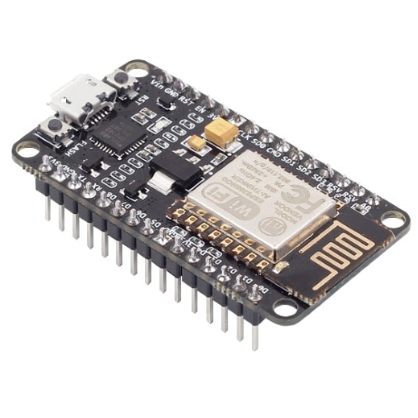
**Métodos**

* **Descrição dos módulos/hardware empregados contendo imagens e correta indicação de referências (citações e referências)**
* **NodeMcu ESP8266**

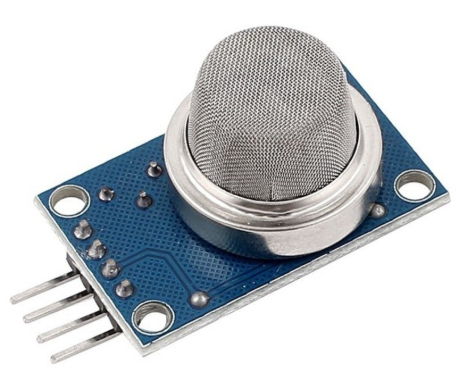
Os módulos utilizados para o desenvolvimento do projeto foram, a development board NodeMcu ESP8266 (Figura 1), onde este é um acoplado de outras duas placas, sendo elas, a NodeMCU Devkit 1.0 e o módulo ESP-12E. O NodeMcu ESP8266 foi utilizado como microcontrolador e principal agente para estabelecimento da comunicação via WiFi do circuito com a rede no sistema, através dele também foi possível o monitoramento da leitura do sensor utilizando o protocolo MQTT, a escolha por este microcontrolador foi devido a sua fama no meio IoT, seu suporte a utilização dos protocolos MQTT e TCP/IP e sua praticidade, assim como citado na publicação sobre a utilização do módulo ESP8266 (Murta, Gustavo, 2018)



**Figura 1. NodeMcu ESP8266**

* **Sensor de gás e fumaça MQ-2**

O sensor MQ-2 (Figura 2) tem a capacidade de detectar a concentração de vários gases combustíveis e fumaça no ambiente, como por exemplo, o Butano, principal componente do gás de cozinha, o sensor também é capaz de captar outros elementos, tais como GLP, Metano, Propano, Hidrogênio, Álcool e Gás Natural, ele capta a quantidade desses elementos através da concentração, em partículas por milhão (ppm) destes presentes no ambiente, sua capacidade de leitura e monitoramento vai de 200 a 10000ppm



**Figura 2. Sensor MQ-2**

* **Buzzer piezo sonalarme s-30v lf**

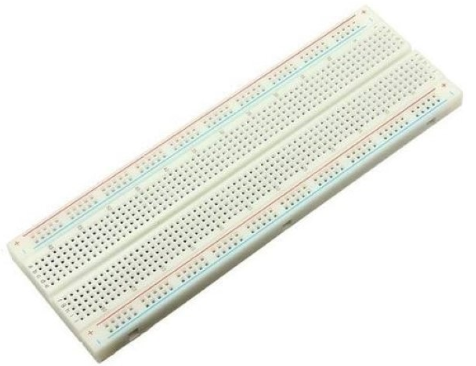
Escolhido para ser o atuador do projeto, o Buzzer (Figura 3) tem como principal objetivo no sistema a notificação de um vazamento de gás por meio sonoro para o indivíduo mais próximo, emitindo um som semelhante a um alarme, seu principal funcionamento é devido ao cristal contido em seu nome, o Piezo, o qual pode ser manipulado aplicando alterações na frequência da voltagem que passa pelos fios da peça, gerando assim diferentes efeitos sonoros, conforme citado na publicação sobre seu funcionamento (DiCola, Tony, 2018)



**Figura 3. Buzzer Piezo**

* **Protoboard 830 pinos**

Utilizada como matriz de contato, a protoboard (Figura 4) é uma peça com base em plástico, responsável pela ligação e transmissão dos sinais elétricos entre os componentes do circuito, através de sua função, nela o sensor foi acoplado diretamente, a placa escolhida foi a de 830 pinos, devido a garantia de espaço suficiente para comportar todos os elementos do projeto



**Figura 4. Protoboard 830 pinos**

* **LED difuso 5mm vermelho**

Utilizado como indicador visual do projeto, se trata de um pino de LED comum (Figura 5), o qual é ligado a todo o circuito através da protoboard e irá responder aos sinais captados pelo sensor através do processamento lógico do projeto



**Figura 5. LED difuso vermelho**

* **Cabos jumper macho/macho e cabos jumper macho/fêmea**

**Responsáveis por interligar, seja através ou não, da protoboard, os cabos jumper são fios feitos de cobre com pinos de conexão em suas pontas, eles estão disponíveis nas composições: macho/macho, macho/fêmea e fêmea/fêmea, no projeto foram utilizados apenas cabos macho/macho (Figura 7) e macho/fêmea (Figura 6)**



**Figura 6. Cabos jumper macho/fêmea**



**Figura 7. Cabos jumper macho/macho**

* **Descrição dos protocolos de comunicação, serviços utilizados, e outros detalhes que sejam importantes para o entendimento das tecnicas utilizadas para o uso do protocolo MQTT**
* **TCP/IP**

O protocolo TCP/IP se baseia em uma comunicação entre servidor e cliente. Basicamente sem este protocolo, não há possibilidade de comunicação entre computadores, pois estes necessitam de uma linguagem padrão, então para isso serve o protocolo TCP/IP. Este protocolo TCP (Protocolo de Controle de Transmissão) tem como função dividir as mensagens que serão enviadas à outra máquina em pequenas mensagens e então encaminhá-las via internet. O protocolo que vem acompanhado, o IP (Protocolo de Interconexão) é um guia, no qual fornece o caminho (endereço) da entrega dos pacotes de mensagens, este protocolo é utilizado pelo protocolo MQTT

* **MQTT**

O protocolo MQTT é responsável por enviar micro mensagens, no formato de char, para dispositivos que podem variar desde sensores até smartphones. Este protocolo consiste em uma comunicação entre servidor e cliente, no qual pode tanto enviar, quanto receber insights.

Ele realiza essa comunicação através de métodos de publish e subscribe, onde cada dispositivo utilizado na comunicação pode utilizar esses métodos para envio e recebimento de informações

Conforme citado no artigo (Automation, Inductive, 2018) “A leveza e eficiência do MQTT tornam possível aumentar significativamente a quantidade de dados que estão sendo monitorados ou controlados”, o protocolo também utiliza e se baseia no protocolo TCP/IP

* **Mosquitto**

O mosquitto é um serviço gratuito que foi utilizado como o serviço broker MQTT do projeto, um broker service é um serviço que recebe as informações geradas pela leitura dos componentes de um sistema, onde estas podem ser processadas e enviadas para outros dispositivos e/ou serviços, foi utilizado por ser uma solução open source, possuir uma boa avaliação por parte dos usuários e ter os requisitos para processamento de informações necessários para o funcionamento do projeto

* **MQTT Dash**

Um aplicativo para smartphone, gratuito, que foi utilizado como serviço receptor das informações provenientes dos tópicos de publish enviados através do mosquitto, utilizando também a biblioteca PubSubClient, o aplicativo suporta a criação de dashboards de monitoramento e controle, onde foram utilizados elementos de monitoramento da atividade de leitura do sensor e alerta para o usuário, nele é onde os tópicos são criados, que seriam os locais para os quais o MQTT consegue enviar suas mensagens, através de métodos de publish e subscribe, cada informação foi monitorada por dois tópicos diferentes, sendo um para monitoramento da leitura do gás presente no ambiente e outro como forma de alerta para o usuário

* **Descrição do software desenvolvido, fluxograma de funcionamento, descrição arquitetural, caso de uso ou qualquer outro item ou diagrama que possa deixar claro o que foi implementado**

O principal objetivo do software desenvolvido é obter os dados de leitura capturados a partir do sensor de gás e fumaça MQ-2, e ter esses dados em constante verificação, e assim que esses dados ultrapassam um determinado valor estabelecido no código o sistema terá a responsabilidade de comunicar ao usuário enviando sinais para alteração de estado de atuadores e também envio constante dos dados para outros serviços e dispositivos por meio do protocolo MQTT

O software utilizado foi desenvolvido através da plataforma Arduino IDE, utilizando as bibliotecas ESP8266WiFi e PubSubClient, onde a primeira oferece uma série de classes, métodos e propriedades para o estabelecimento da comunicação através do WiFi utilizando um microcontrolador que contenha um módulo WiFi, ou apenas um módulo WiFi, enquanto a segunda é responsável por prover um cliente para aplicação dos métodos de publish e subscribe, principais métodos utilizados pelo protocolo MQTT, para envio e recebimento de informações durante a comunicação entre dois ou mais dispositivos

Após a importação das bibliotecas são declaradas as variáveis com as informações para autenticação na rede WiFi assim como a constante que do tipo WiFi, que será utilizada pelos métodos da biblioteca PubSubClient para manipulação das informações através dos métodos de publish e subscribe. Após a declaração das variáveis é feita a definição dos principais métodos de conexão aos principais componentes que estabelecerão a comunicação entre dispositivos diferentes, sendo o WiFi e o MQTT Broker, respectivamente

Os níveis de gás no ambiente serão constantemente capturados pelo sensor MQ-2, e um retorno contendo a quantidade em partículas por milhão (ppm) é exibida no monitor serial, assim como armazenada em uma variável que será utilizada mais tarde no código

São então definidos os pontos de conexão, entradas analógicas e digitais que foram estabelecidas no circuito fisíco, assim como a definição de suas funções (INPUT, OUTPUT), juntamente são também iniciadas a taxa de transferência em bits por segundo para transmissão serial, o estabelecimento do server e porta que serão utilizados pelo MQTT, assim como a chamada da função de call-back que irá ser responsável por construir a relação para recebimento do tópico que será enviado e impressão das informações contidas no payload

Por último, na função de loop, é onde todo o processamento das informações ocorre, primeiramente é feita a declaração das variáveis char que poderão compor conteúdos de payloads, que por sua vez poderão ser enviados junto aos tópicos através dos métodos publish ou subscribe, logo após ocorre a chamada da função que irá tentar estabelecer a conexão com o WiFi e o MQTT Broker Service, após o estabelecimento das conexões é feita a leitura dos dados obtidos pelo sensor através da porta analógica A0, onde estas são exibidas no monitor serial, e também através do método sprintf(), armazenadas concatenadas ao array de char responsável pelo armazenamento da quantidade, em partículas por milhão, de gás presente no ambiente, a qual será enviada por meio de um dos tópicos

Após a captura desse valor, é estabelecida uma estrutura condicional, a qual irá verificar a quantidade de gás presente no ambiente e comparar com um determinado valor, caso esta ultrapasse o valor estabelecido, ocorre o acionamento de uma série de ações, sendo o acionamento do LED, a chamada do método publish para o tópico que irá notificar o usuário através do Dashboard sobre o vazamento de gás utilizando um array de char com uma mensagem como payload e o acionamento do buzzer, respectivamente

Caso contrário, o LED irá se apagar, a mensagem de alerta será alterada, também através do método publish, por uma mensagem informando um estado de normalização do gás presente no ambiente e o buzzer também será desligado, respectivamente, todo esse processo para envio das informações será em função de um delay de 1 segundo, estabelecido ao final da estrutura da função de loop

**Conclusões**

* **Objetivos propostos foram alcançados?**

Sim, conseguimos alcançar nossos objetivos. No início parecia algo quase impossível, pois após poucos dias de aula com esta nova matéria, recebemos a notícia sobre o projeto e de início ficamos assustados pois não conhecíamos quase nada sobre Arduino. Mas com as aulas, estudos, vídeos e artigos adquiridos conseguimos atingir o que era esperado e ficamos muito felizes, pois foi uma grande conquista. Além do projeto influenciar em nossos currículos e vidas profissionais, pela sua grandiosidade e importância do projeto.

Assim como o objetivo de monitoramento da quantidade de gás no ambiente, e o principal objetivo, influenciado pela necessidade de segurança proposta pelo projeto também foi alcançada, assimilando o aviso para o usuário por meios sonoros e visuais, o que possibilitaria a prevenção de acidentes domésticos e/ou empresariais durante o cotidiano

* **Quais as vantagens e desvantagens do projeto?**

As vantagens são várias, porém vamos citar algumas delas. O conhecimento e experiência adquiridos no projeto influenciou muito, pois após a execução do mesmo, notamos a importância não somente do Hardware, mas também o quanto um projeto desses poderia salvar vidas e evitar acidentes. Podemos dizer também que a complexidade foi uma grande vantagem, pois através dela, nos desafiamos e mergulhamos na matéria e no conteúdo para atingir o máximo esperado pelo professor.

Entre as desvantagens temos um imprevisto ocorrido que foi a epidemia causada pelo COVID-19, no qual influenciou bastante pois afastou tanto os alunos quanto os professores, colocando as partes envolvidas afastadas e dificultando algumas atividades. Pelo fato do projeto ser desenvolvido fisicamente, tivemos que dividir algumas tarefas até mesmo para manter a distância social estabelecida pela OMS (Organização Mundial da Saúde) no qual afetou um pouco o desenvolvimento.

* **O que deve/poderia ser feito para melhorar o projeto.**

Bom, o Hardware está totalmente pronto e funcional, porém poderia ser feito algo relacionado ao aspecto visual, que seria no caso implementar uma carcaça, uma aparência visual, cobrindo assim, os fios e todo hardware. Deixando basicamente pronto para utilizar em um domicílio normal, com uma proteção, uma boa aparência.

Juntamente a carcaça, poderíamos também incluir um método de proteção/prevenção adicional, implementando mais um atuador, além do Buzzer, como um braço mecânico, por exemplo, que poderia ser acoplado perto do componente que abre e fecha a passagem do gás de cozinha entre o botijão e o fogão para que os piores cenários sejam evitados de forma automática também

**Resultados:**

* **Descrição do produto final e apresentação do funcionamento:**

O produto final tem como objetivo garantir a segurança assim como o bem-estar das pessoas. O intuito deste produto é prevenir possíveis acidentes e riscos, alertando-os de vazamentos de gases no qual é bastante perigoso e causadores de acidentes atualmente no Brasil. Lembrando que este produto detecta tanto gases quanto fumaças, assim podendo ser utilizado próximo de fornos elétricos, torradeiras, sanduicheiras, além de fogões.

Ele é capaz de detectar diversos tipos de gases e fumaças. O sensor de gás recebe informações sobre o nível de gás e fumaças presente no ar, obtendo esses dados em partículas por milhão(ppm), no qual o valor será armazenado em uma variável. E este será comparado a um valor pré-estabelecido em um estrutura condicional, onde caso atinja esse valor, acionará o restante das funcionalidades do sistema.

Sendo assim ele irá comparar o nível de gás presente no ambiente com o limite estabelecido no código, caso ultrapassado ele ativará um aviso audiovisual através de um LED, um alarme e a notificação pelo Dashboard de monitoramento, alertando da detecção de vazamento de gás e prevenindo possíveis riscos e acidentes.

* **Principais problemas enfrentados e como foram resolvidos?**

Um dos problemas iniciais observados foi pensar na forma como seriam realizados os testes de leitura dos dados pelo sensor, tendo em vista que a fonte de matéria-prima do que iria gerar os dados para o sensor é perigosa quando manipulada fora de um ambiente controlado, que no caso seria o butano, principal componente do gás de cozinha, com isso foi realizada uma pesquisa sobre o funcionamento do sensor MQ-2, onde foi verificado, que uma fonte menor do gás poderia ser utilizada para realização dos testes, sendo esta proveniente de um isqueiro, em que seu fluído é composto por: Butano (C₄H₁₀) Nafta (C₆H₆) Querosene.

Uma das maiores dificuldades presentes durante o desenvolvimento do projeto, foi o fator singular relacionado a pandemia de COVID-19 que se alastrou pelo mundo, devido ao distanciamento social obrigatório, toda a parte de hardware teve que ser desenvolvida apenas por um componente do grupo, havendo auxílio apenas remoto da outra parte, porém, isso foi contornado com o planejamento e divisão das tarefas, onde um dos componentes do grupo ficou responsável por toda a documentação e desenvolvimento do artigo, enquanto a outra ficou responsável pelo desenvolvimento do hardware e código do projeto, onde ao final houve a necessidade de integração entre ambas as partes, para preenchimento de lacunas no artigo e um maior acordo entre o funcionamento do projeto físico e a proposta que havia sido estabelecida

* **Referências**

Murta, Gustavo (2018) “Guia completo do NodeMCU – ESP12 – Usando Arduino IDE”, <https://blog.eletrogate.com/nodemcu-esp12-usando-arduino-ide-2/>, March

Automation, Inductive (2018) “What is MQTT?”, <https://inductiveautomation.com/resources/article/what-is-mqtt>, July