

**UNIVERSIDADE DE TAUBATÉ**

**Vinicius Leite Lemes**

**Giovanni Anjoletto da Cunha**

**ESTAÇÃO METEOROLÓGICA INTELIGENTE PARA  
LANÇAMENTOS DE VLS**

**Taubaté**

**2023**

**Vinicius Leite Lemes**  
**Giovanni Anjoletto da Cunha**

**ESTAÇÃO METEOROLÓGICA INTELIGENTE PARA  
LANÇAMENTOS DE VLS**

Trabalho de Graduação apresentado para obtenção do Certificado de Graduação do curso de Engenharia Elétrica e Eletrônica do Departamento de Engenharia Mecânica da Universidade de Taubaté.

Orientador (a): Prof. Dr. Prof. Antônio Faria Netto

**Taubaté**  
**2023**

**Grupo Especial de Tratamento da Informação - GETI**  
**Sistema Integrado de Bibliotecas – SIBi**  
**Universidade de Taubaté - Unitau**

L552e

Lemes, Vinicius Leite

Estação meteorológica inteligente para lançamentos de VLS / Vinicius Leite Lemes, Giovanni Anjoletto da Cunha. -- 2023.  
56 f. : il.

Monografia (graduação) – Universidade de Taubaté, Departamento de Engenharia Mecânica e Elétrica, 2023.

Orientação: Prof. Dr. Antonio Faria Neto, Departamento de Instituto Básico de Ciências Exatas.

1. Acesso Remoto. 2. Dados. 3. Informação. 4. Internet das Coisas. I. Universidade de Taubaté. Departamento de Engenharia Mecânica e Elétrica. Graduação em Engenharia Elétrica e Eletrônica. II. Cunha, Giovanni Anjoletto da.

CDD – 621.3

**Vinicius Leite Lemes**  
**Giovanni Anjoleta da Cunha**

**ESTAÇÃO METEOROLÓGICA INTELIGENTE PARA  
LANÇAMENTOS DE VLS**

Trabalho de Graduação apresentado para obtenção do Certificado de Graduação do curso de Engenharia Elétrica e Eletrônica do Departamento de Engenharia Mecânica da Universidade de Taubaté.

**DATA:** \_\_\_\_\_

**RESULTADO:** \_\_\_\_\_

**BANCA EXAMINADORA:**

Prof. Dr. Prof. Antônio Faria Netto

UNIVERSIDADE DE TAUBATÉ

Assinatura: \_\_\_\_\_

Prof. Me Sandro Botossi dos Santos

UNIVERSIDADE DE TAUBATÉ

Assinatura: \_\_\_\_\_

Prof. Dr. Francisco Antônio Lotufo

UNIVERSIDADE ESTADUAL PAULISTA

Assinatura: \_\_\_\_\_

(16 de novembro de 2023)

Dedicamos este Trabalho a nossa família e amigos e, agradecemos os mesmos pela compreensão e apoio durante está longa e dura jornada, aonde eles foram de suma importância para a conclusão deste trabalho.

Vinicius e Giovanni

## **AGRADECIMENTOS**

Em primeiro lugar agradeço a Deus, fonte da vida e da graça. Agradeço pela minha vida, minha inteligência, meus amigos.

À Universidade de Taubaté – UNITAU, que ofereceu um excelente ambiente educacional com profissionais qualificados

Ao nosso orientador, *Prof. Dr. Antônio Faria Netto* por todo o incentivo e motivação na orientação deste trabalho.

Aos nossos familiares, que apesar das dificuldades enfrentadas, sempre incentivaram e motivaram.

Aos Professores Msc Sandro Botossi dos Santos e Dr Francisco Antônio Lotufo por aceitarem compor a banca examinadora.

Às funcionárias da Secretaria, Limpeza e da Segurança pela dedicação, presteza e principalmente pela vontade de ajudar.

## RESUMO

A estação meteorológica inteligente é baseada no conceito de Internet das Coisas que é o modo como os equipamentos estão conectados e se comunicam entre si e com usuários. Na atualidade, este conceito abre caminho para a utilização de sensores inteligentes e aplicativos que trabalham de forma eficiente em rede. Os benefícios desta tecnologia incluem a possibilidade de troca de informações entre múltiplos usuários além do uso benéfico dos dados, permitindo a previsão de falhas e a extração de características do sistema através do uso de inteligência artificial. Este projeto visa integrar uma instrumentação de meteorologia com uma plataforma de Internet das coisas. As principais metas deste trabalho são a prova de conceito deste tipo de tecnologia, a possibilidade de acesso remoto a informações e também o uso de sensores e sistemas de processamento, para garantir a eventual implementação em larga escala. Ao longo deste trabalho foi montado com sucesso a estação meteorológica capaz de adquirir dados de umidade, temperatura, pressão atmosférica, velocidade do vento, quantidade de precipitação e probabilidade de chuva. Esta estação pode ser acessada através de uma conexão Wi-Fi e os dados coletados são enviados diretamente a um banco de dados em uma nuvem. Através do banco de dados os usuários podem ter o acesso remoto aos dados climáticos para consulta em tempo real ou extração de parâmetros de longo prazo.

**Palavras Chaves:** Acesso Remoto. Dados. Informação. Internet das Coisas.

## **ABSTRACT**

The smart weather station is based on the concept of the Internet of Things, which is the way in which equipment is connected and communicates with each other and with users. Currently, this concept paves the way for the use of intelligent sensors and applications that work efficiently in a network. The benefits of this technology include the possibility of exchanging information between multiple users in addition to the beneficial use of data, allowing the prediction of failures and the exclusion of system characteristics through the use of artificial intelligence. This project aims to integrate meteorology instrumentation with an Internet of Things platform. The main goals of this work are the proof of concept of this type of technology, the possibility of remote access to information and also the use of sensors and processing systems, to guarantee an eventual large-scale implementation. Throughout this work, a meteorological station capable of acquiring data on humidity, temperature, atmospheric conditions, wind speed, number of occurrences and pressure probability was successfully set up. This station can be accessed via a Wi-Fi connection and the data found is sent directly to a database in a cloud. Through the database, users can have remote access to climate data for real-time consultation or long-term parameter protection.

Keywords: Remote Access. Data. Information. Internet of Things.

## LISTA DE FIGURAS

Figura 1 - Montagem Protoboard .....	16
Figura 2 - Montagem do protótipo desenvolvido da estação meteorológica .....	17
Figura 3 - Central que armazena a comunicação dos sensores com o ESP8266 ....	17
Figura 4 - Base Básica de Meteorologia II .....	18
Figura 5 - Sensor de umidade e temperatura modelo DHT11 .....	19
Figura 6 - Diagrama de conexão entre o sensor DHT11 e a plataforma NodeMCU .	19
Figura 7 - Anemômetro digital .....	20
Figura 8 - Diagrama de conexão entre o Sensor Anemômetro e o NodeMCU .....	20
Figura 9 - Sensor Pluviométrico .....	21
Figura 10 - Diagrama de conexão entre o Sensor Pluviométrico e o NodeMCU.....	21
Figura 11 - Sensor de Pressão Barométrica .....	22
Figura 12 - Diagrama de conexão entre o Sensor de Pressão Barométrica e o NodeMCU.....	22
Figura 13 – Sensor de Chuva.....	23
Figura 14 – Diagrama de conexão entre Sensor de Chuva e o NodeMCU .....	23
Figura 15 – Bateria Li-Íon 18650 .....	23
Figura 16 – Indicador de tensão.....	24
Figura 17 – Painel Solar.....	24
Figura 18 – Carregador de Bateria TP4056 .....	25
Figura 19 – Módulo MT3608 .....	26
Figura 20 – Plataforma de Internet das Coisas NodeMCU .....	27
Figura 21 – Interface de programação IDE-Arduino utilizada no desenvolvimento do projeto (Versão 1.8.39.0).....	27
Figura 22 – Interface de desenvolvimento da página Web utilizando o Visual Studio Code (Versão 1.83.1) .....	28
Figura 23 – Banco de dados do projeto da estação meteorológica utilizando o Firebase .....	29
Figura 24 – Anemômetro.....	30
Figura 25 – Fixação do anemômetro com parafusos direto na madeira .....	30
Figura 26 – Painel solar fotovoltaico .....	31
Figura 27 – Base para fixação dos painéis solares fotovoltaicos .....	31
Figura 28 – Pluviômetro .....	32

Figura 29 – Fixação do Pluviômetro.....	32
Figura 30 – Circuito de controle .....	33
Figura 31 – Circuito de alimentação.....	34
Figura 32 – Custo para construção da Estação Meteorológica.....	35
Figura 33 – Versão final da estação meteorológica .....	35
Figura 34 – Diagrama de contexto do projeto mostrando a estação meteorológica baseada em Internet das Coisas.....	36
Figura 35 – Dados de temperatura e Umidade relativa no dia 06/07/2023 .....	37
Figura 36 – Classificação do funcionamento da estação meteorológica durante o período de testes de longo prazo, entre os dias 17/05 e 07/07.....	38
Figura 37 – Gráfico de Chuva Acumulada Mensal .....	39
Figura 38 – Umidade por Hora - INMET .....	39
Figura 39 – Umidade Relativa por hora BBMET .....	40
Figura 40 – Temperatura por Hora - INMET .....	40
Figura 41 – Temperatura por Hora BBMET.....	41
Figura 42 – Layout da página Web que mostra os dados da estação meteorológica em tempo real .....	41

## LISTA DE TABELAS

Tabela 1 - Especificação elétrica da plataforma meteorológica .....	15
Tabela 2 - Especificação elétrica do carregador de baterias TP4056 .....	25
Tabela 3 - Especificação elétrica do Módulo MT3608 .....	26

## LISTA DE ABREVIATURAS E SIGLAS

INMET	Instituto Nacional de Meteorologia
Wi-Fi	Wireless Fidelity
IAE	Instituto de Aeronáutica e Espaço
CTA	Centro Tecnológico da Aeronáutica
BM	Brisa Marítima
ZCIT	Zona de Convergência Intertropical
DOL	Distúrbio Ondulatório de Leste
VCAN	Vórtice Ciclônico Em Altos Níveis
LI	Linhas de Instabilidade
SF	Sistemas Frontais
CLS	Camada Limite Superficial
VLS	Veículo Lançador de Satélite
GB	Gigabytes
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
IoT	Internet of Things
BB MET II	Base Básica de Meteorologia II
CC	Corrente Contínua
CA	Corrente Alternada
V	Volt
C	Celsius
USB	Universal Serial Bus
Km/h	Quilômetro por Hora
hPa	Hectopascal
I2C	Inter-Integrated Circuit
GPIO	General Purpose Input/Output
PWM	Pulse Width Modulation
mAh	miliampere-hora
IDE	Integrated Development Environment
cm	Centímetro
m	Metro
km	Quilômetro

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
<b>2 REVISÃO BIBLIOGRÁFICA.....</b>	<b>13</b>
<b>3 MATERIAIS E METODOS .....</b>	<b>15</b>
3.1 SENSORES UTILIZADOS .....	18
3.1.1 Sensor de umidade e temperatura .....	19
3.1.2 Sensor Anemômetro.....	19
3.1.3 Sensor Pluviométrico.....	20
3.1.4 Sensor de Pressão Barométrica.....	21
3.1.5 Sensor de Chuva.....	22
3.1.6 Bateria Li-Íon 18650 .....	23
3.1.7 Painéis Solares .....	24
3.1.8 Carregador de Bateria TP4056 .....	24
3.1.9 Módulo MT3608 .....	25
3.2 PLATAFORMA DE INTERNET DAS COISAS E PROCESSAMENTO DOS DADOS .....	26
3.3 ACESSO WEB E BANCO DE DADOS.....	28
<b>4 DESENVOLVIMENTO .....</b>	<b>30</b>
<b>5 RESULTADOS.....</b>	<b>36</b>
<b>6 CONCLUSÃO .....</b>	<b>43</b>
<b>7 REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>44</b>
<b>APENDICE 1 .....</b>	<b>45</b>
<b>APENDICE 2.....</b>	<b>47</b>
<b>APENDICE 3.....</b>	<b>49</b>
<b>APENDICE 4.....</b>	<b>51</b>
<b>APENDICE 5.....</b>	<b>54</b>

## 1 INTRODUÇÃO

Em 1971 foi criada uma organização no Ministério da Aeronáutica, o Instituto de Aeronáutica e Espaço (IAE), no então Centro Tecnológico da Aeronáutica (CTA), com a missão de gerir, desenvolver e aprimorar as atividades espaciais.

Para o sucesso de qualquer lançamento ou operação aeroespacial, é de vital importância o conhecimento das condições meteorológicas da região. Essas condições influenciam ou modificam a Trajetória dos veículos em análise.

Existem alguns fatores climáticos que podem ser determinantes na região Tropical como: a brisa marítima (BM); a zona de convergência intertropical (ZCIT); o distúrbio ondulatório de Leste (DOL); o vórtice ciclônico em altos níveis (VCAN); as linhas de instabilidade (LI) e os sistemas frontais (SF).

Temperatura e Umidade podem influenciar no funcionamento de alguns componentes de veículos aeroespaciais, já que a composição química pode sofrer alterações como: dilatações, expansão ou alguma anomalia que modifique a sua função específica, assim como ocorreu no lançamento do ônibus espacial Challenger, culminando no derradeiro acidente. Também influenciam na dinâmica dos ventos.

A circulação de brisa é induzida e dirigida localmente pelo gradiente de temperatura (mecanismo térmico) que se estabelece perpendicular à linha da costa. As intensidades das brisas marítimas e terrestres estão diretamente relacionadas às variações diurnas e às variações sazonais. A brisa tende a dominar a circulação local quando os ventos globais são mais fracos, tal como é o caso dos ventos alísios

O conhecimento da variação temporal da velocidade e direção do vento é de grande interesse para as atividades operacionais nos lançamentos de foguetes já que ela implica na trajetória dos veículos. Por isso é necessário o controle criterioso antes do lançamento de qualquer aeronave.

## 2 REVISÃO BIBLIOGRÁFICA

O programa espacial brasileiro, por mais que exista competências individuais para criar o mesmo nível que de outras nações, falta a visão para fomentar as benesses que são atribuídas para com o avanço do programa, benefícios como: independência econômica, política e além de elevar o nível de educação da sociedade brasileira (PALMERIO, 2017).

Com o intuito de fomentar e ajudar no avanço do programa nacional brasileiro, o projeto tem como intuito trazer um melhor custo benefício para a análise de dados cruciais e evitar a dependência de tecnologia estrangeira, que muitas vezes são adquiridas com preço exorbitantes. Assim sendo, a estação meteorológica visa auxiliar na coleta de dados meteorológicos que são muito variáveis no Brasil.

O interesse científico em estudar a CLS (Camada Limite Superficial) está relacionado às significativas transferências de momentum, calor sensível, calor latente, gases e aerossóis, entre a superfície da terra e a atmosfera, os quais são vitais à manutenção da vida na Terra e também para o entendimento dos processos meteorológicos e climatológicos. E que interferem significativamente na manutenção climática. O lançamento de um satélite brasileiro, por um veículo nacional, a partir de um centro espacial no Brasil compõe a campanha denominada Missão Espacial Completa Brasileira (MECB) (FERREIRA GISLER, 2009).

A melhor maneira de viabilizar a coleta desses dados para auxiliar no lançamento do VLS é com o desenvolvimento de uma estação meteorológica que possa concatenar todos os sensores necessários, de modo que, esses dados após coletados tenham um fácil acesso por parte do usuário, e que a gama de dados seja a maior possível.

Um banco de dados em tempo real é um sistema de banco de dados que usa processamento em tempo real para lidar com cargas de trabalho cujo estado muda constantemente. Isso difere dos bancos de dados tradicionais que contêm dados persistentes, em sua maioria não afetados pelo tempo (GHARSELLAOUI, 2019).

O banco de dados utilizado para o projeto, foi o firebase, que é um banco de dados gratuito com uma limitação de espaço de 10GB para hosting, o que é suficiente para a aplicabilidade da estação meteorológica. A utilização do firebase

viabiliza utilizar páginas de internet feitas com base na linguagem HTML, como no caso da página que foi criada para que ocorra a visualização dos dados coletados em tempo real, além dessa linguagem foi utilizado CSS para melhorar o layout da página e JavaScript com o intuito de fazer comunicação dos dados que estão no firebase com o banco de dados.

IoT tem tudo a ver com conectar dispositivos, ou “coisas”, como você pode chamá-los, à Internet e, em seguida, analisar os dados coletados desses dispositivos para extrair um valor agregado. (GHARSELLAOUI,2019).

(VIEBRANTZ, 2017) realizou um trabalho que serviu como base para a construção da estação, pois em seu trabalho ele utiliza como base a ideia de IoT, e concatena alguns sensores com o firebase, e depois demonstra seus resultados em uma aplicação HTML.

### 3 MATERIAIS E MÉTODOS

Para o desenvolvimento da estação meteorológica foram utilizados os sensores de umidade relativa do ar e temperatura (DHT11), o sensor Anemômetro, sensor Pluviométrico, sensor de Pressão Barométrica, sensor de chuva e a placa de desenvolvimento NodeMCU que opera com o chip ESP8266. No NodeMCU estará contida as linhas de código que permitem a integração dos sensores e a comunicação remota. O NodeMCU tem como característica principal as possibilidades de conexão via Wi-Fi que faz com que seja a plataforma de conexão de Internet das Coisas do projeto. A tabela 1 lista as especificações técnicas da plataforma desenvolvida.

Tabela 1 – Especificação elétrica da plataforma meteorológica.

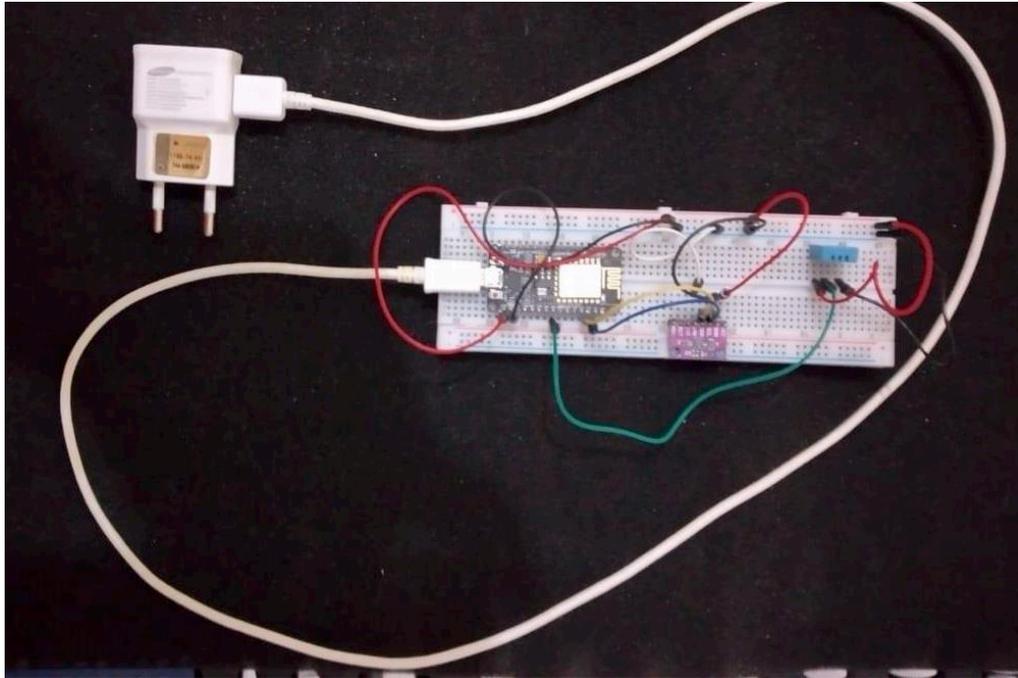
<b>Característica</b>	<b>Valor</b>	<b>Unidade</b>
Tensão de alimentação	5	volt [V]
Corrente	500	mili ampere [mA]
Escala de medição de temperatura	0° a 50°	celsius [°C]
Escala de medição de umidade	20% a 95%	porcentagem [%]
Escala de medição da velocidade do vento	0km/h a 200km/h	quilometro por hora [km/h]
Escala de medição de pluviosidade	0mm a 145mm	milímetro [mm]
Escala de medição de pressão barométrica	30000Pa a 110000Pa	pascoal [Pa]
Escala de medição do sensor de chuva	0% a 100%	porcentagem [%]
Tipo de conexão	Wi-Fi	802.11 b/g/n
Taxa de transmissão	2.4	giga hertz [Ghz]

Fonte: Próprio autor.

Na Figura 1 é possível visualizar a montagem que integra os sensores mencionados acima com a plataforma NodeMCU. Essa é a 1º versão do protótipo.

Este conjunto atende as especificações listadas na Tabela 1. Na primeira versão, de desenvolvimento do projeto, a montagem foi realizada em um protoboard para alocar todos os componentes da montagem da estação e ainda assim possibilitar alterações naturais no período de desenvolvimento. A alimentação do sistema foi realizada por meio de um carregador de celular USB.

Figura 1 – Montagem Protoboard



Fonte: Próprio autor.

A segunda versão do protótipo foi feita em um esqueleto de madeira apresentado na figura 2 e figura 3, onde foi alocado os sensores:

- Anemômetro;
- Pluviômetro;
- DHT11;
- NodeMCU ESP8266;
- BMP180;
- Sensor de Chuva;
- Circuito de Alimentação;
- Placas Fotovoltaicas.

Figura 2 – Montagem do protótipo desenvolvido da estação meteorológica



Fonte: Próprio autor.

Figura 3 – Central que armazena a comunicação dos sensores com o ESP8266



Fonte: Próprio autor.

A ideia em um primeiro instante era validar a coleta de dados pelos sensores, e então, a estética foi deixada de lado em um primeiro momento, após a coleta e validação dos dados foi pensado uma forma de melhor alocar os componentes que fazem parte do projeto.

Conforme foi sendo aperfeiçoado, a sua estrutura foi sendo moldada para uma melhor adaptabilidade em campo, conferindo a ela, praticidade de instalação, tamanho diminuto e organização, pois todos os sensores foram agrupados em um só

estamento, sendo assim, foi concebida a sua Terceira versão apresentada na figura 4, que recebeu a denominação de BB MET II (Base Básica de Meteorologia II).

Figura 4 – Base Básica de Meteorologia II



Fonte: Próprio autor.

Junto da 3ª versão do protótipo, foi implementado uma fonte de energia sustentável para a alimentação do sistema, basicamente, durante o dia o fornecimento de energia é feito por placas fotovoltaicas que deixam um banco de baterias em flutuação e, no período da noite ocorre a troca da alimentação, entrando em uso a carga das baterias. Faz se necessário lembrar que, é possível alimentar o circuito através de uma fonte CA/CC, que vem junto do protótipo, sendo assim uma terceira via de alimentação para o projeto.

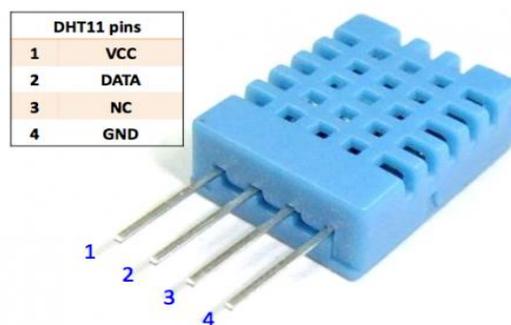
### 3.1 SENSORES UTILIZADOS

Esta subseção descreve os detalhes dos sensores utilizados no desenvolvimento da plataforma meteorológica.

### 3.1.1 Sensor de umidade e temperatura

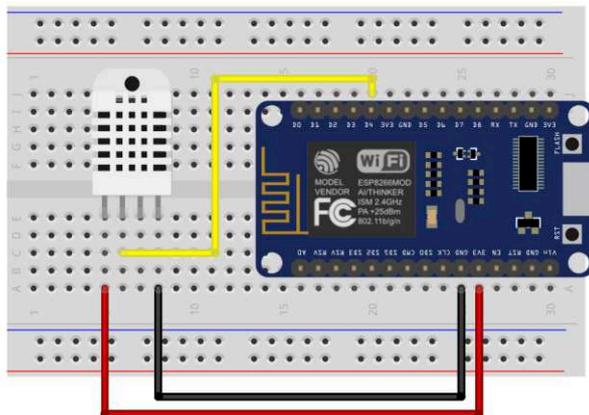
O sensor de umidade e temperatura modelo DHT11 pode ser visualizado na Figura 5 é capaz de medir a temperatura e umidade relativa do ar. Este sensor é alimentado com 3.3V. Conforme listado na Tabela 1, sua faixa de temperatura é de 0° a 50° e sua precisão é de 2°C. A umidade é medida na faixa entre 20% a 95% com precisão de 5%. Na Figura 6 é possível também visualizar a conexão deste sensor com o NodeMCU.

Figura 5 – Sensor de umidade e temperatura modelo DHT11



Fonte: Manual de Instruções.

Figura 6 – Diagrama de conexão entre o sensor DHT11 e a plataforma NodeMCU



Fonte: Próprio autor.

### 3.1.2 Sensor Anemômetro

O Sensor Anemômetro ilustrado na figura 7 é composto por um Reed Switch que funciona como uma chave, a cada volta completa do ímã alocado na base circular das pás possibilita o chaveamento gerando assim um pulso. O circuito é

alimentado com 3.3V, a capacidade de leitura da velocidade do vento vai de 0 km/h até aproximadamente 200 km/h.

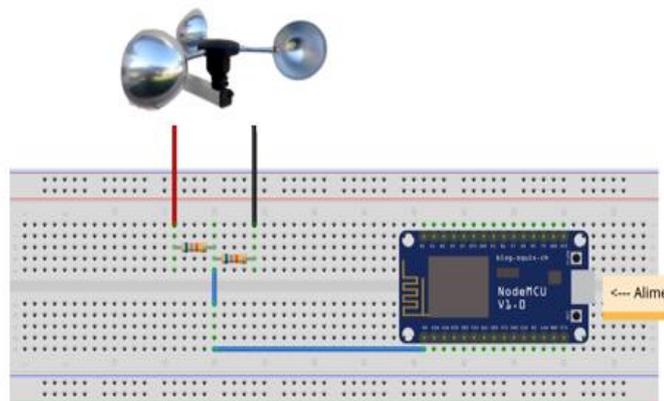
Na figura 8, é possível visualizar um dos esquemas típicos de utilização do sensor, com a correta discriminação de sua pinagem, para a correta ligação do anemômetro com o NodeMCU.

Figura 7 – Anemômetro digital



Fonte: Manual de Instruções.

Figura 8 – Diagrama de conexão entre o Sensor Anemômetro e o NodeMCU



Fonte: Próprio autor.

### 3.1.3 Sensor Pluviométrico

O Sensor Pluviométrico mostrado na figura 9 é composto por um Reed Switch que funciona como uma chave, dentro do sensor existe um compartimento piramidal comumente chamado de balsa, que se enche de água até certo nível e faz com que ele oscile para esquerda ou direita, assim fazendo com que o ímã varie sua posição e acione o Reed Switch.

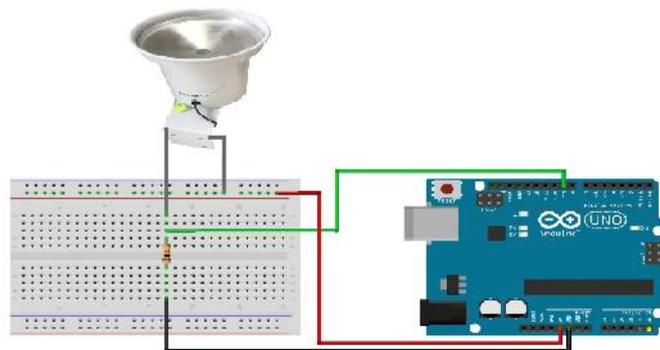
Na figura 10, é possível visualizar o esquema de conexão do sensor com o NodeMCU.

Figura 9 – Sensor Pluviométrico



Fonte: Manual de Instruções.

Figura 10 – Diagrama de conexão entre o Sensor Pluviométrico e o NodeMCU

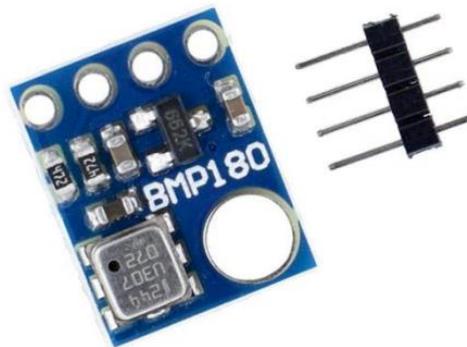


Fonte: Próprio autor.

### 3.1.4 Sensor de Pressão Barométrica

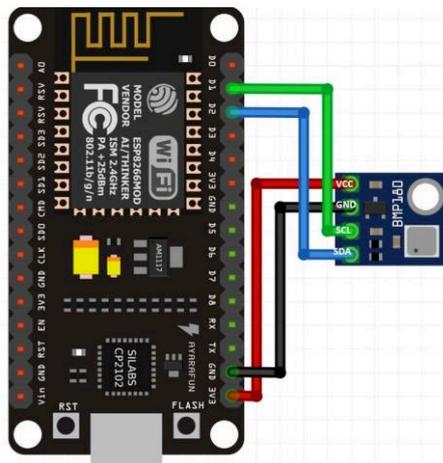
O BMP180 apresentado na figura 11, é um sensor digital utilizado para detectar a pressão atmosférica e temperatura, possui uma faixa de medição de 300 a 1100hPa, proporciona uma precisão absoluta de 2,5hPa e ruído de até 0,03hPa, além de fornecer uma medição de temperatura na faixa entre -45°C e 85°C, com precisão de +-2°C. Para comunicação utiliza a interface I2C Inter-Integrated Circuit, na figura 12 é possível visualizar sua ligação com o NodeMCU.

Figura 11 – Sensor de Pressão Barométrica



Fonte: Manual de Instruções.

Figura 12 – Diagrama de conexão entre o Sensor de Pressão Barométrica e o NodeMCU



Fonte: Próprio autor.

### 3.1.5 Sensor de Chuva

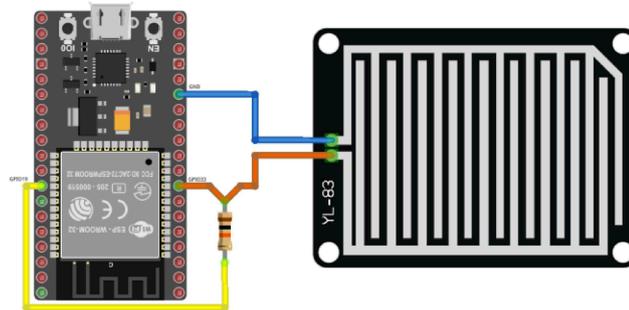
O Sensor de Chuva apresentado na figura 13, é um módulo eletrônico desenvolvido com a finalidade de detectar gotas de chuva em uma placa que faz parte do mesmo. Caso não seja detectado gotas de água na superfície da placa, a saída digital do sensor se mantém em nível alto e quando o sensor detectar alguma gota de água sobre a superfície, a saída digital altera para nível baixo. Para medir a variação de água sobre a superfície, faz-se o uso do pino analógico disponível no sensor em conjunto com um microcontrolador que possua conversor analógico digital. Na figura 14 é representada a conexão entre o sensor e o NodeMCU.

Figura 13 – Sensor de Chuva



Fonte: Manual de Instruções.

Figura 14 – Diagrama de conexão entre Sensor de Chuva e o NodeMCU



Fonte: Próprio autor.

### 3.1.6 Bateria Li-Íon 18650

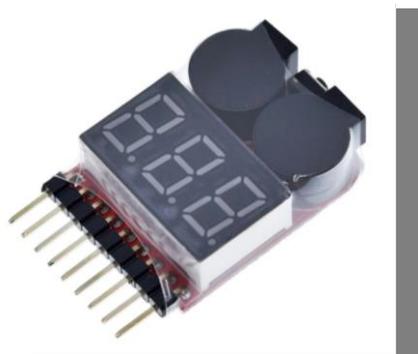
Bateria recarregável mostrada na figura 15, é composta por células de íons de Lítio, sua principal característica é ter longa vida útil. É utilizada para alimentar os componentes elétricos durante os períodos de baixa incidência solar, integrado às bateria foi instalado um módulo MDL0029 mostrado na figura 16, possui um visor que indica a tensão nas células da bateria e emite um aviso sonoro quando a tensão estiver abaixo do recomendado.

Figura 15 – Bateria Li-Íon 18650



Fonte: Manual de Instruções.

Figura 16 – Indicador de tensão



Fonte: Manual de Instruções.

### 3.1.7 Painéis Solares

Também conhecido como módulo fotovoltaico ou célula solar, o painel solar apresentado na figura 17 é um dispositivo que converte a energia da luz do sol em eletricidade utilizando o efeito fotovoltaico. Cada painel solar é composto por várias células solares interconectadas, que são feitas de materiais semicondutores, como o silício. Fornece uma tensão de saída de 1,25V cada, foram utilizadas 3 placas, para alimentar os componentes da Estação Meteorológica possibilitando a recarga das baterias.

Figura 17 – Painel Solar



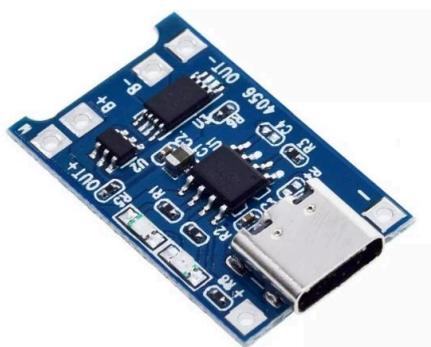
Fonte: Manual de Instruções.

### 3.1.8 Carregador de Bateria TP4056

O Carregador de Bateria TP4056 mostrada na figura 15, é utilizado para carregar as baterias com a tensão fornecida pelos painéis fotovoltaicos. O TP4056 foi desenvolvido para solucionar problemas com falta de carga em baterias, ele possibilita que baterias de lítio (Li-Ion), 18650 por exemplo, sejam recarregadas

diretamente no compartimento onde se encontram, sem necessitar removê-las. A tabela 2 lista as especificações técnicas do TP4056, o diferencial desse modelo é a presença de saídas específicas para uma bateria Li-Ion com tensão entre 3.7 e 4.2V e pinos próprios para a carga, possibilitando que ele gerencie a energia de carregamento e permita a utilização de imediato da energia em pinos próprios, sendo totalmente compatível com baterias 18650 recarregáveis. O TP4056 conta com leds indicadores de carga, sendo um led vermelho que indica que está em processo de carregamento e um led verde indica que a carga está completa.

Figura 18 – Carregador de Bateria TP4056



Fonte: Manual de Instruções.

Tabela 2 – Especificação elétrica do carregador de baterias TP4056.

<b>Tensão de Entrada</b>	5V
<b>Tensão de Carregamento</b>	4,2V (+ ou -) 1%
<b>Corrente Máxima de Carregamento</b>	1000mA
<b>Tensão de Proteção contra Descarga Excessiva</b>	2,5V
<b>Corrente de proteção contra sobrecorrente de baterias</b>	3 <sup>a</sup>
<b>Tamanho</b>	28 * 1,73mm

Fonte: Próprio autor.

### 3.1.9 Módulo MT3608

O Módulo Step-Up CC/CC apresentado na figura 19 é um componente que consegue converter uma tensão baixa de entrada em uma tensão alta na saída, que, por sua vez, pode ser regulada por meio de um trimpot. Uma das principais

diferenças desse modelo para os outros é o ajuste que pode ser feito na tensão de saída por meio do trimpot, sua especificação elétrica é mostrada na tabela 3.

Figura 19 – Módulo MT3608



Fonte: Manual de Instruções.

Tabela 3 – Especificação elétrica do Módulo MT3608

<b>Tensão de Entrada</b>	2V à 24V
<b>Tensão de saída ajustável</b>	2,5 à 28V
<b>Regulação de carga</b>	+/- 0,5%
<b>Eficiência</b>	91%
<b>Diferença mínima entre a entrada e saída</b>	0,5V
<b>Corrente máxima</b>	2A

Fonte: Próprio autor.

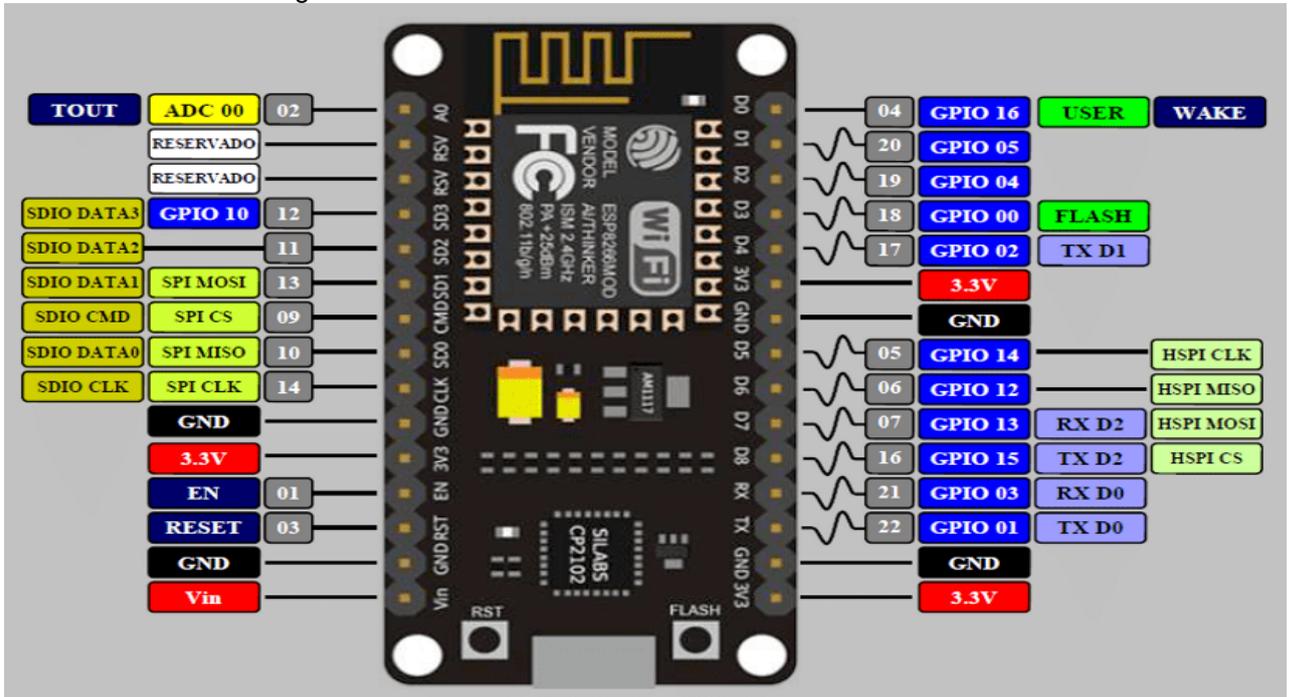
### 3.2 PLATAFORMA DE INTERNET DAS COISAS E PROCESSAMENTO DOS DADOS

A Figura 20 mostra em detalhes a plataforma NodeMCU com o chip ESP8266. Este dispositivo é também alimentado com tensão nominal de 3,3V. Esta plataforma possui 17 interfaces GPIO que podem ser configuradas como entrada ou saída digitais, existem também 4 interfaces de saída PWM e uma entrada para leitura de dados analógicos. O consumo de corrente desta plataforma é de 170mA. Há ainda a possibilidade de operação com bateria, neste caso, considerando uma bateria de 3,7V com 1000mAh, o NodeMCU tem autonomia de 20 horas desconsiderando o consumo dos dispositivos periféricos de I/O (sensores).

O NodeMCU permite a conexão Wi-Fi que fica com 20% da sua capacidade de processamento. Para a lógica de programação foi utilizada a IDE (Integrated

Development Environment) do Arduino que pode ser visualizada na Figura 21 e que utiliza a linguagem C/C++ para programação.

Figura 20 – Plataforma de Internet das Coisas NodeMCU



Fonte: Próprio autor.

Figura 21 – Interface de programação IDE-Arduino utilizada no desenvolvimento do projeto (Versão 1.8.39.0)

A captura de tela mostra a interface de programação IDE-Arduino com o seguinte código de exemplo:

```

#include <ArduinoJson.h>
#include <FirebaseArduino.h>
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <Adafruit_CCS811.h>

#include <NTPClient.h>
#include <WiFiUdp.h>

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

//----Definições do sensor de CO2----//
Adafruit_CCS811 ccs;

//----Definições do FIREBASE----//
#define FIREBASE_HOST "projeto-iae-3.firebaseio.com"
#define FIREBASE_AUTH "Lkuf3UyutY8kUqZNTu8UsA9P6biIYYAPvngqHWX"

//----Definições da REDE----//
#define WIFI_SSID "Familia Leite 2.4G"
#define WIFI_PASSWORD "60646064"

//----Definições da PINAGEM DHT11----//
#define DHT_PIN D5
  
```

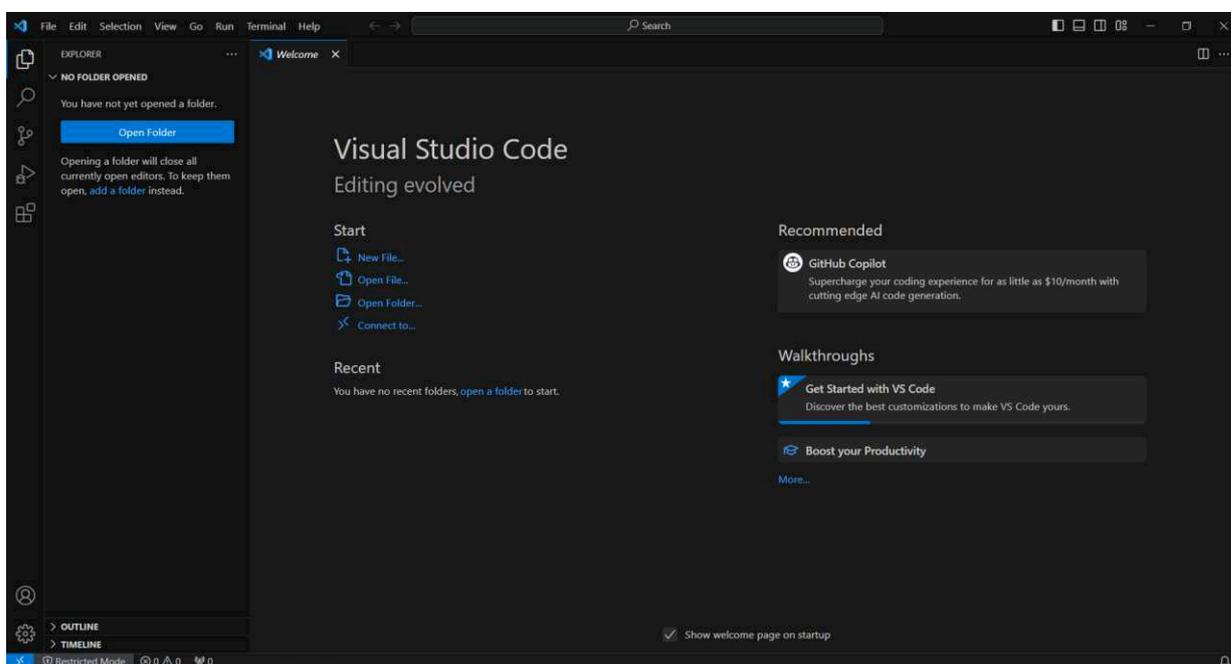
Fonte: Próprio autor.

### 3.3 ACESSO WEB E BANCO DE DADOS

A página Web criada para acessar os dados da plataforma foi montada utilizando o Visual Studio, que é mostrado na Figura 22. A linguagem de programação adotada foi o Java Script que comunica o Banco de Dados com a interface Web. O layout da página foi feito usando as linguagens HTML e CSS que são duas das principais tecnologias para a criação de páginas da Web.

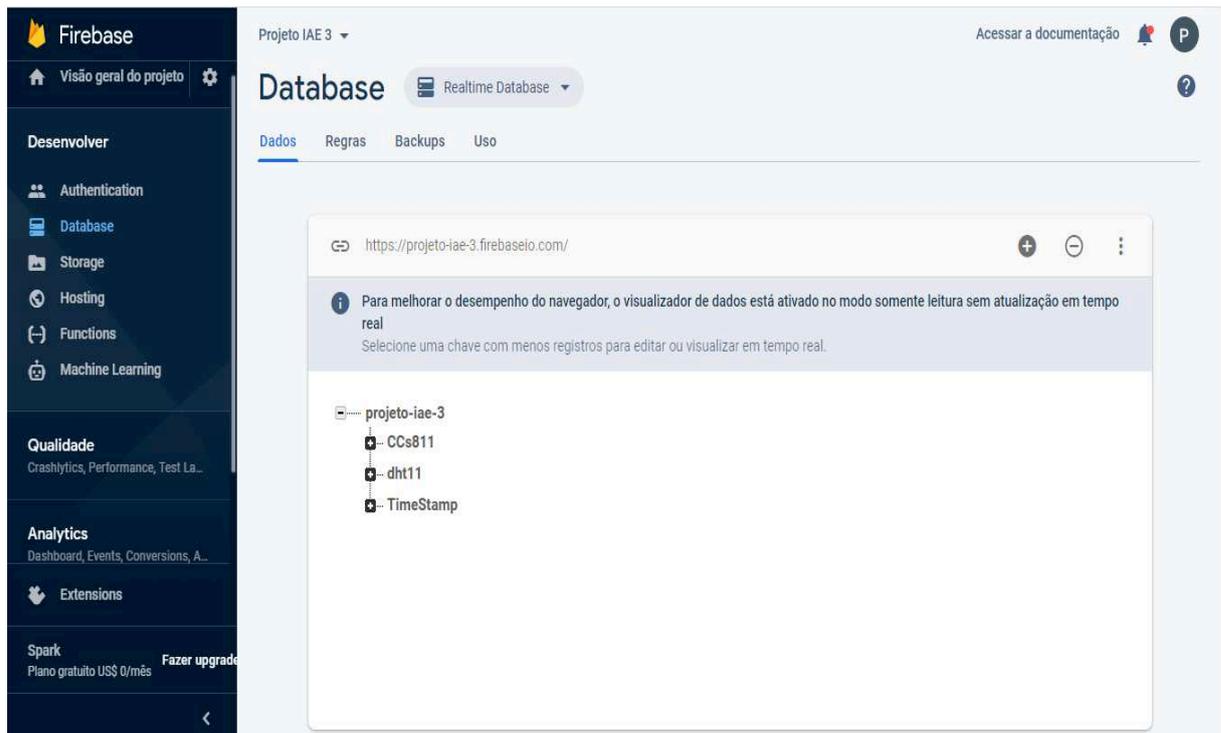
O banco de dados escolhido para armazenar os dados adquiridos pelos sensores da estação meteorológica foi o Firebase. A Figura 23 mostra o banco de dados do projeto. Além do armazenamento, este banco de dados permite a leitura e acesso aos dados em tempo real.

Figura 22 – Interface de desenvolvimento da página Web utilizando o Visual Studio Code (Versão 1.83.1)



Fonte: Próprio autor.

Figura 23 – Banco de dados do projeto da estação meteorológica utilizando o Firebase



Fonte: Próprio autor.

## 4 DESENVOLVIMENTO

Este tópico visa descrever como foi montada a estação meteorológica

O processo de construção da Base Básica, contou com uma estrutura de madeira 35cmx20cm que com espaço suficiente para a fixação dos sensores e devida alocação do sistema de alimentação.

Figura 24 – Anemômetro



Fonte: Próprio autor.

Para a fixação do sensor anemômetro foram feitos dois furos na caixa e utilizado 2 parafusos e 2 porcas para a sua fixação.

Figura 25 – Fixação do anemômetro com parafusos direto na madeira



Fonte: Próprio autor.

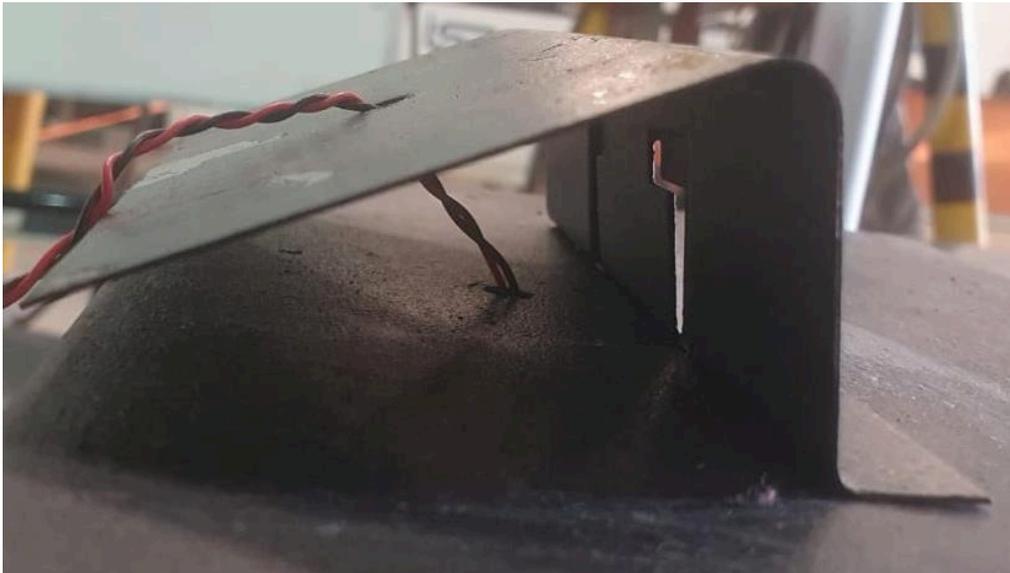
Para a alimentação baseada em energia renovável foram utilizadas 3 placas fotovoltaicas que fornecem 1,25V cada, elas foram ligadas em série para e sua saída é ligada ao TP 4056 que é um regulador de tensão que permite manter as baterias em flutuação e utilizar a outra saída para alimentar o circuito de comando.

Figura 26 – Painel solar fotovoltaico



Fonte: Próprio autor.

Figura 27 – Base para fixação dos painéis solares fotovoltaicos



Fonte: Próprio autor.

Para fixação do sensor pluviômetro foram feitos 4 furos na madeira, e utilizados 4 parafusos e 4 porcas.

Figura 28 – Pluviômetro



Fonte: Próprio autor.

Figura 29 – Fixação do Pluviômetro

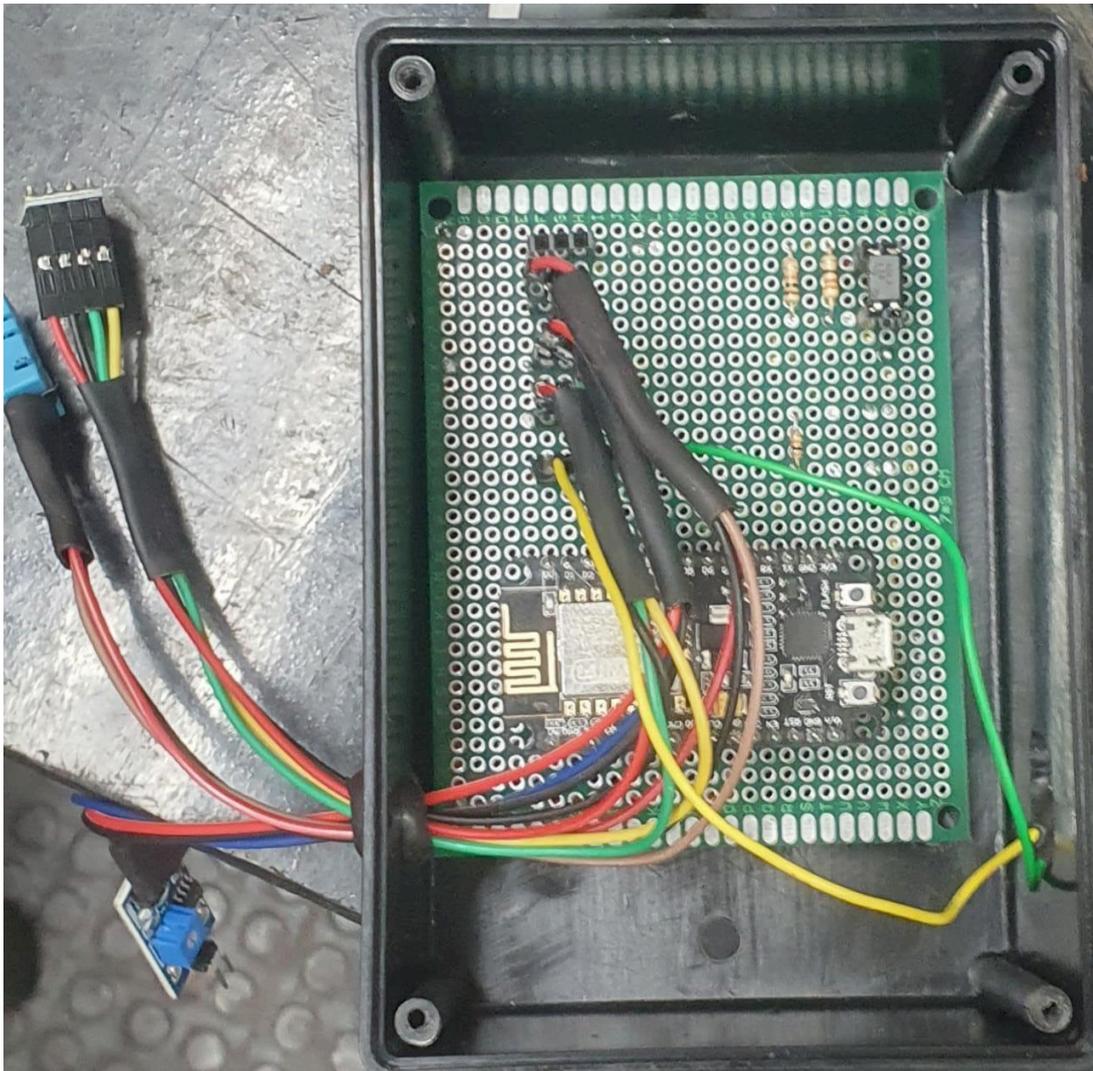


Fonte: Próprio autor.

Foi feito um circuito de controle para os sensores utilizados que foram alocados em uma caixa de plástico.

O circuito foi fabricado de uma forma a facilitar futuras manutenções, pois é fácil de se retirar os sensores e também substituir o ESP8266. Cada sensor tem um o seu circuito descriminado na placa de controle e foi fabricado um chicote para cada.

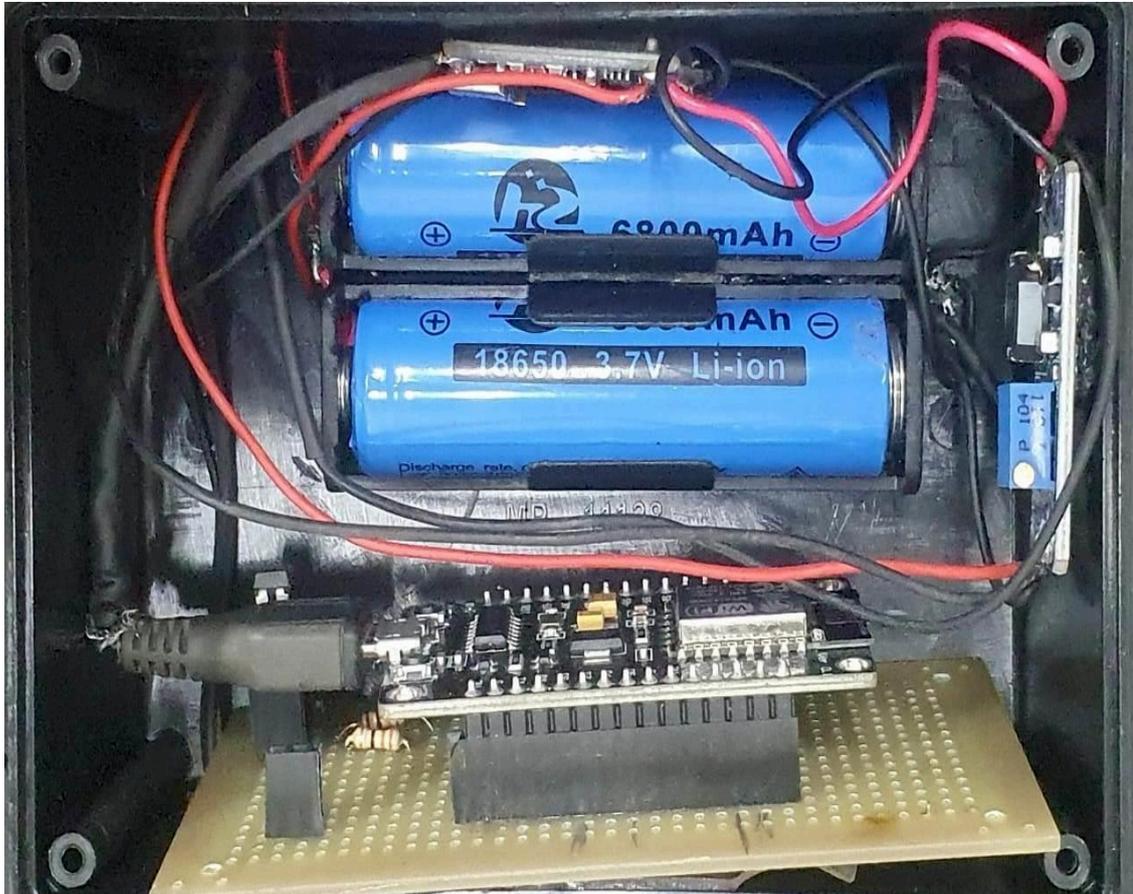
Figura 30 – Circuito de controle



Fonte: Próprio autor.

O circuito de alimentação é composto por um banco de baterias e o TP4056 que faz o controle de tensão da flutuação do banco e também alimenta os circuitos de controle. Juntamente do circuito de alimentação ficou alocado mais um ESP8266 que também é responsável auxiliar no controle dos sensores.

Figura 31 – Circuito de alimentação



Fonte: Próprio autor.

A figura 32 apresenta uma tabela com os valores investidos na construção da estação meteorológica inteligente, tendo como principal objetivo um projeto de baixo custo sem perder a confiabilidade nas medições, foi utilizado sensores e componentes presentes no mercado.

A versão final da Estação meteorológica ficou conforme mostrado na figura 33.

Figura 32 – Custo para construção da Estação Meteorológica

<b>BBMET II</b>		
<b>Sensores</b>	<b>Quantidade</b>	<b>Valor</b>
DHT 11	1	R\$ 28,00
BMP 180	1	R\$ 5,90
Sensor de Chuva	1	R\$ 8,70
Anemometro	1	R\$ 274,70
Pluviometro	1	R\$ 341,70
ESP8266	2	R\$ 65,20
<b>Componentes</b>	<b>Quantidade</b>	<b>Valor</b>
Placas	2	R\$ 10,00
Caixa Plástica	2	R\$ 36,00
Bateria	2	R\$ 29,90
TP4056	1	R\$ 4,80
MT3608	1	R\$ 7,00
MDL0029	1	R\$ 14,70
Fonte Chaveada	1	R\$ 49,00
<b>Total</b>		<b>R\$ 826,60</b>

Fonte: Próprio autor.

Figura 33 – Versão final da estação meteorológica



Fonte: Próprio autor.

## 5 RESULTADOS

Nesta seção são apresentados os resultados obtidos com o projeto. A Figura 34 mostra um diagrama de contexto da plataforma integrada aos sensores sendo acessada remotamente.

Figura 34 – Diagrama de contexto do projeto mostrando a estação meteorológica baseada em Internet das Coisas



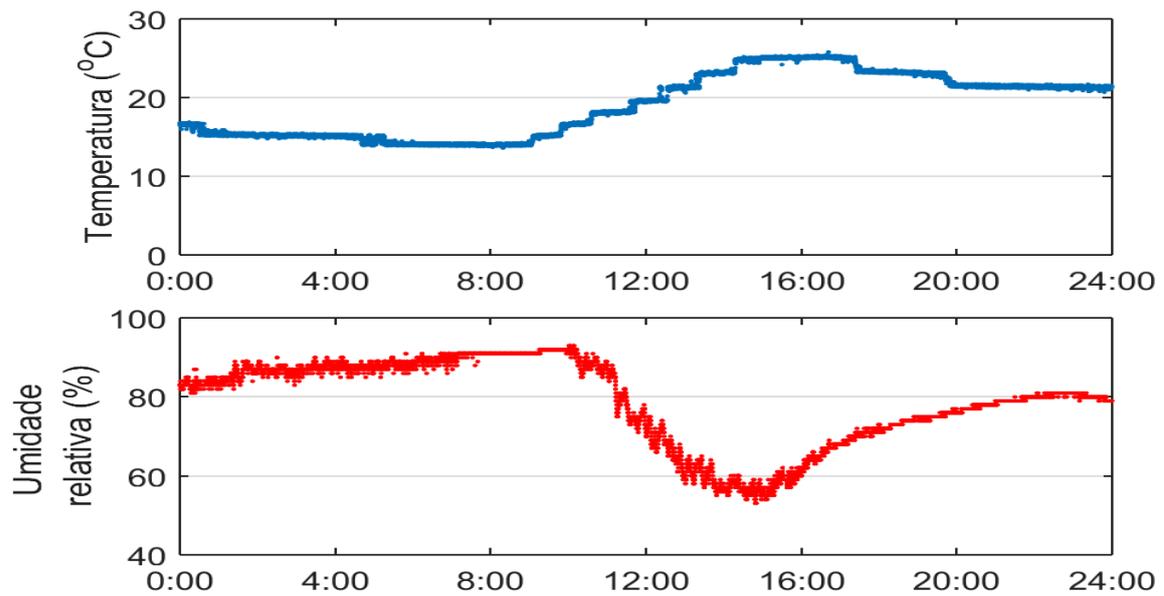
Fonte: Próprio autor.

Pode-se observar na Figura 35 os dados de temperatura e umidade relativa para um período de 24 horas durante o dia 6 de julho de 2023. Os valores e comportamento das curvas estão coerentes com o esperado, de acordo com o verificado no site do Instituto Nacional de Meteorologia (IBMET), endereço <https://tempo.inmet.gov.br/GraficosDiarios/A728>. A estação estava instalada na cidade de Taubaté. Nos gráficos observa-se que a temperatura máxima atingida foi de 25.8 °C aproximadamente as 16:00 enquanto a umidade relativa mínima foi de 53% em torno das 15:00. Este exemplo mostra que este tipo de estação nos centros de lançamento poderia realizar estas medidas e os dados poderiam ser acessados pelos especialistas do IAE de forma remota e instantânea.

Além disto, o sistema foi testado continuamente entre os dias 17/05/2023 e 07/07/2023.

Durante este período, de 53 dias, a plataforma operou de forma esperada a maior parte do tempo. A Figura 36 mostra o calendário com a classificação de operação dos dias. Nesta Figura a cor verde representa que a estação operou nominalmente como se esperava e em vermelho os dias em que houve algum problema na captura, envio ou armazenagem dos dados.

Figura 35 – Dados de temperatura e Umidade relativa no dia 06/07/2023



Fonte: Próprio autor.

As variações de umidade interferem em vários aspectos importantes para a estrutura do VLS e para o seu lançamento, e elas são:

- Vento de altitude, são ventos encontrados acima de 1000m, e no brasil as medições são feitas até 30km para ter um controle e a umidade interfere na formação desses ventos;
- Proteção de equipamentos, tendo uma análise da umidade ao qual os equipamentos serão submetidos durante sua vida;
- Condições ambientais para o instante de lançamento;
- A umidade pode interferir na estrutura de motores do VLS, no caso, se o motor for envelopado bobinado, que na sua composição tem resina e fibras, causando alguma alteração molecular, e assim degradando o material

Temperaturas, ou melhor nas suas variações durante o dia, por isso é essencial fazer uma análise dos dois gráficos juntos e, vale lembrar que em lugares com maior umidade será maior a amplitude térmica (a diferença entre a maior e a menor temperatura) costuma ser menor, pois as variações costumam ser mais constantes.

Figura 36 – Classificação do funcionamento da estação meteorológica durante o período de testes de longo prazo, entre os dias 17/05 e 07/07

MAIO						
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
JUNHO						
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
JULHO						
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Fonte: Próprio autor.

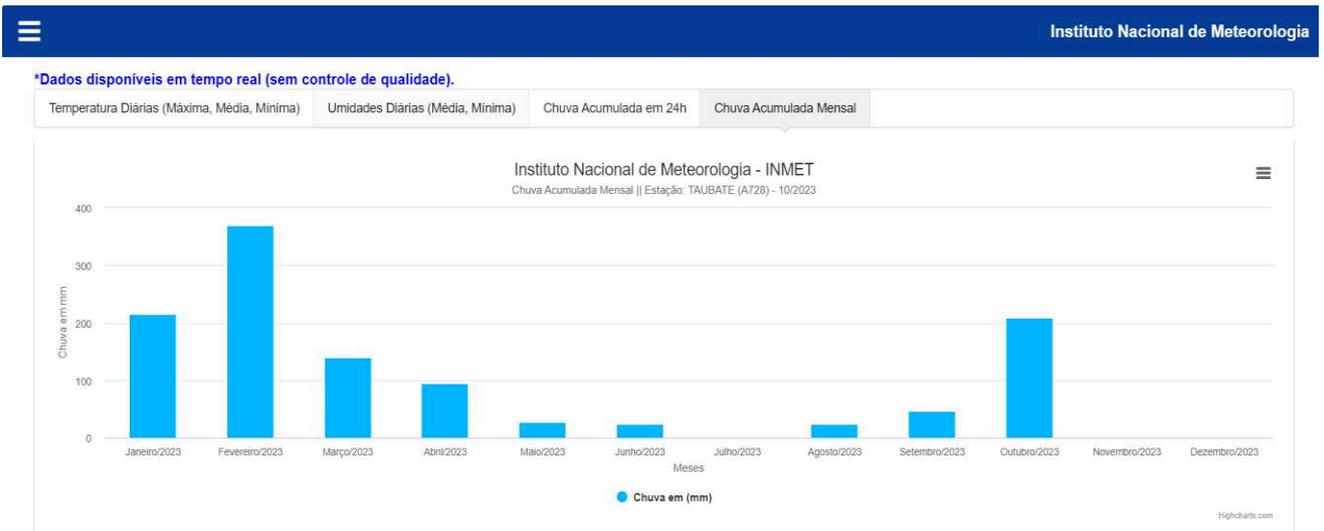
A seguir pode-se comprovar a eficiência do sensor DHT11 com bases nos dados que foram coletados no decorrer desse período. O sensor DHT11 é responsável por coletar os dados referentes a Umidade relativa do ar e de temperatura.

Foi utilizado os dados fornecidos pelo o INMET (INSTITUTO NACIONAL DE METEOROLOGIA).

É necessário salientar que o site do INMET não disponibiliza gráfico para pressão atmosférica, probabilidade de chuva e velocidade do vento.

Por esse motivo foi utilizado a temperatura e umidade para corroborar a eficiência dos dados que foram obtidos durante esse período.

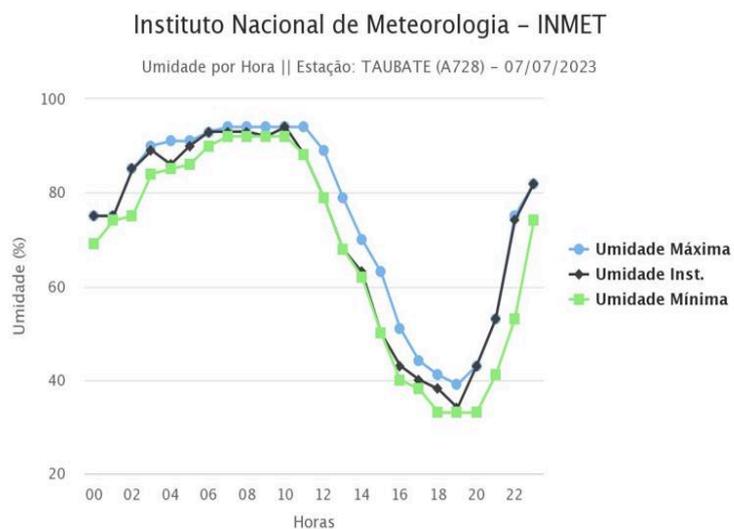
Figura 37 – Gráfico de Chuva Acumulada Mensal



Fonte: Instituto Nacional de Meteorologia INMET.

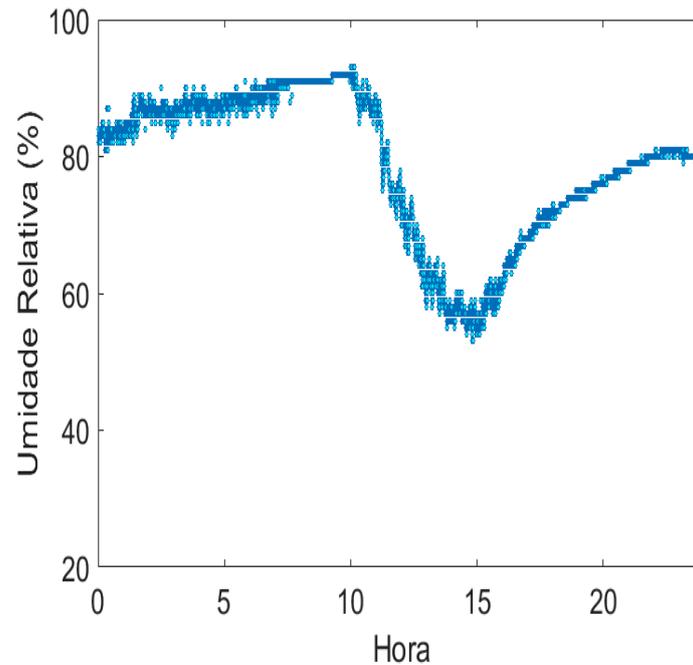
Nas figuras a seguir pode ser visualizado os gráficos referentes a Umidade, sendo a figura 38 referente ao que é disponibilizado no INMET e a figura 39 o gráfico que foi plotado através dos dados adquiridos pelo sensor.

Figura 38 – Umidade por Hora - INMET



Fonte: Instituto Nacional de Meteorologia INMET.

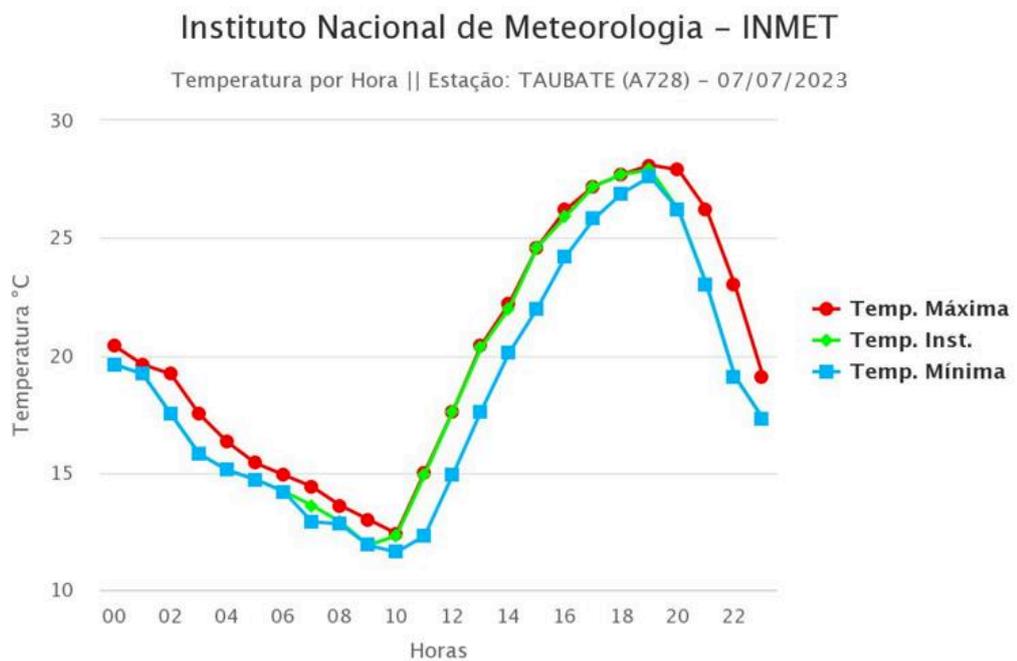
Figura 39 – Umidade Relativa por hora BBMET



Fonte: Próprio autor.

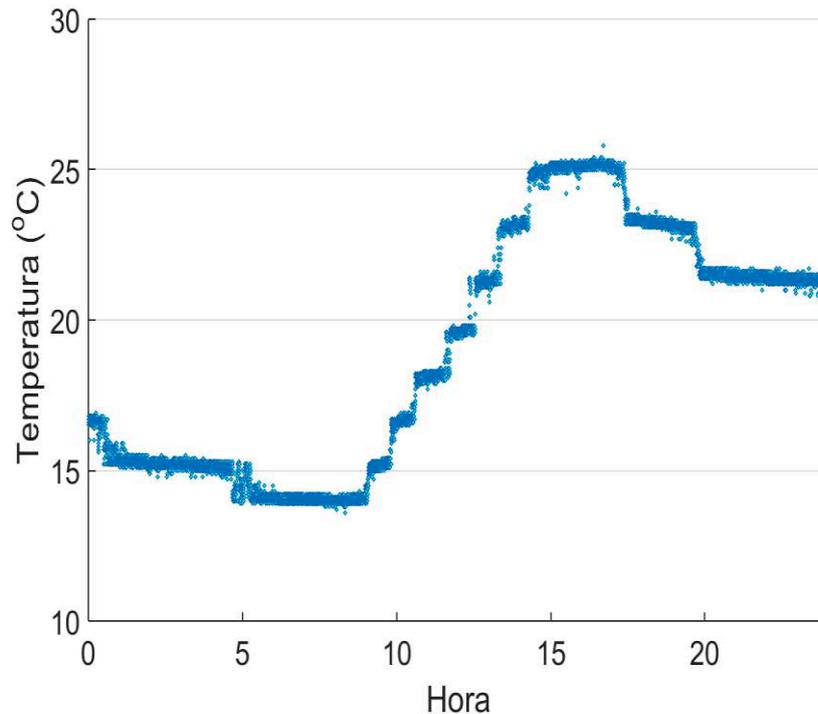
Nas figuras a seguir pode ser visualizado os gráficos referentes a Temperatura, sendo a figura 40 referente ao que é disponibilizado no INMET e a figura 41 o gráfico que foi plotado através dos dados adquiridos pelo sensor.

Figura 40 – Temperatura por Hora - INMET



Fonte: Instituto Nacional de Meteorologia INMET.

Figura 41 – Temperatura por Hora BBMET



Fonte: Próprio autor.

A figura 42 apresenta a interface Web desenvolvida onde se observa na estação os valores de mínimo, máximo, médio e atual de medição para os dados de umidade temperatura e pressão. Juntamente dessas informações, no cabeçalho são mostrados os dados relativos a data e hora.

Estes dados podem ser acompanhados em tempo real, para isto basta acessar o link <https://projeto-iae-3.firebaseio.com/>.

Figura 42 – Layout da página Web que mostra os dados da estação meteorológica em tempo real

Data / Hora: 2023/10/18 21:27:01				
	Mínimo	Máximo	Média	Atual
Umidade	49 %	95 %	82,4 %	95 %
Temperatura	18,4°C	41°C	23,0°C	27,2°C
Velocidade do Vento	0 Km/h	146,4084 Km/h	4,2 Km/h	0 Km/h
Pluviômetro	0 mm	1 mm	1,0 mm	1 mm
Pressão Atmosférica	94350 Pa	95679 Pa	95303,7 Pa	95615 Pa
Probabilidade de Chuva	0 %	80 %	19,1 %	50 %

Fonte: Próprio autor.

Importante mencionar que o projeto foi desenvolvido pensando em uma arquitetura de Internet das Coisas que prevê a possibilidade do uso de múltiplas estações no futuro. Então pode ser ampliado o número de tabelas que permitiram visualizar um número ainda maior de estações e, melhorando a confiabilidade dos dados, já que a gama dos mesmos será maior.

## 6 CONCLUSÃO

Este trabalho mostrou de forma sucinta o desenvolvimento de uma estação meteorológica de pequeno porte e de baixo custo baseada em plataforma de Internet das Coisas. Os resultados obtidos com esta plataforma mostraram a efetividade do projeto com relação à coleta de uma alta taxa de dados, que é alocada em um banco de dados, de forma remota e instantânea. Claramente este desenvolvimento foi uma prova de conceito de um sistema que pode no futuro auxiliar o lançamento de foguetes no Centro de Lançamento de Alcântara no Maranhão e no Centro de Lançamento da Barreira do Inferno no Rio Grande do Norte.

Este trabalho permitiu que, a formação e consolidação de uma série de conceitos relacionados às áreas de Linguagem de programação, Front End, Circuitos integrados e Meteorologia.

Trouxe também uma vivência singular, por parte dos alunos, de temáticas da área espacial, relacionadas às áreas de eletrônica e computação, neste caso aplicado a temática da meteorologia. Vale citar por exemplo que ao longo deste projeto os alunos se familiarizaram com 3 linguagens de programação diferentes além da bagagem obtida com a utilização de circuitos integrados e instrumentação.

Esta experiência é um grande diferencial devido ao conhecimento obtido neste tipo de atividade prática, que dificilmente se experimentaria em um ambiente de sala de aula.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

OLIVEIRA, Sergio. **Internet das coisas**: com ESP8266, Arduino e Raspberry Pi. São Paulo: Novatec, 2017.

GHARSELLAOUI, Bayrem. **Internet of Things prototyping with Firebase**: how to do more with less. 19 jan. 2019.

Disponível em: <https://medium.com/free-code-camp/iot-prototyping-with-firebase-doing-more-with-less-2f5c746dac8b>. Acesso em: 5 jul. 2020.

KATYAL, Amber; YADAV, Ravi; PANDEY, Manoj. **Wireless Arduino Based Weather Station**, [s. l.], 2016.

VIEBRANTZ, Alvaro. Sensoriamento e Controle Realtime com Firebase e ESP8266. **Medium**, 2019. Disponível em:

<https://medium.com/@alvaroviebrantz/sensoriamento-realtime-com-firebase-e-esp8266-6e54b9bff1c1>. Acesso em: 23 set. 2020.

THOMSEN, Adilson. Saiba como funciona o módulo TP4056. **Maker Hero**, 2022. Disponível em: <https://www.makerhero.com/blog/saiba-como-funciona-o-modulo-tp4056/>. Acesso em: 23 fev. 2023.

GUSTAVO ABREU MURTA, José. Sensores DHT11 e DHT22: Guia Básico dos Sensores de Umidade e Temperatura. **Blog Eletrogate**, 2019. Disponível em: <https://blog.eletrogate.com/sensores-dht11-dht22/>. Acesso em: 23 set. 2020.

OLIVEIRA, Euler. Como usar com Arduino – Sensor de Pressão e Temperatura BMP180. **Master Walker**, 2020. Disponível em:

<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-pressao-e-temperatura-bmp180>. Acesso em: 17 fev. 2023.

PALMERIO, Ariovaldo Felix. **Introdução a tecnologia de Foguetes**. São José dos Campos: SindCT, 2017.

FERREIRA GISLER, Carlos Alberto . **ANALISE DO PERFIL DE VENTO NA CAMADA LIMITE SUPERFICIAL E SISTEMAS METEOROLOGICOS ATUANTES NO CENTRO DE LANÇAMENTO DE ALCANTARA**. São José dos Campos: INPE, 2009.

## APENDICE 1

### PROGRAMAÇÃO ESP8266 PARA 5 SENSORES

```
//===Definições de BIBLIOTECAS===//

Serial.print(" [km/h] ");
Serial.println();
//vm = vm + speedwind;
if (speedwind > vmax)
{
  vmax = speedwind;
}

ESP.wdtFeed(); // feeds the dog
yield();
delay(3100); // Tem que ser maior que o periodo
}

void addcount()
{
  counter++;
}

//---Inicio do Loop----//

void loop()
{

  // Obtem os dados dos sensores
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float pr = bmp.readPressure();
  float w = speedwind;
  float sc = Porcento;
  float pl = (REEDCOUNT*0.25);

  //BMP180 - Pressão//

  Serial.print("Pressão: ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa (Pascal)");

  Serial.println("-----");
  delay(2000);

  //Obtem os dados do sensor de chuva
```

```
chuva = analogRead(PinSensor);  
Porcento = map(chuva, 1023, 0, 0, 100);
```

```
Serial.println("Sensor de Chuva");  
Serial.print(Porcento);  
Serial.println("%");
```

```
delay(1000);
```

```
if (Porcento >= 20)  
{  
  Serial.println("Chovendo!");  
  digitalWrite(13, HIGH);  
  delay(1000);  
}
```

```
else  
{  
  digitalWrite(13, LOW);  
}
```

```
// Obtem os dados dos sensores DHT11  
Serial.print("Umidade: ");  
Serial.print(h);  
Serial.print("%");  
Serial.println();
```

```
Serial.print("Temperatura: ");  
Serial.print(t);  
Serial.print("C");  
Serial.println();
```

```
// Dados Anemometro  
windvelocity();  
Serial.println();
```

```
delay(1000);
```

```
// Dados Pluviometro  
pluv();  
Serial.println();
```

```
delay(1000)
```

## APENDICE 2

### PROGRAMAÇÃO ESP8266 PARA O SENSOR PLUVIOMETRO

```
//===Definições de BIBLIOTECAS===//

#include <ArduinoJson.h>
#include <FirebaseArduino.h>
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <Wire.h>

}
Serial.println();
Serial.print("Conectado: ");
Serial.println(WiFi.localIP());

//----NTP CLiet----//
timeClient.begin(); // inicializa NTPClient

timeClient.setTimeOffset(-3600 * 3); // Seta para GTM -3 (horario de Brasilia)
while (!timeClient.update())
{ // Altera GTM
  timeClient.forceUpdate();
}
}

//---Inicio do Loop---//

void loop()
{
  val = digitalRead(REED); //Read the status of the Reed switch

  if ((val == LOW) && (old_val == HIGH)){ //Check to see if the status has
changed

  delay(10); // Delay put in to deal with any "bouncing" in the
switch.

  REEDCOUNT = REEDCOUNT + 1; //Add 1 to the count of bucket tips

  old_val = val; //Make the old value equal to the current value
  Serial.print("Medida de chuva (contagem): ");
  Serial.print(REEDCOUNT);/*0.2794);
```

```

    Serial.println(" pulso");
    Serial.print("Medida de chuva (calculado): ");
    Serial.print(REEDCOUNT*0.25);
    Serial.println(" mm");

}

else {

old_val = val;          //If the status hasn't changed then do nothing

}

//----Manda para o firebase a indice de Pluviometro----//

String PluvID = Firebase.pushFloat("Pluviometro/Pluv: ",
(REEDCOUNT*0.25));
if (Firebase.failed())
{
    Serial.print("[ERRO] ao enviar /Pluviometro/Pluv falhou:");
    Serial.println(Firebase.error());
    return;
}
Serial.print("[INFO] enviado: /Pluviometro/Pluv \tkey: ");
Serial.println(PluvID);
Serial.println();
delay(100);
}

```

### APENDICE 3

## PROGRAMAÇÃO DO SITE

```
<!DOCTYPE html>
<html>
<head>
```

```
    <td id="PI_atual" ></td>
```

```
</tr>
```

```
</div>
<!--container-->
<!-- Firebase App -->
<script>
```

```
    // Função para atualizar um gráfico comum
    function atualizarGráfico(gráfico, elementoAtual, maxRegistros = 4330) {
        var dataAtual = new Date();
        var horas = dataAtual.getHours().toString().padStart(2, '0');
        var minutos = dataAtual.getMinutes().toString().padStart(2, '0');
        var segundos = dataAtual.getSeconds().toString().padStart(2, '0'); // Obtendo
os segundos
        var horaAtual = `${horas}:${minutos}:${segundos}`;

        // Adicionar a nova hora aos rótulos
        gráfico.data.labels.push(horaAtual);

        // Obter o valor atual do elemento
        var valorAtual = parseFloat(elementoAtual.innerText); // Converte para
número

        // Adicionar o valor aos dados do gráfico
        gráfico.data.datasets[0].data.push(valorAtual);

        // Limite o número de rótulos e dados para manter o gráfico limpo
        if (gráfico.data.labels.length > maxRegistros) {
            gráfico.data.labels.shift();
            gráfico.data.datasets[0].data.shift();
        }
    }
```

```
// Atualizar o gráfico
grafico.update();
}

// Função para criar um novo gráfico
function criarGrafico(elemento, maximo = 0) {
  return new Chart(elemento, {
    type: 'line',
    data: {
      labels: [],
      datasets: [{
        data: [],
        backgroundColor: 'white',
        borderColor: '#00ff8d',
        fill: false,
      }]
    },
    options: {
      legend: {
        display: false
      },
      scales: {
        yAxes: [{
          ticks: {
            min: 0,
            max: maximo // Define o máximo se necessário
          }
        }]
      }
    }
  });
}

</body>
</html>
```

## APENDICE 4

## PROGRAMAÇÃO CSS PARA O LAYOUT DA PÁGINA

```
body{
  background: #212529;
}

#container{
  margin:2%;
}

#cnpq{
  padding-left: 20px;
  padding-bottom: 35px;
  width: 120px;
  transition: all 1.5s ease-in-out;
}
#cnpq:hover {
  transform: rotateY(180deg);
}

#iae{
  width: 75px;
  transition: all 1.5s ease-in-out;
}
#iae:hover {
  transform: rotateY(180deg);
}

#unitau{
  width: 75px;
  transition: all 1.5s ease-in-out;
}
#unitau:hover {
  transform: rotateY(180deg);
}

#autor{
  color: #ffffffcc;
  float: right;
}

.box{
  color: #d4d4d4;
  text-align: center;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
```

```
}

@keyframes example {
  25% {left:0px; top:0px;}
  0% {left:400px; top:0px;}
}

h3{
  color: #00ff8d;
}

.tempo{
  color: white;
  background-color: #00ff902e;
  display: flex;
  flex-direction: column;
  justify-content: space-evenly;
  align-items: center;
  border-radius: 8px 8px 0 0;
  height: 37px;
}

.titulo-coluna{
  color: #00ff8d;
}
h2{
  color: white;
}

#graficos{
  color: #9e9e9e;
  text-align: center;
  margin: 50px;
  font-size: x-large;
  text-align: -webkit-center;
}

#tt-gr{
  margin-top: 33px;
}

.gr{
  width:100%;
  max-width:510px;
}

h1{
  color:#e9f0e9;
```

```
margin-left:43%;
}
#plots{
margin:auto;
background-color: #2e2d2d;
}

footer{
height: 100px;
background-color: #2d2d32;
margin-top: 46px;
}
```

```
@media only screen and (max-width: 1220px) {
```

```
#graficos {
margin: 2%;
}
.gr{
max-width: 100%;
}
```

```
#pluviometro{
max-width:510px;
}
}
```

```
@media only screen and (max-width: 750px) {
```

```
.titulo-coluna{
font-size: 15px;
}

#graficos {
margin: 3%;
}

.gr{
max-width: none;
}
}
```

## APENDICE 5

### COMUNICAÇÃO DA PÁGINA COM O FIREBASE

```

(function() {
  // Configuração do meu App
  const schuElement = document.getElementById("Pc_atual");

  let timeStampArray=[]; //Array é deifnido pela chave; para deocarar uma variável
no Js melhor usar let
  // Sync objects changes
  timeStampRef.limitToLast(8640).on('value', function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var childData = childSnapshot.val();
      timeStampArray.push(childData);
    });
    console.log(timeStampArray);

    let last = timeStampArray.length;

    timeStampElement.innerHTML = timeStampArray[last-1]
  });

  let tempArray=[]; //Array é deifnido epla chave; para declarar uma variável no Js
melhor usar let
  // Sync objects changes
  tempRef.limitToLast(8640).on('value', function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var childData = childSnapshot.val();
      tempArray.push(childData);
    });
    console.log(tempArray);

    let humArray=[];
  //Sync objects changes
  humRef.limitToLast(8640).on('value', function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var childData = childSnapshot.val();
      humArray.push(childData);
    });
    console.log("humidade: " + humArray);
  });

  console.log(b180Array);

  let last = b180Array.length;

```

```
    b180Element.innerText = b180Array[last-1] + " Pa";
    b180_max.innerText = Math.max(... b180Array) + " Pa";
    b180_min.innerText = Math.min(... b180Array) + " Pa";
    b180_media.innerText = media(b180Array).toFixed(1) + " Pa";
  });

  let VVentoArray=[];
  //Sync Objects changes
  VVentoRef.limitToLast(8640).on('value', function(snapshot) {
  snapshot.forEach(function(childSnapshot) {
    var childData = childSnapshot.val();
    VVentoArray.push(childData);
  });

  console.log(VVentoArray);

  let last = VVentoArray.length;
});

  console.log(PluvArray);

  let last = PluvArray.length;

  PluvElement.innerText = PluvArray[last-1] + " mm";
  Pluv_max.innerText = Math.max(... PluvArray) + " mm";
  Pluv_min.innerText = Math.min(... PluvArray) + " mm";
  Pluv_media.innerText = media(PluvArray).toFixed(1) + " mm";
});
})();
```

