



UNIVERSIDADE FEDERAL DO TOCANTINS

CAMPUS DE PALMAS

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Pedro Henrique Cavalcante Noletto

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA MONITORAMENTO
UTILIZANDO CONCEITOS DE INTERNET DAS COISAS**

Palmas-TO

2020

PEDRO HENRIQUE CAVALCANTE NOLETO

**DESENVOLVIMENTO DE UMA APLICAÇÃO PARA MONITORAMENTO
UTILIZANDO CONCEITOS DE INTERNET DAS COISAS**

Monografia foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, Curso de Engenharia Elétrica para obtenção do título de Bacharel em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: Me. Alex Vilarindo Menezes

Palmas-TO

2020

<https://sistemas.uft.edu.br/ficha/>

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

N791d Noletto, Pedro Henrique Cavalcante.
Desenvolvimento de uma aplicação para monitoramento utilizando conceitos de internet das coisas. / Pedro Henrique Cavalcante Noletto. – Palmas, TO, 2020.
47 f.
Monografia Graduação - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Engenharia Elétrica, 2020.
Orientador: Alex Vilarindo Menezes Menezes
1. Monitoramento. 2. Internet das Coisas. 3. Arduino. 4. Node-red. I.
Título

CDD 621.3

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).

FOLHA DE APROVAÇÃO

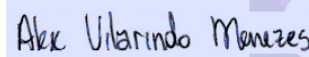
PEDRO HENRIQUE CAVALCANTE NOLETO

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA MONITORAMENTO UTILIZANDO CONCEITOS DE INTERNET DAS COISAS

Monografia foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, Curso de Engenharia Elétrica para obtenção do título de Bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de aprovação: 03 / 08 / 2020

Banca Examinadora


Digitally signed by ALEX VILARINDO MENEZES:
01374778133
DN: C=BR, O=ICP-Brasil, OU=Secretaria da
Receita Federal do Brasil - RFB, OU=RFB e-CPF
A3, OU=(EM BRANCO), OU=31230362000100,
CN=ALEX VILARINDO MENEZES:01374778133
Reason: TCC - Pedro Noleto
Location: Palmas-TO
Date: 2020-08-05 15:47:07
Foxit Reader Version: 9.7.0

Prof. Me. Alex Vilarindo Menezes, UFT (Orientador)



Prof. Dr. Marcus André Pereira Oliveira, IFTO



Prof. Dr. Humberto Xavier de Araújo, UFT

Palmas, 2020

Dedico à minha mãe Irene Ribeiro Cavalcante.

AGRADECIMENTOS

Aos meus familiares pelo apoio, paciência e incentivo ao longo dessa longa caminhada que é a graduação. Minha eterna gratidão à todos.

Ao meu orientador Prof. Me. Alex Vilarindo Menezes, pelo auxílio e direcionamento no desenvolvimento deste trabalho, revisão do texto e disponibilidade de tempo e local para o desenvolvimento do trabalho.

Ao Prof. Dr. Marcus André Pereira Oliveira por toda ajuda fornecida, sempre de maneira muito solícita procurou contribuir com boas sugestões e fornecimento de material próprio. Seguramente suas intervenções foram fundamentais na execução desse trabalho.

Ao Prof. Me. Tiago da Silva Almeida pelas sugestões de ferramentas que foram utilizadas ao longo do trabalho.

Aos amigos que não pouparam esforços em me ajudar com materiais de estudo, fornecimento de equipamentos e sugestões, sou grato ao Lucas Cajueiro, Alex Dias da Cruz, Faronak Kharaghani Amorim, Vanessa Clímaco Oliveira Gomes, Thiago Silva Pereira, Thierry Silva Pereira, Rafael Chagas da Costa, Breno Quaresma Praça Leal e Wltagnan Silva Lima.

RESUMO

A sociedade está passando por um processo tecnológico irreversível na maneira de se relacionar com o mundo. A revolução tecnológica ora em curso está trazendo à tona tecnologias que alterarão a forma como trabalhamos, vivemos e nos relacionamos. Nesse contexto, o termo “internet das coisas” (do inglês *Internet of Things*, IoT) se destaca entre as muitas inovações emergentes da revolução tecnológica. Aprendizado de máquinas, armazenamento na nuvem, inteligência artificial e monitoramento são algumas das muitas possibilidades que a IoT oferece, trazendo consigo novas oportunidades e participando efetivamente do processo de ruptura com os modelos tradicionais de negócio. Este trabalho consiste no desenvolvimento e análise de uma aplicação de monitoramento da temperatura ambiente por meio da integração de diversas plataformas *open source* e da utilização de ferramentas de IoT, permitindo o armazenamento na nuvem e ações baseadas nos dados, bem como a geração de gráficos dos dados obtidos em computador e *smartphone*. Por fim, são feitas considerações à respeito da infraestrutura proposta e seu potencial de aplicação nas diversas áreas da engenharia elétrica.

Palavras-chaves: Internet das Coisas. *NODE-RED*. *ThingSpeak*. *Arduino*.

ABSTRACT

Society is going through an irreversible technological process in the way of relating to the world. The technological revolution now underway is bringing to the fore technologies that will change the way we work, live and relate. In this context, the term “internet of things” (IoT) stands out among the many innovations emerging from the technological revolution. Machine learning, cloud storage, artificial intelligence and monitoring are some of the many possibilities that IoT offers, bringing with it new opportunities and participating effectively in the process of breaking with traditional business models. This work consists in the development and analysis of an application for monitoring the ambient temperature through the integration of several open source platforms and the use of IoT tools, allowing for cloud storage and data-based actions, as well as the generation of graphs of the data obtained on computer and smartphone. Finally, considerations are made regarding the proposed infrastructure and its potential for application in the various areas of electrical engineering.

Key-words: Internet of Things. NODE-RED. ThingSpeak. Arduino.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Placa do Arduino UNO..... | 19 |
| Figura 2 - IDE Arduino..... | 20 |
| Figura 3 - Fluxo em NODE-RED..... | 21 |
| Figura 4 - Nós do NODE-RED..... | 22 |
| Figura 5 - Funcionamento básico do MQTT..... | 25 |
| Figura 6 - Site do ThingSpeak..... | 28 |
| Figura 7 - Cadastro no ThingSpeak..... | 28 |
| Figura 8 - Acesso ao ThingSpeak..... | 29 |
| Figura 9 - Password ThingSpeak..... | 29 |
| Figura 10 - Criando canais..... | 30 |
| Figura 11 - Chaves ThingSpeak..... | 31 |
| Figura 12 - Chaves MQTT API Key..... | 32 |
| Figura 13 - Circuito montado..... | 33 |
| Figura 14 - Fluxo no NODE-RED..... | 35 |
| Figura 15 - Criando um novo canal no ThingSpeak..... | 36 |
| Figura 16 - Assinando canais por smartphone..... | 37 |
| Figura 17 - Análise gráfica no NODE-RED..... | 38 |
| Figura 18 - Análise gráfica no ThingSpeak sem tratamento..... | 38 |
| Figura 19 - Análise gráfica no ThingSpeak com tratamento..... | 39 |
| Figura 20 - Análise gráfica no IoT Monitor Widget..... | 39 |
| Figura 21 - Notificação baseada em evento..... | 40 |
| Figura 22 - Exibindo dados no Excel..... | 40 |

LISTA DE ABREVIATURAS E SIGLAS

API : *Application Programming Interface*

AWS: *Amazon Web Services*

BNDES: Banco Nacional do Desenvolvimento

CA: Corrente Alternada

CC: Corrente Contínua

CSS: *Cascading Style Sheets*

CSV: *Comma-separated values*

HTML: *Hypertext Markup Language*

IBM: *International Business Machines*

ID: *Identify*

IDE: *Integrated Development Environment*

IoT: Internet das Coisas, *Internet of Things*

IP: *Internet Protocol*

JS: JavaScript

JSON: *JavaScript Object Notation*

M2M: *Machine to Machine*

MCTIC: Ministério da Ciência, Tecnologia, Inovações e Comunicações

MQTT: *Message Queuing Telemetry Transport*

Msg: Mensagem

PWM: *Pulse Width Modulation*

QoS: *Quality of Service*

RAM: *Random Access Memory*

TCP: *Transport Control Protocol*

USB: *Universal Serial Bus*

Zip: *Zone Information Postal*

LISTA DE SÍMBOLOS

V: Volt
°C: Graus Celsius

SUMÁRIO

| | |
|---|----|
| 1 INTRODUÇÃO | 13 |
| 1.1 REVOLUÇÃO TECNÓLOGICA E INTERNET DAS COISAS | 13 |
| 1.2 PROBLEMA DE PESQUISA | 14 |
| 1.2.1 Hipótese | 15 |
| 1.2.2 Delimitação do escopo | 15 |
| 1.2.3 Justificativa | 15 |
| 1.3 OBJETIVOS | 16 |
| 1.3.1 Objetivos Gerais | 16 |
| 1.3.2 Objetivos Específicos | 16 |
| 1.4 METODOLOGIA | 16 |
| 1.5 ESTRUTURA DO TRABALHO | 17 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 18 |
| 2.1 COMPUTAÇÃO EM NUVEM | 18 |
| 2.2 ARDUÍNO | 18 |
| 2.2.1 Programação na IDE do Arduino UNO | 19 |
| 2.3 NODE-RED | 21 |
| 2.3.1 Termos recorrentes | 22 |
| 2.4 MQTT | 24 |
| 2.4.1 Termos recorrentes | 26 |
| 2.5 THINGSPEAK | 27 |
| 2.5.1 Cadastro na plataforma | 28 |
| 2.5.2 Criação de canais | 29 |
| 2.5.3 Disparo de ações por eventos | 30 |
| 2.5.4 Publicando dados em um canal | 30 |
| 2.5.5 Aplicativos para smartphone | 32 |
| 3 DESENVOLVIMENTO DA APLICAÇÃO | 33 |
| 3.1 MONTAGEM DO CIRCUITO | 33 |
| 3.2 PROGRAMAÇÃO NO NODE-RED | 35 |
| 3.3 UTILIZAÇÃO DA PLATAFORMA THINGSPEAK | 36 |
| 3.4 UTILIZAÇÃO DE APLICATIVOS PARA SMARTPHONE | 37 |
| 3.5 IMPLEMENTAÇÃO DO DISPARO DE AÇÕES POR EVENTOS | 37 |
| 4 RESULTADOS | 38 |
| 4.1 VISUALIZAÇÃO GRÁFICA DOS DADOS EM TEMPO REAL | 38 |

| | |
|---|----|
| 5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS | 41 |
| REFERÊNCIAS | 42 |
| APÊNDICE A - INSTALAÇÃO DO NODE-RED NO WINDOWS | 45 |
| APÊNDICE B - CÓDIGO DO FLUXO NO NODE-RED | 46 |

1 INTRODUÇÃO

1.1 REVOLUÇÃO TECNÓLOGICA E INTERNET DAS COISAS

A palavra revolução remete às transformações radicais na estrutura de uma sociedade, bem como às mudanças profundas e abruptas que redefiniram o curso da história (DICIO, 2019). Revoluções já ocorreram anteriormente e modificaram a maneira do ser humano interagir com o mundo. Por exemplo, a revolução agrícola foi responsável pela sedentarização e civilização do homem, culminando no surgimento dos antigos impérios, enquanto que a revolução industrial foi responsável pela substituição dos meios de produção artesanais por processos industriais, alavancando a economia e ciência dos países que a implementaram primeiro, de tal maneira que ainda hoje pode-se perceber claramente as lacunas socioeconômicas existentes entre países desenvolvidos e em desenvolvimento.

A revolução tecnológica é um fenômeno contemporâneo que está se consolidando em razão do seu potencial de aplicação nos mais diversos campos das atividades humanas. De fato, nas duas últimas décadas tecnologias como internet, *smartphones*, redes sociais e aprendizado de máquinas estão alterando substancialmente aspectos da vida cotidiana, tais como, as interações sociais, política, economia, etc.

Nesse contexto, a Internet das Coisas (*Internet of Things* – IoT, em inglês) se destaca entre as muitas inovações emergentes dos avanços na microeletrônica, embarcados, sensoriamento e comunicação. Pode ser definida como uma extensão da internet atual, na qual objetos dotados de sensores inteligentes e *softwares* embarcados podem interagir entre si e com o usuário de maneira sistêmica, coletando e processando informações do ambiente e da rede aos quais estão conectados.

A Internet das Coisas é um novo paradigma que combina aspectos e tecnologias oriundas de diferentes abordagens. Computação onisciente, computação penetrante, protocolos de internet, tecnologias de sensores, tecnologias da comunicação e dispositivos embutidos são mesclados juntos em ordem para formar um sistema onde mundos real e digital se encontram e estão continuamente em interação simbiótica. (BORGIA, 2014).

As inovações propostas pela IoT desafiam o *status quo* dos modelos de negócios existentes. O mercado de IoT deverá movimentar de 4 a 11 trilhões de dólares em 2025. Nesse cenário de grandes oportunidades, priorizam-se ideias inovadoras, redesenhando assim o modelo tradicional da indústria (MCKINSEY & COMPANY 2015).

Com a internet das coisas, inúmeras possibilidades de monitoramento, controle e melhoria de processos surgiram. Os segmentos de mercado que sofrerão influência disso são os mais variados possíveis, exercendo especial influência em nichos com forte grau de emprego de tecnologia, como setor agropecuário e produção industrial, por exemplo. (BERTOLETI, 2019)

Muitos países já se deram conta do poder disruptivo que a IoT impõe à economia global. De fato, estratégias de governo estão sendo consideradas para garantir a competitividade das suas indústrias e o fortalecimento de sua economia.

O Brasil já começou a traçar um plano estratégico para o país, à respeito do tema IoT. Com o objetivo de realizar um diagnóstico e propor políticas públicas no tema para o Brasil, em outubro de 2017 o Banco Nacional de Desenvolvimento Econômico e Social (BNDES), em parceria com o Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC) publicou um relatório do estudo “Internet das Coisas: um plano de ação para o Brasil”. No qual são apresentadas as aspirações de IoT para o Brasil, bem como as estratégias de desenvolvimento aplicadas em contextos específicos, a saber, cidades, saúde, rural e indústria.

1.2 PROBLEMA DE PESQUISA

O monitoramento de processos em tempo real, processamento de dados e tomada de decisões baseadas nos dados de entrada são algumas das muitas possibilidades oferecidas pela IoT. Esses aspectos da IoT vêm ao encontro com muitas demandas da engenharia elétrica, pois diversas soluções de engenharia poderiam ser exploradas utilizando conceitos da IoT.

Algumas empresas e órgãos de pesquisa como *Amazon Web Services (AWS)*, *Microsoft Azure*, *IBM*, *Google* já publicaram arquiteturas para soluções IoT baseadas em suas concepções à respeito do tema; entretanto, o cenário atual para um estudante ainda é muito nebuloso. O processo de desenvolvimento de algumas aplicações se torna muito difícil devido ao excesso de informações fragmentadas, soma-se a isso o fato de que a área de IoT exige um conhecimento multidisciplinar de eletrônica, redes e programação.

Dessa forma, deve-se investigar questões como:

- É viável integrar ferramentas *open source* e plataformas IoT com vistas a desenvolver aplicações em engenharia elétrica como monitoramento de temperatura?
- Quais seriam as ferramentas e plataformas capazes de compor uma aplicação IoT? Quais suas funções específicas?

- As soluções encontradas corresponderiam às demandas dos engenheiros?

Desse modo, explorar estas questões pode ser uma boa maneira de abordar essa problemática e contribuir efetivamente no desenvolvimento de projetos de engenharia elétrica.

1.2.1 Hipótese

Diante das potencialidades oferecidas pela IoT e de sua compatibilidade com as demandas da engenharia elétrica, fica evidente que esse tema não deve ficar em segundo plano, mas estar no centro de interesse de estudantes de engenharia, no que diz respeito ao desenvolvimento de aplicações que facilitem o trabalho e agreguem valor aos produtos desenvolvidos.

Isso permite supor que o desenvolvimento de uma infraestrutura para monitoramento permitirá que diversos conceitos relacionados à IoT sejam explorados através do uso de ferramentas *open source* de maneira dinâmica e simplificada, contribuindo assim para o desenvolvimento de projetos com soluções de engenharia.

1.2.2 Delimitação do escopo

Esse trabalho se propõe a apresentar uma metodologia de aplicação para monitoramento de temperatura por meio de sensores, apresentação gráfica dos dados e registro dos dados obtidos. Além do envio de dados para armazenamento na nuvem e ações baseadas nos dados, bem como a integração de diversas ferramentas *open source* e plataformas de IoT.

1.2.3 Justificativa

A expectativa de aumento no número de dispositivos interconectados, além do grande volume de dados disponíveis irá gerar oportunidades para a criação de novos serviços que irão refletir em benefícios para a sociedade, meio ambiente, economia e cidadãos individuais (BORGIA, 2014).

Nessa perspectiva, o desenvolvimento de uma aplicação IoT que realize o monitoramento de temperatura pode se constituir em uma ótima ferramenta para o desenvolvimento de soluções em engenharia elétrica. Além disso, com a utilização de recursos disponibilizados pela IoT tem-se mais poder de processamento e infraestrutura para desenvolver soluções mais complexas (acesso a banco de dados, aprendizado de máquinas, armazenamento na nuvem, inteligência artificial, etc). Soma-se a isso o fato de o custo para implementação dessas soluções ser relativamente baixo, pois as ferramentas e plataformas

utilizadas serão *open source*, bem como os equipamentos de *hardware* serem também acessíveis tanto pelo baixo custo como (plataforma *Arduino*, ESP-8266, ESP-32, etc.) quanto pela massificação do uso (*smartphones, notebooks, tablets*).

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver e analisar uma aplicação de monitoramento de temperatura por meio da integração de diversas plataformas *open source* e da utilização de ferramentas de IoT, permitindo o armazenamento na nuvem e ações baseadas nos dados, bem como a geração de gráficos dos dados obtidos.

1.3.2 Objetivos Específicos

- Determinar quais protocolos de comunicação melhor atendem à aplicação;
- Armazenar os dados obtidos na nuvem e disponibilizá-los através de registro;
- Desenvolver um *dashboard* capaz de possibilitar a visualização dos dados de entrada em tempo real e dados históricos, por meio de gráficos;
- Desenvolver um sistema de notificações baseadas em eventos específicos no monitoramento;

1.4 METODOLOGIA

Estudo das áreas do conhecimento relacionadas à execução do trabalho, a saber:

- Internet das coisas;
- Linguagens de programação e suas ferramentas de desenvolvimento;
- Infraestrutura de redes e seus protocolos;
- Microcontroladores;
- Plataformas IoT;
- Desenvolvimento da aplicação e realização de testes.

1.5 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado da seguinte maneira:

- O Capítulo 1 trata da apresentação do trabalho, problematização, definição e delimitação do tema a ser abordado e objetivos.
- O Capítulo 2 apresenta a revisão bibliográfica e fundamentação teórica sobre o tema. Este capítulo expõe quais as soluções e plataformas utilizadas para a resolução do problema proposto além das características destas soluções.
- O Capítulo 3 descreve os procedimentos executados na montagem do circuito e no desenvolvimento da aplicação.
- O Capítulo 4 apresenta e discute os resultados obtidos, avaliando se todos os objetivos foram cumpridos.
- O Capítulo 5 trata das considerações finais e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 COMPUTAÇÃO EM NUVEM

A computação em nuvem é o fornecimento de serviços de computação como: servidores, armazenamento, banco de dados, rede, *software*, análise e inteligência pela internet “nuvem” (Microsoft Azure, 2020). É difícil definir onde a nuvem inicia e onde termina, pelo fato de boa parte de seus conceitos serem usados desde os primórdios da internet.

Nos primórdios da internet, todas as empresas que queriam estar presentes na internet precisavam instalar seus próprios servidores, isso demandava altos custos com mão de obra especializada, garantia do fornecimento contínuo de eletricidade ao parque computacional, procedimentos de *backup*, etc. Aos poucos esses serviços começaram a ser terceirizados, isto é, transferidos para *datacenters* externos. Inicialmente serviços *web* e *e-mails* foram disponibilizados por meio de servidores externos, em seguida provedores de aplicações e bancos de dados. Atualmente, serviços de processamento podem ser terceirizados, dispondo de diversas funcionalidades como criar máquinas virtuais e migração entre servidores de virtualização (OLIVEIRA, 2017).

Dispositivos IoT normalmente possuem limitações de processamento, armazenamento, baixa confiabilidade e alta taxa de erros, por isso elementos centralizadores com maior poder de processamento e disponibilidade devem compor a arquitetura da solução, ficando a cargo desses elementos centralizadores tratar os dados recebidos, enquanto que os dispositivos IoT apenas realizam a aquisição e envio dos dados. A adoção da computação em nuvem ajuda a reduzir custos operacionais, a executar a infraestrutura da aplicação com mais eficiência e a escalonar conforme as demandas do cliente mudam. (OLIVEIRA, 2017)

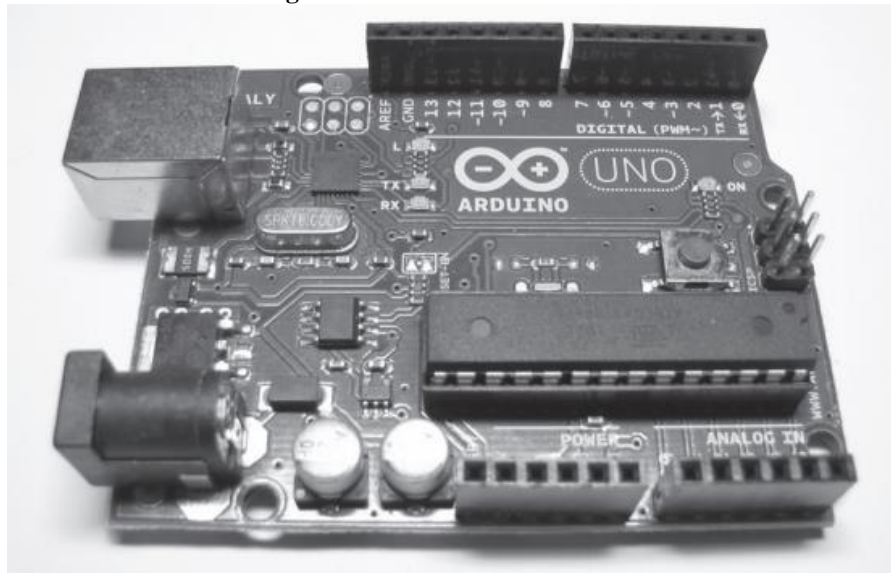
2.2 ARDUÍNO

O *Arduino* é uma plataforma de prototipagem de *hardware* e *software open source* que surgiu na Itália em 2005 (ALVES, 2013). Possui como componente central um microcontrolador da família ATmega. O microcontrolador é muito semelhante a um computador de pequeno porte, pois compartilha de diversas similaridades como: memória RAM, memória *flash*, *clock*, entre outros. Possui também um software que consiste em uma linguagem de programação e um ambiente derivados do *Processing*¹ (OLIVEIRA e ZANETTI,

¹ O *Processing* é uma linguagem de programação open source com ambiente de desenvolvimento integrado, com aplicações em artes eletrônicas e comunidades de projetos visuais.

2015). Foi desenvolvido para ser uma ferramenta de prototipagem eletrônica de baixo custo, que permitiria que pessoas não especialistas em programação e/ou em eletrônica pudessem desenvolver aplicações de objetos e ambientes interativos. Na **Figura 1** abaixo é mostrada um modelo de placa *Arduino*.

Figura 1 – Placa do Arduino UNO.



Fonte: McRoberts, 2011.

A placa possui pinos de entrada/saída que permitem a conexão com outros circuitos e sensores:

- 14 pinos de entrada e saída digital (pinos 0-13): pinos com funcionamento de acordo com as definições no sketch criado na IDE.
- 6 pinos de entradas analógicas (pinos A0 - A5): Esses pinos são dedicados a receber valores analógicos, por exemplo, a tensão de um sensor. O valor a ser lido deve estar na faixa de 0 a 5 V.
- 6 pinos de saídas analógicas (pinos 3, 5, 6, 9, 10 e 11): São pinos digitais que podem ser programados para serem utilizados como saídas analógicas, utilizando modulação PWM.

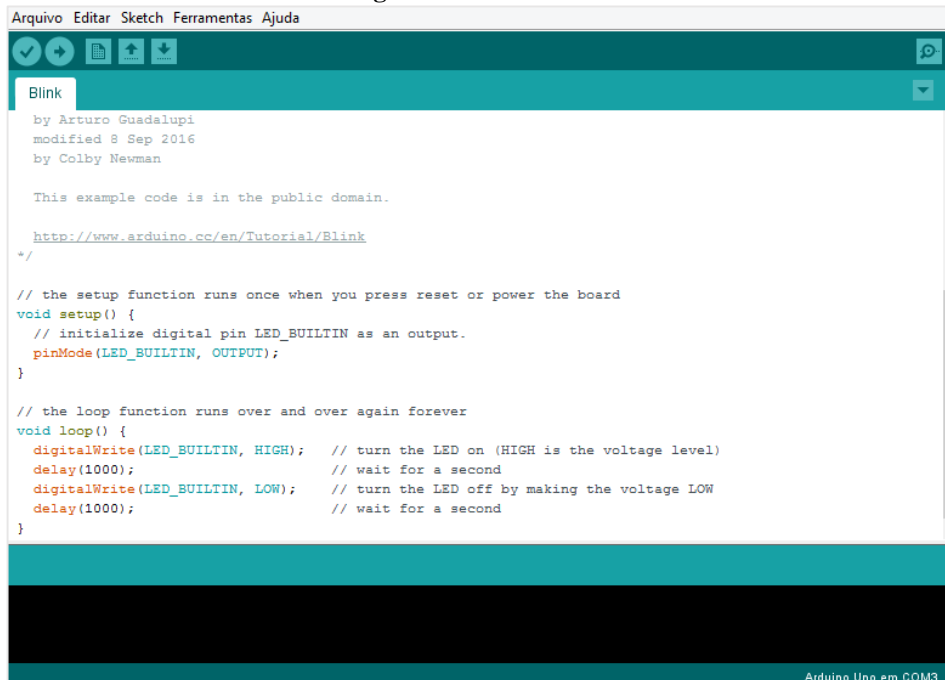
A alimentação da placa pode ser feita a partir da porta USB do computador ou através de uma fonte de alimentação elétrica CA/CC, na qual se recomenda-se uma tensão de 9 a 12 volts.

2.2.1 Programação na IDE do Arduino UNO

Devido ao seu alto grau de abstração, a IDE do Arduino possibilita que usuários com pouca experiência no desenvolvimento de *softwares* possam criar projetos. Na IDE os

programas “*sketches*” podem ser escritos e carregados para a placa, que por sua vez irá interagir com o que estiver conectado aos seus pinos. A seguir, na **Figura 2**, pode-se verificar a IDE do *Arduino UNO*.

Figura 2 - IDE Arduino.



The image shows a screenshot of the Arduino IDE interface. The title bar reads 'Arquivo Editar Sketch Ferramentas Ajuda'. The main window displays the 'Blink' sketch code. The code includes comments about the author (Arturo Guadalupi, modified 8 Sep 2016, by Colby Newman) and a public domain notice. The code defines a setup function to initialize the LED_BUILTIN pin as an output and a loop function that turns the LED on for 1000ms and off for 1000ms. The IDE status bar at the bottom right indicates 'Arduino Uno em COM3'.

```

Arquivo Editar Sketch Ferramentas Ajuda
Blink
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino Uno em COM3

```

Fonte: Própria, 2020.

Existem elementos comuns a qualquer programa desenvolvido na IDE do *Arduino*:

- **Variáveis:** são declaradas de forma global (fora dos *loops* do programa) ou local (dentro de algum *loop* específico);
- **Funções:** são blocos de código criados pelo programador e irão realizar comandos definidos;
- **Bibliotecas:** são incluídas no início do programa e são compostas por funções desenvolvidas pela comunidade de programadores ou criadores da plataforma.

Os programas desenvolvidos na IDE do *Arduino* são divididos em duas partes principais:

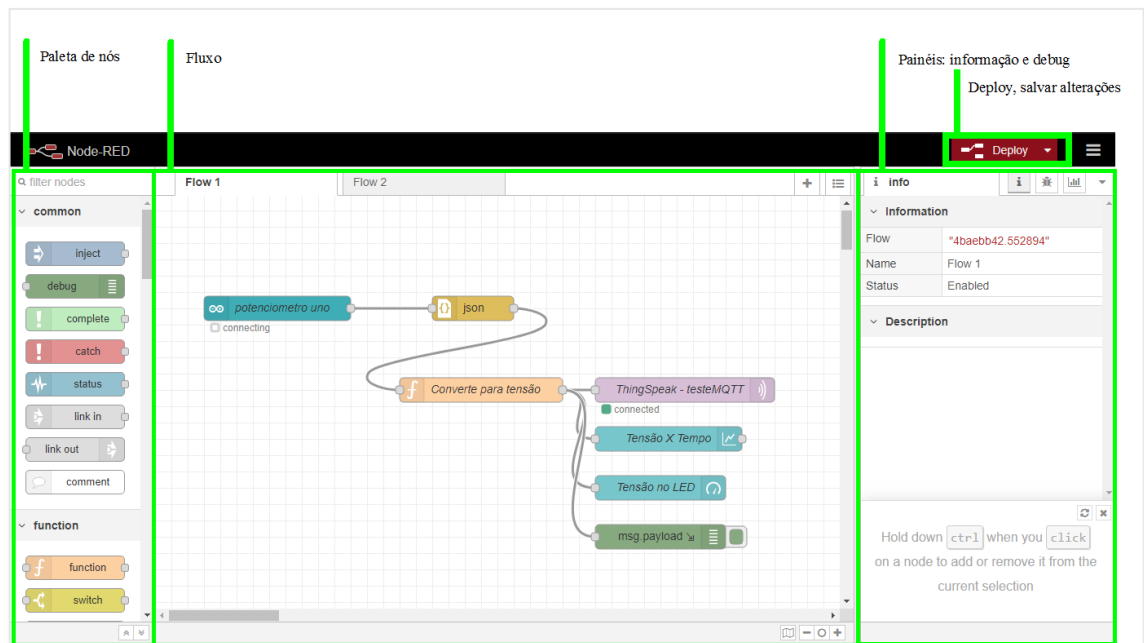
- ***void setup()*:** inicializa as variáveis, modos dos pinos, velocidade de transmissão serial, entre outros;
- ***void loop()*:** as instruções contidas nesse loop serão executadas repetidamente;

2.3 NODE-RED

É um editor de fluxos *open source* desenvolvido pela IBM em *JavaScript* através da utilização do *framework NODE-JS*. Viabiliza a prototipagem de projetos de IoT, interligando a internet (*softwares*) às coisas (*hardware*). Pelo seu alto nível de abstração, o *NODE-RED* permite utilizar tecnologias complexas sem a necessidade de se aprofundar até os mínimos detalhes em todas elas, pois a maior parte da programação é desenvolvida através de funções pré-definidas “nós”, onde são necessários apenas algumas configurações de parâmetros.

O editor de fluxos de *NODE-RED* consiste em uma simples interface em HTML, acessível de qualquer navegador, onde arrastando e conectando nós entre si, é possível definir um fluxo que ofereça um serviço. Pelo fato dos nós serem guardados em formato JSON, eles podem facilmente ser importados e exportados.

Figura 3 – Fluxo em NODE-RED.



Fonte: Própria, 2020.

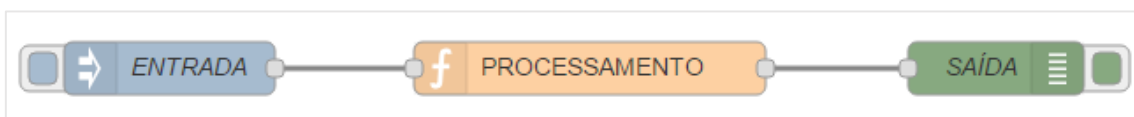
Como pode-se facilmente verificar esse ambiente é composto por:

- **Paleta de nós:** exibe todos os nós que temos disponíveis na instalação. Existe um repositório de nós desenvolvido por outros usuários e pode-se até criar e instalar nós personalizados.
- **Fluxo:** permite arrastar nós da paleta e conectá-los. Dessa forma, cria-se o fluxo da operação.
- **Painéis de informação e debug:** exibe detalhes da configuração de cada nó, além das mensagens que fluem nos nós.
- **Deploy, salvar alterações:** Compilação do projeto *NODE-RED*. Botão localizado no canto superior direito da interface *web*.

O *NODE-RED* dispõe de basicamente três tipos de nós, que podem ser classificados em:

- **Nós de entrada:** permitem que os dados sejam inseridos em um aplicação.
- **Nós de saída:** permitem o envio de dados para fora de uma aplicação.
- **Nós de processamento:** permitem a manipulação dos dados que entram no nó para posteriormente enviar novos dados para o próximo estágio.

Figura 4: Nós do NODE-RED.



Fonte: <https://www.filipeflop.com/blog/primeiros-passos-node-red-arduino-uno/>.

2.3.1 Termos recorrentes

- *Background:* Aplicação executada por um código fonte não visível na *dashboard*. Corresponde à programação em blocos da interface *web*.
- *Client-ID:* Identificação de um usuário do sistema.
- *Dashboard:* Painel de instrumentos gerados a partir dos blocos.
- *Deploy:* Compilação do projeto *NODE-RED*. Botão localizado no canto superior direito da interface *web*.

- *Done*: Botão de finalização das janelas de edição de cada bloco a ser programado.
- *End-nodes*: Dispositivos de aplicação embarcados com sensor, microcontrolador *Arduino* por exemplo.
- *Flow*: Estrutura de blocos de programação *NODE-RED* que se interligam para prover um caminho à informação.
- *Gauge*: Mostrador de valores no *dashboard*.
- *GitHub*: *Site* para se disponibilizar projetos à comunidade por meio de repositórios.
- *Groups*: Blocos que podem conter outros diversos blocos em um *dashboard*.
- *Interface web NODE-RED*: Interface de programação onde serão alocados os blocos de programação.
- *json*: Sigla para *JavaScript Object Notation*. Estrutura de objetos para envio de informações indexadas.
- *Manage Palette*: Gerenciamento de *plugins* disponíveis e instalados no *NODE-RED*.
- *Menu de blocos*: Localizado ao lado esquerdo da interface web contendo os blocos possíveis e disponíveis para a programação *NODE-RED*.
- *Menu principal*: Botão característico de Menu localizado no canto superior direito da interface *web*.
- *MQTT*: Sigla para *Message Queueing Telemetry Transport*.
- *MQTT broker*: Serviço que gerencia usuários, senhas, tópicos e aplicações de uma comunicação do tipo *MQTT*.
- *MQTT client*: Conecta-se ao servidor *MQTT broker* para receber e/ou enviar dados.
- *MQTT server*: Servidor *MQTT* propriamente dito.
- *msg.payload*: Variável principal de um *flow* que armazena as variáveis a serem transmitidas de um bloco a outro.
- *NODE-RED*: Servidor para criação de aplicação. Pode ser instalado localmente ou em nuvem.
- *Parsing*: Análise ou decomposição de uma variável em outra desejada.
- *Password*: Senha de usuário correspondente ao cadastro do usuário.

- *Port*: Porta de um servidor em que se está disponibilizado um serviço.
- *Publish*: Ação de publicar em um determinado *Topic* para enviar dados da aplicação ao *end-node*.
- *QoS*: Sigla para *Quality of Service*.
- *Repositório*: Lugar em que se encontram os arquivos de um projeto no *GitHub*, por exemplo.
- *Subscribe*: Ação de se inscrever em um determinado *Topic* para receber os dados do *end-node* à aplicação.
- *Tab*: Cada *Tab* representa uma página dentre as diversas possíveis em um *dashboard*.
- *Topic*: Endereçamento em que se está disponibilizado o acesso à aplicação por meio do *MQTT broker*.
- *Username*: Nome de usuário correspondente ao cadastro do usuário no *Network Server*.
- *Update*: Botão de atualização das janelas de edição de cada bloco a ser programado.
- *Widgets*: Blocos mostradores em um *dashboard*, o *Gauge* é um deles, por exemplo.
- *.zip*: Estrutura de compactação de arquivos

O *NODE-RED* é compatível com diversos microcontroladores como: *Arduino*, *Raspberry Pi*, *ESP8266*, *ESP32*. Tem suporte à protocolos essenciais nos desenvolvimentos IoT como: *HTTP*, *MQTT*, *TCP/IP*, entre outros.

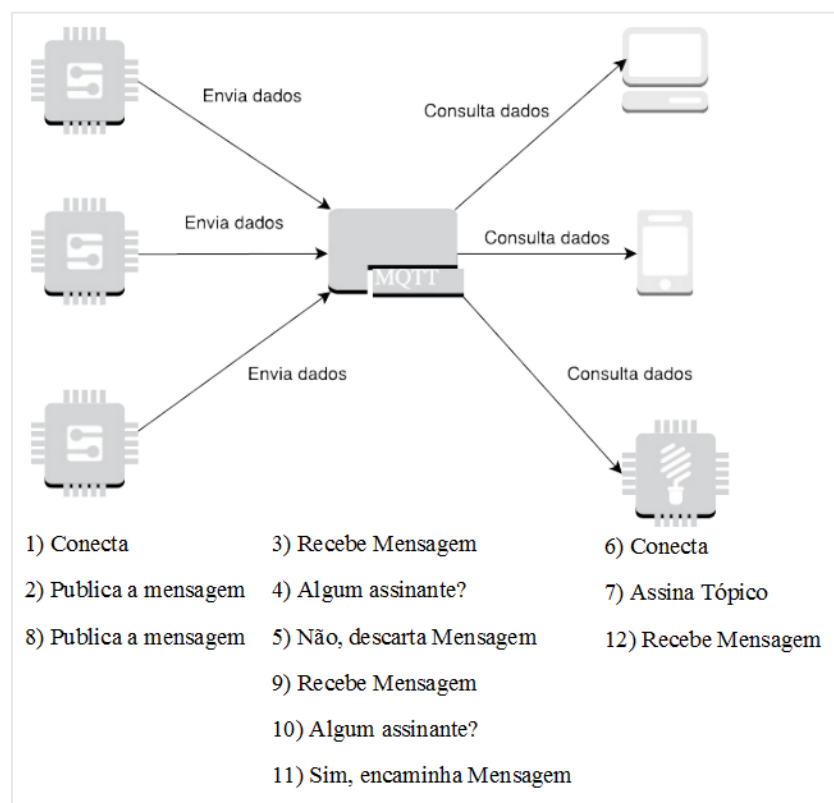
2.4 MQTT

O protocolo *MQTT Message Queing Telemetry Transport*, foi criado por Andy Stranford-Clark (IBM) e Arlen Nipper em 1999. É um protocolo de mensagens leve, *open source*, simples e projetado para ser de fácil implementação e administração. Ideal para cenários com alta latência, instabilidade na comunicação e casos em que são exigidos pacotes de dados e/ou largura de banda mínimos. Apesar de todas essas vantagens o protocolo *MQTT* garante qualidade de serviço, segurança e flexibilidade. Pela sua simplicidade pode ser usado em *hardwares* com poder de processamento limitado, como os microcontroladores.

É baseado em TCP/IP e possui *payload* menor que o HTTP. Essas características tornam o MQTT ideal para a comunicação M2M (*machine to machine*) e está se tornando rapidamente um padrão para IoT pois suas particularidades favorecem o envio de mensagens em ambientes instáveis (SOUZA, 2018).

Sua arquitetura está baseada em três componentes básicos: *publisher*, *subscriber* e *broker*. Para que uma informação possa ser recebida por um dispositivo é necessário que ele esteja registrado em um *broker*, isto é, seja reconhecido como um assinante (*subscriber*) de um determinado tópico para onde as informações publicadas são encaminhadas. De maneira análoga, para que em dispositivo possa publicar dados em um *broker* é necessário que esteja registrado como publicador (*publisher*) no qual poderá enviar mensagens para um tópico específico no *broker*. A **Figura 5** a seguir mostra a operação do MQTT.

Figura 5: Funcionamento básico do MQTT.



Fonte: Própria, 2020.

O protocolo MQTT possui três níveis de Qualidade de Serviço, no qual define-se com qual nível de persistência as mensagens serão entregues ao assinante. Abaixo estão listadas os níveis de Qualidade de Serviço e quando utilizá-los

1. No máximo uma vez (QoS 0):

- As conexões são estáveis;
- A perda de alguma mensagem é aceitável;
- As mensagens não precisam ser persistidas;

2. Ao menos uma vez (QoS 1):

- Todas as mensagens precisam ser recebidas, mesmo que eventualmente duplicadas;
- O tráfego adicional gerado por eventuais mensagens duplicadas é mais aceitável do que o tráfego gerado pelo QoS 2;

3. Exatamente uma vez (QoS 2):

- A precisão no recebimento da mensagem é crítico;
- A sobrecarga de tráfego e processamento é justificável.

2.4.1 Termos recorrentes

- *Message*: pacote de dados trafegando entre clientes e *broker*
- *Topic*: repositório de dados único acessível aos clientes, instantâneo e sem histórico
- *Publish*: ação de enviar para o *broker* uma mensagem para um determinado tópico
- *Subscribe*: ação de solicitar receber mensagens de um determinado tópico
- *Unsubscribe*: ação de solicitar não receber mais mensagens de um determinado tópico
- *Payload*: dados enviados na mensagem
- *QoS (Quality of Service)*: nível de serviço sobre o qual o *broker* deve receber e/ou transmitir os dados
 - 0 - *At most once* (no máximo uma vez)
 - 1 - *At least once* (pelo menos uma vez)
 - 2 - *Exactly once* (exatamente uma vez)

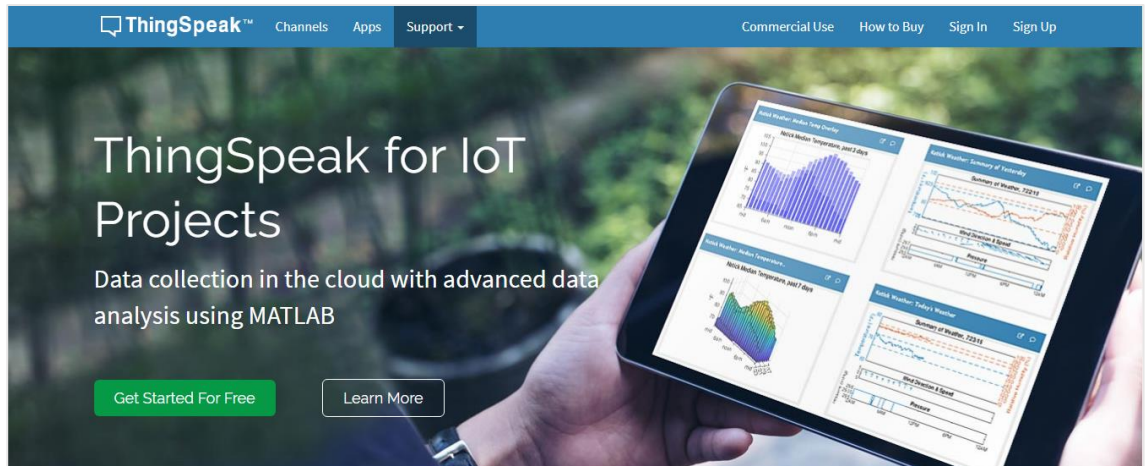
- *Last will / will message* (última vontade): mensagem enviada pelo *broker* caso o cliente perca a conexão
- *Ping*: teste periódico de conexão cliente-broker
- *Session*: período no qual o cliente está conectado no *broker*
- *Retain*: indica se uma mensagem deve ser mantida pelo *broker* até uma próxima ser recebida
- *Keep alive*: intervalo de tempo máximo sem comunicação. O cliente deve enviar *ping* antes de atingir esse intervalo

2.5 THINGSPEAK

O *Thingspeak* é uma plataforma online fornecida pela *Mathworks*, na qual estão disponibilizados diversos serviços relacionados a IoT como armazenamento de dados, API para comunicação, suporte ao protocolo M2M (*machine to machine*), MQTT (*Message Queeing Telemetry Transport*), integração com o *software* MATLAB, plataformas de visualização de dados, integração com *Twitter*, disparo de ações por eventos, disparo de ações periódicas, entre outros. As diversas funcionalidades oferecidas pela *Mathworks* são intuitivas e agregam valor às aplicações.

Uma utilização típica do *Thingspeak* é como *datalog*, no qual registram-se grandezas obtidas por sensoriamento. O emprego do *datalog* possibilita o monitoramento ambiental e aplicações industriais. A plataforma *ThingSpeak* disponibiliza canais para coleta e tratamento dos dados. Após o registro dos dados, o *Thingspeak* disponibiliza a análise por *softwares* matemáticos, visualização de gráficos indicativos do comportamento dos dados e oferece *plugins* para criar arquivos HTML, *JavaScript* ou CSS personalizados (OLIVEIRA, 2017).

Figura 6: Site do ThingSpeak



Fonte: <https://thingspeak.com/>.

Na página inicial do site da *Mathworks* (<https://thingspeak.com/>) se encontram as opções de cadastro e acesso à conta do usuário. A seguir serão apresentados as etapas para a criação de um canal na plataforma *ThingSpeak* para recebimento de dados.

2.5.1 Cadastro na plataforma

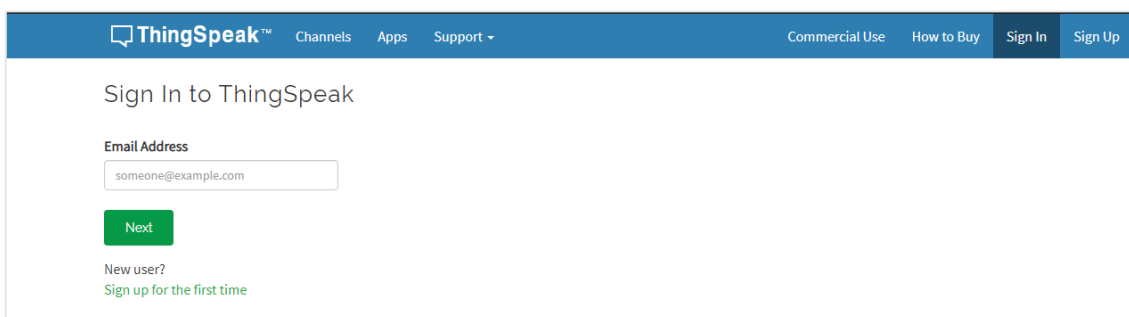
Para realizar o cadastro de um novo usuário na plataforma, deve-se clicar no canto superior da tela em *Sign up*. A **Figura 7** a seguir exibe as informações requisitadas que deverão ser preenchidas. Em seguida, deve-se confirmar o cadastro no *e-mail* informado.

Figura 7: Cadastro no ThingSpeak

Fonte: https://thingspeak.com/users/sign_up.

Após a confirmação da autenticidade da conta do usuário, deve-se clicar em *Sign in* para ter acesso à área restrita ao usuário. Em seguida, preencher o campo do *e-mail* previamente cadastrado. Ver **Figura 8**.

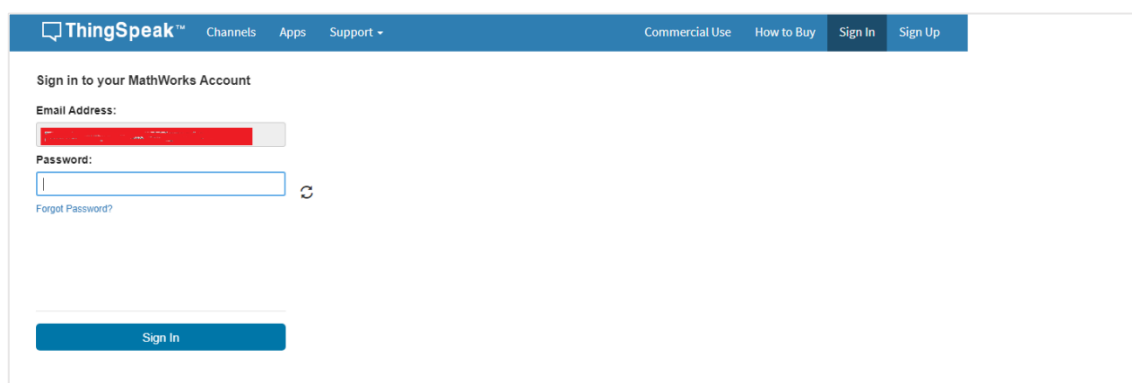
Figura 8: Acesso ao ThingSpeak

The screenshot shows the 'Sign In to ThingSpeak' page. At the top, there is a navigation bar with the ThingSpeak logo and links for 'Channels', 'Apps', 'Support', 'Commercial Use', 'How to Buy', 'Sign In', and 'Sign Up'. The main heading is 'Sign In to ThingSpeak'. Below it, there is a form with an 'Email Address' field containing 'someone@example.com' and a green 'Next' button. At the bottom, there is a link for 'New user? Sign up for the first time'.

Fonte: https://thingspeak.com/users/sign_up.

Logo após, deve-se colocar a senha cadastrada. Ver **Figura 9**.

Figura 9: Password ThingSpeak.

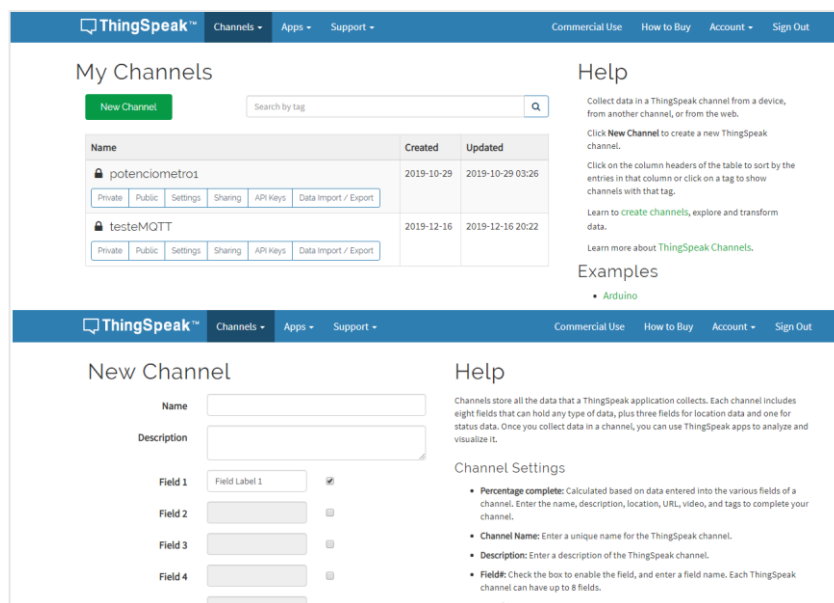
The screenshot shows the 'Sign in to your MathWorks Account' page. At the top, there is a navigation bar with the ThingSpeak logo and links for 'Channels', 'Apps', 'Support', 'Commercial Use', 'How to Buy', 'Sign In', and 'Sign Up'. The main heading is 'Sign in to your MathWorks Account'. Below it, there is a form with an 'Email Address' field (partially obscured by a red bar) and a 'Password' field. There is a 'Forgot Password?' link and a 'Sign In' button at the bottom.

Fonte: <https://thingspeak.com/>.

2.5.2 Criação de canais

Para criação de um canal no *ThingSpeak* deve-se estar cadastrado na plataforma. Na página inicial clique em *Channels > My Channels*. Agora pode-se criar um novo canal clicando em *New Channel*; e configurar os campos conforme as especificidades da aplicação a ser desenvolvida. Ver **Figura 10**.

Figura 10: Criando canais.



Fonte: <https://thingspeak.com/channels/new/>.

Após configurado, o novo canal estará pronto para receber dados enviados por diversas plataformas IoT. Nesse ambiente os dados recebidos poderão ser visualizados por meio de gráficos, painéis de instrumentos, entre outros. Também estará disponível a opção de importar/exportar dados de um banco de dados próprio. Pode-se publicar em um canal até oito diferentes informações enviadas em um mesmo procedimento de *publish*.

2.5.3 Disparo de ações por eventos

Na página inicial do *ThingSpeak*, na aba *Apps* existem diversos recursos que podem agregar valor à aplicação. Nessa aba a *Mathworks* disponibiliza serviços que tornam possível definir filtros, alvos e notificações específicas baseadas nas atividades de um canal.

O processo de criar notificações inteligentes envolve diversas etapas que consistem no uso do serviço *React* para disparar uma requisição HTTP para o serviço *web* chamado *IFTTT* no *ThingHTTP*. Na documentação disponibilizada no *ThingSpeak* estão detalhados todas as etapas necessárias para a utilização do recurso de notificação por *e-mail*, *twitter*, etc.

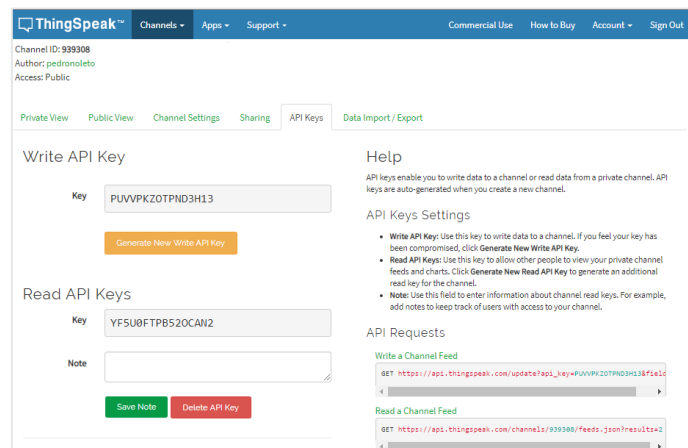
2.5.4 Publicando dados em um canal

Todo canal possuirá uma identificação própria chamada *ID Channel* essa identificação será muito importante para os processos de *publish* que eventualmente forem feitos nesse canal. Para que um dispositivo realize a publicação de dados em um determinado tópico é necessário

que tenha a permissão para tal. Uma **chave de escrita** (*WRITE API Key*) é requisitada e deverá ser fornecida para habilitar o dispositivo a publicar no tópico correto utilizando o *broker* MQTT do *ThingSpeak*.

De maneira análoga existem **chaves de leitura** (*READ API Key*) são necessárias para habilitar um dispositivo qualquer a assinar um determinado tópico correspondente a um canal do *ThingSpeak*, dessa forma terá acesso aos dados e gráficos contidos em um canal privado. Na aba *API Keys* do canal estão disponíveis as chaves de escrita e leitura conforme exibido na **Figura 11**.

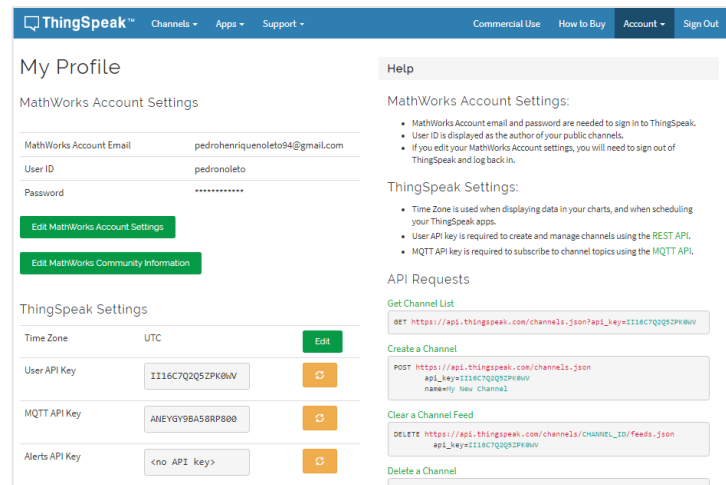
Figura 11: Chaves ThingSpeak



Fonte: https://thingspeak.com/channels/939308/api_keys.

Nas configurações de conta do usuário encontra-se uma chave chamada **MQTT API Key**, essa chave é utilizada no processo de conexão com o *broker* MQTT. Para obtenção da permissão de conexão junto ao *broker* é necessário uma senha, nesses casos a senha é a **MQTT API Key**. Ver **Figura 12**.

Figura 12: Chaves MQTT API Key



Fonte: <https://thingspeak.com/account/profile>.

Conforme explicitado anteriormente, para realizar uma conexão de um cliente MQTT junto a um *broker* deve-se proceder da seguinte maneira:

- *Host*: mqtt.thingspeak.com;
- *Port*: 1883;
- *Username*: qualquer valor (recomenda-se preencher com o nome de usuário no *ThingSpeak*);
- *Password*: MQTT API Key;
- *Cliente ID*: não necessário;
- *Topic*: channels/{channelID}/publish/{WRITE API Key};

2.5.5 Aplicativos para smartphone

Existem diversos aplicativos que são compatíveis com o protocolo MQTT e podem ser facilmente configurados para receber os dados provenientes dos canais criados no *ThingSpeak*. Nesses aplicativos podem ser utilizadas *dashboards* que disponibilizam a assinatura de tópicos, a visualização dos dados em tempo real e em intervalos de tempo definidos, geração de gráficos e publicação de dados através do protocolo MQTT.

3 DESENVOLVIMENTO DA APLICAÇÃO

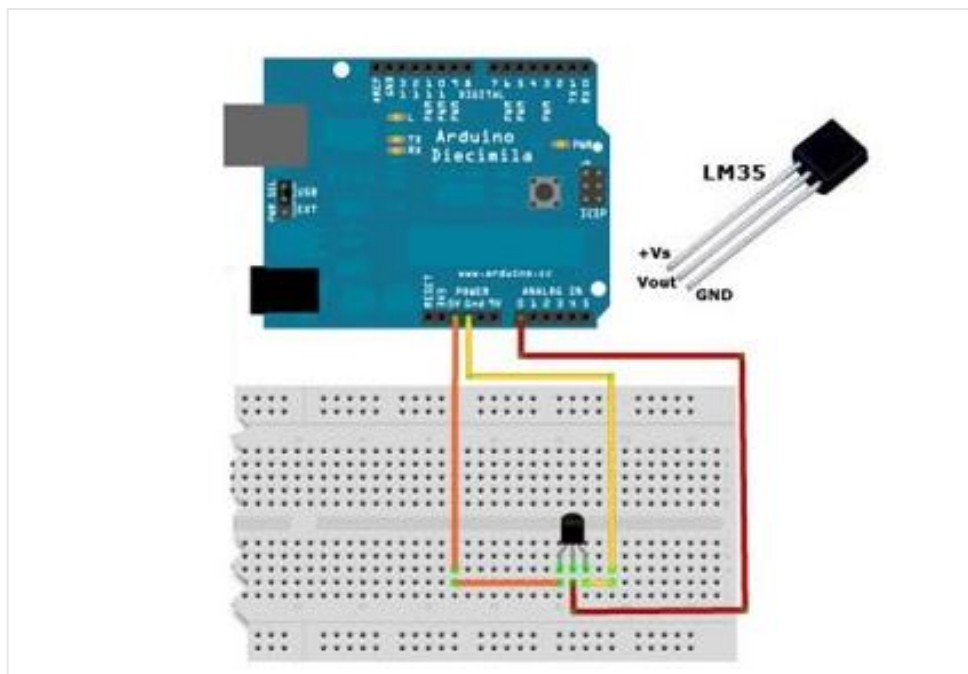
3.1 MONTAGEM DO CIRCUITO

Para o desenvolvimento desta aplicação serão usados:

- Sensor de temperatura LM35;
- *Jumpers*;
- Placa *Arduino UNO*;
- *Proboard*;
- Cabo USB;
- *Notebook*;

Na montagem do circuito, o microcontrolador se conecta ao LM35 através da protoboard e dos jumpers. A **Figura 13** apresenta o esquema de montagem do circuito.

Figura 13: Circuito montado.



Fonte: <https://ricardo-sequeira.com/arduino-lm35/>.

O LM35 é um sensor de temperatura que apresenta uma resposta linear, com $10 \text{ mV}/^{\circ}\text{C}$. Opera num intervalo de -55°C à 150°C , quando utilizado no modo escala completa a leitura de

tensão irá variar entre -55 mV e 150 mV. Segundo o *datasheet* do LM35, na configuração básica de utilização o sensor possui um *offset* de temperatura, que corresponde a 2°C quando a sua saída é de 0V. Portanto, essa aplicação só está habilitada a monitorar temperaturas a partir de 2°C.

Conforme apresentado na **Figura 13**, os terminais externos estarão conectados à alimentação +5v e ao GND do Arduino, enquanto que o terminal central estará ligado ao pino de entrada analógica, nesse caso o pino A0.

Internamente o *Arduino* possui um conversor A/D de 10 bits. Dessa forma, os valores lidos na porta estarão contidos no intervalo de 0 à 1023 (ou seja, 1024 valores possíveis), onde o 0 corresponde a 0 V e o 1023 corresponde a 5 V. O LM35 não necessita de qualquer calibração externa para fornecer os dados; então para se obter a temperatura ambiente aferida pelo LM35 é necessário realizar os seguintes cálculos.

Seja **V** a tensão na porta analógica e **A** o valor lido na porta analógica. Dessa forma temos que:

$$V = A \frac{5}{1024} \quad (1)$$

Por fim, seja **T** a temperatura ambiente e sabendo que:

$$1^\circ\text{C} = 10\text{mV} \quad (2)$$

Desse modo, a temperatura ambiente é dada pela seguinte equação:

$$T = A \frac{5}{1024} * \frac{1}{0,01} \quad (3)$$

$$T \cong A * 0,4883 \quad (4)$$

Após a completa instalação da IDE do Arduino deve-se conectá-lo ao *notebook* através do cabo USB e compilar o exemplo *StandardFirmata*. O exemplo *StandardFirmata* é um dos muitos exemplos disponibilizados pela biblioteca *Firmata* e pode ser facilmente encontrado em: File > Exemplos > *Firmata* > *StandardFirmata*.

A biblioteca *Firmata* implementa o protocolo *Firmata*, que é um protocolo de comunicação com microcontroladores à partir de um software em um computador. Esse protocolo pode ser implementado em *firmware* em qualquer arquitetura de microcontrolador, bem como software em qualquer pacote de computador. A utilização do protocolo *Firmata* na IDE do *Arduino* habilita a integração entre a placa do *Arduino* e o *NODE-RED* instalado localmente.

Controlar o microcontrolador por meio do protocolo *Firmata* tem algumas vantagens que irão contribuir com o desenvolvimento dessa aplicação, dentre as quais podem se destacar:

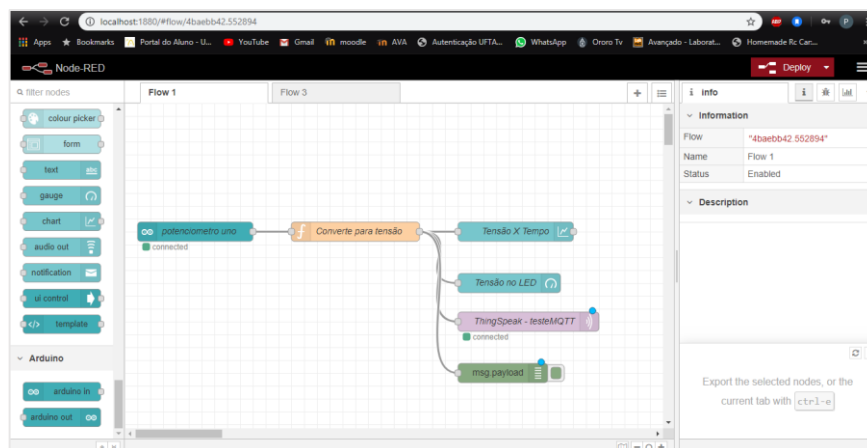
- O programa implementado não estará limitado pela memória RAM ou *Flash* do microcontrolador;
- O programa poderá ser implementado em qualquer linguagem e não estará restrito apenas ao C++, dessa forma diversas funcionalidades poderão ser incluídas no projeto, facilitando sua execução e produzindo melhores resultados;

3.2 PROGRAMAÇÃO NO NODE-RED

Para esta aplicação foi definido que o editor *NODE-RED* iria realizar toda a comunicação entre o microcontrolador e a plataforma IoT na nuvem, além da geração dos gráficos dos dados em tempo real. O processo de instalação do *NODE-RED* está detalhado no **APÊNDICE A**.

Foi desenvolvido um fluxo no *NODE-RED* responsável pela coleta, decodificação, tratamento e visualização dos dados. Ver **Figura 14**.

Figura 14: Fluxo no NODE-RED



Fonte: Própria, 2020.

O *NODE-RED* já disponibiliza um conjunto abrangente de nós na sua instalação, entretanto alguns nós precisam ser instalados manualmente. Para isso deve-se clicar no canto superior direito da tela no menu, em seguida selecionar a opção *manage palette*. Logo após, deve-se clicar em *install*, digitar “arduino” e instalar o pacote **node-red-node-arduino**. Por fim, os nós do novo pacote instalado estarão adicionados à paleta de nós. De maneira análoga, foi instalado o **pacote node-red-dashboard**.

No fluxo desenvolvido foram acrescentados apenas os dois pacotes supracitados, os demais nós já estavam disponíveis na paleta de nós. Os dados lidos na porta A0 são recebidos e tratados no nó (*arduino in*), em seguida os dados são enviados para o próximo nó (*function*), onde é feita a conversão do valor enviado pelo conversor A/D para graus Celsius. Por fim, a informação é enviada para nós de debug, geração de gráficos e para a plataforma *ThingSpeak*, onde a *stream* de dados será armazenada, visualizada e analisada. Os fluxos desenvolvidos podem ser facilmente importados por outros usuários, desde que se tenha o arquivo *flow.json*. O conteúdo completo do fluxo desenvolvido está disponível no **APÊNDICE B**.

3.3 UTILIZAÇÃO DA PLATAFORMA THINGSPEAK

Para essa aplicação foi criado o canal **testeMQTT** na plataforma *ThingSpeak*, para o qual os dados do monitoramento foram enviados. Após a montagem do circuito, foram realizados testes para avaliar a integração entre as plataformas envolvidas na aplicação. Ao fim de vários ajustes foi realizado o monitoramento da temperatura ambiente em minha residência durante alguns dias. Ver **Figura 15**.

Figura 15: Criando um novo canal no ThingSpeak.

Fonte: Própria, 2020.

Nesse processo o Arduino foi alimentado pelo *notebook*, no qual estava instalado o *NODE-RED*, os dados aferidos pelo sensor foram transmitidos via USB ao *notebook* por meio do protocolo *Firmata*. Desse modo, os dados eram prontamente interpretados pelo *NODE-RED* que realizava a recepção, conversão, análise gráfica e envio para a plataforma *ThingSpeak*.

3.4 UTILIZAÇÃO DE APLICATIVOS PARA SMARTPHONE

A utilização da plataforma *ThingSpeak* em conjunto com aplicativos com suporte ao protocolo MQTT, como o *IoT Monitor Widget*, confere mais mobilidade à aplicação. Nesses aplicativos era realizada a assinatura dos tópicos no canal do *ThingSpeak*, assim as atualizações ocorriam simultaneamente na plataforma e nos *dashboards* dos aplicativos. Ver **Figura 16**.

Figura 16: Assinando canais por smartphone.



Fonte: Própria, 2020.

3.5 IMPLEMENTAÇÃO DO DISPARO DE AÇÕES POR EVENTOS

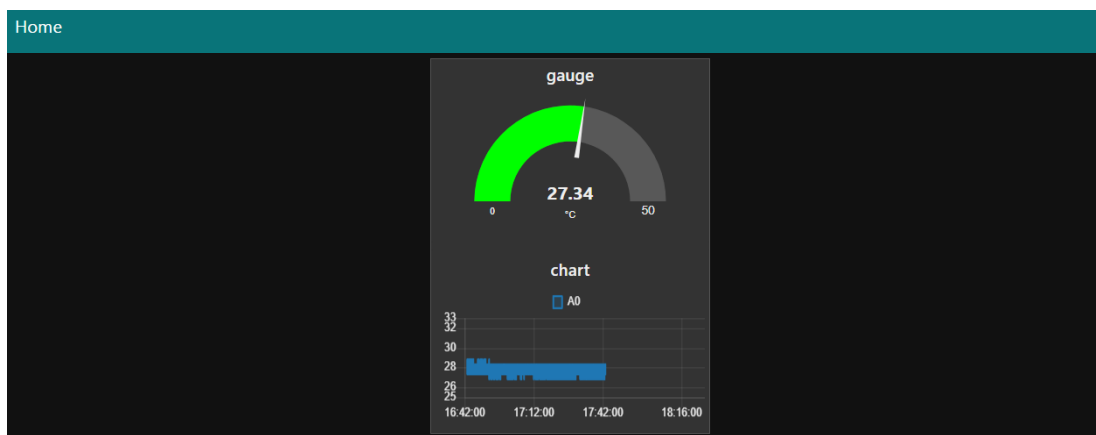
Por fim, as funcionalidades de disparo de ações baseadas nos dados foram incorporadas à aplicação. Foi configurada uma condição no serviço *React* através dos procedimentos citados anteriormente. Após configurada, a aplicação estará habilitada para o disparo de ações baseadas nos dados aferidos. Nesse caso, foi configurado o envio de notificações por *e-mail* quando a temperatura ambiente ultrapassasse os 33°C.

4 RESULTADOS

4.1 VISUALIZAÇÃO GRÁFICA DOS DADOS EM TEMPO REAL

As variações observadas na temperatura eram imediatamente visualizadas nos gráficos gerados pelo *NODE-RED*. Devido à alta transmissão de dados e o comportamento caracteristicamente caótico da temperatura em torno de um valor central, observou-se que o gráfico gerado apresentou um aspecto difuso e de difícil interpretação. Ver **Figura 17**.

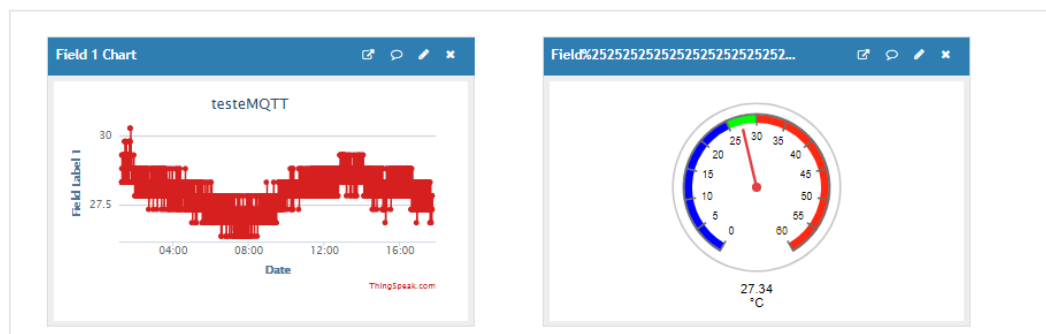
Figura 17: Análise gráfica no NODE-RED.



Fonte: Própria, 2020.

O comportamento caótico nos gráficos da temperatura também foram observados no canal testeMQTT. Apesar de a taxa de transmissão dos dados ser muito menor, a cada 15 segundos, à medida que o tempo de monitoramento aumentava o comportamento caótico da temperatura era novamente observado. Ver **Figura 18**.

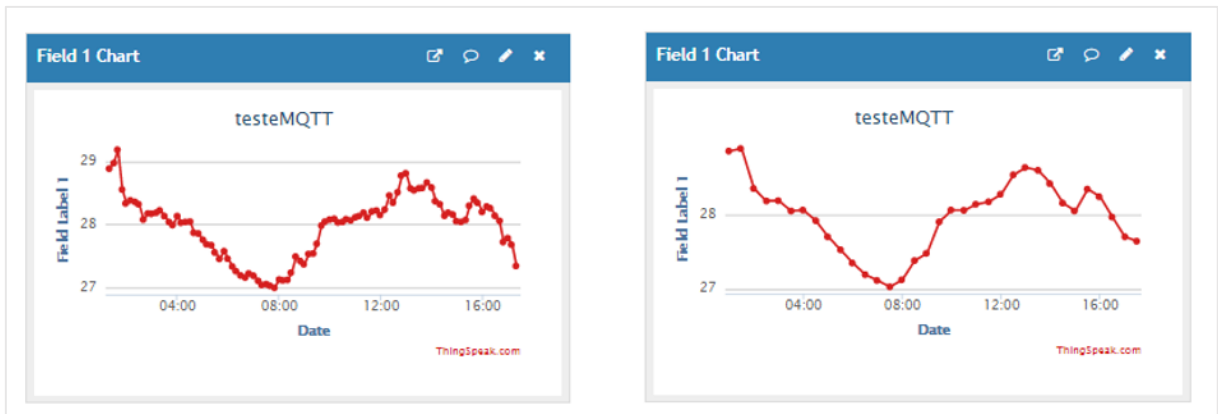
Figura 18: Análise gráfica no ThingSpeak sem tratamento.



Fonte: Própria, 2020.

À vista disso, utilizou-se os recursos disponíveis na plataforma *ThingSpeak* para a suavização das curvas. Por meio da utilização de valores médios em intervalos de tempo previamente definidos, foi possível obter gráficos com curvas mais suavizadas e que exibiam com mais clareza as variações da temperatura ao longo do dia. Ver **Figura 19**.

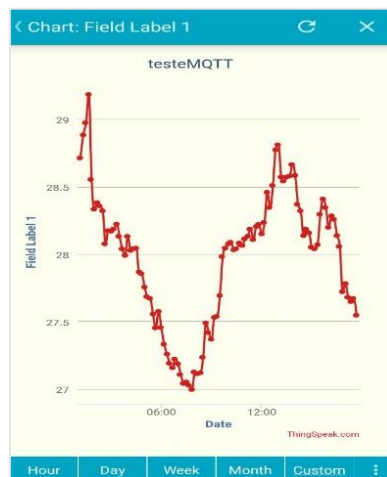
Figura 19: Análise gráfica no ThingSpeak com tratamento.



Fonte: Própria, 2020.

Os aplicativos possuem diversas funcionalidades dentre as quais se destacam a suavização das curvas por processos de média em intervalos definidos e a geração de gráficos à partir de uma data pré-estabelecida. A **Figura 20** apresenta os gráficos obtidos no dia 15 de fevereiro de 2020.

Figura 20: Análise gráfica no IoT Monitor Widget.



Fonte: Própria.

As ações consistem no envio de mensagens de texto para o e-mail cadastrado quando a temperatura ultrapassasse os 33 °C. A **Figura 21** apresenta o conteúdo de uma notificação recebida por *e-mail*.

Figura 21: Notificação baseada em evento.



Fonte: Própria, 2020.

Todos os dados enviados ao canal do *ThingSpeak* podem ser baixados em formato CSV através da aba Data Import/Export no canal testeMQTT. Dessa forma, podem ser interpretados em diferentes programas como Excel, Google Sheets, Apple Numbers, OpenOffice Calc, etc. Nesse caso temos os dados baixados exibidos no Excel. Ver **Figura 22**.

Figura 22: Exibindo dados no Excel.

| | A | B | C | D | E | F | G | H | I | J | K |
|-------|--|---|---|---|---|---|---|---|---|---|---|
| 19483 | 2020-02-20 15:27:39 -03,19482,24.41,,, | | | | | | | | | | |
| 19484 | 2020-02-20 15:27:54 -03,19483,24.9,,, | | | | | | | | | | |
| 19485 | 2020-02-20 15:28:09 -03,19484,24.41,,, | | | | | | | | | | |
| 19486 | 2020-02-20 15:28:24 -03,19485,23.93,,, | | | | | | | | | | |
| 19487 | 2020-02-20 15:28:39 -03,19486,24.41,,, | | | | | | | | | | |
| 19488 | 2020-02-20 15:28:55 -03,19487,24.41,,, | | | | | | | | | | |
| 19489 | 2020-02-20 15:29:10 -03,19488,23.93,,, | | | | | | | | | | |
| 19490 | 2020-02-20 15:29:25 -03,19489,24.9,,, | | | | | | | | | | |
| 19491 | 2020-02-20 15:29:41 -03,19490,23.93,,, | | | | | | | | | | |
| 19492 | 2020-02-20 15:29:56 -03,19491,24.41,,, | | | | | | | | | | |
| 19493 | 2020-02-20 15:30:11 -03,19492,24.41,,, | | | | | | | | | | |
| 19494 | 2020-02-20 15:30:26 -03,19493,24.41,,, | | | | | | | | | | |

Fonte: Própria, 2020.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Esse trabalho apresentou todo o processo de desenvolvimento de um sistema de monitoramento de temperatura. No decorrer do período de testes, o sistema se comportou de maneira estável na coleta, armazenamento e tratamento dos dados. De fato, verificou-se que a integração de ferramentas *open source* é viável no contexto acadêmico devido ao baixo custo dos equipamentos envolvidos na aplicação e do uso de ferramentas gratuitas. O acesso remoto aos dados por *smartphone* conferiu mais mobilidade à aplicação, tornando mais simples o controle dos processos monitorados. Por fim, a notificação por eventos agrega valor ao projeto por automatizar o procedimento de verificação do comportamento que a aplicação apresenta. Ao final desse processo verificou-se que a integração das plataformas IoT ocorreu de maneira satisfatória permitindo que todos objetivos estabelecidos previamente fossem alcançados.

A infraestrutura proposta nessa aplicação é capaz de realizar o monitoramento de diversas grandezas físicas que utilizem sensores analógicos e digitais na sua aferição, sendo necessário para isso que alguns ajustes sejam feitos, principalmente no nó *function* do *NODE-RED*. O microcontrolador *Arduino UNO* utilizado nessa aplicação foi suficiente para o monitoramento, entretanto a utilização de microcontroladores mais modernos e compatíveis com a IDE do *Arduino*, como o ESP8266 e o ESP32, pode ser indispensável no desenvolvimento de aplicações mais complexas.

REFERÊNCIAS

- A REVOLUÇÃO TECNOLÓGICA 4.0. **Canaltech**. 2017. Disponível em: <https://canaltech.com.br/mercado/a-revolucao-tecnologica-40-92576/>. Acesso em: 01/01/2020.
- ALVES, R. M. S.; ARMANDO L. C., Pinto, M. C.; SAMPAIO, F. F. & Elia, M. F. **Uso do Hardware Livre Arduino em Ambientes de Ensino-aprendizagem**. Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 162- 187, 2013.
- ATZORI. Luigi. LERA. Antonio. MORABITO. Giacomo. **The Internet of Things: A survey**. 2010. Disponível em: <https://www.cs.mun.ca/courses/cs6910/IoT-Survey-Atzori-2010.pdf>. Acesso em: 15/02/2020.
- BARBOSA. Matheus. CONCEITO DE INFRAESTRUTURA TECNOLÓGICA DE TI. **Desenvolvedor Matheus**. Disponível em: <https://www.desenvolvedormatheus.com.br/conceito-de-infraestrutura-tecnologica/>. Acesso em 17/01/2020.
- BERTOLETI, Pedro. **Projetos com ESP32 e LoRa**. 1. ed. São Paulo: NCB, 2019.
- BORGIA, E. **The internet of things vision: Key features, applications and open issues**. Computer Communications, 2014.
- CONFIGURING THE MQTT PUBLISH NODE. **Steves internet guide**. 2019. Disponível em: <http://www.steves-internet-guide.com/configuring-the-mqtt-publish-node/>. Acesso em: 14/10/2019.
- EXEMPLO PRÁTICO PLATAFORMA THINGSPEAK. **Medium**. 2019. Disponível em: <https://medium.com/@tgeip5/exemplo-pr%C3%A1tico-plataforma-thingspeak-923e44ff71ca>. Acesso em: 07/01/2020.
- FIRMATA PROTOCOL DOCUMENTATION. **GitHub** .2019. Disponível em: <https://github.com/firmata/protocol>. Acesso em: 22/01/2020.
- GUIMARÃES. G. S. B. **Internet das Coisas: Desenvolvimento de um Medidor de Potência de Equipamentos Elétricos**. 2017. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2017. [Orientador: Prof. Dr. Guilherme Augusto Silva Pereira].
- BAUER. Harald, PATEL. Mark, VIERIRA. Jan. INTERNET DAS COISAS: OPORTUNIDADES E DESAFIOS. **McKinsey & Company**. 2015. Disponível em: <https://www.mckinsey.com/br/our-insights/internet-of-things-opportunities-and-challenges-for-semiconductor-companies>. Acesso em: 03/02/2020.
- INTERNET DAS COISAS: UM DESENHO DO FUTURO. **Proof**. 2017. Disponível em: <https://www.proof.com.br/blog/internet-das-coisas/>. Acesso em: 07/02/2020.
- JUNIOR. O. R. **Sistema de monitoramento residencial baseado em Internet das Coisas**. 2017. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Estadual de Londrina, Londrina, 2017. [Orientador: Prof. Dr. Francisco Granziera Junior].
- KONESKI. E. M. **Ambiente de comunicação segura para Internet das Coisas com a utilização do MQTT e TLS**. 2018. 72 f. Trabalho de Conclusão de Curso (Bacharelado em

Gestão da Informação) – Universidade Federal de Santa Catarina, Florianópolis, 2018. [Orientador: Prof. Dr. Carlos Becker Westphall].

McROBERTS. Michael. **Arduino Básico**. Tradução Rafael Zanolli. São Paulo. Novatec Editora, 2011.

MONITORAMENTO DE INTERNET DAS COISAS NA ERA DIGITAL. **OpServices**. 2017. Disponível em: <https://www.opservices.com.br/monitoramento-de-internet-das-coisas/>. Acesso em: 13/01/2020.

NODE-RED. **Aprendiendo Arduino**, ano. Disponível em: <https://aprendiendoarduino.wordpress.com/2018/11/20/node-red/>. Acesso em: 18/01/2020.

O QUE É COMPUTAÇÃO EM NUVEM. **Microsoft Azure**. 2020. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>. Acesso em: 22/11/2019.

OLIVEIRA, Cláudio Vieira. ZANETTI, Humberto Piovesana. **Arduino Descomplicado – Como Elaborar Projetos de Eletrônica**. São Paulo. Érica, 2015.

OLIVEIRA. Sérgio de. **Internet das coisas com ESP8366, Arduino e Raspberry PI**. 1. ed. São Paulo: Novatec, 2017.

POTENCIALIDADES E DIFICULDADES EM IOT: CONTEXTO BRASILEIRO, SETORES ESTRATÉGICOS E A TECNOLOGIA LORA. **CSEM**. 2017. Disponível em: <https://csembrasil.com.br/potencialidades-e-dificuldades-em-iot-contexto-brasileiro-setores-estrategicos-e-tecnologia-lora/>. Acesso em: 11/02/2020.

QUAL A IMPORTÂNCIA DO MONITORAMENTO NA ERA DA INTERNET DAS COISAS. CANAL **Westcon**. 2020. Disponível em: <https://blogbrasil.westcon.com/qual-a-importancia-do-monitoramento-na-era-da-internet-das-coisas>. Acesso em: 15/01/2020.

RELATÓRIO DO PLANO DE AÇÃO. **BNDES**. Disponível em: <https://www.bndes.gov.br/wps/wcm/connect/site/269bc780-8cdb-4b9b-a297-53955103d4c5/relatorio-final-plano-de-acao-produto-8-alterado.pdf?MOD=AJPERES&CVID=m0jDUok>. Acesso em: 05/03/2020.

REVOLUÇÃO. In: **DICIO, Dicionário Online de Português**. Porto: 7Graus, 2019. Disponível em: <https://www.dicio.com.br/revolucao/>. Acesso em: 05/03/2020.

RONCHETTI. L. S. **O valor da internet das coisas para as organizações estudo de casos múltiplos**. 2017. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Administração de Empresas) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. [Orientador: Prof. Dr. Antonio Carlos Gastaud Maçada].

SILVA. A. P. M.; RIBEIRO. L. E. **Sensoriamento remoto de potência para o sistema fotovoltaico conectado à rede do escritório verde da UTFPR**. 2017. 48 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Curitiba, 2017. [Orientador: Prof. Dr. Glauber Gomes de Oliveira Brante].

SOUZA. Fábio. **Controle dispositivos remotamente com o ESP8266**. Ebook. 2018. Disponível em: <https://docero.com.br/doc/5cnv05>. Acesso em: 10/01/2020.

UMA ARQUITETURA DE REFERÊNCIA PARA A INTERNET DAS COISAS - PARTE 1. **InfoQ**. 2016. Disponível em: <https://www.infoq.com/br/articles/internet-of-things-reference-architecture/>. Acesso em: 05/02/2020.

UPLOAD DATA TO THINGSPEAK THROUGH MQTT WITH NA ESP8266. **Arduinodiy**. 2017. Disponível em : <https://arduinodiy.wordpress.com/2017/04/05/upload-data-to-thingspeak-through-mqtt-with-an-esp8266/>. Acesso em: 13/01/2020.

REACT TO CHANNEL ACTIVITY WITH NOTIFICATION FROM IFTTT. **ThingSpeak Documentation**. Disponível em: <https://www.mathworks.com/help/thingspeak/use-ifttt-to-send-text-message-notification.html>. Acesso em: 07/02/2020.

WAHLBRINCK. A. J. **Monitoramento da rede elétrica por meio de Aplicação web e IoT**. 2018. 88 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) – Universidade do Vale do Taquari, Lajeado, 2018. [Orientador: Prof. Ms. Alexandre Stürmer Wolf].

APÊNDICE A - INSTALAÇÃO DO NODE-RED NO WINDOWS

Para utilizar o *NODE-RED* é necessário que o *Framework* NODE-JS esteja devidamente instalado no computador. O processo de instalação do *NODE-JS* é simples e rápido. As versões para *download* estão disponíveis no endereço <https://nodejs.org/en/>. É possível verificar se a instalação do *NODE-JS* ocorreu com sucesso, para isso deve-se abrir o *Prompt* de comando do *Windows* (executando como administrador) e proceder conforme os passos a seguir:

Digite:

- `node --version` (deverá aparecer a versão do *NODE-JS* instalado);
- `npm --version` (deverá aparecer a versão do gerenciador de pacotes `npm` instalado);

Ainda com o *Prompt* de comando do *Windows* sendo executado no modo administrador deve-se proceder conforme os passos a seguir para instalação do *NODE-RED*:

Digite:

- `npm install -g --unsafe-perm node-red`

Dessa forma o `npm` instalará automaticamente o *NODE-RED*.

Após essas etapas, o *NODE-RED* estará habilitado no computador. Para inicializá-lo digite o comando:

- `node-red`

Desse modo o editor *NODE-RED* já pode ser executado em qualquer navegador, desde que o *Prompt* de comando permaneça aberto.

Para acessar o *NODE-RED* abra o navegador e digite na barra de endereços: <http://localhost:1880>.

APÊNDICE B - CÓDIGO DO FLUXO NO NODE-RED

```
[{"id":"4baebb42.552894","type":"tab","label":"Flow
1","disabled":false,"info":""},{ "id":"fc532b57.5bdb28","type":"arduino
in","z":"4baebb42.552894","name":"Sensor de
Temperatura","pin":"0","state":"ANALOG","arduino":"831b97aa.1c8408","x":120,"y":200,"
wires":[["66445e12.5d4d4"]]}, {"id":"1fbe3ff3.059a7","type":"debug","z":"4baebb42.552894
","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","
x":590,"y":340,"wires":[]}, {"id":"a714d385.1adea","type":"mqtt
out","z":"4baebb42.552894","name":"ThingSpeak -
testeMQTT","topic":"channels/939308/publish/fields/field1/PUVVVPKZOTPND3H13","qos":
"0","retain":"false","broker":"bc7f39ec.e467f8","x":630,"y":200,"wires":[]}, {"id":"66445e12.
5d4d4","type":"function","z":"4baebb42.552894","name":"Converte para
temperatura","func":"msg.payload = parseFloat(msg.payload)*0.48828125;\nmsg.payload =
parseFloat(msg.payload.toFixed(2))\nmsg.payload.toFixed(2).replace(\".\",\"\");\nreturn
msg;","outputs":1,"noerr":0,"x":360,"y":200,"wires":[["a714d385.1adea","1fbe3ff3.059a7","b
98cd088.089c","c83e80bc.3fa42"]]}, {"id":"b98cd088.089c","type":"ui_chart","z":"4baebb42.
552894","name":"Temperatura X
Tempo","group":"2cf8bad4.5bc086","order":0,"width":0,"height":0,"label":"chart","chart
Type":"line","legend":"true","xformat":"HH:mm:ss","interpolate":"linear","nodata":"grafico",
"dot":false,"ymin":25,"ymax":33,"removeOlder":18,"removeOlderPoints":"","removeO
lderUnit":3600,"cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#
2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs
":1,"x":620,"y":260,"wires":[]]}, {"id":"c83e80bc.3fa42","type":"ui_gauge","z":"4baebb42.5
52894","name":"Temperatura em
Palmas","group":"2cf8bad4.5bc086","order":1,"width":0,"height":0,"gtype":"gage","title":"ga
uge","label":"°C","format":"{{ value }}","min":0,"max":50,"colors":["#80fff","#00ff00","#f
f8000"],"seg1":25,"seg2":30,"x":630,"y":300,"wires":[]}, {"id":"831b97aa.1c8408","type"
:"arduino-board","z":"","device":"COM3"}, {"id":"bc7f39ec.e467f8","type":"mqtt-
broker","z":"","name":"mqtt.thingspeak.com:1883","broker":"mqtt.thingspeak.com","port":1
883,"clientid":"pedronoieto","usetls":false,"compatmode":false,"keepalive":60,"cleansessi
on":true,"birthTopic":"","birthQos":0,"birthPayload":"","closeTopic":"","closeQos":0,"cl
osePayload":"","willTopic":"","willQos":0,"willPayload":""}, {"id":"2cf8bad4.5bc086","typ
```

```
e":"ui_group","z":"","name":"Entradas  
analógicas","tab":"6e5cb30c.55180c","disp":false,"width":"6","collapse":false},{  
"id":"6e5cb30c.55180c","type":"ui_tab","z":"","name":"Home","icon":"dashboard",  
"disabled":false,"hidden":false}]
```