



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE ENERGIAS RENOVÁVEIS

LUCAS SILVA MENESES

PROTÓTIPO DE ESTAÇÃO METEOROLÓGICA UTILIZANDO PLATAFORMA IOT

FORTALEZA

2023

LUCAS SILVA MENESES

PROTÓTIPO DE ESTAÇÃO METEOROLÓGICA UTILIZANDO PLATAFORMA IOT

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Energias Renováveis do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Energias Renováveis.

Orientador: Prof. Dr. Francisco Nivaldo Aguiar Freire.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M488p Meneses, Lucas Silva.

Protótipo de estação meteorológica utilizando plataforma IOT / Lucas Silva Meneses. – 2023.
88 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia,
Curso de Engenharia de Energias Renováveis, Fortaleza, 2023.

Orientação: Prof. Dr. Francisco Nivaldo Aguiar Freire.

1. Estação meteorológica. 2. Iot. I. Título.

CDD 621.042

LUCAS SILVA MENESES

PROTÓTIPO DE ESTAÇÃO METEOROLÓGICA UTILIZANDO PLATAFORMA IOT

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Energias Renováveis do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Energias Renováveis.

Aprovada em: 07/12/2023.

BANCA EXAMINADORA

Prof. Dr. Francisco Nivaldo Aguiar
Freire (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Amanda Pereira Monteiro
Universidade Federal do Ceará (UFC)

Prof. Me. Paulo Herbert França Maia Júnior
Faculdade CEPEP (CEPEP)

A Deus e à minha família.

AGRADECIMENTOS

Gostaria de agradecer a toda a minha família pela educação, por todos os ensinamentos e por todo o apoio nos momentos bons e ruins que sempre me deram ao longo minha trajetória.

Gostaria de agradecer também a todos os meus amigos da universidade que fizeram parte desta caminhada, a ajuda deles foi fundamental para que eu chegasse até aqui. Aos meus amigos de infância, que me acompanham desde os tempos de colégio, também gostaria de deixar meu agradecimento por todos esses anos de amizade e apoio.

Ademais, gostaria de agradecer ao meu professor orientador Nivaldo, que me ajudou muito durante minha trajetória acadêmica e a quem sou muito grato, e também agradecer aos amigos do Laboratório de Filmes Finos e Energias Renováveis da Universidade Federal do Ceará, em especial ao amigo Paulo Herbert, que muito me ensinou ao longo dos projetos em que trabalhamos juntos no laboratório.

E obrigado a Universidade Federal do Ceará e a todo o corpo docente por todas as oportunidades e ensinamentos fornecidos ao longo desses anos.

"A imaginação é mais importante que o conhecimento. O conhecimento é limitado. A imaginação envolve o mundo." (Albert Einstein)

RESUMO

O presente trabalho visa desenvolver um protótipo de uma estação meteorológica utilizando plataformas IoT, alinhado com duas grandes tendências globais: a de criação de “dispositivos inteligentes” e a de desenvolvimento de ferramentas que auxiliem no desenvolvimento de projetos de geração de energia a partir de fontes renováveis. A estação meteorológica desenvolvida será capaz de coletar dados de velocidade do vento, de temperatura, de umidade e de incidência de radiação solar global do local em que ela for instalada; será capaz de armazenar os dados coletados em um data logger, será capaz de armazenar os dados coletados remotamente em sistemas de computação em nuvem e poderá ser acompanhada remotamente e em tempo real pela internet, com dashboard de visualização. A estação será controlada por duas placas microcontroladoras, uma placa Arduino Mega 2560 e uma placa ESP 01, que serão responsáveis por integrar todos os sensores, comandar a coleta, comandar o armazenamento dos dados e comandar o compartilhamento dos dados com o sistema de computação em nuvem. Para coletar a velocidade do vento, a estação terá em seu sistema um anemômetro. Para coletar a temperatura e a umidade, a estação contará com um sensor DHT11, capaz de fornecer dados sobre estes dois agentes meteorológicos. Já para coletar a radiação solar global, a estação contará com um piranômetro.

Palavras-chave: Estação Meteorológica; Arduino; ESP; IoT; Sensores; Hall; DHT11; Placa Solar; Anemômetro; Piranômetro; Termômetro; Higrômetro; Velocidade do Vento; Temperatura; Umidade; Radiação Solar.

ABSTRACT

The present work aims to develop a prototype of a weather station using IoT platforms, aligned with two major global trends: the creation of 'smart devices' and the development of tools that assist in renewable energy projects. The developed weather station will be capable of collecting data on wind speed, temperature, humidity, and global solar radiation incidence at its installation location. It will be able to store the collected data in a data logger, remotely store the collected data in cloud computing systems, and can be monitored remotely and in real-time over the internet with a visualization dashboard. The station will be controlled by two microcontroller boards, an Arduino Mega 2560 and an ESP 01 board, which will be responsible for integrating all sensors, commanding data collection, data storage, and data sharing with the cloud computing system. To measure wind speed, the station will have an anemometer in its system. For temperature and humidity data collection, the station will utilize a DHT11 sensor capable of providing information on these two meteorological agents. Finally, to measure global solar radiation, the station will incorporate a pyranometer.

Keywords: Weather Station; Arduino; ESP; IoT; Sensors; Hall; DHT11; Solar Panel; Anemometer; Pyranometer; Thermometer; Hygrometer; Wind Speed; Temperature; Humidity; Solar Radiation.

LISTA DE FIGURAS

Figura 1 – Gráfico de emissão de CO ₂ originário da queima de combustíveis fósseis e processos industriais	14
Figura 2 – Médias de temperatura global entre 1979 a 2023	15
Figura 3 – Concentração Média Mensal de CO ₂ em PPM	15
Figura 4 – Participação anual das fontes renováveis no aumento da capacidade de geração de energia	16
Figura 5 – Crescimento da capacidade de geração de energia	17
Figura 6 – Sensor de efeito Hall	21
Figura 7 – Estrutura completa do anemômetro, com o sensor, carcaça e ímãs	21
Figura 8 – Sensor DHT11	22
Figura 9 – Estrutura que abriga o sensor DHT11	23
Figura 10 – Placa solar utilizada como sensor do piranômetro	23
Figura 11 – Placa Arduino Mega 2560	25
Figura 12 – IDE da Plataforma Arduino e exemplo de código escrito na linguagem padrão	26
Figura 13 – Placa ESP 01	26
Figura 14 – Esquema de ligação entre dispositivos para realização de comunicação serial	27
Figura 15 – Diagrama de conexões do protótipo da estação meteorológica	30
Figura 16 – Diagrama lógico do sistema do protótipo	31
Figura 17 – Amostra de dados salvos na nuvem	34
Figura 18 – Amostra de dados salvos no data logger	34
Figura 19 – Gráfico gerado no Excel com os dados de corrente salvos no data logger . .	35
Figura 20 – Gráfico gerados no próprio sistema de nuvem com os dados de corrente lá salvos	35
Figura 21 – Dados de referência dos dias de coleta	36
Figura 22 – Local de coleta no dia 04/10/2023 - 16:20 e 16:27	37
Figura 23 – Gráficos de coleta de corrente entre 16:00 - 16:59 dos dias 01 e 02/10/2023 .	37
Figura 24 – Gráficos de coleta de corrente entre 16:00 - 16:59 dos dias 04, 05 e 06/10/2023	38
Figura 25 – Estrutura montada do termômetro/higrômetro	39
Figura 26 – Módulo de leitura e escrita de dados no cartão sd e cartão sd	40
Figura 27 – Sistema de nuvem AdaFruit	40
Figura 28 – Comparação entre os dados coletados e os dados de referência do dia 01/10/2023	41

Figura 29 – Comparação entre os dados coletados e os dados de referência do dia 02/10/2023	42
Figura 30 – Comparação entre os dados coletados e os dados de referência do dia 04/10/2023	42
Figura 31 – Comparação entre os dados coletados e os dados de referência do dia 05/10/2023	43
Figura 32 – Comparação entre os dados coletados e os dados de referência do dia 06/10/2023	43
Figura 33 – Obtenção da equação	44
Figura 34 – Circuito do protótipo	46
Figura 35 – Protótipo em funcionamento	47
Figura 36 – Data logger do sistema	48
Figura 37 – Banco de dados remoto do sistema	48
Figura 38 – Dashboard de monitoramento da estação	49

LISTA DE ABREVIATURAS E SIGLAS

CPU	Unidade central de processamento (<i>Central Processing Unit</i>)
IDE	Ambiente de desenvolvimento integrado
IEA	International Energy Agency
INMET	Instituto Nacional de Meteorologia
IoT	Internet das coisas
IRENA	International Renewable Energy Agency
LAFFER	Laboratório de Filmes Finos e Energias Renováveis
NOAA	National Oceanic and Atmospheric Administrations
RPM	Rotações por minuto
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UFC	Universidade Federal do Ceará
Wi-Fi	Wireless Fidelity

LISTA DE SÍMBOLOS

<i>RPM</i>	Rotações por minuto
<i>TG</i>	Total de giros do eixo do anemômetro
<i>S</i>	Total de sinais do sensor hall
<i>R</i>	Raio do anemômetro em metros
<i>V</i>	Velocidade do vento em quilômetros por hora
<i>rad</i>	Radiação solar global em watt hora por metro quadrado
<i>i</i>	Corrente elétrica gerada na placa solar em miliampéres

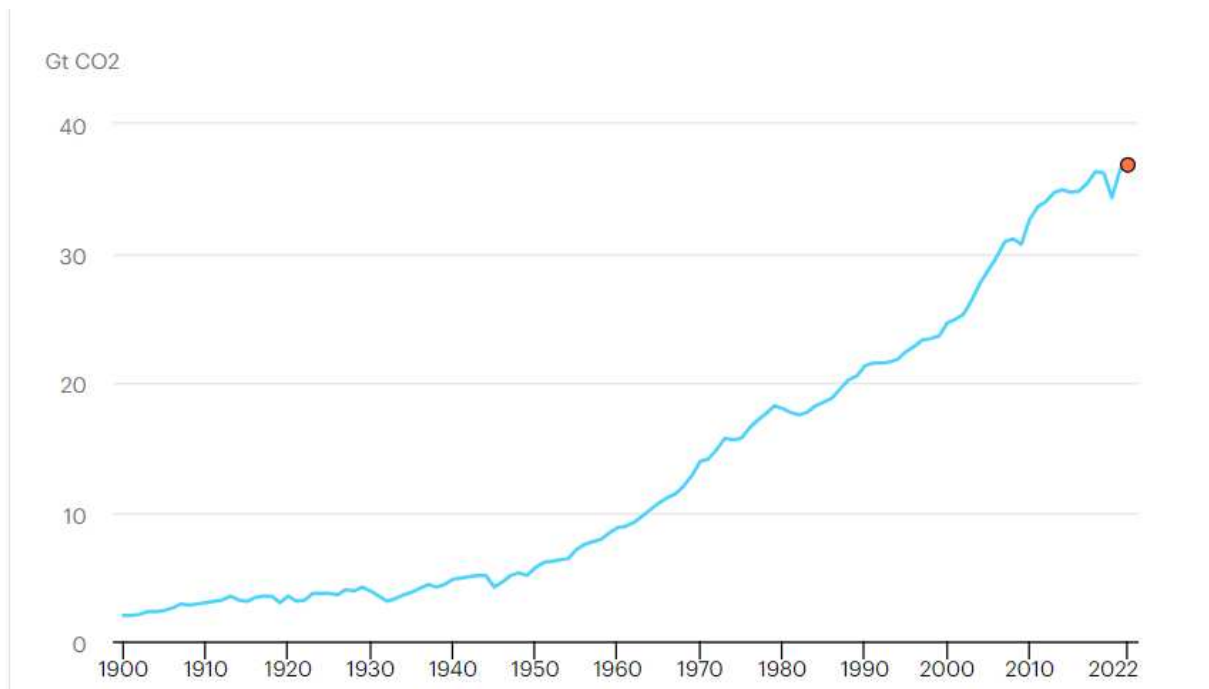
SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Estação Meteorológica	19
2.2	IoT	19
2.3	Sensores	20
2.3.1	<i>Anemômetro</i>	20
2.3.2	<i>Termômetro e Higrômetro</i>	21
2.3.3	<i>Piranômetro</i>	22
2.4	Placas Microcontroladoras	24
2.4.1	<i>Arduino</i>	24
2.4.1.1	<i>IDE e Linguagem de programação Arduino</i>	25
2.4.2	<i>ESP</i>	25
2.5	Comunicação Serial	27
2.6	Sistema de computação em nuvem	28
3	METODOLOGIA	29
3.1	Materiais	29
3.2	Diagrama esquemático do protótipo da estação	30
3.3	Diagrama lógico do sistema do protótipo da estação	30
3.4	Construção do anemômetro e o cálculo da velocidade do vento	32
3.5	Construção do piranômetro	33
3.5.1	<i>Coleta dos dados de corrente</i>	33
3.5.2	<i>Filtragem dos dados coletados de corrente</i>	35
3.6	Construção do termômetro e do higrômetro	37
3.7	Construção do data logger	38
3.8	Sistema de computação em nuvem AdaFruit	38
4	RESULTADOS E DISCUSSÕES	41
4.0.1	<i>Resultados da obtenção da equação para cálculo da radiação solar global</i>	41
4.0.2	<i>Resultados finais do trabalho</i>	45
5	CONCLUSÕES E TRABALHOS FUTUROS	50
	REFERÊNCIAS	51

1 INTRODUÇÃO

Toda a sociedade moderna depende da eletricidade para se sustentar e continuar a se desenvolver. Porém, após décadas de desenvolvimento baseado em uma matriz energética composta primariamente por fontes de energia oriundas de combustíveis fósseis, que são extremamente poluentes, as consequências negativas começaram a surgir. Segundo a National Oceanic and Atmospheric Administrations (NOAA), os 10 anos mais quentes já registrados ocorreram entre 2010 e 2022. De acordo com a International Energy Agency (IEA), em 2022, o mundo atingiu um novo recorde de emissões de CO₂ originários da queima de combustíveis fósseis e processos industriais, com 36,8 bilhões de toneladas lançadas na atmosfera, como pode ser visto na figura 1.

Figura 1 – Gráfico de emissão de CO₂ originário da queima de combustíveis fósseis e processos industriais

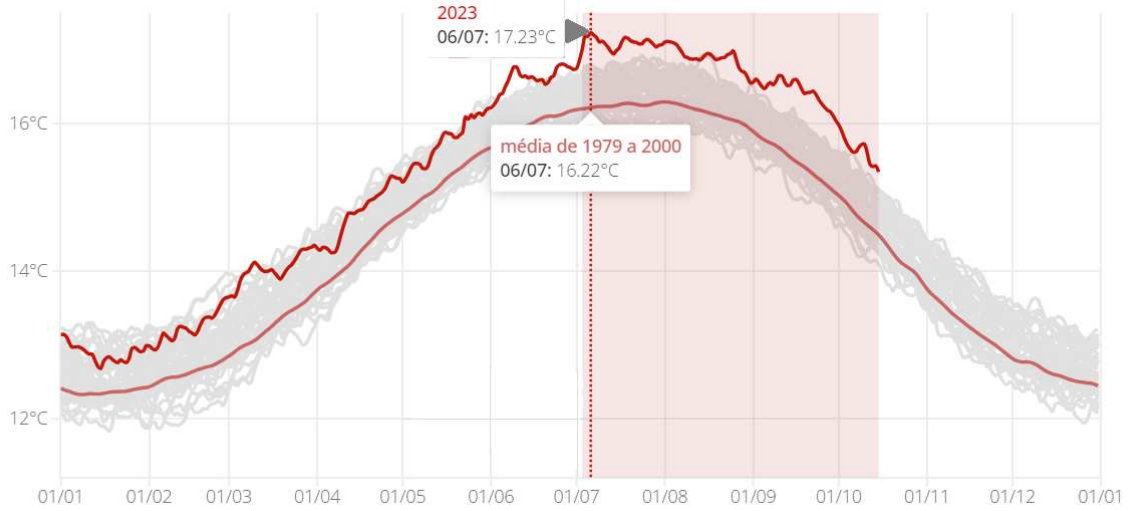


Fonte: (IEA, 2023).

O mês de julho de 2023 bateu três vezes o recorde de dia mais quente já registrado na história, como pode ser observado na figura 2. No dia 3 de julho, a temperatura média global registrada foi de 17,01°C, superando a marca anterior de 16,92°C, registrada em 2016. Porém, o dia 4 de julho bateu o recorde novamente, atingindo uma média de 17,18°C. Por fim, no dia 6 de julho o recorde foi batido mais uma vez, com a média de temperatura global atingindo 17,23°C. A concentração média mensal de dióxido de carbono vem em constante crescimento nas últimas

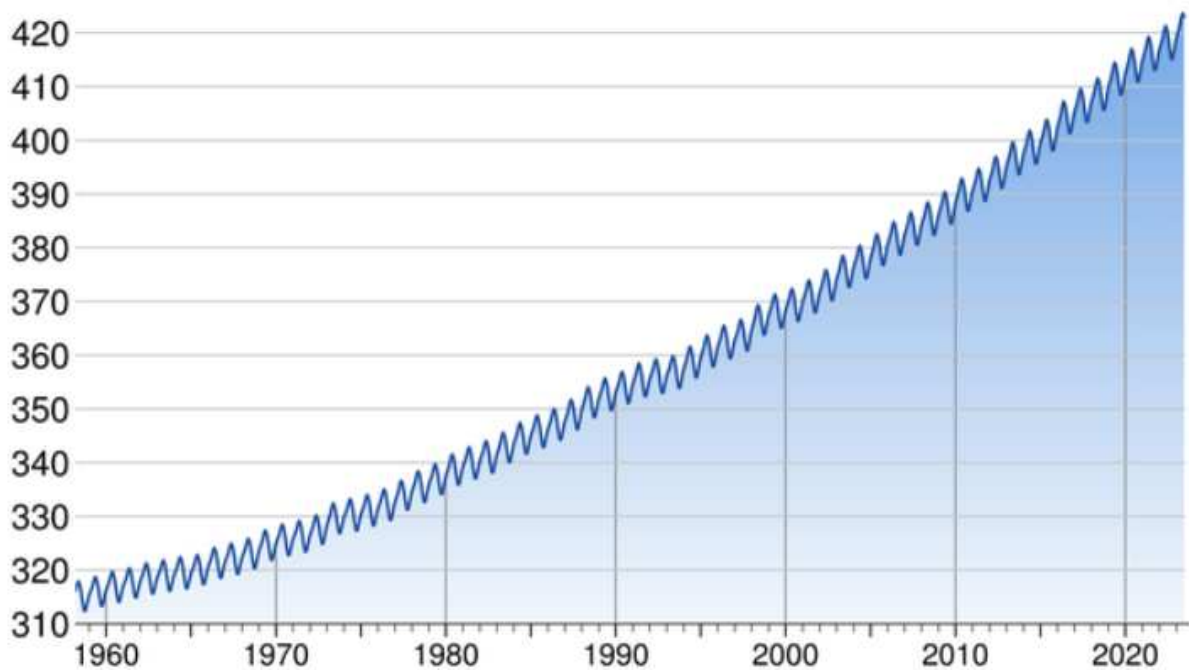
décadas, batendo recordes ano após ano, como pode ser observado na figura 3.

Figura 2 – Médias de temperatura global entre 1979 a 2023



Fonte: (GLOBO, 2023).

Figura 3 – Concentração Média Mensal de CO₂ em PPM

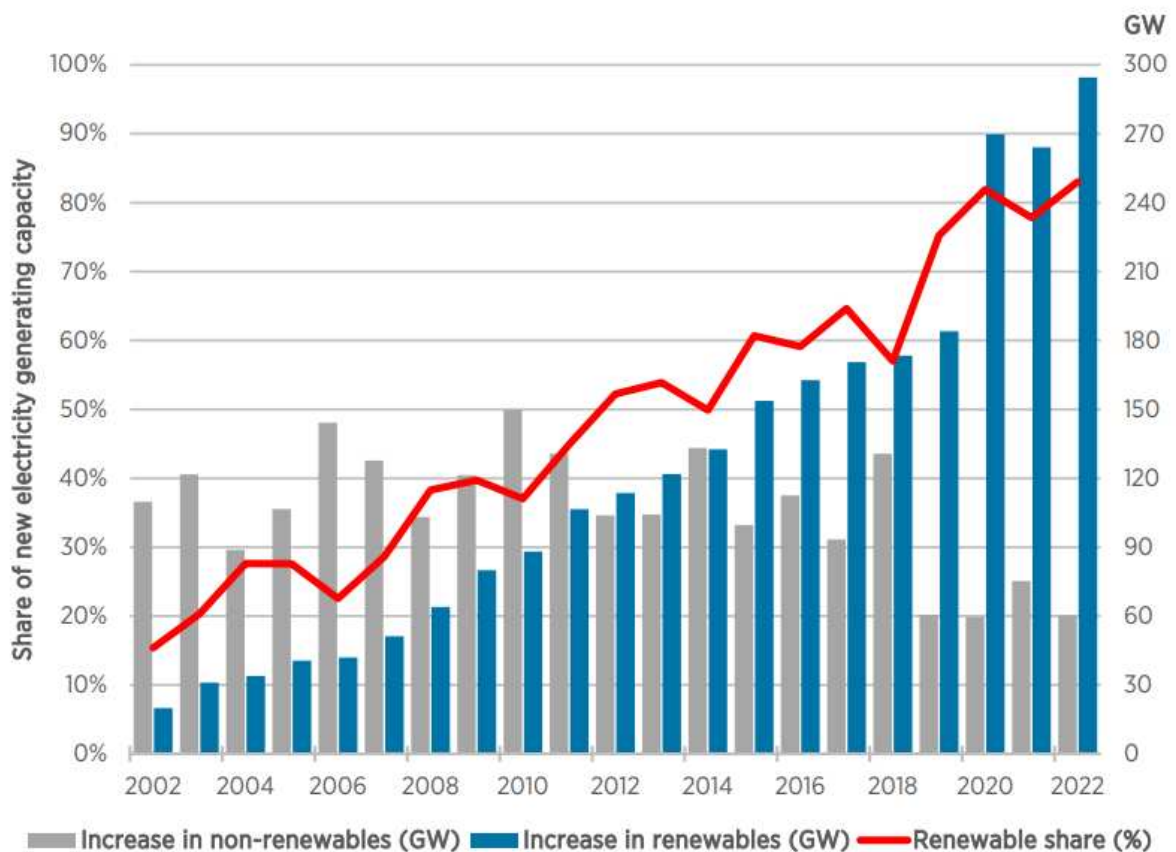


Fonte: (OCEANOGRAPHY, 2023).

Toda essa poluição vem tornando cada vez mais frequente a ocorrência de fenômenos naturais extremos, como secas, tempestades, queimadas, dentre outros. Com isso, se torna urgente a diminuição da emissão de gases que aumentam o efeito estufa na atmosfera. É uma das principais formas de tentar amenizar os desequilíbrios ambientais causados pela poluição

atmosférica é o investimento em fontes renováveis de energia. Nos últimos anos, temos visto um grande aumento do interesse global no desenvolvimento de fontes renováveis de geração de energia. Os investimentos estão cada vez maiores, governos de todo o mundo estão dando incentivos e adotando políticas e regulamentações favoráveis às energias renováveis e os avanços tecnológicos estão tornando cada vez mais eficientes e acessíveis os sistemas de geração de energia oriundos dessas fontes. Como pode ser visto na figura 4 e na figura 5, com base em dados da International Renewable Energy Agency (IRENA), a capacidade de geração de energia por fontes renováveis vem crescendo de forma acelerada nos últimos anos.

Figura 4 – Participação anual das fontes renováveis no aumento da capacidade de geração de energia

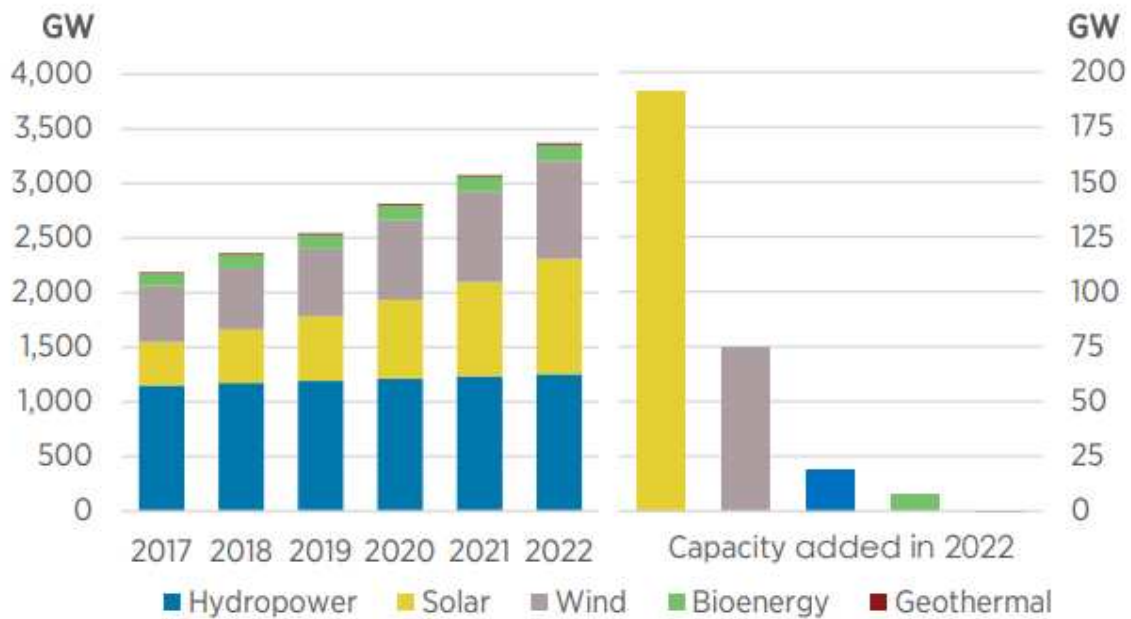


Fonte: (IRENA, 2023a).

A capacidade de geração de energia a partir de fontes renováveis cresceu 9.6% em 2022 em relação ao ano anterior, com destaque para fontes de energia solar e eólica, que representaram 90% deste aumento.

Neste cenário de franca expansão, de investimentos e de incentivos à implantação de sistemas de geração de energia a partir de fontes renováveis, o presente trabalho teve como motivação o desenvolvimento de uma ferramenta para auxiliar o desenvolvimento de projetos

Figura 5 – Crescimento da capacidade de geração de energia



Fonte: (IRENA, 2023a).

de geração de energia dessa natureza. Com isso, surgiu a ideia de desenvolver um protótipo de estação meteorológica, ferramenta fundamental para o desenvolvimento desses tipos de sistemas, principalmente sistemas de energia solar e de energia eólica, uma vez que o monitoramento dos agentes meteorológicos é de vital importância para a análise da viabilidade do projeto, para a realização de um dimensionamento correto, para garantir a eficiência e para monitorar a geração de energia.

O projeto foi desenvolvido tendo em vista o objetivo de ser de baixo custo, porém capaz de fornecer dados confiáveis, capaz de armazenar os dados coletado e capaz de compartilhar remotamente os dados coletados de forma fácil e eficiente, além de exibir de uma forma amigável uma visão geral do que está sendo coletado pela estação para o usuário. Dessa forma, foi utilizada como uma das centrais controladoras da estação uma placa Arduino, largamente utilizada em projetos de prototipagem, que possui custo acessível e um grande leque de módulos compatíveis com o seu sistema.

Também alinhado com o objetivo do projeto, o protótipo da estação foi desenvolvido utilizando o conceito de Internet das coisas (IoT), que vem ganhando destaque no mundo nos últimos anos, trazendo oportunidades e inovações nas mais diversas áreas. Por meio da IoT, é possível desenvolver dispositivos capazes de compartilhar dados com outros dispositivos e com outros sistemas através da internet, recurso de extrema utilidade para o armazenamento e compartilhamento dos dados coletados de forma remota, além de tornar possível a exibição

destes dados para o usuário final em um formato intuitivo e de fácil acesso em qualquer lugar que possua acesso a internet. Para tornar o protótipo um dispositivo inteligente, foi escolhida uma placa Esp para ser a segunda central controladora da estação, também de baixo custo, que possui capacidade de realizar conexão Wireless Fidelity (Wi-Fi) e, assim, permite que a estação troque dados com outros dispositivos inteligentes e com sistemas de computação em nuvem através da internet.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão abordados os conceitos fundamentais que definem as ferramentas utilizadas no projeto e que nortearam o planejamento e o desenvolvimento do protótipo da estação meteorológica.

2.1 Estação Meteorológica

Uma estação meteorológica é um conjunto de ferramentas capazes de realizar o monitoramento de grandezas meteorológicas, utilizada para fins de coleta de dados sobre a meteorologia de uma determinada região. Segundo o Instituto Nacional de Meteorologia (INMET) (2023), temos que:

A observação meteorológica é uma avaliação ou uma medida de um ou vários parâmetros meteorológicos. As observações são sensoriais quando são adquiridas por um observador sem ajuda de instrumentos de medição, e instrumentais, em geral chamadas medições meteorológicas, quando são realizadas com instrumentos meteorológicos. Portanto, os instrumentos meteorológicos são equipamentos utilizados para adquirir dados meteorológicos (termômetro/temperatura do ar, pressão atmosférica/barômetro, higrômetro/umidade relativa do ar etc.). A reunião desses instrumentos em um mesmo local, é denominada estação meteorológica.

A estação meteorológica do presente trabalho será capaz de aferir as grandezas de velocidade do vento, temperatura, umidade do ar e radiação solar global. Para isso, contará com os seguintes sensores: anemômetro, termômetro, higrômetro e piranômetro.

2.2 IoT

A IoT é um termo que se refere a sistemas e dispositivos físicos interligados a sensores, softwares ou outros tipos de tecnologia, capazes de se conectar e trocar dados com outros sistemas e dispositivos através da internet. A estes dispositivos capazes de se comunicar por meio da internet, dá-se o nome de dispositivos inteligentes.

Os avanços tecnológicos, que promoveram maior acessibilidade a microcontroladores, desenvolvimento de sensores de baixo custo, popularização de conectividade de alta velocidade e aumento da capacidade de processamento de dados em nuvem, fez com que a IoT experimentasse um crescimento exponencial nos últimos anos, tornando esta tecnologia uma das mais importantes do século XXI. Com sua enorme capacidade de coleta de dados, os dispositivos inteligentes auxiliarão fortemente processos de tomadas de decisão e processos relacionados à

automação de tarefas, representando um grande potencial para aumentar significativamente a produtividade, a eficiência e a economia na utilização de recursos em todas as áreas. Com isso, é esperado que, nos próximos anos, eles desempenhem um papel cada vez mais importante tanto no cotidiano das pessoas, quanto na operação de empresas e de indústrias.

2.3 Sensores

Um sensor é um dispositivo que irá reagir de alguma forma com algum fenômeno externo resultando em alguma forma de sinal mensurável para o ser humano. Segundo (FUENTES, 2005), um sensor pode ser definido como:

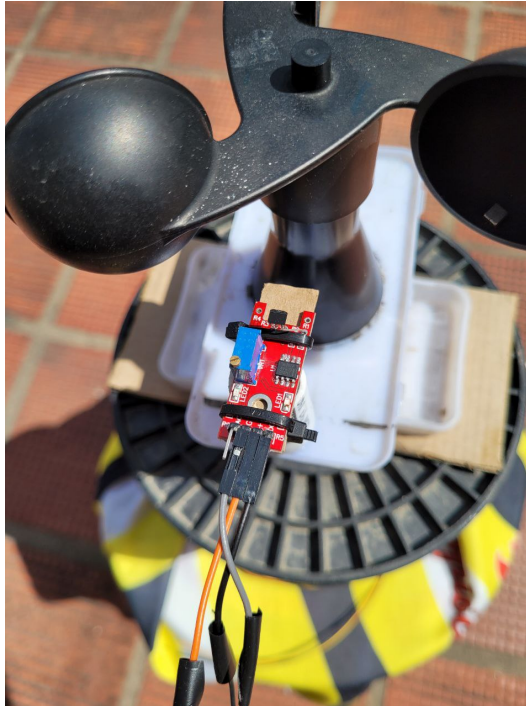
Dispositivo que converte uma grandeza física de qualquer espécie em outro sinal que possa ser transmitido a um elemento indicador, para que este mostre o valor da grandeza que está sendo medida ou que seja inteligível para o elemento de comparação de um sistema de controle.”.

A seguir, serão apresentados os sensores que realizarão o monitoramento dos quatro agentes meteorológicos presentes no protótipo desenvolvido neste trabalho.

2.3.1 *Anemômetro*

Um anemômetro é utilizado para se aferir a velocidade do vento. No presente trabalho, o anemômetro utilizado será uma junção de um sensor magnético de efeito hall, comprado pelo autor do projeto; um conjunto de ímãs, retirados de um leitor de cd; e uma carcaça de um anemômetro defeituoso, disponibilizado pelo Laboratório de Filmes Finos e Energias Renováveis (LAFFER) da Universidade Federal do Ceará (UFC). A carcaça utilizada conta com 3 pás conectadas a um eixo livre para girar. Na extremidade dessas pás, existem coletores de vento que facilitam o movimento de giro do eixo quando neles incidir vento. Anexados a esses coletores, estão os ímãs, que, ao passarem sobre o sensor hall (apresentado na figura 6) acoplado ao sistema durante o movimento circular causado pela ação do vento, farão com que o sensor emita um sinal que será coletado pela placa gerenciadora do sistema e que será utilizado para calcular as Rotações por minuto (RPM) do anemômetro e, a partir do RPM, calcular a velocidade do vento. Para sustentar o sensor hall, uma estrutura plástica foi adicionada ao sistema, além de mais uma estrutura plástica acoplada à base do anemômetro para melhor fixação da ferramenta durante os testes experimentais. A estrutura completa do anemômetro pode ser observada na figura 7.

Figura 6 – Sensor de efeito Hall



Fonte: Próprio autor

Figura 7 – Estrutura completa do anemômetro, com o sensor, carcaça e ímãs



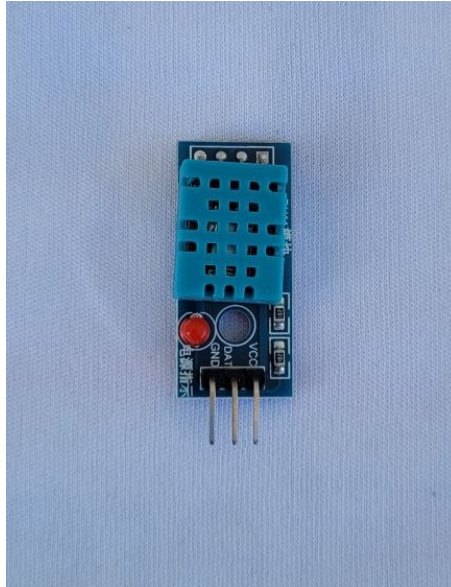
Fonte: Próprio autor

2.3.2 *Termômetro e Higrômetro*

O termômetro é uma ferramenta utilizada para se aferir a temperatura do local no qual o instrumento está localizado. Já o higrômetro é uma ferramenta utilizada para se aferir a

umidade do ar do local no qual o instrumento está localizado. No presente trabalho, o sensor DHT11 (apresentado na figura 9) será o responsável por realizar a aferição de ambas as grandezas. Este sensor foi comprado pelo autor do projeto.

Figura 8 – Sensor DHT11



Fonte: Próprio autor

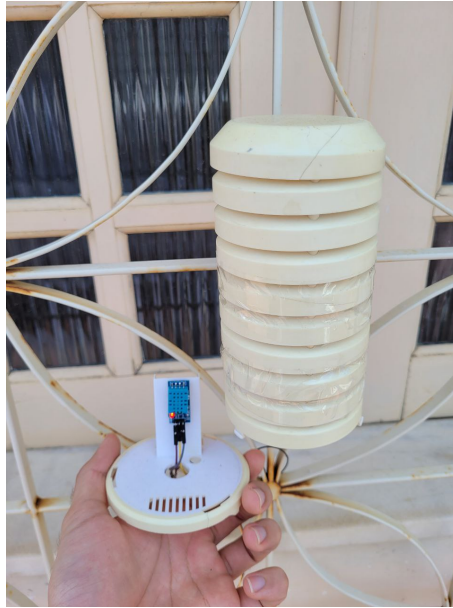
O sensor DHT11 é um dispositivo com capacidade de interação com diversas placas microcontroladoras disponíveis no mercado, incluindo a placa Arduino Mega 2560, escolhida para ser utilizada no projeto. De acordo com dados fornecidos pela fabricante do dispositivo, seu intervalo de aferição para temperaturas vai de 0 até 50°C, com precisão que vai de -2°C até +2°C. Já seu intervalo de aferição para a umidade vai de 20% até 90%, com precisão que vai de -5% até +5%.

Para completar a estrutura do termômetro/higrômetro, o sensor DHT11 ficará abrigado em uma estrutura que o protegerá do meio externo e garantirá que as aferições do sensor não sejam prejudicadas por possíveis interferências externas, além de facilitar a sua fixação nos mais diversos lugares. Essa estrutura pode ser observada na figura ??.

2.3.3 Piranômetro

Um piranômetro é uma ferramenta utilizada para se aferir a quantidade de radiação solar que incide sobre uma superfície durante um determinado período de tempo. No presente trabalho, o piranômetro utilizado foi desenvolvido pelo autor, sendo projetado a partir da relação entre a corrente produzida por uma placa solar exposta ao sol (apresentada na figura 10) ao longo

Figura 9 – Estrutura que abriga o sensor DHT11



Fonte: Próprio autor

do dia e valores de referência para a incidência de radiação solar esperada para o local de teste. A placa solar utilizada foi comprada pelo autor do projeto. Para realizar a aferição da corrente gerada pela placa solar, foi utilizado o sensor INA219, também comprado pelo autor do projeto, conectado junto a placa solar.

Figura 10 – Placa solar utilizada como sensor do piranômetro



Fonte: Próprio autor

2.4 Placas Microcontroladoras

Um microcontrolador é um circuito integrado que reúne uma Unidade central de processamento (*Central Processing Unit*) (CPU), memórias, tanto voláteis quanto não voláteis, e entradas e saídas que permitem conexões com periféricos para troca de dados. Já as placas microcontroladoras são placas de circuito que possuem um microcontrolador acoplados ao seu circuito, além de inúmeros outros componentes eletrônicos que tem como finalidade facilitar a interação entre o usuário e o microcontrolador, facilitando, assim, o desenvolvimento de projetos. No presente projeto, as placas microcontroladoras escolhidas foram as placas Arduino mega 2560 e a ESP 01.

2.4.1 Arduino

O Arduino é uma plataforma de prototipagem eletrônica desenvolvida em 2005, na Itália, pelos pesquisadores Massimo Banzi, David Cuartielles, Tom Igoe e David Mellis. Esta plataforma foi desenvolvida com o objetivo de se criar um dispositivo eletrônico de baixo custo que permitisse que estudantes, profissionais, entusiastas e pessoas interessadas em geral pudessem desenvolver desde projetos eletrônicos simples até projetos eletrônicos complexos de forma mais barata, mais simples e mais acessível. “O Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software de fácil utilização. É destinado a qualquer pessoa que esteja criando projetos interativos.” (ARDUINO, 2023).

A placa Arduino utilizada foi a placa Arduino mega 2560, disponibilizada pelo LAF-FER, sendo uma das escolhidas para o projeto devido à alta capacidade do seu microprocessador integrado.

De acordo com informações do fabricante, a placa Arduino mega 2560 possui integrada o microprocessador ATmega2560, que tem como grandes destaques sua maior capacidade de memória e maior disponibilidade de portas de entrada e saída de dados em relação a modelos mais básicos da placa, se tornando ideal para o presente projeto, pois, tanto permite que todos os recursos estipulados para o protótipo da estação sejam desenvolvidos, quanto permite futuras adições de recursos à estação.

Para desenvolver projetos utilizando plataformas Arduino, junto da placa também foram desenvolvidos seu próprio Ambiente de desenvolvimento integrado (IDE) e sua própria linguagem de programação, criando-se, assim, uma interface amigável de interação entre o

Figura 11 – Placa Arduino Mega 2560



Fonte: Próprio autor

usuário e a placa.

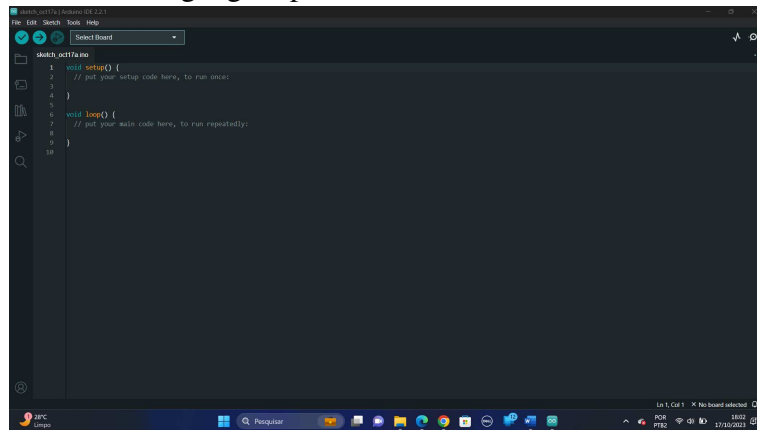
2.4.1.1 IDE e Linguagem de programação Arduino

A IDE do Arduino é o ambiente de desenvolvimento integrado da plataforma, sendo a ferramenta de software que fornece a interface que permite a escrita, a compilação e o envio dos códigos de programação para a placa. É por ela que todos os comandos a serem executados por todo o sistema serão escritos, a partir da linguagem de programação padrão da plataforma. Sua tela inicial pode ser vista na figura 12. A linguagem de programação utilizada no desenvolvimento de projetos para o Arduino é específica da plataforma, baseada na linguagem C++ simplificada, o que permite uma curva de aprendizagem mais rápida mesmo para aqueles com pouco conhecimento sobre programação.

2.4.2 ESP

Já a segunda placa microcontroladora escolhida para a estação foi a placa ESP 01, apresentada na figura 13. Esta placa foi comprada pelo autor do trabalho. Produzida pela empresa Espressif Systems, destaque no segmento de microcontroladores para sistemas IoT, ela possui integrada ao seu circuito o microprocessador ESP8266, que tem como diferencial a capacidade

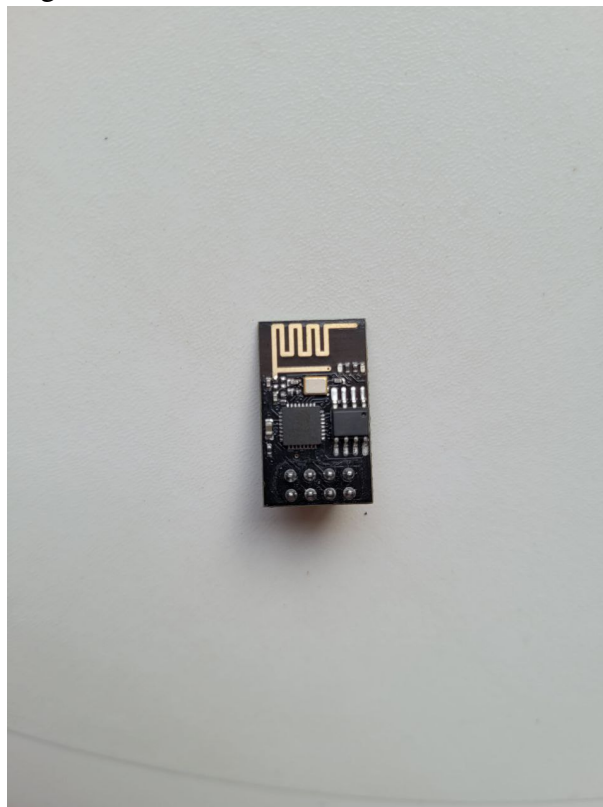
Figura 12 – IDE da Plataforma Arduino e exemplo de código escrito na linguagem padrão



Fonte: Próprio autor

de se conectar a redes Wi-Fi. Esta foi a característica que fez com que esta placa fosse escolhida para fazer parte do projeto. A sua capacidade de se conectar a redes Wi-Fi é o que torna o protótipo da estação um dispositivo alinhado com o conceito de IoT, elevando o projeto ao nível de dispositivo inteligente. Isso é o que permite que o protótipo seja capaz de se trocar dados tanto com sistemas de nuvem, quanto com quaisquer outros dispositivos inteligentes.

Figura 13 – Placa ESP 01



Fonte: Próprio autor

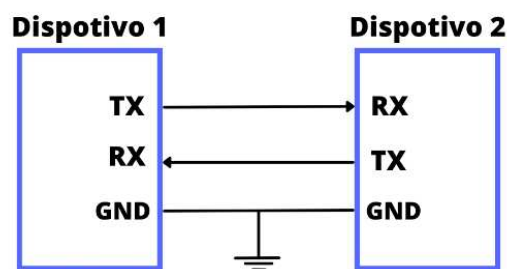
Apesar de possuir IDE e linguagem de programação próprias, ela também é compa-

tível com a IDE e com a linguagem de programação das plataformas Arduino, facilitando sua utilização no desenvolvimento de projetos.

2.5 Comunicação Serial

Para utilizar duas placas microcontroladoras no projeto, é necessário que haja uma comunicação entre elas para que cada uma realize sua devida função da forma correta e no momento correto. Para realizar esta comunicação, o método de transmissão de dados utilizado foi o método de comunicação serial, compatível com o hardware de ambas as placas do projeto. A comunicação serial consiste em um método que utiliza um único canal de comunicação para transmitir dados de um bit sequencialmente. Neste método de comunicação, o dispositivo que envia os dados, chamado de transmissor, converte os dados que se pretende enviar em uma sequência de bits. O dispositivo que recebe os dados, chamado de receptor, lê os bits que foram enviados pelo canal de comunicação e interpreta cada bit enviado sequencialmente, reconstruindo a informação enviada e a utilizando de acordo com o que o receptor está programado para fazer. O protocolo de comunicação serial utilizado pelas placas do projeto é o protocolo *Universal Asynchronous Receiver/Transmitter* (UART), cujo esquema de conexão entre os dispositivos está representado na figura 14. Sobre este protocolo, é importante que se saiba que se trata de um protocolo assíncrono e que requer que os dispositivos conectados estabeleçam a mesma velocidade de transmissão de dados para o sucesso da comunicação.

Figura 14 – Esquema de ligação entre dispositivos para realização de comunicação serial



Fonte: (HERO, 2023)

2.6 Sistema de computação em nuvem

Os sistemas de computação em nuvem, comumente chamados apenas de nuvem, são sistemas capazes de fornecer recursos de computação de forma remota, mantendo comunicação com o(s) sistema(s) que o utiliza(m) através apenas de conexão com a internet. Estes sistemas são capazes de fornecer servidores, bancos de dados para armazenamento, infraestrutura de rede, dentre outros recursos, apenas por meio de conexão com a internet. Sistemas de nuvem possuem diversos pontos positivos. Como principal vantagem, dentro do escopo do presente projeto, temos a capacidade de armazenar dados remotamente. A partir da possibilidade de armazenar os dados coletados remotamente, se torna possível o acompanhamento à distância em tempo real do que está sendo coletado pela estação, além de ser possível também acessar à distância os dados coletados, permitindo que estudos e análises sejam feitas sem a necessidade de deslocamento até o local de instalação da estação.

O sistema de computação em nuvem escolhido para ser integrado ao presente trabalho é mantido pela empresa AdaFruit, uma das grandes empresas do ramo da eletrônica e da IoT, que disponibiliza soluções de nuvem de alta confiabilidade, com versões gratuitas e pagas para quem precisa deste tipo de serviço para seus projetos. Para o presente projeto, as soluções fornecidas pelo plano gratuito da empresa já foram suficientes para a sua realização, sendo, portanto, o tipo de plano escolhido para o seu prosseguimento.

3 METODOLOGIA

Para a construção do protótipo da estação meteorológica, a metodologia empregada consistiu em criar um circuito que integrasse os sensores com as placas microcontroladoras responsáveis por gerenciar o funcionamento da estação e em desenvolver o sistema de operação a ser executado pelas placas microcontroladoras, responsável por definir como deve ser feito o gerenciamento da comunicação entre as duas placas, o gerenciamento da coleta dos dados, o gerenciamento do armazenamento local em um data logger e o gerenciamento da comunicação com o sistema de computação em nuvem escolhido para o armazenamento remoto e monitoramento em tempo real da estação.

3.1 Materiais

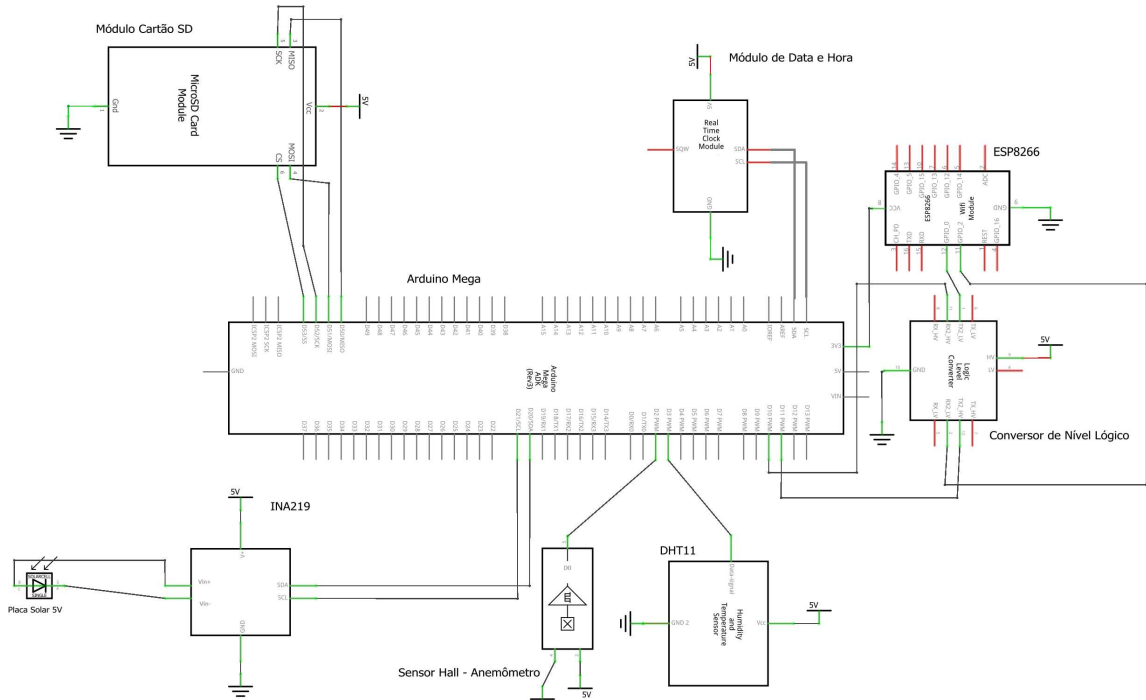
A seguir, serão listados todos os materiais utilizados na construção do protótipo da estação.

- a) Placa Arduino Mega 2560;
- b) placa ESP 01;
- c) protoboard;
- d) jumpers;
- e) fios;
- f) sensor de efeito hall;
- g) sensor DHT11;
- h) sensor INA219;
- i) módulo sd card;
- j) cartão microsd;
- k) placa solar 5V;
- l) conversor de nível lógico;
- m) carcaça para alojar o sensor DHT11;
- n) carcaça de um anemômetro;
- o) leds;
- p) fonte para protoboard;
- q) módulo rtc;

3.2 Diagrama esquemático do protótipo da estação

Na figura 15, é apresentado o diagrama esquemático do protótipo da estação, demonstrando detalhadamente como foram feitas todas as conexões entre os sensores e as placas microcontroladoras na construção do projeto.

Figura 15 – Diagrama de conexões do protótipo da estação meteorológica



fritzing

Fonte: Próprio autor

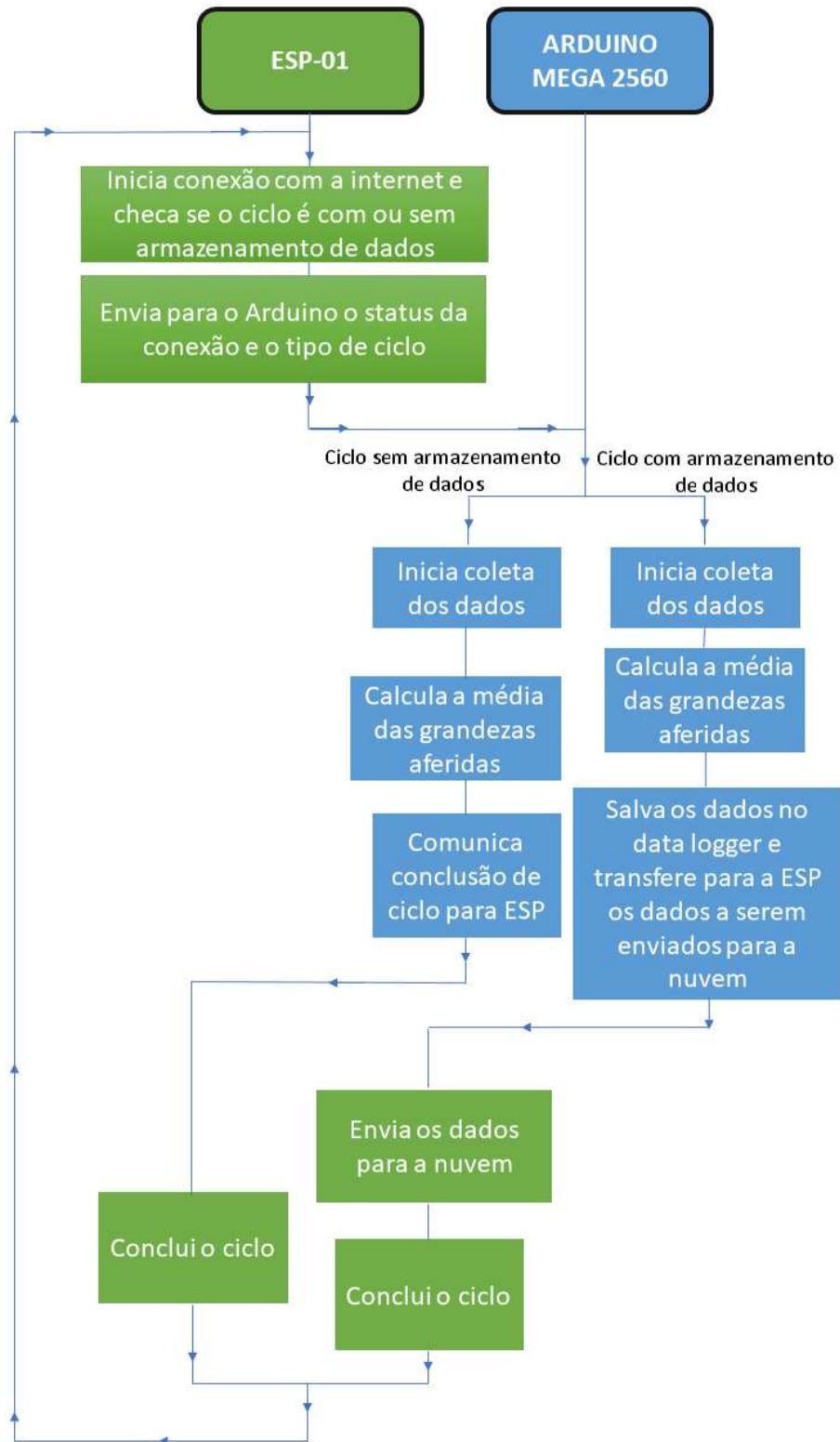
A partir deste diagrama, a montagem do circuito projetado para a estação pode ser realizada por qualquer pessoa interessada em trabalhar com o projeto.

3.3 Diagrama lógico do sistema do protótipo da estação

Na figura 16, é apresentado o diagrama lógico do sistema que opera o protótipo da estação desenvolvida neste projeto.

A partir deste esquema, é possível entender o funcionamento lógico do programa responsável por comandar as ações da estação. Também é possível entender como funcionam os ciclos de trabalho do protótipo. Da forma como foi configurado o sistema, a estação coleta continuamente dados dos sensores a ela conectados. Sempre que os dados são coletados, eles são utilizados para se calcular a média de cada grandeza aferida na estação naquele dia até o momento daquela coleta. Nos ciclos sem armazenamento, a estação coleta dados apenas para cálculo das

Figura 16 – Diagrama lógico do sistema do protótipo



Fonte: Próprio autor

médias. Nos ciclos com armazenamento, a estação continua realizando a coleta para calcular as médias das variáveis. Porém, após o cálculo das médias, estes valores são armazenados no

data logger e na nuvem. Nas configurações de teste definidas para o protótipo, para cada quatro ciclos de coleta sem armazenamento, a estação realiza um ciclo com armazenamento dos dados.

O sistema foi desenvolvido de forma que as coletas e o armazenamento de dados no data logger são gerenciados pela placa Arduino, que também fica responsável por gerenciar a transmissão dos dados coletados para a placa ESP 01. A placa ESP 01, por sua vez, fica responsável por realizar a comunicação da estação com o sistema de computação em nuvem que armazena remotamente os dados coletados e fornece uma interface gráfica amigável em formato de dashboard para o usuário poder realizar o acompanhamento remoto em tempo real do trabalho da estação.

3.4 Construção do anemômetro e o cálculo da velocidade do vento

Conforme já comentado, a construção do anemômetro do projeto foi realizada com o acoplamento de ímãs nas pás coletoras de vento conectadas ao eixo do anemômetro e com a fixação de um sensor de efeito hall instalado logo abaixo destas pás. O intuito da criação desta estrutura é fazer com que o sensor de efeito hall envie um sinal para a placa microcontroladora cada vez que um ímã das pás passe sobre ele. A partir da força do vento, as pás irão girar e, com os sinais emitidos pelo sensor de efeito hall, se torna possível realizar o cálculo do RPM do eixo do anemômetro e, com este valor, se torna possível aferir a velocidade do vento da localidade.

Para se calcular o RPM do anemômetro no momento das aferições de velocidade do vento, foi definido que o sistema coletaria a quantidade de giros do eixo por um intervalo de tempo de 5 segundos. Após os 5 segundos de coleta, tem-se que o sistema terá dado x voltas. Ao final do período, o sistema calculará o valor de RPM aferido durante o período de coleta de giros. Para realizar este cálculo, a seguinte fórmula foi implementada ao algoritmo do sistema:

$$RPM = \frac{TG}{t} \times 60 \quad (3.1)$$

, onde TG representa o total de giros do eixo durante o período da coleta, e pode ser calculado a partir da seguinte equação:

$$TG = \frac{S}{3} \quad (3.2)$$

, onde S é o total de sinais do sensor hall. Este cálculo é necessário, pois em uma volta completa

do eixo do anemômetro, o sensor é ativado 3 vezes. Isso fornece maior precisão à medição final de velocidade.

Com o valor de RPM calculado, temos a seguinte relação entre velocidade linear e velocidade angular, que nos permite calcular qual a velocidade do vento que passou pelo anemômetro:

$$V = \frac{2 \times \pi \times R \times RPM}{60} \times 3,6(km/h) \quad (3.3)$$

, onde R é o raio do anemômetro em metros.

A partir da implementação destes passos ao algoritmo que comanda o sistema da estação, temos a realização do cálculo da velocidade do vento.

3.5 Construção do piranômetro

Como já abordado anteriormente, o piranômetro deste projeto foi desenvolvido baseado na relação entre a corrente gerada por uma placa solar exposta ao sol, conectada a um sensor INA219 para medição da sua corrente, e valores de referência esperados para a incidência de radiação solar total no local em que o protótipo foi desenvolvido. Uma equação foi gerada a partir da relação entre estes valores de corrente produzida e radiação solar total esperada, e esta equação foi implementada ao algoritmo do sistema de controle da estação para realizar o cálculo da radiação solar total. Os dados de referência para a radiação solar total esperada foram obtidos do site de dados meteorológicos Tu Tiempo, que fornece estimativas sobre vários agentes meteorológicos para inúmeras cidades do mundo.

3.5.1 Coleta dos dados de corrente

Seguindo a metodologia de desenvolvimento do piranômetro, foram realizadas coletas de dados de corrente da placa solar por 5 dias: os dias 01/10/2023, 02/10/2023, 04/10/2023, 05/10/2023 e 06/10/2023. O ponto de coleta foi a casa do autor do projeto, localizada na cidade de Fortaleza-CE. As coletas foram realizadas entre os horários de 05:00 até as 18:00 horas, totalizando 13 horas de coleta diária. Os dados foram coletados com um intervalo de tempo de 1 minuto, totalizando 60 dados coletados por hora e 780 dados coletados por dia.

Aproveitando-se da necessidade de coleta e armazenamento desses dados, os sistemas de data logger e de nuvem da estação foram utilizados para realizar o armazenamento dos dados

coletados com o objetivo de já testar seus funcionamentos. Para armazenar os dados dos dias 1 e 2 de outubro de 2023, utilizou-se o sistema de data logger da estação. Já para armazenar os dados dos dias restantes, utilizou-se o sistema de nuvem. Na figura 17, podemos observar uma amostra de dados de corrente (mA) que foram salvos no sistema de nuvem, e na figura 18, podemos observar uma amostra de dados de corrente (mA) que foram salvos no data logger.

Figura 17 – Amostra de dados salvos na nuvem

Timestamp	Current (mA)	Other Data
2023/10/06 11:58:06AM	192.000000	0.0, 0.0, 0.0
2023/10/06 11:57:05AM	191.000000	0.0, 0.0, 0.0
2023/10/06 11:56:03AM	192.399994	0.0, 0.0, 0.0
2023/10/06 11:55:02AM	193.800003	0.0, 0.0, 0.0
2023/10/06 11:54:01AM	193.100006	0.0, 0.0, 0.0
2023/10/06 11:52:59AM	192.800003	0.0, 0.0, 0.0
2023/10/06 11:51:58AM	192.000000	0.0, 0.0, 0.0
2023/10/06 11:50:57AM	191.399994	0.0, 0.0, 0.0
2023/10/06 11:49:55AM	191.699997	0.0, 0.0, 0.0
2023/10/06 11:48:54AM	191.500000	0.0, 0.0, 0.0
2023/10/06 11:47:53AM	192.100006	0.0, 0.0, 0.0
2023/10/06 11:46:51AM	191.000000	0.0, 0.0, 0.0
2023/10/06 11:45:50AM	190.600006	0.0, 0.0, 0.0

Fonte: Próprio autor

Figura 18 – Amostra de dados salvos no data logger

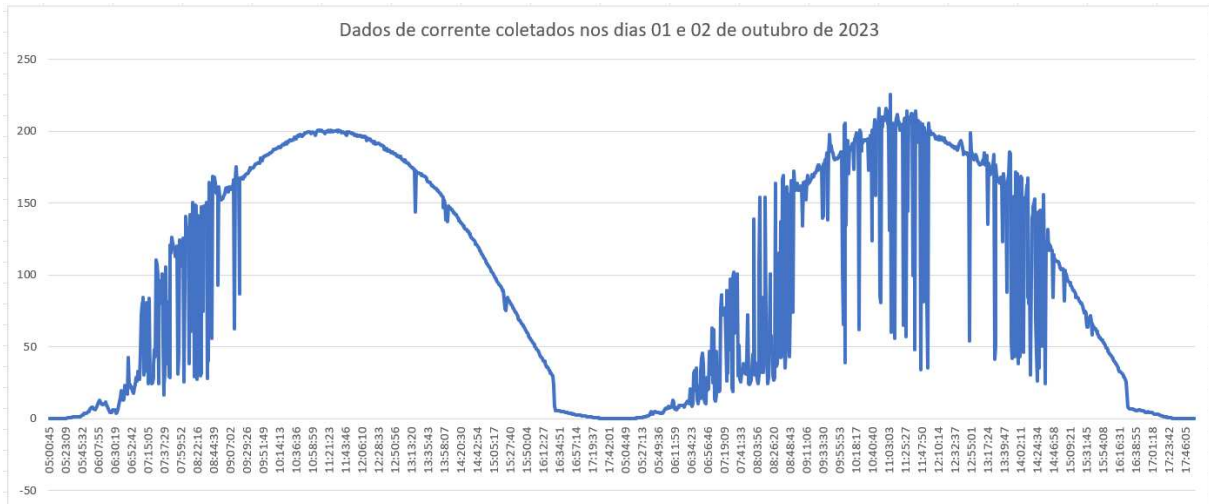
```

PIRAN
Arquivo Editar Exibir
755,01.10.2023,17:48:07,0,0
756,01.10.2023,17:49:08,0,0
757,01.10.2023,17:50:09,0,0
758,01.10.2023,17:51:10,-0,1
759,01.10.2023,17:52:11,0,0
760,01.10.2023,17:53:12,-0,1
761,01.10.2023,17:54:13,0,0
762,01.10.2023,17:55:14,-0,1
763,01.10.2023,17:56:15,-0,1
764,01.10.2023,17:57:16,-0,1
765,01.10.2023,17:58:18,-0,1
766,01.10.2023,17:59:19,-0,1
1,02.10.2023,05:00:45,0,0
2,02.10.2023,05:01:46,0,0
3,02.10.2023,05:02:47,0,0
4,02.10.2023,05:03:48,0,0
5,02.10.2023,05:04:49,0,0
6,02.10.2023,05:05:50,0,0
7,02.10.2023,05:06:52,0,0
8,02.10.2023,05:07:53,0,0
9,02.10.2023,05:08:54,0,0
10,02.10.2023,05:09:55,0,1
11,02.10.2023,05:10:56,0,1
12,02.10.2023,05:11:57,0,1
13,02.10.2023,05:12:58,0,1
14,02.10.2023,05:13:59,0,2
15,02.10.2023,05:15:00,0,2
16,02.10.2023,05:16:01,0,3
17,02.10.2023,05:17:02,0,3
18,02.10.2023,05:18:03,0,4
19,02.10.2023,05:19:04,0,4
20,02.10.2023,05:20:05,0,5
21,02.10.2023,05:21:06,0,5
22,02.10.2023,05:22:07,0,6
23,02.10.2023,05:23:08,0,8
24,02.10.2023,05:24:10,0,7
25,02.10.2023,05:25:11,0,8
26,02.10.2023,05:26:12,0,9
27,02.10.2023,05:27:13,1,0
28,02.10.2023,05:28:14,1,0
29,02.10.2023,05:29:15,1,2
Ln 1, Col 1
100% Windows (CRLF) UTF-8
    
```

Fonte: Próprio autor

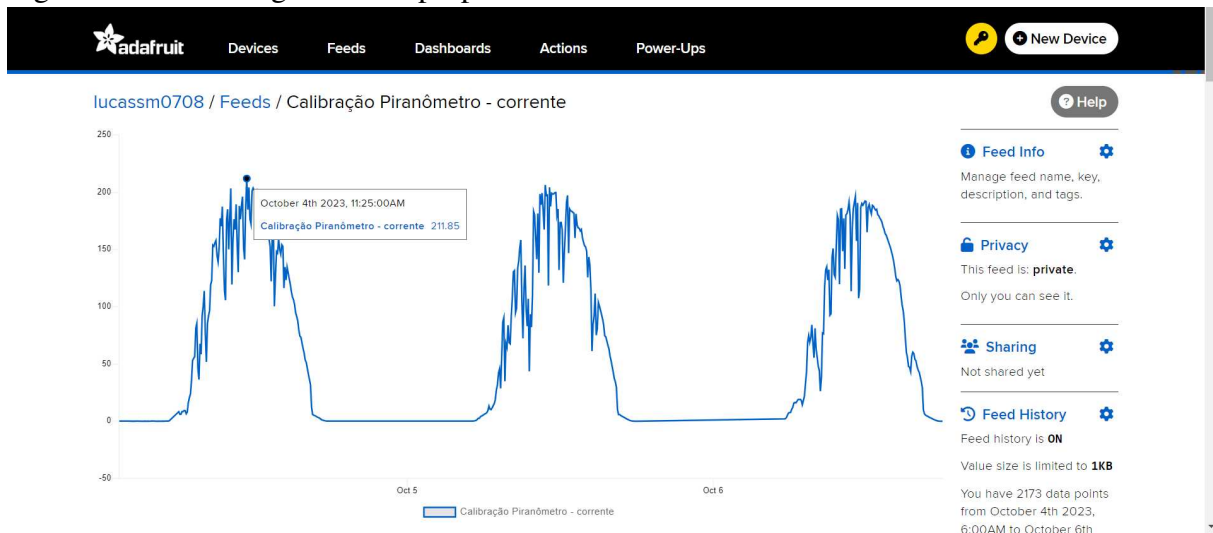
O desenho das curvas geradas pelos gráficos obtidos (apresentados nas figuras 19 e 20) a partir dos dados coletados tendem a formar uma parábola com concavidade voltada para

Figura 19 – Gráfico gerado no Excel com os dados de corrente salvos no data logger



Fonte: Próprio autor

Figura 20 – Gráfico gerados no próprio sistema de nuvem com os dados de corrente lá salvos



Fonte: Próprio autor

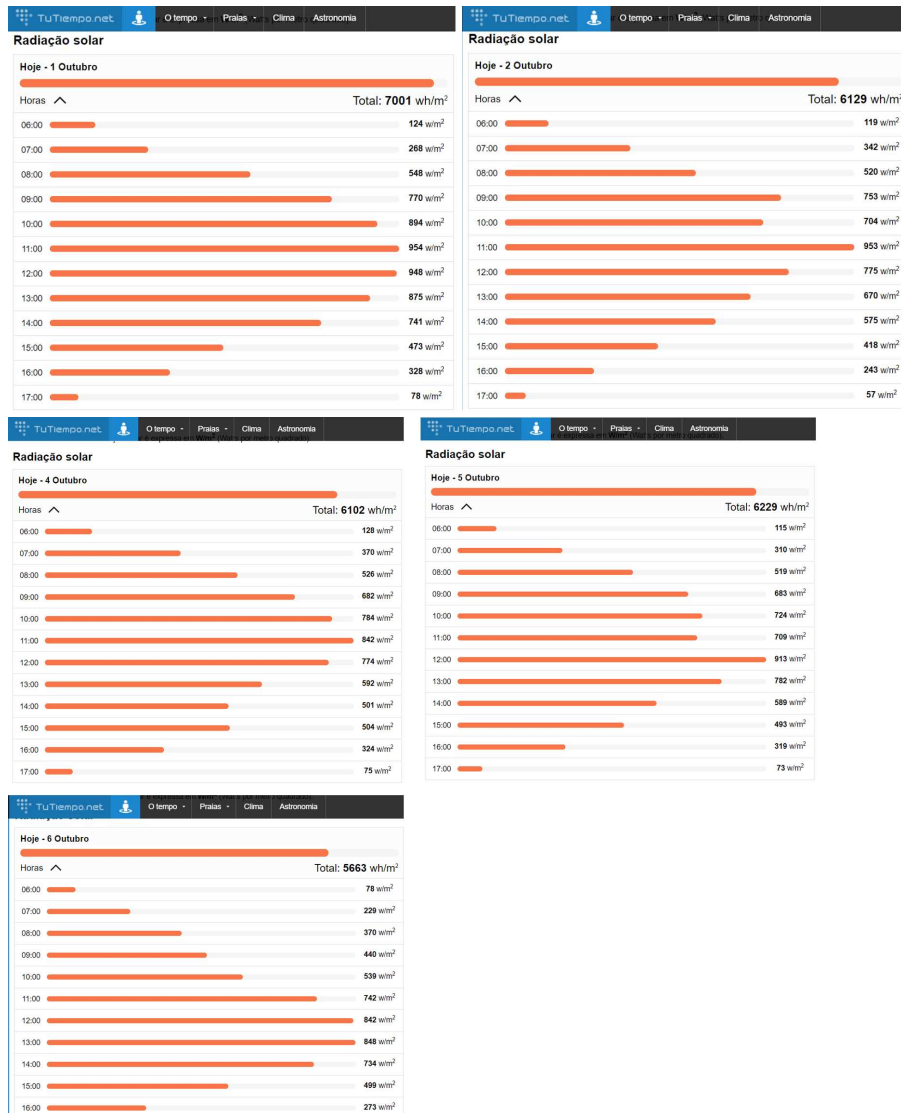
baixo, indicando que ele possui a mesma natureza que um gráfico esperado para a radiação solar global ao longo do dia. A parábola não é perfeita devido a fatores como nuvens, sombreamentos e fenômenos atmosféricos que, inevitavelmente, ocorrem ao longo do dia e interferem na incidência da radiação solar na superfície terrestre. Com isso, fica comprovado que o método de obtenção da radiação solar global é válido, sendo necessário apenas a realização da calibração do método.

3.5.2 Filtragem dos dados coletados de corrente

Os dados de referência utilizados (apresentados na figura 21), do site Tu Tiempo, fornecem estimativas de radiação solar global para cada intervalo de uma hora, de 06:00 as 17:00 horas. Com isso, a primeira filtragem feita nos dados coletados foi selecionar apenas os

dados coletados entre 06:00 e 17:59. A segunda filtragem foi realizada com base no tempo de exposição solar ao qual a placa foi exposta no local de coleta. Durante todo o período de sol dos dias em que as coletas foram realizadas, apenas a partir de algum horário entre 16 e 17 horas o local em que a placa solar estava instalada não incidia sol. Em uma observação presencial no dia 04/10/2023, notou-se que entre 16:20 e 16:30, o local ficava completamente na sombra pelo resto do dia, como pode ser visto na figura 22. Para confirmar esta observação, analisou-se os dados coletados entre 16:00 e 16:59 horas de todos os dias de coleta. A análise dos dados confirmou a observação, como pode ser visto nas figuras 23 e 24, com a queda brusca de geração de corrente observada entre 16:20 e 16:30.

Figura 21 – Dados de referência dos dias de coleta



Fonte: <https://pt.tutiempo.net/radiacao-solar/fortaleza.html>

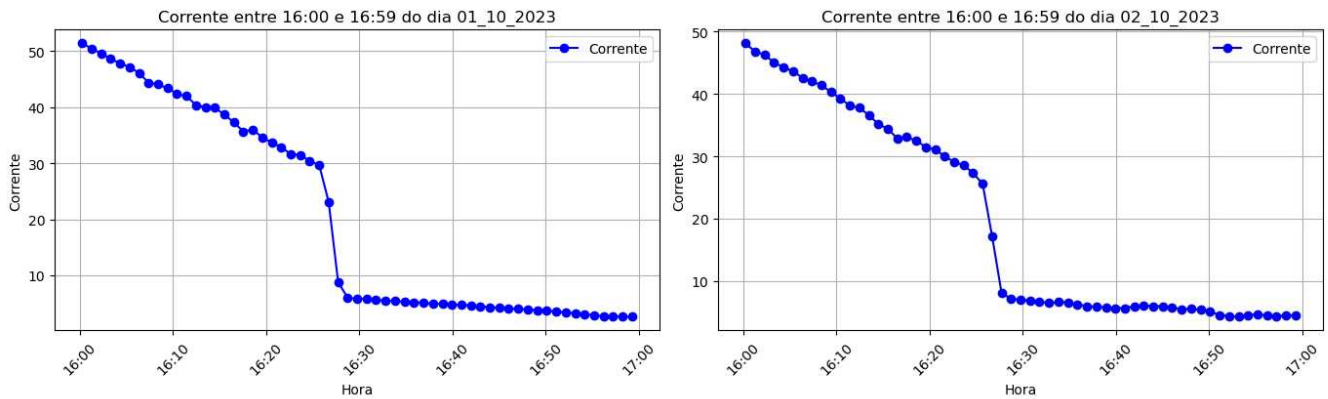
Com isso, foi decidido que os dados coletados a partir de 16:00 também seriam

Figura 22 – Local de coleta no dia 04/10/2023 - 16:20 e 16:27



Fonte: Próprio autor

Figura 23 – Gráficos de coleta de corrente entre 16:00 - 16:59 dos dias 01 e 02/10/2023



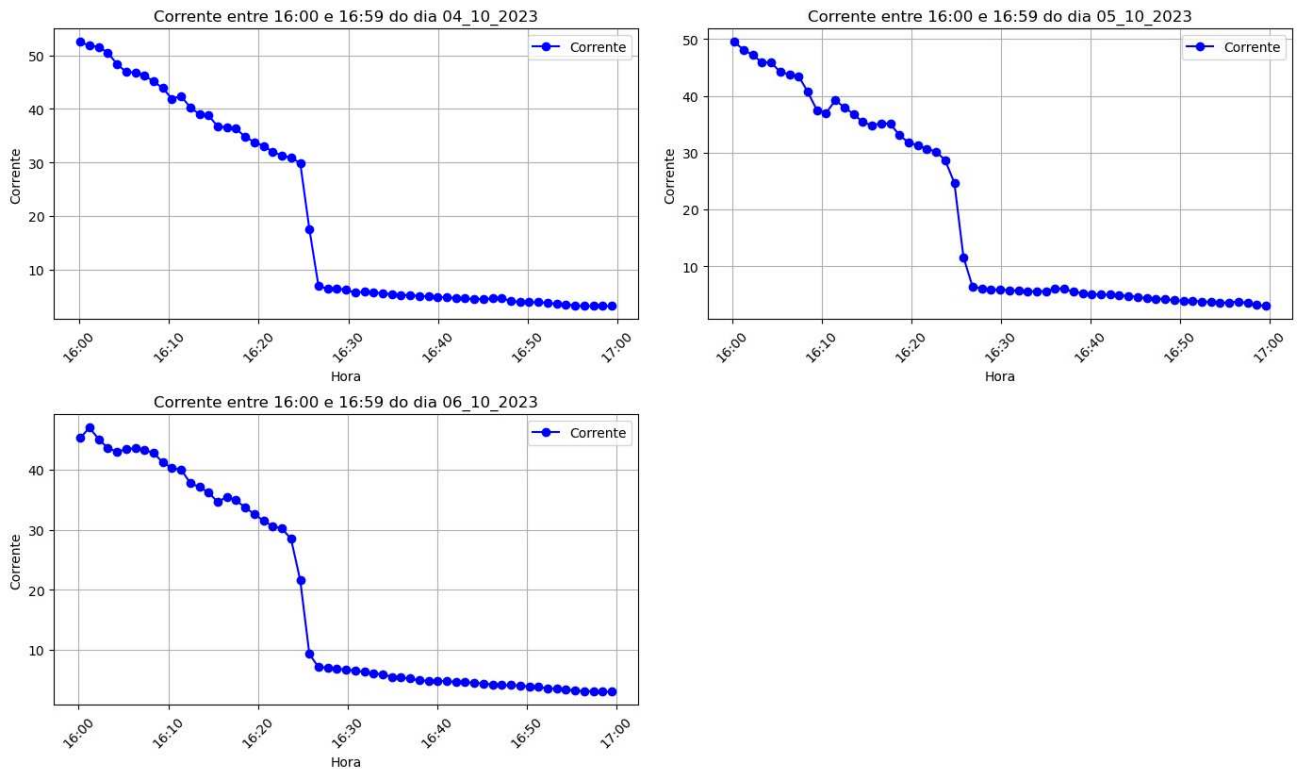
Fonte: Próprio autor

desconsiderados, uma vez que não representariam índices corretos de incidência de radiação solar.

3.6 Construção do termômetro e do higrômetro

Para a construção do termômetro e do higrômetro do protótipo deste projeto (apresentado na figura 25), apenas a anexação do sensor DHT11 à carcaça de proteção contra o meio externo e o desenvolvimento do algoritmo de programação da operação do protótipo foram

Figura 24 – Gráficos de coleta de corrente entre 16:00 - 16:59 dos dias 04, 05 e 06/10/2023



Fonte: Próprio autor

necessárias, uma vez que o próprio sensor é adquirido já com a capacidade de entregar os valores de temperatura e de umidade do ar do local sem a necessidade de conversões.

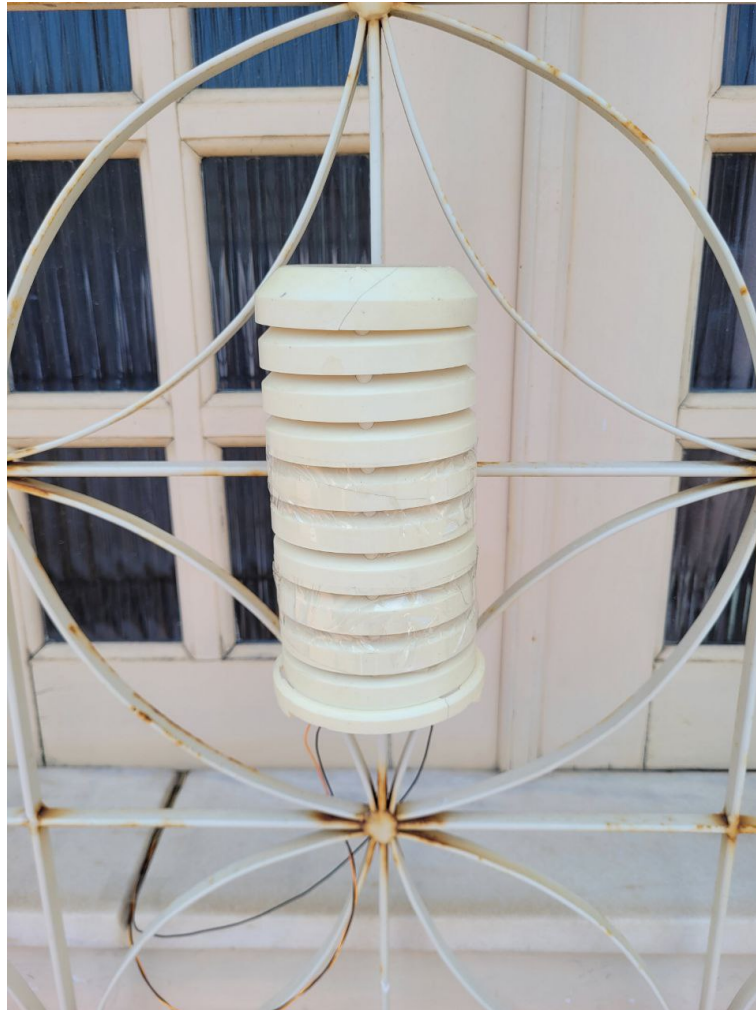
3.7 Construção do data logger

Para desenvolver o data logger, foram utilizados um módulo adaptador de leitura e escrita de dados em cartões microSD e um cartão microSD. Ambos foram concedidos pelo LAFFER. Na construção, foi necessária a criação do circuito para conectar as saídas do módulo sd às devidas entradas da placas microcontroladora Arduino, e o desenvolvimento do algoritmo de programação responsável por gerenciar o salvamento dos dados.

3.8 Sistema de computação em nuvem AdaFruit

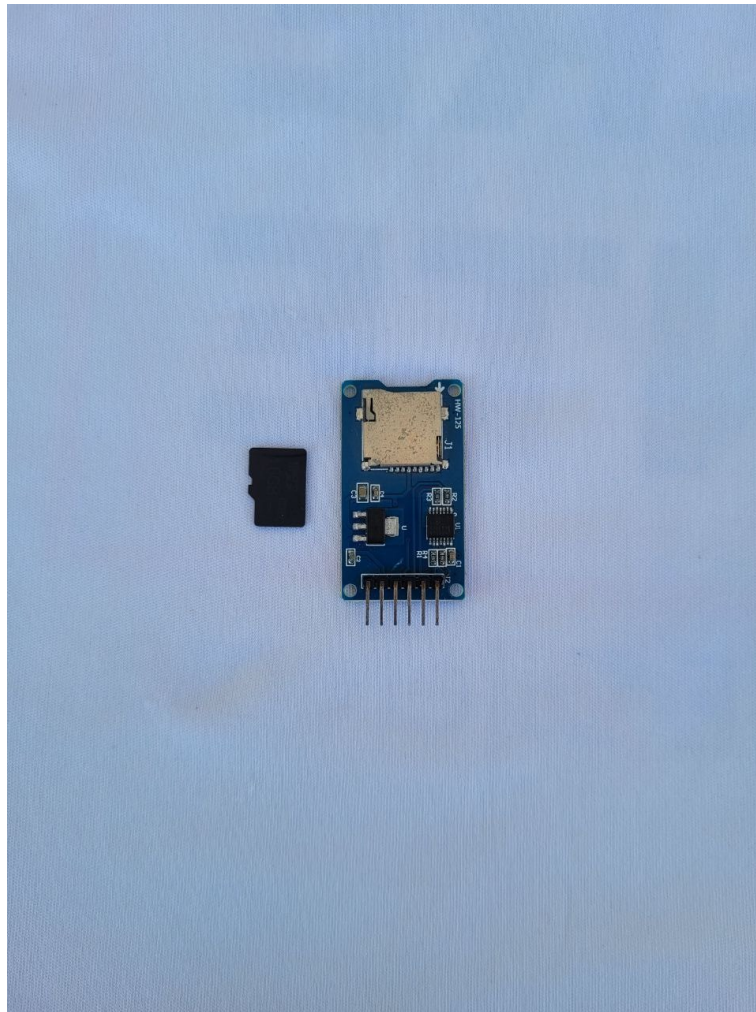
O sistema de computação em nuvem utilizado foi o disponibilizado em versão gratuita pela desenvolvedora AdaFruit. A comunicação entre o protótipo da estação e a nuvem é gerenciada pela placa ESP 01, que recebe os dados coletados pela placa Arduino e os envia para o sistema de nuvem ofertado pela empresa AdaFruit, por meio de conexão com a internet.

Figura 25 – Estrutura montada do termômetro/higrômetro



Fonte: Próprio autor

Figura 26 – Módulo de leitura e escrita de dados no cartão sd e cartão sd



Fonte: Próprio autor

Figura 27 – Sistema de nuvem AdaFruit

The screenshot displays the AdaFruit IO user interface for user 'lucasm0708'. At the top, there is a navigation bar with links for Shop, Learn, Blog, Forums, LIVE!, AdaBox, and IO. The user's name 'Hi, Lucas Meneses' and account options are visible on the right. Below the navigation bar, the user's profile 'lucasm0708 / Overview' is shown, along with a 'Help' button. A green notification box indicates the user is on a Basic plan and offers an upgrade to AIO+ for \$10/month. The main content area is divided into several sections: 'Account Status' with progress bars for Devices (0 of 2), Groups (2 of 5), Feeds (10 of 10), Dashboards (2 of 5), and Data Rate (0 of 30); 'Live Errors' showing 'No errors since page load.'; 'Live Data' showing 'No data since page load.'; and 'Connections' showing 'No connections.' A 'My Dashboards' section lists a dashboard named 'Estação Meteorológica'.

Fonte: Próprio autor

4 RESULTADOS E DISCUSSÕES

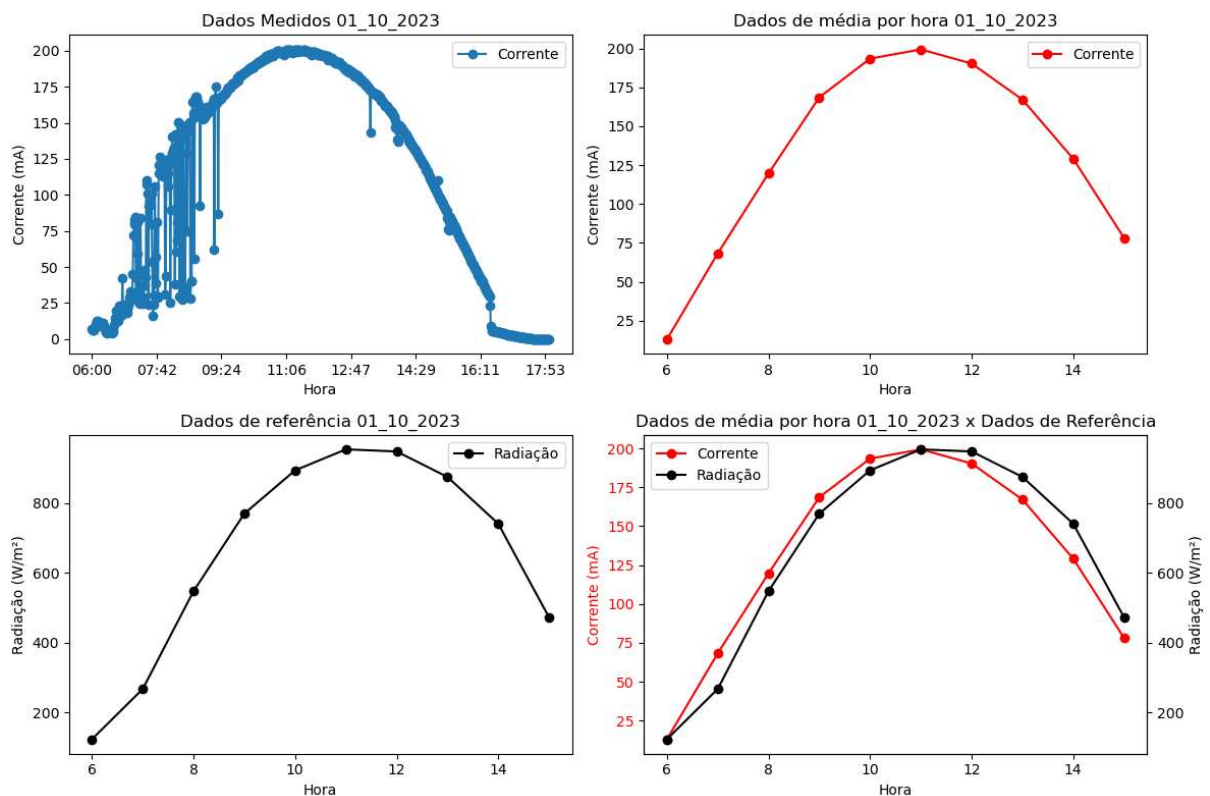
Nesta seção, discutiremos os resultados atingidos com o método de obtenção da equação de determinação da radiação solar global em função da corrente gerada pela placa solar, e os resultados gerais atingidos no presente trabalho.

4.0.1 Resultados da obtenção da equação para cálculo da radiação solar global

Após a filtragem dos dados de corrente coletados, sobraram aqueles obtidos entre 06:00 e 15:59 dos dias de coleta, e, conseqüentemente, foram utilizados apenas os valores dos dados de referência para o mesmo intervalo de tempo para comparação e obtenção da equação.

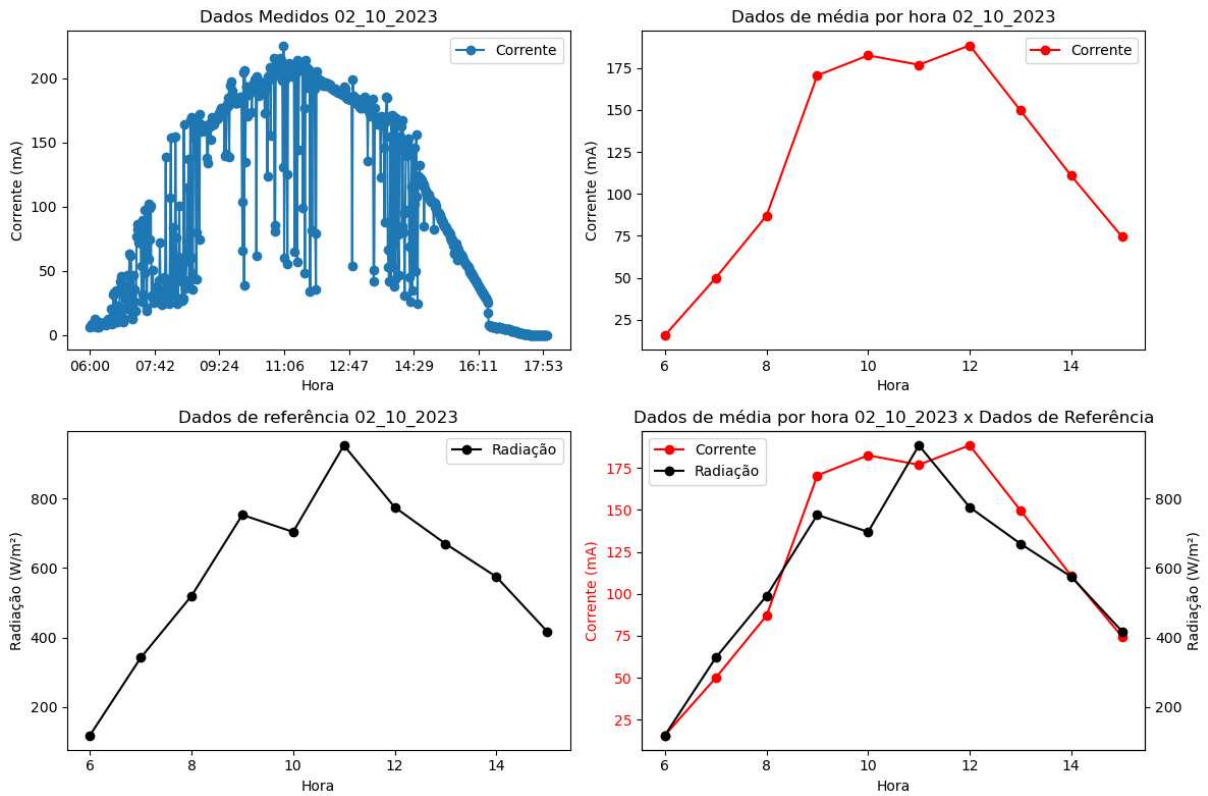
Como os dados de referência são fornecidos por estimativas de radiação solar global para cada intervalo de hora, foi calculada a média de corrente gerada por hora a partir dos dados coletados, como pode ser visto nas figuras 28, 29, 30, 31, 32. Após o cálculo das médias, foram gerados gráficos com os dados dessas médias e com os dados de referência para possibilitar a comparação de comportamento entre eles e, assim, encontrarmos em qual dia a previsão fornecida pelo site mais se aproximou do comportamento meteorológico do local de coleta.

Figura 28 – Comparação entre os dados coletados e os dados de referência do dia 01/10/2023



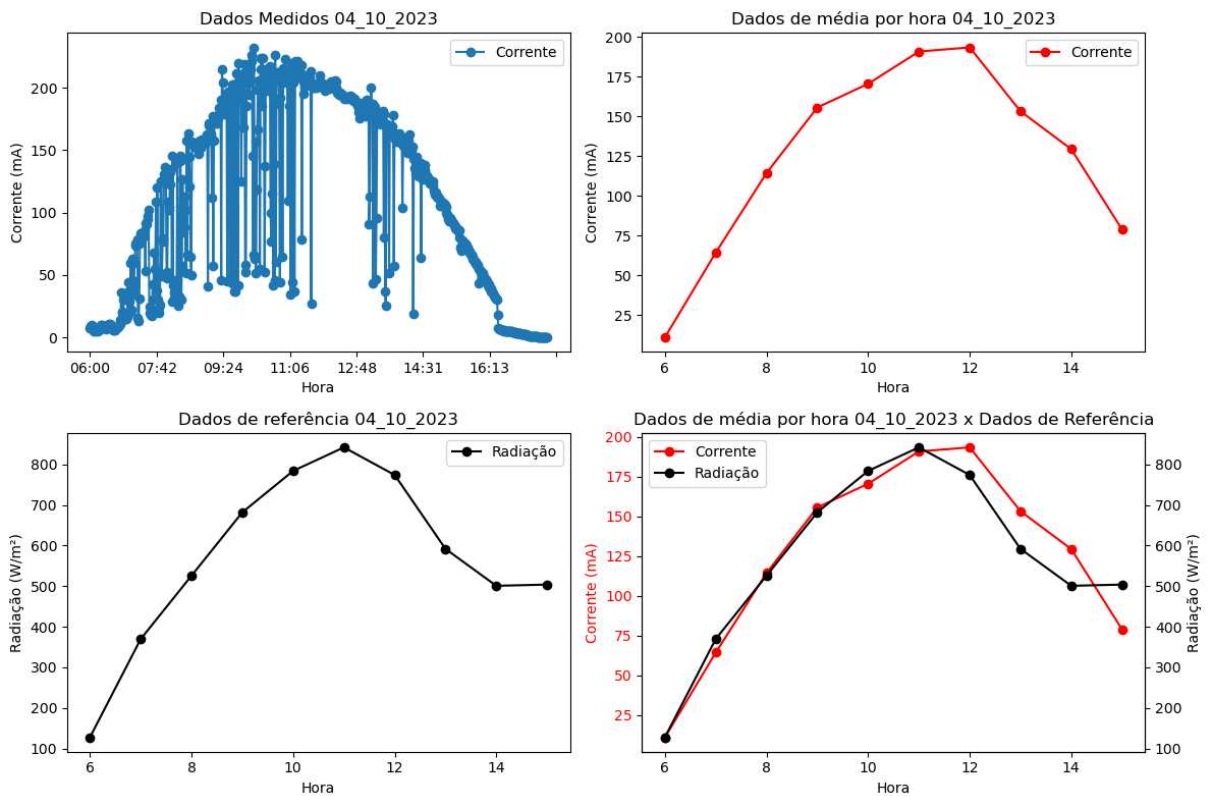
Fonte: Próprio autor

Figura 29 – Comparação entre os dados coletados e os dados de referência do dia 02/10/2023



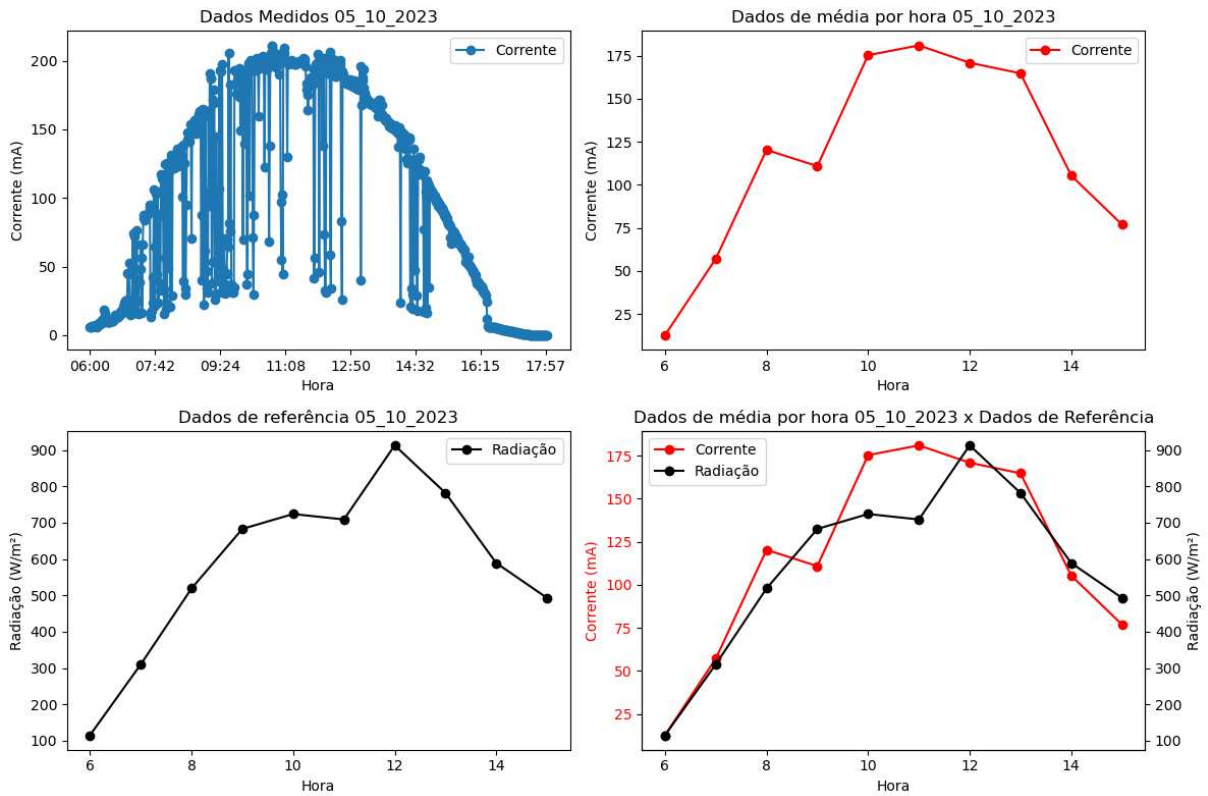
Fonte: Próprio autor

Figura 30 – Comparação entre os dados coletados e os dados de referência do dia 04/10/2023



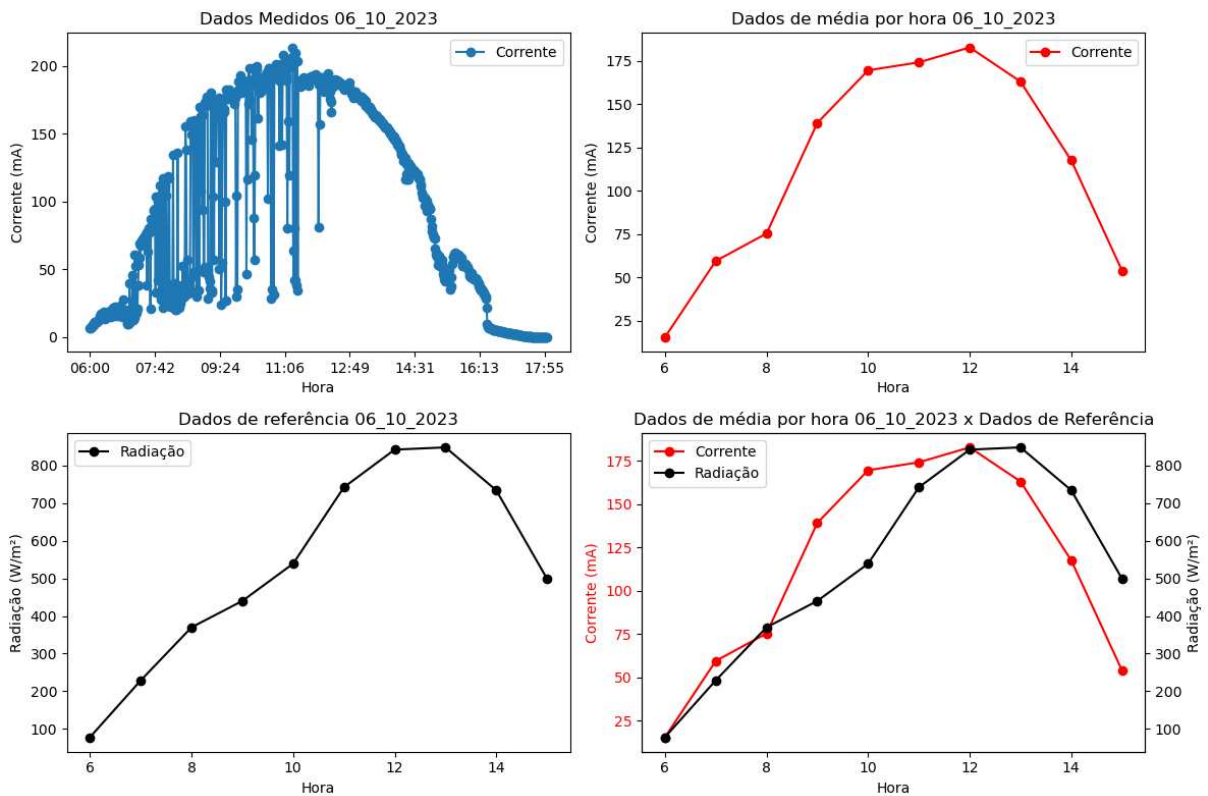
Fonte: Próprio autor

Figura 31 – Comparação entre os dados coletados e os dados de referência do dia 05/10/2023



Fonte: Próprio autor

Figura 32 – Comparação entre os dados coletados e os dados de referência do dia 06/10/2023

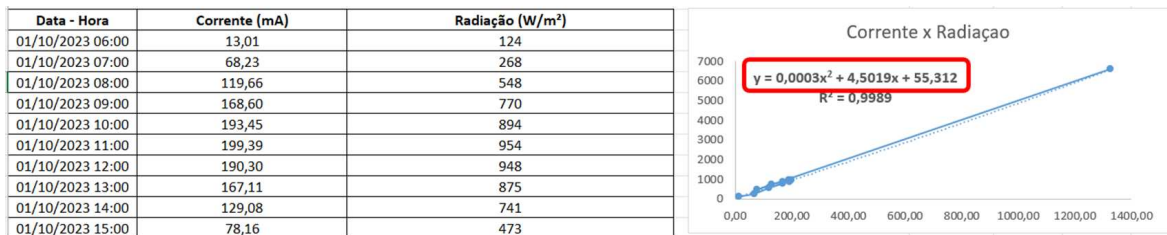


Fonte: Próprio autor

Como ficou constatado pela análise dos gráficos gerados, o primeiro dia de coletas foi o dia em que a previsão fornecida pelo site mais se aproximou das condições meteorológicas do local de coletas. Isso se deve, principalmente, pelo fato de este dia ter sido um dia de céu aberto, com poucas nuvens ao longo do dia e, conseqüentemente, menos interferência na incidência de radiação solar. Nos outros dias de coleta, como pode ser observado nos gráficos, tanto nos dados coletados quanto nos dados de referência existem irregularidades no formato da parábola esperada para uma curva de radiação solar global, o que indica que foram dias com mais fenômenos de interferência na incidência de radiação solar; e essas irregularidades, apesar de terem um padrão semelhante, ocorrem em momentos distintos nos gráficos gerados com os dados de referência e nos gerados com dados medidos.

Com isso, utilizando o software de planilhas eletrônicas Microsoft Excel e os dados coletados no primeiro dia de coletas (demonstrado na figura 33), a previsão de incidência de radiação solar global foi determinada a partir da seguinte equação:

Figura 33 – Obtenção da equação



Fonte: Próprio autor

$$rad = 0.0003 x i^2 + 4.5019 x i + 55.312(W/m^2) \quad (4.1)$$

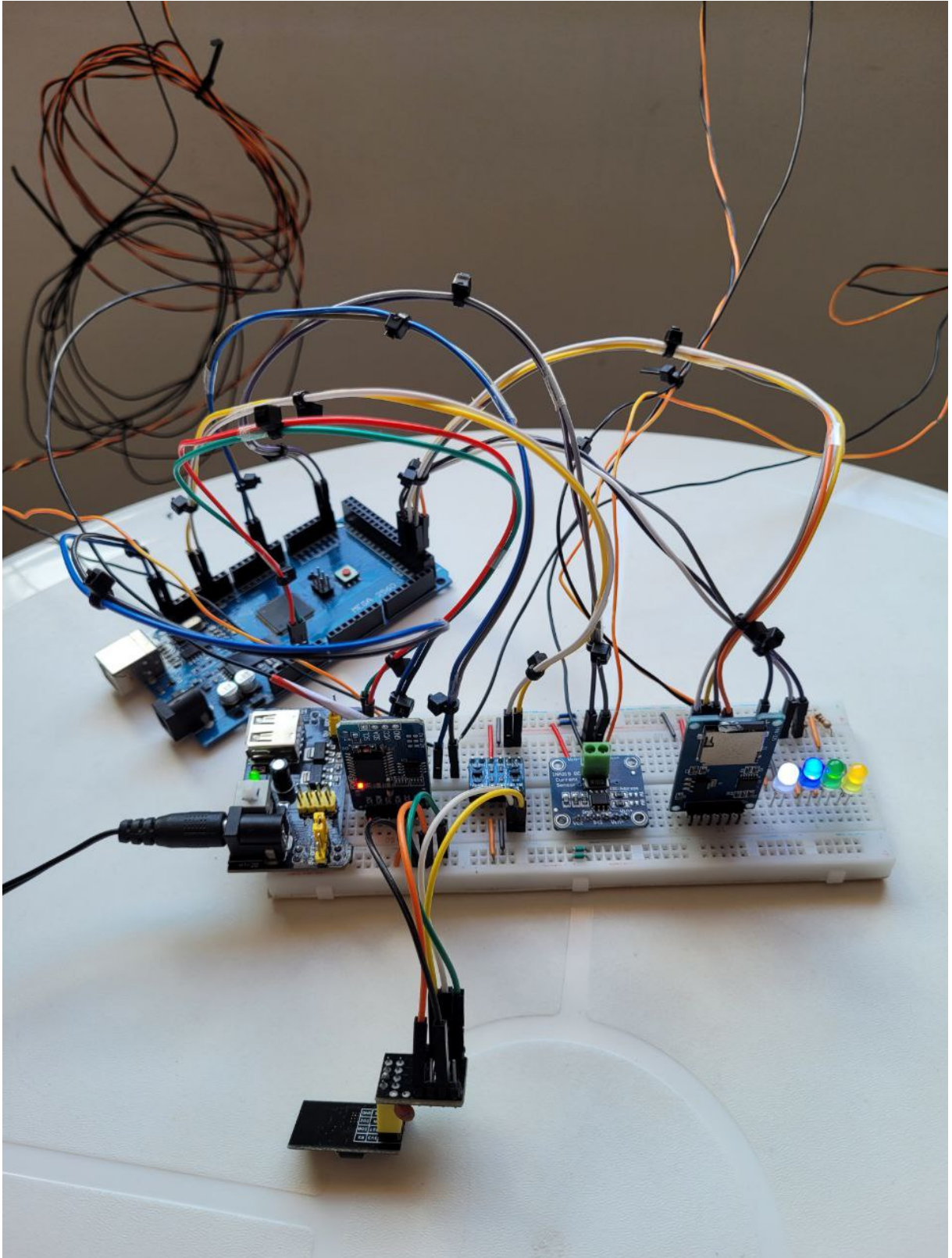
, onde i é o valor da corrente gerada pela placa solar.

Vale discutir que o modelo ideal de coleta de dados de referência para garantir a maior precisão possível da equação de cálculo da radiação solar total seria colocar um piranômetro no mesmo local em que está a placa solar utilizada para obtenção dos valores de corrente gerados. Uma vez que os valores obtidos pelo site mencionado são apenas estimativas gerais para a cidade pesquisada, a precisão da equação obtida ficará reduzida, pois, logicamente, o valor de referência generalizado não será preciso para todos os pontos da cidade, resultando nas diferenças de comportamento ao longo do dia observada nas figuras 29, 30, 31, 32.

4.0.2 Resultados finais do trabalho

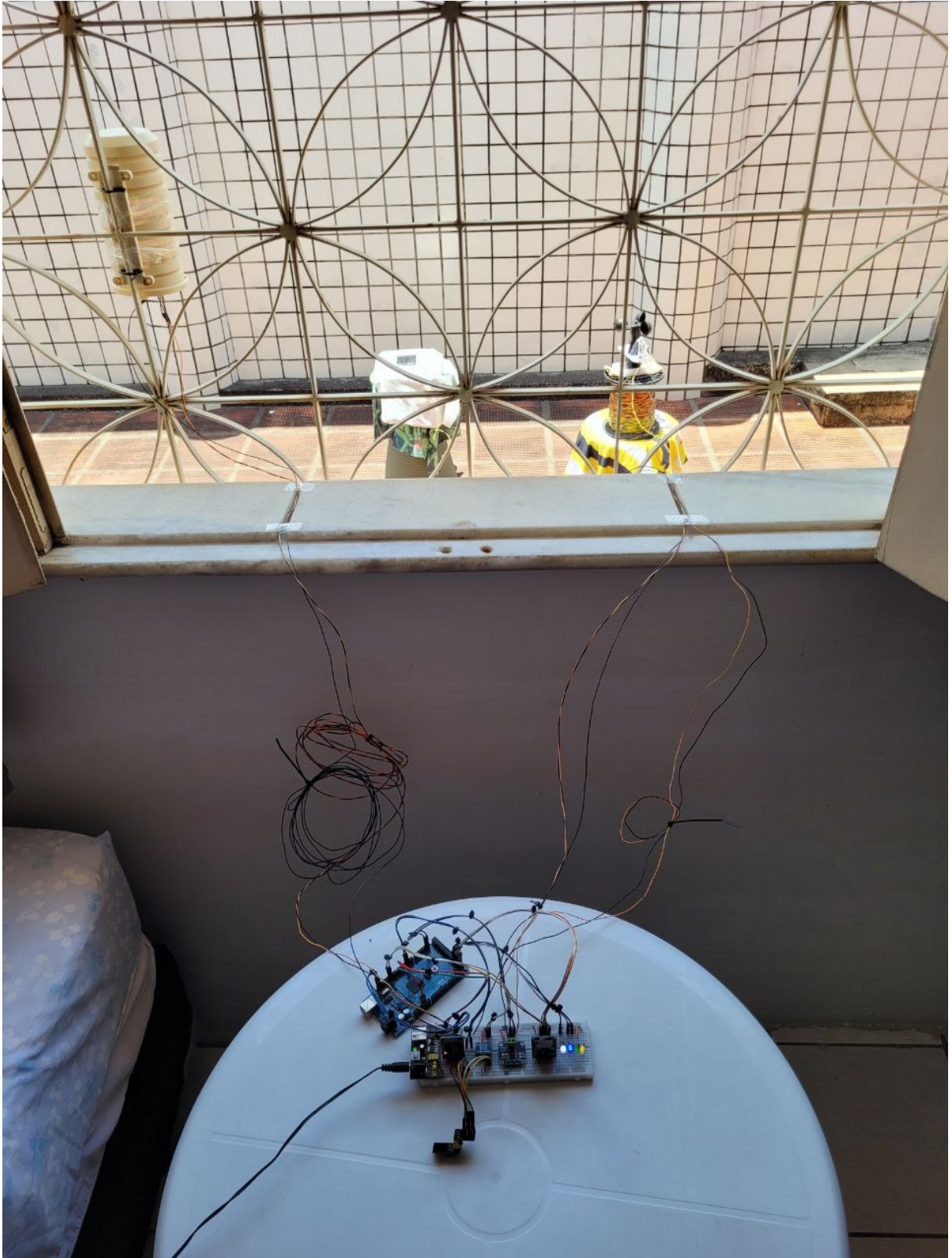
Após a montagem do circuito (demonstrado na figura 34) e instalação do sistema de operação do protótipo da estação em ambas as placas microcontroladoras, a estação foi inicializada no dia 11/10/2023 (demonstrado na figura 35). Do dia 11/10/2023 até o dia 14/10/2023, o protótipo ficou em funcionamento em caráter de teste, com alguns recursos sendo utilizados de cada vez para garantir que cada parte do sistema estaria funcionando de acordo com o esperado. Dados deste período estão armazenados no sistema de nuvem do projeto como forma de acompanhamento do funcionamento durante o período de testes. Durante o período de testes, verificou-se que os sistemas conectados ao protótipo estavam funcionando individualmente de acordo com o planejado. Com isso, a partir do dia 15/10/2023, o protótipo foi inicializado em caráter definitivo, com todos os seus sensores e sistemas atuando em conjunto para realização do teste final de funcionamento do protótipo. De 00:30 do dia 15/10/2023 até as 15:00 do dia 20/10/2023, o sistema ficou ligado ininterruptamente, e seu funcionamento foi um sucesso. Todos os sensores atuaram da forma que se esperava, os dados foram coletados com sucesso, o armazenamento no data logger ocorreu com sucesso (demonstrado na figura 36) e a comunicação com o sistema de nuvem também ocorreu com sucesso (demonstrado na figura 37). Também foi possível acompanhar remotamente o funcionamento da estação através da interface de dashboard (demonstrado na figura 38) desenvolvida pelo autor no sistema de nuvem.

Figura 34 – Circuito do protótipo



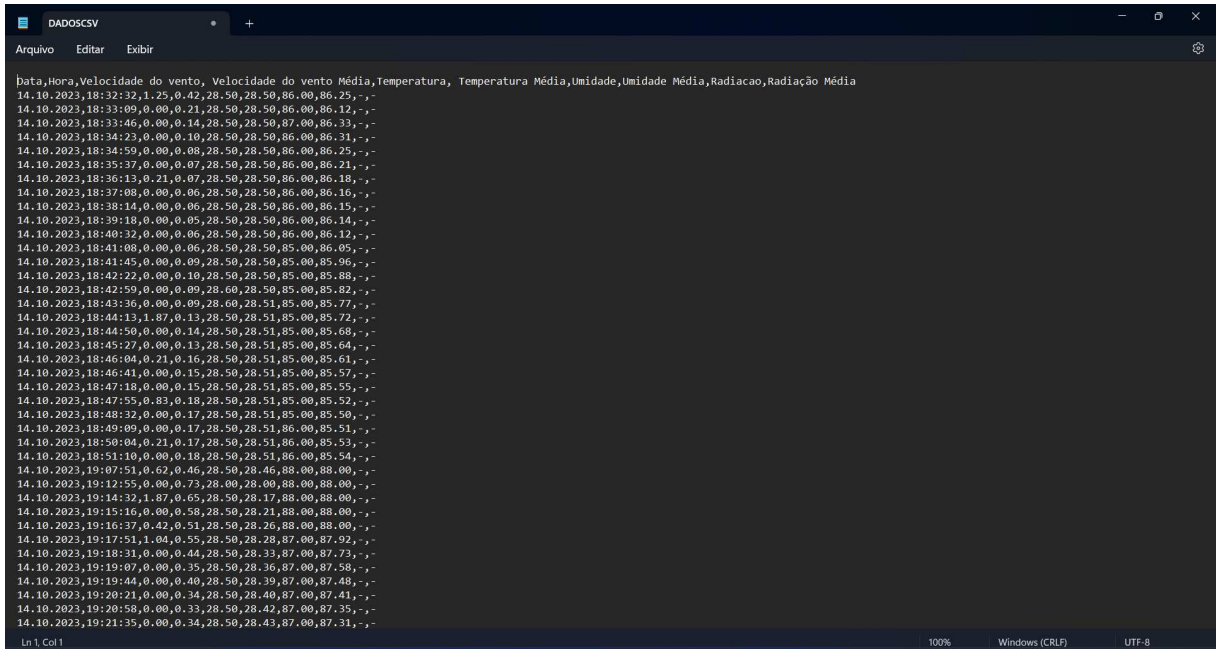
Fonte: Próprio autor

Figura 35 – Protótipo em funcionamento



Fonte: Próprio autor

Figura 36 – Data logger do sistema



Fonte: Próprio autor

Figura 37 – Banco de dados remoto do sistema

New Feed New Group

Default

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> Calibração Piranômetro - corrente	calibracao-piranometro-c...	0.000000	11 days ago
<input type="checkbox"/> Radiação Solar - Atual	radiacao-solar-atual	583.400024	about 4 hours ago
<input type="checkbox"/> Radiação Solar - Média	radiacao-solar-media	274.019989	about 4 hours ago
<input type="checkbox"/> Radiação Solar - Total	radiacao-solar-total	0	about 19 hours ago
<input type="checkbox"/> Temperatura - Atual	temperatura-atual	33.799999	about 4 hours ago
<input type="checkbox"/> Temperatura - Média	temperatura-media	29.730000	about 4 hours ago
<input type="checkbox"/> Umidade - Atual	umidade-atual	65.000000	about 4 hours ago
<input type="checkbox"/> Umidade - Média	umidade-media	82.959999	about 4 hours ago
<input type="checkbox"/> Velocidade Do Vento - Atual	velocidade-do-vento-atual	0.000000	about 4 hours ago
<input type="checkbox"/> Velocidade Do Vento - Média	velocidade-do-vento-med...	0.160000	about 4 hours ago

Fonte: Próprio autor

Figura 38 – Dashboard de monitoramento da estação



Fonte: Próprio autor

5 CONCLUSÕES E TRABALHOS FUTUROS

O projeto de desenvolvimento de um protótipo de estação meteorológica utilizando plataforma IoT foi realizado com sucesso. Todos os objetivos iniciais propostos foram atingidos, com a construção de um dispositivo inteligente, capaz de coletar dados a cerca de agentes meteorológicos de forma confiável, com custo reduzido, com capacidade de armazenar dados localmente e remotamente e com uma interface de monitoramento remoto em tempo real de fácil utilização por parte do usuário.

Como trabalhos futuros, é possível melhorar ainda mais o sistema de calibração dos sensores, trazendo ainda mais precisão para os dados coletados; construção de um corpo projetado especificamente para a estação em uma impressora 3d; desenvolvimento de uma placa de circuito impresso para ligação do circuito da estação; adição de sensores na estação para coleta de dados de mais agentes meteorológicos e aplicação de técnicas de *machine learning* para análise de dados para realização de previsões do tempo a partir dos dados coletados.

As possibilidades são enormes, e com o constante desenvolvimento e maior acessibilidade de tecnologias, o leque de opções para desenvolvimento de projetos a partir da presente estação ficará cada vez maior. A estação ficará em operação no LAFFER coletando dados meteorológicos do local, e ficará disponível para futuros trabalhos e aprimoramentos.

REFERÊNCIAS

ARDUINO. **About Arduino**. United States: Arduino, 2023. Disponível em: <https://www.arduino.cc/en/about>. Acesso em: 14 out. 2023.

FUENTES, R. C. **Apostila de Automação Industrial**. Santa Maria: Universidade Federal de Santa Maria, 2005.

GLOBO. **'Era da fevura global': gráficos mostram 'oceanos com febre', recordes de calor e gelo derretendo, tudo agora**. Brazil: Globo.com, 2023. Disponível em: <https://1nk.dev/YEZbQ>. Acesso em: 13 out. 2023.

HERO, M. **Comunicação serial Arduino via protocolo UART**. Brazil: Maker Hero, 2023. Disponível em: <https://www.makerhero.com/blog/comunicacao-serial-arduino-via-protocolo-uart/>. Acesso em: 14 out. 2023.

IEA. **CO2 Emissions in 2022**. France: International Energy Agency, 2023.

IRENA. **Renewable capacity statistics 2023**. Abu Dhabi: International Renewable Energy Agency, 2023.

IRENA. **Renewable energy statistics 2023**. Abu Dhabi: International Renewable Energy Agency, 2023.

METEOROLOGIA, I. N. de. **ESTAÇÕES AUTOMÁTICAS**. Brasília: Instituto Nacional de Meteorologia, 2023. Disponível em: <https://portal.inmet.gov.br/servicos/esta%C3%A7%C3%B5es-autom%C3%A1ticas>. Acesso em: 12 out. 2023.

OCEAN, N.; ADMINISTRATION, A. **The globe had its sixth-warmest year on record**. United States: National Ocean and Atmospheric Administration, 2023. Disponível em: <https://search.usa.gov/search?utf8=%E2%9C%93&affiliate=noaa.gov&query=warmest+years&commit=>. Acesso em: 12 out. 2023.

OCEANOGRAPHY, S. I. of. **The globe had its sixth-warmest year on record**. United States: Scripps Institution of Oceanography, 2023. Disponível em: <https://scrippsco2.ucsd.edu/>. Acesso em: 12 out. 2023.

ORACLE. **O que é IoT?** United States: Oracle, 2023. Disponível em: [https://www.oracle.com/br/internet-of-things/what-is-iot/#:~:text=A%20Internet%20das%20Coisas%20\(IoT\)%20descreve%20a%20rede%20de%20objetos,comuns%20a%20ferramentas%20industriais%20sofisticadas](https://www.oracle.com/br/internet-of-things/what-is-iot/#:~:text=A%20Internet%20das%20Coisas%20(IoT)%20descreve%20a%20rede%20de%20objetos,comuns%20a%20ferramentas%20industriais%20sofisticadas). Acesso em: 14 out. 2023.

APÊNDICE A – CÓDIGOS-FONTE UTILIZADOS NO PROJETO

Código-fonte 1 – Código-fonte da placa Arduino

```
1 #include <DHT.h>
2 #include <DHT_U.h>
3 #include <SPI.h>
4 #include <SD.h>
5 #include <DS3231.h>
6 #include <SoftwareSerial.h>
7 #include <Adafruit_INA219.h>
8
9 String status_internet_txt;
10 float status_internet_aux;
11 bool status_internet;
12 int aguardo = 50;
13 int contador_vel = 0;
14 int contador_temp = 0;
15 int contador_umd = 0;
16 int contador_rad = 0;
17 int variaveis = 10;
18 int dia_atual;
19 float aferir_rad = 1.00;
20 String salvar_dados_txt;
21 float salvar_dados;
22 #define led_g 46
23 #define led_b 47
24 #define led_w 48
25
26 const float pi = 3.14159265359;
27 int periodo = 5000;
28 int ativacoes = 0;
29 int const_sensor = 3;
```

```
30 int raio = 46;
31 int voltas = 0;
32 int RPM = 0;
33 float velocidade_do_vento = 0.0;
34 float velocidade_do_vento_media = 0.0;
35 float velocidade_do_vento_media_aux = 0.0;
36
37 #define DHTPIN 3
38 #define DHTTYPE DHT11
39 DHT dht(DHTPIN, DHTTYPE);
40 float temperatura = 0.0;
41 float temperatura_media = 0.0;
42 float temperatura_media_aux = 0.0;
43 float umidade = 0.0;
44 float umidade_media = 0.0;
45 float umidade_media_aux = 0.0;
46
47 float corrente = 0;
48 float radiacao = 0;
49 float radiacao_media = 0;
50 float radiacao_media_aux = 0;
51
52 #define CS 53
53 File arquivo;
54 File arquivo_csv;
55
56 DS3231 rtc(SDA, SCL);
57 Time data_hora;
58
59 SoftwareSerial mySerial(10, 11);
60
61 Adafruit_INA219 ina219;
```

```
62
63 void setup() {
64     pinMode(2, INPUT);
65     digitalWrite(2, HIGH);
66     pinMode(led_g, OUTPUT);
67     pinMode(led_b, OUTPUT);
68     pinMode(led_w, OUTPUT);
69
70     Serial.begin(9600);
71     mySerial.begin(9600);
72
73     while(!Serial);
74     while(!mySerial);
75     delay(aguardo);
76     LimparBuffer();
77     delay(aguardo);
78
79     dht.begin();
80     rtc.begin();
81
82     data_hora = rtc.getTime();
83     dia_atual = data_hora.date;
84
85     if (data_hora.hour >= 6 && data_hora.hour < 18) {
86         aferir_rad = 1.00;}
87     else {aferir_rad = 0.00;}
88
89     if (!ina219.begin()) {
90         Serial.println("INA219 n o foi inicializado!");
91         while (1) { delay(10); }
92     }
93     ina219.setCalibration_16V_400mA();
```



```
93
94  if(SD.begin(CS)) {
95      digitalWrite(led_g, HIGH);
96  }
97
98  Serial.println("
99      =====
100     ");
101  Serial.println("
102     ESTA  O INICIALIZADA !!
103
104     ");
105  Serial.println("
106     =====
107     ");
108  Serial.println();
109  LimparBuffer();
110
111 }
112
113 void loop() {
114     Aguardar_resposta();
115     delay(aguardo);
116     Checar_transmissao_recebida_esp();
117     Checar_conexao_datalogger(false);
118     data_hora = rtc.getTime();
119     if (salvar_dados) { Ciclo_com_armazemento(); }
120     else { Ciclo_sem_armazenamento(); }
121 }
122
```

```
117 void Checar_transmissao_recebida_esp() {
118     status_internet_txt = mySerial.readStringUntil(';');
119     delay(aguardo);
120     salvar_dados_txt = mySerial.readStringUntil(';');
121     delay(aguardo);
122     LimparBuffer();
123     status_internet_aux = status_internet_txt.toFloat();
124     delay(aguardo);
125     salvar_dados = salvar_dados_txt.toFloat();
126     if (status_internet_aux) {
127         digitalWrite(led_b, HIGH);
128         status_internet = true;
129
130     } else {
131         digitalWrite(led_b, LOW);
132         status_internet = false;
133
134     }
135
136 }
137
138 void Ciclo_sem_armazenamento() {
139     Exibir_iniciando_coleta();
140     delay(aguardo);
141     Coletar_dados();
142     Exibir_finalizando_coleta();
143     Exibir_resultado_ciclo();
144     mySerial.write(status_internet);
145     delay(aguardo);
146     Aguardar_entrega();
147     delay(aguardo);
148 }
```

```
149 }
150
151 void Ciclo_com_armazejamento() {
152     Exibir_iniciando_coleta();
153     digitalWrite(led_w, HIGH);
154     delay(aguardo);
155     Coletar_dados();
156     Exibir_finalizando_coleta();
157     Exibir_resultado_ciclo();
158     Armazenamento_dos_dados();
159     digitalWrite(led_w, LOW);
160
161 }
162
163 void Armazenamento_dos_dados() {
164     Armazenar_coleta_datalogger();
165
166     if (status_internet) {
167         Enviar_dados_esp();
168
169     } else {
170         mySerial.write(status_internet);
171         Aguardar_entrega();
172         delay(aguardo);
173
174     }
175
176 }
177
178 void Iniciar_contagem_rpm_anemometro() {
179     ativacoes = 0;
180     attachInterrupt(0, addcontador, FALLING);
```

```
181     long inicio_da_execucao_da_funcao = millis();
182     while (millis() < inicio_da_execucao_da_funcao + periodo)
183         {}
184
185     detachInterrupt(0);
186
187 }
188
189 void addcontador() {
190     ativacoes++;
191 }
192
193 void Calcular_RPM() {
194     RPM = ((ativacoes/const_sensor) * 60) / (periodo / 1000);
195 }
196
197 float Aferir_velocidade_do_vento() {
198     Calcular_RPM();
199     float vel;
200     vel = (((2 * pi * raio * RPM) / 60) / 1000) * 3.6;
201     return vel;
202 }
203
204
205
206 float Aferir_temperatura() {
207     float t;
208     t = dht.readTemperature();
209     return t;
210 }
211 }
```

```
212
213 float Aferir_umidade() {
214     float h;
215     h = dht.readHumidity();
216     return h;
217 }
218
219 float Aferir_radiacao() {
220     float rad;
221     corrente = ina219.getCurrent_mA();
222     rad = 0.0003*corrente*corrente + 4.5019*corrente +
223         55.312;
224     return rad;
225 }
226
227 void Exibir_iniciando_coleta() {
228     if (salvar_dados) {
229         Serial.println();
230         Serial.println("
231             -----
232             ");
233         Serial.print("INICIANDO COLETA E PREPARANDO OS DADOS
234             PARA ARMAZENAMENTO - ");
235         Data_hora();
236         Serial.print(" ...");
237     } else {
238         Serial.println();
239         Serial.println("
240             -----
241             ");
242         Serial.print("INICIANDO COLETA ");
```

```
238     Data_hora();
239     Serial.print(" ...");
240 }
241
242
243 }
244
245 void Exibir_finalizando_coleta() {
246     Serial.println("   FINALIZADA.");
247     Serial.println("
-----
");
248
249 }
250
251 void Exibir_velocidade_do_vento(float dado, float
    dado_media) {
252     Serial.print("Velocidade do Vento (Instant nea): ");
253     if(dado < 10) Serial.print("0");
254     Serial.print(dado);
255     Serial.print("km/h   <----->   ");
256     Serial.print("Velocidade do Vento (M dia): ");
257     Serial.print(dado_media);
258     Serial.println("km/h");
259
260 }
261
262 void Exibir_temperatura(float dado, float dado_media) {
263     Serial.print("Temperatura (Instant nea): ");
264     Serial.print(dado);
265     Serial.print(" C           <----->   ");
266     Serial.print("Temperatura (M dia): ");
```

```
267 Serial.print(dado_media);
268 Serial.println(" C ");
269
270 }
271
272 void Exibir_umidade(float dado, float dado_media) {
273     Serial.print("Umidade (Instantanea): ");
274     Serial.print(dado);
275     Serial.print("%          <-----> ");
276     Serial.print("Umidade (Media): ");
277     Serial.print(dado_media);
278     Serial.println("%");
279
280 }
281
282 void Exibir_radiacao(float dado, float dado_media) {
283     Serial.print("Radiao Solar (Instantanea): ");
284     Serial.print(dado);
285     Serial.print("W/m          <-----> ");
286     Serial.print("Radiao Solar (Media): ");
287     Serial.print(dado_media);
288     Serial.println("W/m ");
289
290 }
291
292 void Armazenar_coleta_datalogger() {
293     if (Checar_conexao_datalogger(true)) {
294         delay(aguardo);
295         Salvar_arquivo_txt();
296         delay(aguardo);
297         Salvar_arquivo_csv();
298         delay(aguardo);
```

```
299     }
300
301 }
302
303 bool Checar_conexao_datalogger(bool exibir) {
304     if(!SD.begin(CS))
305     {
306         Mensagem_erro_conexao_modulo_sd();
307         digitalWrite(led_g, LOW);
308         return false;
309     }
310
311     else
312     {
313         if(exibir) {Mensagem_sucesso_conexao_modulo_sd(exibir)
314             ;}
315         digitalWrite(led_g, HIGH);
316         return true;
317     }
318 }
319
320 void Salvar_arquivo_txt() {
321     arquivo = SD.open("dados.txt", FILE_WRITE);
322
323     if(arquivo)
324     {
325         Salvar_dados_txt();
326     }
327
328     else
329     {
```



```
330     Mensagem_de_erro_ao_gravar_o_arquivo_txt();
331 }
332
333     arquivo.close();
334 }
335
336 void Salvar_arquivo_csv() {
337     arquivo_csv = SD.open("dadoscsv.txt", FILE_WRITE);
338
339     if(arquivo_csv)
340     {
341         Salvar_dados_csv();
342     }
343
344     else
345     {
346         Mensagem_de_erro_ao_gravar_o_arquivo_csv();
347     }
348
349     arquivo_csv.close();
350 }
351
352 void Salvar_dados_txt() {
353     Serial.println("Salvando os dados no SD card...");
354
355     arquivo.println("
356         -----
357         ");
358     arquivo.print("COLETA");
359     arquivo.print(" -> ");
360     arquivo.print(rtc.getDateStr());
361     arquivo.print(" | ");
```

```
360 arquivo.println(rtc.getTimeStr());
361 arquivo.println("
    -----
    ");
362 arquivo.println();
363
364 arquivo.print("Velocidade do Vento: ");
365 arquivo.print(velocidade_do_vento);
366 arquivo.print("km/h <-----> ");
367 arquivo.print("Velocidade do Vento (M dia): ");
368 arquivo.print(velocidade_do_vento_media);
369 arquivo.println("km/h");
370
371 arquivo.print("Temperatura: ");
372 arquivo.print(temperatura);
373 arquivo.print(" C <-----> ");
374 arquivo.print("Temperatura do Local Media: ");
375 arquivo.print(temperatura_media);
376 arquivo.println(" C ");
377
378 arquivo.print("Umidade: ");
379 arquivo.print(umidade);
380 arquivo.print("% <-----> ");
381 arquivo.print("Umidade do Local M dia: ");
382 arquivo.print(umidade_media);
383 arquivo.println("%");
384
385 if (aferir_rad) {
386     arquivo.print("Radia o Solar: ");
387     arquivo.print(radiacao);
388     arquivo.print("W/m <-----> ");
389     arquivo.print("Radia o Solar M dia: ");
```

```
390     arquivo.print(radiacao_media);
391     arquivo.println("W/m  ");
392
393 }
394
395 arquivo.println("
    -----
    ");
396 arquivo.println();
397 arquivo.println();
398 arquivo.println();
399
400 Serial.print("Dados TXT Salvo! - ");
401
402 }
403
404 void Salvar_dados_csv() {
405     if(arquivo_csv.size() == 0)
406     {
407         Cabecalho_csv();
408         Corpo_csv();
409     }
410     else
411     {
412         Corpo_csv();
413     }
414 }
415
416
417 void Cabecalho_csv() {
418     arquivo_csv.println("Data,Hora,Velocidade do vento,
        Velocidade do vento M dia ,Temperatura, Temperatura
```

```
        M dia ,Umidade,Umidade M dia ,Radiacao Instant nea ,
        Radia o M dia");
419 }
420
421 void Corpo_csv() {
422
423     arquivo_csv.print(rtc.getDateStr());
424     arquivo_csv.print(",");
425     arquivo_csv.print(rtc.getTimeStr());
426     arquivo_csv.print(",");
427     arquivo_csv.print(velocidade_do_vento);
428     arquivo_csv.print(",");
429     arquivo_csv.print(velocidade_do_vento_media);
430     arquivo_csv.print(",");
431     arquivo_csv.print(temperatura);
432     arquivo_csv.print(",");
433     arquivo_csv.print(temperatura_media);
434     arquivo_csv.print(",");
435     arquivo_csv.print(umidade);
436     arquivo_csv.print(",");
437     arquivo_csv.print(umidade_media);
438     arquivo_csv.print(",");
439
440     if (aferir_rad) {
441         arquivo_csv.print(radiacao);
442         arquivo_csv.print(",");
443         arquivo_csv.println(radiacao_media);
444     }
445
446     else {
447         arquivo_csv.print("-");
448         arquivo_csv.print(",");
```

```
449     arquivo_csv.println("-");
450 }
451
452 Serial.println("Dados CSV Salvo!");
453 Serial.println("
454     -----
455     ");
456
457 if (!status_internet) {
458     Serial.println();
459     Serial.println();
460 }
461 }
462
463 void Mensagem_erro_conexao_modulo_sd() {
464     Serial.println("
465     -----
466     ");
467     Serial.println("0 Data Logger n o est conectado
468     corretamente!");
469     Serial.println("
470     -----
471     ");
472 }
473
474 void Mensagem_sucesso_conexao_modulo_sd(bool exibir) {
475     if(exibir) {
476         Serial.println("
477         -----
478         ");
```

```
471     Serial.println("Conex o do Data Logger funcionando
           corretamente!");
472     Serial.println("
           -----
           ");
473 }
474
475 }
476
477 void Mensagem_de_erro_ao_gravar_o_arquivo_txt() {
478     Serial.println("Erro ao abrir o arquivo TXT para gravar
           os dados!");
479     Serial.println("
           -----
           ");
480 }
481
482 void Mensagem_de_erro_ao_gravar_o_arquivo_csv() {
483     Serial.println("Erro ao abrir o arquivo CSV para gravar
           os dados!");
484     Serial.println("
           -----
           ");
485     if (!status_internet) {
486         Serial.println();
487         Serial.println();
488     }
489 }
490
491 void Enviar_dados_esp() {
492     Transferir_dados();
493 }
```

```
494 }
495
496 void Transferir_dados() {;
497     float dados[variaveis] = {data_hora.hour, aferir_rad,
498         velocidade_do_vento, velocidade_do_vento_media,
499         temperatura, temperatura_media, umidade, umidade_media
500         , radiacao, radiacao_media};
501
502     if (!aferir_rad) {
503         dados[8] = 0.0;
504         dados[9] = 0.0;
505     }
506
507     String dados_txt_string = "";
508     for (int i = 0; i < variaveis; i++) {
509         dados_txt_string.concat(dados[i]);
510         dados_txt_string.concat(";");
511     }
512
513     int j = dados_txt_string.length();
514     char dados_txt_char[j];
515     dados_txt_string.toCharArray(dados_txt_char, j);
516
517     LimparBuffer();
518     mySerial.write(dados_txt_char);
519     delay(aguardo);
520     Serial.println("Dados Enviados Para a Placa ESP01!");
521     Serial.println("
522         -----
523         ");
524     Serial.println();
```

```
521 Serial.println();
522 Aguardar_entrega();
523 delay(aguardo);
524
525 }
526
527 void LimparBuffer() {
528     while(mySerial.available())
529     {
530         mySerial.read();
531     }
532 }
533
534 void Aguardar_entrega() {
535     while (mySerial.available()) {
536     }
537
538 }
539
540 void Aguardar_resposta() {
541     while (!mySerial.available()) {
542     }
543
544 }
545
546 void Coletar_dados() {
547     Checar_mudanca_de_dia();
548     delay(aguardo);
549     Iniciar_contagem_rpm_anemometro();
550     delay(aguardo);
551     Calcular_medias();
552
```



```
553 }
554
555 void Exibir_resultado_ciclo() {
556     if(salvar_dados) {
557         Exibir_velocidade_do_vento(velocidade_do_vento,
558             velocidade_do_vento_media);
559
560         Exibir_temperatura(temperatura, temperatura_media);
561
562         Exibir_umidade(umidade, umidade_media);
563
564         if (data_hora.hour >= 6 && data_hora.hour < 18) {
565             aferir_rad = 1.00;
566             Exibir_radiacao(radiacao, radiacao_media);
567
568         } else {
569             aferir_rad = 0.00;
570         }
571     } else {
572         Serial.println("M dias do dia at o momento:");
573         Serial.println("
574             -----
575             ");
576         Serial.print("Velocidade do vento: ");
577         Serial.print(velocidade_do_vento_media);
578         Serial.println("Km/h");
579         Serial.print("Temperatura: ");
580         Serial.print(temperatura_media);
581         Serial.println(" C ");
582         Serial.print("Umidade: ");
583         Serial.print(umidade_media);
```

```
582     Serial.println("%");
583     Serial.print("Radiação Solar: ");
584     Serial.print(radiacao_media);
585     Serial.println("W/m ");
586     Serial.println("
        -----
        ");
587     Serial.println();
588
589 }
590
591 }
592
593
594 void Checar_mudanca_de_dia() {
595     if (dia_atual != data_hora.date) {
596         dia_atual = data_hora.date;
597         Zerar_medias();
598     }
599 }
600
601 }
602
603 void Zerar_medias() {
604     velocidade_do_vento_media_aux = 0;
605     temperatura_media_aux = 0;
606     umidade_media_aux = 0;
607     radiacao_media_aux = 0;
608     contador_vel = 0;
609     contador_temp = 0;
610     contador_umd = 0;
611     contador_rad = 0;
```

```
612
613 }
614
615 void Calcular_medias() {
616     velocidade_do_vento = Aferir_velocidade_do_vento();
617     if (!isnan(velocidade_do_vento) && velocidade_do_vento >
618         0) {
619         velocidade_do_vento_media_aux += velocidade_do_vento;
620         contador_vel++;
621     }
622     if (contador_vel) {velocidade_do_vento_media =
623         velocidade_do_vento_media_aux / contador_vel;}
624     else {velocidade_do_vento_media = 0;}
625
626     temperatura = Aferir_temperatura();
627     if (!isnan(temperatura)) {
628         temperatura_media_aux += temperatura;
629         contador_temp++;
630     }
631     if (contador_temp) {temperatura_media =
632         temperatura_media_aux / contador_temp;}
633     else {temperatura_media = 0;}
634
635     umidade = Aferir_umidade();
636     if (!isnan(umidade)) {
637         umidade_media_aux += umidade;
638         contador_umd++;
639     }
640     if (contador_umd) {umidade_media = umidade_media_aux /
641         contador_umd;}
642 }
```

```
640     else {umidade_media = 0;}
641
642
643     if (aferir_rad) {
644         radiacao = Aferir_radiacao();
645         if (!isnan(radiacao)) {
646             radiacao_media_aux += radiacao;
647             contador_rad++;
648         }
649         if (contador_rad) {radiacao_media = radiacao_media_aux /
        contador_rad;}
650     else {radiacao_media = 0;}
651
652     }
653 }
654
655
656 void Data_hora() {
657     Serial.print(" -> ");
658     Serial.print(rtc.getDateStr());
659     Serial.print(" | ");
660     Serial.print(rtc.getTimeStr());
661
662 }
```

Código-fonte 2 – Código-fonte da placa ESP

```
1 // #define IO_USERNAME NOME DO USU RIO
2 // #define IO_KEY CHAVE DO USUARIO
3
4 // #define WIFI_SSID NOME DA REDE
5 // #define WIFI_PASS SENHA
```

```
6 #include "AdafruitIO_WiFi.h"
7
8 #if defined(USE_AIRLIFT) || defined(
    ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
9     defined(ADAFRUIT_PYPORTAL)
10 #if !defined(SPIWIFI_SS)
11 #define SPIWIFI SPI
12 #define SPIWIFI_SS 10
13 #define NINA_ACK 9
14 #define NINA_RESETN 6
15 #define NINA_GPIO0 -1
16 #endif
17 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID,
    WIFI_PASS, SPIWIFI_SS,
18                     NINA_ACK, NINA_RESETN, NINA_GPIO0, &
    SPIWIFI);
19 #else
20 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID,
    WIFI_PASS);
21 #endif
22
23 #include <SoftwareSerial.h>
24
25
26 SoftwareSerial mySerial(0, 2);
27
28 int intervalo = 30000;
29 int aux_rad_total = 0;
30 int aux_rad_total_2 = 0;
31 unsigned long aux_intervalo = 35000;
32 int aguardo = 50;
33 int subir_datos = 0;
```

```
34 String dados = "";
35 String velocidade_vento_txt = "";
36 String temperatura_txt = "";
37 String umidade_txt = "";
38 String radiacao_txt = "";
39 String velocidade_vento_media_txt = "";
40 String temperatura_media_txt = "";
41 String umidade_media_txt = "";
42 String radiacao_media_txt = "";
43 String aferir_rad_txt = "";
44 String internet_envio;
45 String hora_txt;
46
47 float vel_vento;
48 float temp;
49 float umid;
50 float rad;
51 float vel_vento_media;
52 float temp_media;
53 float umid_media;
54 float rad_media;
55 float hora;
56 float aferir_rad;
57
58 int status_internet = 1;
59 int tempo_espera_conexao = 15000;
60 int aux_conexao = 25000;
61 unsigned long aux_tempo = 0;
62
63 AdafruitIO_Feed *velocidade_do_vento = io.feed("Velocidade
    Do Vento - Atual");
64 AdafruitIO_Feed *temperatura = io.feed("Temperatura - Atual
```

```
");
65 AdafruitIO_Feed *umidade = io.feed("Umidade - Atual");
66 AdafruitIO_Feed *radiacao = io.feed("Radiação Solar -
    Atual");
67 AdafruitIO_Feed *velocidade_do_vento_media = io.feed("
    Velocidade Do Vento - Média");
68 AdafruitIO_Feed *temperatura_media = io.feed("Temperatura -
    Média");
69 AdafruitIO_Feed *umidade_media = io.feed("Umidade - Média"
    );
70 AdafruitIO_Feed *radiacao_media = io.feed("Radiação Solar
    - Média");
71 AdafruitIO_Feed *radiacao_total = io.feed("Radiação Solar
    - Total");
72
73 void setup() {
74
75     delay(25000);
76     Serial.begin(9600);
77     mySerial.begin(9600);
78     while(!Serial);
79     while(!mySerial);
80     delay(aguardo);
81
82     Iniciar_conexao();
83
84     if (status_internet) {Mensagem_io_status_text();}
85     else {Mensagem_falha_na_conexao();}
86
87     LimparBuffer();
88     aux_tempo = millis();
89
```

```
90 }
91
92 void loop() {
93     io.run();
94     Checar_conexao_wifi();
95     if (aux_tempo >= aux_intervalo + intervalo) {
96         Solicitar_dados_e_realizar_upload_nuvem();}
97     else {
98         Realizar_comunicacao_com_arduino();
99         aux_tempo += 7500;
100     }
101 }
102
103 void Enviar_status() {
104     internet_envio = String(status_internet) + ";" + String(
105         subir_dados) + ";";
106     int j = internet_envio.length();
107     char dados_txt_char[j];
108     internet_envio.toCharArray(dados_txt_char, j);
109     LimparBuffer();
110     mySerial.write(dados_txt_char);
111     delay(aguardo);
112
113
114 }
115
116 void Solicitar_dados_e_realizar_upload_nuvem() {
117     subir_dados = 1.0;
118     LimparBuffer();
119     Enviar_status();
```



```
120     Aguardar_entrega();
121     delay(aguardo);
122     Aguardar_resposta();
123     delay(aguardo);
124     Conferir_resposta_arduino();
125     Checar_conexao_wifi();
126     subir_dados = 0;
127     aux_intervalo = aux_tempo;
128
129 }
130
131 void Realizar_comunicacao_com_arduino() {
132     Enviar_status();
133     Aguardar_entrega();
134     delay(aguardo);
135     Aguardar_resposta();
136     delay(aguardo);
137     LimparBuffer();
138     delay(aguardo);
139
140 }
141
142 void Envio_dos_dados_para_nuvem() {
143     if (io.status() == AIO_CONNECTED) {
144         hora_txt = mySerial.readStringUntil(';');
145         hora = hora_txt.toFloat();
146         aferir_rad_txt = mySerial.readStringUntil(';');
147         aferir_rad = aferir_rad_txt.toFloat();
148         velocidade_vento_txt = mySerial.readStringUntil(';');
149         velocidade_vento_media_txt = mySerial.readStringUntil('
            ');
150         temperatura_txt = mySerial.readStringUntil(';');
```

```
151     temperatura_media_txt = mySerial.readStringUntil(';');
152     umidade_txt = mySerial.readStringUntil(';');
153     umidade_media_txt = mySerial.readStringUntil(';');
154     if(aferir_rad) {
155         radiacao_txt = mySerial.readStringUntil(';');
156         radiacao_media_txt = mySerial.readStringUntil(';');
157     }
158
159
160     vel_vento = velocidade_vento_txt.toFloat();
161     vel_vento_media = velocidade_vento_media_txt.toFloat();
162     temp = temperatura_txt.toFloat();
163     temp_media = temperatura_media_txt.toFloat();
164     umid = umidade_txt.toFloat();
165     umid_media = umidade_media_txt.toFloat();
166     if (aferir_rad) {
167         rad = radiacao_txt.toFloat();
168         rad_media = radiacao_media_txt.toFloat();
169     }
170
171
172     Serial.println("
    -----
    ");
173     Serial.println("Esta o Conectada internet -
    Enviando dados para a nuvem...");
174     Serial.println("
    -----
    ");
175
176     Serial.print("Enviando Velocidade Do Vento -> ");
177     Serial.print(vel_vento);
```

```
178 Serial.println(" Km/h");
179 velocidade_do_vento->save(vel_vento);
180 Serial.print("Enviando Velocidade Do Vento M dia -> ")
    ;
181 Serial.print(vel_vento_media);
182 Serial.println(" Km/h");
183 velocidade_do_vento_media->save(vel_vento_media);
184
185 Serial.print("Enviando Temperatura -> ");
186 Serial.print(temp);
187 Serial.println(" C ");
188 temperatura->save(temp);
189 Serial.print("Enviando Temperatura M dia -> ");
190 Serial.print(temp_media);
191 Serial.println(" C ");
192 temperatura_media->save(temp_media);
193
194 Serial.print("Enviando Umidade -> ");
195 Serial.print(umid);
196 Serial.println(" %");
197 umidade->save(umid);
198 Serial.print("Enviando Umidade M dia -> ");
199 Serial.print(umid_media);
200 Serial.println(" %");
201 umidade_media->save(umid_media);
202
203 if (aferir_rad) {
204     aux_rad_total = 0;
205     aux_rad_total_2 = 0;
206     Serial.print("Enviando Radia o Solar Global -> ");
207     Serial.print(rad);
208     Serial.println(" Wh/m ");
```

```
209     radiacao->save(rad);
210     Serial.print("Enviando Radiação Solar Global Média
211     -> ");
211     Serial.print(rad_media);
212     Serial.println(" Wh/m ");
213     radiacao_media->save(rad_media);
214 } else {
215     if (hora >= 18) {
216         if (aux_rad_total == 0) {
217             radiacao_total->save(rad_media*12);
218             aux_rad_total = 1;
219         }
220     } else {
221         if (aux_rad_total_2 == 0) {
222             radiacao_total->save(0);
223             aux_rad_total_2 = 1;
224         }
225     }
226 }
227
228
229     Mensagem_dados_enviados();
230 }
231
232 else {
233     Mensagem_estacao_desconectada();
234 }
235
236 }
237
238 void Iniciar_conexao() {
239     Serial.println("
```

```
-----  
    ");  
240 Serial.print("Conectando ao Adafruit IO ");  
241 io.connect();  
242 while(io.status() < AIO_CONNECTED) {  
243     Serial.print(".");  
244     delay(300);  
245     if (millis() > aux_conexao + tempo_espera_conexao) {  
246         status_internet = false;  
247         break;  
248     }  
249 }  
250  
251 }  
252  
253 void Conferir_resposta_arduino() {  
254     if (status_internet) {  
255         Envio_dos_dados_para_nuvem();  
256     }  
257  
258     else {  
259         status_internet = false;  
260         Mensagem_estacao_desconectada();  
261     }  
262  
263 }  
264  
265 void Checar_conexao_wifi() {  
266     if (io.status() == AIO_CONNECTED) {  
267         status_internet = true;  
268     }  
269
```

```
270     else {
271         status_internet = false;
272     }
273
274 }
275
276 void LimparBuffer() {
277     while(mySerial.available())
278     {
279         mySerial.read();
280     }
281 }
282
283 void Aguardar_entrega() {
284     while (mySerial.available()) {
285         }
286 }
287
288 void Aguardar_resposta() {
289     while (!mySerial.available()) {
290
291     }
292 }
293
294 void Mensagem_estacao_conectada() {
295     Serial.println("
296         -----
297         ");
298     Serial.println("Esta o conectada na internet");
299     Serial.println("
300         -----
301         ");
```

```
298 Serial.println();
299
300 }
301
302 void Mensagem_estacao_desconectada() {
303     Serial.println("
304         -----
305         ");
306     Serial.println("A esta o est sem conex o com a
307         internet. Os dados coletados ser o salvos somente no
308         Data Logger.");
309     Serial.println("
310         -----
311         ");
312     Serial.println();
313 }
314
315 void Mensagem_dados_enviados() {
316     Serial.println("
317         -----
318         ");
319     Serial.println("Dados enviados para a nuvem com sucesso!"
320     );
321     Serial.println("
322         -----
323         ");
324     Serial.println();
325     Serial.println();
326 }
327
328 }
```

```
319 void Mensagem_io_status_text() {
320     Serial.println();
321     Serial.println("
        -----
        ");
322     Serial.println(io.statusText());
323
324 }
325
326 void Mensagem_falha_na_conexao() {
327     Serial.println();
328     Serial.println("
        -----
        ");
329     Serial.println("N o foi poss vel se conectar
        Internet. Os dados coletados ser o salvos somente no
        Data Logger at o estabelecimento da conex o.");
330     Serial.println("
        -----
        ");
331     Serial.println();
332     Serial.println();
333
334 }
```