

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM INTERNET DAS COISAS

JORDANO CHEROBIM

**PLATAFORMA PARA CONTROLE DE CONCENTRAÇÃO DE GÁS
GLP**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2018

JORDANO CHEROBIM

**PLATAFORMA PARA CONTROLE DE CONCENTRAÇÃO DE GÁS
GLP**

Monografia de Especialização, apresentada ao Curso de Especialização em Internet das Coisas, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Marcos Eduardo Pivaro Monteiro

CURITIBA

2018



UNIVERSIDADE

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Diretoria de Pesquisa e Pós-graduação
Departamento Acadêmico de Eletrônica
Curso de Especialização em Internet das Coisas



TERMO DE APROVAÇÃO

PLATAFORMA PARA CONTROLE DE CONCENTRAÇÃO DE GÁS GLP

por

JORDANO CHEROBIM

Esta monografia foi apresentada em 30 de novembro de 2018 como requisito parcial para a obtenção do título de Especialista em Internet das Coisas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Marcos Eduardo Pivaro Monteiro
Orientador

Prof. M.Sc. Danillo Leal Belmonte
Membro titular

Prof. M.Sc. Omero Francisco Bertol
Membro titular

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica -

Dedico este trabalho aos meus pais Ivano Cherobim e Luci Maria dos Santos Cherobim, e à minha namorada Fernanda de Goes Furmam que tanto me apoiaram em todos os momentos.

AGRADECIMENTOS

Agradeço à Deus pela força e disposição para sair de minha cidade aos finais de semana e conseguir superar todos os desafios encontrados pelo caminho.

Agradeço aos meus colegas pela companhia e compartilhamento de conhecimentos que me auxiliaram nas mais diversas disciplinas durante o curso.

Agradeço também ao Prof. Dr. Marcos Eduardo Pivaro Monteiro primeiramente pela passagem de conhecimento durante suas disciplinas e também por ter aceitado prestar a orientação para esta monografia, pela qual dedicou muito de seu tempo.

Finalmente gostaria de agradecer ao colega Patrick Suarez Fonseca pela ajuda que prestou em todas as disciplinas do curso, simplesmente pelo fato de poder ajudar. Também pela parceria durante o projeto prático, sem dúvida imprescindível para a realização do projeto.

RESUMO

CHEROBIM, Jordano. **PLATAFORMA PARA CONTROLE DE CONCENTRAÇÃO DE GÁS GLP**. 2018. 46 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A utilização de gases liquefeitos de petróleo (GLP) pelas indústrias implica no risco inerente de explosões ou danos à saúde dos funcionários. Por esse motivo, é de suma importância a realização de aferições frequentes para medir a quantidade de gás presente no ar em pontos estratégicos. Para realizar tais aferições, tipicamente se faz necessário um profissional com equipamento adequado para tal atividade, o que envolve custos e expõe o funcionário à diversos riscos. Neste trabalho, é realizado o desenvolvimento de um protótipo de um sensor GLP capaz de realizar tais aferições, que potencialmente pode diminuir os custos para as empresas e aumentar a segurança de seus funcionários. O protótipo criado comunica-se através de um servidor central utilizando o protocolo IEEE 802.11, que é popularmente conhecido como Wi-Fi. O servidor central armazena as informações obtidas do sensor, e fornece uma visualização amigável para um acompanhamento diário e para uma eventual tomada de decisão quando o nível do gás ultrapassar um dado limite aceitável. Com este trabalho, é testada a viabilidade da solução proposta, são identificados possíveis problemas e é oferecida uma base sólida para a criação de um produto final.

Palavras-chave: Sensor de GLP. Arduino. Gases.

ABSTRACT

CHEROBIM, Jordano. **GLP GASES SENSOR DEVELOPMENT IN INDUSTRY**. 2018. 46 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The use of liquefied petroleum gases (LPG) by industries involves the inherent risk of explosions or to the health of the employees. Thus, it is essential to frequently perform measurements of the amount of gas present in the air at strategic points. In order to carry out such measurements, it is typically necessary to have a professional with suitable equipment for such activity, which involves costs and exposes the employee to various risks. This work aims to develop a prototype of an LPG sensor capable of performing such measurements, which can potentially reduce costs for companies and increase the safety of their employees. The prototype created communicates through a central server using the IEEE 802.11 protocol, which is popularly known as Wi-Fi. The central server stores the information obtained from the sensor and provides a friendly view for daily monitoring, allowing a possible action when the gas level exceeds an acceptable limit. In this work, the viability of the proposed solution is tested, possible problems are identified, and a solid foundation is offered for the creation of an end product.

Keywords: GLP Sensor. Arduino. Gases.

LISTA DE FIGURAS

FIGURA 1	– Arduino Uno	18
FIGURA 2	– Arduino Mega	19
FIGURA 3	– Arduino Pro Mini	20
FIGURA 4	– ESP8266	21
FIGURA 5	– Diagrama do sensor MQ-2	24
FIGURA 6	– MQTT	25
FIGURA 7	– ThingSpeak com MQTT	27
FIGURA 8	– Diagrama Arduino Mega	30
FIGURA 9	– Teste de conexão Arduino Mega	31
FIGURA 10	– Dados no ThingSpeak	32
FIGURA 11	– Nó com o sensor de GLP	32
FIGURA 12	– Gravação do Arduino Pro Mini com o FTDI232	34
FIGURA 13	– Nó com Arduino Pro Mini	35
FIGURA 14	– ThingSpeak	36
FIGURA 15	– Canais ThingSpeak	36
FIGURA 16	– API Keys ThingSpeak	37
FIGURA 17	– Sensor de Gás 001	39
FIGURA 18	– Sensor de Gás 002	40
FIGURA 19	– Sensor de Gás 003	40
FIGURA 20	– Interface Web	41

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
AT	Comando básico para teste de conexão
CH ₄	Gás Metano
CO	Gás Carbônico
CSS	<i>Cascading Style Sheets</i>
EPI	Equipamento de Proteção Individual
GLP	Gases Liquefeitos de Petróleo
H ₂	Gás Hidrogênio
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
MIT	<i>Massachussets Institute of Tecnology</i>
MVC	<i>Model View Controller</i>
OSI	<i>Open System Interconnection</i>
PWM	<i>Pulse Width Modulation</i>
RFID	Identificadores de rádio frequência
RTOS	<i>Real Time Operating Systems</i>
SGBD	Sistema Gerenciador de Banco de Dados
SRAM	memória estática de acesso aleatório
TCP/IP	Conjunto de protocolos de comunicação entre computadores em rede
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 MOTIVAÇÃO	12
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO DOCUMENTO	14
2 METODOLOGIA	15
2.1 INTERNET DAS COISAS	15
2.2 COMPUTAÇÃO EM NUVEM	16
2.3 ARDUINO	17
2.3.1 Arduino Uno	17
2.3.1.1 Especificações técnicas	17
2.3.1.2 Programação	18
2.3.1.3 Fonte de alimentação	18
2.3.2 Arduino Mega	18
2.3.2.1 Especificações técnicas	18
2.3.2.2 Programação	19
2.3.2.3 Fonte de alimentação	19
2.3.3 Arduino Pro Mini	20
2.3.3.1 Especificações técnicas	20
2.3.3.2 Programação	20
2.3.3.3 Fonte de alimentação	20
2.4 ESP8266	21
2.5 COMANDOS AT	23
2.6 SENSOR MQ-2	23
2.7 MQTT	24
2.8 HTTP	26
2.9 THINGSPEAK	26
2.10 FRAMEWORK LARAVEL	27
3 DESENVOLVIMENTO	29
3.1 EQUIPAMENTOS UTILIZADOS	29
3.2 MONTAGEM DO CIRCUITO DE DESENVOLVIMENTO	29
3.2.1 Codificação Utilizando Arduino	30
3.3 MONTAGEM DOS NÓS	33
3.3.1 Circuito	33
3.3.2 Gravação	34
3.4 BROCKER MQTT ONLINE	35
4 RESULTADOS	38
4.1 AFERIÇÕES OBTIDAS	38
4.1.1 Sensor de Gás 001	38

4.1.2 Sensor de Gás 002	39
4.1.3 Sensor de Gás 003	39
4.2 DISCUSSÃO	40
4.3 INTERFACE WEB	41
5 TRABALHOS FUTUROS	42
5.1 MICROCONTROLADOR	42
5.2 SENSOR DE GLP	42
5.3 SERVIDOR NA NUVEM	43
6 CONCLUSÃO	44
REFERÊNCIAS	45

1 INTRODUÇÃO

O Gás Liquefeito de Petróleo (GLP), popularmente conhecido como gás de cozinha, é amplamente utilizado pela sociedade moderna e está presente em vários setores como: no setor comercial, como em hospitais, restaurantes, hotéis e lavanderias; no setor Siderúrgico para fundição, corte e solda de metais; no setor Petroquímico, como na fabricação de borracha e polímeros; no setor Agropecuário como na secagem de grãos, controle de pragas e aquecimento de ambientes. O GLP é uma das frações mais leves do petróleo e sua queima é considerada muito limpa, com uma emissão de poluentes muito baixa. Devido a esses atributos, pode ser utilizado em cozinhas ou em aplicações industriais sensíveis a poluentes (PETROBRAS, 2013).

Os gases GLP são misturas de moléculas de carbono e hidrogênio que podem passar para o estado líquido (condensáveis) em virtude de uma pressão submetida às moléculas em seu processo de envazamento. Em suas condições normais, o GLP encontra-se em estado gasoso. Sua principal fonte de obtenção é o petróleo e apesar de não possuir cor e nem odor próprio, por motivos de segurança, ainda dentro da refinaria onde é extraído, uma substância com cheiro característico do grupo das mercaptanas é adicionada ao GLP, para que possíveis vazamentos possam ser percebidos pelas pessoas. Pois apesar de não ter risco tóxico, o gás GLP ao ser inalado pode causar uma sensação anestésica, produzindo assim efeitos colaterais indesejados (SILVA, 2018; ABREU, 2015).

Apesar do GLP ser utilizado em vários setores da sociedade, sua principal aplicação em nível mundial é no cozimento de alimentos, sendo que, no Brasil, cerca de 80% da sua utilização é destinada ao uso doméstico. O Brasil representa um dos maiores mercados de GLP, consumindo anualmente valores estimados na ordem de 6 milhões de toneladas.

O engarrafamento é feito em recipientes, normalmente feitos de aço. O tamanho desses recipientes depende da sua aplicação, sendo os bujões de gás de 13kg os mais utilizados para uso residencial. No caso da utilização em prédios e condomínios, suas embalagens podem ser consideravelmente maiores, e quando se trata de indústrias que utilizam o gás como matéria prima, podem ser transportados e armazenados em tanques de até milhares de metros cúbicos

de volume (ABREU, 2015).

Esses gases, apesar de não serem tóxicos, são perigosos devido à suas características físico-químicas em todo seu ciclo de utilização, bem como no seu transporte. Essas substâncias, quando em contato com o ar, formam uma mistura explosiva que pode entrar em ignição com muita facilidade e causar graves acidentes, desde domésticos até em seu uso comercial. Para evitar acidentes, os tanques possuem indicadores de temperatura e pressão, e também são armazenados em áreas bem ventiladas e com um rigoroso controle de possíveis fontes de ignição (ABREU, 2015).

Uma vez que os processos de verificação de níveis de gases GLP contidos nos ambientes industriais são realizados de forma manual, o presente trabalho dedica-se a tornar automático esse processo, visando diminuir o risco para as pessoas, além de aumentar a confiabilidade das informações obtidas e a velocidade de *feedback* em situações de risco.

1.1 MOTIVAÇÃO

As empresas do setor industrial que utilizam gases GLP em alguma parte do processo produtivo possuem grande preocupação com a segurança. Essa preocupação se dá devido à possibilidade de vazamentos nas tubulações que levam os gases, desde as chamadas “fazendas de gases”, que são lugares ao ar livre que servem como depósito de gases, até as linhas de produção, onde os produtos serão fabricados.

Levando em consideração essa preocupação, as empresas se utilizam de ferramentas para aferir a quantidade de gás contido no ambiente. Como vivenciado pela equipe em sua experiência, em algumas empresas o processo de aferição de gases é realizado de forma manual com equipamentos específicos onde um funcionário visita pontos estratégicos da empresa por onde passam tubulações de gases para realizar essa aferição. Caso encontre uma quantidade de gás acima do valor aceitável, imediatamente este entra em contato com o setor de manutenção para que possam tomar as devidas precauções com o intuito de evitar eventuais acidentes.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

1.2.1 OBJETIVO GERAL

Desenvolver um equipamento a partir de módulos microcontroladores que forneça maior confiabilidade e rapidez, e menor custo na transmissão dos dados sobre a quantidade de gases GLP contidos nos diversos setores da empresa.

1.2.2 OBJETIVOS ESPECÍFICOS

- Criar um protótipo para aferição de gases GLP e a disponibilização das informações em tempo real através de uma interface gráfica;
- Aumentar a confiabilidade na passagem das informações obtidas através da utilização dos sensores para aferições de gases GLP, evitando o envio de informações errôneas que potencialmente ocorre em processos manuais;
- Obter mais agilidade e rapidez ao informar possíveis níveis de quantidade de gases acima do aceitável para o setor de manutenção;
- Possibilitar a análise dos dados obtidos em tempo real através de uma aplicação web;
- Reduzir custos referentes ao tempo gasto com um funcionário designado para realizar o processo de aferição de gases.

1.3 JUSTIFICATIVA

Visto a importância do processo de aferição da quantidade de gases GLP presentes dentro de uma indústria, é possível visualizar vários problemas durante a execução do processo: Considerando que é necessário ao menos uma pessoa para realizar o processo de aferição manualmente, essa pessoa mesmo utilizando Equipamentos de Proteção Individual (EPI)'s adequados para a atividade, se expõe a uma atividade com um alto grau de risco. Os dados obtidos através da aferição são suscetíveis a erros, tanto no manuseio do equipamento de aferição, quanto na transmissão dessa informação até o sistema. As informações dessas aferições são obtidas poucas vezes ao dia, pois esse processo pode levar um tempo considerável para ser concluído, fazendo com que a empresa não tenha a informação da quantidade de gases em tempo real, o que leva a um tempo de resposta muito grande para uma eventual tomada de decisão.

1.4 ESTRUTURA DO DOCUMENTO

No Capítulo 2 serão mostrados aspectos teóricos e conceitos trazidos de livros, artigos e manuais técnicos utilizados, iniciando desde o conceito de Internet das Coisas, até as ferramentas e *softwares* utilizados para realizar a comunicação dos sensores com uma interface web. No Capítulo 3 serão tratados os equipamentos de *hardware* utilizados, assim como a montagem dos seus circuitos especificando a nomenclatura dos pinos. Posteriormente, será apresentada a codificação e a calibragem do sensor, juntamente com a montagem dos circuitos, gravação dos nós e o envio das informações através de um broker MQTT online. No Capítulo 4 são indicados os dados resultantes do trabalho, as dificuldades encontradas com os equipamentos utilizados e também uma discussão sobre as aferições obtidas e como esses dados são mostrados ao usuário. No Capítulo 5 são apresentadas alterações a serem realizadas futuramente visando o aperfeiçoamento do projeto, transformando o protótipo em um produto que possa ser usado no âmbito comercial e industrial, com maior desempenho e confiabilidade no envio das informações. Por fim, o Capítulo 6 apresenta as conclusões deste trabalho.

2 METODOLOGIA

Neste capítulo serão apresentados conceitos e documentações utilizadas durante o estudo para auxiliar na compreensão das ferramentas, componentes e ambientes aplicados ao projeto.

2.1 INTERNET DAS COISAS

O termo Internet das Coisas (IoT, do inglês *Internet of Things*), apesar de ser um tema atual, não é novo para a literatura relacionada. Este termo foi criado pelo pesquisador Kevin Ashton, do Instituto de Tecnologia de Massachusetts (MIT, do inglês *Massachusetts Institute of Technology*) em 1999 durante uma apresentação para executivos da Procter & Gamble quando falava em identificar produtos de uma empresa de forma eletrônica e assim facilitar o processo logístico da empresa por meio de identificadores de rádio frequência (RFID, do inglês *Radio-Frequency IDentification*). Ashton foi co-fundador do Auto-ID Center e trabalhou na criação de padrões para sensores e acreditava que com a IoT, tudo estaria conectado entre si em uma grande rede de comunicação (CARVALHO et al., 2016).

De forma mais simplificada, esse conceito pode ser entendido como um conjunto de tecnologias incluindo *hardwares* e *softwares* que possibilitem a interligação de equipamentos de forma que a troca de informações e ações ocorra através da internet. Entretanto segundo Oliveira (2017), não basta ligar equipamentos pela internet, mas também tornar esses equipamentos inteligentes. E isso pode ser realizado através da programação desses componentes para que, na ocorrência de determinado fato, o dispositivo atue sem a necessidade de intervenção humana.

O mercado de IoT tende a crescer muito nos próximos anos com o objetivo de simplificar e facilitar os processos diários das pessoas, sejam eles em casa, na rua ou na empresa. A partir dessa simplificação, os equipamentos começarão a gerar mais dados e poderão ser utilizados em outros nichos de mercados, o que criará novas possibilidades de negócios.

2.2 COMPUTAÇÃO EM NUVEM

De forma básica, a computação em nuvem nada mais é que uma grande utilização da virtualização para construir um modelo de negócio. A computação como um todo passou por inúmeras transformações e ciclos que variavam entre a centralização e descentralização dos sistemas. No início da informática, em meados dos anos 60, eram comuns os chamados mainframes, que nada mais eram que computadores de grande porte, na época utilizado para processar e armazenar grandes volumes de informações. O acesso dessas informações era feito através de terminais compostos por telas e periféricos de entrada, normalmente um teclado. Mais tarde surgiram os computadores pessoais, o que permitiu a descentralização do poder de processamento, onde cada usuário acessava seu próprio computador, sem ser mais necessário um mainframe (CARISSIMI, 2015).

Ainda mais tarde surgiu a ideia de conectar os computadores em uma grande rede em que esses dispositivos compartilhariam informações e recursos através de servidores. Exemplos disso são servidores de impressão e de arquivos. A partir desse modelo uma nova ideia surgiu de ofertar serviços de forma remota seguindo o modelo de cliente-servidor, onde no servidor é contido um recurso em específico, e vários clientes o acessam através da rede para realizar a solicitação do serviço ofertado (CARISSIMI, 2015).

A computação em nuvem acontece quando utiliza-se recursos como armazenamento, processamento ou acesso a partir de servidores externos (OLIVEIRA, 2017).

Alguns pontos que tornam a computação em nuvem uma vantagem são: Redução de custos com compra e manutenção de equipamentos de hardware e software. Menor complexidade de configuração, pois na maioria das vezes as ferramentas em nuvem possuem uma configuração predefinida com uma interface amigável ao usuário. Menor quantidade de pessoas necessárias para realizar manutenção da infraestrutura interna (AZURE, 2018).

Entretanto existem desvantagens como: Dependência de uma velocidade de internet razoável para que as aplicações não possuam uma latência acima do máximo aceitável para uma dada aplicação. Falta de segurança na transmissão de arquivos, pois apesar de geralmente os acessos serem criptografados, pelo simples fato de estarem diretamente ligados a internet, isso já basta para que sejam vulneráveis a ataques cibernéticos.

Analisando os pontos positivos e negativos com relação a computação em nuvem é possível notar que a viabilidade da utilização da computação em nuvem varia conforme a necessidade da aplicação.

2.3 ARDUINO

O Arduino é uma plataforma de desenvolvimento de sistemas embarcados aberta e de baixo custo. Nessa plataforma podem ser adicionados vários componentes eletrônicos desenvolvidos para executar atividades específicas, e assim incrementar novas funções à plataforma. A sua popularização é devido a facilidade de desenvolvimento dentro da plataforma, assim como inúmeros módulos prontos para utilização e com baixo custo (OLIVEIRA, 2017).

Como o Arduino não está atrelado à nenhum fabricante, ele é tão popular e também por isso se tornou uma plataforma de sistemas embarcados acessível a estudantes, entusiastas e até curiosos. Com a popularização do Arduino, começaram a ser produzidos e utilizados em larga escala, módulos chamados também de shields, e por este motivo tornou-se muito acessível em questão de preço.

Dentre seus módulos é possível encontrar uma infinidade de sensores, desde a medição de temperatura ambiente, até os mais complexos sensores de gases inflamáveis, entre outros. Também atuadores como os módulos de relé e acionadores que funcionam em lâmpadas e motores. Com relação à comunicação é possível encontrar shields de conexão Wifi, Ethernet, Bluetooth e rádio frequência.

Entre os módulos mais conhecidos estão o Arduino Uno, Mega e Pro Mini, os dois primeiros focados no desenvolvimento e o Arduino Pro Mini utilizado para produção de sistemas embarcados.

2.3.1 ARDUINO UNO

2.3.1.1 ESPECIFICAÇÕES TÉCNICAS

O Arduino Uno mostrado na Figura 1 é baseado no microcontrolador ATmega328P e possui 14 entradas/saídas digitais das quais 6 são utilizadas como saídas de Modulação de Largura de Pulso (PWM, do inglês *Pulse Width Modulation*) e 6 entradas analógicas a uma frequência de 16 MHz. O ATmega328 possui 32KB de memória interna, sendo 0.5KB sendo ocupado pelo *bootloader* e 2KB de memória estática de acesso aleatório (SRAM, do inglês *Static Random Access Memory*). Essa placa também se diferencia de outras por não precisar utilizar o conversor FTDI de porta USB para serial.

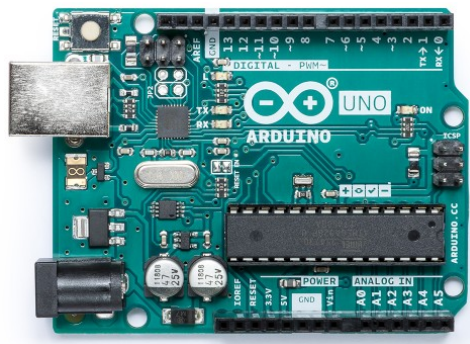
2.3.1.2 PROGRAMAÇÃO

Para realizar a programação dentro do Arduino Uno basta que seja conectado via um cabo USB ao computador e com a utilização da própria IDE do Arduino. Para utilizá-lo basta acessar o menu "Ferramentas / Placa / Arduino Uno". O Arduino Uno já é pré-programado com um gerenciador de inicialização que permite fazer upload de novos códigos sem necessitar de um hardware externo.

2.3.1.3 FONTE DE ALIMENTAÇÃO

Para sua alimentação elétrica pode ser utilizada a mesma porta de transferência de dados de barramento serial universal (USB, do inglês *Universal Serial Bus*) e também uma entrada de energia própria, sendo a fonte selecionada automaticamente conforme é conectada. A segunda opção pode ser ligada através de um adaptador AC/DC ou uma bateria. A placa pode operar externamente com uma fonte de 6 à 20 volts, sendo que o intervalo de energia ideal é entre 7 e 12 volts (ARDUINO, 2018).

Figura 1: Arduino Uno



Fonte: Arduino (2018).

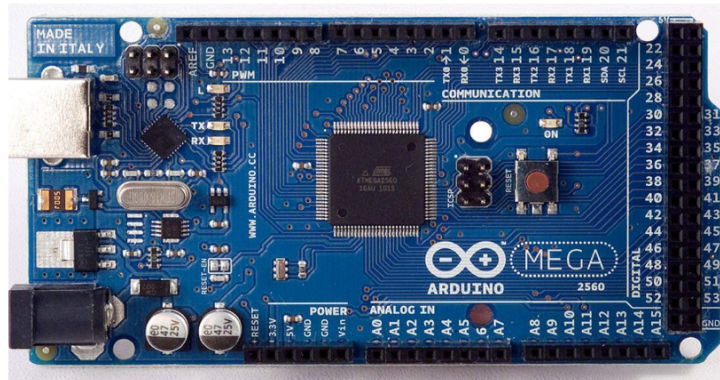
2.3.2 ARDUINO MEGA

2.3.2.1 ESPECIFICAÇÕES TÉCNICAS

O Arduino Mega indicado na Figura 2 é baseado no microcontrolador ATmega2560 e possui 54 entradas/saídas digitais sendo 15 delas utilizadas como saídas PWM e 16 entradas analógicas. O que o diferencia de outros modelos mais básicos de Arduino é que o Arduino Mega possui 4 portas seriais, o que fornece uma solução muito mais completa para projetos complexos. Possui o armazenamento de 256 KB, sendo 8KB utilizados para a *bootloader*. Sua

memória SRAM é de 8 KB e assim como o Arduino Uno, essa placa também não precisa utilizar um conversor FTDI externo (ARDUINO, 2018).

Figura 2: Arduino Mega



Fonte: Arduino (2018).

2.3.2.2 PROGRAMAÇÃO

Na questão de programação e envio do código, o Arduino Mega tem o mesmo funcionamento que o Arduino Uno, basta que seja conectado via um cabo USB ao computador e com a utilização da própria IDE do Arduino. Para utilizá-lo basta acessar o menu "Ferramentas / Placa / Arduino Mega". O Arduino Mega já é pré-programado com um gerenciador de inicialização que permite fazer upload de novos códigos sem necessitar de um hardware externo.

2.3.2.3 FONTE DE ALIMENTAÇÃO

Assim como o Arduino Uno, o Arduino Mega pode ter sua alimentação de energia via porta USB ou uma fonte externa através de um adaptador AC/DC ou bateria, determinada de forma automática a partir de sua ligação. A placa suporta uma alimentação externa de 6 à 20 volts, sendo que menos de 7 volts geraria uma instabilidade no funcionamento da placa. A alimentação ideal fica na faixa de 7 à 12 volts. Os pinos de energia contidos na placa são Vin de entrada de energia, 5V de saída regulada, de 3.3V também de saída regulada porém de menor energia e GND que é a ligação neutra da placa.

2.3.3 ARDUINO PRO MINI

2.3.3.1 ESPECIFICAÇÕES TÉCNICAS

O Arduino Pro Mini, que aparece na Figura 3, por sua vez possui um tamanho consideravelmente pequeno, porém baseado no mesmo microcontrolador ATmega2560 que o Arduino Mega, possuindo 54 entradas/saídas digitais, 15 saídas PWM e 16 entradas analógicas. Além de também possuir 4 portas seriais mas devido ao seu tamanho pequeno, sua fonte de energia não é provida através de uma porta USB e sim através de sua pinagem. Por possuir um tamanho reduzido sua memória de armazenamento é de 32 KB, sendo 2 KB usados como *bootloader* e 2 KB de memória SRAM (ARDUINO, 2018).

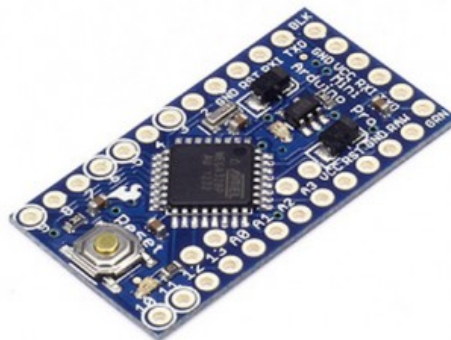
2.3.3.2 PROGRAMAÇÃO

O Arduino Pro Mini assim como os Arduinos anteriores, pode ser programado utilizando a IDE do Arduino, porém ao contrário dos anteriores, o mesmo não possui uma conexão direta com o computador através de uma porta USB, necessitando então de um conversor de porta serial para USB.

2.3.3.3 FONTE DE ALIMENTAÇÃO

A alimentação de energia para o Arduino Pro Mini pode ser realizada através de um cabo/placa FTDI ou um regulador de tensão 3.3V ou 5V no pino VCC.

Figura 3: Arduino Pro Mini

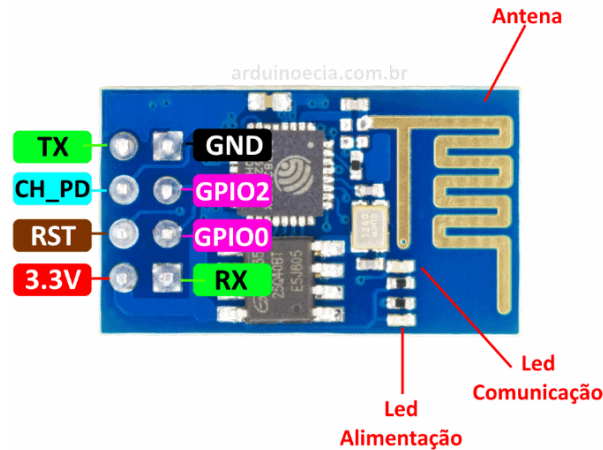


Fonte: Arduino (2018).

2.4 ESP8266

O principal conceito atrás da internet das coisas é a conectividade dos nós e recursos. Essa conectividade em rede, pode ou não ser realizada com a internet dependendo do modelo e a necessidade de cada aplicação.

Figura 4: ESP8266 ESP-01



Fonte: Cia (2015).

O ESP8266 é um microcontrolador da fabricante chinesa Espressif Systems. O ESP8266 se apresenta como solução para quem busca um microcontrolador com WiFi embutido que tenha um tamanho pequeno e que possa ser utilizado de forma compatível com uma grande variedade de microcontroladores. Ele utiliza o protocolo Wifi 802.11 para enviar e receber dados e possui dimensões pequenas, o que é ideal para a utilização dentro de aplicações de Internet das Coisas (CURVELLO, 2015).

O consumo de energia desse microcontrolador é de 20uA com o modo *sleep* ativo que seria o consumo mínimo do equipamento. Com o microcontrolador ligado a um AP WiFi esse consumo chega a 50mA. Outras especificações importantes desse dispositivo podem ser vistas na Tabela 2.

Esse microcontrolador possui como vantagens para sua utilização, fatores como preço: Pode ser encontrado em sites chineses por exemplo, por cerca de 3 dólares, o que torna um dos principais microcontroladores utilizados quando se trata de Internet das Coisas. Outra vantagem importante é que possui três ambientes de desenvolvimento disponíveis, sendo um baseado na linguagem brasileira Lua e que possui integração de maneira simplória; Ambiente do Arduino, que inclui todas as bibliotecas e interfaces necessárias, incluindo também aplicações de exemplo já construídas dentro do Arduino; Ambiente baseado em sistema operacional de tempo real (RTOS, do inglês *Real Time Operating Systems*), permitindo assim a criação de aplicações

de forma profissional e podendo explorar todo seu potencial em aplicações de tempo real. O último fator importante a ser citado sobre o microcontrolador é a grande disponibilidade de módulos periféricos, onde todos os módulos popularizados para o Arduino são compatíveis com o ESP8266, permitindo assim que a integração entre os mesmos seja realizada sem grande esforço.

Existem no mercado vários módulos que se utilizam do microcontrolador ESP8266 e que podem ser encontrados em lojas internacionais por preços muito baixos e também em sites nacionais, com um valor não tão baixo comparando com sites chineses, porém em comparação com outros microcontroladores podem ser considerados baratos. O módulo mais simples que pode ser encontrado é o ESP-01. Este possui apenas o módulo de memória externa entre 512kB e 1MB, sendo mais que suficiente para a maioria dos casos reais. Possui apenas 8 pinos incluindo os pinos de alimentação e de comunicação serial. Para realizar a gravação no mesmo é necessário um adaptador USB. Mais informações sobre os pinos que compõe o ESP-01 podem ser encontradas na Tabela 1.

Tabela 1: Pinos disponíveis no módulo ESP-01

Pino	Característica
TX	Sinal TX do módulo, ligado com o sinal RX do microcontrolador
GND	Saída para o sinal Terra
CH_PD	Pino usado para gravação ou atualização do firmware
RST	Pino de reset
3.3V	Pino de entrada de alimentação de energia
RX	Sinal RX do módulo, ligado com o TX do microcontrolador
GPIO	2 pinos de I/O (Entrada/Saída)

Fonte: Oliveira (2017).

Tabela 2: Especificações do microcontrolador Wifi ESP8266

Característica	Valor
Voltage	3.3V
Current consumption	10uA – 170mA
Flash memory attachable	16MB max (512K normal)
Processor	Tensilica L106 32 bit
Processor speed	80-160MHz
RAM	32K + 80K
GPIOs	17 (multiplexed with other functions)
Analog to Digital	1 input with 1024 step (10 bit) resolution
802.11 support	b/g/n/d/e/i/k/r
Maximum concurrent TCP connections	5

Fonte: Kolban (2016).

O equipamento utiliza apenas 20% de seu processamento total dedicado à comunicação

Wifi, sendo os outros 80% dedicados a aplicação do usuário. O espaço livre de memória para o programa principal é de 4MB (CURVELLO, 2015).

2.5 COMANDOS AT

Os comandos AT são instruções executadas para acionar o módulo ESP8266 quando conectado a um outro equipamento, como o Arduino. Esses comandos são baseados nos comandos que compõe o "Padrão Hayes" criado em 1981 que era utilizado para execução de funções relacionando computadores com linhas telefônicas, permitindo a configuração de modems. Apesar de terem sido criados em 1981, os comandos estão presentes em vários equipamentos periféricos, inclusive sendo usados para a configuração de firmwares (CAMPOS, 2018).

Os comandos que serão demonstrados no presente trabalho aparecem citados na Tabela 3.

Tabela 3: Comandos AT utilizados para realizar a comunicação wifi entre Arduino e broker

Comando	Definição
AT+RST	Reiniciar módulo
AT+CWMODE	Ler/configurar modo de operação.
AT+CWJAP	Acessar o Access Point
AT+CIPMUX	Ativa/desativa múltiplas conexões sendo 0 para desativado
AT+CIPSTART	Estabelece uma conexão TCP ou registra uma porta UDP
AT+CIPSEND	Especifica o tamanho da string que será enviada
GET	Envio de dados

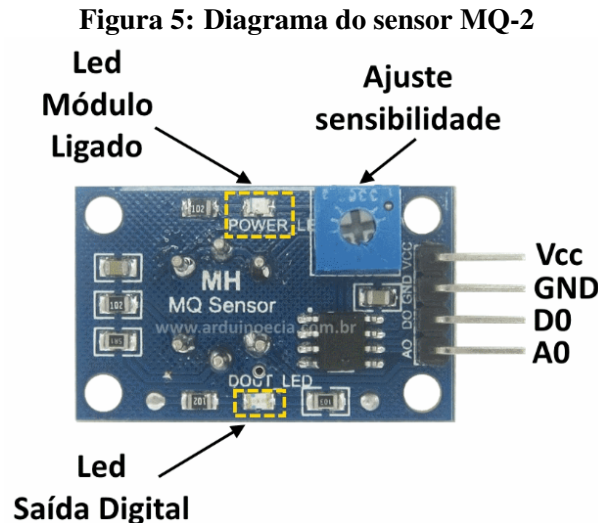
Fonte: Team (2015).

2.6 SENSOR MQ-2

Segundo Oliveira (2017), um sensor que deveria estar presente em todas as moradias é o sensor de gases inflamáveis, pois o mesmo pode significar a diferença entre a vida e a morte, pois pode detectar facilmente o vazamento de gases conhecidos como "gases de cozinha" e assim evitar riscos à segurança dos moradores.

O MQ-2 é um sensor de gases usado tanto em aplicações industriais como residenciais. Com ele é possível detectar gases como H₂ (Gás Hidrogênio), GLP, CH₄ (Gás Metano), CO (Gás Carbônico), Álcool, Fumaça e Propano. Sua sensibilidade pode ser ajustada dependendo da necessidade de cada aplicação (JIANKAILI, 2015).

Esse sensor possui a vantagem de ser muito sensível a presença de gases GLP, Propano e Hidrogênio, assim como garantir uma grande longevidade, ser um módulo de baixo custo e de fácil implementação. Pode ser aplicado em residências, indústrias ou comércios.



Fonte: Jiankai.li (2015).

2.7 MQTT

Segundo Yuan (2017) o MQTT é um protocolo de comunicação criado inicialmente para vincular sensores em *pipelines* de petróleo a satélites. Foi desenvolvido para ser extremamente leve e transportar mensagens de *publish/subscribe*. Pode-se dizer que o MQTT é um excelente protocolo para a viabilização da Internet das Coisas devido a sua leveza que permite sua implementação na parte física dos dispositivos, funcionando muito bem com larguras de bandas limitadas e de alta latência.

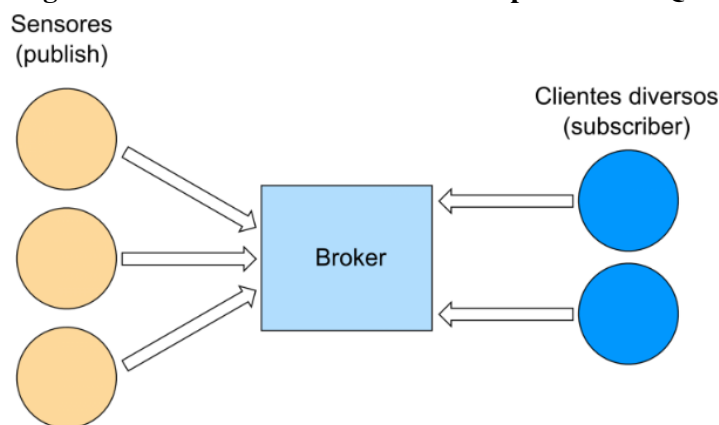
Alguns dos motivos para não se trabalhar com o Protocolo de Transferência de Hipertexto (HTTP, do inglês *Hypertext Transfer Protocol*) que é um dos protocolos de Web mais conhecidos e discriminados, é que o HTTP é um protocolo que trabalha de forma síncrona. Ou seja, ao cliente enviar uma requisição para o servidor, o mesmo espera o retorno da informação. Para servidores web isso não é um problema, mas quando se fala em Internet das Coisas é necessário pensar em mínimo consumo de banda de internet e de energia, pois um dos princípios desse conceito é o funcionamento e autonomia dos nós pelo maior tempo possível. Por isso a aplicação de um protocolo síncrono não faria sentido. O HTTP também é um protocolo grande, com muitos cabeçalhos e regras, o que não se encaixaria para redes que visem a comunicação mínima de poucos bytes.

O MQTT possui três tipos de elementos, sendo o primeiro deles um broker que trabalha como um servidor que sempre deve estar disponível para fazer a interligação desde a geração dos dados, até a sua publicação. O segundo elemento é o Publisher MQTT que seria um dispositivo de Internet das Coisas ou "nó" que irá realizar a geração dos dados e os enviará até o broker, para que ele realize a entrega a seus determinados destinos. E finalmente o Subscriber MQTT que nada mais é que o "assinante" de determinado serviço disponibilizado pelo Publisher Broker através do broker. A partir das informações obtidas a partir desse serviço são mostradas via interface gráfica (OLIVEIRA, 2017).

De forma mais simplista, essa estrutura de *Publish/Subscribe* do MQTT tem por definição duas entidades na rede que são: *broker* e clientes. O *broker* é um *software* servidor que recebe mensagens de todos os nós clientes e faz o roteamento das mensagens entre eles. Os clientes são quaisquer dispositivos que possam se comunicar com o *broker*. (YUAN, 2017) Pode-se dizer que o broker tem atuação da mesma maneira que um servidor *web* ou Sistema Gerenciador de Banco de Dados (SGBD), porém com um modelo bem simplificado pois trabalham em sua maioria com dados simples (OLIVEIRA, 2017).

Nesse padrão o cliente se inscreve no tópico que deseja receber informações como pode ser visto na Figura 6 e somente receberá enquanto estiver inscrito. Após inscrito no tópico, o Broker MQTT tem a informação de que aquele cliente em específico deseja receber a informação do tópico pré-determinado. Com isso, sempre que o tópico possuir alguma atualização de dados, enviará essa informação ao seu cliente.

Figura 6: Modelo de funcionamento do protocolo MQTT



Fonte: Embarcados (2018).

2.8 HTTP

O HTTP é um protocolo da camada de aplicação presente tanto no modelo Interconexão de sistemas abertos (OSI, do inglês *Open System Interconnection*) quanto no modelo TCP/IP e define vários métodos para a requisição de dados. O HTTP é um protocolo de transmissão de hipermídia como Linguagem de Marcação de Hipertexto (HTML, em inglês Hypertext Markup Language). Ele foi criado para realizar a comunicação entre navegadores e servidores, mas também pode ser utilizado para outros propósitos. Funciona utilizando o modelo clássico de cliente-servidor, assim o cliente envia uma requisição e aguarda, enquanto o servidor recebe a requisição, a processa e envia uma mensagem de retorno para a comunicação ser completada. O protocolo HTTP define métodos para a requisição de dados, que indicam para o servidor, qual a ação a ser realizada no momento da chegada da mensagem.

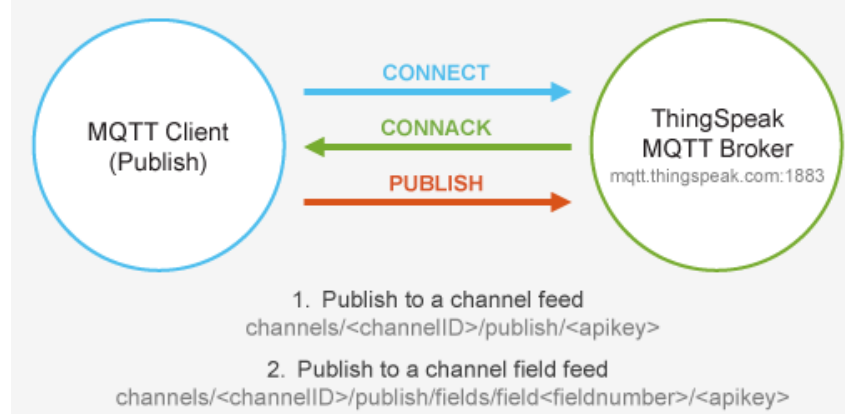
Essas requisições podem acontecer a partir de vários métodos como o *GET* que envia informações de requisição dentro da própria Localizadora Padrão de Recursos (URL, em inglês *Uniform Resource Locator*). O *POST* que tem a mesma função que o *GET*, porém as informações não são enviadas de forma explícita na URL. O *PUT* utilizado para atualizar um recurso em específico e *DELETE* para remover um recurso específico (GARCIA, 2018).

2.9 THINGSPEAK

O ThingSpeak é um plataforma para Internet das Coisas que permite a coleta e armazenamento de dados providos pelos sensores em nuvem. Ele também contempla ferramentas de análise e visualização dos dados em tempo real (MATHWORKS, 2018). Além disso o ThingSpeak permite a análise dos dados por softwares matemáticos, por gráficos e também permitindo a criação de visualizações personalizadas através da programação HTML, JavaScript e Folha de Estilo em Cascatas (CSS, em inglês *Cascading Style Sheets*). Dentre as ações já prontas para a utilização dentro da plataforma, estão o envio de *tweeters* a partir de condições definidas, controles temporizados, reações à eventos específicos e comandos via protocolo HTTP.

A plataforma fornece a Interface de Programação de Aplicativos (API, em inglês *Application Programming Interface*) REST e MQTT para a troca de informações através da criação de canais, sejam para o envio ou recebimento de informações e é compatível com *hardwares* como Arduino, Raspberry, entre outros. É possível visualizar seu funcionando com o protocolo MQTT utilizando a plataforma ThingSpeak, na Figura 7 (MATHWORKS, 2018).

Figura 7: Modo de funcionamento do ThingSpeak com protocolo MQTT



Fonte: ThingSpeak (2018).

Na figura 7 é possível visualizar o cliente do lado esquerdo realizando a comunicação com o servidor Broker MQTT do ThingSpeak. Nessa comunicação, o cliente se conecta ao broker, recebe o retorno da conexão realizada com sucesso e então o item 1 representa o momento em que o cliente informa ao broker que está se cadastrando para enviar uma informação. No item 2, o cliente envia a informação propriamente dita para o servidor broker MQTT.

2.10 FRAMEWORK LARAVEL

Frameworks são bibliotecas de códigos padronizados, que tendem a entregar inúmeras funcionalidades com a escrita do mínimo de código possível. Ou seja, tarefas do dia a dia que o programador precisaria reescrever sempre se preocupando com segurança, reuso de código, melhor maneira de codificar, estão todas prontas para uso. Segundo Willians (2015) os *frameworks* possuem o objetivo de agilizar nosso trabalho, evitando a reutilização de código e trazendo funcionalidades prontas. Além de atuar na redução de custos por ser amplamente difundido, o que significa que em suas versões estáveis, já foram testados exaustivamente por vários desenvolvedores, reduzindo assim o número de testes realizados em suas funcionalidades.

O Laravel é um exemplo de *framework* baseado na linguagem de programação PHP, sendo assim, utilizado para desenvolvimento *web*. Ele utiliza a arquitetura de *software* Modelo Visão e Controle (MVC, em inglês *Model View Controller*) que separa o desenvolvimento de *software* em camadas, cada uma com um tratando uma parte do projeto. A camada *Model* é responsável pela interação direta com o banco de dados, recebendo e enviando informações.

A camada *View* é a camada onde são apresentadas de forma gráfica, as informações obtidas pela camada *Model*. Já a camada *Controller* que é a camada lógica, é responsável por fazer o intermédio entre a camada *Model* e a camada *View*. Além disso, o Laravel é composto por ferramentas prontas para a conexão com os principais bancos de dados do mercado como MySQL, SQL Server, Oracle e Postgress (RAMOS, 2015). Possui um sistema de autenticação por e-mail ou até mesmo com redes sociais de simples implementação, assim como uma estrutura extremamente organizada, o que facilita muito o trabalho e navegação em sua estrutura durante o dia a dia. Essas facilidades proporcionam uma grande praticidade e robustez na criação de aplicações.

3 DESENVOLVIMENTO

Para o desenvolvimento do projeto foram utilizados diversos recursos de *hardware* e *software* que serão apresentados nas seguintes seções deste capítulo, bem como a forma em que o projeto foi desenvolvido.

3.1 EQUIPAMENTOS UTILIZADOS

Para o desenvolvimento do projeto, foram necessários os seguintes equipamentos de hardware:

- Sensor de gás MQ-2;
- Arduino Mega 2560;
- Arduino Pro Mini;
- ESP8266 Modelo ESP.01;
- Módulo FTDI 232;
- Fonte ajustável de 3.3V até 5V;
- Resistores de 10 ons;
- Jumpers;

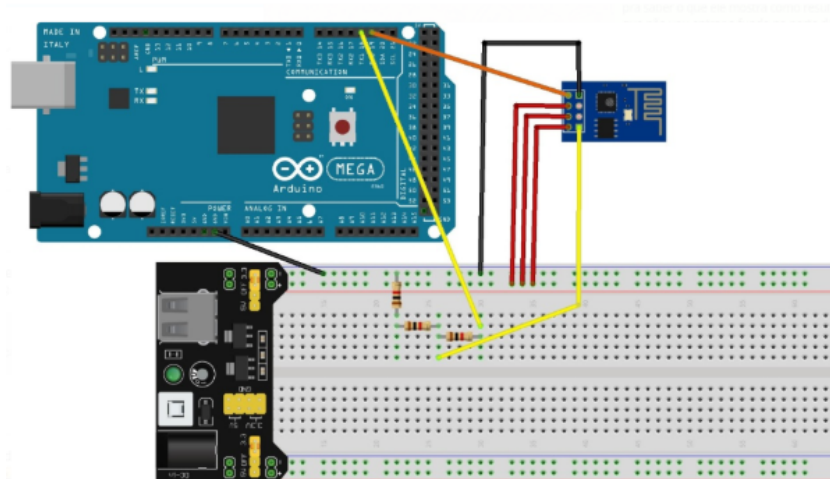
3.2 MONTAGEM DO CIRCUITO DE DESENVOLVIMENTO

O primeiro protótipo foi criado com o Arduino Mega que possibilita uma maior facilidade no processo de desenvolvimento. Isso se dá devido à gravação do código desenvolvido em um computador para o Arduino Mega através de um cabo USB simples.

Foram utilizados nesse primeiro protótipo um Arduino Mega, um módulo de wifi ESP8266, um shild de fonte ajustável entre 3.3V até 5V para utilização na protoboard e 3 resistores de 10 ons cada. A ligação dos pinos do ESP8266 foram realizadas da seguinte forma: O pino GND ligado diretamente no pino GND da fonte ajustável, o pino TX foi ligado no pino Rx1 do Arduino, os pinos CH_PD, RST e 3.3V foram ligados diretamente no pino VCC da fonte ajustável e por fim o pino RX foi ligado no divisor de tensão que foi formado pelos 3 resistores em série de 10 ons que por sua vez foram ligados na outra ponta no pino Tx1 do Arduino. O pino que ainda não foi citado nesse diagrama é o GND do Arduino que foi ligado no pino GND da fonte ajustável. A nomenclatura dos pinos pode ser analisada em seus respectivos diagramas nas Figuras 2 e 4.

Foi realizada a montagem do protótipo inicialmente apenas do Arduino Mega juntamente com o ESP8266 para realizar o teste de conexão Wifi do Arduino e ESP8366 com o servidor do ThingSpeak, como mostrado no diagrama e foto real na Figura 8.

Figura 8: Diagrama Arduino Mega com ESP8266



Fonte: Arduino (2018).

3.2.1 CODIFICAÇÃO UTILIZANDO ARDUINO

Com o circuito montado, foi executado o código apresentado na Figura 9 dentro da IDE (*Integrated Development Environment*) de desenvolvimento do Arduino para acessar o monitor serial, executar os comandos AT (Comando básico para teste de conexão) e realizar o teste de conexão.

Acessando a tela para o teste de conexão, foram informados os dados da rede Wifi, os dados do servidor do ThingSpeak e o comando *GET* através do qual serão enviados os dados

Figura 9: Código executado para acessar monitor serial e executar comandos AT

The screenshot shows the Arduino IDE interface. The top window displays the sketch code for 'teste_serial_arduino_mega'. The code is as follows:

```

void setup() {
  Serial1.begin(115200);
  Serial.begin(115200);
}

void loop() {
  if(Serial.available())
    Serial1.write(Serial.read());
  if(Serial1.available())
    Serial.write(Serial1.read());
}

```

The bottom window shows the serial monitor output, which includes the text 'Carregado.' and memory usage information: 'O sketch usa 2056 bytes (0%) de espaço de armazenamento para prog...' and 'Variáveis globais usam 341 bytes (4%) de memória dinâmica, deixand...'. The status bar at the bottom indicates 'Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) em COM5'.

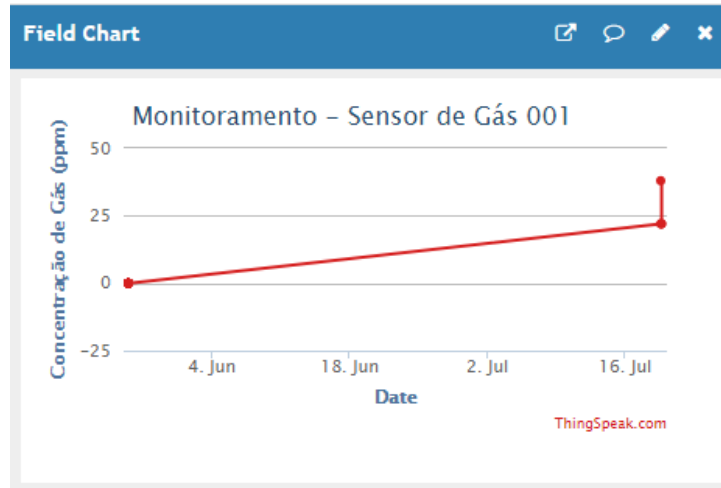
Fonte: Arduino (2018).

para o servidor. Mais detalhes sobre como obter os dados para realizar a conexão estão na seção ThingSpeak(Projeto).

1. AT+RST
2. AT+CWMODE=3
3. AT+CWJAP=REDEWIFI,SenhaDaRede
4. AT+CIPMUX=0
5. AT+CIPSTART="TCP","184.106.153.149",80
6. AT+CIPSEND=75
7. GET https://api.thingspeak.com/update?api_key=P2SEXL8UPEPD2URM&field1=22

Após a execução dos comandos, é obtido o retorno de que a conexão do nó com o servidor foi realizada com sucesso. A partir da conexão realizada, é possível visualizar na Figura 10 o envio do dado para o ThingSpeak em seu próprio *dashboard*.

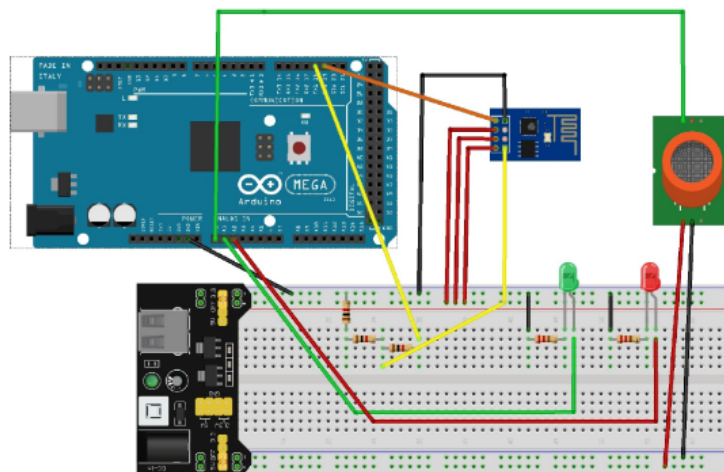
Figura 10: Visualizando o dado recebido no ThingSpeak



Fonte: ThingSpeak (2018).

Tendo a estrutura de conexão completa e funcionando corretamente, foi realizada a ligação do sensor MQ-2 que irá realizar as aferições de gases, assim realizando a montagem completa do nó apresentado na figura 11.

Figura 11: Nó com Sensor de GLP



Fonte: Petrobras (2013).

Neste novo diagrama foram adicionados um led verde, um led vermelho, 2 resistores e o Sensor MQ-2. Os pinos positivos dos leds verde e vermelho foram ligados no Arduino nos pinos A1 e A2 respectivamente. Os pinos negativos dos leds foram ligados aos resistores e

posteriormente ao pino GND da fonte ajustável. O pino VCC do sensor MQ-2 ligado na saída de 5V e o pino GND ligado no pino GND da fonte ajustável, e o pino A0 do sensor ligado no pino A0 do Arduino. Os detalhes das nomenclaturas dos pinos no Sensor MQ-2 podem ser vistos na Figura 5.

Nesse novo contexto onde foi adicionado o sensor MQ-2, enquanto o nível de gás GLP presente no ambiente estiver dentro do nível aceitável, o led verde permanecerá aceso. No caso do nível de gás presente no ambiente aumentar de forma a ultrapassar o nível aceitável, o led verde se apagará e o led vermelho se acenderá.

É necessário frisar que para o sensor MQ-2 funcionar de maneira otimizada, é necessário realizar um processo de calibragem no mesmo. Para isso utilizou-se o método encontrado no datasheet do sensor MQ-2, que define uma variável RS como a resistência do ar e uma variável $R0$ que define a resistência inicial do sensor, que servirá como valor de referência. Ao ligar o sensor, o mesmo foi programado através da IDE do Arduino para retornar a leitura média entre 100 aferições. Como o sensor só pode detectar valores entre 0 e 1023, essa média é dividida por 1024 e como o sensor possui uma saída analógica que varia entre 0V e 5V considerando a faixa do nível de GLP 300ppm a 10.000ppm, esse resultado é multiplicado pelo valor 5.0, obtendo assim o resultado de $R0$, que será o valor utilizado como referência para as posteriores aferições nesse nó.

Para encontrar o valor de RS e após isso obter o valor resultante final de aferição são executadas as seguintes equações:

$$sensorVolt = \frac{sensorValue}{1024 \cdot 5.0} \quad (1)$$

$$RS = \frac{5.0 - sensorVolt}{sensorVolt} \quad (2)$$

$$resultValue = \frac{RS}{R0} \quad (3)$$

3.3 MONTAGEM DOS NÓS

3.3.1 CIRCUITO

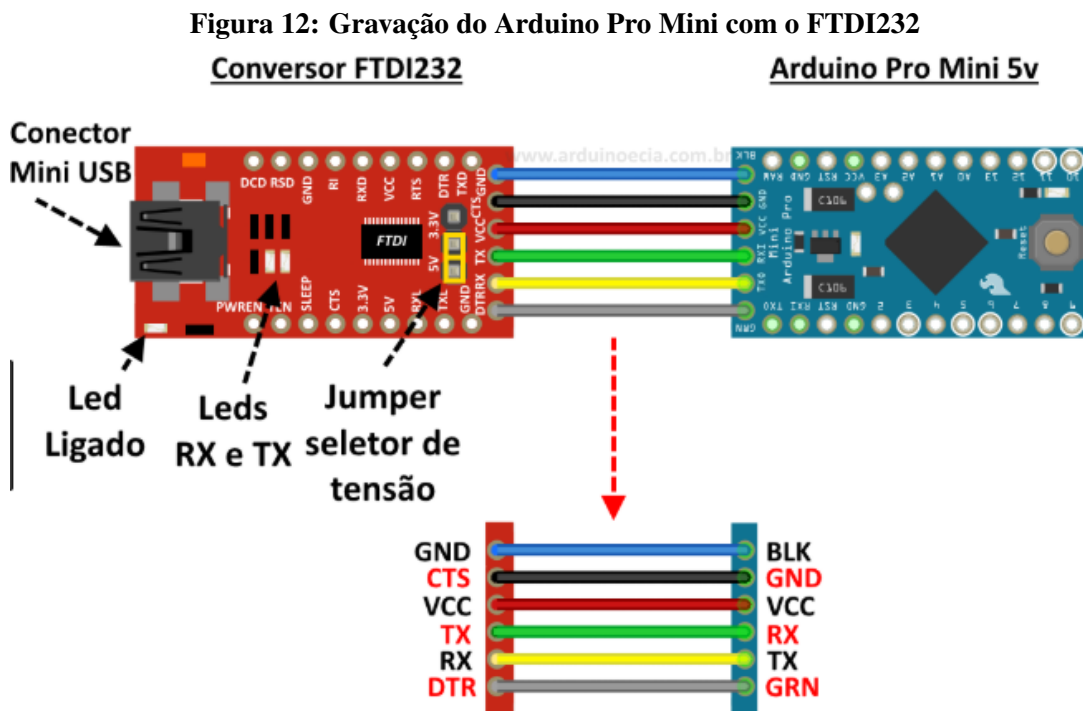
Depois do protótipo completo e testado, foram realizados testes utilizando três nós, porém substituindo o Arduino Mega por uma placa com praticamente as mesmas características,

mas com um tamanho reduzido, o Arduino Pro Mini mostrado na Figura 13. Isso permitirá futuramente ao nó possuir um tamanho reduzido.

3.3.2 GRAVAÇÃO

Porém para realizar essa mudança é necessário que todo o código produzido dentro da IDE do Arduino seja gravada dentro do Arduino Pro Mini, assim como realizado com o Arduino Mega. Mas para realizar essa gravação, é necessário um módulo conversor FTDI232 para que o cabo de transmissão de dados seja ligado no computador e outro nele. E a partir dele, através de seus pinos, os dados serão gravados dentro do Arduino Pro Mini.

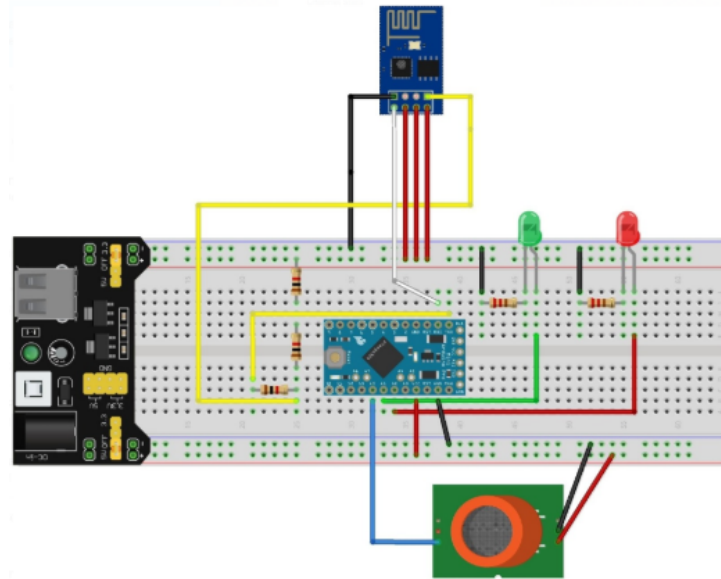
Contudo, como o Arduino Mega trabalha com entradas analógicas em uma faixa de tensão entre 0 e 5V e o Arduino Pro Mini trabalha com entradas analógicas na faixa de 0 a 7V, foi necessário o ajuste para realizar a equivalência dessas duas faixas com o objetivo de fazer com que as aferições vindas do Arduino Pro Mini mostrassem dados equivalentes aos obtidos no Arduino Mega. Para realizar a gravação, do código compilado no computador, para o Arduino Pro Mini, foi necessária a seguinte ligação intermediária com o FTDI232 como mostrada na Figura 12.



Fonte: Arduino (2018).

Após a gravação, foi realizada a montagem do mesmo protótipo que aparece na Figura 11, mas com a diferença de utilizar o Arduino Pro Mini que possui um tamanho

Figura 13: Nó com Arduino Pro Mini



Fonte: Arduino (2018).

consideravelmente menor e com características parecidas no que diz respeito ao processamento de informações como mostrado na Figura 13.

3.4 BROCKER MQTT ONLINE

Para ser utilizado como um *broker online* foi escolhido o Thingspeak que funciona como uma plataforma que recebe os dados do sensor, os apresenta de uma forma intuitiva através de gráficos, os analisa e, a partir dessa análise, pode retornar algum tipo de ação, dependendo da necessidade da aplicação. Além desses recursos, o ThingSpeak tem a vantagem de poder ser utilizado de forma gratuita para aplicações pequenas. A partir do cadastro na plataforma, já é possível registrar vários canais que nada mais são que os nós com sensores que se comunicarão com o *broker online* através do MQTT.

A criação de um canal é simples, basta informar o nome do nó e o nome do sensor que enviará as informações. Mas é possível o preenchimento de outras várias informações que possibilitam o cadastro do nó de forma mais detalhada como mostrado na Figura 14.

A manutenção desses canais pode ser realizada na tela principal “Meus canais” mostrada na Figura 15.

Na opção *API Keys* como na Figura 16 temos acesso às chaves para leitura e escrita nos canais, ou seja, para enviar a um dado de um nó até o *broker*, é necessário que o nó conheça

Figura 14: Criação de um novo canal no Thingspeak

The screenshot shows the 'New Channel' page on the Thingspeak website. The page is divided into two main sections: 'New Channel' on the left and 'Ajuda' (Help) on the right.

New Channel Form:

- Nome:** A text input field for the channel name.
- Descrição:** A text area for the channel description.
- Campo 1-8:** Eight fields for defining data fields. Each field has a label input (e.g., 'Rótulo do campo 1') and a checkbox to enable it.
- Metadata:** A text area for entering metadata.
- Tags:** A text area for entering tags, with a note '(Tags are comma separated)'. Below it is a 'URL' field.
- Link to GitHub:** A text area for entering a GitHub repository URL.
- Altitude:** A text area for entering the channel's altitude.

Ajuda (Help) Section:

Channel Settings:

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Using the Channel:

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

Fonte: ThingSpeak (2018).

Figura 15: Manutenção dos canais

The screenshot shows the 'Meus Canais' (My Channels) page. At the top, there is a 'New Channel' button and a search bar labeled 'Search by tag'. Below is a table listing the user's channels.

Nome	Criada em	Updated
Monitoramento - Sensor de Gás 001 Private Públicos Settings Sharing API Keys Data Import / Export	2018-05-20	2018-06-14 22:45
Monitoramento - Sensor de Gás 002 Private Públicos Settings Sharing API Keys Data Import / Export	2018-05-20	2018-06-14 22:45
Monitoramento - Sensor de Gás 003 Private Públicos Settings Sharing API Keys Data Import / Export	2018-05-20	2018-06-14 22:46

Fonte: ThingSpeak (2018).

a chave de escrita do broker, dessa forma também a aplicação *web* deve conhecer a chave de leitura desse canal, para que possa acessar o dado e apresentar ao usuário.

Figura 16: API Keys

API Requests

Update a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=CZBWKIZJIHUIP09Q&field
```

Get a Channel Feed

```
GET https://api.thingspeak.com/channels/500662/feeds.json?results=2
```

Get a Channel Field

```
GET https://api.thingspeak.com/channels/500662/fields/1.json?results
```

Get Channel Status Updates

```
GET https://api.thingspeak.com/channels/500662/status.json
```

Fonte: ThingSpeak (2018).

Nesta mesma Figura 16 é apresentada a tela onde é possível obter as requisições do tipo *GET* para realizar o envio dos dados do nó para o servidor, assim como obter esses dados do servidor para a aplicação *web*.

4 RESULTADOS

O projeto inicialmente tinha como ideia a utilização do módulo Arduino Uno para a fabricação do protótipo. Porém foi percebida uma dificuldade muito grande ao realizar a conexão entre o módulo Arduino Uno com módulo de Wifi ESP-01, pois esse Arduino em específico trabalha com taxas de 9.600 baud e o módulo ESP-01 trabalha com 115.200 baud. Sendo assim foi realizada a troca do Arduino em questão por um Arduino intermediário que possui o suporte a taxas mais altas e assim foi possível comprovar com os mesmos testes, que o problema foi sanado.

Nos capítulos anteriores foi feita a apresentação dos conceitos utilizados para a construção dos protótipos no Capítulo 2, uma análise sobre o projeto de construção dos nós, bem como sua comunicação com o *brocker* online e a apresentação dos dados em tempo real. Agora serão comentados os resultados obtidos através dos protótipos.

4.1 AFERIÇÕES OBTIDAS

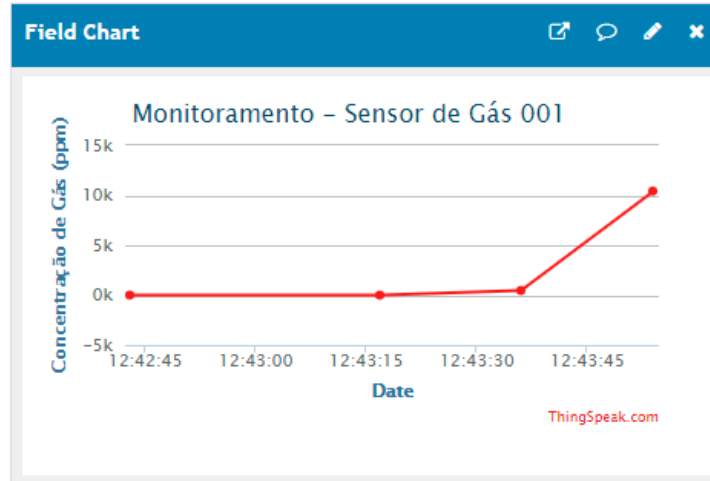
Para simular realização de uma aferição do nível de GLP presente em um ambiente fechado, utilizando os três nós criados, cada nó foi colocado dentro de um pequeno recipiente de plástico com uma pequena abertura no fundo para a inserção de um isqueiro, através do qual foi liberado o gás. Os três nós foram submetidos a testes diferentes, sendo o primeiro deles com a presença intensa de GLP, ou seja, foi liberado gás durante 10 segundos dentro do ambiente. No segundo, foi liberado o GLP durante 5 segundos. No terceiro, foi liberado o GLP durante 5 segundos e posteriormente foi retirado o isqueiro, fazendo com que houvesse uma passagem para o GLP ser dissipado aos poucos.

4.1.1 SENSOR DE GÁS 001

A figura 17 apresenta os dados do primeiro teste de aferição recebidos através dos gráficos padrão contidos na plataforma Thingspeak. Neste teste, o nó foi colocado dentro do

recipiente pequeno de plástico com uma abertura para a incidência de GLP através de um isqueiro. Foi possível notar que antes da aplicação do GLP, os valores permaneceram muito próximos a 0 e após a aplicação, alcançou-se o pico de 10.000ppm atingido nesse teste.

Figura 17: Sensor de Gás 001



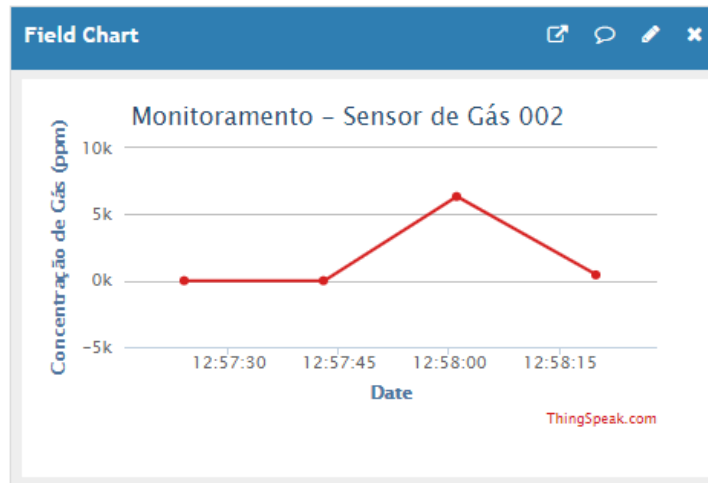
Fonte: ThingSpeak (2018).

4.1.2 SENSOR DE GÁS 002

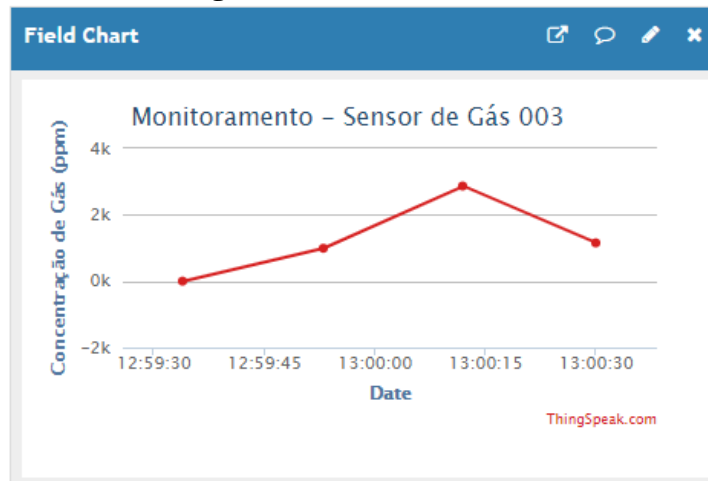
A figura 18 apresenta os dados do segundo teste de aferição recebidos através dos gráficos padrão contidos na plataforma ThingSpeak. Neste teste, o nó foi colocado dentro do recipiente pequeno de plástico com uma abertura para a incidência de GLP através de um isqueiro, porém a quantidade de gás aplicada foi menor que no primeiro teste. Foi possível notar que antes da aplicação do GLP, os valores permaneceram muito próximos a 0 e após a aplicação, alcançou-se os valores maiores e próximos a 5.000ppm.

4.1.3 SENSOR DE GÁS 003

A figura 19 apresenta os dados do terceiro teste de aferição recebidos através dos gráficos padrão contidos na plataforma ThingSpeak. Como nos testes anteriores, o nó foi colocado dentro do recipiente pequeno de plástico com uma abertura para a incidência de GLP através de um isqueiro, mas neste teste o gás foi aplicado aos poucos e após isso, foi permitido que o GLP se dissipasse aos poucos. Foi possível notar que durante a aplicação gradual do GLP, os valores começaram a subir e atingiram aproximadamente 3.000ppm e após a abertura da fresta o valor foi diminuindo.

Figura 18: Sensor de Gás 002

Fonte: ThingSpeak (2018).

Figura 19: Sensor de Gás 003

Fonte: ThingSpeak (2018).

4.2 DISCUSSÃO

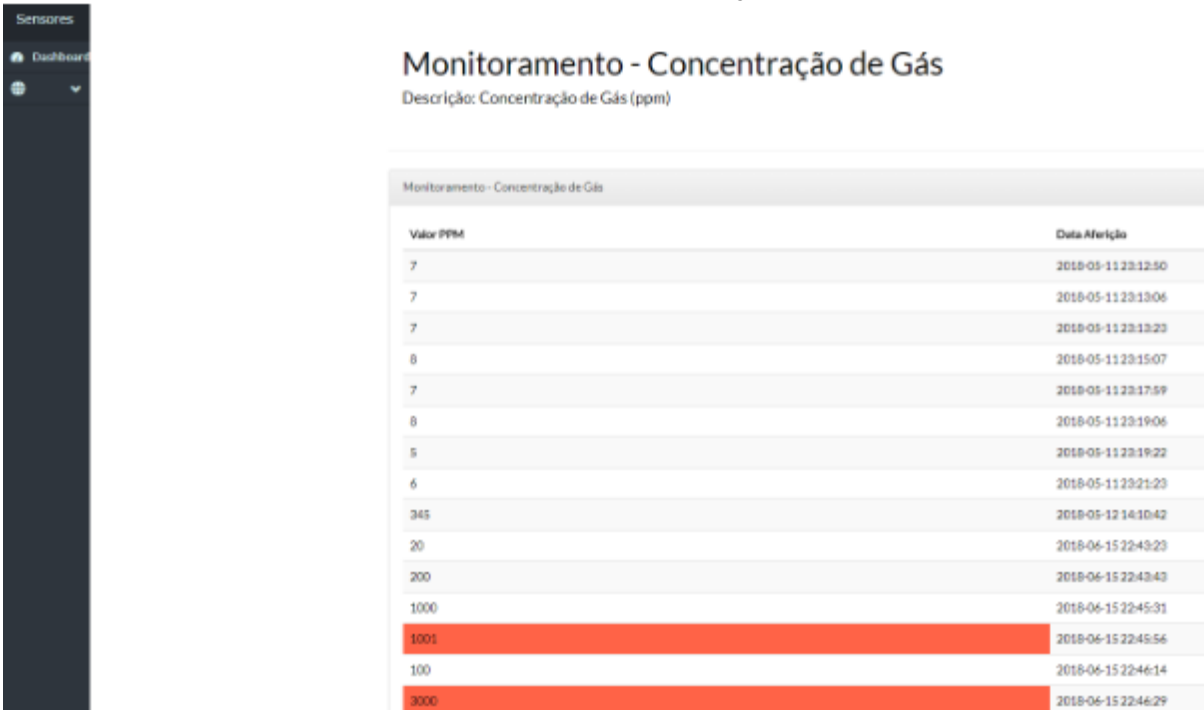
Após a aferição em cada um dos três nós, foi possível observar a aferição em tempo real de cada nó individualmente como seguem nas Figuras 17, 18 e 19. Para verificar o funcionamento dos nós, foi colocado um isqueiro perto de cada nó para que a concentração do gás fosse maior, assim podendo obter os resultados visualizados nos gráficos, onde os valores mais baixos que foram zero aconteceram na ausência total do GLP e os valores de aproximadamente 10.000ppm foram obtidos quando foi liberado uma maior quantidade de GLP dentro do recipiente. A condição mediana foi obtida quando a quantidade de GLP incidida foi menor que no primeiro teste. Pode-se perceber isso tanto no segundo ou terceiro teste, onde

no segundo foi incidida uma quantidade menor de maneira ininterrupta atinjindo o valor médio de 5.000ppm e no terceiro teste incidindo o GLP com algumas interrupções, atingindo valores entre 0ppm e 4.000ppm de forma gradativa.

4.3 INTERFACE WEB

Para obter uma visualização personalizada foi criado uma página *Web* contendo o histórico das informações obtidas através das aferições. A figura 20 mostra um histórico recente das aferições. Apesar da frequência de aferições serem realizadas de forma contínua, a atualização desses dados direto na página web pode ser alterada conforme preferência ou necessidade. É possível notar que quando o nível de GLP encontrado em qualquer aferição atingir o limite máximo do padrão aceitável, que pode ser alterado de acordo com a necessidade e tolerância de vazamentos de gases, neste exemplo definido como valores acima de 1.000ppm, o registro ficará destacado, apontando dessa forma que alguma atitude deve ser tomada para a resolução do problema.

Figura 20: Interface Web criada para visualizar a situação atual de cada né em tempo real



Valor PPM	Data Aferição
7	2018-05-11 23:12:50
7	2018-05-11 23:13:06
7	2018-05-11 23:13:23
8	2018-05-11 23:15:07
7	2018-05-11 23:17:59
8	2018-05-11 23:19:06
5	2018-05-11 23:19:22
6	2018-05-11 23:21:23
345	2018-05-12 14:10:42
20	2018-06-15 22:43:23
200	2018-06-15 22:43:43
1000	2018-06-15 22:45:31
1001	2018-06-15 22:45:56
100	2018-06-15 22:46:14
3000	2018-06-15 22:46:29

Fonte: Autoria própria.

5 TRABALHOS FUTUROS

O protótipo produzido neste trabalho utilizou recursos de *hardware* e *software* visando a construção de uma ferramenta útil para determinados mercados, mas que também fossem de simples montagem e programação, e de baixo custo. Para a construção de um produto que possa ser testado no mercado comercial, são necessárias algumas alterações para tornar o projeto mais confiável no que diz respeito ao processo de envio de informações pelo nó, o que envolve a utilização de sensores e *hardwares* certificados.

5.1 MICROCONTROLADOR

Os microcontroladores utilizados durante a o projeto do protótipo foram escolhidos pela sua facilidade de programação, simplicidade e compatibilidade na integração com sensores e outros microcontroladores. Porém, para a construção do produto final é cabível a utilização de um microcontrolador mais especializado para internet das coisas. Neste caso, um microcontrolador que entrega confiabilidade em sua construção e segue os princípios da internet das coisas é o CC2650 da Texas Instruments que, em seu desenvolvimento, levou em consideração um bom desempenho juntamente com a economia de energia. Possíveis trabalhos futuros incluem construir o produto final de menor tamanho possível, de maneira a retirar a placa wifi ESP8266 do projeto uma vez que a placa da Texas já faz comunicação através de *bluetooth* e wifi de forma nativa, diminuindo assim o tamanho do produto.

5.2 SENSOR DE GLP

O módulo de sensor de GLP, apesar de possuir uma calibragem para levar em consideração o ponto de referência no ambiente e ser amplamente utilizado em projetos parecidos, pode apresentar algumas falhas na precisão de suas aferições, pois esse módulo especificamente não atende as regras da NR-33 que regulamenta aspectos referentes à detecção de gases e calibração de sensores. As empresas que fazem as aferições de GLP de forma manual utilizam equipamentos robustos e certificados pela NR-33, e por isso seu custo é extremamente

elevado. Por exemplo, modelos simples como o Akron Kr4100, custam em média R\$3.000,00. Uma alternativa é adequar o sensor dentro das normas através da adição de complementos de *hardware* que faça a redundância da aferição e de complementos de *software* que realize os testes e homologação da aferição em tempo real.

5.3 SERVIDOR NA NUVEM

O servidor utilizado para o protótipo foi o Thingspeak por ser de fácil implementação e ser gratuito para um número pequeno de nós. Entretanto, para aplicações comerciais são necessários mais canais e mais espaço para o armazenamento dos dados coletados para posterior análise. Outras empresas, como a Amazon e o Google, possuem produtos específicos para IoT que atendem a essa necessidade e também oferecem opções de análise mais completa dos dados, assim como opções próprias de *dashboards* para a visualização das informações pelos usuários.

6 CONCLUSÃO

Neste trabalho, foram relatados aspectos importantes relacionados a uma solução para aferição de gases GLP a partir de um protótipo reunindo módulos e sensores conectados, e com o objetivo de transmitir informações relacionadas à quantidade de gases presentes no meio em tempo real. Foram levados em consideração para a viabilidade do projeto questões como a redução de gastos financeiros com material humano, diminuição de riscos constantes à saúde dos funcionários e baixo custo com a utilização de módulos e sensores compatíveis com Arduino. Também foi apresentada uma breve visão sobre os conceitos e equipamentos utilizados durante o projeto, assim como a configuração e montagem do protótipo e servidor web.

Posteriormente, foram feitas análises sobre a criação do protótipo, a transmissão dos dados e sua apresentação no servidor web. Nesse contexto, foi levado em consideração aspectos sobre a montagem das placas, sensor e módulos, e também sobre a transmissão dos dados para o servidor. No aspecto geral, pode-se perceber que, além das economias e diminuição de riscos citados anteriormente, também deve ser levado em consideração que é possível obter o nível de gás em tempo real através de uma interface gráfica criada por meio de ferramentas gratuitas e, assim, tornando visíveis as possíveis alterações fora da normalidade de níveis de gases GLP. É necessário salientar também que, como cada parte do projeto é modular, é possível fazer alterações tanto em partes do *hardware* quanto do *software*, sem resultar em quaisquer mudanças no resultado esperado.

REFERÊNCIAS

- ABREU, F. **Gás Liquefeito de Petróleo (GLP)**. 2015. Disponível em: <<https://betaeq.com.br/index.php/2015/09/30/gas-liquefeito-de-petroleo-glp/>>. Acesso em: 27 ago. 2018.
- ARDUINO. **Arduino**. 2018. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 28 jun. 2018.
- AZURE, W. **Windows Azure**. 2018. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>>. Acesso em: 16 jun. 2018.
- CAMPOS, A. **ESP8266: Comandos AT**. 2018. Disponível em: <<https://br-arduino.org/2015/11/esp8266-comandos-at.html>>. Acesso em: 28 ago. 2018.
- CARISSIMI, A. **Desmistificando a Computação em Nuvem**. 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/erad-rs/2015/002.pdf>>. Acesso em: 29 ago. 2018.
- CARVALHO; SOUZA; LIMA. **Internet das Coisas e Sua Aplicação em Bibliotecas**. [S.l.]: Revista Gestão.Org, 2016. ISSN 16791827.
- CIA, A. e. **Conecte o Arduino na rede wireless 802.11 com o módulo ESP8266**. 2015. Disponível em: <<https://www.arduinoocia.com.br/2015/03/arduino-modulo-wireless-esp8266.html>>. Acesso em: 29 jun. 2018.
- CURVELLO, A. **Embarcados**. 2015. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 29 jun. 2018.
- EMBARCADOS. **Embarcados - Sua fonte de informação sobre sistemas embarcados**. 2018. Disponível em: <<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>>. Acesso em: 18 mai. 2018.
- GARCIA, T. **MDN Webdocs**. 2018. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>>. Acesso em: 1 jun. 2018.
- JIANKAILI. **Seeed Studio**. 2015. Disponível em: <<https://docs.particle.io/as-sets/datasheets/electronsensorkit/MQ-2.pdf>>. Acesso em: 15 jun. 2018.
- KOLBAN, N. **Kolban's Book on ESP8266**. [S.l.: s.n.], 2016.
- MATHWORKS. **MathWorks**. 2018. Disponível em: <<https://www.mathworks.com/help/thingspeak/>>. Acesso em: 12 jul. 2018.
- OLIVEIRA, S. d. **Internet das Coisas com ESP8266, Arduino e Raspberry PI**. [S.l.]: Novatec, 2017.
- PETROBRAS. **Gás Liquefeito de Petróleo. Informações Técnicas**. 2013. Disponível em: <<http://sites.petrobras.com.br/minisite/assistenciatecnica/public/downloads/manual-tecnico-gas-liquefeito-petrobras-assistencia-tecnica-petrobras.pdf>>. Acesso em: 27 ago. 2018.

RAMOS, A. **O que é MVC?** <https://tableless.com.br/mvc-afinal-e-o-que/>: Tableless, 2015.

SILVA, A. L. S. da. **Gás Liquefeito de Petróleo (GLP)**. 2018. Disponível em: <<https://www.infoescola.com/quimica/gas-liquefeito-de-petroleo-glp/>>. Acesso em: 27 ago. 2018.

TEAM, E. S. I. **ESP8266 AT Instruction Set**. <https://static.efetividade.net/img/esp8266-at-63240.pdf>: Espressif Systems IOT Team, 2015.

THINGSPEAK. **ThingSpeak Documentation**. 2018. Disponível em: <<https://www.mathworks.com/help/thingspeak/>>. Acesso em: 20 mai. 2018.

WILLIAMS, W. **Laravel 5 Essencial - Alta Produtividade no mundo Real**. [S.l.]: Code Education, 2015.

YUAN, M. **IBM**. 2017. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 15 jun. 2018.