uma das principais razões para utilizá-lo neste projeto.

#### 6.2.4 Sensor de tensão

A medição da tensão é fundamental para a análise do consumo de energia, como é o caso do presente trabalho. Isto porque, caso a mesma não fosse realizada e o grupo adotasse como padrão a tensão da rede (110 ou 127 V), o cálculo do consumo não seria fidedigno uma vez que a tensão não seria a real.

Esta pode ser medida através de um divisor de tensão ou através de um transformador de potencial, cuja utilização mais comum se dá em aplicações que possuem tensões elevadas (maior que 500V).

O sensor escolhido para esta função foi o **sensor de tensão AC 0 a 250Vac – ZMPT101B**, comprado já montado no módulo conforme Figura 30.

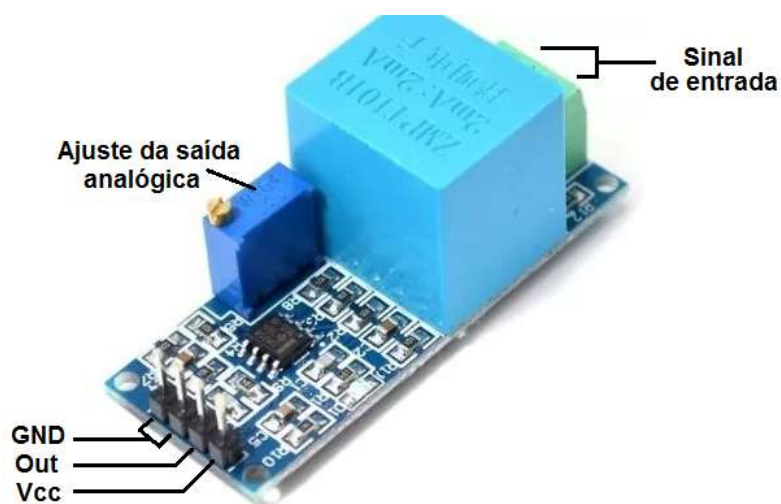


**Figura 30** – Sensor de tensão AC 0 a 250Vac ZMPT101B.  
(Fonte: CURTO CIRCUITO, 2018)

Além de realizar medições de 0 a 250 Vca, este sensor também pode ser utilizado apenas como detector de tensão, identificando se uma luz está acesa ou apagada, se um botão está ligado ou desligado, etc. Neste caso, é enviado um sinal digital para o microcontrolador: quando há detecção de tensão, tem-se estado alto; quando não há detecção, tem-se estado baixo.

É constituído basicamente pelo transformador ZMPT101B, que possui a função de limitar a corrente em 2 mA, e um circuito contendo amplificador operacional para corrigir o deslocamento da saída analógica. Possui também um potenciômetro que é responsável pelo ajuste da saída analógica. Pode ser alimentado com tensões de alimentação entre 5 a 30 Vcc e a temperatura de operação varia entre -40 a 70 °C. O sinal de saída é analógico e apresenta uma relação linear com relação à tensão de entrada. Além disso, é importante ressaltar que este sensor possui uma precisão de leitura de  $\pm 0,5\%$ .

De uma forma mais clara é possível visualizar abaixo o sensor utilizado. Conforme Figura 31, o módulo é composto por 4 pinos, sendo: três de alimentação (Vcc e GND) e um de saída; possui um conector no qual são inseridos os cabos para medição (representado pelo sinal de entrada) e também um ajuste da saída analógica, conforme mencionado anteriormente.



**Figura 31** – Sensor de tensão AC 0 a 250Vac ZMPT101B (pinagem).  
(Fonte: Adaptado de MERCADO LIVRE, 2018)

### 6.2.5 Sensor de corrente

De forma análoga ao sensor de tensão abordado anteriormente, o sensor de corrente foi utilizado no projeto com o intuito de realizar a medição da corrente elétrica no circuito para possibilitar a análise do consumo de energia.

Existe uma diversa gama de sensores de corrente no mercado, divididos basicamente em: sensores invasivos e não invasivos. Para utilizar um sensor invasivo, é necessário abrir o circuito e conectá-lo em série com a carga, enquanto o não invasivo não requer a abertura do circuito, bastando apenas envolver o condutor onde deseja-se realizar a medição no orifício do sensor em questão. Esta é uma grande vantagem, pois a medição da corrente ocorre sem necessidade de contato elétrico com o circuito. Sua aplicação é voltada principalmente para o monitoramento de corrente elétrica ou para proteções de circuitos, como por exemplo, motores e iluminação.

Os sensores de corrente não invasivos utilizam propriedades magnéticas para que a medição seja realizada e podem ser considerados um transformador de corrente. Estes são divididos em 2 modelos, mostrados nas Figuras 32 (a) e (b): *split-core* (núcleo dividido), não possui possibilidade de abertura para passagem do cabo; e *solid-core* (núcleo sólido), possui possibilidade de abertura para passagem do cabo.



**Figura 32** – Transformadores de corrente: (a) Modelo *solid-core*; (b) Modelo *split-core*.  
(Fonte: VIDA DE SILICIO, 2017)

Devido à maior facilidade de conexão em circuitos e à sua versatilidade, optou-se por utilizar o modelo *split-core*. Um dos sensores mais comuns deste modelo são os chamados SCT-013, da Marca YHDC. SCT é a sigla para *split-core current transformer* (transformador de corrente de núcleo dividido). Existem inúmeros

modelos desta família, que se distinguem basicamente pelo valor de corrente máxima a ser medida e o tipo de saída do sensor (podendo ser tensão ou corrente).

No presente projeto, optou-se por utilizar o modelo **SCT-013-020**, ilustrado na Figura 33, que pode medir corrente de até 20 A. Seu sinal de saída é através da tensão, possuindo uma relação de espiras de 20A/1V, ou seja, quando está circulando uma corrente alternada de até 20 A no condutor, o sensor fornece uma tensão de saída de 1 V. Em virtude disso, o mesmo lê apenas corrente alternada e fornece na saída uma tensão alternada senoidal correspondente a corrente medida.

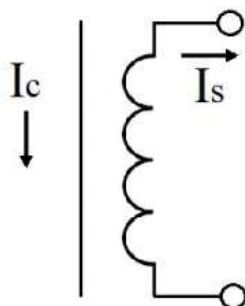


**Figura 33** – Sensor de corrente SCT-013-020.  
(Fonte: FILIPEFLOP, 2018)

Seu princípio de funcionamento utiliza três conceitos básicos: a Lei de Ampère, características da corrente alternada e a Lei de Faraday. Segundo a Lei de Ampère, todo o condutor energizado induz ao seu redor um campo magnético proporcional à corrente; A corrente alternada tem como característica sua forma de onda senoidal, variando sua intensidade ao longo do tempo. Como o campo magnético é proporcional à corrente, seu valor também irá oscilar ao passar do tempo; Já pela Lei de Faraday, um campo magnético variante ao longo do tempo induz uma força eletromotriz gerando, conseqüentemente, uma corrente elétrica proporcional à intensidade do campo magnético.

Estas três propriedades explicam a atuação do transformador de corrente, que é basicamente um conjunto de espiras inseridas ao redor de um condutor no qual se deseja medir a corrente. Internamente, o sensor possui em seu interior um

enrolamento de bobinas e um núcleo magnético. Como exemplo deste projeto, o enrolamento primário é o condutor utilizado para a medição no circuito, onde ele é envolvido pelo núcleo do TC, e o enrolamento secundário são as espiras alojadas no interior da caixa do sensor de corrente, vide figura 34.



**Figura 34** – Esquema para exemplificar o SCT-013.  
(Fonte: VIDA DE SILICIO, 2017)

Através da figura acima é possível constatar que o transformador de corrente terá em seus polos uma corrente alternada induzida  $I_s$ , que é considerada proporcional a corrente alternada  $I_c$  que percorre o condutor ao qual pretendemos fazer a medição de corrente elétrica. Além disso, a corrente induzida do TC será inversamente proporcional ao número de espiras que existe no interior do sensor. As relações de transformação e o cálculo da corrente dos transformadores pode ser feito baseado nas Equações 6.1 e 6.2 a seguir:

$$\text{Relação de espiras} = \frac{N^\circ \text{ de espiras no primário}}{N^\circ \text{ de espiras no secundário}} \quad (6.1)$$

$$\text{Corrente no secundário} = \text{Corrente no primário} \times \text{Relação de espiras} \quad (6.2)$$

### 6.3 Atuador

Enquanto os sensores são responsáveis pela aquisição de dados, os atuadores são utilizados para realizar determinadas ações em um sistema sendo necessário apenas estabelecer comandos para a sua atuação. Existem diversos tipos de atuadores no mercado, sejam aqueles que possuem a função de produzir movimentos, como exemplo, cilindros e motores, ou qualquer outro elemento que realize uma operação no sistema com base em uma programação preestabelecida por um dispositivo.

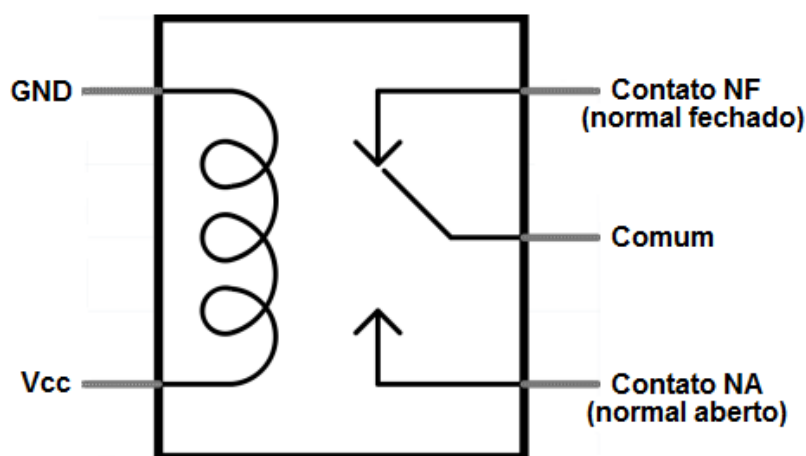
Neste trabalho, apenas um atuador será utilizado: o módulo relé. A seguir será explicado suas características e seu funcionamento.

### 6.3.1 Módulo relé

Os sistemas inteligentes geralmente demandam um controle de cargas de diferentes potências, como resistências, lâmpadas, motores etc. O dispositivo mais utilizado para realizar a função de ligar e desligar estas cargas são os relés.

Um relé é um componente eletromecânico, ou seja, possui uma parte elétrica e outra mecânica. É semelhante a um interruptor, com a possibilidade de ser ligado ou desligado, permitindo ou interrompendo a passagem da corrente elétrica.

A Figura 35 ilustra o esquema interno do relé. Existem dois pinos conectados a uma bobina, onde é ligado a alimentação – GND e Vcc. Além disso, há três pinos de saída: contato NF, comum, contato NA. Quando a bobina não está energizada (relé “desligado”), o contato NF permanece fechado e o contato NA permanece aberto, ou seja, o comum está conectado ao NF. Quando a bobina é energizada (relé “ligado”), o contato NF é aberto e o contato NA é fechado – comum conectado ao NA. Isto ocorre devido ao campo eletromagnético criado pela bobina. Vale ressaltar que a bobina é 100% isolada dos contatos, ou seja, o relé possibilita a isolamento dos circuitos, garantindo, assim, proteção e segurança.



**Figura 35** – Esquema de ligação interna de um relé.  
(Fonte: AUTOCORE ROBOTICA, 2018)

Os relés são muito utilizados atualmente já montados em uma placa – denominados módulos relés. Estes fornecem um circuito auxiliar para acionamento

do relé e proteção dos microcontroladores, como Arduino e ESP32, facilitando sua utilização.

O mercado oferece inúmeros modelos de módulos relés, que geralmente variam de acordo com seu número de canais. Para o presente projeto, optou-se por utilizar o módulo relé de 2 canais, conforme Figura 36, devido à demanda do sistema.



**Figura 36** – Módulo relé de 2 canais.  
(Fonte: FILIPEFLOP, 2018)

Este módulo possui tensão de operação de 5 Vcc e pode controlar cargas de até 220 Vac. Sua corrente de operação varia entre 15 e 20 mA, podendo suportar cargas de até 10 A em até 250 Vca ou 30 Vcc. Seu tempo de resposta varia entre 5 a 10 ms. Possui LED's que indicam seu estado (ligado ou desligado) em cada canal. Um detalhe importante é que os relés são ativados em nível baixo, ou seja, quando o estado da porta estiver em LOW, o relé será acionado.

Possui também dois optoacopladores, sendo um para cada canal, utilizados para isolar partes de um circuito, principalmente ao trabalhar com diferentes tensões. Como não possui contato elétrico, nem mecânico, funciona como um sistema de segurança, protegendo o microcontrolador em caso de descargas elétricas.

A Figura 37 abaixo representa a pinagem do módulo relé utilizado. Os pinos JD-Vcc, Vcc e GND, localizados no canto superior esquerdo, permitem que a placa seja conectada a uma fonte externa para alimentação do optoacoplador,

possibilitando a proteção do circuito. Os pinos GND e Vcc, localizados no canto inferior esquerdo, são responsáveis pela alimentação dos relés, enquanto os pinos IN1 e IN2 são referentes às entradas dos relés: IN1 aciona o relé 1 e IN2 aciona o relé 2. Por fim, os contatos NF, comum e NA, localizados à direita, são responsáveis pelo acionamento da carga.



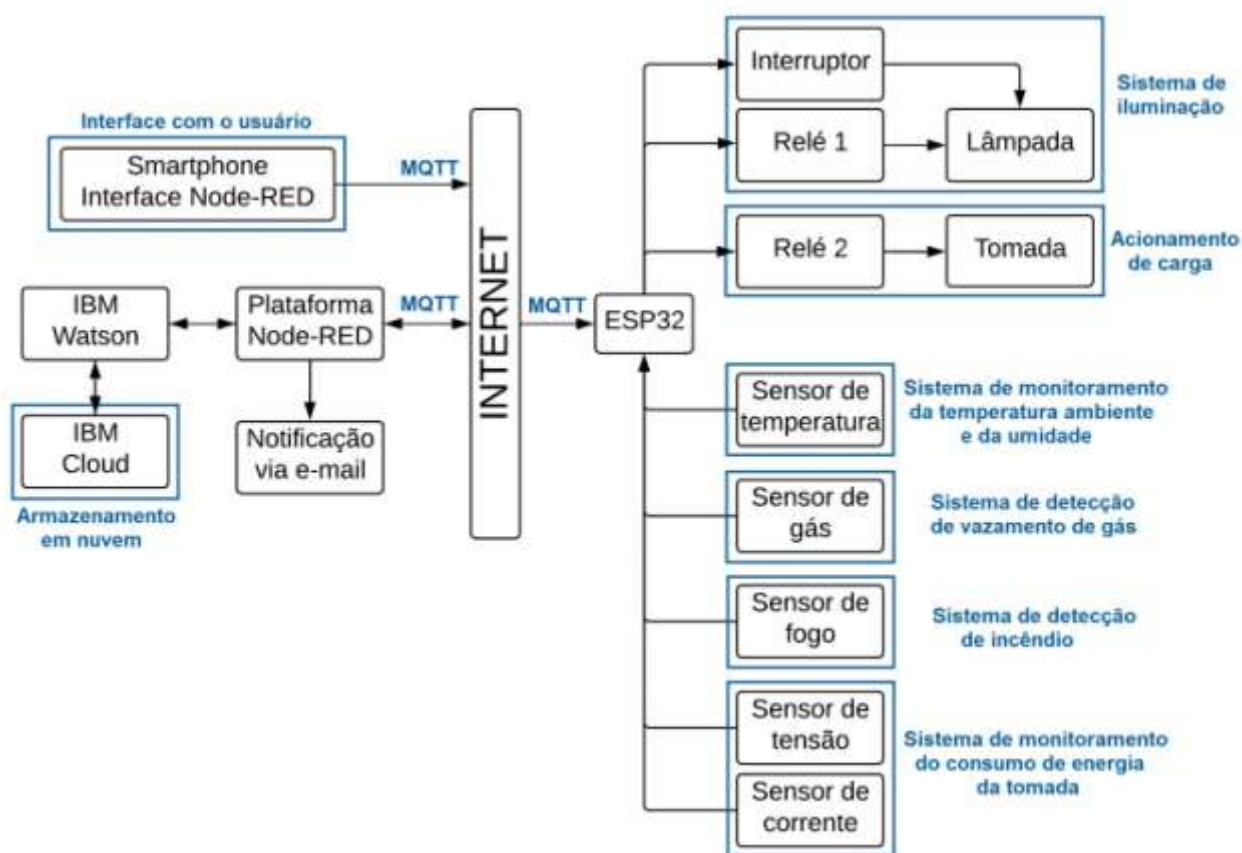
**Figura 37** – Pinagem do módulo relé de 2 canais.  
(Fonte: FILIPEFLOP, 2013)

## 7 ETAPAS DE DESENVOLVIMENTO DO PROJETO

A ideia inicial do projeto consiste em demonstrar as diversas aplicações existentes atualmente no campo da IoT, bem como as possibilidades de serem implementadas em qualquer tipo de edifício, sejam estes de uso residencial, comercial, industrial, entre outros.

Pensando nisso, a proposta consistia na elaboração de um protótipo que, apesar de possuir uma concepção simples, é capaz de executar uma variedade de funções que podem ser exploradas e potencializadas em grande escala.

Para melhor compreensão do funcionamento do projeto, foi elaborado o diagrama de blocos representado na Figura 38 onde é possível visualizar os componentes, serviços e protocolo utilizados.



**Figura 38** – Diagrama de blocos das etapas do projeto.  
(Fonte: Autores, 2018)

Este diagrama pode ser dividido em duas partes, sendo elas: a parte física, representada pelo microcontrolador, sensores e atuadores; e a parte virtual, representada pelos serviços, protocolo e interface *web*.

As principais funcionalidades do sistema são:

- Controle de iluminação;
- Controle do acionamento de uma carga;
- Monitoração de temperatura e umidade;
- Detecção de gás;
- Detecção de incêndio;
- Medição do consumo de energia da carga.

A seguir será apresentado o detalhamento do projeto. As partes constitutivas podem ser subdivididas em: montagem do circuito físico, programação na IDE do Arduino, cadastro na nuvem, programação no Node-Red e protótipo finalizado.

## **7.1 Montagem do Circuito Físico**

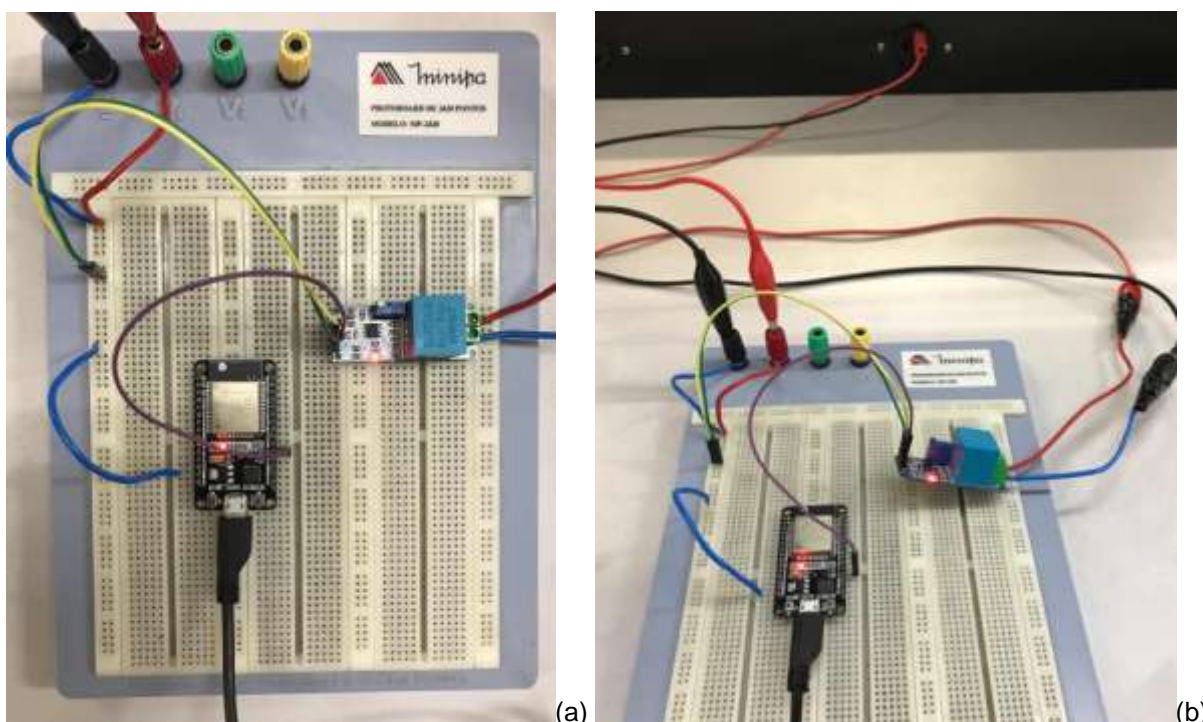
### **7.1.1 Testes nos sensores**

Antes de inserir os sensores na configuração final do projeto, é necessário realizar testes individuais com o intuito de verificar o correto funcionamento dos dispositivos. Neste ponto, o laboratório de Eletrônica da UniFOA foi de fundamental importância para a utilização de instrumentos de medição, bem como demais componentes necessários para a conclusão desta etapa.

#### **7.1.1.1 Sensor de tensão**

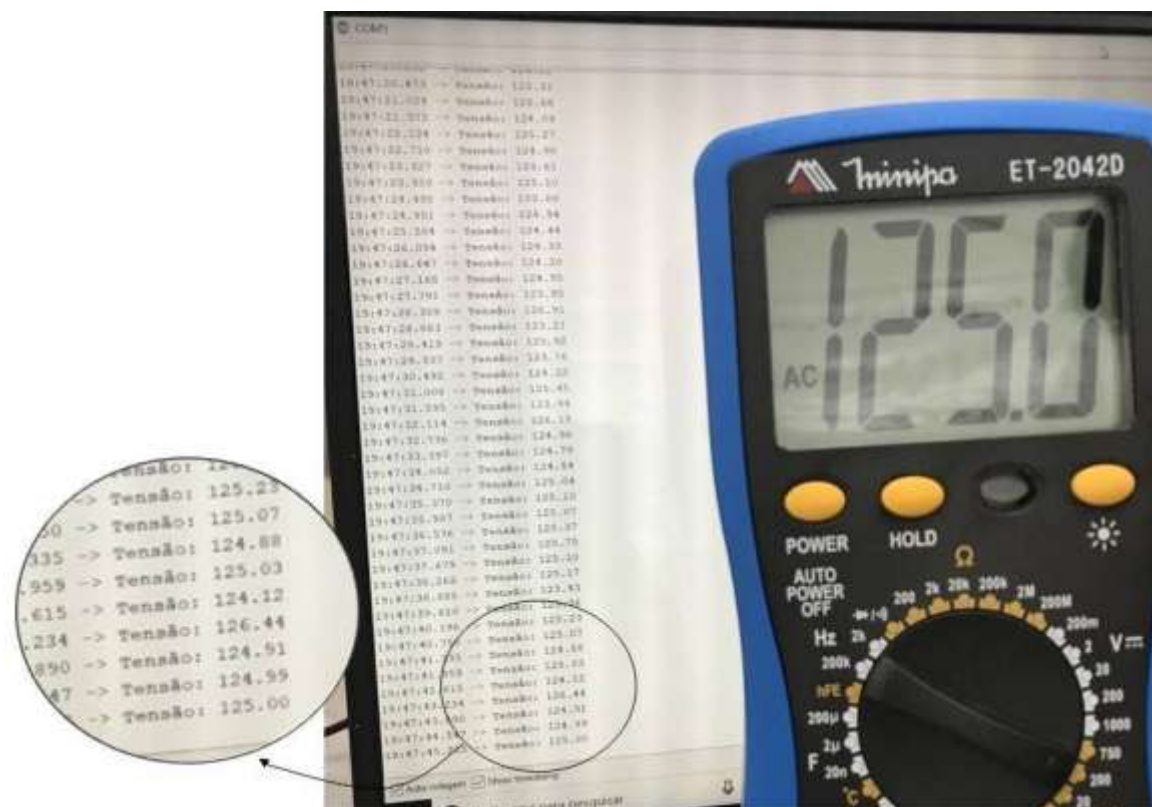
A ideia inicial era realizar a medição em uma das tomadas existentes na bancada do laboratório, que disponibilizam tensão de 127 V. O ESP32 foi alimentado via cabo USB conectado diretamente no computador, para visualização dos valores lidos pelo sensor, que, por sua vez, foi alimentado através de uma fonte externa de 5 Vcc. Para equalizar os referenciais de tensão dos dispositivos, o GND da fonte foi conectado ao GND do ESP32. Esta ligação é necessária sempre que temos uma diferença de potencial entre as alimentações de um sensor/atuator e do microcontrolador conectado aos mesmos. Esta ligação foi realizada para todos os testes e não será mais citada ao longo do capítulo.

A montagem do circuito pode ser visualizada nas Figuras 39 (a) e (b). A ligação do sensor foi realizada da seguinte forma: em sua entrada, foram conectados dois fios (vermelho e azul) que fazem a conexão com a tomada. O pino correspondente a alimentação positiva é representado através do fio verde, o da alimentação negativa pelo fio amarelo e a saída pelo fio roxo, interligado ao ESP32 através da GPIO 4, uma porta ADC (*analog digital converter*), que possui a característica de converter o sinal analógico para digital com precisão de 12 bits.



**Figura 39** – Teste do sensor de tensão. (a) Circuito montado na *protoboard* visto de cima. (b) Circuito completo - medição da tomada.  
(Fonte: Autores, 2018)

Através do código registrado no ESP32, realizou-se a medição da tensão RMS da tomada, conforme ilustrado na Figura 40. Com os resultados obtidos, pode-se concluir que o valor lido pelo sensor está de acordo com o esperado, uma vez que a tensão da rede é 127 V. Para conferir o valor exato de alimentação da bancada, foi utilizado um multímetro que apresentou a mesma leitura do sensor. É importante ressaltar que isso não ocorre em todos os casos, visto que o sensor de tensão utilizado possui uma margem de precisão  $\pm 0,5\%$ , podendo haver uma pequena diferença entre o valor medido e o valor lido.



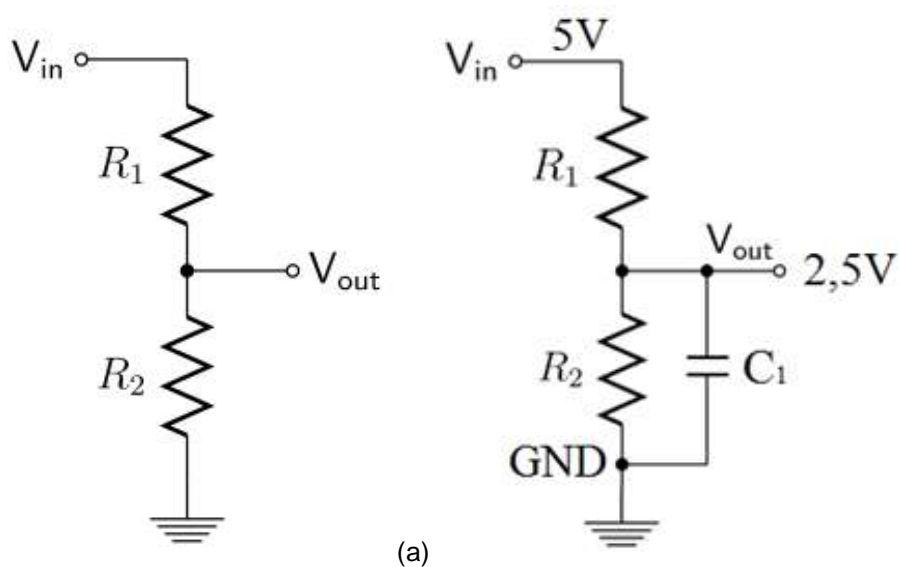
**Figura 40** – Comparação dos valores medidos pelo sensor de tensão e multímetro.  
(Fonte: Autores, 2018)

### 7.1.1.2 Sensor de corrente

Para o sensor de corrente, primeiramente foi necessário determinar uma carga de teste. Um ferro a vapor foi escolhido, devido à sua potência máxima, que, segundo o fabricante (Black+Decker), é 1200 W. Porém, antes de realizar o experimento, foi necessário ajustar o sinal de saída do sensor.

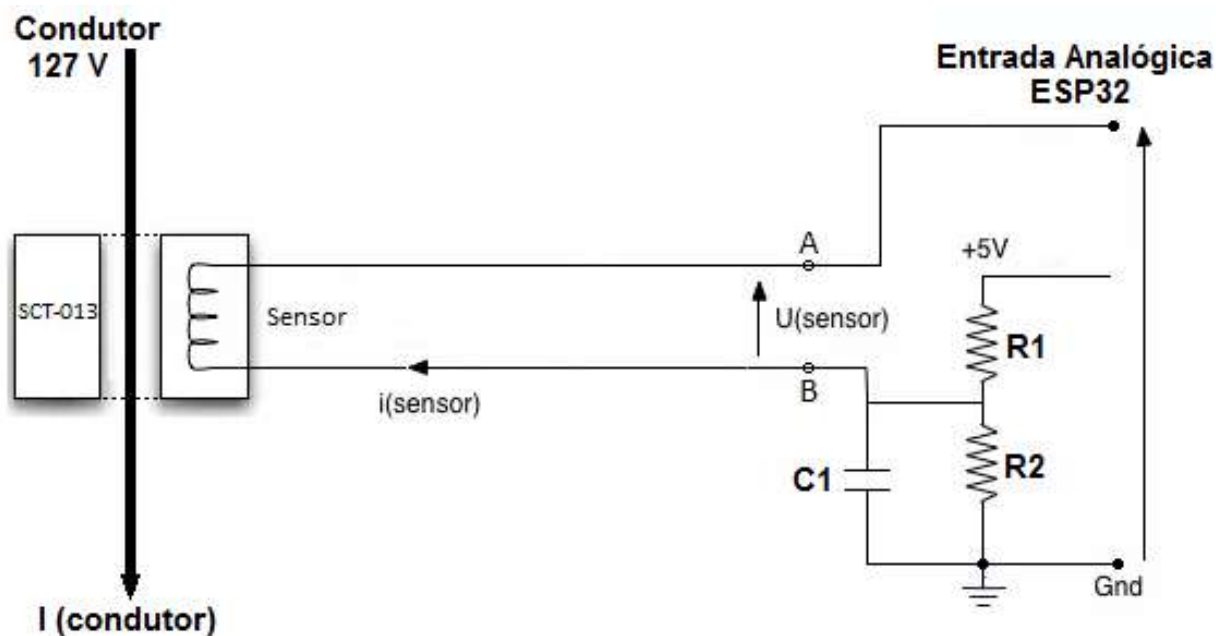
A saída do sensor de corrente é dada em forma de tensão AC e varia entre -1 e +1 V. Como o ESP32 não consegue medir tensão negativa faz-se necessário transformar estes valores de forma que toda a forma esteja no ciclo positivo.

Para isso, utilizou-se um divisor de tensão, vide Figura 45 (a), com alimentação ( $V_{in}$ ) 5 Vcc fornecida por uma fonte externa. Os resistores R1 e R2 possuem resistências iguais para que a tensão sobre eles seja a mesma. Para o projeto foram considerados resistores de 10k $\Omega$ . Em paralelo ao R2, foi adicionado um capacitor de 100 $\mu$ F com o objetivo de estabilizar a tensão sobre este resistor, conforme Figura 41 (b).



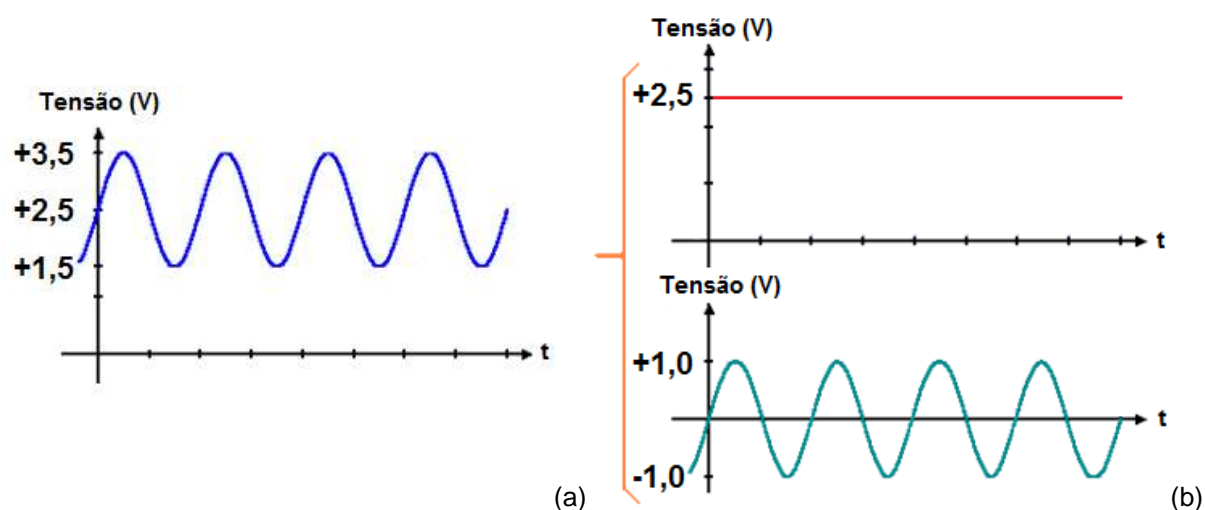
**Figura 41** – Circuito auxiliar para a utilização do sensor de corrente. (a) Divisor de tensão. (b) Divisor de tensão com o capacitor integrado.  
(Fonte: VIDA DE SILICIO, 2017)

O circuito final está ilustrado na Figura 42:



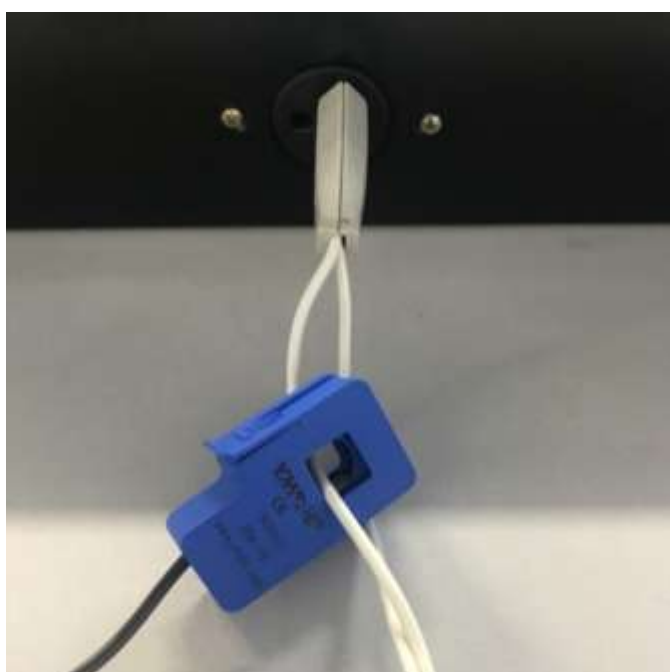
**Figura 42** – Ligação do sensor de corrente com o circuito final.  
(Fonte: Adaptado de VIDA DE SILICIO, 2017)

O divisor de tensão irá gerar uma tensão contínua de 2,5 V que será sobreposta ao sinal de saída do sensor. O resultado será uma senóide sem valores negativos, conforme ilustrado nas Figuras 43 (a) e (b).



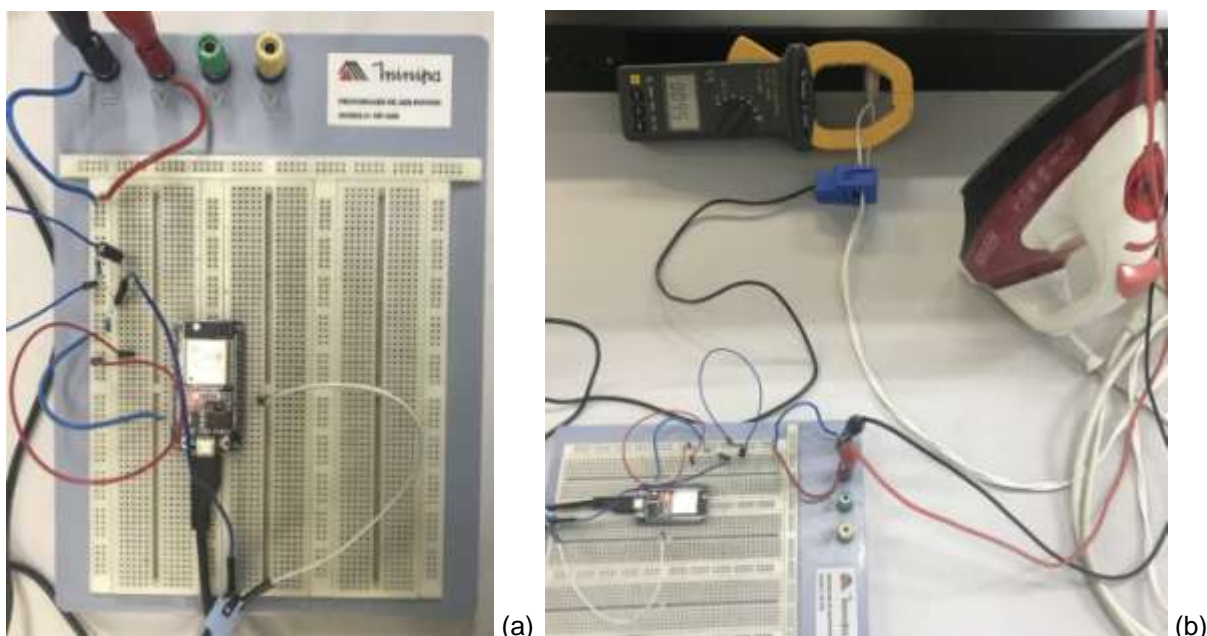
**Figura 43** – Sobreposição de formas de onda. (a) Onda completa. (b) Componentes CC e AC.  
(Fonte: Adaptado de FEIRA DE CIÊNCIAS, 1999)

É importante ressaltar a forma de conexão do sensor. A corrente elétrica produz um campo magnético com mesmo sentido. Nos dois fios de alimentação de um dispositivo, suas correntes possuem sentidos opostos e, conseqüentemente, seus campos magnéticos também. Isso faz com que, ao medir os dois fios juntos, os campos sejam anulados, não sendo possível mensurar a corrente. Por este motivo, este sensor não pode ser conectado aos dois fios de alimentação. A conexão deve ser realizada em apenas 1 fio, conforme Figura 44.



**Figura 44** – Forma de conexão correta entre o sensor de corrente e o condutor.  
(Fonte: Autores, 2018)

A montagem do circuito pode ser visualizada nas Figuras 45 (a) e (b). Como o cabo do sensor de corrente possui em sua saída um conector P2, foi utilizado um adaptador para realizar a conexão na *protoboard* sem que fosse necessário cortar o cabo. Este adaptador foi conectado a dois fios, sendo o primeiro (fio azul) conectado entre os resistores R1 e R2 e o segundo (fio branco) ligado na GPIO 4 do ESP32. Este, por sua vez, foi alimentado via cabo USB conectado ao computador.



**Figura 45** – Teste do sensor de corrente. (a) Circuito montado na *protoboard* visto de cima. (b) Circuito completo – medição da carga.  
(Fonte: Autores, 2018)

Para identificar qual seria a máxima corrente do ferro, utilizou-se a Equação 4.11 isolando a corrente. Como a tensão da bancada foi medida e conferida anteriormente (teste do sensor de tensão), adotou-se o valor de 125 V. O fator de potência é 1 pois trata-se de carga resistiva e a potência máxima é 1200 W, conforme informado anteriormente. Aplicando estes valores na fórmula, obteve-se corrente igual a 9,6 A.

$$I = \frac{1200}{125 \times 1} = 9,6 \text{ A}$$

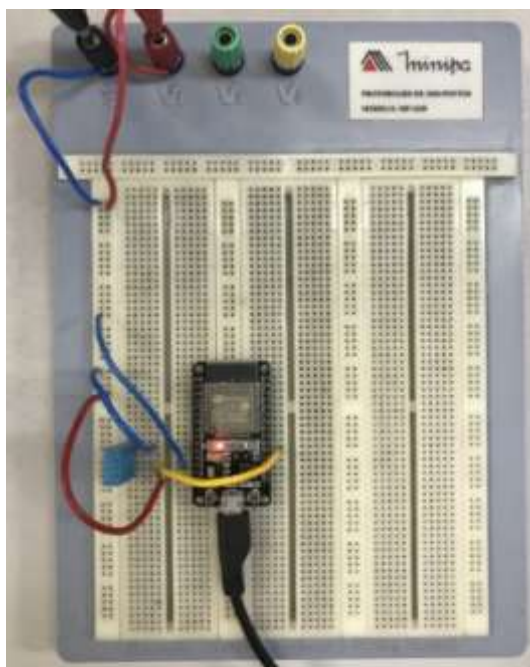
Através do código registrado no ESP32 (que será explicado posteriormente), realizou-se a medição da corrente RMS da tomada e foi possível verificar que o resultado obtido estava de acordo com o esperado, bem como o valor mostrado no alicate amperímetro – Figura 46.



**Figura 46** – Comparação dos valores medidos pelo sensor de corrente e alicate amperímetro. (Fonte: Autores, 2018)

### 7.1.1.3 Sensor de temperatura e umidade

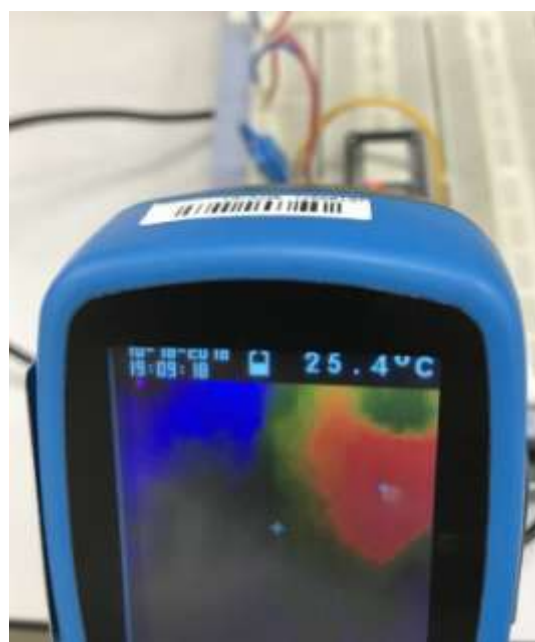
O teste realizado no sensor de temperatura e umidade DHT11 é feito de uma forma bem simples e fácil. É necessário apenas realizar a ligação dos três pinos existentes no dispositivo, conforme ilustrado na Figura 47, sendo eles: alimentação positiva (representada pelo fio vermelho), alimentação negativa (representado pelo fio azul) e sinal de dados enviados para o ESP32 (representado pelo fio amarelo) cuja informação é processada através da porta GPIO 4 do microcontrolador. Vale ressaltar que existe um quarto pino que não deve ser conectado e, portanto, não é utilizado no circuito.



**Figura 47** – Teste do sensor de temperatura e umidade - Circuito montado na *protoboard*.  
(Fonte: Autores, 2018)

A visualização dos resultados obtidos pode ser demonstrada na Figura 52 (a). Apesar de este sensor realizar a medição tanto de temperatura quanto de umidade, a conferência dos dados será focada apenas nos valores de temperatura. Para isso, utilizou-se um termostato, através do qual é possível demonstrar a veracidade do valor lido, conforme Figura 48 (b).

Timestamp	Temperatura	Umidade
21:23:15.760	-> Temperatura 24.00	Umidade ==> 54.00
21:23:16.773	-> Temperatura 24.00	Umidade ==> 54.00
21:23:18.063	-> Temperatura 24.00	Umidade ==> 54.00
21:23:19.048	-> Temperatura 24.00	Umidade ==> 54.00
21:23:20.308	-> Temperatura 24.00	Umidade ==> 54.00
21:23:21.320	-> Temperatura 24.00	Umidade ==> 54.00
21:23:22.607	-> Temperatura 24.00	Umidade ==> 54.00
21:23:23.584	-> Temperatura 24.00	Umidade ==> 54.00
21:23:24.866	-> Temperatura 24.00	Umidade ==> 54.00
21:23:25.885	-> Temperatura 24.00	Umidade ==> 54.00
21:23:27.135	-> Temperatura 24.00	Umidade ==> 54.00
21:23:28.152	-> Temperatura 24.00	Umidade ==> 54.00
21:23:29.409	-> Temperatura 24.00	Umidade ==> 54.00
21:23:30.430	-> Temperatura 24.00	Umidade ==> 54.00
21:23:31.681	-> Temperatura 24.00	Umidade ==> 54.00
21:23:32.694	-> Temperatura 24.00	Umidade ==> 54.00
21:23:33.982	-> Temperatura 24.00	Umidade ==> 55.00
21:23:34.967	-> Temperatura 24.00	Umidade ==> 55.00
21:23:36.226	-> Temperatura 24.00	Umidade ==> 55.00
21:23:37.240	-> Temperatura 24.00	Umidade ==> 55.00
21:23:38.529	-> Temperatura 24.00	Umidade ==> 55.00
21:23:39.513	-> Temperatura 24.00	Umidade ==> 55.00

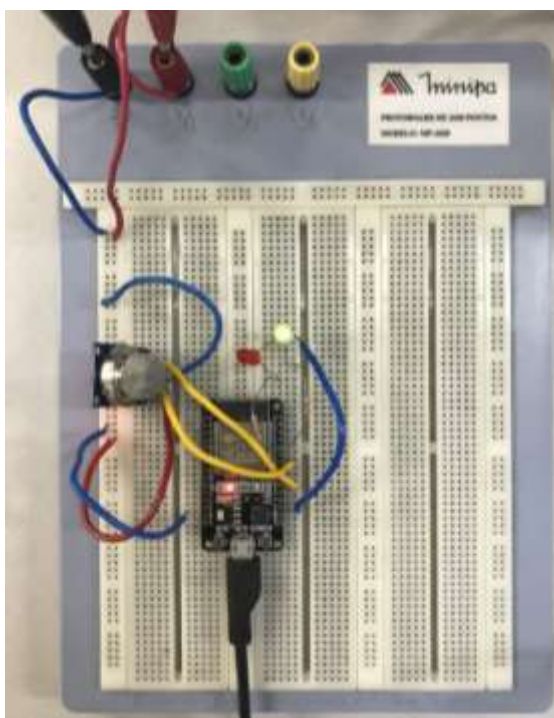


**Figura 48** – Comparação dos valores de temperatura.  
(a) Leitura do sensor DHT11. (b) Leitura do termostato.  
(Fonte: Autores, 2018)

É importante ressaltar que em aplicações reais é necessário utilizar mais de um sensor a fim de garantir uma maior confiabilidade e precisão ao sistema. Quando se pretende medir a temperatura de um ambiente, o valor correto será o resultado da média de todas as leituras dos sensores implementados, uma vez que pontos específicos podem apresentar drásticas variações. Como exemplo, pode-se citar um cômodo contendo um ar condicionado: a temperatura será mais fria no ponto mais próximo ao equipamento em relação à outra localidade mais distante do mesmo, cuja temperatura será superior.

#### 7.1.1.4 Sensor de gás

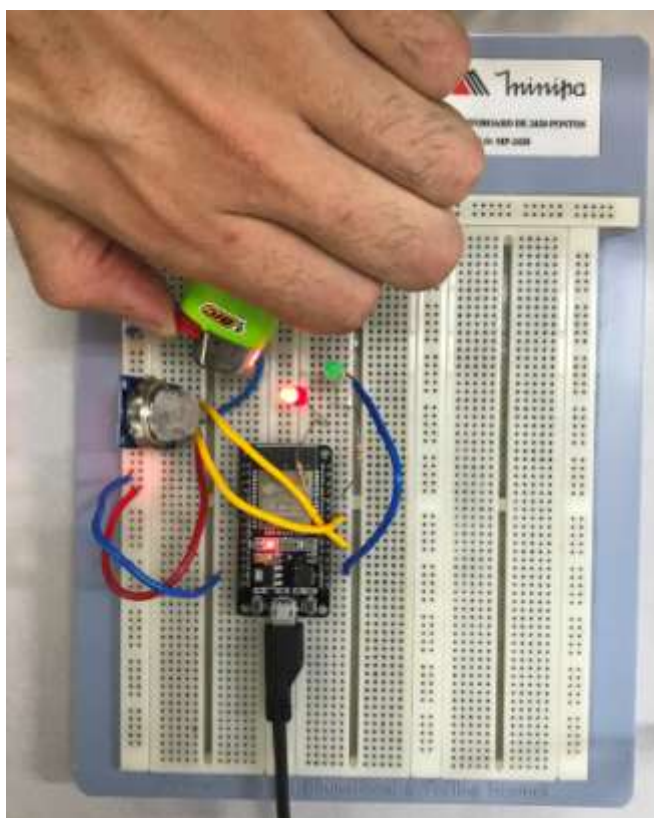
O circuito teste do sensor de gás está representado na Figura 49. Este possui quatro pinos, sendo eles: alimentação positiva, representada pelo fio vermelho; alimentação negativa, representada pelo fio azul; saída analógica (ligado na GPIO 4) e saída digital (ligado na GPIO 18), ambas representadas pelo fio amarelo. O ESP32 está conectado no computador via USB para fazer a leitura dos valores do sensor no monitor serial. Além disso, foram inseridos dois LED's ligados em série com resistores de 270  $\Omega$ , sendo o primeiro de cor vermelha (ligado na GPIO 19) e o segundo de cor verde (ligado na GPIO 21).



**Figura 49** – Teste do sensor de gás - Circuito montado na *protoboard*.  
(Fonte: Autores, 2018)

Para a lógica da programação gravada no ESP32, adotou-se um valor como parâmetro para o acionamento dos LED's, sendo este 3.000 ppm. Este parâmetro, porém, é inserido na programação em forma percentual e equivale a 30% do valor máximo lido pelo sensor (10.000 ppm). Quando o pino analógico do sensor detectar um valor inferior ao estipulado, o LED verde ficará aceso, acusando a inexistência de gás ou fumaça. Caso contrário, o LED verde se apaga enquanto o LED vermelho é ligado indicando a presença de gás inflamável ou fumaça no ambiente.

Para provar o funcionamento do sensor, foi utilizado um isqueiro que possui gás butano em seu interior, conforme ilustrado na Figura 50.



**Figura 50** – Teste no sensor de gás - LED vermelho aceso: Concentração acima de 3.000 ppm de gás/fumaça detectado.  
(Fonte: Autores, 2018)

Os dados obtidos estão representados na Figura 51. Existem duas leituras para este sensor, uma referente à saída digital e outra referente à saída analógica. A leitura da saída analógica é dada em bits e, como a conversão de valores CA/CC tem precisão de 12 bits, o valor lido pode variar de 0 a 4.096 bits ( $2^{12}$ ). A relação entre o número de bits e a leitura em percentual e ppm é linear e, por

isso, realizando uma regra de três simples, pode-se verificar que o valor configurado de 30% = 3.000 ppm  $\cong$  1.229 bits. A leitura da saída digital é mais simples e sua resposta é 0 ou 1, tendo sido utilizada apenas para verificação de sua funcionalidade. Sua regulação é realizada através do potenciômetro. Através dos valores comparativos mostrados abaixo e, novamente realizando regra de três simples, pode-se verificar que o potenciômetro está ajustado para detectar valores acima de aproximadamente 4.858 ppm (considerando valor 0 – gás detectado – para leituras acima de 1.990 bits).

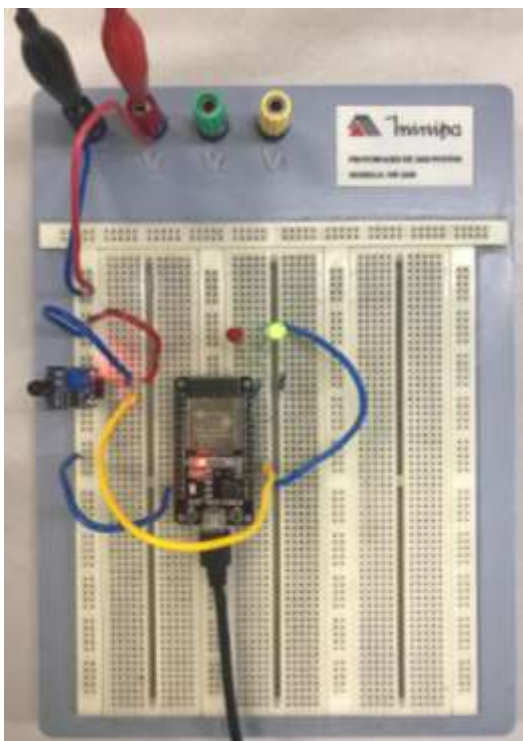


**Figura 51** – Leitura das saídas do sensor de gás.  
(Fonte: Autores, 2018)

### 7.1.1.5 Sensor de fogo

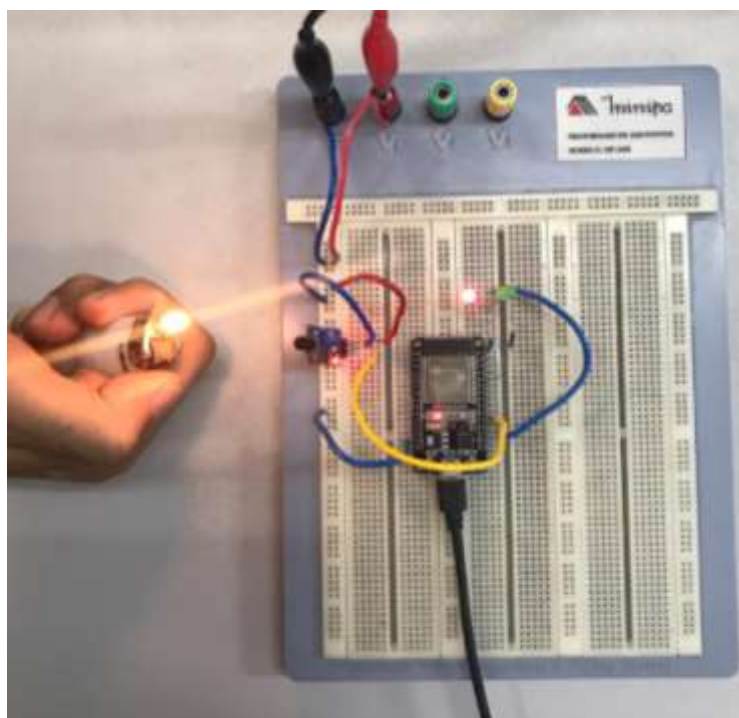
O circuito teste do sensor de fogo é bem simples e está demonstrado na Figura 52. Os fios vermelho e azul estão sendo utilizados para realizar a alimentação

do sensor, sendo Vcc e GND, respectivamente. O fio amarelo está fazendo a conexão entre o pino de saída e a GPIO 4 do ESP32. Este, por sua vez, está sendo alimentado através do cabo USB do computador para fazer a leitura dos dados obtidos no serial monitor. Além disso, foram inseridos dois LED's ligados em série com resistores de 270  $\Omega$ , sendo o primeiro de cor vermelha (ligado na GPIO 19) e o segundo de cor verde (ligado na GPIO 21).



**Figura 52** – Teste do sensor de fogo - Circuito montado na *protoboard*.  
(Fonte: Autores, 2018)

A saída deste sensor é apenas digital, portanto só haverá dois níveis de sinal: alto (1) ou baixo (0). A lógica foi configurada da seguinte forma: enquanto não houver detecção de fogo, o sensor indicará nível alto e o LED verde ficará aceso; quando o fogo for detectado, o LED verde se apaga e o LED vermelho é ligado. Para realizar o experimento, utilizou-se um isqueiro simulando a presença de fogo no ambiente – Figura 53.



**Figura 53** – Teste no sensor de gás - LED vermelho acesso: Fogo detectado.  
(Fonte: Autores, 2018)

A leitura da saída digital pode ser visualizada através da Figura 54, comprovando o funcionamento correto do sensor e da lógica de programação.

```
COM5
19:10:37.276 -> Porta digital: 1
19:10:37.377 -> Porta digital: 1
19:10:37.447 -> Porta digital: 1
19:10:37.547 -> Porta digital: 1
19:10:37.648 -> Porta digital: 1
19:10:37.751 -> Porta digital: 1
19:10:37.852 -> Porta digital: 1
19:10:37.952 -> Porta digital: 1
19:10:38.055 -> Porta digital: 0
19:10:38.089 -> Fogo detectado !!!
19:10:38.155 -> Porta digital: 0
19:10:38.188 -> Fogo detectado !!!
19:10:38.256 -> Porta digital: 0
19:10:38.289 -> Fogo detectado !!!
19:10:38.355 -> Porta digital: 0
19:10:38.389 -> Fogo detectado !!!
19:10:38.459 -> Porta digital: 0
19:10:38.493 -> Fogo detectado !!!
19:10:38.561 -> Porta digital: 0
19:10:38.595 -> Fogo detectado !!!
```

Sensor detectou a presença de fogo no ambiente

**Figura 54** – Leitura da saída do sensor de fogo.  
(Fonte: Autores, 2018)

### 7.1.2 Disposição dos componentes

Após a realização de todos os testes nos sensores, foi necessário projetar e desenvolver a parte física do protótipo. Os autores optaram por utilizar duas placas de acrílico, denominadas Placa 1 e Placa 2, as quais serão detalhadas a seguir.

**Placa 1:** Composta por uma lâmpada, um interruptor e uma tomada. A lâmpada poderá ser ligada tanto pelo interruptor, quanto via *dashboard*, que possui também a função de indicar seu estado (acesa ou apagada). A tomada ficará disponível para acionamento de uma carga qualquer, e somente nela o consumo de energia será medido, ou seja, os sensores de tensão e corrente estarão apenas monitorando esta carga. Os três dispositivos são conectados pelo módulo relé para serem acionados e por isso ele será inserido atrás da placa, com suas devidas ligações.

A Placa 1 está representada na Figura 55.



**Figura 55** – Placa 1 - Lâmpada, interruptor e tomada.  
(Fonte: Autores, 2018)

**Placa 2:** Composta por três sensores: o de temperatura e umidade, cuja função é apenas de monitoração; o de fogo, responsável pela detecção de incêndio; e o de gás, que tem a característica de detectar alguma fumaça ou gás inflamável presente no ambiente.

A Placa 2 está representada na Figura 56.



**Figura 56** – Placa 2 - Sensores de temperatura e umidade, fogo e gás.  
(Fonte: Autores, 2018)

### 7.1.3 Conexão dos dispositivos ao microcontrolador

Após a inserção de todos os componentes presentes nas Placas 1 e 2, foi necessário pensar em como seriam feitas as ligações com o ESP32. Os autores optaram por utilizar duas placas de fenolite perfurada: na primeira foi inserido o ESP32, o regulador de tensão e um LED de sinalização indicando a conexão com o servidor; na segunda placa foi inserido o sensor de tensão, o conector do sensor de corrente e o circuito divisor de tensão criado para o mesmo, conforme já explicado anteriormente.

Em seguida, estas duas placas foram alocadas em uma caixa a fim de melhor organiza-los, conforme ilustrado na Figura 57.



**Figura 57** – Caixa com o circuito devidamente montado nas placas de fenolite.  
(Fonte: Autores, 2018)

## 7.2 Programação na IDE do Arduino

Na IDE do arduino foram inseridos as programações necessárias para: fazer a comunicação com o servidor da IBM e seus clientes, utilizando para isso o protocolo MQTT; estabelecer a conexão e a configuração do WiFi; realizar a conexão do ESP32 com os diversos sensores implantados neste projeto, enviando e recebendo dados e transformando o código dos dispositivos utilizados para a linguagem *json*, para serem interpretados pela plataforma Node-RED.

Algumas bibliotecas foram necessárias para essa etapa, nas quais se destacam principalmente as descritas abaixo:

**Wifi.h** – Permite que o microcontrolador se conecte à internet;

**WiFiManager.h** – Biblioteca inscrita por cima do WiFi.h que é responsável por fazer o gerenciamento de conexões wireless;

**DNSServer.h** – Implementa um servidor DNS simples no qual os clientes podem abrir um servidor da Web em execução no ESP32 usando um nome de domínio, não um endereço IP;

**WebServer.h** – Baseada em servidores Web, permite a manipulação de métodos http;

**PubSubClient.h** – Biblioteca cliente capaz de criar mensagens simples de publicação / assinatura para um servidor que suporta o protocolo MQTT;

**ArduinoJson.h** – Utilizada para criar *strings* armazenadas na memória de programação e em seguida utilizadas no Node-RED.

Para que haja a comunicação via protocolo MQTT do broker da IBM com o ESP32, é necessário definir algumas *strings*. Estas são criadas a partir das informações do site do servidor e de chaves geradas pelo cadastro do dispositivo no mesmo. Além disso, é necessário definir tópicos para que haja a publicação ou assinatura dos dispositivos envolvidos no projeto, ou seja, o microcontrolador e seus sensores, o *broker* e a plataforma Node-RED.

A configuração do WiFi é necessária uma vez que todas as informações serão enviadas através da internet. Devido a isso, foi necessário utilizar algumas bibliotecas e são elas: Wifi.h, WiFiManager.h, DNSServer.h e WebServer.h. Existem dois modos de operação para a conexão do microcontrolador na rede, que pode ser tanto em ponto de acesso quanto em modo estação. Ambos utilizam a função callback para responder a esses dois eventos. O nome da rede configurada para conectar o ESP32 é “BI-ESP-AP”.

Para inserir os sensores no projeto foi necessário definir os pinos de entradas analógicos e digitais dos mesmos. Além disso, para que esses sensores sejam interpretados pela plataforma Node-Red e com isso realizar as funções desenvolvidas no projeto foi necessário que eles estivessem na linguagem Json dentro da linguagem C++ na IDE do Arduino, utilizando com isso o auxílio da biblioteca ArduinoJson.h.

Os sensores utilizados no projeto foram: de temperatura e umidade, fogo, gás, tensão e corrente, sendo que ambos possuem códigos exclusivos para realizar determinadas funções, porém configurados de forma a ficarem juntos em uma mesma programação. Além disso, foram

integrados também um módulo relé de 2 canais, sendo um responsável por acionar uma lâmpada e o outro pelo acionamento de uma carga qualquer ligada à tomada.

Para realizar o monitoramento da temperatura e umidade foi necessário inserir a biblioteca DHT.h na programação. Em seguida foram declarados o objeto que realiza a leitura de ambas as variáveis, “dht (14, DHT11)”, ligado na GPIO 14 do ESP32 e as variáveis para guardar os seus valores “*temperature*” e “*humidity*”. As leituras realizadas pelo sensor foram feitas de acordo com uma condição *if/else* estabelecida no código. Seus valores deverão ser convertidos em linguagem json para serem publicados e posteriormente interpretados e programados no Node-RED.

Para o código do sensor de fogo é necessário definir o pino digital de entrada (pino\_D0), que será usado na GPIO 21 do microcontrolador, e declarar a variável responsável por guardar os valores lidos pelo sensor – “valor\_dig”. Estes são gerados no monitor serial e interpretados da seguinte forma: o fogo é detectado quando o sensor indica sinal baixo (0); para sinal alto (1), não há incidência de fogo no local. Os valores serão convertidos para a linguagem json e enviados para o Node-RED.

Em relação ao código do sensor de gás, apenas o pino analógico foi utilizado (pin\_a0), sendo este conectado na GPIO 32. A variável responsável por guardar os valores lidos pelo sensor foi declarada como “valor\_analog”. Seu resultado será apresentado em percentual de concentração de gás, sendo 0% equivalente a 0 ppm e 100% equivalente a 10.000 ppm. Estes valores serão convertidos em linguagem json e enviados para o Node-RED.

Os relés foram incluídos no projeto com o intuito de acionar cargas, sendo elas: uma lâmpada e um aparelho qualquer conectado à tomada disponível na Placa 1. Para que isso fosse possível, é necessário realizar a programação para gerenciar a atuação destes relés. Estes foram definidos “pinRele1”, ligado na GPIO 25 e “pinRele2” conectado na GPIO 26.

O sensor de corrente e tensão foram conectados aos cabos da

tomada da Placa 1. A saída do sinal da corrente foi conectada à GPIO 15, enquanto a de tensão na GPIO 34. A variável responsável por guardar os valores lidos pelo sensor foi declarada são “*current*” e “*voltage*”, respectivamente. Estes valores serão convertidos em linguagem json e enviados para o Node-RED. Além disso, ainda no ESP é realizada o cálculo da potência, através do produto das leituras destes dois sensores.

O código completo utilizado encontra-se disponível no Apêndice A.

## 7.3 Integração do Servidor

### 7.3.1 Serviço de Internet das Coisas no IBM Cloud

Para utilizar os serviços do IBM Cloud, proposto neste projeto, é necessário primeiramente realizar a inscrição no IBMid através do site: <https://www.ibm.com/cloud-computing/bluemix/pt>, vide Figura 58. Com esta inscrição é possível adquirir acesso aos aplicativos da IBM, serviços, Internet das coisas entre outros.



**Figura 58** – Tela inicial da IBM Cloud.  
(Fonte: Autores, 2018)

Ao acessar o site, é necessário se inscrever, ou seja, preencher alguns campos colocando o nome, sobrenome e uma senha, vide Figura 59.

**Criar sua conta IBM**

Acesso a Testes, Registros, Kits, Serviços e API.

Já tem uma conta IBM? Conectar

Email \*

Primeiro nome \*

Sobrenome \*

País ou região \* (?)

Configurar uma senha \*

- Mínimo de 8 caracteres
- Um caractere minúsculo
- Um caractere maiúsculo
- Um número

A IBM pode usar meus dados de contato para manter-me informado sobre produtos, serviços e ofertas:

por e-mail.

É possível retirar seu consentimento de marketing para qualquer momento enviando um e-mail para [netsupp@us.ibm.com](mailto:netsupp@us.ibm.com). Além disso, é possível obter e receber e-mails de marketing ao clicar no link de assinatura de cada um desses e-mails.

Mais informações sobre o processo podem ser encontradas na Declaração de Privacidade IBM. Ao enviar esse formulário, reconfigurar e ler a Declaração de Privacidade IBM.

Eu aceito os Termos e Condições do produto contidos nesse formulário de registro.

**Figura 59** – Criando uma conta no IBMid.  
(Fonte: Autores, 2018)

Após concluir o cadastro do IBMid, é possível criar uma conta no IBM Cloud, considerado o servidor em nuvem utilizado neste projeto. O registro ocorre através do site: <https://console.bluemix.net/registration/> ilustrado na Figura 60. Deve-se inserir o e-mail do cadastro anterior, preencher novos campos e finalizar a inscrição.

**Figura 60** – Criando uma conta no IBM Cloud.  
(Fonte: Autores, 2018)

Após a finalização das inscrições no IBMid e IBM Cloud, é necessário acessar o menu de catálogos para adquirir os serviços da Internet das Coisas.

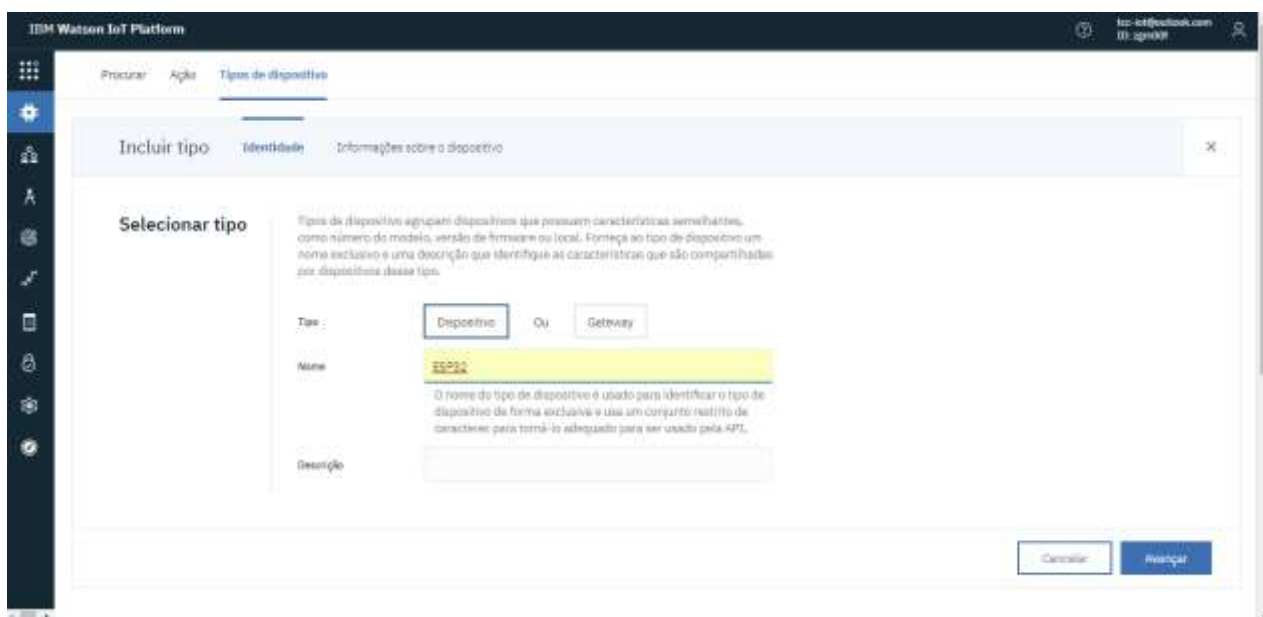
### 7.3.2 Cadastrando o ESP32 na plataforma de IoT do IBM Cloud

A próxima etapa realizada é cadastrar na Plataforma IoT do IBM Cloud os dispositivos que irão ser utilizados no projeto. Neste trabalho, o dispositivo de cadastro será o ESP32. Para fazer isto, é necessário selecionar a aba “Tipo de Dispositivo” e clicar na opção “+ Incluir tipo de dispositivo”, vide Figura 61.



**Figura 61** – Cadastrando os dispositivos na plataforma IoT.  
(Fonte: Autores, 2018)

A Figura 62 mostra os campos que foram preenchidos, como por exemplo o nome e as informações da placa do dispositivo.



The screenshot shows the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Procurar', 'Ajuda', and 'Tipos de dispositivos'. Below that, a dialog box titled 'Incluir tipo' is open, with tabs for 'Identidade' and 'Informações sobre o dispositivo'. The 'Identidade' tab is selected. The dialog contains a 'Selecionar tipo' section with a text box explaining that device types group devices with similar characteristics. Below this, there are three fields: 'Tipo' with a dropdown menu showing 'Dispositivo' selected, 'Nome' with a text input containing 'ESP32', and 'Descrição' with an empty text input. At the bottom right of the dialog, there are 'Cancelar' and 'Finalizar' buttons.

**Figura 62** – Preenchimento dos dados para identificação de dispositivo.  
(Fonte: Autores, 2018)

Após inclusão do ESP32 na Plataforma IoT do IBM Cloud, é necessário fazer alguns registros de identificação deste dispositivo que será utilizado posteriormente na Plataforma do Node-Red bem como na IDE do arduino. Nas abas seguintes do registro do ESP32, é necessário criar (ou gerar automaticamente) algumas chaves de identificação do dispositivo que serão utilizadas nos passos seguintes da programação do projeto. É importante guardar a chave de *token* de autenticação, que no presente trabalho foi gerada automaticamente, conforme Figura 67.




serão posteriormente utilizadas na programação do Node-RED bem como na IDE do Arduino.

## 7.4 Programação no Node-Red

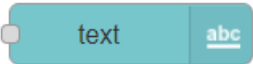


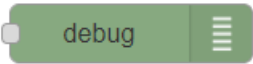
O Node-RED foi utilizado para realizar a conexão entre o ESP32 e a *dashboard* criada. Além disso, o mesmo promove a interação com os serviços *onlines* através da chave API.

Abaixo são apresentados todos os nós utilizados na criação do *flow* e suas respectivas funções.

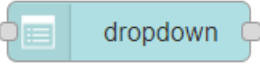
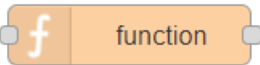
### ➤ Nós de entrada

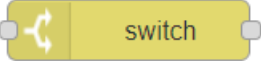
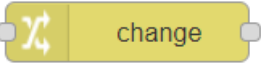
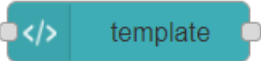
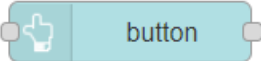
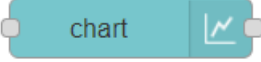
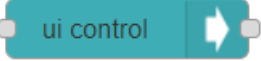
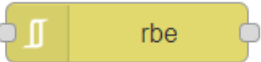
- 
Pode ser usado com o *Watson IoT Platform* para receber eventos ou comandos enviados de dispositivos, ou atualizações de status referentes aos componentes ou aplicativos;

### ➤ Nós de saída

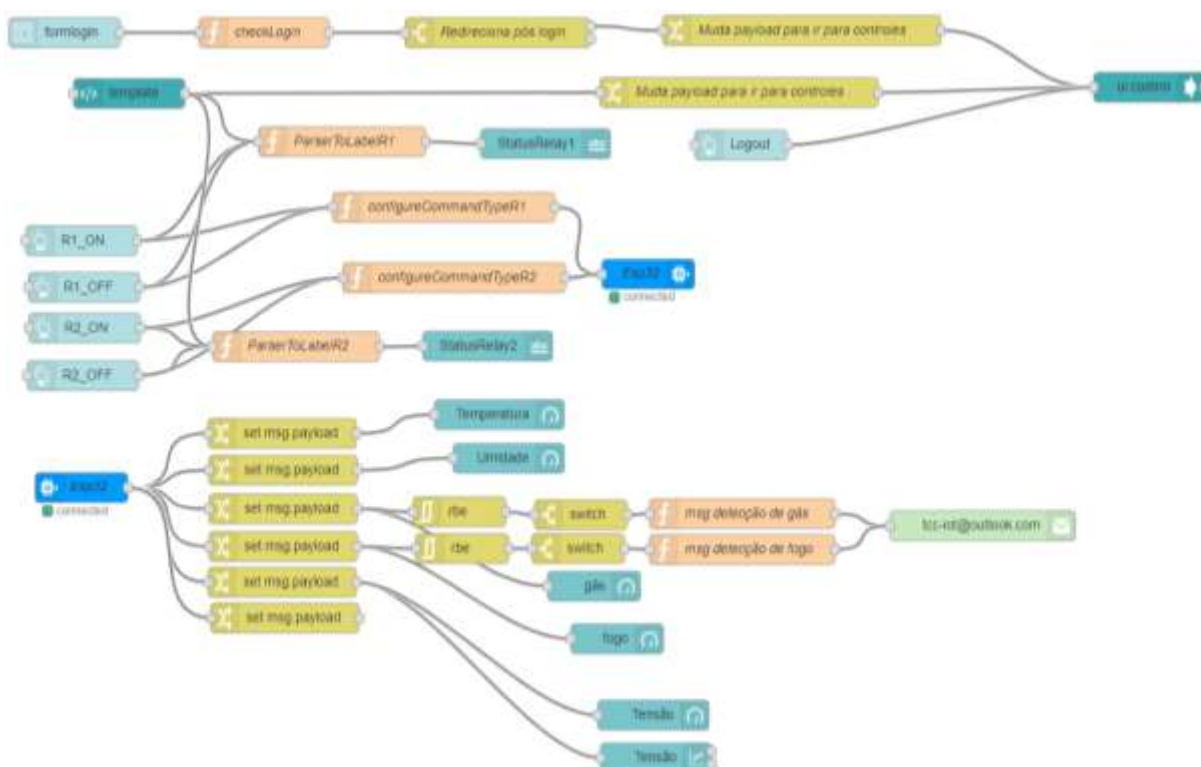
- 
Exibe um campo de texto não editável à interface do usuário;
- 
Adiciona um gráfico tipo velocímetro à interface do usuário;
- 
Pode ser usado com o *Watson IoT Platform* para enviar comandos a um dispositivo ou enviar um evento em nome de um dispositivo;
- 
Exibe propriedades de mensagem selecionadas na guia da barra lateral de depuração e, opcionalmente, no *log* de tempo de execução;

### ➤ Nós de processamento

- 
Adiciona uma caixa de seleção suspensa à interface do usuário;
- 
Possui funções JavaScript para executar as mensagens recebidas pelo nó;

	switch	Roteia mensagens com base em seus valores de propriedade ou posição de sequência;
	change	Define, altera, exclui ou move propriedades de uma mensagem, contexto de fluxo ou contexto global;
	template	Nó genérico que usa diretivas de HTML e de material angular. Pode ser usado para criar um elemento de interface de usuário dinâmico;
	button	Adiciona um botão à interface do usuário;
	chart	Plota os valores de entrada em um gráfico. Este pode ser um gráfico de linhas baseado em tempo, de barras, colunas ou de pizza;
	ui control	Permite controle dinâmico do painel;
	rbe	Reporta por nó de exceção – apenas envia dados se a carga útil foi alterada.

A Figura 65 a seguir mostra o *flow* em seu estado atual da programação desenvolvida.



**Figura 65** – Flow criado no Node-RED.  
(Fonte: Autores, 2018)

O bloco *dropdown* (“*login*”) tem a função de representar a tela de bloqueio para que o usuário acrescente o *login* e senha predeterminados e assim tenha acesso ao seu projeto. Este bloqueio foi feito através do bloco *function* (“*checklogin*”). Caso o login e a senha inseridos estejam corretos, o *switch* (“*redireciona*”) habilita o acesso à página e o *change* (“*muda payload para ir para os controles*”) abre a interface com o usuário.

A interface em questão possui botões para ligar (*ON*) e desligar (*OFF*) a lâmpada. Estes botões foram inseridos através do bloco *button* (“*R1\_ON / R1\_OFF*”). Para realizar a ação a partir do acionamento desses botões virtuais, foi necessário programar o bloco *function* (“*ParsertolabelR1*”). Ao clicar no botão *ON* para ligar a carga, o estado da lâmpada automaticamente é alterado na *dashboard*. Quem realiza essa função é o bloco *text* (“*statusrelay1*”). Ao mesmo tempo, outro bloco *function* (“*configurecommandtypeR1*”) entra em ação, enviando o código *commandR1* para acender a lâmpada. Esse código será interpretado pela linguagem C++ contida no microcontrolador ESP32, que enviará o comando de acionar o relé com base na programação existente.

Para o monitoramento da temperatura, umidade, detecção de gás e chama, foi necessário criar *strings* de cada sensor, a fim de serem identificados no Node-RED pela programação em Json. O nó *ibmiotin* (“*esp32*”) foi utilizado para o recebimento dos valores que serão monitorados/registrados no *dashboard*. Observando os nós na figura anterior, é possível verificar que um único bloco está recebendo todas as informações referentes aos sensores. Por este motivo, é necessário filtrá-las para que sejam separadas em cada grandeza e assim poder associar a respectiva ação a ser tomada. Para esse filtro, será usado o bloco *change* (“*set.message.payload*”).

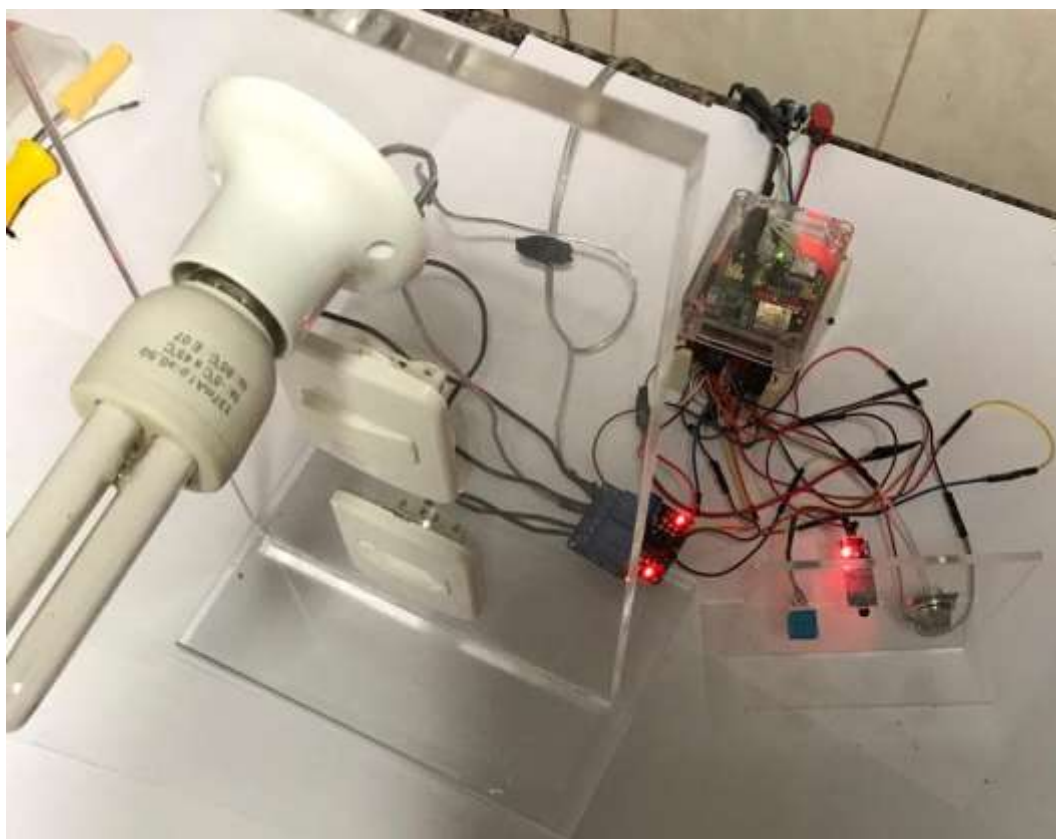
Os valores de temperatura e umidade estão sendo utilizados apenas para o monitoramento em tempo real, não tendo, portanto, vínculo com nenhuma tomada de decisão. Estes valores são apresentados em forma de gráfico tipo velocímetro gerado através do nó *gauge*.

Para a detecção de gás, foi utilizado o bloco *switch* que realizará a lógica de alarmes, programada com as seguintes condições: se o sensor detectar

concentração de gás/fumaça igual ou inferior a 5% este é apenas registrado na *dashboard*; caso seja maior que 5%, além do registro na *dashboard*, um e-mail será enviado para o destinatário desejado. Para que este e-mail não seja enviado repetidas vezes enquanto o sensor detecta o que seria uma condição indesejada para o sistema, foi utilizado o bloco *rbe* com a função “*block unless value changes*”, que bloqueia estas repetições, não enviando mais nenhum e-mail enquanto o valor do *feedback* não retornar para a faixa admissível. O sensor de fogo foi programado de maneira similar ao de gás, com a diferença de que o pino é digital então só envia sinal alto ou baixo.

### 7.5 Protótipo Finalizado

Após a realização de toda a programação, tanto na IDE do Arduino quanto no Node-RED, e de serem feitas todas as ligações dos componentes utilizados no projeto, é possível visualizar a configuração final do protótipo através das Figuras 66.



**Figura 66** – Protótipo finalizado.  
(Fonte: Autores, 2018)

Os resultados obtidos podem ser visualizados através da *dashboard* criada. Estes estão sendo demonstrados através de gráficos, indicando o monitoramento da temperatura e umidade, chama, gás e botões dos relés para o acionamento das cargas, vide Figura 67.



**Figura 67** – *Dashboard* criada.  
(Fonte: Autores, 2018)

## 7.6 Análise dos Resultados

Após a finalização do protótipo, ocorreram alguns imprevistos que não eram esperados pelo grupo. Isso porque, durante a realização dos testes individuais de cada sensor utilizado neste projeto, não foi apresentado nenhum problema, tanto na instalação quanto na medição, pois esta também não demonstrava erros expressivos.

Porém, ao realizar a conexão de todos os componentes na placa e a integração com a plataforma Node-RED, verificou-se que o valor da medição de corrente não gerava *strings* na programação da IDE do Arduino e com isso a leitura não era realizada pelo Node-RED. Conseqüentemente, este valor não aparecia na *dashboard*. A solução adotada foi visualizar as medições de tensão e corrente apenas no monitor serial, conforme ilustrado na Figura 68, e com isso realizar o

cálculo do consumo da carga, que era o objetivo principal do grupo. Contudo, a perspectiva é que essas medições sejam inseridas na interface gráfica futuramente.

```
COM5
|
|
|
Tensão: 124.38
Corrente: 8.55
potencia ativa: 1029.01
Tensão: 124.41
Corrente: 8.54
potencia ativa: 1029.07
Tensão: 124.29
Corrente: 8.56
potencia ativa: 1030.09
Tensão: 124.46
Corrente: 8.55
potencia ativa: 1057.86
Tensão: 126.15
Corrente: 8.64
potencia ativa: 1028.14
Tensão: 124.33
Corrente: 8.54
potencia ativa: 1028.46
Tensão: 124.29
Corrente: 8.55
potencia ativa: 1032.62
Tensão: 124.61
Corrente: 8.56
potencia ativa: 1031.32
Tensão: 124.55
Corrente: 8.55
potencia ativa: 1032.66
```

**Figura 68** – Cálculo da tensão, corrente e potência de uma carga (ferro elétrico).  
(Fonte: Autores, 2018)

Em relação aos outros sensores e atuadores utilizados no projeto foram obtidos resultados positivos, a programação obteve sucesso e os valores puderam ser visualizados na interface gráfica, possibilitando assim uma interação com o usuário.

## **8 APLICABILIDADE DO PROTÓTIPO**

Conforme já mencionado anteriormente, a proposta do presente trabalho consistia em elaborar um protótipo simples, porém aplicável a inúmeros ramos, podendo ser utilizado em grande escala para os mais diversos setores, desde projetos para amadores até em processos complexos.

Afim de elucidar melhor esta ideia, este capítulo se dedicará a descrever algumas possíveis aplicações com os sensores utilizados.

### **8.1 Monitoramento da Temperatura e Umidade**

Fazer o controle da temperatura e umidade em um ambiente é imprescindível para diversos setores e por isso as aplicações são inúmeras. Podem-se citar as indústrias alimentícias e de bebidas, restaurantes, supermercados, padarias, enfim, todos os setores que requerem um correto monitoramento dessas variáveis em sua cadeia produtiva para impedir que os alimentos sejam estragados ou ingeridos de forma incorreta pelos seus consumidores.

Outro exemplo seria o ramo da saúde, como hospitais, laboratórios ou postos, que requerem um controle preciso das geladeiras que contenham vacinas, medicamentos ou amostras de exames clínicos, que exigem um correto manuseio e condicionamento da temperatura do ambiente. Em locais de internação, é de extrema importância que a temperatura seja mantida em níveis baixos para minimizar o risco de contaminação dos pacientes hospitalizados.

Uma terceira aplicação muito comum é em climatizadores, que realizam a renovação do ar em um determinado ambiente. Esta é crucial em casos em que existem poucas saídas de ar ou em que haja poluição do ar devido à emissão de máquinas industriais. Além disso, profissionais que trabalham em um ambiente cuja temperatura é elevada tendem a reduzir sua produtividade. Pesquisas revelam que os riscos de ocorrência de acidente aumentam quando os operadores das máquinas trabalham sob temperaturas desconfortáveis.

As vantagens que a tecnologia IoT podem oferecer são vastas. É possível fazer o monitoramento real, fornecer dados automáticos das condições ambientais desses locais, receber alertas em casos de alterações ou temperaturas que estejam

abaixo/acima do esperado entre outros, tudo isso sendo gerado a partir de uma interface gráfica e interativa com o usuário, bastando apenas acessar a internet. Isso permite com que não ocorram desperdícios, reduz o custo e melhora a qualidade dos produtos e serviços oferecidos para o consumidor.

## **8.2 Monitoramento da Concentração de Gás**

Existem diversas aplicações que exigem o controle da detecção de gases inflamáveis, tóxicos e asfixiantes em um ambiente. No ramo de alimentos e bebidas, por exemplo, constantemente a existência de gás se faz presente ou até mesmo é gerada a partir das etapas de produção de uma substância, seja no momento de processamento, tratamento de resíduo, armazenamento entre outros. Um dos processos com maior incidência de gás está relacionado ao sistema de refrigeração, pois este é muito suscetível a vazamentos de gás ocasionando riscos aos funcionários do local ou provocando emissões nocivas para a atmosfera.

Outro exemplo que pode ser citado são as indústrias químicas. Estas também geram uma ampla gama de gases inflamáveis e tóxicos em seus processos de fabricação ou em seus subprodutos, provocando incidentes em casos onde não há o seu correto controle e detecção.

As aplicações que envolvem a monitoração da concentração de gás geralmente estão diretamente ligadas à saúde e ao bem-estar do ser humano. Por este motivo, é importante que haja o monitoramento real e até mesmo a sinalização em casos de vazamento de gás para impedir que desastres aconteçam.

Uma das tecnologias que podem proporcionar soluções e melhorias para esses ambientes é a internet das coisas, capaz de proporcionar eficiência, praticidade e rapidez para coletar dados, processar informações e tomar medidas preventivas.

## **8.3 Monitoramento da Presença de Fogo**

Outra forma de controle importante a ser feita diz respeito à prevenção e o combate a incêndios que deve ser realizada em diversos tipos de ambientes. Alguns exemplos, além de edifícios comerciais e residenciais, são os museus e bibliotecas, que precisam ter uma segurança em suas instalações para que sejam

preservados os patrimônios e a cultura do local bem como a saúde física das pessoas. Nestes ambientes, faz-se necessário a existência de tecnologias capazes de detectar rapidamente a incidência de fogo e ferramentas de prevenção, alguns exemplos são: sinalizações de rotas de fuga, iluminação, extintores e hidrantes, portas corta fogo, detectores de fumaça e afins.

Uma aplicação interessante foi realizada nos jogos olímpicos do Rio em 2016, onde as 24 salas de tecnologias, que eram responsáveis pelo processamento e transmissão dos jogos de diferentes modalidades, tiveram suas estruturas protegidas contra o incêndio utilizando os princípios da Internet das Coisas, com os chips M2M. A tecnologia utilizada foi uma ampola do *Bonpet* que, apesar de ser de origem japonesa, já está presente e instalada no Brasil há alguns anos. Seu funcionamento ocorre da seguinte forma: quando a temperatura do ambiente sobe ou detecta algum sinal de fumaça nas salas, a ampola aquece (simulando a chama de fogo) e inicia o processo de liberação do líquido *Bonpet* (processo de extinção). Apesar de ter sido implantado nesse evento em específico, esta tecnologia pode ser incorporada em indústrias, escritórios com monitoramento à distância e até em pequenas residências.

#### **8.4 Consumo de Energia Elétrica**

Realizar o controle sobre o consumo de energia elétrica é uma das principais atuações da IoT atualmente e as perspectivas tendem a crescer para o setor elétrico.

Segundo Paulo Victor Peixoto Noronha, gestor de oferta TI corporativo da Oi: No setor de energia são instalados sensores no ambiente que coletam dados como corrente e voltagem, seja da energia que vem da distribuidora, seja da que sai do quadro elétrico para os equipamentos que devem ser monitorados. Através dessas detecções é possível saber em tempo real como estão sendo distribuída a energia nos pontos de consumo (lojas, casas, ruas), quais são as falhas elétricas existentes ou os pontos considerados mais críticos entre outros. Dessa forma, é possível solucionar os problemas de forma mais fácil e rápida, reduzir custos e atenuar os efeitos através da instalação de geradores ou baterias em locais considerados críticos, que sejam mais suscetíveis a falhas.

A *smart grid* é um exemplo que faz com que as distribuidoras tenham informações detalhadas sobre o comportamento de seus clientes, o que permite garantir a qualidade de energia e antecipar demandas de manutenção. Automação de subestações, instalação de medidores e tomadas inteligentes são outros exemplos que auxiliam no monitoramento da rede e maximizam os benefícios da IoT na operação.

Conclui-se que medir o consumo de energia elétrica é essencial em qualquer instalação desde aplicações simples como uma residência, onde se pode medir a potência dos aparelhos utilizados, até em aplicações mais complexas como uma subestação ou concessionária. Em ambos os casos isso gera redução de custos, eficiência energética e até mesmo contribui para a sustentabilidade.

Vale salientar que um dos pontos mais importantes neste processo é a correção do fator de potência para um nível acima do exigido pela concessionária (atualmente 0,92), uma vez que um baixo fator de potência acarreta em um desperdício de energia elétrica. Este processo é realizado com o intuito de impedir uma cobrança adicional. Nem todos os consumidores, porém, pagam por esse tipo de multa. Em geral, a maioria das residências de pequeno porte está isenta de pagar uma multa, enquanto condomínios (residenciais ou comerciais), indústrias de pequeno, médio e grande porte, pagam um adicional todos os meses junto com as suas contas de consumo.

## 9 CONCLUSÃO

A crescente conectividade e a busca pela comunicação entre dispositivos eletrônicos proporcionam cada vez mais espaço para que a Internet das Coisas seja inserida nos diversos ramos da sociedade. Pode-se dizer que esta tecnologia melhora o cotidiano das pessoas, promovendo conforto, praticidade, agilidade e eficiência para seus usuários. O conceito de Edifícios Inteligentes vai de encontro com esta ideia e tem a função de aplicar a IoT para criar serviços mais inteligentes e interdependentes, capazes de processar informações do ambiente e assim tomar ações pré-programadas de acordo com a necessidade do local.

As possibilidades para os edifícios inteligentes são vastas, uma vez que existem diferentes tipos com funções, demandas e necessidades diferenciadas. Podem-se citar hospitais, *shopping centers*, indústrias, comércios, residências e outras diferentes variedades de escolhas de locais para empregarem esta recente tecnologia. Sabendo disto, o presente projeto teve como finalidade utilizar aplicações que, apesar de simples e gerais, pudessem ser implantadas em praticamente todos os tipos de edifícios.

A proposta do projeto foi realizar um estudo teórico e prático para que o leitor tivesse um melhor entendimento do assunto abordado. Devido a isso, criou-se um protótipo que acomodasse todos os dispositivos utilizados, tanto a parte dos sensores e atuadores quanto a do microcontrolador. Este, por sua vez, foi escolhido devido a sua praticidade, suporte e eficiência para atender toda a ideia do grupo.

Cada componente do circuito foi testado individualmente para o perfeito funcionamento do protótipo. Através destes testes, o desempenho dos sensores foi considerado satisfatório, de acordo com o esperado para atender o objetivo central do trabalho. Após esta etapa, o sistema foi unificado, sendo esta considerada a parte mais complicada, devido à necessidade de integrar em perfeita harmonia todos os dispositivos utilizados: sensores, atuadores, ESP32 e plataforma Node-RED. Apesar de alguns inconvenientes, todos os problemas foram solucionados e o objetivo final foi alcançado com êxito.

Uma das aplicações mais importantes do protótipo está relacionada ao consumo de energia, uma vez que isto impacta diretamente nas despesas do

consumidor. Além disso, métodos para economia de energia são de suma importância atualmente, uma vez que a demanda de energia elétrica no Brasil está aumentando consideravelmente e a capacidade instalada não está acompanhando este crescimento.

Ao verificar os valores de tensão e corrente obtidos pelos sensores, pôde-se perceber que estes são confiáveis uma vez que não apresentaram grandes variações quando comparados aos instrumentos de medição utilizados para a conferência. Portanto, conclui-se que a medição da potência está condizente com a realidade. Porém, é importante deixar claro que a medição do consumo realizado neste trabalho foi baseada em uma carga resistiva, cujo fator de potência é equivalente a 1. Para cargas reativas como, por exemplo, motores elétricos, transformadores, elevadores entre outros, o FP é diferente de 1, sendo necessário verificar qual é o fator correspondente para que seja levado em conta ao calcular-se a potência.

A fim de ampliar a pesquisa e aperfeiçoar o protótipo, pode-se mencionar algumas sugestões para implementações futuras. A primeira delas é inserir o cálculo do fator de potência da carga ligada na tomada, que pode ser realizada através da medição do tempo de resposta dos sensores de tensão e de corrente. Dessa forma, seria possível verificar se o mesmo está dentro do valor estabelecido por norma e, em caso negativo, é possível tomar medidas para a correção deste fator, evitando multas. A segunda seria acrescentar sensores/atuadores com novas funcionalidades como, por exemplo, fazer o controle da climatização do ambiente, do ar condicionado entre outros. A terceira seria buscar novas formas de interação e comunicação com o usuário, aprimorando com isso a *dashboard* gerada.

## 10 REFERENCIAS BIBLIOGRÁFICAS

ARDUINO. **Arduino Software (IDE)**. Disponível em: <<https://www.arduino.cc/en/Guide/Environment#serialmonitor>>. Acesso em: 13 set. 2018.

ARDUINO E CIA. **Alarme sensor de gás com o módulo MQ-2**. Disponível em: <<https://www.arduinoocia.com.br/2015/01/alarme-sensor-de-gas-modulo-mq-2.html>>. Acesso em: 13 set. 2018.

CANAL WESTCON. **Como a Internet das Coisas irá transformar os edifícios inteligentes?** Disponível em: <<https://blogbrasil.westcon.com/como-a-internet-das-coisas-ira-transformar-os-edificios-inteligentes>>. Acesso em: 20 set. 2018.

CANAL WESTCON. **O que são Edifícios Inteligentes?** Disponível em: <<https://blogbrasil.westcon.com/o-que-sao-edificios-inteligentes>>. Acesso em: 25 set. 2018.

CANDIDO, Gradimilo. **Sensor de gás MQ-135 e a família de sensores MQ**. Disponível em: <<https://portal.vidadesilicio.com.br/sensor-de-gas-mq-135/>>. Acesso em: 25 set. 2018.

COELHO, Darlene Figueiredo Borges; CRUZ, Victor Hugo do Nascimento. **Edifícios Inteligentes: uma visão das tecnologias aplicadas**. São Paulo: Blücher, 2017.

CREDER, Hélio. **Instalações Elétricas**, 16ª edição. LTC, 02/2016.

CTE. **Eldorado Business Tower é o primeiro edifício certificado Leed® Platinum da América Latina**. Disponível em: <<http://www.cte.com.br/noticias/2009-09-03-eldorado-business-tower-e-o-primeiro-e/>>. Acesso em: 02 set. 2018.

CURTO CIRCUITO. **Módulo Sensor de Tensão AC - ZMPT101B**. Disponível em: <<https://www.curtocircuito.com.br/modulo-sensor-de-tensao-ac-zmpt101b.html>>. Acesso em: 10 set. 2018.

CURVELLO, André. **ESP32 – Um grande aliado para o Maker IoT**. Disponível em: <<https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>>. Acesso em: 02 set. 2018.

DEMETRAS, Ezequiel. **SCT-013 - Sensor de Corrente Alternada com Arduino**. Disponível em: <<https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>>. Acesso em: 22 out. 2018.

ELETROGATE. **Sensor de chama / fogo**. Disponível em: <<https://www.eletrogate.com/sensor-de-chama-fogo>>. Acesso em: 22 out. 2018.

EPE. **Projeção da demanda de energia elétrica para os próximos 10 anos (2017-2026)**. Disponível em: <[http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-245/topico-261/DEA%20001\\_2017%20-%20Proje%C3%A7%C3%B5es%20da%20Demanda%20de%20Energia%20El%C3%A9trica%202017-2026\\_VF\[1\].pdf](http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-245/topico-261/DEA%20001_2017%20-%20Proje%C3%A7%C3%B5es%20da%20Demanda%20de%20Energia%20El%C3%A9trica%202017-2026_VF[1].pdf)>. Acesso em: 25 set. 2018.

ESPRESSIF SYSTEMS. **About *Espressif***. Disponível em: <<https://www.espressif.com/en/company/about-us/who-we-are>>. Acesso em: 02 set. 2018.

ESPRESSIF SYSTEMS. **ESP32 – A Different IoT Power and Performance**. Disponível em: <<https://www.espressif.com/en/products/hardware/esp32/overview>>. Acesso em: 02 set. 2018.

EVANS, Dave. **A Internet das Coisas: Como a próxima evolução da Internet está mudando tudo**. Disponível em: <[https://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iiot\\_ibsg\\_0411final.pdf](https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iiot_ibsg_0411final.pdf)>. Acesso em: 19 mai. 2018.

FILHO, SILVA, Matheus da. **Fundamentos de Eletricidade**. LTC, 02/2007.

FILIPEFLOP. **Sensor de Corrente Não Invasivo 20A SCT-013**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-corrente-nao-invasivo-20a-sct-013/>>. Acesso em: 07 out. 2018.

FILIPEFLOP. **Sensor de Chama Fogo 760 a 1100 nm**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-chama-fogo/>>. Acesso em: 07 out. 2018.

FILIPEFLOP. **Sensor de Gás MQ-2 Inflamável e Fumaça**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-gas-mq-2-inflamavel-e-fumaca/>>. Acesso em: 07 out. 2018.

FILIPEFLOP. **Sensor de Umidade e Temperatura DHT11**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/#tab-description>>. Acesso em: 08 out. 2018.

GALERIA DA ARQUITETURA. **Eldorado Business Tower**. Disponível em: <<https://www.galeriadaarquitetura.com.br/slideshow/newslideshow.aspx?idproject=2177&index=4>>. Acesso em: 09 out. 2018.

IBM CLOUD. **How the IBM Cloud console works**. Disponível em: <<https://console.bluemix.net/docs/overview/ui.html#ui>>. Acesso em: 22 ago. 2018.

INTERNATIONAL ENERGY AGENCY. **Global Status Report 2017**. Disponível em: <<https://europa.eu/capacity4dev/unep/documents/global-status-report-2017>>. Acesso em: 16 mar. 2018.

LACERDA, Flávia. **Arquitetura da Informação Pervasiva: projetos de ecossistemas de informação na Internet das Coisas.** Disponível em: <http://repositorio.unb.br/handle/10482/19646>>. Acesso em: 20 set. 2018.

LIMA, Izabelle. **Módulo Relé.** Disponível em: <http://autocorerobotica.blog.br/modulo-rele/>>. Acesso em: 10 out. 2018.

LOBITSKY, Gabriel. **Edifícios inteligentes: faça as malas, você vai morar em um.** Disponível em: <http://computerworld.com.br/edificios-inteligentes-faca-malas-voce-vai-morar-em-um>>. Acesso em: 02 jun. 2018.

LOES, João; STEINHAUSER, Rafael. **O que é a internet das coisas e como ela mudará suas vidas.** Disponível em: <http://tecnologia.terra.com.br/o-que-e-a-internet-das-coisas-e-como-ela-mudara-a-sua-vida,3e61c3b90c8ca410VgnVCM3000009af154d0RCRD.html>>. Acesso em: 08 de set. 2018.

MADEIRA, Daniel. **Primeiros passos com o Node-RED e Arduino UNO.** Disponível em: <https://www.filipeflop.com/blog/primeiros-passos-node-red-arduino-uno/>>. Acesso em: 05 set. 2018.

MADEIRA, Daniel. **Criando dashboard para Arduino no Node-RED.** Disponível em: <https://www.filipeflop.com/blog/dashboard-para-arduino-no-node-red/>>. Acesso em: 07 set. 2018.

MANCINI, Mônica. **Internet das Coisas: História, Conceitos, Aplicações e Desafios.** Disponível em: <https://pmisp.org.br/slideshow/2617-internet-das-coisas>>. Acesso em: 02 jun. 2018.

MNATI, Mohannad Jabbar; BOSSCHE, Alex Van den Bossche; CHISAB, Raad Farhood. **A Smart Voltage and Current Monitoring System for Three Phase Inverters Using an Android Smartphone Application.** Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5424749/>>. Acesso em: 14 out. 2018.

MAKER.IO. **Introduction to the Arduino IDE.** Disponível em: <https://www.digikey.com/en/maker/blogs/2018/introduction-to-the-arduino-ide>>. Acesso em: 13 set. 2018.

MASTERWALKER ELECTRONIC SHOP. **Sensor Detector de Fogo / chama Infravermelho.** Disponível em: <https://www.masterwalkershop.com.br/sensor-detector-de-fogochama-infravermelho-5mm>>. Acesso em: 07 out. 2018.

MOURA, Victor de Macedo. **O que é Computação Cognitiva?** Disponível em: <https://blog.betalabs.com.br/o-que-e-computacao-cognitiva/>>. Acesso em: 13 set. 2018.

MOTA, KLEBER. **O que é Arduino? Anatomia da placa Uno e do IDE.** Disponível em: <https://www.klebermota.eti.br/2017/06/23/o-que-e-arduino-anatomia-da-placa-uno-e-do-ide/>>. Acesso em: 13 set. 2018.

NAKAMURA, Juliana. **Monitoramento reduz o consumo de energia em prédios corporativos.** Disponível em:

<[https://www.aecweb.com.br/cont/m/rev/monitoramento-reduz-o-consumo-de-energia-em-predios-corporativos\\_15084\\_10\\_0](https://www.aecweb.com.br/cont/m/rev/monitoramento-reduz-o-consumo-de-energia-em-predios-corporativos_15084_10_0)>. Acesso em: 14 out. 2018.

NETTO, Luiz Ferraz. **Sinais Elétricos e suas Formas de Onda.** Disponível em: <[http://www.feiradeciencias.com.br/sala15/15\\_07.asp](http://www.feiradeciencias.com.br/sala15/15_07.asp)>. Acesso em: 14 out. 2018.

NODE-RED. **Node-RED - Flow-based programming for the Internet of Things.** Disponível em: <<https://nodered.org/#features>>. Acesso em: 01 set. 2018.

NUNES, Renato; SÊRRO, Carlos. **Edifícios Inteligentes: Conceitos e Serviços.** Disponível em: <[http://www.domobus.net/ei\\_docs/edif\\_int.pdf](http://www.domobus.net/ei_docs/edif_int.pdf)>. Acesso em: 16 mar. 2018.

OLIVEIRA, Euler. **Conhecendo o NodeMCU-32S ESP32.** Disponível em: <<http://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32/>>. Acesso em: 21 ago. 2018.

PBE EDIFICA. Manual para o Entendimento da Etiquetagem de Edificações pelo Gestor Público. Disponível em: <[http://www.pbeedifica.com.br/sites/default/files/Manual\\_Gestor\\_Publico\\_20140613\\_1.pdf](http://www.pbeedifica.com.br/sites/default/files/Manual_Gestor_Publico_20140613_1.pdf)>. Acesso em: 14 out. 2018.

PORTAL VGV. **Rio de Janeiro ganha primeiro prédio verde na Categoria Gold: Ventura Corporate Towers.** Disponível em: <<https://www.portalvgv.com.br/site/rio-de-janeiro-ganha-primeiro-predio-verde-na-categoria-gold-ventura-corporate-towers>>. Acesso em: 14 out. 2018.

PROCEL INFO. **Selo PROCEL Edificações.** Disponível em: <<http://www.procelinfo.com.br/main.asp?View={8E03DCDE-FAE6-470C-90CB-922E4DD0542C}>>. Acesso em: 14 out. 2018.

REVISTA MEIO AMBIENTE – INDUSTRÍA E SUSTENTABILIDADE. **IoT e faz surgir nova geração de edifícios inteligentes.** Disponível em: <<http://rmai.com.br/iot-e-faz-surgir-nova-geracao-de-edificios-inteligentes/>>. Acesso em: 20 set. 2018.

REYES, Angel Tomala. **O que é IBM Cloud?** Disponível em: <<https://www.ibm.com/developerworks/br/cloud/library/cl-bluemixfoundry/index.html>>. Acesso em: 21 ago. 2018.

ROBOCORE. **ESP32 - WiFi & Bluetooth.** Disponível em: <<https://www.robocore.net/loja/produtos/esp32-wifi-bluetooth.html>>. Acesso em: 15 set. 2018.

SANTOS, Bruno P; SILVA, Lucas A. M.; CELES, Clayson S. F. S.; NETO, João B. Borges; PERES, Bruna S.; VIEIRA, Marcos Augusto M.; VIEIRA, Luiz Filipe M.; GOUSSEVSKAIA, Olga N.; LOUREIROINTERNET, Antonio A. F. **Internet das Coisas: da Teoria à Prática.** Disponível em: <<http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>. Acesso em: 19 mai. 2018.

SANTOS, Mariana Feres dos; ABASCAL, Eunice Helena Sguizzardi. **Certificação LEED e arquitetura sustentável: edifício Eldorado Business Tower.** Disponível em: <<http://www.vitruvius.com.br/revistas/read/arquitextos/12.140/4126>>. Acesso em: 14 out. 2018.

SATO, S. K. **Mobilidade, comunicação e consumo: expressões da telefonia celular em Angola, Brasil e Portugal.** Disponível em: <<http://www.teses.usp.br/teses/disponiveis/27/27153/tde-01062015-163806/pt-br.php>>. Acesso em: 13 de set. 2018.

SCARLETT, Kane. **Fundamentos do IBM Cloud: Inclua um mundo de serviços de nuvem em seu aplicativo.** Disponível em: <<https://www.ibm.com/developerworks/br/cloud/library/cl-bluemix-fundamentals-introducing-cloud-services/index.html>>. Acesso em: 21 ago. 2018.

SEGURA, Mauro. **Bradesco é estrela da campanha mundial da IBM.** Disponível em: <<http://www.maurosegura.com.br/bradesco-e-estrela-da-campanha-mundial-da-ibm/>>. Acesso em: 01 set. 2018.

SOUZA, Carlos Leite de; AWAD, Juliana Di Cesare Marques. **Cidades sustentáveis, cidades inteligentes:** Desenvolvimento sustentável num planeta urbano. Porto Alegre: Bookman, 2012.

VERTIGO TECNOLOGIA. **O que é API? Entenda de uma maneira simples.** Disponível em: <<https://vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples>>. Acesso em: 08 set. 2018.

TANG, Chun Bin. **Explore o MQTT e o serviço de Internet of Things no IBM Cloud.** Disponível em: <<https://www.ibm.com/developerworks/br/cloud/library/cl-mqtt-bluemix-iot-node-red-app/index.html>>. Acesso em: 18 jul. 2018.

THOMSEN, Adilson. **Controlando lâmpadas com Módulo Relé Arduino.** Disponível em: <<https://www.filipeflop.com/blog/controle-modulo-rele-arduino/>>. Acesso em: 14 out. 2018.

THOMSEN, Adilson. **Tutorial Sensor de Chama com Arduino.** Disponível em: <<https://www.filipeflop.com/blog/sensor-de-chama-com-arduino/>>. Acesso em: 14 out. 2018.

VENTURA CORPORATE TOWERS. **Ventura Corporate Towers.** Disponível em: <<http://br.tishmanspeyer.com/properties/ventura-corporate-towers>>. Acesso em: 14 out. 2018.

YUAN, Michael. **Conhecendo o MQTT.** Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 18 jul. 2018.

WU, Geng; TALWAR, Shilpa; JOHNSON, Kerstin; HIMAYAT, Nageen; JOHNSON, Kevin. ***M2M: From mobile to embedded internet.*** IEEE Commun. Mag., [s.l.], v. 49, n. 4, abr. 2011. Institute of Electrical & Electronics Engineers (IEEE). Disponível em: <<http://dx.doi.org/10.1109/mcom.2011.5741144>>. Acesso em: 25 set. 2018.

## APÊNDICE A

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <DNSServer.h>
#include <WebServer.h>
#include <WiFiManager.h>
#include <DHT.h>
#include "EmonLib.h"

EnergyMonitor emon1;

// Altere estes dados pelos obtidos ao cadastrar o novo dispositivo no Watson IoT
const String ORG = "zgm00f";
const String DEVICE_TYPE = "ESP32";
const String DEVICE_ID = "001";
#define DEVICE_TOKEN "tcctoken"

// Pinos dos Relés
#define pinRele1 25
#define pinRele2 26
#define pinLED 12 //LED para mostrar conexão

const String CLIENT_ID = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
const String MQTT_SERVER = ORG + ".messaging.internetofthings.ibmcloud.com";
const String QUICK_START = "d:quickstart:arduino:";

#define COMMAND_TOPIC_R1 "iot-2/cmd/commandR1/fmt/json"
#define COMMAND_TOPIC_R2 "iot-2/cmd/commandR2/fmt/json"
#define TOPIC_NAME "iot-2/evt/status/fmt/json"
#define INTERVAL 100

WiFiClient wifiClient;
PubSubClient client(MQTT_SERVER.c_str(), 1883, wifiClient);

//Tempo em que o último envio foi feito
long lastPublishTime = 0;

//Objeto que realiza a leitura da temperatura e da umidade
DHT dht(14, DHT11);

//Variável para guardarmos o valor da temperatura
float temperature = 0;

//Variável para guardarmos o valor da umidade
float humidity = 0;

//sensor de chama

```

## APÊNDICE A

```

float pino_chama = 21;
float fire = 0;

//sensor de gás
float pin_gas = 32;
float gas = 0;

//sensor de corrente
float pino_sct = 15;
float current = 0;

//sensor de tensão
#define VOLT_CAL 148.7
float voltage = 0;

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(pinRele1, OUTPUT);
  pinMode(pinRele2, OUTPUT);
  pinMode(pinLED, OUTPUT);

  Serial.begin(9600);
  digitalWrite(pinLED, LOW);

  Serial.println("Lendo dados do sensor...");

  Serial.begin(9600);
  pinMode(pino_chama, INPUT);

  Serial.begin(9600);
  pinMode(pin_gas, INPUT);

  Serial.begin(9600);
  emon1.current(pino_sct, 19);

  Serial.begin(9600);

  emon1.voltage(34, VOLT_CAL, 1.7);

  WiFiManager wifiManager;
  wifiManager.setAPCallback(configModeCallback);
  wifiManager.setSaveConfigCallback(saveConfigCallback);
  //wifiManager.resetSettings();
  wifiManager.autoConnect("BI-ESP-AP");
  connectMQTTServer();
}

```

## APÊNDICE A

```

void loop() {
  client.loop();
  {
    //Tempos agora em milisegundos
    long now = millis();

    //Se o tempo desde o último envio for maior que o intervalo de envio
    if (now - lastPublishTime > INTERVAL) {
      //Atualizamos o tempo em que ocorreu o último envio
      lastPublishTime = now;
      //Fazemos a leitura da temperatura e umidade
      readSensorDHT11();
      Serial.print("Publish message: ");
      //Criamos o json que enviaremos para o server mqtt
      String msg = createJsonString();
      Serial.println(msg);
      //Publicamos no tópico onde o servidor espera para receber
      //e gerar o gráfico
      client.publish(TOPIC_NAME, msg.c_str());
    }
    delay (500);
  }
  {
    long now = millis();
    if (now - lastPublishTime > INTERVAL) {

      lastPublishTime = now;

      readSensorChama();

      Serial.print("Publish message: ");

      String msg = createJsonString();
      Serial.println(msg);

      client.publish(TOPIC_NAME, msg.c_str());
    }
    delay (500);
  }
  {
    long now = millis();
    if (now - lastPublishTime > INTERVAL) {

      lastPublishTime = now;

      readSensorGas();
      Serial.print("Publish message: ");

      String msg = createJsonString();

```

**APÊNDICE A**

```

Serial.println(msg);

  client.publish(TOPIC_NAME, msg.c_str());
}
delay (500);
}

{
  long now = millis();
  if (now - lastPublishTime > INTERVAL) {

    lastPublishTime = now;

    readSensorCorrente();
    Serial.print("Publish message: ");

    String msg = createJsonString();
    Serial.println(msg);

    client.publish(TOPIC_NAME, msg.c_str());
  }
  delay (500);
}

{
  long now = millis();
  if (now - lastPublishTime > INTERVAL) {

    lastPublishTime = now;

    void readSensorTensao();
    Serial.print("Publish message: ");

    String msg = createJsonString();
    Serial.println(msg);

    client.publish(TOPIC_NAME, msg.c_str());
  }
  delay (500);
}
}

//Função responsável pela conexão ao servidor MQTT
void connectMQTTServer() {
  Serial.println("Conectando ao servidor MQTT...");

  if (client.connect(CLIENT_ID.c_str(), "use-token-auth", DEVICE_TOKEN)) {
    Serial.println("Conectado ao Broker MQTT...");
  }
}

```

**APÊNDICE A**

```

client.setCallback(callback);

client.subscribe(COMMAND_TOPIC_R1);
client.subscribe(COMMAND_TOPIC_R2);
digitalWrite(pinLED, HIGH);
} else {
  Serial.print("erro = ");
  Serial.println(client.state());
  connectMQTTServer();
}
}

void callback(char* topic, unsigned char* payload, unsigned int length) {
  Serial.print("topico ");
  Serial.println(topic);

  StaticJsonBuffer<30> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(payload);

  if ( !root.success() ) {
    Serial.println("Erro no Json Parse");
    return;
  }

  int value = root["value"];

  if (strcmp(topic, COMMAND_TOPIC_R1) == 0) {
    digitalWrite(pinRele1, value);
  } else {
    digitalWrite(pinRele2, value);
  }
}

void configModeCallback( WiFiManager *myWiFiManager) {
  Serial.println("Entrou no modo de configuração");
  Serial.println(WiFi.softAPIP());
  Serial.println(myWiFiManager->getConfigPortalSSID());
}

void saveConfigCallback() {
  Serial.println("Configuração salva");
  Serial.println(WiFi.softAPIP());
}

//Função responsável por realizar a leitura
//da temperatura e umidade
void readSensorDHT11() {
  float value;
  //Faz a leitura da temperatura
  value = dht.readTemperature();
}

```

## APÊNDICE A

```

//Se o valor lido é válido
if (!isnan(value)) {
  //Armazena o novo valor da temperatura
  temperature = value;
}
//Faz a leitura da umidade
value = dht.readHumidity();
//Se o valor for válido
if (!isnan(value)) {
  //Armazena o novo valor da umidade
  humidity = value;
}
delay (1000);
}
void readSensorChama() {
  float value;
  value = digitalRead(pino_chama);
  //Se o valor lido é válido
  if (!isnan(value)) {
    //Armazena o novo valor da temperatura
    fire = value;
  }
  delay (1000);
}
void readSensorGas() {
  float value;
  value = analogRead(pin_gas);
  value = map(value, 0, 4095, 0, 100);
  //Se o valor lido é válido
  if (!isnan(value)) {
    //Armazena o novo valor da temperatura
    gas = value;
  }
  delay (1000);
}

void readSensorCorrente() {
  float Irms = emon1.calcIrms(1480);
  float value;

  if (!isnan(value)) {
    //Armazena o novo valor da temperatura
    current = Irms;
  }
  delay(1000);
}
void readSensorTensao() {
  emon1.calcVI(20, 500);
  float supplyVoltage = emon1.Vrms;
}

```

**APÊNDICE A**

```
float value;
value = supplyVoltage;

if (!isnan(value)) {
    //Armazena o novo valor da temperatura
    voltage = supplyVoltage;
}
delay(1000);
}

//Função responsável por criar
//um Json com os dados lidos
String createJsonString() {
    String data = "{";
    data += "\"d\": {";
    data += "\"temperature\":";
    data += String(temperature);
    data += ",";
    data += "\"fire\":";
    data += String(fire);
    data += ",";
    data += "\"gas\":";
    data += String(gas);
    data += ",";
    data += "\"current\":";
    data += String(current);
    data += ",";
    data += "\"voltage\":";
    data += String(voltage);
    data += ",";
    data += "\"humidity\":";
    data += String(humidity);
    data += "}";
    data += "}";
    return data;
}
```

### ANEXO I – ORÇAMENTO DO PROTÓTIPO

Componente	Qty.	Valor Unit.	Valor Total
Node MCU-32S ESP32	1	R\$ 64,90	R\$ 64,90
Sensor de Umidade e Temperatura DHT11	1	R\$ 15,90	R\$ 15,90
Sensor de Gás MQ-2 Inflamável e Fumaça	1	R\$ 15,90	R\$ 15,90
Sensor de Chama Fogo 760 a 1100 nm	1	R\$ 9,90	R\$ 9,90
Sensor de Corrente Não Invasivo 20A SCT-013	1	R\$ 58,90	R\$ 58,90
Sensor de tensão AC 0 a 250Vac ZMPT101B	1	R\$ 44,90	R\$ 44,90
Placa Fenolite Perfurada	2	R\$ 2,40	R\$ 4,80
Caixa Plástica Tampa Transparente 100x68x50mm	1	R\$ 26,90	R\$ 26,90
Fonte Ajustável Protoboard	1	R\$ 10,90	R\$ 64,90
<b>Total</b>			<b>R\$ 253,00</b>

## ANEXO II – DATASHEET DO ESP32

### 2. Pin Definitions

## 2. Pin Definitions

### 2.1 Pin Layout



Figure 1: ESP32-WROOM-32 Pin Layout (Top View)

### 2.2 Pin Description

ESP32-WROOM-32 has 38 pins. See pin definitions in Table 2.

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8

## ANEXO II – DATASHEET DO ESP32

### 2. Pin Definitions

Name	No.	Type	Function
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SD0/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD1/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

**Notice:**

\* Pins SCK/CLK, SD0/SD0, SD1/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on the module and are not recommended for other uses.

### 2.3 Strapping Pins

ESP32 has five strapping pins, which can be seen in Chapter 6 Schematics:

## ANEXO II – DATASHEET DO ESP32

### 1. Overview

Categories	Items	Specifications
	Audio	CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I <sup>2</sup> C, LED PWM, Motor PWM, I <sup>2</sup> S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage/Power supply	2.7 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Recommended operating temperature range	-40 °C ~ +85 °C
	Package size	(18.00±0.10) mm x (25.50±0.10) mm x (3.10±0.10) mm

## ANEXO III – DATASHEET DO SENSOR DE TEMPERATURA E UMIDADE

# AOSONG

Temp, Humidity & Dew point measurement experts

### 1、 Product Overview

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity, Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller.



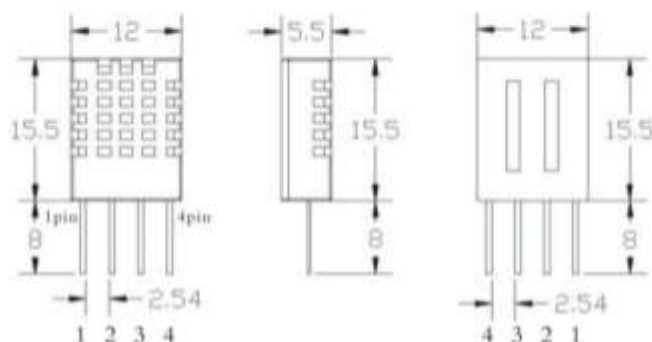
### 2、 Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

### 3、 Features

Low cost, long-term stability, relative humidity and temperature measurement, excellent quality, fast response, strong anti-interference ability, long distance signal transmission, digital signal output, and precise calibration.

### 4、 Dimensions (unit: mm)



**ANEXO III – DATASHEET DO SENSOR DE TEMPERATURA E UMIDADE****AOSONG**

Temp., Humidity &amp; Dew point measurement experts

**5. Product parameters**

## Relative humidity

Resolution: 16Bit

Repeatability:  $\pm 1\%$  RHAccuracy: At 25°C  $\pm 5\%$  RH

Interchangeability: fully interchangeable

Response time: 1 / e (63%) of 25°C 6s

1m / s air 6s

Hysteresis:  $< \pm 0.3\%$  RHLong-term stability:  $< \pm 0.5\%$  RH / yr in

## Temperature

Resolution: 16Bit

Repeatability:  $\pm 0.2^\circ\text{C}$ Range: At 25°C  $\pm 2^\circ\text{C}$ 

Response time: 1 / e (63%) 10S

## Electrical Characteristics

Power supply: DC 3.5 ~ 5.5V

Supply Current: measurement 0.3mA standby 60 $\mu$  A

Sampling period: more than 2 seconds

## Pin Description

1, the VDD power supply 3.5 ~ 5.5V DC

2 DATA serial data, a single bus

3, NC, empty pin

4, GND ground, the negative power

## ANEXO IV – DATASHEET DO SENSOR DE GÁS

### MQ-2 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is  $\text{SnO}_2$ , which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

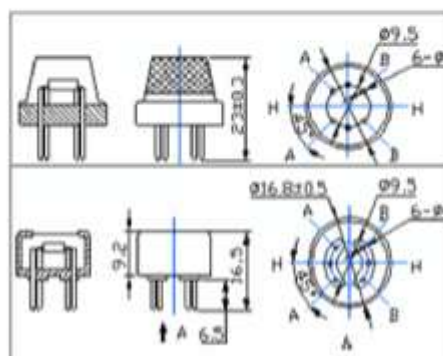
#### Character

- \* Good sensitivity to Combustible gas in wide range
- \* High sensitivity to LPG, Propane and Hydrogen
- \* Long life and low cost
- \* Simple drive circuit

#### Application

- \* Domestic gas leakage detector
- \* Industrial Combustible gas detector
- \* Portable gas detector

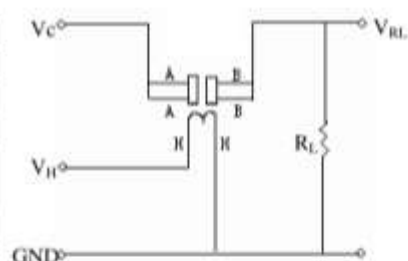
#### Configuration



#### Technical Data

Model No.		MQ-2	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Combustible gas and smoke	
Concentration		300-10000ppm ( Combustible gas)	
Circuit	Loop Voltage	$V_L$	$\leq 24V$ DC
	Heater Voltage	$V_H$	$5.0V \pm 0.2V$ AC or DC
	Load Resistance	$R_L$	Adjustable
Character	Heater Resistance	$R_H$	$31\Omega \pm 3\Omega$ ( Room Tem. )
	Heater consumption	$P_H$	$\leq 900mW$
	Sensing Resistance	$R_S$	$2K\Omega - 20K\Omega$ (in 2000ppm C.I.L. )
	Sensitivity	$S$	$R_S(\text{in air})/R_S(1000\text{ppm isobutane}) \geq 5$
	Slope	$\alpha$	$\leq 0.6 (R_{500\text{ppm}}/R_{3000\text{ppm}} \text{ CH}_4)$
Condition	Tem. Humidity	$20^\circ\text{C} \pm 2^\circ\text{C}$ ; $65\% \pm 5\% \text{RH}$	
	Standard test circuit	$V_C: 5.0V \pm 0.1V$ $V_H: 5.0V \pm 0.1V$	
	Preheat time	Over 48 hours	

#### Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage ( $V_H$ ) and test voltage ( $V_C$ ).  $V_H$  used to supply certified working temperature to the sensor, while  $V_C$  used to detect voltage ( $V_{RL}$ ) on load resistance ( $R_L$ ) whom is in series with sensor. The sensor has light polarity,  $V_C$  need DC power.  $V_C$  and  $V_H$  could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable  $R_L$  value is needed:  
Power of Sensitivity body ( $P_S$ ):  
$$P_S = V_C^2 \times R_S / (R_S + R_L)^2$$

## ANEXO IV – DATASHEET DO SENSOR DE GÁS

Resistance of sensor( $R_s$ ):  $R_s=(V_c/V_{RL}-1)\times R_L$

### Sensitivity Characteristics

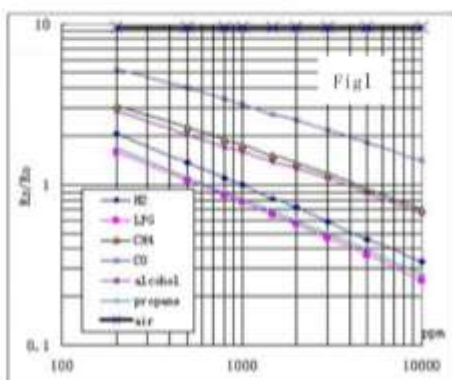


Fig.1 shows the typical sensitivity characteristics of the MQ-2, ordinate means resistance ratio of the sensor ( $R_s/R_0$ ), abscissa is concentration of gases.  $R_s$  means resistance in different gases,  $R_0$  means resistance of sensor in 1000ppm Hydrogen. All test are under standard test conditions.

### Influence of Temperature/Humidity

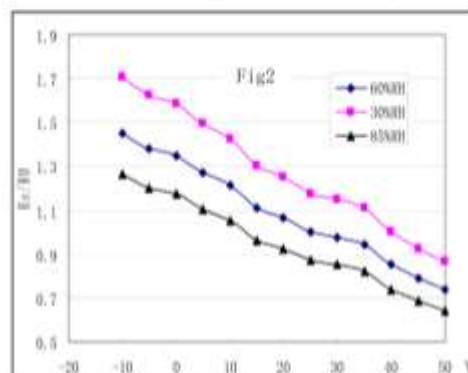
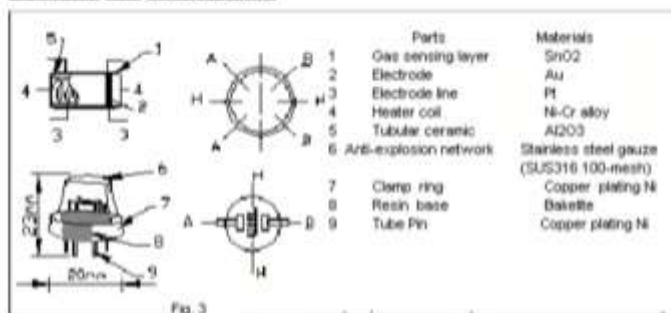


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor ( $R_s/R_0$ ),  $R_s$  means resistance of sensor in 1000ppm Butane under different tem. and humidity.  $R_0$  means resistance of the sensor in environment of 1000ppm Methane, 20°C/65%RH

### Structure and configuration



Structure and configuration of MQ-2 gas sensor is shown as Fig. 3, sensor composed by micro AL<sub>2</sub>O<sub>3</sub> ceramic tube, Tin Dioxide (SnO<sub>2</sub>) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

## ANEXO V – DATASHEET DO SENSOR DE CORRENTE

## Split-Core Current Transformer



## Model: SCT-013 Series

Characteristics: Opening size: 13mm×13mm

Non-linearity±3% ( 10%—120% rated input current )

1.5m leading wire. Diagram for standard three-output

Current output type or Voltage output type  
( Voltage output type built-in sampling resistor )

Purpose: Suitable for the current measuring  
monitoring and protection of AC motor  
lighting equipment air compressor and so on.

Core material: Ferrite

Mechanical strength: The number of switching is not less than 1000 times  
(Test under 20°C).

Safety index: Dielectric strength 6000V AC/1min

Fire resistance property: in accordance with UL94-V0

Working temperature: -25°C—+70°C

Outline size diagram: (in mm)

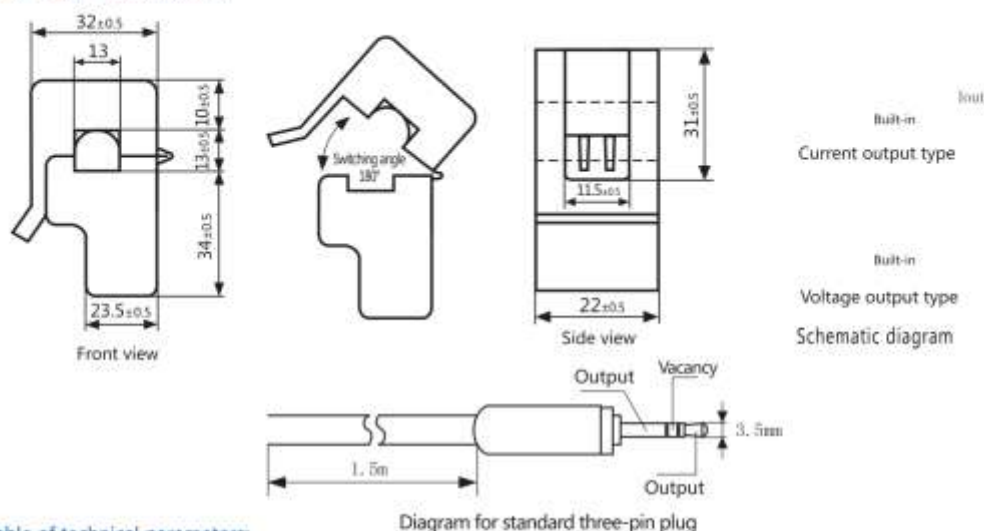


Table of technical parameters:

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output mode	Current/33mA	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-070
Input current	0-25A	0-30A	0-50A	0-60A	
Output mode	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V	

Output mode: Voltage output type built-in sampling resistor; Current output type built-in protective diode;  
Forbidden to be opening operating for current type.

## ANEXO VI – DATASHEET DO SENSOR DE TENSÃO

### ZMPT101B VOLTAGE TRANSFORMER

#### Applications

- \* Sensing Overload Current
- \* Ground fault detection
- \* Metering
- \* Analog to Digital Circuits

#### Electrical Specification

Primary Current	2mA
Secondary Current	2mA
Turns Ratio	1000:1000
Phase Angle Error	$\leq 20'$ (50 $\Omega$ )
Current Range	0 ~ 3mA
Linearity	0.1%
Accuracy Class	0.2
Rated Burden	$\leq 200 \Omega$
Frequency Range	50 ~ 60Hz
Dielectric Level	3000VAC/min
DC Resistance at 20°C	110 $\Omega$

#### Mechanical Specification

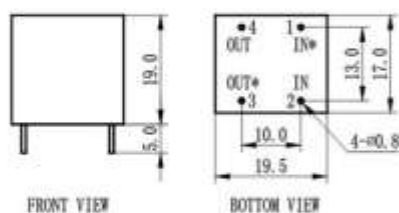
Cup	PBT
Encapsulant	Epoxy
Terminals	Pin $\phi 0.80$ mm
Tolerance	$\pm 0.2$ mm
Approx. Weight	13g
Case	Carton

#### Environment Specification

Storage Temperature	-40°C ~ +130°C
Insulation Resistance	>100 M $\Omega$

#### Dimensions (mm)

Same Polarity \*

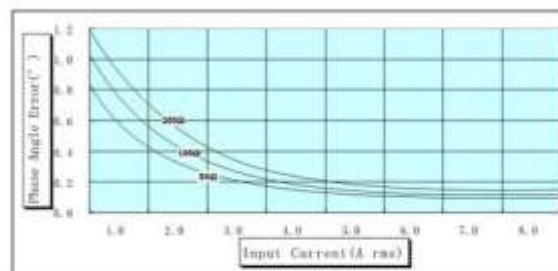
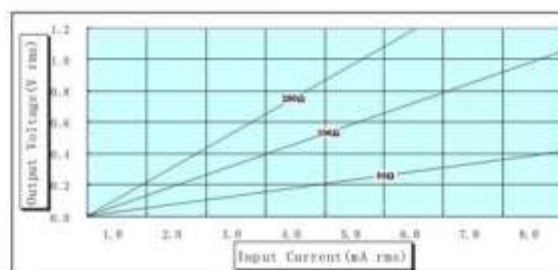


#### Label



Product Illustration

#### Output Characteristic



**Description:** Micro Precision Voltage Transformers, with low price, small size and easy PCB mounting, are mainly used in electrical energy meters, household electrical equipment, industrial apparatuses, electrical testing equipment and relay protection, widely acclaimed as well.