

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

PATRICK SUAREZ FONSECA

**GÁSNAMEIDA: PLATAFORMA DE CONTROLE DE
CONCENTRAÇÃO DE GÁS**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2018

PATRICK SUAREZ FONSECA

**GÁSNAME DIDA: PLATAFORMA DE CONTROLE DE
CONCENTRAÇÃO DE GÁS**

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



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

GÁSNAMEDIDA: PLATAFORMA DE CONTROLE DE CONCENTRAÇÃO DE GÁS

por

PATRICK SUAREZ FONSECA

Esta monografia foi apresentada em 13 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 -

AGRADECIMENTOS

Agradeço a todos os professores, amigos de turma, coordenação, direção e demais funcionários da instituição pela paciência com a qual me guiaram até o fim da jornada. Sem eles esse texto jamais poderia ser escrito.

Agradecimento especial ao professor Dr. Marcos Eduardo Pivaro Monteiro, pela orientação nesse trabalho.

Agradecimento também especial ao meu antes colega de turma e agora amigo Jordano Cherobim, pela parceria no projeto, pelas dicas preciosas e todos os trabalhos que desenvolvemos em equipe, no decorrer de todo o curso.

RESUMO

FONSECA, Patrick Suarez. **GÁSNAMEDIDA: plataforma de controle de concentração de gás.** 2018. 73 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

É de conhecimento público e geral que o Gás Liquefeito de Petróleo (GLP) é utilizado e manipulado em grande parte dos setores como casas, restaurantes, centros comerciais e indústrias. Tal gás é utilizado para diversos fins, como no aquecimento do espaço, da água, dos alimentos e na manufatura de produtos. Devido à alta volatilidade desse gás é muito comum que vazamentos aconteçam e os ambientes se tornem insalubres e perigosos. Particularmente na indústria, sistemas de controle são tipicamente utilizados para que sejam feitas as tomadas de decisão de manutenção e substituição das máquinas e demais elementos que compõem a planta. Nesse contexto, o objetivo deste trabalho é desenvolver e construir um protótipo de plataforma de medição dos níveis de gás no ambiente.

Palavras-chave: Gás. GLP. Vazamento. Controle. Instrumentação.

ABSTRACT

FONSECA, Patrick Suarez. **GÁSNAMEDIDA: gas concentration control platform.** 2018. 73 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

It is well known that Liquefied Petroleum Gas (LPG) is used and manipulated in many sectors such as houses, restaurants, shopping malls, and industries. Such gas is used for various purposes, such as heating space, water, food and for the manufacture of products. Due to the high volatility of this gas, it is very common that leaks happen, causing the environments to become unhealthy and dangerous. Particularly in industry, control systems are typically used to make decision-making for maintenance and replacement of machines and other elements of the plant. In this context, this work aims to develop and build a prototype platform for measuring gas levels in the environment.

Keywords: Gas. LPG. Leakage. Control. Instrumentation.

LISTA DE FIGURAS

Figura 1 - Diagrama em blocos da solução	21
Figura 2 - Diagrama de pinos do Arduino Uno R3.....	29
Figura 3 - Diagrama de pinos do Arduino MEGA 2560	31
Figura 4 - Diagrama de pinos do Arduino Pro Mini.....	32
Figura 5 - Diagrama de pinos do ESP-01.....	36
Figura 6 - Sensor de gás MQ-2	38
Figura 7 - IDE do Arduino com o código de testes do ESP8266	47
Figura 8 - Monitor serial do Arduino Uno R3	48
Figura 9 - IDE do Arduino com o código de testes do ESP8266 no Arduino Mega 2560	49
Figura 10 - Dashboard do ThingSpeak obtido com comandos AT	51
Figura 11 - IDE de gravação do Arduino Pro Mini	56
Figura 12 - Tela “Meus Canais” do ThingSpeak.....	57
Figura 13 - Tela “New Channel” do ThingSpeak	58
Figura 14 - Canal criado no ThingSpeak.....	59
Figura 15 - Chaves de escrita, leitura e API´s do ThingSpeak.....	59
Figura 16 - Testes com o Packet Sender	61
Figura 17 - Dashboard do Thingspeak alimentado pelo Packet Sender	61
Figura 18 - Dashboard do ThingSpeak - testes com o nó 1	64
Figura 19 - Dashboard do ThingSpeak - testes com o nó 2	64
Figura 20 - Dashboard do ThingSpeak - testes com o nó 3	65

LISTA DE DIAGRAMAS

Diagrama 1 - Arduino Uno conectado ao ESP8266	42
Diagrama 2 - Arduino Mega 2560 conectado ao ESP8266.....	43
Diagrama 3 - Calibração do sensor MQ-2 com o Arduino Mega 2560	43
Diagrama 4 - Bloco "nó" completo.....	44
Diagrama 5 - Hardware utilizado para a gravação do Arduino Pro Mini.....	45
Diagrama 6 - Protótipo do bloco "nó" com Arduino Pro Mini	45

LISTA DE FOTOGRAFIAS

Fotografia 1 - Circuito Arduino Uno conectado ao ESP8266.....	46
Fotografia 2 - Circuito Arduino Mega 2560 conectado ao ESP8266	49
Fotografia 3 - Circuito para calibração do sensor MQ-2 com Arduino Mega 2560	52
Fotografia 4 - Protótipo do bloco "nó" construído	54
Fotografia 5 - Circuito utilizado para a gravação do Arduino Pro Mini.....	55
Fotografia 6 - Bloco "nó" construído com Arduino Pro Mini.....	57
Fotografia 7 - Protótipo da plataforma completa	62

LISTA DE GRÁFICOS, QUADROS E TABELAS

Gráfico 1 - Curva R_s/R_o em função da concentração de gás em ppm	39
Quadro 1 - Descrição e função dos pinos do ESP-01	36
Quadro 2 - Comandos AT básicos	37
Quadro 3 - Sequência de comandos para comunicação entre Arduino e ThingSpeak	50
Tabela 1 - Preço total de cada nó do protótipo.....	41

LISTA DE SIGLAS

AP	<i>Access Point</i> (ou ponto de acesso)
BD	Banco de Dados
CNTP	Condições Normais de Temperatura e Pressão
CSS	<i>Cascading Style Sheets</i> (ou folhas de estilo em cascatas)
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i> (ou memória somente leitura programável e apagável eletricamente)
GLP	Gás Liquefeito de Petróleo (ou <i>liquefied petroleum gas</i>)
HART	<i>Highway Addressable Remote Transducer</i> (ou via de dados endereçável por transdutor remoto)
HTML	<i>HyperText Markup Language</i> (ou linguagem de marcação de hipertexto)
HTTP	<i>HyperText Transfer Protocol</i> (ou protocolo de transferência de hipertexto)
ISP	<i>In-System Programming</i> (ou programação serial em circuito)
LPG	<i>Liquefied Petroleum Gas</i> (ou gás liquefeito de petróleo)
PWM	<i>Pulse Width Modulation</i> (ou modulação por largura de pulso)
RAM	<i>Random Access Memory</i> (ou memória de acesso aleatório)
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> (ou protocolo de controle de transmissão/protocolo de internet)
WirelessHART	<i>Wireless Highway Addressable Remote Transducer</i> (ou via de dados endereçável por transdutor remoto sem fio)

LISTA DE ACRÔNIMOS

AS-i	<i>Sensor Interface Actuator</i> (ou atuador de interface de sensor)
Cloud	Nuvem, elemento que recebe informações de dispositivos IoT
CoAP	<i>Constrained Application Protocol</i> (ou protocolo de aplicação restrita)
Datacenter	Centro de Processamento de Dados
DeviceNet	<i>Device Network</i> (ou rede de dispositivos)
DIP	<i>Dual In-line Package</i> (ou pacote duplo em linha)
Ethernet	Arquitetura de interconexão para redes locais
EthernetIndustrial	Arquitetura de interconexão para redes locais industriais
IoT	<i>Internet of Things</i> (ou internet das coisas)
JavaScript	Linguagem de script para páginas Web
MQTT	<i>Message Queuing Telemetry Transport</i> (ou transporte de telemetria de fila de mensagens)
PROFIBUS	<i>Process Field Bus</i> (ou barramento de campo de processo)
SSID	<i>Service Set Identifier</i> (ou identificadores do conjunto de serviços)
Web	<i>World Wide Web</i> (ou rede mundial de computadores)
Wi-Fi	<i>Wireless Fidelity</i> (ou internet sem fio)
Zigbee	Rede criada pelo IEEE em conjunto com a ZigBee Alliance

SUMÁRIO

1 INTRODUÇÃO	15
1.1 PROBLEMA	17
1.2 OBJETIVOS.....	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivos Específicos.....	18
1.3 JUSTIFICATIVA.....	18
1.4 ESTRUTURA DO TRABALHO	19
2 METODOLOGIA.....	20
2.1 FUNDAMENTAÇÃO TEÓRICA	21
2.1.1 Internet das Coisas e a Nuvem.....	21
2.1.2 Computação em Nuvem (Cloud Computing)	22
2.1.3 Plataformas Comerciais de Nuvem.....	23
2.1.4 Datalog na Nuvem	24
2.2 TECNOLOGIAS UTILIZADAS	24
2.2.1 ThingSpeak.....	24
2.2.2 Arduino	25
2.2.3 Módulos Periféricos Arduino	25
2.2.4 Arduino Uno R3	26
2.2.4.1 Microcontrolador	26
2.2.4.2 Alimentação	27
2.2.4.3 Interface de programação	27
2.2.4.4 Interface com o usuário.....	28
2.2.4.5 Diagrama de pinos	28
2.2.5 Arduino Mega 2560.....	29
2.2.5.1 Microcontrolador	29
2.2.5.2 Alimentação	30
2.2.5.3 Interface de programação	30
2.2.5.4 Interface com o usuário.....	30
2.2.5.5 Diagrama de pinos	30
2.2.6 Arduino Pro Mini	31
2.2.6.1 Microcontrolador	31
2.2.6.2 Alimentação	31
2.2.6.3 Interface de programação	32
2.2.6.4 Interface com o usuário.....	32
2.2.6.5 Diagrama de pinos	32
2.2.7 Módulo Wi-Fi ESP8266.....	33
2.2.7.1 ESP8266.....	33
2.2.7.2 Módulos com o ESP8266.....	34

2.2.7.3	Considerações sobre o modelo ESP.01	35
2.2.7.4	Diagrama de pinos do ESP8266	36
2.2.7.5	Comandos AT	36
2.2.8	Sensor de Gás MQ-2	38
2.2.8.1	Cálculo da inclinação da curva LPG	39
3	PROJETO.....	40
3.1	RECURSOS UTILIZADOS.....	40
3.1.1	Componentes.....	40
3.1.2	Gastos com o Protótipo	41
3.2	DIAGRAMAS	41
3.2.1	Diagrama 1: Arduino Uno R3 Conectado ao ESP8266.....	42
3.2.2	Diagrama 2: Arduino Mega 2560 Conectado ao ESP8266.....	42
3.2.3	Diagrama 3: Arduino Mega 2560, ESP8266 e Sensor MQ-2.....	43
3.2.4	Diagrama 4: Bloco Nó Completo.....	44
3.2.5	Diagrama 5: Circuito de Gravação do Arduino Pro Mini	44
3.2.6	Diagrama 6: Protótipo do Bloco Nó com Arduino Pro Mini	45
3.3	PROTOTIPAGEM	46
3.3.1	Primeiro Protótipo: Interface Arduino Uno R3 - ESP8266.....	46
3.3.2	Segundo Protótipo: Interface Arduino Mega 2560 - ESP8266.....	48
3.3.3	Terceiro Protótipo: Interface Arduino Mega 2560 - ESP8266 - MQ-2.....	52
3.3.4	Integração do Bloco Nó	54
3.3.5	Integração do Bloco Nó - Arduino Pro Mini.....	54
3.3.5.1	Circuito de gravação do Arduino Pro Mini.....	55
3.3.5.2	Gravação do Arduino Pro Mini	55
3.3.5.3	Quarto protótipo: Arduino Pro Mini.....	56
3.4	CONCENTRADOR	57
3.5	SOFTWARE.....	57
3.5.1	Criação do Canal no ThingSpeak.....	57
3.5.2	Testes e Encapsulamento do Bloco “BD/Cloud”	60
3.5.3	Integração	62
4	RESULTADOS	63
4.1	SENSOR DE GÁS 001	63
4.2	SENSOR DE GÁS 002	64
4.3	SENSOR DE GÁS 003	65
4.4	DISCUSSÃO.....	65
5	CONSIDERAÇÕES FINAIS	67
6	TRABALHOS FUTUROS	68
6.1	MICRONTROLADOR.....	68
6.2	SENSOR.....	68
6.3	COMUNICAÇÃO WI-FI.....	69
6.4	CONCENTRADOR	69

6.5 BD/CLOUD	70
6.6 PROTOCOLO DE APLICAÇÃO.....	70
REFERÊNCIAS.....	72

1 INTRODUÇÃO

O “gás de cozinha”, como é conhecido popularmente o gás liquefeito de petróleo por causa de sua utilização principal na cocção de alimentos, é uma das frações mais leves do petróleo e sua queima é muito limpa, com baixíssima emissão de poluentes. Por causa dessas características, ele é utilizado em ambientes fechados, como na cozinha de casa, ou em aplicações industriais sensíveis a poluentes, como na fabricação de vidros, cerâmicas e alimentos (PETROBRÁS, 2018).

Em substituição à lenha, querosene e óleo combustível, o GLP desempenha um importante papel na preservação ambiental, pois se trata de um combustível de elevado rendimento energético e possui uma combustão muito eficiente, que não produz resíduos tóxicos, contribuindo para o progresso socioeconômico e desenvolvimento sustentável do país (MOURA, 2012).

Define-se como Gás Liquefeito do Petróleo (GLP), a mistura formada, em sua quase totalidade, por moléculas de carbono e hidrogênio (hidrocarbonetos) de três a quatro átomos de carbono que, embora gasosos nas Condições Normais de Temperatura e Pressão (CNTP), podem ser liquefeitos por resfriamento e/ou compressão (PETROBRÁS, 2013).

Os componentes do GLP podem ser comercializados separadamente como propano e propeno, butanos e butenos. A queima do GLP é limpa, comparada aos combustíveis mais pesados, com reduzido nível de emissão de particulados, SO_x e NO_x. Produz também baixo nível de emissões de CO₂ por sua alta proporção Hidrogênio/Carbono (PETROBRÁS, 2013).

A principal aplicação do GLP, em nível mundial, é no cozimento de alimentos, representando no Brasil cerca de 80% do consumo deste derivado. Como uso doméstico, pode-se citar ainda a calefação da água, o aquecimento de ambientes e atividades de lazer. Pode também ser utilizado como matéria-prima nos seguintes segmentos (PETROBRÁS, 2013):

- Comercial: hospitais, lavanderias, restaurantes, padarias, hotelaria, cozimento de alimentos, aquecimento de água, esterilização e climatização (PETROBRÁS, 2013);

- Siderúrgico: fundição, corte e solda de metais (custo menor que o acetileno, seu competidor) (PETROBRÁS, 2013);
- Petroquímico: fabricação de borracha, polímeros, álcoois e éteres (PETROBRÁS, 2013);
- Combustível industrial: indústria de vidros (moldagem, solda e acabamento), indústria cerâmica (queima e secagem), indústria de papel e celulose (secagem) e indústria alimentícia (PETROBRÁS, 2013);
- Agropecuário: secagem de grãos, controle de pragas e queima ervas daninhas, aquecimento e esterilização de ambiente de criação de animais (PETROBRÁS, 2013).

O GLP não é corrosivo, poluente e nem tóxico, mas se inalado em grande quantidade produz efeito anestésico e também asfixia, pois empurra o gás respirável do ambiente em que se encontra. O GLP não possui cor nem odor próprio, mas por motivo de segurança nele é adicionado a substância Mercaptano, ou Tiol, ainda nas refinarias, para facilitar sua detecção (PETROBRÁS, 2013).

Para evitar o acúmulo de GLP no ambiente, regras para armazenagem de botijões devem ser rigorosamente seguidas. Para facilitar a detecção, é obrigatória a adição de um agente odorizante ao GLP comercializado em botijões. Os odorizantes tradicionalmente utilizados são compostos a base de enxofre, conhecidos como mercaptans. O produto utilizado pela Petrobras é o etil-mercaptan (C_2H_5SH), numa proporção de 12 g/m³ de GLP. No combate a incêndios causados pela combustão do GLP, deve-se empregar extintores de água em neblina, pó químico ou CO₂, não sendo recomendável a extinção do fogo sem antes estancar o vazamento, devendo-se manter o recipiente resfriado com água após esta extinção. O vazamento do GLP pode provocar tonteados e irritações no sistema respiratório, olhos e queimaduras na pele (PETROBRÁS, 2013).

Conforme exposto por um dos principais fabricantes brasileiros de GLP, a Petrobrás, a detecção e controle primários de vazamentos desse gás é feito de maneira rudimentar, com a exposição do elemento humano a níveis nem sempre seguros de concentração do elemento. Dessa maneira, em aplicações críticas, torna-se necessário um tratamento automatizado do problema.

O presente trabalho trata essa questão, ou seja, a substituição dos sensores humanos (olfato e tato) por um sistema eletrônico que faz a detecção e medição da

concentração de gás no ambiente. No comércio já existem produtos que fazem a medição dos níveis de concentração desse gás em ambientes, porém se apresentam a elevados custos e nem sempre adaptados aos objetivos desse projeto. O objetivo deste projeto é construir um protótipo de baixo custo que faça as medições de concentração de gás e divulgue esses resultados em algum banco de dados que armazene e deixe disponível aos operadores para o devido tratamento matemático.

1.1 PROBLEMA

Em indústrias que utilizam Gás Liquefeito de Petróleo (GLP) na manufatura de produtos é muito comum que vazamentos aconteçam nos diversos equipamentos e máquinas que compõem a planta. Dependendo da concentração desse gás em determinado segmento, torna-se necessária a paralisação da linha de produção para que um determinado equipamento ou bloco do sistema seja submetido a manutenção ou substituição. Em muitos segmentos da indústria esse controle é feito de forma manual, com a exposição do elemento humano a níveis altos de toxicidade, insalubridade e periculosidade, tornando o sistema susceptível a erros de calibração, aferição e medição.

Através da experiência, conhecimento e vivência da equipe deste projeto com atividades deste segmento, constatou-se que na prática é necessário o controle dos níveis desse gás através de algum sistema automatizado que possa, em tempo real, fazer a aferição dos níveis de concentração de gás em diversos ambientes e sem a necessidade do elemento humano.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

A proposta é a elaboração e construção de um protótipo para um produto comercial capaz de implementar a medição dos níveis de gás na planta da indústria. Nós simples e de baixo custo monetário que possam ser dispostos em locais estratégicos da planta e que concentrem as informações coletadas em um banco de dados para que se possa processar o tratamento matemático específico para o

controle. Também será feita uma descrição detalhada de como o protótipo foi construído para que possa ser transformado em um produto final utilizando os conceitos vistos durante o curso de pós-graduação.

1.2.2 Objetivos Específicos

Para atender ao objetivo geral neste trabalho de conclusão de curso os seguintes objetivos específicos serão abordados:

- Construir um protótipo, de baixo custo monetário, altamente escalável e que tenha a capacidade de alimentar qualquer elemento concentrador com as informações advindas do meio;
- Implementar a comunicação dos nós com o banco de dados utilizando a internet sem fio (Wi-Fi, do inglês *Wireless Fidelity*);
- Utilizar uma *Cloud* de baixo custo monetário para a implementação do banco de dados;
- Relatar e descrever detalhadamente como o protótipo foi construído para que o mesmo possa ser transformado em um produto final utilizando os conceitos vistos durante o curso de pós-graduação.

1.3 JUSTIFICATIVA

Como é de conhecimento geral, o Gás Liquefeito de Petróleo (GLP) é utilizado nos mais variados ambientes como residências, restaurantes, centros comerciais, indústria e em praticamente todas as aplicações onde se deseja economia de energia elétrica. Também é de conhecimento geral que o controle desse gás é deveras rudimentar, constantemente impondo riscos às pessoas que o manipulam ou que eventualmente entram em contato por acidente.

Exposto isso, conclui-se que esse estudo é enfatizado para uso industrial, mas pode ser utilizado em quaisquer outras áreas nas quais esse tipo de gás é utilizado. A metodologia de desenvolvimento deste projeto pode ser aplicada a qualquer área que necessite um controle de concentração de GLP e pode também ser aplicada ao controle de concentração de outros gases, desde que as partes constituintes sejam substituídas por elementos que tratem especificamente cada gás particular.

1.4 ESTRUTURA DO TRABALHO

O trabalho tem a seguinte estrutura:

- **Capítulo 1 – Introdução:** são apresentados o problema, os objetivos do projeto, a justificativa e a estrutura geral do trabalho.
- **Capítulo 2 – Metodologia:** serão apresentados a divisão dos blocos do sistema e os aspectos de interconexão entre esses blocos. As tecnologias e dispositivos utilizados no projeto serão esmiuçadas em suas características fundamentais.
- **Capítulo 3 – Projeto:** serão apresentados os requisitos e diagramas dos circuitos que serão utilizados durante todo o projeto. Também serão apresentadas detalhadamente como foi a construção de cada bloco do sistema, as interfaces de entrada e saída, aspectos de interconexão e testes de cada bloco.
- **Capítulo 4 – Resultados:** serão apresentados os testes com a plataforma implementada e uma análise crítica dos resultados.
- **Capítulo 5 – Trabalhos futuros:** serão expostos aspectos industriais da transformação do protótipo em produto comercial.
- **Capítulo 6 – Considerações finais:** será feito um balanço geral do trabalho.

2 METODOLOGIA

Para o desenvolvimento do protótipo da plataforma GÁSNAMEIDA foram adotadas algumas premissas fundamentais, a saber: construir um produto que atenda às necessidades do cliente, evitando retrabalho e utilizando o mínimo de recursos monetários possíveis.

Esta solução é formada pelos seguintes componentes:

- **Nó:** é o bloco que fica em contato direto com o ambiente no qual se quer medir os níveis de concentração do gás. Trata-se de um sistema que contém um microcontrolador atuando em conjunto com o sensor de gás. O nó fica latente a maior parte do tempo. Após um determinado tempo o microcontrolador sai do estado de latência, faz a medição do nível de gás no sensor e volta à latência. Esse ciclo é repetido por tempo indeterminado;
- **Concentrador:** para cada ambiente é necessário um concentrador de informações, algum elemento que colete as informações de cada nó e envie para outro elemento que irá armazenar essas informações e deixará os dados no formato necessário para o devido tratamento matemático;
- **Banco de dados:** elemento que armazena as informações de todos os sensores da planta em tempo real. Essas informações ficam à disposição da equipe que monitora o sistema e toma decisões de manutenção ou substituição de equipamentos de acordo com os protocolos de segurança de cada instituição.

A interface entre os nós e o concentrador será sem fio, através de módulos Wi-Fi, de forma altamente escalável, sem sistemas redundantes. Trabalhos futuros incluem a implementação de redundância e criação de interfaces mais robustas se as condições do meio forem críticas no que diz respeito à propagação de ondas eletromagnéticas. Como já se sabe, ambientes ruidosos impõe a necessidade de implementação de protocolos mais robustos para a comunicação.

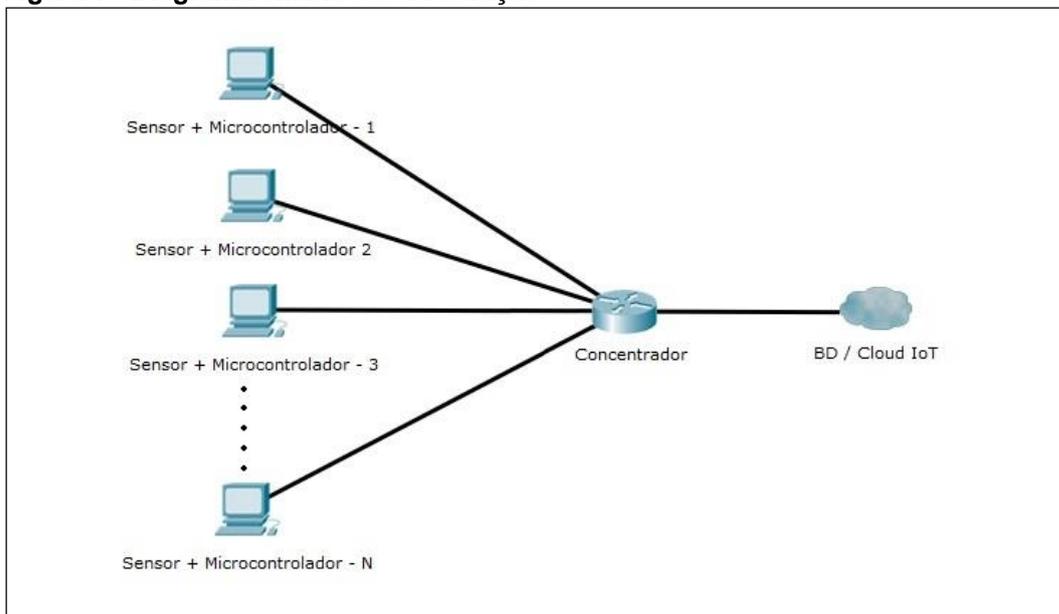
A interface entre o concentrador e o banco de dados se fará através da internet. Propõe-se a utilização de uma nuvem IoT de aplicação comercial de baixo custo ou até mesmo grátis para o armazenamento dos dados e apresentação do painel para os operadores do sistema. Posteriormente, para atender às necessidades de cada instituição em particular, pode-se implementar um sistema de banco de dados

proprietário, sem acesso externo às informações. Dependendo da criticidade do sigilo de cada instituição esse tipo de abordagem pode se tornar necessária.

A interface com o usuário final será feita através de uma *Cloud* que fornece painéis onde o usuário pode visualizar os gráficos de concentração de gás no ambiente onde o sensor está instalado. Essa *Cloud* fornece ferramentas de exportação dos dados em arquivos e também ferramentas para o processamento matemático da massa de dados obtida.

A solução proposta é ilustrada na Figura 1, onde se pode observar as conexões entre os blocos sensor/microcontrolador, concentrador e Banco de Dados (BD).

Figura 1 - Diagrama em blocos da solução



Fonte: Autoria própria, utilizando o software Cisco Packet Tracer.

2.1 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentadas a introdução teórica e o detalhamento dos aspectos fundamentais de cada tecnologia que será utilizada.

2.1.1 Internet das Coisas e a Nuvem

Computação em nuvem é essencial para o funcionamento efetivo e de baixo custo dos dispositivos de Internet da Coisas (IoT, do inglês *Internet of Things*). A

“nuvem” garante a confiabilidade dos sistemas a baixo custo. Os dispositivos podem permanecer em ambientes hostis e de baixa confiabilidade, desde que seus dados sejam enviados periodicamente para a nuvem. Esse modelo é econômico, pois dispensa a presença de servidores e possibilita a utilização de canais de comunicação de baixa confiabilidade (OLIVEIRA, 2017).

2.1.2 Computação em Nuvem (Cloud Computing)

Um termo que tem feito sucesso nos últimos tempos na computação é *Cloud Computing*, ou computação em nuvem. É difícil dizer onde começa a nuvem e até onde vai, visto que boa parte dos seus conceitos já é usada desde os primórdios da internet. Basicamente, pode-se dizer que a “nuvem” está sendo utilizada se os serviços essenciais de rede, seja armazenamento, processamento ou acesso, são hospedados em servidores externos (OLIVEIRA, 2017).

O conceito de servidor externo deve considerar o cenário inicial da internet, no qual todas as empresas que queriam estar presentes na internet precisavam instalar seus próprios servidores, que podem ser de armazenamento, banco de dados, rede mundial de computadores (web, do inglês *World Wide Web*) ou email. Isso exigia um bom parque computacional das empresas, procedimentos de backup constantes, além de equipamentos para suprir falhas de energia. Com o tempo, esses serviços começaram a ser transferidos para Centros de Processamento de Dados (Datacenter, do inglês *Data Center*) externos. Em síntese, tem-se que computação em nuvem é, basicamente, terceirizar os servidores (OLIVEIRA, 2017).

Os primeiros serviços a serem terceirizados foram web e email. Várias empresas iniciaram a oferta de terceirização desses serviços há vários anos, incluindo Google e Microsoft. Com o tempo, surgiram também provedores de aplicações e bancos de dados web. A terceirização desses serviços visa obter uma alta disponibilidade dos mesmos a baixo custo. São serviços essenciais para as empresas que ofertam serviços na internet, e é muito caro para uma empresa manter servidores de alta disponibilidade conectados à internet (OLIVEIRA, 2017).

O conceito de nuvem se tornou mais comum quando os serviços de armazenamento e processamento começaram a se tornar populares na rede. Os serviços de armazenamento e compartilhamento de arquivos, como Dropbox,

Microsoft Onedrive ou Google Drive, rapidamente tornaram-se populares, disponibilizando planos básicos gratuitos. E a comunidade começou a salvar seus arquivos na nuvem como opção de backup, acesso remoto e compartilhamento (OLIVEIRA, 2017).

A opção de processamento na nuvem foi inicialmente disponibilizada pela Amazon (OLIVEIRA, 2017). O serviço conta com a opção de criar máquinas virtualizadas, migrar entre servidores de virtualização, garantindo eficiência e confiabilidade às aplicações. A virtualização em nuvem abriu espaço para as empresas terceirizarem todo o seu parque computacional, desde servidores de bancos de dados até aplicações bancárias, de automação comercial ou industrial. Não apenas aplicações web, mas qualquer sistema, desenvolvido em qualquer linguagem (OLIVEIRA, 2017).

Para aplicações de Internet das Coisas, o conceito de nuvem se tornou um complemento importante. Os dispositivos IoT normalmente têm poder de processamento limitado, baixa confiabilidade e alta taxa de erros (OLIVEIRA, 2017). Aplicações críticas não devem ficar hospedadas em dispositivos IoT. Esses dispositivos devem enviar suas informações para um elemento centralizador, com maior poder de processamento e disponibilidade. Se esse elemento estiver na nuvem, as aplicações ganham mais flexibilidade e escalabilidade. Além disso, evitam a necessidade de um servidor local de alta disponibilidade (OLIVEIRA, 2017).

Aplicações que acessam dispositivos IoT podem ter várias interfaces para a nuvem: web, banco de dados ou aplicações específicas de automação industrial, comercial ou residencial. Podem incluir aprendizagem de máquina e algoritmos de alta demanda computacional (OLIVEIRA, 2017).

2.1.3 Plataformas Comerciais de Nuvem

A perspectiva de criar máquinas virtuais, instalar serviços diversos incluindo banco de dados, web, DNS e outros, de forma exclusiva e isolada, só é possível nos serviços pagos de nuvem. Há várias plataformas de nuvem disponíveis. Algumas delas já têm interfaces para IoT que possibilitam novas abordagens e novos serviços. Os principais são: a) Microsoft Azure, b) Google Cloud Platform, c) IBM Watson IoT Platform, e d) Amazon Web Services IoT. A maioria deles oferece um nível básico de

serviço gratuito, possivelmente exigindo a inserção dos dados de cartão de crédito e alertando que haverá cobranças caso excedam-se os níveis de serviços gratuitos (OLIVEIRA, 2017).

2.1.4 Datalog na Nuvem

Uma aplicação típica de IoT é funcionar como datalog, cujo objetivo é apenas registrar uma grandeza obtida por um sensor. Esse registro pode ser usado com vários objetivos, desde monitoramento ambiental a aplicações industriais. Também é possível disparar alarmes ou ações baseadas nos valores registrados. Datalog já foi um processo manual, passou a ser automatizado com armazenamento local, depois por um servidor em rede. O momento atual é de enviar o registro para a nuvem. Enviar para a nuvem economiza toda a infraestrutura necessária para o registro local e, de quebra, facilita o acesso remoto. Diversos serviços de registro de informações IoT estão disponíveis; alguns, de forma gratuita. Eles oferecem interfaces elaboradas, incluindo gráficos e aplicativos para acesso em smartphone (OLIVEIRA, 2017).

2.2 TECNOLOGIAS UTILIZADAS

2.2.1 ThingSpeak

ThingSpeak.com é um serviço de IoT na nuvem que possibilita fazer registro de informações recebidas pelos dispositivos IoT, além de disponibilizar diversas análises e ações sobre esses dados. As análises incluem análise de dados por softwares matemáticos, visualização de gráficos indicativos do comportamento dos dados e plug-ins para criar arquivos Linguagem de Marcação de Hipertexto (HTML, do inglês *HyperText Markup Language*), linguagem de script para páginas Web (JavaScript, do inglês *Java Script*) ou Folhas de Estilo em Cascata (CSS, do inglês *Cascading Style Sheets*) personalizados. As ações incluem o envio de tweetters, que podem ser encaminhados para comandar outras plataformas, além de controles temporizados, reações a eventos e comandos via Protocolo de Transferência de Hipertexto (HTTP, do inglês *HyperText Transfer Protocol*) (OLIVEIRA, 2017).

2.2.2 Arduino

O Arduino consiste não somente de um microcontrolador, mas sim de uma plataforma de desenvolvimento de sistemas embarcados de baixo custo aberta e livre. Assim, não está vinculado a nenhum fabricante específico, embora a maioria dos módulos disponíveis utilize microcontroladores da Atmel.

As referências a Arduino consideram, normalmente, uma placa integrada com um microcontrolador e suas interfaces de entrada e saída, alimentação e comunicação. O módulo mais comum é o Arduino Uno, basicamente, um kit de desenvolvimento para iniciantes. Há diversos outros módulos Arduino, entre os quais, o Arduino Nano, que é voltado para a produção de sistemas embarcados, além de diversos outros módulos compostos de sensores, atuadores e comunicação.

O Arduino teve a primazia de tornar o desenvolvimento de sistemas embarcados acessível a toda espécie de curiosos, entusiastas e estudantes de elétrica, eletrônica e computação. Antes do Arduino, desenvolver um sistema embarcado exigia a compra de um kit de desenvolvimento nem sempre barato, com recursos para programação e gravação dos microcontroladores, além da necessidade de construir uma placa para acomodar os componentes. As placas Arduino vêm prontas para utilização e integração e têm baixo custo; o ambiente de desenvolvimento torna a tarefa mais fácil (OLIVEIRA, 2017).

2.2.3 Módulos Periféricos Arduino

A popularização do Arduino trouxe algumas vantagens importantes para o seu desenvolvimento. Vários módulos, conhecidos como shields, começaram a ser produzidos e consumidos em larga escala, tornando-se também acessíveis e muito úteis na integração de sistemas embarcados. Há vários módulos disponíveis, incluindo sensores, atuadores e módulos de comunicação.

Entre os módulos sensores, é possível destacar o módulo sensor de corrente elétrica, módulo de sonar, módulo detector de movimento, módulo sensor de umidade de solo, módulo sensor de temperatura e umidade do ar, módulo sensor de gases inflamáveis, módulo sensor de luminosidade, módulo sensor infravermelho, entre

dezenas de outros módulos extremamente úteis em aplicações de Internet das Coisas.

Entre os módulos atuadores, os mais comuns são os módulos de relé, que podem aparecer em diversas configurações de número de canais, de um a oito canais. Esses módulos são ótimos para controlar diversos tipos de dispositivos com função de liga/desliga, como lâmpadas, motores e aparelhos eletrônicos. Também são comuns os módulos de controle de servomotores, usados especialmente em robótica.

Os módulos de comunicação incluem módulos Wi-Fi, Ethernet e Bluetooth. Esses módulos servem para interligar o Arduino e outros dispositivos em aplicações de Internet das Coisas. Os módulos Wi-Fi e Ethernet são usados para ligar o Arduino à internet. Para isso estão disponíveis diversas bibliotecas com as pilhas de Protocolo de Controle de Transmissão/Protocolo de Internet (TCP/IP, do inglês *Transmission Control Protocol/Internet Protocol*) e Wi-Fi, que possibilitam a interligação nos diversos modos de Wi-Fi, bem como algumas aplicações TCP/IP, como a web. Há bibliotecas para que o Arduino funcione como cliente web, enviando solicitações e acessando dados na web, ou como servidor, disponibilizando acesso às suas informações por meio do protocolo HTTP.

O módulo Bluetooth é particularmente interessante para comunicação direta com aplicativos em smartphones, que normalmente suportam esse protocolo. Isso permite controlar e acessar informações de diversos dispositivos, desde que estejam próximos, visto que o alcance do Bluetooth é da ordem de poucos metros. O Bluetooth também pode ser usado para acessar periféricos de entrada e saída, como teclados, mouse, microfones e alto-falantes. Nesse caso, no entanto, está limitado àqueles periféricos que são suportados pelas bibliotecas do Arduino (OLIVEIRA, 2017).

2.2.4 Arduino Uno R3

2.2.4.1 Microcontrolador

Esta versão do Arduino usa o microcontrolador ATMEGA328P-PU, que é um microcontrolador com arquitetura de 8 bits, possui 32kB de memória flash, 8kB de Memória de Acesso Aleatório (RAM, do inglês *Random Access Memory*) e 1kB de

memória somente leitura programável e apagável eletricamente (EEPROM, do inglês *Electrically-Erasable Programmable Read-Only Memory*).

Nesta versão, ele é encapsulado no formato Pacote Duplo em Linha (DIP, do inglês *Dual In-line Package*) com 28 pinos, exatamente como os outros circuitos integrados que encaixam no protoboard, sendo possível remover o microcontrolador da placa para utilizá-lo em protótipos ou projetos finais, e até mesmo a substituição dele em caso de danos ao componente (NASCIMENTO, 2017).

2.2.4.2 Alimentação

Esta placa possui um conector fêmea do tipo P4 para alimentação, padrão muito comum nas fontes que acompanham equipamentos eletrônicos. Neste plug pode-se ligar tensões de 7V a 12V, pois neste conector está ligado um regulador de tensão que irá reduzir a tensão aplicada para 5V, tensão de operação da placa. Assim como neste conector, pode-se ligar a tensão de 7V a 12V ao pino Vin da placa, que também está ligado ao mesmo regulador de tensão. A placa também pode ser alimentada pelo conector USB, que disponibiliza para a placa a tensão de 5V. Também é disponibilizada a tensão de 3.3V, porém nesta tensão é possível consumir apenas 150mA (NASCIMENTO, 2017).

2.2.4.3 Interface de programação

A programação desta placa pode ser feita tanto pelo conector USB quanto pelo conjunto de 6 pinos próximos ao microcontrolador. Na operação normal da placa, a programação é feita através do conector USB ligado a um computador. Para possibilitar a comunicação, esta placa utiliza um microcontrolador ATMEGA16u2 ou ATMEGA8u2 que converte o protocolo USB em Serial. O driver para o computador é instalado automaticamente junto com o software Arduino IDE.

Os outros pinos podem ser utilizados para a programação utilizando um protocolo chamado Programação Serial em Circuito (ISP, do inglês *In-system Programming*). Normalmente, esses pinos são utilizados para a gravação do bootloader do microcontrolador, pequeno programa que roda na placa antes do código do programador e que é responsável por receber o programa que vem pela porta serial

e gravar na memória do microcontrolador. Quando o bootloader da placa não está gravado, não é possível programar a placa através da porta USB, e neste caso a única forma de realizar a gravação do programa é através deste protocolo ISP. É possível utilizar outra placa da família Arduino como conversor USB-ISP e realizar essa gravação (NASCIMENTO, 2017).

2.2.4.4 Interface com o usuário

São disponibilizados e nomeados diversos pinos para o usuário utilizar a placa. As entradas e saídas digitais são numeradas de 0 a 13, sendo que 6 delas podem ser utilizadas como Modulação por Largura de Pulso (PWM, do inglês *Pulse Width Modulation*). As entradas analógicas estão numeradas de A0 a A5, e além disso as entradas analógicas podem ser usadas também como entradas e saídas digitais. Há também mais um conjunto de pinos referentes à alimentação da placa, possuindo o pino Vin, GND, +5V, +3.3V e o pino Reset.

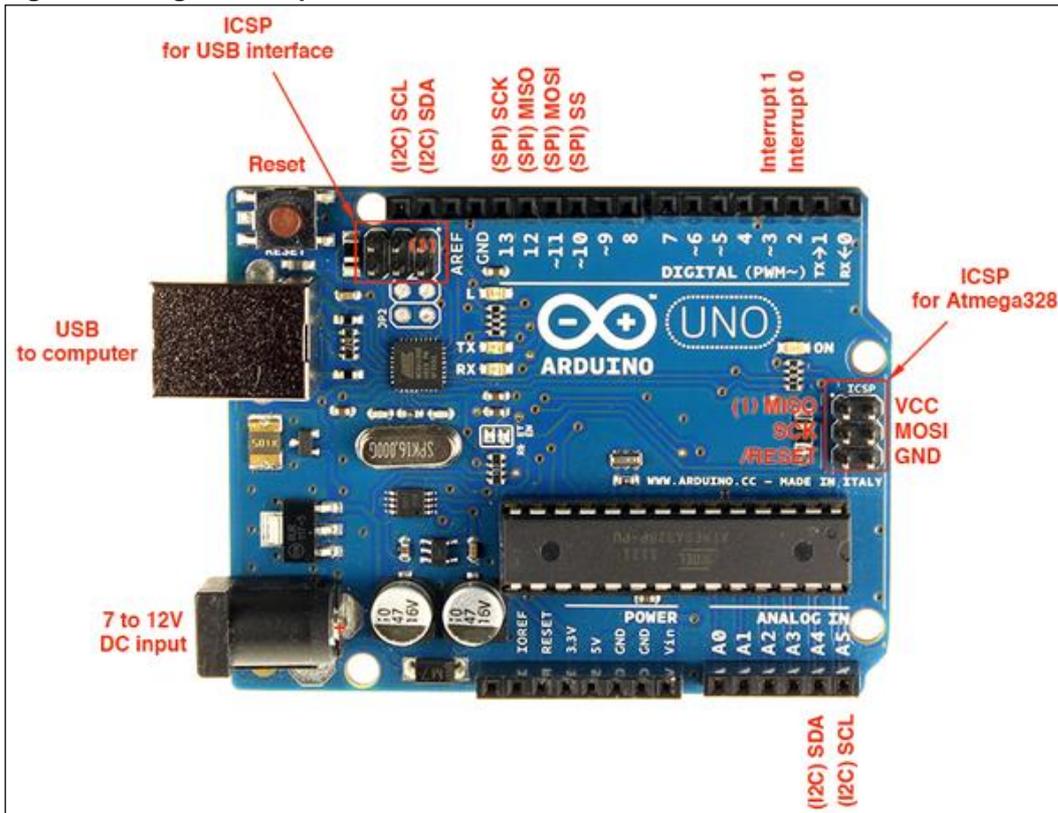
A placa oferece esses pinos de acesso com conectores do tipo fêmea, e as ligações devem ser feitas colocando placas ou jumpers nestes conectores. A disposição dos pinos do Arduino UNO serve de referência para diversos Shields, placas com funcionalidades específicas que se encaixam em cima do Arduino UNO.

Nesta placa também existe um led conectado ao pino digital 13, além de dois leds para a comunicação serial e um led para indicar se a placa está ligada ou não, além de um botão responsável pelo Reset da placa (NASCIMENTO, 2017).

2.2.4.5 Diagrama de pinos

A Figura 2 ilustra o diagrama de pinos da placa Arduino Uno R3.

Figura 2 - Diagrama de pinos do Arduino Uno R3



Fonte: Reali (2012).

2.2.5 Arduino Mega 2560

Esta placa é ideal para quem precisa controlar e ler muitas coisas ao mesmo tempo, pois possui um elevado número de entradas e saídas (NASCIMENTO, 2017).

2.2.5.1 Microcontrolador

Esta placa conta com um microcontrolador ATMEGA2560, que possui 256kB de memória flash (memória que armazena o programa), 8kB de memória RAM e 4kB de memória EEPROM, ou seja, possui uma memória flash 5 vezes maior que o ATMEGA328 e uma memória RAM 4 vezes maior, sendo capaz de receber programas maiores. Além disso, esse microcontrolador é do tipo SMD, ou seja, ele é soldado diretamente na superfície da placa. Possui um número muito superior para entradas e saídas, que será detalhado na interface com o usuário (NASCIMENTO, 2017).

2.2.5.2 Alimentação

Possui o mesmo sistema de alimentação que o Arduino UNO: possui um conector para fonte, além de possuir a entrada USB. Possui regulador de tensão de 5V, porém a tensão de 3.3V disponibilizada para o usuário suporta uma corrente baixa. Pode receber tensões de 7V a 12V no pino Vin ou na conexão para fonte (NASCIMENTO, 2017).

2.2.5.3 Interface de programação

Predominantemente utiliza mesma interface de programação que o Arduino UNO, utilizando o microcontrolador ATMEGA16U2, porém algumas placas podem possuir o chip CH340. A programação é feita através de uma porta USB (NASCIMENTO, 2017).

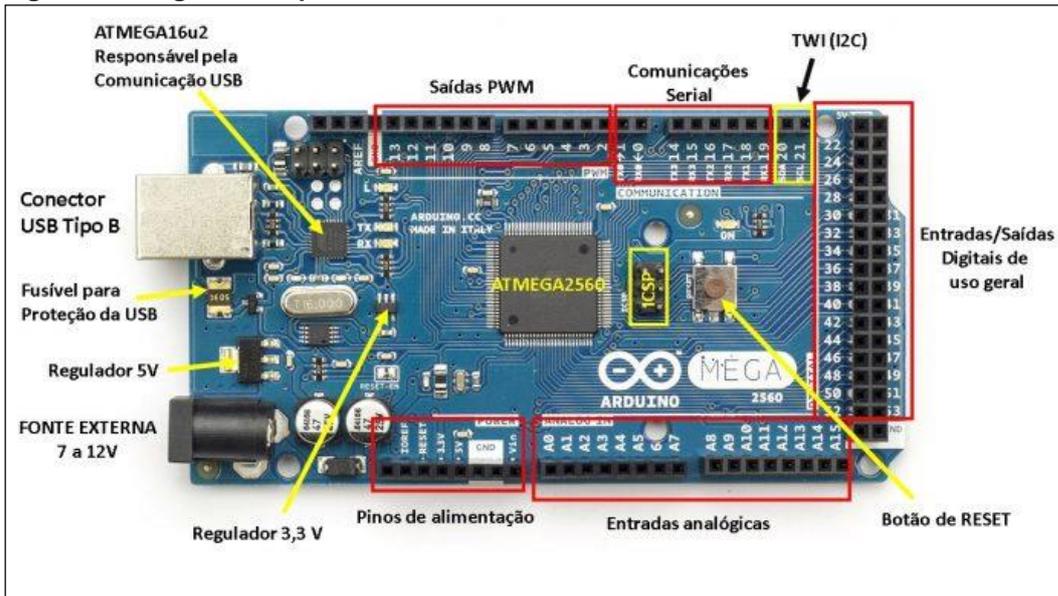
2.2.5.4 Interface com o usuário

O Arduino MEGA possui conectores do tipo fêmea e possui parte de seu layout idêntico ao Arduino UNO. Esta placa possui 16 entradas analógicas, 54 entradas/saídas digitais (sendo que 15 podem ser PWM), sendo todos eles nomeados e numerados, além de mais canais de comunicação com outros dispositivos: são 4 portas Seriais. Possui um conjunto de pinos dedicados à comunicação, juntando as Seriais, o I2C e o SPI (mais protocolos de comunicação) (NASCIMENTO, 2017).

2.2.5.5 Diagrama de pinos

A Figura 3 ilustra o diagrama de pinos da placa Arduino MEGA 2560.

Figura 3 - Diagrama de pinos do Arduino MEGA 2560



Fonte: Souza (2014).

2.2.6 Arduino Pro Mini

Esta é uma placa de baixo custo, mais recomendada para utilização em projetos e protótipos por seu tamanho e preço. Pouco recomendada para o aprendizado devido à dificuldade de programar a placa devido à falta de porta USB (NASCIMENTO, 2017).

2.2.6.1 Microcontrolador

Seguindo na mesma linha, a placa Pro Mini utiliza o ATMEGA328P-AU com as mesmas características de memória e processamento das placas anteriores (NASCIMENTO, 2017).

2.2.6.2 Alimentação

Esta placa pode ser alimentada utilizando o pino RAW, que é o equivalente ao Vin nas outras placas. A tensão de alimentação pode ser de 5V a 12V. Uma diferença desta placa é que não fica disponível ao usuário a tensão de 3.3V (NASCIMENTO, 2017).

2.2.6.3 Interface de programação

Com a finalidade de reduzir a placa e também os custos, nesta placa não existe a comunicação USB diretamente com o computador. Para programar esta placa é necessário um conversor USB-Serial. Esta placa dispõe dos pinos necessários para conexão diretamente com o conversor USB (NASCIMENTO, 2017).

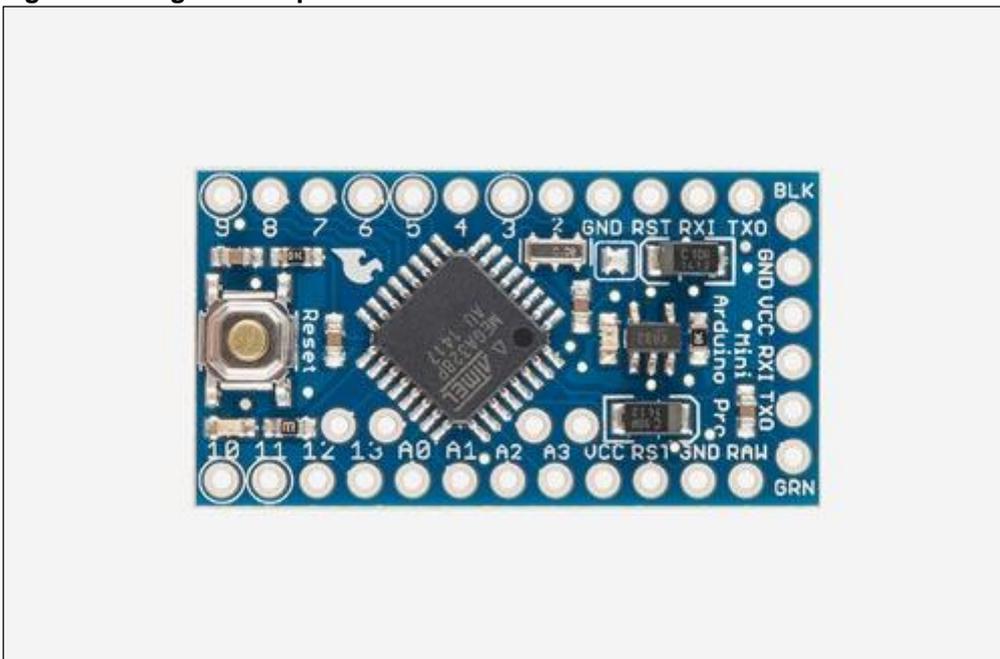
2.2.6.4 Interface com o usuário

Esta placa costuma possuir pinos do tipo macho, sendo encaixável no protoboard quando os pinos A4 e A5 não estão soldados. Além disso, existe a possibilidade de conectar pinos do tipo fêmea, apesar de pouco comum. Existe nela as marcações de cada um dos pinos da placa, facilitando a montagem. Possui uma organização de pinos diferente das outras placas (NASCIMENTO, 2017).

2.2.6.5 Diagrama de pinos

A Figura 4 ilustra o diagrama de pinos da placa Arduino Pro Mini.

Figura 4 - Diagrama de pinos do Arduino Pro Mini



Fonte: Arduino (2018).

2.2.7 Módulo Wi-Fi ESP8266

2.2.7.1 ESP8266

O microcontrolador ESP8266, produzido pela fabricante chinesa Espressif, é um microcontrolador de 32 bits que inclui um núcleo microprocessado Tensilica L106, que funciona na largura de banda de 80 MHz, podendo chegar a 160 MHz. O processamento da pilha de protocolos Wi-Fi usa 20% da capacidade de processamento desse processador. Com isso, 80% dessa capacidade pode ser utilizada em aplicações do usuário. A memória disponível para os dados dos programas tem cerca de 50kB, já descontado o espaço necessário para o padrão Wi-Fi. A memória disponível para o programa principal é de 4MB, em área acessível à atualização em funcionamento, também conhecida como *Over-The-Air* (OTA); além de mais 512kB que não contam com esse recurso e só podem ser atualizados via cabo, em procedimento de atualização.

A tensão nominal de funcionamento é de 3,3 volts. O consumo de energia conta com 20 μ A no modo sleep e cerca de 50mA conectado a um AP Wi-Fi em modo de recepção. Na transmissão em potência máxima de 17dBm, a corrente de consumo pode chegar a 170mA. Considerando uma bateria recarregável de 3,7V e 1.000mAh de capacidade, pode manter o funcionamento do ESP8266 por cerca de vinte horas ininterruptas se não mantiver uma taxa alta de transmissão. É possível, ainda, considerar aplicações nas quais o microcontrolador entra no modo sleep sempre que não há demanda. Isso pode reduzir o tempo de funcionamento para uma pequena parcela do tempo, aumentando consideravelmente o tempo de vida de uma bateria alimentando esse microcontrolador.

Em relação às interfaces de entrada e saída, o ESP8266 tem 17 interfaces GPIO que podem ser configuradas como entradas ou saídas digitais. Tem também quatro interfaces de saída PWM e uma entrada analógica com 10 bits de precisão, por uma tensão de 3,3 volts, que é a tensão de funcionamento do ESP8266.

Como interfaces de comunicação, há interfaces seriais síncronas, SPI, I2C e I2S e assíncronas USART, que podem ser usadas para interligar uma interface RS-232. E, claro, não se pode esquecer da interface Wi-Fi, que pode atuar no modo Ponto de Acesso (AP, do inglês *Access Point*), como cliente, no modo ad hoc ou Wi-Fi Direct.

O ESP8266 tem, ainda, atualização *Over The Air* (OTA), que possibilita que seu firmware seja atualizado pela rede Wi-Fi, em vez de ser atualizado exclusivamente pela porta USB/Serial. Essa característica é muito importante na atualização de software para correção de erros, já depois de implantado. É possível que os dispositivos sejam atualizados periodicamente, assim como ocorre com os sistemas operacionais modernos (OLIVEIRA, 2017).

2.2.7.2 Módulos com o ESP8266

O módulo ESP.01 é o módulo mais simples disponível para implementação de comunicação Wi-Fi em microcontroladores que não possuem essa funcionalidade. Além do microcontrolador ESP8266, ele tem apenas o módulo de memória externa, encontrado em versões de 512kB ou 1MB, suficiente para a maioria das aplicações. Tem apenas oito pinos, que incluem alimentação, comunicação serial UART e alimentação de 3,3V. Para ser gravado, é necessário o uso de adaptadores USB. O ESP-01 é muito usado como módulo de comunicação Wi-Fi para o Arduino e até para o Raspberry Pi, devido ao baixo custo. No entanto ele é capaz de realizar as tarefas de forma independente, visto que também tem processador interno com capacidade de processamento superior à da maioria dos módulos Arduino. É ideal para aplicações mais simples como acender uma lâmpada ou ler um sensor do tipo liga/desliga.

O próximo módulo disponível é o ESP-12, com suas variantes ESP12-E e ESP-12F, cuja diferença se localiza apenas na pinagem. Esse módulo foi construído para ser integrado a uma placa de circuito impresso que pode agregar outros componentes. Vários outros módulos comerciais integram o ESP-12 em seus módulos. Tem 11 pinos GPIO, interfaces seriais SPI, I2C e UART. Contém memória flash de 4MB e possibilita explorar todos os recursos do ESP8266. Devido a aspectos físicos esse módulo não é tão fácil de ser utilizado na prototipação de projetos. Seu uso é mais direcionado na produção em média e larga escala, integrando as placas de circuito impresso que podem incluir outros componentes.

O próximo módulo relacionado utiliza o ESP-12 em uma placa que agrega recursos adicionais, especialmente um conversor USB-serial, para facilitar a gravação do software e um regulador de tensão de 5 volts para 3,3V, o que possibilita ligar os

módulos diretamente em interfaces USB, bem como em fontes de 5V, bem mais comuns que as fontes de 3,3V.

O módulo mais popular é o módulo chamado NodeMCU, que agrega com ele uma plataforma de desenvolvimento em linguagem Lua. Essa linguagem, desenvolvida no Brasil por professores da PUC-Rio, é interpretada e tem uma sintaxe de fácil aprendizagem e integração.

Todos esses módulos podem ser integrados em um módulo Arduino típico, como Arduino Uno, Nano, etc., desde que tenha um módulo adicional de rede Wi-Fi. Esses módulos podem ser facilmente encontrados e já são de domínio de boa parte dos desenvolvedores Arduino. O próprio ESP8266 pode ser usado como interface Wi-Fi no módulo ESP-01. É um claro desperdício de recursos, visto que o ESP8266 dá conta de todo o processamento, mas o Arduino pode ser necessário por causa de suas interfaces e portas (OLIVEIRA, 2017).

2.2.7.3 Considerações sobre o modelo ESP.01

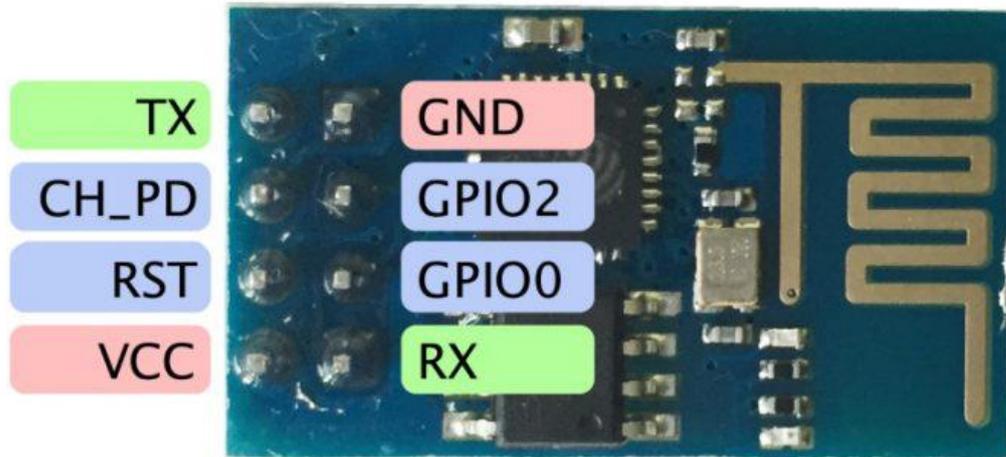
Para se construir um protótipo de rede de sensores sem fio e de baixo custo monetário foi utilizado o módulo ESP8266 ESP.01. Esse modelo da família ESP8266 é muito popular no universo da IoT pois possibilita conexão sem fio que atende às especificações do protocolo wireless 802.11 b/g/n e pode ser acoplado a quaisquer modelos de microcontroladores da família Arduino. O componente funcional principal do módulo é a comunicação Wi-Fi, mas esse módulo também possui portas GPIO que, a princípio, poderiam ser utilizadas para implementar outras funções ao módulo, mas descobriu-se ao longo da pesquisa que essas portas são muito limitadas e que não são tão confiáveis quanto a comunicação wireless propriamente dita.

É importante ressaltar que esse módulo foi utilizado, em um primeiro momento, no seu estado fundamental, da maneira como veio do distribuidor ou fabricante. Aspectos como atualização de firmware para a utilização de bibliotecas mais elaboradas não foram abordados neste trabalho. Para se fazer a abordagem inicial do funcionamento do módulo ESP8666 ESP.01 foi necessário fazer um aparte de como esse módulo trabalha no seu estado funcional mais básico.

2.2.7.4 Diagrama de pinos do ESP8266

A Figura 5 ilustra o modelo ESP-01 com destaque para o seu diagrama de pinos.

Figura 5 - Diagrama de pinos do ESP-01



Fonte: Alves (2016).

A descrição e função dos pinos do ESP-01 são apresentadas no Quadro 1.

Quadro 1 - Descrição e função dos pinos do ESP-01

Pino	Descrição	Função
1	GND	Sinal de Terra GND
2	GPIO2	I/O que pode ser controlada pelo firmware
3	GPIO0	Pode ser controlado pelo firmware e deve ser colocado em nível baixo (GND) para modo update ou em nível alto para operação normal
4	Rx	Sinal de Rx do módulo, a ser conectado no Tx do microcontrolador
5	Tx	Sinal de Tx do módulo, a ser conectado no Rx do microcontrolador
6	CH_PD	Sinal de habilitação do chip (chip enable), usado para gravação de firmware ou atualização. Deve ser mantido em nível ALTO para operação normal
7	RESET	Sinal de Reset/Restart acionado em nível baixo (GND)
8	VCC	Tensão de alimentação 3,3V. Módulo consome até 300mA

Fonte: Curvello (2015).

2.2.7.5 Comandos AT

Comandos AT são a forma básica de configurar e acionar o ESP8266 quando ele está sob controle de um equipamento externo (como um Arduino, por exemplo). Os atuais comandos AT são descendentes diretos do chamado "Padrão Hayes", de

1981, usado para permitir que os computadores pessoais interagissem com conexões telefônicas controlando diretamente um modem. As definições básicas do Hayes (incluindo o prefixo AT, que significava "atenção") estão presentes nas linguagens de comando de muitos aparelhos modernos, incluindo vários periféricos de computadores – e dificilmente alguém que não seja um programador de drivers ou de firmware precisa chegar a conhecê-las, mas quando um Arduino está sendo programado para enviar instruções a um módulo Wi-Fi ESP8266, os usuários estão no papel de programadores de firmware, e é necessário dominar a sintaxe dos comandos AT aceitos pela família ESP (br-arduino.org).

Em Efetividade (2015), tem-se um documento que descreve o conjunto completo de instruções AT para o módulo ESP8266. É a referência oficial publicada pela Espressif, que é o nome corporativo por trás da família ESP. Disponível em: <<https://br-arduino.org/> br-arduino.org>. Acesso em: 07 nov. 2018.

De toda a tábua de comandos AT disponíveis, para as funcionalidades que foram implementadas nesse projeto, foi necessário pesquisar os comandos básicos relacionados no Quadro 2 (em ordem alfabética).

Quadro 2 - Comandos AT básicos

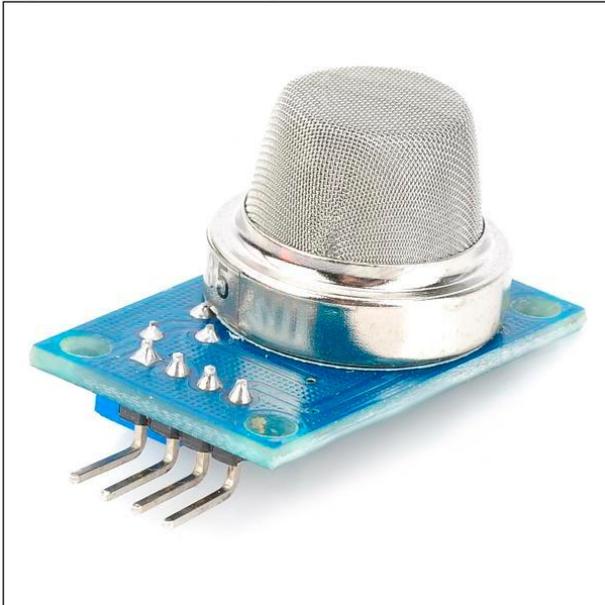
Comando	Descrição
AT+CIPCLOSE	Encerra a conexão TCP ou UDP. Para o modo de conexão múltipla é necessário especificar o “=id” após a instrução
AT+CIPMUX	Ativa/desativa o modo multiplex (até 4 conexões). Para múltiplas conexões é necessário especificar o “=mode” após a instrução
AT+CIPSEND	Define o comprimento da string de dados que será enviada. Para conexão única é necessário especificar o “=length” da string e para múltiplas conexões é necessário especificar o “=id,length” após a instrução
AT+CIPSTART	Inicia uma conexão como cliente. Para conexão única é necessário especificar o “=type,addr,port” após a instrução e para múltiplas conexões é necessário especificar o “=id,type,addr,port” após a instrução
AT+CWJAP?	Imprime os Identificadores do Conjunto de Serviço (SSID, do inglês <i>Service Set Identifier</i>) dos pontos de acesso disponíveis que possivelmente podem ser conectados ao ESP8266
AT+CWJAP=ssid,pwd	Comanda o ESP8266 para se conectar a um SSID com a senha fornecida
AT+CWLAP	Pesquisa pontos de acesso disponíveis com condições específicas
AT+CWMODE=mode	Define o modo como será operado o ESP8266. 1=modo de Estação (cliente). 2=modo Access Point. 3=modo Access Point + Estação
AT+RST	Inicializa/reinicializa o módulo ESP8266

Fonte: Traduzido e adaptado de Efetividade (2015).

2.2.8 Sensor de Gás MQ-2

Para a transformação da concentração, ausência ou presença de gás no meio em sinais elétricos foi utilizado o sensor de Gás MQ-2, conforme ilustra a Figura 6.

Figura 6 - Sensor de gás MQ-2



Fonte: FilipeFlop (2018).

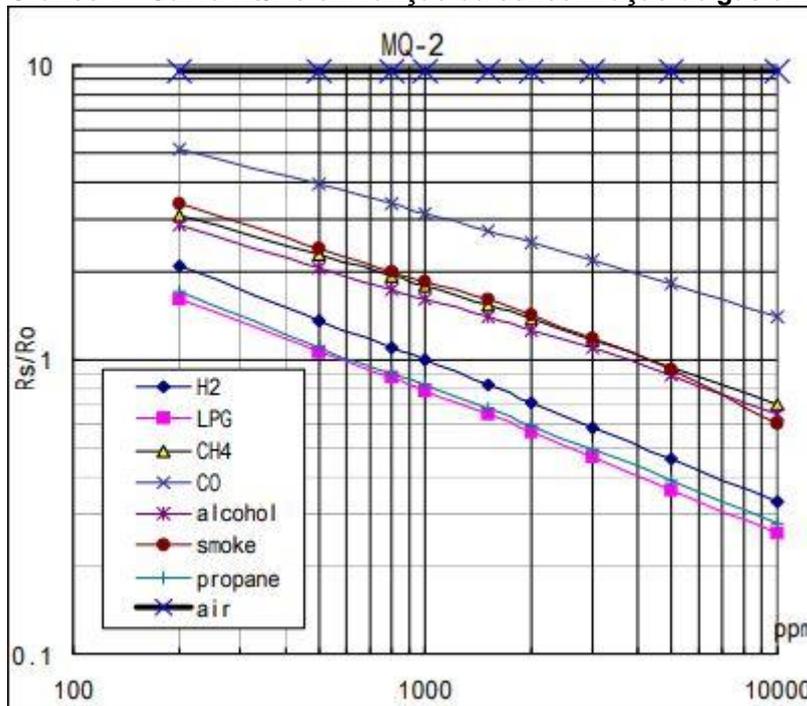
Esse sensor é fabricado pela empresa chinesa Hanwei e possui características práticas que o fizeram muito popular no universo de aferição de concentração de gás em protótipos e soluções domésticas, a saber:

- Detecção de gases inflamáveis: GLP, Metano, Propano, Butano, Hidrogênio, Álcool, Gás Natural e outros inflamáveis;
- Detecção de fumaça;
- Concentração de detecção: 300-10.000ppm;
- Tensão de operação: 5V;
- Sensibilidade ajustável via potenciômetro;
- Saída Digital e Analógica;
- Fácil instalação;
- Comparador LM393;
- Led indicador para tensão;
- Led indicador para saída digital;
- Dimensões: 32 x 20 x 15mm.

2.2.8.1 Cálculo da inclinação da curva LPG

O datasheet do sensor MQ-2 contém a curva R_s/R_o em função da concentração de gás em ppm (FILIPEFLOP, 2018), ilustrada no Gráfico 1.

Gráfico 1 - Curva R_s/R_o em função da concentração de gás em ppm



Fonte: Sampaio (2016).

Tomando-se o ponto extremo superior da curva LPG do Gráfico 1, pode-se estabelecer as coordenadas de p_1 no plano cartesiano (SAMPAIO, 2016):

$$p_1: (x_1, y_1): (\log 200, \log 1.7): (2.3, 0.23)$$

Tomando-se o ponto extremo inferior da curva LPG do Gráfico 1, pode-se estabelecer as coordenadas de p_2 no plano cartesiano:

$$p_2: (x_2, y_2): (\log 10000, \log 0.27): (4, -0.57)$$

Dessa maneira é possível calcular a inclinação da curva LPG no Gráfico 1:

$$\text{inclinação} = (y_2 - y_1)/(x_2 - x_1) = (-0.57 - 0.23)/(4 - 2.3) = -0.47$$

3 PROJETO

Este capítulo descreve e detalha o processo de construção de todos os blocos que constituem o protótipo da plataforma. São apresentados também os diagramas dos circuitos que foram montados no trabalho. Em cada etapa são apresentadas as entradas, saídas e os testes de cada bloco separadamente.

3.1 RECURSOS UTILIZADOS

Nesta seção são expostos e detalhados os recursos utilizados no projeto, os componentes necessários e os gastos com a construção do protótipo.

3.1.1 Componentes

Para a execução do projeto foram necessários os seguintes componentes:

- Arduino Mega 2560 - 1 unidade;
- Arduino Pro Mini - 3 unidades;
- Arduino Uno R3 - 1 unidade;
- Computador pessoal com Windows 10 - 1 unidade;
- Conta free registrada no ThingSpeak - 1 unidade;
- ESP8266 - Modelo ESP.01 - 3 unidades;
- Fonte 12V/1A - 2 unidades;
- Led verde - 3 unidades;
- Led vermelho - 3 unidades;
- Link ADSL com modem/roteador - 1 unidade;
- Placa FTDI 232 - 1 unidade;
- Protoboard - 3 unidades;
- Resistor de 1k Ω - 9 unidades;
- Resistor de 220 Ω - 6 unidades;
- Sensor de gás MQ-2 - 3 unidades;
- Shield fonte ajustável protoboard 3,3V - 5V - 3 unidades.

Além disso, um computador pessoal, os protoboards e as fontes foram utilizados ao longo de todo o projeto. O Arduino Uno R3 e Arduino MEGA 2560 foram utilizados na fase inicial de prototipação e os Arduino Pro Mini foram utilizados para a implementação dos “nós” completos. A placa FDTI 232 foi utilizada para gravar o código do programa, via IDE Arduino, nas placas Arduino Pro Mini. O bloco descrito como “nó” foi formado pela interligação dos componentes sensor de gás MQ-2, Arduino Pro Mini, ESP8266, leds e resistores. O sensor de gás faz a leitura da concentração de GLP no meio, o Arduino faz o processamento e formatação das informações e transmite via ESP8266 para o bloco “concentrador”. O bloco descrito como “concentrador” é um modem/roteador ADSL que roteia as informações para a Cloud IoT ThingSpeak.

3.1.2 Gastos com o Protótipo

A Tabela 1 exibe um demonstrativo do cálculo do preço total de cada nó do protótipo. Os componentes foram todos comprados via loja virtual, disponível em: <<https://www.filipeflop.com>>, acesso em: 07 nov. 2018.

Tabela 1 - Preço total de cada nó do protótipo

Item	Quantidade	Componente	Preço Unitário (R\$)
1	1	Arduino Pro Mini	22,9
2	1	ESP8266 – Modelo E.01	26,9
3	1	Sensor de gás MQ-2	14,9
Preço total			64,7

Fonte: Autoria própria.

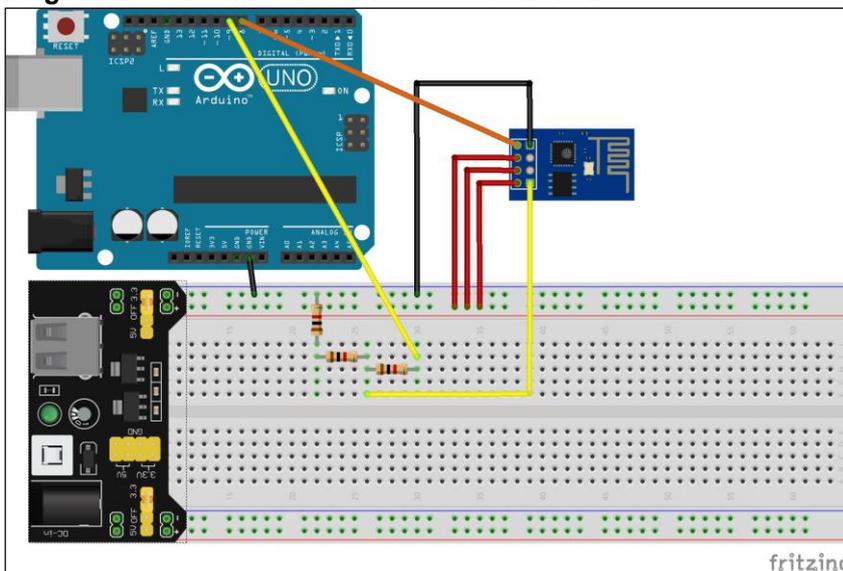
3.2 DIAGRAMAS

No decorrer do projeto foram construídos os circuitos ilustrados nos Diagramas de 1 a 6.

3.2.1 Diagrama 1: Arduino Uno R3 Conectado ao ESP8266

O Diagrama 1 ilustra o circuito utilizado na fase inicial do projeto, basicamente para experiências com comandos AT. O pino 1 (GND) do ESP8266 foi ligado ao GND da fonte de 3.3V. Os pinos 6 (CH_PD), 7 (RST) e 8 (Vcc) do ESP8266 foram ligados ao Vcc da fonte de 3.3V. O pino 5 (Tx) do ESP8266 foi ligado diretamente ao pino 8 (porta de uso comum) do Arduino Uno R3. O pino 4 (Rx) do ESP8266 foi ligado na saída do divisor de tensão formado pelos três resistores de 10kΩ. Esse divisor de tensão tem a função de acoplar a tensão de saída do pino 9 (funcionando como Tx) do Arduino Uno R3 ao pino 4 (Rx) do ESP8266. O GND do Arduino Uno R3 foi ligado diretamente no GND da fonte de 3.3V.

Diagrama 1 - Arduino Uno conectado ao ESP8266



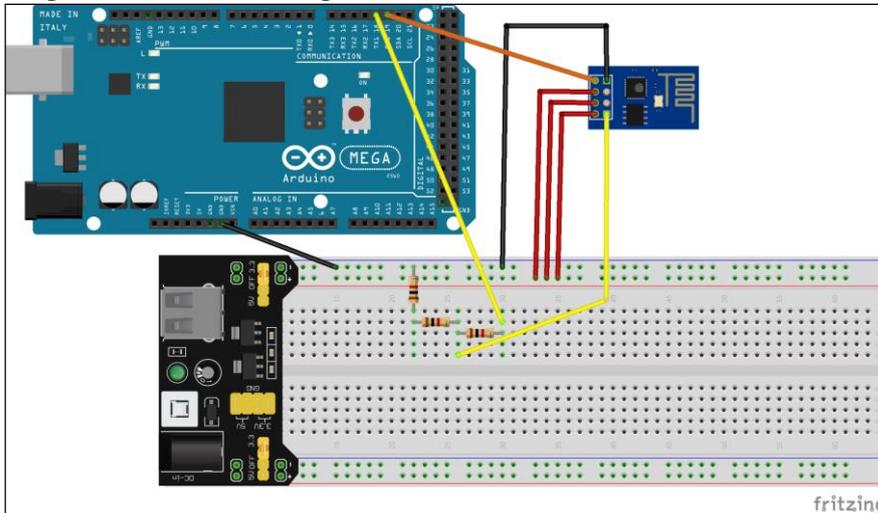
Fonte: Autoria própria, utilizando o software Fritzing.

3.2.2 Diagrama 2: Arduino Mega 2560 Conectado ao ESP8266

O Diagrama 2 ilustra o circuito utilizado na fase inicial de prototipação, para experiências com comandos AT e na comunicação do conjunto Arduino/ESP8266 com o ThingSpeak. O pino 1 (GND) do ESP8266 foi ligado ao GND da fonte de 3.3V. Os pinos 6 (CH_PD), 7 (RST) e 8 (Vcc) do ESP8266 foram ligados ao Vcc da fonte de 3.3V. O pino 5 (Tx) do ESP8266 foi ligado diretamente ao pino 19 (Rx1) do Arduino Mega 2560. O pino 4 (Rx) do ESP8266 foi ligado na saída do divisor de tensão formado pelos três resistores de 10kΩ. Esse divisor de tensão tem a função de acoplar

a tensão de saída do pino 18 (Tx1) do Arduino Mega 2560 ao pino 4 (Rx) do ESP8266. O GND do Arduino Mega 2560 foi ligado diretamente no GND da fonte de 3.3V.

Diagrama 2 - Arduino Mega 2560 conectado ao ESP8266

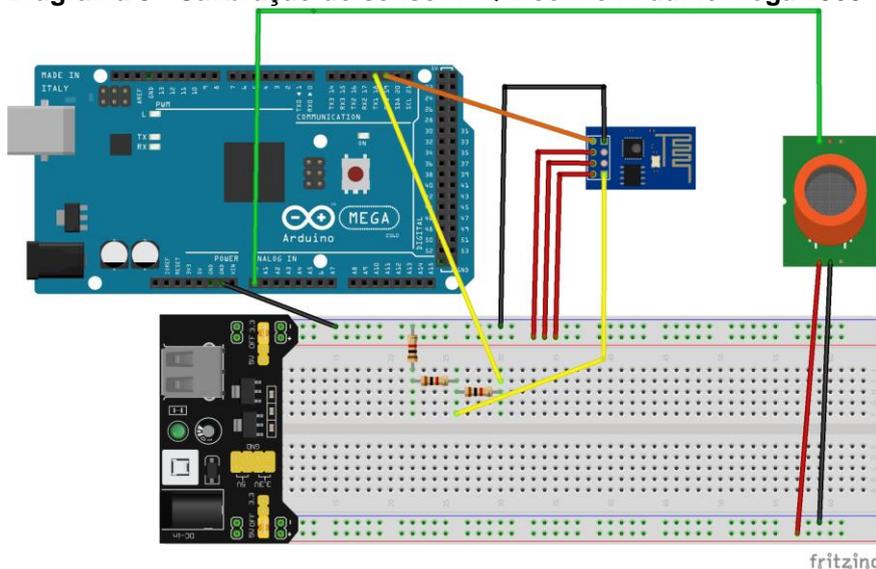


Fonte: Autoria própria, utilizando o software Fritzing.

3.2.3 Diagrama 3: Arduino Mega 2560, ESP8266 e Sensor MQ-2

O Diagrama 3 ilustra o circuito utilizado para a fase de calibração do sensor de gás MQ-2. É composto pelo circuito do Diagrama 2, exposto e detalhado no item 3.2.2, e pelo sensor de gás MQ-2. O pino 1 (Vcc) do sensor MQ-2 foi ligado no Vcc da fonte de 5V, o pino 2 (GND) foi ligado no GND da fonte de 5V e o pino 4 (A0) foi ligado ao pino A0 (entrada analógica) do Arduino Mega 2560.

Diagrama 3 - Calibração do sensor MQ-2 com o Arduino Mega 2560

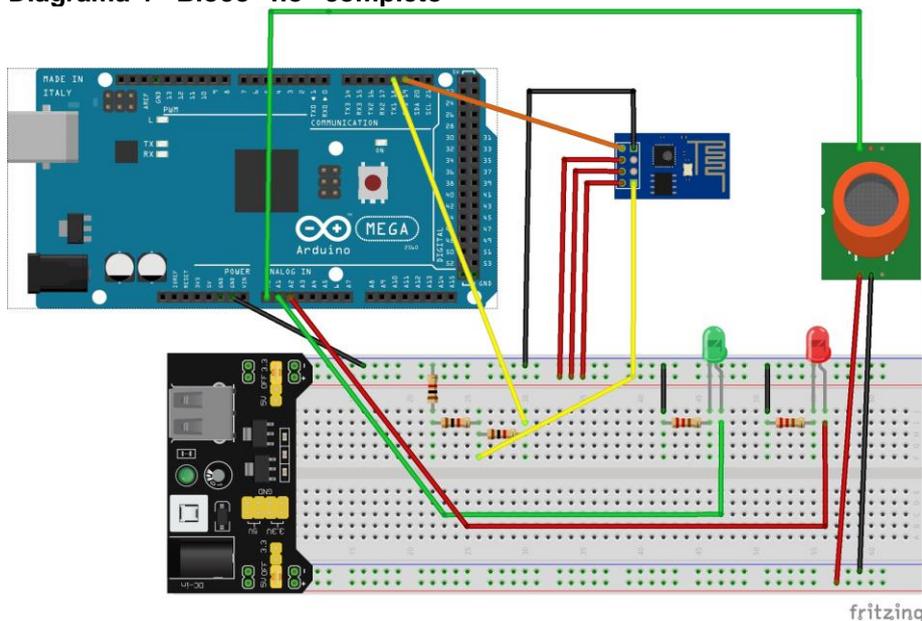


Fonte: Autoria própria, utilizando o software Fritzing.

3.2.4 Diagrama 4: Bloco Nó Completo

O Diagrama 4 ilustra o circuito utilizado para construção do protótipo completo utilizando o Arduino Mega 2560. É composto pelo circuito do Diagrama 3, exposto e detalhado nos tópicos 3.2.2 e 3.2.3 deste trabalho, e pelos leds verde e vermelho. O led verde foi ligado à porta analógica A1 e o led vermelho foi ligado à porta analógica A2 do Arduino Mega 2560. Esses leds têm a função de indicar a ausência de uma determinada concentração de gás no meio (led verde) ou a presença dessa concentração de gás no meio (led vermelho). No ar limpo o led verde fica aceso. O contato do sensor com o nível de concentração de gás determinado apaga o led verde e acende o led vermelho. O nível crítico que comuta do led verde para o led vermelho foi determinado na fase de escrita do código de programação.

Diagrama 4 - Bloco "nó" completo

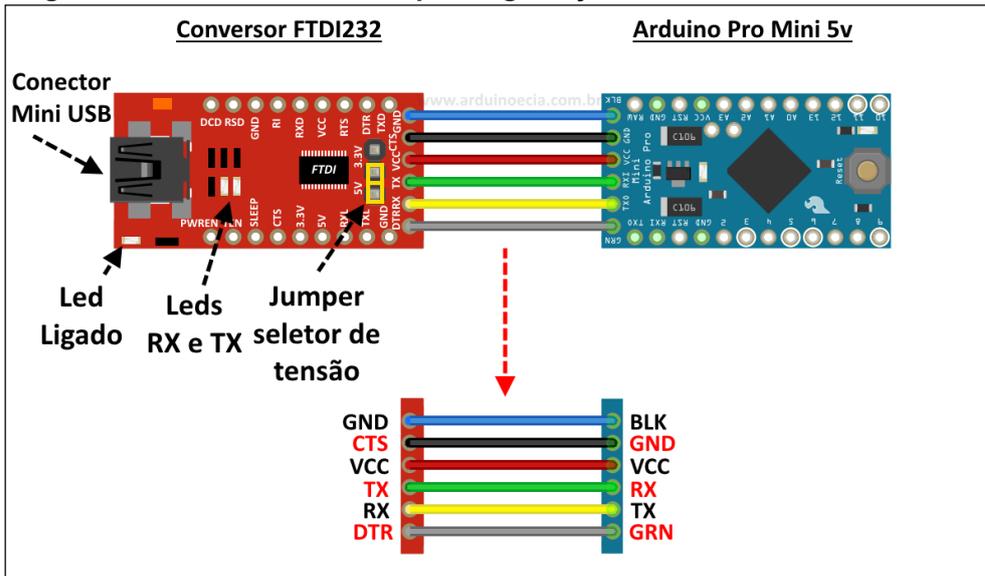


Fonte: Autoria própria, utilizando o software Fritzing.

3.2.5 Diagrama 5: Circuito de Gravação do Arduino Pro Mini

O Diagrama 5 ilustra o circuito utilizado para a gravação do Arduino Pro Mini através do conversor FTDI232. O conversor é ligado pino a pino, diretamente no Arduino Pro Mini, conforme ilustrado no diagrama 5. O conector Mini USB do conversor FTDI232 foi ligado diretamente ao computador através do cabo USB e utilizado para programar o Arduino Pro Mini através da IDE Arduino.

Diagrama 5 - Hardware utilizado para a gravação do Arduino Pro Mini

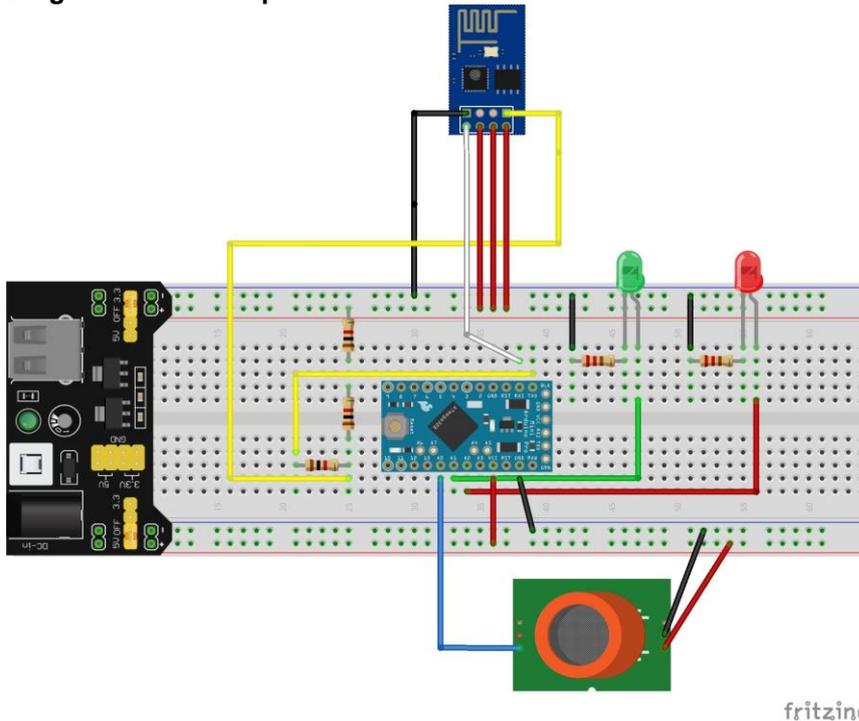


Fonte: Arduino (2014).

3.2.6 Diagrama 6: Protótipo do Bloco Nó com Arduino Pro Mini

O Diagrama 6 ilustra, basicamente, o mesmo arranjo exposto e detalhado no Diagrama 4, tópico 3.2.4, com a substituição pino a pino do Arduino Mega 2560 pelo Arduino Pro Mini. Esse foi o circuito final do projeto e foram produzidas três unidades do mesmo.

Diagrama 6 - Protótipo do bloco "nó" com Arduino Pro Mini



Fonte: Autoria própria, utilizando o software Fritzing.

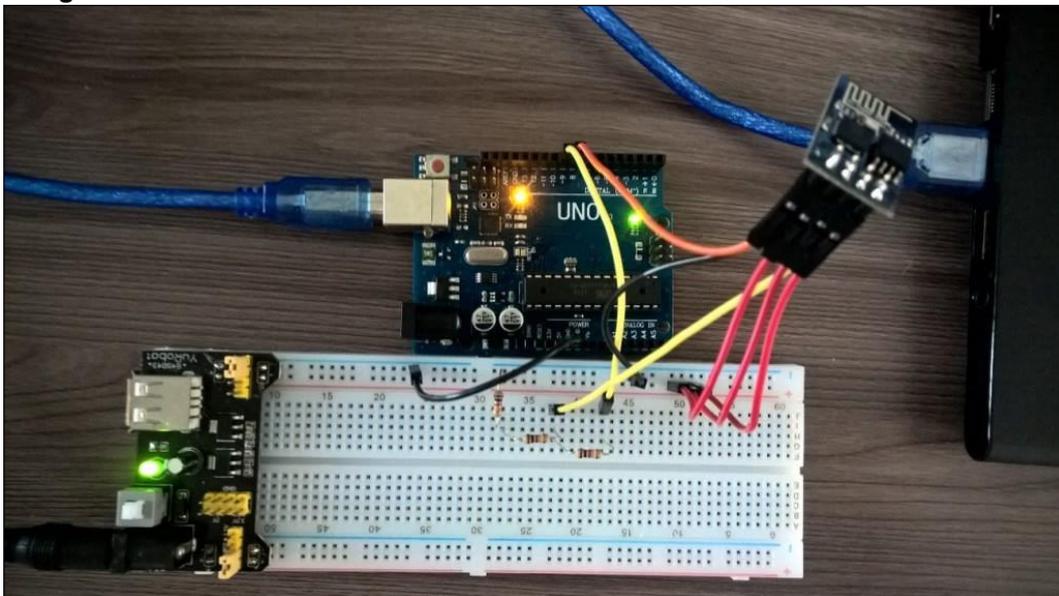
3.3 PROTOTIPAGEM

3.3.1 Primeiro Protótipo: Interface Arduino Uno R3 - ESP8266

O processo de depuração do código no Arduino Pro Mini é complexo pois esse módulo oferece somente um canal de comunicação serial, sem a possibilidade de visualização do funcionamento do código em tempo real de depuração/compilação. A metodologia de gravação do código de programação também é mais complexa. Sendo assim foi pesquisado um módulo da família Arduino um pouco mais prático em termos de desenvolvimento de protótipos, o Arduino Uno R3. Este módulo também apresenta somente um canal de comunicação serial mas, através da utilização da biblioteca “AltSoftSerial.h”, pode-se emular um canal a mais de comunicação serial através de portas comuns do módulo.

Para se fazer os testes com os comandos AT no módulo ESP8266 foi executada a montagem do circuito do Diagrama 1, exposto e detalhado no tópico 3.2.1 deste trabalho. O circuito real montado é apresentado na Fotografia 1.

Fotografia 1 - Circuito Arduino Uno conectado ao ESP8266



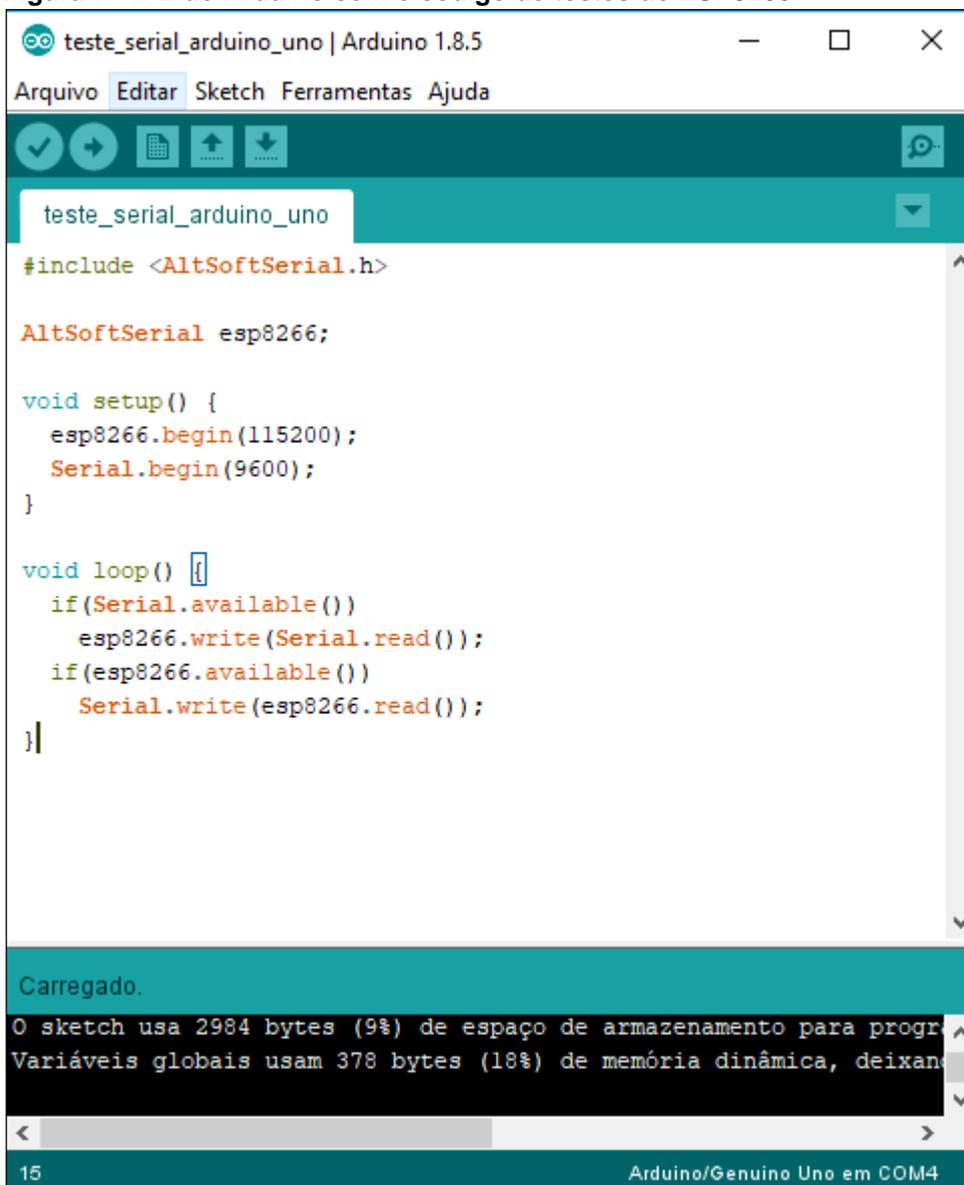
Fonte: Autoria própria.

A construção desse circuito foi o passo inicial para a construção do bloco denominado “nó”. É necessário aqui fazer-se um aparte com relação ao divisor de tensão que foi utilizado para a ligação entre o Tx do Arduino e o Rx do ESP8266. O Arduino primariamente trabalha com 5V de tensão em seus pinos e o ESP8266

trabalha com 3,3V. Dessa maneira foi necessário o divisor de tensão como intermediário. Para a construção do protótipo esse artifício basta, mas em um projeto profissional é necessário um tratamento mais refinado na adequação dessas tensões, através de reguladores de tensão em chips dedicados a essa tarefa.

Após a montagem do circuito do Diagrama 1 e/ou Fotografia 1, foi escrito um código simples na IDE do Arduino para que se pudesse visualizar os comandos AT serem rodados no ESP8266 e ao mesmo tempo todo o processo ser visualizado no Monitor serial, conforme ilustram as Figuras 7 e 8.

Figura 7 - IDE do Arduino com o código de testes do ESP8266



```
teste_serial_arduino_uno | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda

teste_serial_arduino_uno

#include <AltSoftSerial.h>

AltSoftSerial esp8266;

void setup() {
  esp8266.begin(115200);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available())
    esp8266.write(Serial.read());
  if(esp8266.available())
    Serial.write(esp8266.read());
}

Carregado.
O sketch usa 2984 bytes (9%) de espaço de armazenamento para progr
Variáveis globais usam 378 bytes (18%) de memória dinâmica, deixan

15 Arduino/Genuíno Uno em COM4
```

Fonte: Autoria própria.

Figura 8 - Monitor serial do Arduino Uno R3

Fonte: Autoria própria.

Esse código utiliza a serial Arduino para enviar os comandos AT para a serial do computador e emula por software (biblioteca “AltSoftSerial.h”) outra serial utilizando-se pinos comuns do Arduino Uno R3 para enviar os mesmos comandos para o módulo ESP8266. Dessa maneira foi possível gravar o código de maneira automática no microcontrolador e fazer a depuração.

Após diversas tentativas sem sucesso com diversos comandos AT, chegou-se à conclusão de que o Arduino Uno R3 não é adequado à prototipação de circuitos que utilizam o ESP8266 nas suas versões mais modernas. Após vasta pesquisa na internet, descobriu-se que os módulos ESP8266 ESP.01 vendidos atualmente são configurados na fábrica com uma taxa padrão de 115200 baud. Isso gerou um problema na comunicação. A serial emulada por software não trabalha adequadamente com taxas acima de 9600 baud. Isso gerou uma série de experiências com poucos acertos.

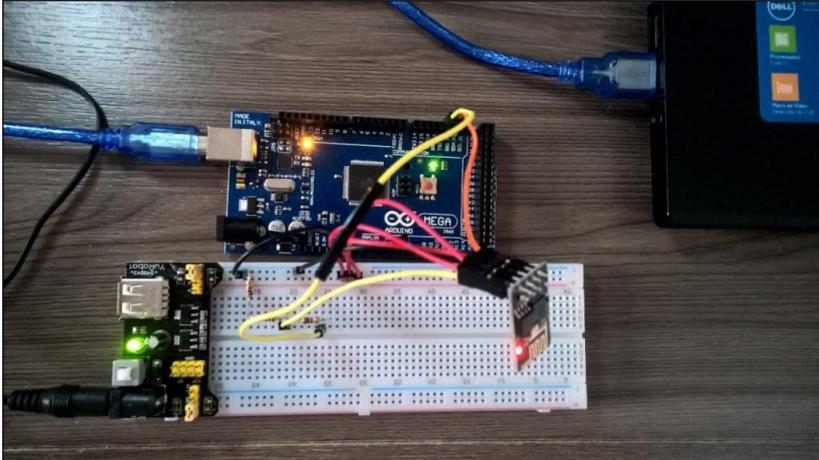
Sendo assim concluiu-se que o desenvolvimento desse protótipo só seria conseguido se fosse utilizado um módulo mais completo que o Arduino Uno R3.

3.3.2 Segundo Protótipo: Interface Arduino Mega 2560 - ESP8266

O próximo na lista da família Arduino, de fácil aquisição, e que conseguiu chegar a um bom termo no desenvolvimento do protótipo foi o Arduino Mega 2560. Esse modelo de shield do Arduino possui quatro seriais implementadas por hardware, o que tornou a escolha dele bastante confortável para a prototipação.

Sendo assim foi montado o circuito do Diagrama 2, exposto e detalhado no tópico 3.2.2 deste trabalho. O circuito real montado é apresentado na Fotografia 2.

Fotografia 2 - Circuito Arduino Mega 2560 conectado ao ESP8266



Fonte: Autoria própria.

Basicamente o circuito é o mesmo que o do Arduino Uno R3, com a exceção de que o par Tx/Rx é ligado à serial 1 do Arduino Mega 2560 e não em portas comuns como foi feito no Arduino Uno R3.

Foi necessária a supressão da biblioteca <AltSoftSerial.h> do código e a troca da denominação das seriais, ficando o código como o ilustrado na Figura 9.

Figura 9 - IDE do Arduino com o código de testes do ESP8266 no Arduino Mega 2560

```

teste_serial_arduino_mega | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda

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

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

Carregado.
O sketch usa 2056 bytes (0%) de espaço de armazenamento para progr
Variáveis globais usam 341 bytes (4%) de memória dinâmica, deixand

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) em COM5

```

Fonte: Autoria própria, print da tela da IDE Arduino.

Essa alternativa de se utilizar o Arduino Mega 2560 na confecção do protótipo tornou o projeto rápido e dinâmico. Através da utilização dos comandos básicos AT pela plataforma Arduino foi possível estabelecer conexão do módulo ESP8266 com o modem ADSL e enviar informações básicas através da rede. Consultando-se a documentação do ESP8266 ESP.01 e do ThingSpeak, chegou-se a um conjunto mínimo necessário de instruções para se enviar um dado do Arduino para o dashboard do ThingSpeak. Foi necessário inicializar o ESP8266, determinar o modo como seria operado, estabelecer conexão com o modem/roteador ADSL, especificar o modo multiplex, estabelecer conexão com o servidor do ThingSpeak, determinar o tamanho da string a ser enviada e enviar a string propriamente dita.

Essa sequência de passos foi conseguida utilizando-se a sequência de comandos AT no Monitor serial da IDE Arduino, especificados no Quadro 3.

Quadro 3 - Sequência de comandos para comunicação entre Arduino e ThingSpeak

Passo	Comando
1	AT+RST
2	AT+CWMODE=3
3	AT+CWJAP="VIVO-5499","1613011747"
4	AT+CIPMUX=0
5	AT+CIPSTART="TCP","184.106.153.149",80
6	AT+CIPSEND=75
7	<pre>GET https://api.thingspeak.com/update?api_key=P2SEXL8UPEPD2URM&field1=345 POST /update HTTP/1.1\r\n Host: api.thingspeak.com\r\n Connection: close\r\n X-THINGSPEAKAPIKEY: "P2SEXL8UPEPD2URM"\r\n Content-Type: application/x-www-form-urlencoded\r\n Content-Length: 39\r\n \r\n\r\n P2SEXL8UPEPD2URM&field1=345\r\n\r\n\r\n\r\n</pre>

Fonte: Autoria própria.

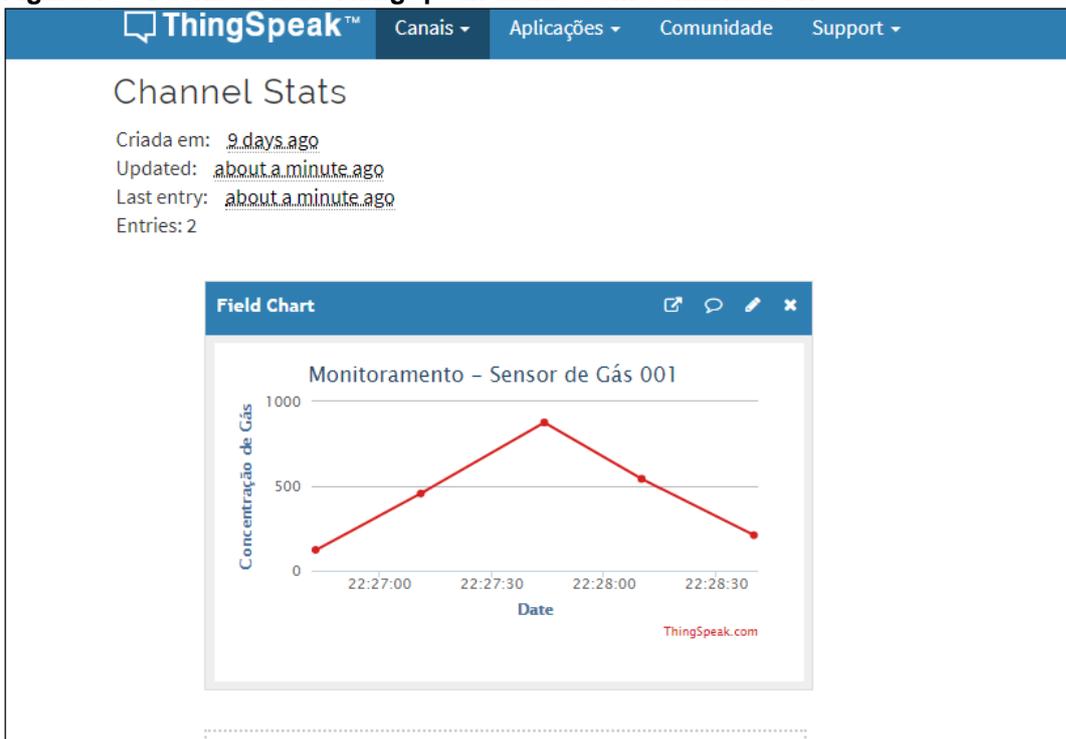
Ainda no Quadro 3, a finalidade dos comandos são:

- Comando 1: responsável por fazer a inicialização do módulo ESP8266 ESP.01;
- Comando 2: define o modo como será operado o ESP8266. 3 = modo Access Point + Estação;
- Comando 3: comanda o ESP8266 para se conectar a um SSID com a senha fornecida;
- Comando 4: ativa/desativa o modo multiplex;
- Comando 5: inicia uma conexão como cliente. TCP na camada de transporte, endereço do servidor do ThingSpeak e porta;

- Comando 6: define o comprimento da string de dados que será enviada. O tamanho da string propriamente dita pôde ser calculado através do site <http://string-functions.com/length.aspx>. Nesse caso chegou-se ao valor 73. Foi necessário adicionar “2” a esse valor por conta dos caracteres [newline] e [return];
- Comando 7: string que foi enviada para o servidor do ThingSpeak. Essa string contém o endereço do servidor do ThingSpeak, a chave de escrita e o valor da concentração de gás a ser impressa no dashboard do ThingSpeak. Foi utilizado o valor genérico e aleatório “345” de concentração de gás para fins de teste do bloco. Nessa etapa foi necessário determinar qual dos métodos HTTP, GET ou POST, seriam utilizados no projeto. Devido ao tamanho do cabeçalho da requisição POST, o tempo de envio das informações foi maior quando comparado com o método GET. De qualquer maneira os dois métodos poderiam ser utilizados, por conta da limitação de update da ferramenta ThingSpeak, que é de 15 segundos. Desta maneira foi determinada a utilização do método POST, pois não tem restrições quanto ao tipo e nem ao tamanho da requisição.

Após os testes através dos comandos AT, utilizando-se valores aleatórios para a concentração de gás, foi obtido o resultado da Figura 10.

Figura 10 - Dashboard do ThingSpeak obtido com comandos AT



Fonte: Autoria própria.

MQ e também teve que fazer a aferição do sensor antes de poder utilizá-lo. Essa parte do presente trabalho foi adaptada com base no artigo referenciado.

R_s/R_o é o produto entre a resistência do sensor no momento da medição pela resistência no ar livre de gases. Para serem obtidas medições reais, foi necessário tomar dois pontos da curva denominada Gás Liquefeito de Petróleo (LPG, do inglês *Liquefied Petroleum Gas*) no Gráfico 1, exposto e detalhado no tópico 2.2.8.1 deste trabalho, e estabelecer o coeficiente de inclinação da curva.

Tomando-se o ponto p1, ponto p2 e a inclinação da curva, foi possível estabelecer o vetor que representa a curva LPG no Gráfico 1, exposto e detalhado no tópico 2.2.8.1 deste trabalho:

$$vetor = \{x, y, inclinação\} = \{2.3, 0.23, -0.47\}$$

Para medir a concentração de gás no ambiente, em ppm, foram necessários o R_s/R_o da medição instantânea, o vetor levantado e relacioná-los conforme a fórmula:

$$10^{(((\log(R_s/R_o) - y_1)/inclinação) + x_1)} \text{ [ppm]}$$

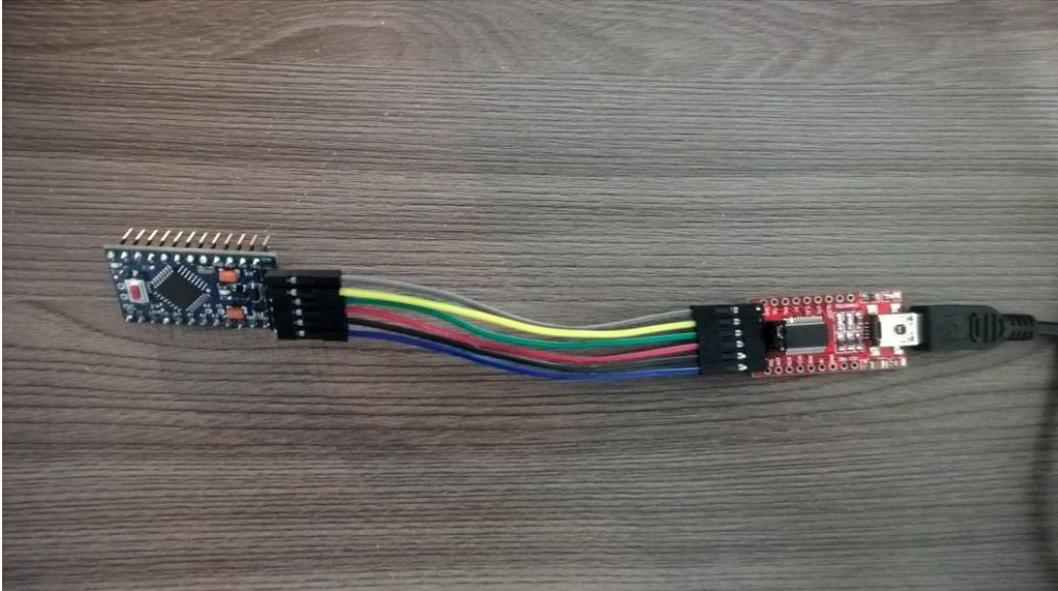
Estabelecida essa metodologia, foi possível adaptar o código descrito no artigo para a utilização no presente trabalho. É importante aqui que algumas observações sobre esse sensor sejam feitas. Para a construção do protótipo foi utilizado o sensor na forma original, ou seja, nas condições em que veio do distribuidor ou fabricante. Não foram observados aspectos de calibração e atendimento às normas de segurança vigentes.

Para aplicações reais, com sensores comerciais mais robustos, existem normas como a NBR IEC 60079-29-2 – Detecção de gases em atmosferas explosivas e a NR33 – Norma trabalhista que regulamenta o trabalho em espaços confinados, que estabelecem critérios de uso e calibração.

3.3.5.1 Circuito de gravação do Arduino Pro Mini

A gravação do Arduino Pro Mini pôde ser feita através de um conversor RS232-TTL, como o mostrado no Diagrama 5, exposto e detalhado no tópico 3.2.5 deste trabalho. O circuito real montado é apresentado na Fotografia 5.

Fotografia 5 - Circuito utilizado para a gravação do Arduino Pro Mini



Fonte: Autoria própria.

3.3.5.2 Gravação do Arduino Pro Mini

Nesta etapa do trabalho é necessário fazer algumas observações. No decorrer da pesquisa foi descoberto que existe uma diferença estrutural entre a entrada analógica do Arduino Pro Mini com relação ao Arduino Mega 2560, que diz respeito ao intervalo de tensões nos quais trabalham. Enquanto o Arduino Mega 2560 varia a entrada analógica entre 0 e 5V, o Arduino Pro Mini varia entre 0 e 7V.

Essa informação foi crucial para o transporte e adaptação do código de um microcontrolador para o outro e obter a equivalência entre as medições em uma plataforma e outra. Vale ainda ressaltar que o código de programação do Arduino já foi elaborado pensando-se nessas necessárias adaptações.

Feitas as considerações do método de gravação e ajustes de tensões nas portas analógicas, foi possível gravar os três blocos protótipos através da IDE Arduino, conforme ilustra a Figura 11.

Figura 11 - IDE de gravação do Arduino Pro Mini

```

sketch_jun18a | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda
Carregar
sketch_jun18a §
//=====
#define PIN_MQ2 A0 //define pin sensor MQ-2
#define VRL_VALOR 5 //resistência de carga
#define RO_FATOR_AR_LIMPO 9.83 //resistência do sensor em ar limpo
#define ITERACOES_CALIBRACAO 50 //numero de leituras para calibrar
#define ITERACOES_LEITURA 5 //numero de leituras para análise
#define GAS_LPG 0
float LPGCurve[3] = {2.3,0.20,-0.47}; //curva LPG aproximada k
//pl: (log200, log1.6),
//inclinacao = (Y2-Y1)/X
//vetor={x, y, inclinacao}

float Ro = 10;
//=====
String apiKey = "1GD6BAOX8F8PXRUL"; // Edit this API key according to your account
String Host_Name = "VIVO-5023"; // Edit Host_Name
String Password = "7048194320"; // Edit Password

//int analog_pin=A0;

```

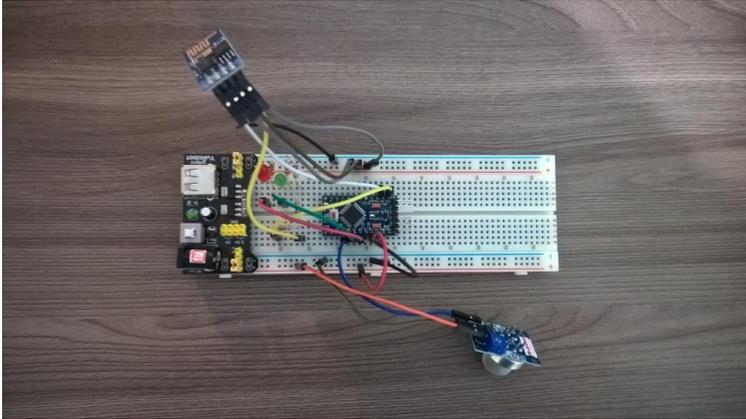
164 Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz) em COM5

Fonte: Autoria própria.

3.3.5.3 Quarto protótipo: Arduino Pro Mini

Foram construídos três protótipos de “nó” completo, conforme ilustra o Diagrama 6, exposto e detalhado no tópico 3.2.6 deste trabalho. Um dos circuitos reais montados é apresentado na Fotografia 6.

Fotografia 6 - Bloco "nó" construído com Arduino Pro Mini



Fonte: Autoria própria.

3.4 CONCENTRADOR

O bloco denominado “Concentrador” constitui-se de um link ADSL doméstico com roteador/modem que provê acesso à internet.

3.5 SOFTWARE

Neste tópico são descritos os processos de criação do canal no ThingSpeak e os testes e encapsulamento do bloco “BD/Cloud”.

3.5.1 Criação do Canal no ThingSpeak

Após o registro de usuário grátis na plataforma, é possível a criação de um datalog para a aplicação. Acessando a guia “Canais”, visualiza-se o menu “Meus Canais”, que contém um botão denominado “New Channel”, conforme ilustra a Figura 12.

Figura 12 - Tela “Meus Canais” do ThingSpeak



Fonte: Autoria própria.

Após clicar o botão “New Channel”, a tela de criação do canal se apresenta e é possível preencher os campos com os dados do sensor que se quer monitorar. No presente trabalho foi necessário apenas preencher os dados “Nome” e “Campo 1” do canal, mas existe a opção de preencher diversas outras informações sobre o sensor a ser criado, como “Descrição”, “Metadata”, “Tags”, “URL”, “Altitude”, “Latitude”, “Longitude” e “ID do vídeo”, como demonstrado na Figura 13.

Figura 13 - Tela “New Channel” do ThingSpeak

The screenshot shows the 'New Channel' page on ThingSpeak. The form includes the following fields and options:

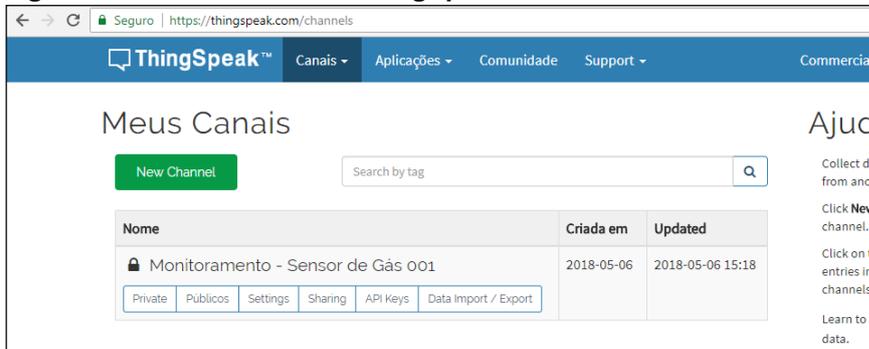
- Nome:** Monitoramento - Sensor de Gás 001
- Descrição:** (empty)
- Campo 1:** Concentração de Gás
- Campo 2:** (empty)
- Campo 3:** (empty)
- Campo 4:** (empty)
- Campo 5:** (empty)
- Campo 6:** (empty)
- Campo 7:** (empty)
- Campo 8:** (empty)
- Metadata:** (empty)
- Tags:** (empty) (Tags are comma separated)
- URL:** http://
- Altitude:** (empty)
- Show Location:**
- Latitude:** 0.0
- Longitude:** 0.0
- Show Video:**
- Video Type:** YouTube Vimeo
- ID do vídeo:** http://
- Show Status:**

At the bottom of the form is a green **Save Channel** button. The right sidebar contains help text and 'Channel Settings' instructions.

Fonte: Autoria própria.

Após o preenchimento dos dados básicos do sensor, é necessário pressionar o botão “Save Channel”. Dessa maneira, quando se acessa a guia “Canais” -> “Meus Canais” novamente, pode-se visualizar o canal criado, conforme ilustra a Figura 14.

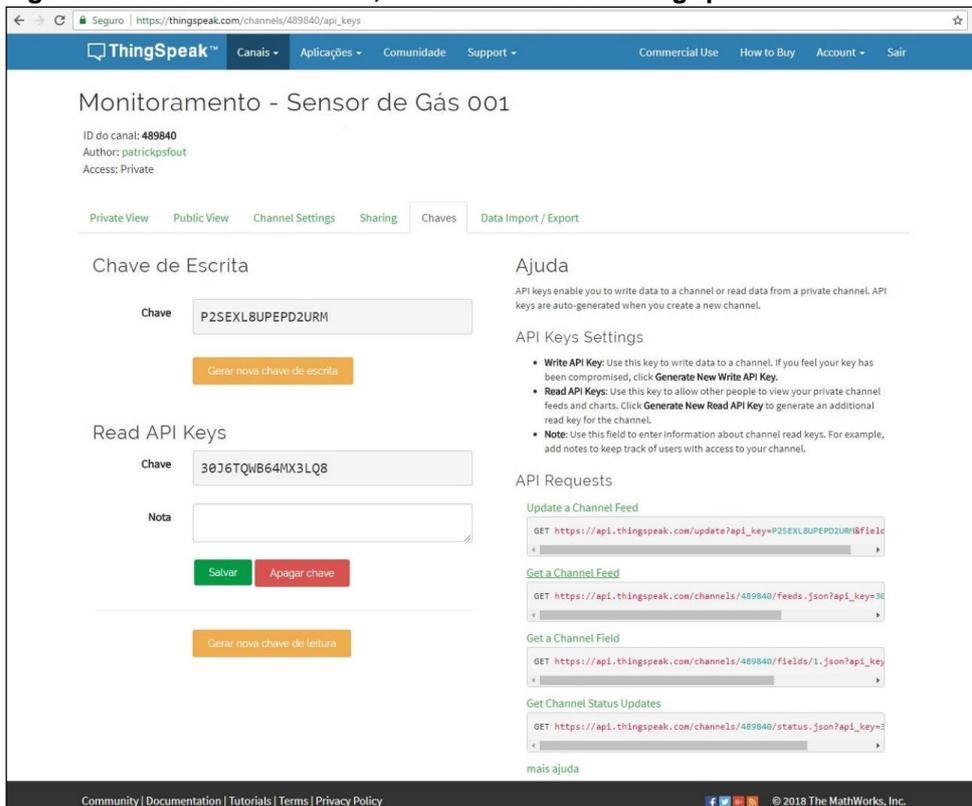
Figura 14 - Canal criado no ThingSpeak



Fonte: Autoria própria.

Ao se acessar o canal criado, os dados do sensor podem ser visualizados, assim como o Dashboard principal que contém o gráfico de concentração de gás ao longo do tempo. Acessando a guia “Chaves” do Dashboard é possível visualizar todos os dados necessários para o projeto do bloco denominado “nó” da plataforma de medição de concentração de gás, conforme ilustrado na Figura 15.

Figura 15 - Chaves de escrita, leitura e API's do ThingSpeak



Fonte: Autoria própria.

Aqui é importante se fazer um aparte. Ainda na Figura 15 é esboçado todo o conjunto de informações necessárias para alimentar o dashboard do sensor, como ID do canal, chaves de escrita e leitura, API Requests. Em projetos de leitura de sensores as informações fundamentais estão em “Chave de Escrita” e “API Requests – Update a Channel Feed”, pois são as que delineiam toda a construção do bloco denominado “nó”.

Observando-se a API Request denominada pelo Thingspeak como “Update a Channel Feed”, visualiza-se a seguinte string:

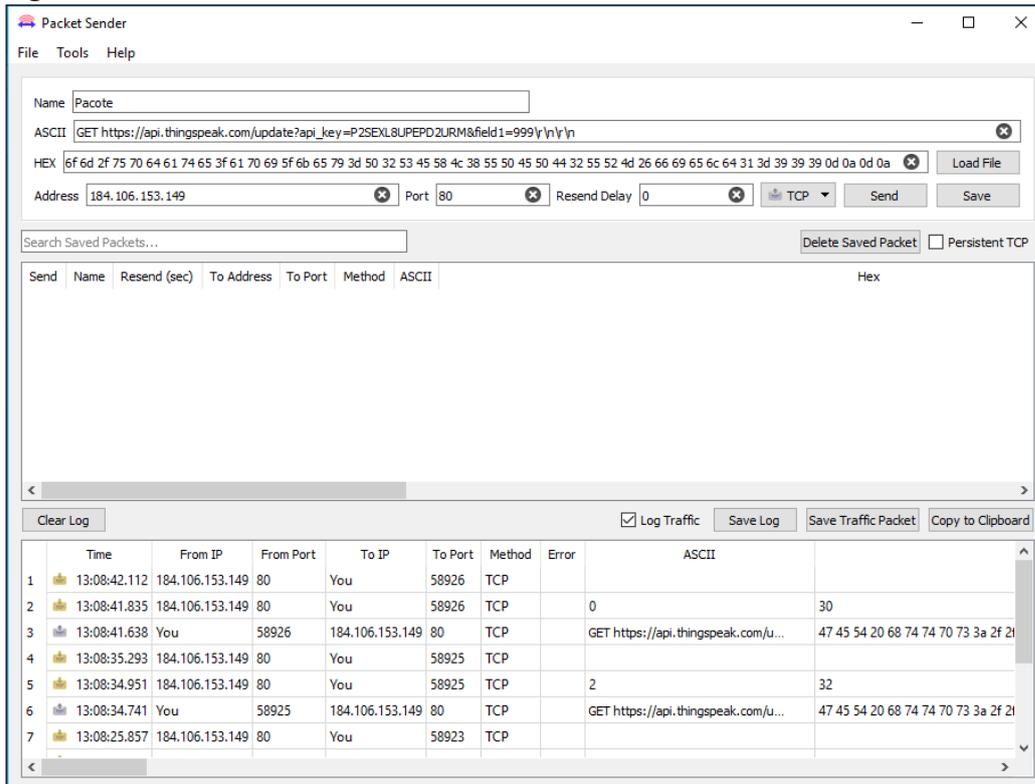
```
GET https://api.thingspeak.com/update?api_key=P2SEXL8UPEPD2URM&field1=0
```

Essa string contém o endereço do servidor do Thingspeak para o qual se deseja enviar a informação, a chave de escrita e o campo que contém o valor coletado pelo sensor.

3.5.2 Testes e Encapsulamento do Bloco “BD/Cloud”

Para encapsular o bloco do sistema denominado “BD/Cloud” foram executados testes para a alimentação do dashboard do ThingSpeak e validação da string que será enviada pelo bloco denominado “nó”. Para este fim foi utilizado o aplicativo Packet Sender, que é grátis e pode ser utilizado sem restrições para aplicações pessoais e acadêmicas. No aplicativo é necessário apenas adicionar a string no campo “ASCII”, informar o endereço e porta do servidor do ThingSpeak, que são 184.106.153.149 e 80, respectivamente, informar que se trata de TCP na camada de transporte e clicar em “Send”. A informação é imediatamente encaminhada, conforme ilustrado na Figura 16.

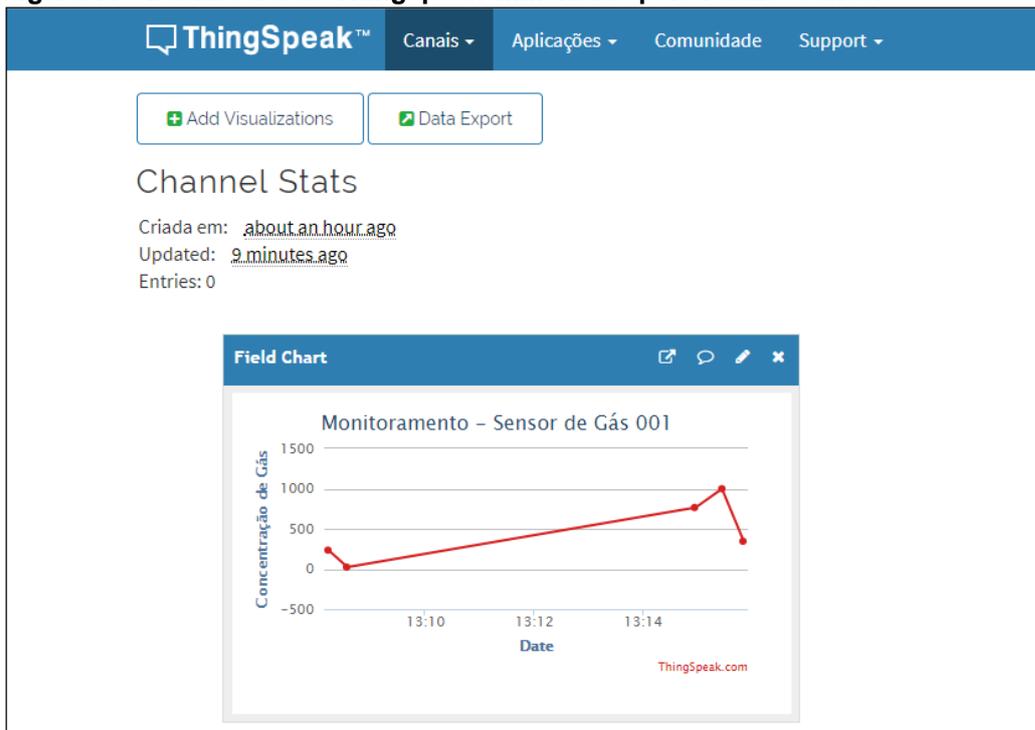
Figura 16 - Testes com o Packet Sender



Fonte: Autoria própria.

Após a transmissão de diversos valores aleatórios no campo “field1” da string, foi observado o resultado no dashboard do ThingSpeak, conforme Figura 17.

Figura 17 - Dashboard do Thingspeak alimentado pelo Packet Sender



Fonte: Autoria própria.

Observações:

1. Por característica de projeto, a taxa de atualização do Thingspeak é de 15 segundos, ou seja, quaisquer informações que forem enviadas para a plataforma no intervalo de tempo entre duas atualizações serão perdidas;
2. Após a string de informações é necessário especificar o término do payload, adicionando-se a string “\r\n\r\n”.

3.5.3 Integração

A Fotografia 7 mostra o hardware completo da plataforma. São os três nós executando medições independentemente e cada um deles enviando as informações de concentração de gás no meio para o ThingSpeak.

Fotografia 7 - Protótipo da plataforma completa



Fonte: Autoria própria.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos utilizando a plataforma de medição completa. Validados os testes iniciais com um bloco “nó”, foi executada a construção de mais dois blocos para os testes da plataforma completa. Para isso foi necessária a mudança de somente uma informação em cada bloco: a chave de escrita do ThingSpeak. No dashboard do ThingSpeak os blocos foram denominados da seguinte maneira:

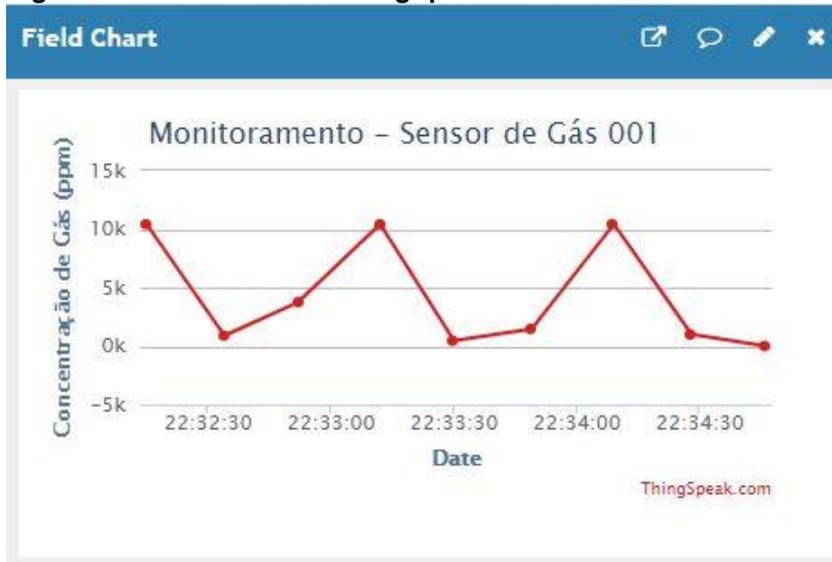
- Monitoramento - Sensor de Gás 001;
- Monitoramento - Sensor de Gás 002;
- Monitoramento - Sensor de Gás 003.

Os blocos foram acomodados em uma sala de aproximadamente 15m x 3m, de maneira que ficaram equidistantes 2m um do outro e também a aproximadamente 2m, cada um, do modem ADSL. Com os blocos em funcionamento e após diversos testes aproximando o gás GLP aos sensores, chegou-se aos resultados descritos nos tópicos 4.1 a 4.3.

4.1 SENSOR DE GÁS 001

A Figura 18 ilustra o dashboard do ThingSpeak com os testes executados no nó 1 da plataforma. É possível observar três picos de medição no valor máximo de 10.000ppm. Esses picos foram conseguidos no momento da aplicação do gás no invólucro de plástico que acomodava o sensor. Após o primeiro pico é possível observar a oscilação da concentração de gás, passando por valores intermediários de aproximadamente 1.000ppm, 2.000ppm e 4.000ppm. Isso foi conseguido utilizando-se um invólucro de plástico que manteve alta concentração de gás nas proximidades do sensor por um período longo de tempo.

Figura 18 - Dashboard do ThingSpeak - testes com o nó 1

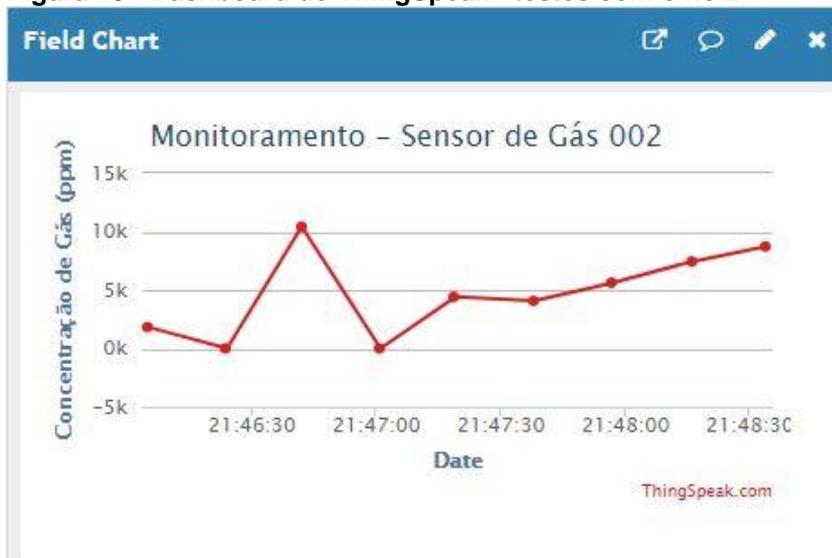


Fonte: Autoria própria.

4.2 SENSOR DE GÁS 002

A Figura 19 ilustra o dashboard do ThingSpeak com os testes executados no nó 2 da plataforma. É possível observar um pico de medição de 10.000ppm no momento da aplicação do gás no sensor. Posteriormente os valores subiram, passando por valores intermediários de aproximadamente 5.000ppm, 6.000ppm, 8.000ppm e 9.000ppm. Esses valores intermediários foram conseguidos aplicando-se gradativamente o gás no invólucro de plástico que acomodava o sensor.

Figura 19 - Dashboard do ThingSpeak - testes com o nó 2

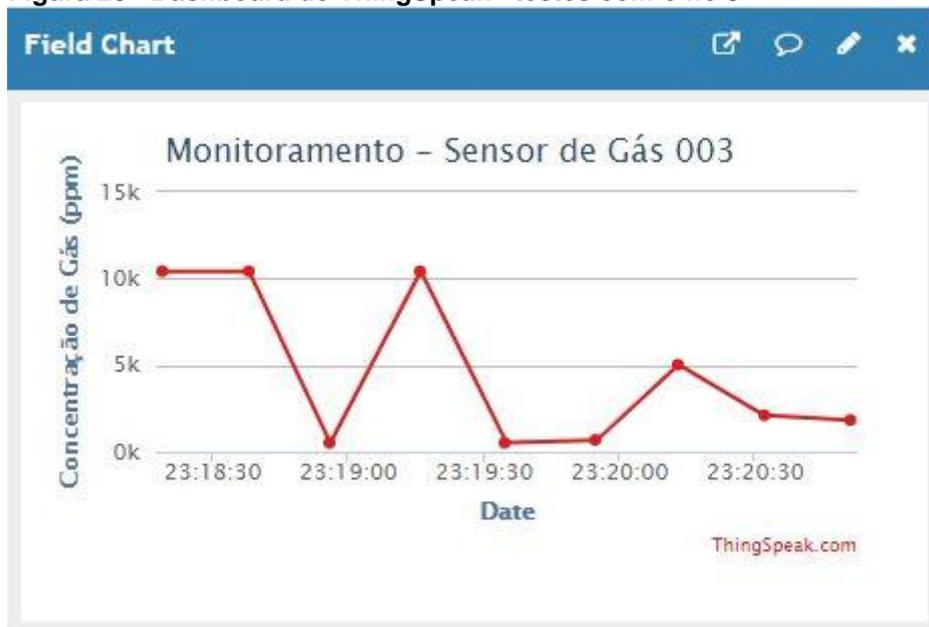


Fonte: Autoria própria.

4.3 SENSOR DE GÁS 003

A Figura 20 ilustra o dashboard do ThingSpeak com os testes executados no nó 3 da plataforma. É possível observar três picos de medição de 10.000ppm e diversos valores intermediários como aproximadamente 500ppm, 2.000ppm, 5000ppm. Essas oscilações foram obtidas enchendo-se e esvaziando-se rapidamente de gás o invólucro de plástico que acomodava o sensor.

Figura 20 - Dashboard do ThingSpeak - testes com o nó 3



Fonte: Autoria própria.

4.4 DISCUSSÃO

Os testes executados foram utilizados para testar as funcionalidades básicas da plataforma. Ao aproximar o gás GLP aos sensores, utilizando-se um isqueiro e posteriormente um spray de gás de isqueiro, pôde-se verificar o funcionamento básico dos circuitos. Nota-se nos gráficos de concentração de gás das Figuras 18, 19 e 20, que o valor máximo de concentração de gás, que é próximo a 10.000ppm, foi atingido em quase todas as medições quando o gás estava presente. Alguns valores intermediários de medida foram obtidos com um invólucro de plástico que mantinha o gás próximo ao sensor em concentrações menores que a máxima. Isso demonstra as dificuldades práticas de levantamento de projetos que envolvem a concentração de materiais voláteis. Em uma implementação comercial/industrial essa metodologia de

testes deve ser mais bem elaborada. Dependendo da criticidade do controle necessário pode ser que empresas especializadas em aferição de sensores devam ser contratadas.

5 CONSIDERAÇÕES FINAIS

A proposta deste trabalho foi a elaboração e construção de um protótipo para uma plataforma capaz de implementar a medição dos níveis de gás GLP em ambientes nos quais isso se faz necessário. Foram expostos os objetivos, as justificativas e a estrutura geral do trabalho. A fundamentação teórica, que apresentou os conceitos e tecnologias que viriam a ser utilizados no projeto, também foi apresentada. Foram apresentados os recursos utilizados, diagramas dos circuitos montados, as etapas de projeto necessárias para se chegar ao protótipo e à integração do mesmo com o software.

Após a execução de exaustivos testes com a plataforma completa, chega-se à conclusão de que todos os requisitos do protótipo foram atendidos. Trata-se de uma plataforma de medição dos níveis de gás em um ambiente e que apresenta os resultados em um dashboard de cliente na nuvem. A metodologia utilizada para a construção desse protótipo pode ser seguida para a construção de quaisquer outras plataformas e utilizando quaisquer outras tecnologias e/ou fabricantes. A descrição completa da experiência também foi adequada para se estabelecer um hardware mínimo para o levantamento do protótipo. Ao se estabelecer a utilização do Arduino Uno R3 não era possível saber que a porta serial emulada do mesmo não funciona adequadamente a taxas acima de 9600 baud. A implementação do protótipo utilizando-se a abordagem em blocos com entradas e saídas bem definidas também foi importante para uma visão de futuro do produto. Com essa abordagem é possível fazer a substituição de cada bloco por soluções proprietárias.

6 TRABALHOS FUTUROS

Na construção do protótipo para validar o projeto, pôde-se utilizar tecnologias simples e economicamente viáveis, mas para se trabalhar com produtos mais robustos, do ponto de vista comercial, é necessária outra abordagem na escolha dos componentes. Requisitos como confiabilidade, economia de energia, protocolos de comunicação implementáveis, ruído do meio, tornam a escolha dos dispositivos um fator crucial para a construção do produto comercial final. Importante é ressaltar que qualquer que seja essa escolha, a metodologia de construção da plataforma é a mesma que foi seguida neste trabalho. Desta maneira, pode-se substituir microcontrolador, sensor, comunicação Wi-Fi, concentrador, BD/Cloud por outros componentes que se adequem às especificações e requisitos de qualquer instituição ou organização.

6.1 MICRONTROLADOR

Embora a utilização da plataforma Arduino seja muito conveniente para a elaboração de protótipos, na prática recomenda-se que a implementação de produtos finais seja feita com a utilização de fabricantes mais especializados. Desta forma pode-se utilizar, por exemplo, o microcontrolador CC2650 da Texas Instruments. Essa linha de chips vem com os requisitos de hardware ainda mais bem dimensionados com relação ao Arduino e já conta com a comunicação Wi-Fi nativa, podendo-se suprimir a utilização do ESP8266.

Preço do kit de desenvolvimento LAUNCHXL-CC2650, vendido no site da Texas Instruments, e que pode ser utilizado para o desenvolvimento do produto final: \$29.00 (USD) ou R\$ 108,75. Dólar a R\$ 3,75 em 31/07/2018.

6.2 SENSOR

O sensor utilizado no protótipo deste trabalho, o MQ-2 da Hanwei Eletronics, é extremamente utilizado em projetos similares, sendo uma das principais referências em medição de concentração de gases tóxicos e/ou inflamáveis. Uma alternativa mais

robusta e com maior precisão para medições de concentração de gases são os sensores baseados em NDIR (Tecnologia de Infravermelho Não Dispersivo).

Existem vários fabricantes e modelos no mercado e os preços variam bastante. Uma característica que faz esse preço subir bastante é se o medidor/detector de gases atende às normas NR-33 (Segurança e Saúde nos Trabalhos em Espaços Confinados). O sensor MQ-2 utilizado neste projeto, que não atende a essas normas, por exemplo, custa aproximadamente R\$ 15,00, enquanto um medidor profissional que atende às normas custa aproximadamente R\$ 4.000,00. Essa comparação pode ser considerada imprecisa, pois o sensor MQ-2 é só uma parte do produto final, mas pode dar uma ideia do quanto se gastaria em desenvolvimento para adequar o protótipo às normas de segurança vigentes. O vendedor do Detector GasAlert MicroClip XL anuncia o produto, em 29/08/2018, por R\$ 4.036,00 (INSTRUMBRASIL, 2018).

6.3 COMUNICAÇÃO WI-FI

Para a prototipação da plataforma pôde-se utilizar, para fins didáticos e explanatórios, um microcontrolador que não tem módulo Wi-Fi nativo, mas para aplicações comerciais e profissionais a escolha de um microcontrolador que tenha Wi-Fi nativo é a melhor recomendação. O kit de desenvolvimento LAUNCHXL-CC2650, mencionado no tópico 5.1, possui Wi-Fi nativo.

6.4 CONCENTRADOR

No protótipo da plataforma foi utilizado um link ADSL ligado a um modem/roteador doméstico. Em condições normais de temperatura, pressão, umidade e com o ambiente livre de ruídos, essa abordagem atende aos requisitos. Quando o ambiente, principalmente o industrial, se distancia do ambiente doméstico, é necessária uma abordagem diferente para esse tipo de solução.

Outras arquiteturas de rede podem ser utilizadas, como as que utilizam outros protocolos como o Via de Dados Endereçável por Transdutor Remoto (HART, do inglês *Highway Addressable Remote Transducer*), Atuador de Interface de Sensor (AS-i, do inglês *Sensor Interface Actuator*), Barramento de Campo de Processo

(PROFIBUS, do inglês *Process Field Bus*), Rede de Dispositivos (DeviceNet, do inglês *Device Network*), Via de Dados Endereçável por Transdutor Remoto Sem Fio (WirelessHART, do inglês *Wireless Highway Addressable Remote Transducer*), Arquitetura de interconexão para redes locais industriais (EthernetIndustrial, do inglês *Ethernet Industrial*), Zigbee e Bluetooth (STEVAN JUNIOR; LEME; SANTOS, 2018).

6.5 BD/CLOUD

Para a implementação do protótipo da plataforma foi utilizada uma cloud bastante popular no monitoramento de sensores, o ThingSpeak. A escolha dessa nuvem para a prototipação foi devido à simplicidade de acesso, por ser grátis para fins acadêmicos e porque implementa a funcionalidade básica necessária para o projeto. Para aplicações comerciais pode-se utilizar a versão paga da ferramenta, que expande as funcionalidades e deixa a ferramenta mais adequada para utilização profissional. Outras nuvens como IBM e Google também podem ser utilizadas.

Dependendo da criticidade de sigilo das informações de cada organização ou empresa, pode-se também pensar em uma abordagem proprietária da funcionalidade, como servidores internos dedicados a esse tipo de implementação.

6.6 PROTOCOLO DE APLICAÇÃO

Na construção do protótipo foi utilizado o protocolo de aplicação HTTP. Devido às limitações do hardware utilizado não foi possível a implementação de outros protocolos. Como a cloud ThingSpeak, na versão grátis, tem limitação de atualização de 15 segundos, foi possível implementar os três nós com a utilização de requisições POST HTTP. Em uma implementação comercial, dependendo da demanda de cada organização, torna-se necessária a utilização de algum protocolo de aplicação mais dinâmico e eficiente para aplicações de Internet das Coisas. Dependendo do número de dispositivos ligados à rede, largura de banda disponível, restrições de consumo de energia dos dispositivos, torna-se mandatória a utilização de algum protocolo específico para Internet das Coisas, como o Transporte de Telemetria de Fila de Mensagens (MQTT, do inglês *Message Queuing Telemetry Transport*) ou Protocolo de Aplicação Restrita (CoAP, do inglês *Constrained Application Protocol*).

De acordo com os relatórios de análise e teste, a transferência de dados MQTT pode transferir dados a uma taxa 20 a 25 vezes mais rápida que as chamadas HTTP REST (PRATA, 2018).

REFERÊNCIAS

ALVES, Josemar. **Configurando GPIO do ESP8266-01**. Novembro 2016. Disponível em: <<https://www.embarcados.com.br/configurando-gpio-do-esp8266-01/>>. Acesso em: 07 nov. 2018.

ARDUINO. **Arduino Pro Mini**. 2018. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardProMini/>>. Acesso em: 07 nov. 2018.

ARDUINO. **Como programar o Arduino Pro Mini usando conversor FTDI**. Setembro 2014. Disponível em: <<https://www.arduinoecia.com.br/2014/09/conversor-ftdi-ft232rl-arduino-pro-mini.html>>. Acesso em: 07 nov. 2018.

CURVELLO, André. **Apresentando o módulo ESP8266**. Abril 2015. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 07 nov. 2018.

EFETIVIDADE. **ESP8266 AT Instruction Set**. Copyright© Espressif Systems IOT Team 2015. Disponível em: <<https://static.efetividade.net/img/esp8266-at-63240.pdf>>. Acesso em: 07 nov. 2018.

FILIPEFLOP. **Sensor de gás MQ-2 inflamável e fumaça**. Agosto 2018. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-gas-mq-2-inflamavel-e-fumaca/>>. Acesso em: 07 nov. 2018.

INSTRUMBRASIL. **Detector de 4 gases (espaço confinado) GasAlert MicroClip XL**. Agosto 2018. Disponível em: <http://www.instrumbrasil.com.br/seguranca-do-trabalho/medidor-de-gas/detector-4-gases-espaco-confinado-gasalert-microclip-xl.phtml?gclid=CjwKCAjwtlXbBRBhEiwAWV-5ngFHdui8FVcpF-OpZOjUP2lvpcwl6DYmNyVCkKtU7EOvqUO2x2_kvhoCJhYQAvD_BwE>. Acesso em: 08 nov. 2018.

MOURA, Antônio Carlos Magalhães. **Novas tendências de consumo de GLP: gás liquefeito de petróleo no Brasil**. Monografia (Pós-graduação) - Centro Universitário do Instituto Mauá de Tecnologia, São Caetano do Sul, SP, 2012. Disponível em: <<https://maua.br/files/monografias/novas-tendencias-de-consumo-de-glp-gas-liquefeito-de-petroleo-no-brasil.pdf>>. Acesso em: 03 nov. 2018.

NASCIMENTO, Gabriel. **Qual a diferença entre os modelos de Arduino? Qual deles devo escolher?** Agosto 2017. Disponível em: <<https://www.sgnrobotica.com/single-post/2017/08/24/Qual-a-diferen%C3%A7a-entre-os-modelos-de-Arduino-Qual-deles-devo-escolher>>. Acesso em: 07 nov. 2018.

OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266**. Arduino e Raspberry PI. São Paulo: Novatec, 2017.

PETROBRÁS. **Gás liquefeito de petróleo: informações técnicas**. Outubro 2013. Disponível em: <<http://sites.petrobras.com.br/minisite/assistenciatecnica/public/downloads/manual-tecnico-gas-liquefeito-petrobras-assistencia-tecnica-petrobras.pdf>>. Acesso em: 04 nov. 2018.

PETROBRÁS. **Gás Liquefeito de Petróleo (GLP)**. Outubro 2018. Disponível em: <<http://www.petrobras.com.br/pt/produtos-e-servicos/produtos/domesticos/gas-liquefeito-de-petroleo-glp/>>. Acesso em: 03 nov. 2018.

PRATA, Renan. **MQTT vs REST: perspectiva de uma implementação IoT**. Junho 2018. Disponível em: <<https://medium.com/@renancprata/mqtt-vs-rest-perspectiva-de-uma-implementa%C3%A7%C3%A3o-iot-4c7e1f26689c>>. Acesso em: 07 nov. 2018.

REALI, Andrea. **Handy Arduino Uno R3 Pinout Diagram**. Maio 2012. Disponível em: <<https://blog.arduino.cc/2012/05/29/handy-arduino-uno-r3-pinout-diagram/>>. Acesso em: 07 nov. 2018.

SAMPAIO, Lucas. **Sensoriamento de gases em tempo real através de sensores MQ-(0-9)***. Novembro 2016. Disponível em: <<https://jualabs.wordpress.com/2016/11/30/sensoriamento-de-gases-em-tempo-real-atraves-de-sensores-mq-0-9/>>. Acesso em: 07 nov. 2018.

SOUZA, Fábio. **Arduino MEGA 2560**. Abril 2014. Disponível em: <<https://www.embarcados.com.br/arduino-mega-2560/>>. Acesso em: 07 nov. 2018.

STEVAN JUNIOR, Sergio L.; LEME, Murilo O.; SANTOS, Max M. D. **Indústria 4.0: Fundamentos, perspectivas e aplicações**. São Paulo: Érica, 2018.