



**UNIVERSIDADE FEDERAL DE ALFENAS**  
Bacharelado Interdisciplinar em Ciência e Tecnologia

**JOÃO GUILHERME CARVALHO ROBERTO**  
**SÉRGIO RODRIGO MAGGION MELIM**

**APRIMORAMENTO E CARACTERIZAÇÃO DE UM PLUVIÔMETRO DE  
BÁSCULA RECONDICIONADO**

**POÇOS DE CALDAS / MG**

**2023**

**JOÃO GUILHERME CARVALHO ROBERTO**  
**SÉRGIO RODRIGO MAGGION MELIM**

**APRIMORAMENTO E CARACTERIZAÇÃO DE UM PLUVIÔMETRO DE  
BÁSCULA RECONDICIONADO**

Trabalho de Conclusão PIEPEx (PCP)  
apresentado como parte dos requisitos para  
obtenção do título de bacharel em ciência e  
tecnologia pela Universidade Federal de  
Alfenas

Orientador: Prof. Dr. Laos Alexandre Hirano

**POÇOS DE CALDAS / MG**

**2023**

Sistema de Bibliotecas da Universidade Federal de Alfenas  
Biblioteca Campus Poços de Caldas

Roberto, João Guilherme Carvalho.

Aprimoramento e caracterização de um pluviômetro de báscula  
recondicionado / João Guilherme Carvalho Roberto, Sérgio Rodrigo Maggion  
Melim. - Poços de Caldas, MG, 2023.

95 f. : il. -

Orientador(a): Laos Alexandre Hirano.

Trabalho de Conclusão de Curso (Graduação em Bacharelado  
Interdisciplinar em Ciência e Tecnologia) - Universidade Federal de Alfenas,  
Poços de Caldas, MG, 2023.

Bibliografia.

1. Pluviômetro de Báscula. 2. ESP8266. 3. IoT. I. Melim, Sérgio Rodrigo  
Maggion. II. Hirano, Laos Alexandre, orient. III. Título.

Ficha gerada automaticamente com dados fornecidos pelo autor.

**JOÃO GUILHERME CARVALHO ROBERTO**  
**SÉRGIO RODRIGO MAGGION MELIM**

**APRIMORAMENTO E CARACTERIZAÇÃO DE UM PLUVIÔMETRO DE**  
**BÁSCULA RECONDICIONADO**

O(A) Presidente da banca examinadora abaixo assina a aprovação do PCP apresentado como parte dos requisitos para obtenção do título bacharel em ciência e tecnologia pela Universidade Federal de Alfenas

Aprovada em: xx de xxxx de 2xxx

Prof. Dr. Xxxxxx xx Xxxxxxxxxx

Assinatura:

Universidade xxxxxxxxxxxxxxxx

Prof. Dr. Xxxxxx xx Xxxxxxxxxx

Assinatura:

Universidade Xxxxxxxxxxxxxx

Prof.<sup>a</sup> Dr.<sup>a</sup> Xxxxxxxxxxxxxx

Assinatura:

Universidade Xxxxxxxxxxxxxx

## RESUMO

O monitoramento eficiente de precipitação é crucial em diversas áreas, incluindo meteorologia, urbanismo e agricultura. Este estudo apresenta o desenvolvimento de um pluviômetro digital avançado, empregando o módulo ESP8266 Wemos D1 Mini para a coleta e transmissão ágil de dados meteorológicos. O objetivo é aprimorar a eficiência, acessibilidade e precisão na monitoração da precipitação, sobretudo em locais onde a coleta manual de dados é complexa. Para assegurar a exatidão das informações meteorológicas, implementamos uma metodologia de testes de calibração, que considera variações de erro em diferentes intensidades de chuva. O dispositivo foi desenvolvido a partir da reforma e adaptação de um pluviômetro RainWise inoperante, integrado ao ESP8266 e a um conjunto de sensores especializados. Paralelamente, um servidor web foi elaborado com CSS, JavaScript e HTML, facilitando o monitoramento remoto dos dados. A programação do ESP8266, realizada em C++, permite o tratamento e armazenamento eficaz dos dados. O resultado é um instrumento capaz de fornecer leituras precisas e atualizadas sobre a precipitação e leituras de sensores adicionais, via interface web. Esta inovação tem potencial para facilitar a coleta e análise de dados de precipitação, trazendo benefícios significativos para vários setores.

Palavras-chave: Esp8266 Wemos D1 Mini, Pluviômetro, Web Server, IoT.

## ABSTRACT

Efficient precipitation monitoring is crucial in various fields, including meteorology, urban planning, and agriculture. This study presents the development of an advanced digital rain gauge, employing the ESP8266 Wemos D1 Mini module for the swift collection and transmission of meteorological data. The objective is to enhance efficiency, accessibility, and accuracy in precipitation monitoring, especially in locations where manual data collection is complex. To ensure the accuracy of meteorological information, we implemented a calibration testing methodology that considers error variations at different rainfall intensities. The device was developed by refurbishing and adapting a non-functional RainWise rain gauge, integrating it with ESP8266 and a set of specialized sensors. Simultaneously, a web server was created using CSS, JavaScript, and HTML, facilitating remote monitoring of the data. The ESP8266 programming, conducted in C++, enables effective data processing and storage. The result is an instrument capable of providing accurate and up-to-date readings on precipitation and additional sensor data through a web interface. This innovation has the potential to streamline the collection and analysis of precipitation data, bringing significant benefits to various sectors. The result is a device capable of providing accurate real-time precipitation data, accessible from anywhere via a web user interface. This contribution has the potential to improve precipitation data collection and analysis, providing significant benefits to a wide range of sectors.

Keywords: ESP8266 Wemos D1 Mini, Rain Gauge, Web Server, IoT.

## LISTA DE FIGURAS

Figura 1 - Representação gráfica dos processos físicos relevantes que afetam na medição da precipitação.....	19
Figura 2 - Pluviômetro basculante .....	21
Figura 3 - Tensão no capacitor aumenta exponencialmente com o tempo.....	24
Figura 4 - Tratamento da báscula em PLA com clorofórmio.....	27
Figura 5 - Câmara de acetona utilizada no tratamento da báscula.....	28
Figura 6 - Calibração da báscula .....	29
Figura 7 - Vista da parte superior do pluviômetro, evidenciando a área de coleta.....	30
Figura 8 - Configuração experimental.....	31
Figura 9 - Esquema em blocos do circuito.....	34
Figura 10 - Sistema de medição de carga por circuito RC.....	35
Figura 11 - Sketch do circuito da PCB.....	37
Figura 12 -Algoritmo do código.....	40
Figura 13 - Pluviômetro aberto evidenciando retenção parcial da água durante o escoamento.....	42
Figura 14 - Pluviômetro aberto destacando parafusos de calibração .....	43
Figura 15 - Báscula original com revestimento de teflon .....	44
Figura 16 - Projeto 3D da nova báscula no Tinkercad.....	45
Figura 17 - Protótipo inicial da báscula em PLA .....	46
Figura 18 - Báscula impressa em ABS .....	47
Figura 19 - Báscula em ABS após o tratamento com acetona .....	47
Figura 20 - Dados coletados pela interface do sistema .....	49
Figura 21 - PCB com o desenho das trilhas antes da corrosão.....	51
Figura 22 - Vista inferior da PCB após a corrosão com os componentes soldados.....	51
Figura 23 - Vista dos componentes soldados na PCB.....	52
Figura 24 - Módulo de bateria construído com BMS 1S .....	52
Figura 25 - Conector Jack fixado embaixo do equipamento para entrada de energia .....	53
Figura 26 – Conector fêmea para conexão do DHT22 .....	55
Figura 27 - Entrada P2 fêmea para adição de um sensor analógico.....	56

Figura 28 – Interface: Página principal .....	57
Figura 29 - Interface: Página "atualizar data e hora" .....	58
Figura 30 - Interface: Página "configurar sensor analógico" .....	59
Figura 31 - Interface: Página "configurar DHT" .....	60
Figura 32 - Interface: Página "limpar registros" .....	61
Figura 33 - Protótipo finalizado .....	63

## LISTA DE EQUAÇÕES

Equação 1 – Função $V_c(t)$ no capacitor em carregamento .....	23
Equação 2 – Incerteza total .....	32
Equação 3 – Porcentagem de carga da bateria .....	54

## LISTA DE TABELAS

Tabela 1 - Análise dos erros percentuais do instrumento .....	49
Tabela 2 - Dados experimentais do tempo de detecção de sinal high pelo ESP8266 com base na tensão de alimentação do circuito RC .....	54
Tabela 3 - Consumo de energia por potência da rede wifi.....	62

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
1.1	OBJETIVOS .....	15
1.1.1	<b>Objetivo Geral</b> .....	<b>15</b>
1.1.2	<b>Objetivos específicos</b> .....	<b>15</b>
<b>2</b>	<b>REVISÃO DA LITERATURA</b> .....	<b>17</b>
2.1	sobre o esp8266.....	17
2.2	Aferição de precipitações .....	18
2.2.1	<b>Pluviômetros</b> .....	<b>19</b>
2.2.2	<b>Diferentes tipos de pluviômetros</b> .....	<b>20</b>
2.3	Aferição de nível de carga em baterias .....	21
2.4	influência da Intensidade da Chuva na Precisão dos Pluviômetros de Báscula 22	
2.5	Circuitos RC e a Função $V_c(t)$ em Capacitores .....	23
2.5.1	<b>Carregamento do Capacitor em Circuitos RC</b> .....	<b>23</b>
2.5.2	<b>A Função <math>V_c(t)</math> no Capacitor</b> .....	<b>23</b>
<b>3</b>	<b>METODOLOGIA</b> .....	<b>25</b>
3.1	Experimentação e Avaliação das Básculas.....	25
3.1.1	<b>Tratamento da báscula de PLA com clorofórmio</b> .....	<b>26</b>
3.1.2	<b>Tratamento da báscula de ABS com acetona</b> .....	<b>27</b>
3.2	Processo de Calibração do Pluviômetro.....	28
3.2.1	<b>Testes de Operação: Avaliação de Erro em Diferentes Intensidades de Chuva</b> 30	
3.3	Circuito .....	32
3.3.1	<b>Estrutura e Componentes do Circuito</b> .....	<b>32</b>
3.4	Módulo de baterias e circuito para aferição de nível de carga .....	34

<b>3.4.1</b>	<b>Sistema de Medição de Carga por circuito RC.....</b>	<b>35</b>
<b>3.4.2</b>	<b>Construção de PCB para suporte ao circuito .....</b>	<b>36</b>
<b>3.4.3</b>	<b>Fabricação da PCB.....</b>	<b>36</b>
3.5	Sensores .....	38
3.6	Estratégia comunicação de dados e Testes de conectividade .....	39
3.7	Programação do sistema.....	40
<b>4</b>	<b>RESULTADOS E DISCUSSÕES .....</b>	<b>41</b>
4.1	Báscula.....	41
<b>4.1.1</b>	<b>Análise Crítica dos Problemas com a Báscula Original.....</b>	<b>41</b>
<b>4.1.2</b>	<b>Tentativas de Melhoria e Decisão pela Construção de uma Nova Báscula</b>	<b>43</b>
<b>4.1.3</b>	<b>Desenvolvimento e Aperfeiçoamento da Nova Báscula em Impressão 3D</b>	<b>45</b>
4.2	Calibração do Pluviômetro .....	48
<b>4.2.1</b>	<b>Calibração da Nova Báscula em ABS .....</b>	<b>48</b>
<b>4.2.2</b>	<b>Avaliação do Erro do Instrumento .....</b>	<b>48</b>
4.3	circuito e Sistema de Registro de Dados.....	50
<b>4.3.1</b>	<b>Placa De Circuito Impresso.....</b>	<b>50</b>
<b>4.3.2</b>	<b>Sistema de Baterias.....</b>	<b>52</b>
<b>4.3.3</b>	<b>Sistema de Medição de Carga por circuito RC.....</b>	<b>53</b>
<b>4.3.4</b>	<b>Incrementação do DHT22 .....</b>	<b>55</b>
<b>4.3.5</b>	<b>Sistema adicional – Entrada Analógica .....</b>	<b>55</b>
4.4	Sistema do pluviômetro .....	56
<b>4.4.1</b>	<b>Sistema de registro e comunicação de dados .....</b>	<b>56</b>
4.5	Testes de Conectividade .....	61
<b>4.5.1</b>	<b>Avaliação da Comunicação sem Fio (Wi-Fi) .....</b>	<b>61</b>
4.6	Protótipo .....	62

<b>5</b>	<b>CONCLUSÃO .....</b>	<b>64</b>
<b>6</b>	<b>SUGESTÃO DE TRABALHOS CORRELACIONADOS.....</b>	<b>65</b>
	<b>REFERÊNCIAS.....</b>	<b>66</b>
	<b>ANEXOS .....</b>	<b>69</b>

## 1 INTRODUÇÃO

No contexto brasileiro, a questão da insuficiência de equipamentos meteorológicos operacionais é particularmente premente, refletindo desafios específicos relacionados à gestão de recursos e infraestrutura tecnológica. Este déficit de equipamentos funcionais tem implicações diretas na capacidade de monitoramento e análise climática, aspectos fundamentais para setores como agricultura, gestão hídrica, e planejamento urbano e ambiental.

A principal barreira no Brasil é o alto custo associado à aquisição e manutenção de estações meteorológicas de ponta. Esta questão é exacerbada por limitações orçamentárias e prioridades concorrentes em políticas públicas. Além disso, a dispersão geográfica do país e a diversidade de seus microclimas demandam uma rede extensa de estações meteorológicas, o que amplifica o desafio financeiro e logístico.

Neste cenário, a estratégia de recondicionamento de equipamentos meteorológicos antigos torna-se uma solução técnica e economicamente viável. Muitos desses dispositivos, apesar de possuírem estruturas físicas resilientes, apresentam defasagem tecnológica, especialmente em seus componentes eletrônicos. A substituição ou atualização desses componentes com tecnologias modernas de baixo custo, como microcontroladores e sistemas de comunicação digital, pode revitalizar esses equipamentos. Isso não só reduz significativamente os custos operacionais, mas também melhora a eficácia na coleta e transmissão de dados meteorológicos.

Esta abordagem oferece um caminho para a expansão e modernização da rede de monitoramento climático no Brasil. Ela permite uma melhor utilização dos recursos existentes, promovendo a sustentabilidade e a eficiência na gestão de dados meteorológicos. Contudo, para que essa estratégia seja bem-sucedida, é necessário um investimento contínuo em pesquisa e desenvolvimento, assim como em programas de capacitação técnica, para assegurar que os profissionais envolvidos possam implementar e manter essas soluções de forma eficaz.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

O objetivo deste trabalho é recondicionar um pluviômetro antigo e integrar um sistema que colete, interprete e armazene dados de precipitação de forma precisa e acessível por meio de um servidor web utilizando o ESP8266.

### 1.1.2 Objetivos específicos

- a) Seleção de equipamentos: Buscar microcontroladores e outros componentes necessários que sejam acessíveis e de boa qualidade, com ênfase na precisão e desempenho dos equipamentos, para garantir a qualidade dos dados obtidos.
- b) Projeto do sistema de recondicionamento: Projetar o sistema de recondicionamento do pluviômetro, integrando o ESP8266 e garantindo a interpretação correta dos pulsos do reed switch, para obter dados precisos de precipitação.
- c) Desenvolvimento do servidor web: Desenvolver um servidor web hospedado no ESP8266, permitindo a visualização e exportação dos dados coletados, com o objetivo de tornar as informações facilmente acessíveis.
- d) Calibração e testes em laboratório controlado: Realizar a calibração do pluviômetro recondicionado em um ambiente controlado, bem como testes para validar a precisão do sistema de recondicionamento, permitindo avaliar a qualidade do projeto e identificar pontos de melhoria.
- e) Integração e Suporte a Sensores Adicionais: Ampliar a funcionalidade do sistema de recondicionamento através da integração e suporte a

sensores adicionais. Este objetivo visa aumentar a versatilidade e a capacidade de monitoramento do pluviômetro, permitindo a coleta e análise de uma gama mais ampla de dados ambientais, além das medições de precipitação.

- f) Nível de carga das baterias: Desenvolver um método de baixo custo para aferição do nível de carga das baterias.

## 2 REVISÃO DA LITERATURA

### 2.1 SOBRE O ESP8266

O ESP8266, desenvolvido pela Espressif Systems em 2015, é um microcontrolador que dá suporte ao chip ESP12 e capacidade de conexão Wi-Fi, tornando-o ideal para soluções em Internet das Coisas (IoT). Este microcontrolador é programável e altamente flexível, permitindo a integração com uma gama variada de dispositivos, incluindo LEDs, displays, sensores entre outros. Sua programação pode ser realizada através do Arduino Integrated Development Environment (IDE), que utiliza uma linguagem baseada em C/C++ (Mehta, 2015), além de outras IDEs e linguagens como MicroPython, ampliando suas possibilidades de uso.

Os sensores, como descritos por Karvinen (2014), são componentes críticos em diversas aplicações tecnológicas, capazes de converter informações ambientais em sinais elétricos para processamento. Existem várias categorias de sensores, como cinéticos, térmicos e luminosos, essenciais em campos que vão desde a automação até equipamentos médicos. No contexto do ESP8266, a integração com sensores comerciais e customizados é vital para a medição de variáveis ambientais, garantindo a precisão e qualidade dos dados.

Os sensores digitais, particularmente, fornecem um sinal de saída de tensão que representa um nível digital, como explicado por Thomazin (2020). Eles podem assumir somente dois estados - "ligado" ou "desligado" -, conferindo-lhes uma precisão superior em comparação aos sensores analógicos. Um exemplo notável é o reed switch, que gera um sinal binário dependendo do estado de contato entre suas lâminas internas na presença ou não de um ímã próximo, permitindo a contagem de pulsos.

## 2.2 AFERIÇÃO DE PRECIPITAÇÕES

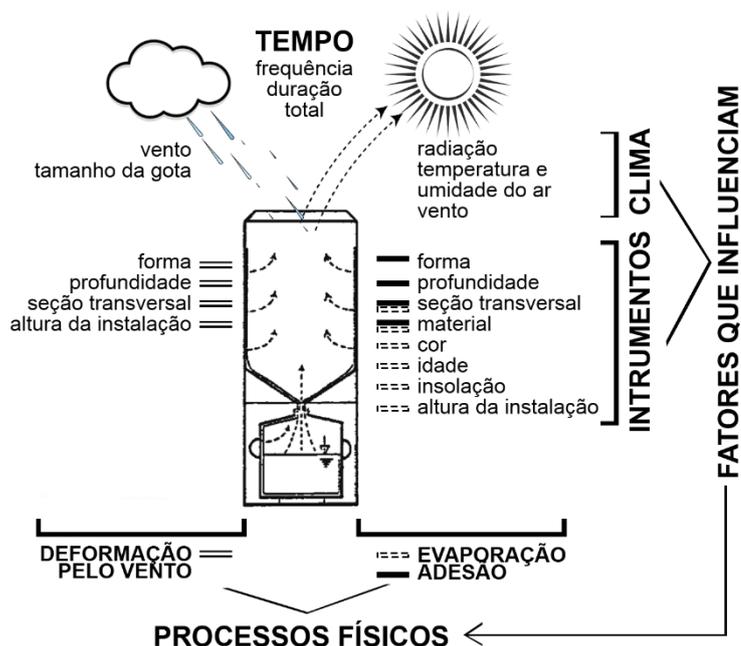
A precipitação é um aspecto importante da climatologia, sendo definida como o produto líquido ou sólido resultante da condensação do vapor de água que cai das nuvens ou é depositado no solo, incluindo chuva, granizo, neve, orvalho, geada e precipitação de neblina (WMO, 2008). As medições são feitas frequentemente a cada hora ou a cada três horas diárias, mas em alguns casos, uma maior resolução temporal é necessária para medir taxas elevadas de precipitação em períodos curtos (WMO, 2008).

O pluviômetro é o instrumento mais comumente usado para medir a precipitação, que é baseado em um recipiente com um funil de seção transversal conhecida voltada para cima, geralmente na forma de um cilindro (WMO, 2008). Ele mede a quantidade de água acumulada, geralmente em milímetros, por unidade de área da seção transversal da entrada da precipitação, geralmente em metros quadrados.

As medições pontuais de precipitação são a principal fonte de dados para análise, pois permitem avaliar a quantidade de chuva em uma determinada região. Entretanto, é importante levar em consideração que cada ponto de medição representa apenas uma área limitada e sua precisão pode ser afetada por diversos fatores, tais como o período de acumulação, a geografia, a homogeneidade da área, a topografia local e o processo de precipitação (WMO, 2008).

A fim de obter uma representação mais precisa da quantidade de precipitação em uma determinada região, é recomendável utilizar uma rede de estações de medição pontual, que permita uma cobertura mais ampla da área em questão. Além disso, é importante considerar os fatores expostos na figura 1 que podem afetar a precisão das medições e buscar mitigá-los, se possível.

Figura 1 - Representação gráfica dos processos físicos relevantes que afetam na medição da precipitação.



Fonte: Grison (2017), adaptada por autores.

### 2.2.1 Pluviômetros

Os pluviômetros são dispositivos de extrema relevância em diversas áreas do conhecimento, especialmente na meteorologia, hidrologia e agricultura, onde se faz necessário um controle detalhado do volume de chuva que cai em uma determinada área. Ayoade (2010) define pluviômetros como instrumentos que coletam e medem a quantidade de precipitação, contribuindo para uma melhor compreensão dos padrões climáticos e possibilitando previsões meteorológicas mais precisas.

Existem dois tipos principais de pluviômetros, cada um com suas características e métodos de medição específicos: o pluviômetro registrador (ou autográfico) e o não-registrador.

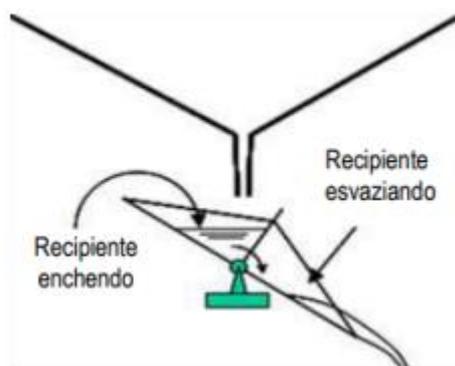
### 2.2.2 Diferentes tipos de pluviômetros

O pluviômetro registrador, também conhecido como autográfico, possui um mecanismo que permite tanto a mensuração do volume como do peso da precipitação que atinge o dispositivo, além de registrar a duração da ocorrência pluviométrica. Ayoade (2010) destaca que existem três variações principais de pluviômetros registradores, diferenciando-se entre si pelos seus mecanismos operacionais: sifão basculante, coletor removível e sistema de coleta por pesagem.

O pluviômetro autográfico de sifão basculante é talvez o modelo mais comum de pluviômetro registrador. À medida que a chuva incide sobre o dispositivo, a água coletada na câmara do aparelho faz com que um flutuador ascenda. Este flutuador está ligado a uma caneta, que, em sua extremidade superior, traça uma linha ascendente em um gráfico, o qual está fixado em um tambor cilíndrico impulsionado por um mecanismo de precisão. Quando a câmara atinge sua capacidade máxima, ela se inclina sobre seu eixo e descarrega a água coletada. Conseqüentemente, o flutuador retorna à posição original e a caneta volta a repousar na base do gráfico. À medida que a chuva prossegue, esse processo se repete, permitindo que cada ciclo completo registre uma quantidade específica de chuva, que é tipicamente de 5 mm.

Modelos mais avançados de pluviômetros utilizam um sistema composto por um ímã fixado em uma balança e um sensor do tipo reed switch. Quando a quantidade de chuva acumulada é suficiente, conforme na figura 2, o ímã é elevado e passa pelo sensor reed switch. Este sensor atua como um interruptor e, ao ser ativado pelo ímã, gera um sinal digital. Este sinal é então registrado, interpretado e armazenado em um dispositivo chamado datalogger para análises futuras. Este processo permite uma medição precisa e confiável da precipitação pluviométrica.

Figura 2 - Pluviômetro basculante



Fonte: Mansano (2015)

Já o pluviômetro não-registrador é mais simples, fornecendo apenas a medida do volume total de precipitação. Nele, a chuva é coletada através de uma abertura de área definida, sendo armazenada em um recipiente interno. A medida do volume de chuva é feita utilizando um frasco graduado ou uma proveta, ambos graduados em milímetros. Este tipo de pluviômetro não registra informações adicionais, como duração ou intensidade da precipitação (AYOADE, 2010).

Em resumo, a escolha do tipo de pluviômetro a ser implementado depende das informações específicas que o pesquisador ou o profissional deseja obter sobre a precipitação. Ambos os tipos têm suas aplicações e utilidades, variando de acordo com a demanda do estudo ou do projeto em questão.

### 2.3 AFERIÇÃO DE NÍVEL DE CARGA EM BATERIAS

O método de estimar o nível de carga de uma bateria por meio da medição de sua tensão, como descrito por Kiehne (2003), baseia-se na relação entre a tensão da bateria e seu estado de carga. Conforme a bateria é utilizada e sua carga diminui, a tensão também cai. Essa relação, que não é linear nem consistente em diferentes tipos de baterias, oferece uma base para estimar o estado de carga.

Para baterias de chumbo-ácido, a tensão em circuito aberto (OCV) pode ser usada para uma estimativa aproximada do estado de carga, como aponta Kiehne (2003). Entretanto, nas baterias de lítio, a situação é mais complexa devido à alta

tensão inicial em circuito aberto, que pode cair significativamente sob carga, conforme explica Kiehne (2003). Assim, a tensão sob carga (CCV) torna-se um indicador mais relevante, sendo influenciada pela corrente de descarga, estado de carga, idade e temperatura da bateria.

Mesmo não sendo o cenário ideal, medindo a tensão ainda é possível obter uma ideia aproximada da quantidade de carga utilizada. Por exemplo, uma bateria de lítio totalmente carregada pode ter uma tensão de cerca de 4.2 volts, que diminui à medida que a carga é consumida. Quando a tensão atinge um certo limiar, como 3.7 volts, isso pode indicar que a bateria está parcialmente descarregada. Porém, é importante considerar que esta é uma aproximação, e outros fatores podem influenciar a leitura da tensão.

Este método é particularmente útil em contextos em que uma precisão absoluta não é necessária, mas uma orientação geral sobre o estado de carga da bateria é suficiente, como em aplicações de monitoramento de baixa criticidade ou em dispositivos onde a precisão do estado de carga não é crítica para o desempenho do sistema.

#### 2.4 INFLUÊNCIA DA INTENSIDADE DA CHUVA NA PRECISÃO DOS PLUVIÔMETROS DE BÁSCULA

A precisão das medições realizadas pelos pluviômetros de balança é significativamente influenciada pela intensidade da chuva. Estudos de Alena et al. (1990), Marsalek (1981) e da Organização Meteorológica Mundial (1983) demonstram que o erro percentual de subestimação nestes instrumentos aumenta de forma não linear com a intensidade da precipitação. Especificamente, para intensidades de chuva superiores a 25 mm por hora, o erro de subestimação pode variar entre 10% e 30%. Tal discrepância é atribuída à incapacidade mecânica da balança de se posicionar rapidamente após alternar a posição, o que leva à subcaptação do volume real de chuva, um problema particularmente acentuado em eventos de chuvas de curta duração e alta intensidade, assim como em tempestades de longa duração com altas intensidades.

## 2.5 CIRCUITOS RC E A FUNÇÃO $V_C(T)$ EM CAPACITORES

Circuitos RC, constituídos pela combinação de um resistor (R) e um capacitor (C), são componentes cruciais em numerosos ramos da física e engenharia. A importância desses circuitos está fundamentalmente ligada ao entendimento da dinâmica de carga e descarga em sistemas eletrônicos. Eles são especialmente relevantes em contextos onde a resposta temporal é um fator decisivo, como na filtragem eficaz de sinais, na precisa temporização em circuitos eletrônicos.

### 2.5.1 Carregamento do Capacitor em Circuitos RC

O processo de carregamento de um capacitor em um circuito RC começa quando a chave é fechada, permitindo que a corrente flua através do resistor para o capacitor. Halliday et al. (2013) descrevem que, inicialmente, o capacitor está descarregado e, à medida que a corrente flui, ele começa a acumular carga, aumentando a diferença de potencial entre suas placas. Esse processo continua até que a tensão no capacitor se iguale à força eletromotriz da fonte.

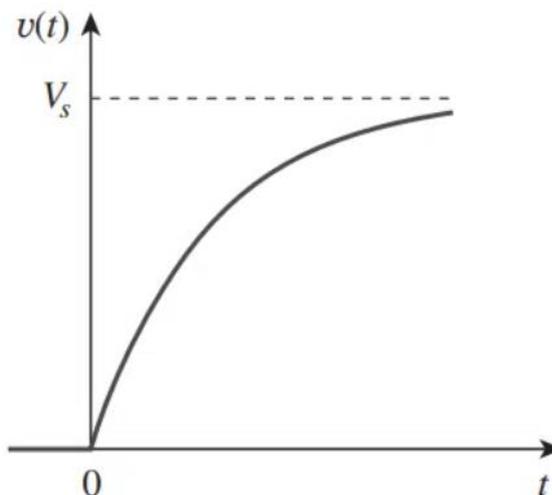
### 2.5.2 A Função $V_C(t)$ no Capacitor

A função  $v_C(t)$  representa a tensão no capacitor em função do tempo durante o processo de carregamento. Segundo Halliday et al. (2013), esta função é dada pela equação 1:

$$V_C(t) = \varepsilon \left( 1 - e^{-\frac{t}{RC}} \right) \quad (1)$$

onde  $\mathcal{E}$  é a força eletromotriz da fonte,  $R$  é a resistência,  $C$  é a capacitância do capacitor, e  $t$  é o tempo. Esta equação mostra que a tensão no capacitor aumenta exponencialmente com o tempo até atingir o valor máximo de  $\mathcal{E}$ . O comportamento gráfico de  $V_c$  pode ser visto na figura 3.

Figura 3 - Tensão no capacitor aumenta exponencialmente com o tempo



Fonte: ALEXANDER (2013)

### 3 METODOLOGIA

#### 3.1 EXPERIMENTAÇÃO E AVALIAÇÃO DAS BÁSCULAS

##### Introdução e Estratégia Experimental:

O objetivo principal desta fase do estudo foi avaliar a eficiência e precisão das básculas em diferentes condições operacionais. A metodologia foi planejada para abranger uma série de testes progressivos, iniciando com a avaliação da balança original, partindo para testes com revestimentos e, por fim, a construção de uma nova balança em último caso.

Em decorrência dos obstáculos identificados nesse estágio, tornou-se necessário a construção de novas básculas utilizando tecnologia de impressão 3D, com materiais como PLA e ABS.

Cada experimento foi planejado para explorar e comparar as características operacionais e de calibração de cada balança, permitindo uma análise abrangente de sua eficiência e precisão em diferentes condições.

##### Experimento 1 - Avaliação Detalhada da Balança Original:

No início do estudo, conduziu-se uma análise detalhada da balança original, construída em plástico PET. O principal objetivo desta fase foi avaliar a capacidade da balança de ser calibrada de maneira precisa e com valor consistente, um fator crucial para garantir a exatidão das medições. Para isso, estabelecemos critérios específicos de avaliação, focando na capacidade da balança de reter água após sua ascensão e na consistência das leituras obtidas. Estes critérios estão interligados e servem como indicadores da estabilidade e confiabilidade do equipamento em condições variadas de uso.

##### Experimento 2 - Implementação de Revestimentos Hidrorepelentes:

Diante de uma balança original fora de condições de operação, a pesquisa progride para a experimentação com revestimentos hidrorepelentes, especificamente o uso de uma manta de teflon. Após a aplicação deste revestimento na balança, a análise segue os mesmos critérios utilizados no Experimento 1. Além disso, incluímos um teste adicional para avaliar a aderência do revestimento de teflon na balança, a fim de assegurar a eficácia e durabilidade da intervenção.

### Experimento 3 - Desenvolvimento de uma Bâscula via Impressão 3D:

Caso os experimentos anteriores não atingirem os padrões de precisão e estabilidade desejados, a metodologia previa a construção de uma nova balsa utilizando tecnologia de impressao 3D.

Nesta fase, dois materiais - ABS e PLA - foram selecionados para a fabricacao, por serem materiais mais comumente utilizados na impressao 3D. A analise comparativa entre os materiais focou-se na eficiencia operacional, precisao da calibracao e durabilidade a longo prazo.

Com o intuito de minimizar a aderencia da agua nas superficies internas da balsa, uma estrategia adicional foi implementada. Para a peca de ABS, realizou-se um acabamento com vapor de acetona, enquanto para a peca em PLA, foi efetuado um acabamento em cloroformio. Essas medidas visam otimizar o desempenho da balsa ao reduzir potenciais obstaculos decorrentes do processo de impressao 3D.

Ainda, em resposta as limitacoes termicas do PLA, destacadas por Södergard e Stolt (2002), com uma temperatura de aplicabilidade em torno de 60°C, realizou-se um teste de resistencia termica. A balsa de PLA foi colocada no painel de um carro sob sol direto, visando simular as altas temperaturas encontradas em um pluviometro e avaliar a resposta do material a essas condicoes extremas.

#### **3.1.1 Tratamento da balsa de PLA com cloroformio**

O processo de aplicacao de cloroformio, detalhado na figura 4, foi implementado como uma etapa essencial no procedimento de acabamento da peca. Esta tecnica foi utilizada a fim de atenuar as ranhuras superficiais inerentes ao processo de impressao 3D, otimizando as propriedades hidrodinamicas da peca, e facilitando assim o escoamento de agua sobre sua superficie.

Figura 4 - Tratamento da báscula em PLA com clorofórmio



Fonte: Elaboração própria (2023)

### 3.1.2 Tratamento da báscula de ABS com acetona

Para o tratamento da superfície da báscula em ABS, mostrado na figura 5, criou-se um ambiente fechado para tratar a báscula com acetona. Uma bandeja com papel absorvente embebido em acetona foi espalhada pela bandeja, sobre a qual a báscula foi posicionada, elevada por um béquer, de modo a não tocar o papel. Cobrindo tudo, outra bandeja foi colocada, formando uma câmara selada que permitia a volatilização e condensação da acetona nas paredes da báscula, sem perda de vapores para o ambiente externo.

Figura 5 - Câmara de acetona utilizada no tratamento da báscula



Fonte: Elaboração própria (2023)

### 3.2 PROCESSO DE CALIBRAÇÃO DO PLUVIÔMETRO

A calibração do pluviômetro, com foco na báscula escolhida como mais adequada, constituiu uma etapa fundamental neste estudo, tendo como objetivo principal assegurar a precisão na medição de volumes de precipitação. O procedimento de calibração foi projetado para garantir uma resposta precisa da báscula a um volume específico de água.

Na metodologia adotada, utilizou-se uma pipeta de 20 ml graduada, selecionada por sua elevada precisão e confiabilidade. O procedimento consistiu em adicionar água de forma controlada na báscula, com o objetivo de alcançar um volume específico de 9,7 ml, como pode ser observado na figura 6. Esse volume foi calculado para simular 0,3 milímetros de precipitação, levando em conta as dimensões da área de coleta do pluviômetro, tendo diâmetro de 8 polegadas da seção de coleta.

Figura 6 - Calibração da báscula

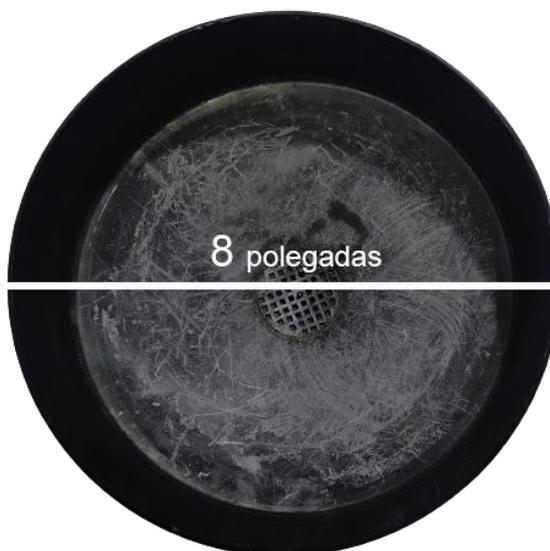


Fonte: Elaboração própria (2023)

Ajustes finos foram feitos nos parafusos da báscula: abaixando-os para aumentar a capacidade de retenção de água, e elevando-os para diminuí-la. Esta abordagem assegurou que cada tombamento da báscula representasse quase que exatamente a quantidade de chuva pretendida, um elemento crucial para a precisão das medições.

Na figura 7, a superfície superior do pluviômetro é exibida, destacando a área de coleta de chuva. Esta área é essencial para assegurar que cada tombamento da báscula corresponda precisamente à quantidade de chuva indicada.

Figura 7 - Vista da parte superior do pluviômetro, evidenciando a área de coleta



Fonte: Elaboração própria (2023)

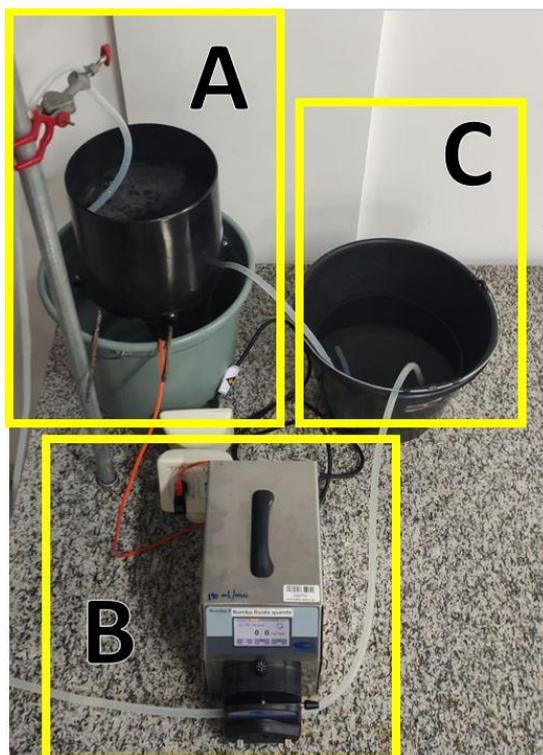
O processo de calibração do equipamento foi efetuado por meio de ajustes no parafuso de calibração da balança. Se a balança fosse ativada com um volume inferior ao volume de água definido, o parafuso de calibração localizado no lado oposto era ajustado para baixo, aumentando assim a sensibilidade do dispositivo. Por outro lado, se a balança suportasse um volume maior que o volume definido sem alterar sua posição, o parafuso no lado oposto era ajustado para cima.

### **3.2.1 Testes de Operação: Avaliação de Erro em Diferentes Intensidades de Chuva**

Utilizando uma bomba peristáltica Solab SI d64, disponível no Laboratório de Ensino de Engenharia do ICT, simulamos diferentes intensidades de chuva para testar a precisão de um pluviômetro. As intensidades foram ajustadas para corresponder a 16,21 mL/min (30 mm/h), 32,42 mL/min (60 mm/h) e 48,63 mL/min (90 mm/h), de acordo com as dimensões do aparelho. Essa configuração nos permitiu testar a balança do pluviômetro em três diferentes intensidades de chuva: 30 mm/h, 60 mm/h e 90 mm/h.

Em nossa configuração experimental, descrita na figura 8, a área "A" representa o pluviômetro e o recipiente de coleta, a área "B" indica a localização da bomba peristáltica, e a área "C" abriga o recipiente com água, servindo de fonte para a bomba. Uma mangueira conecta os recipientes das áreas "A" e "C", mantendo o mesmo nível de água.

Figura 8 - Configuração experimental



Fonte: Elaboração própria (2023)

Considerando a superioridade da precisão da bomba peristáltica em comparação com um pluviômetro convencional, os valores de chuva simulados pela bomba foram tratados como os valores verdadeiros nas análises. Ao final dos testes, empregamos um método para estimar as incertezas associadas ao equipamento, fundamentado em pesquisas e diretrizes estabelecidas.

Para a análise de erros e cálculo da incerteza total, seguimos as orientações de Vuolo (1996). O erro sistemático foi determinado pela diferença entre o valor verdadeiro (aferido pela bomba peristáltica) e a média das aferições do pluviômetro. Já o erro aleatório, considerando o limite de erro com 95% de confiança, foi

estabelecido como duas vezes o desvio padrão ( $2\sigma$ ) das medições, conforme a prática comum na descrição técnica de instrumentos.

Segundo Vuolo (1996), a incerteza total, descrita pela equação 2, permite uma avaliação precisa das incertezas, integrando tanto os erros sistemáticos quanto os aleatórios, essencial para assegurar a confiabilidade dos dados coletados sob diversas condições de intensidade de chuva.

$$\text{Incerteza Total} = \sqrt{(\text{Erro Sistemático})^2 + (\text{Erro Aleatório})^2} \quad (2)$$

Após a conclusão dos testes, considerando o contexto apresentado, foi desenvolvido um método para estimar as incertezas associadas ao equipamento. Este método se baseia nas pesquisas realizadas por Alena et al. (1990), Marsalek (1981) e as diretrizes da Organização Meteorológica Mundial (1983), que coletivamente indicam que ocorre um aumento no erro de subestimação em instrumentos de medição à medida que a intensidade da chuva se intensifica. Assim, para garantir maior precisão e confiabilidade nas medições, torna-se prudente adotar faixas de erro máximo correspondentes a diferentes faixas de intensidade de chuva, calculadas utilizando a equação x.

### 3.3 CIRCUITO

#### 3.3.1 Estrutura e Componentes do Circuito

A construção do circuito para o sistema de pluviômetro foi uma etapa crucial do projeto, desempenhando um papel fundamental na coleta, armazenamento e disponibilização de dados relacionados à chuva, umidade, temperatura e outros parâmetros. O circuito foi projetado em torno do microcontrolador ESP8266, conhecido por sua versatilidade e capacidade de comunicação, integrando-o ao reed

switch, que mede a chuva contando os pulsos gerados por um pêndulo com um ímã, e a outros módulos e sensores.

O armazenamento dos dados é feito em um cartão micro SDHC de 32GB, através de um módulo SD card genérico de 3V, onde os dados são registrados em um documento .txt. Este método oferece uma solução eficaz e prática para o armazenamento de grandes volumes de dados.

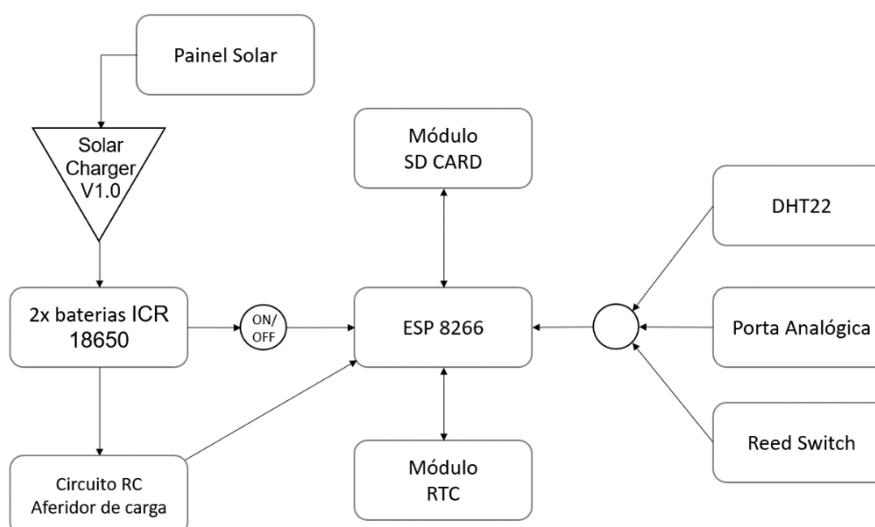
Para garantir a precisão do tempo nos registros, foi incorporado um RTC (Real Time Clock) PCF8563. Esta inclusão é vital, especialmente em casos de reinicialização do sistema, para assegurar a continuidade e precisão do registro temporal.

Foi incrementada uma conexão adicional para o sensor DHT22. Este sensor pode ser integrado facilmente e ativado por meio da interface do usuário. Existe, adicionalmente, uma porta analógica acessível ao usuário. Esta porta é compatível com uma conexão P2 fêmea e é projetada para a conexão e configuração de sensores extras, contribuindo para a flexibilidade do sistema.

Um elemento crucial do circuito é o módulo Solar Charger V1.0. Este módulo foi projetado para receber energia de um painel solar com capacidade de 6V. Ele é utilizado para carregar um conjunto de baterias de 3.7V, que fornece energia ao sistema. Na ausência de geração solar, o módulo permite que as baterias assumam o fornecimento de energia do circuito. Alternativamente, o módulo permite uma fonte de alimentação convencional de até 6V que pode ser utilizada em substituição ao painel solar.

O esquema em blocos do circuito pode ser visualizado na figura 9.

Figura 9 - Esquema em blocos do circuito



Fonte: Elaboração própria (2023)

O diagrama esquemático do circuito, com mais detalhes e especificações está disponível no anexo A.

### 3.4 MÓDULO DE BATERIAS E CIRCUITO PARA AFERIÇÃO DE NÍVEL DE CARGA

Na concepção do módulo de bateria para o projeto em questão, adotou-se a premissa de que a capacidade da bateria deveria ser suficiente para manter o circuito operacional por um período de três dias sem necessidade de recarga. Com esse objetivo, selecionaram-se duas baterias ICR 18650, cada uma com capacidade de 2600mAh. Conectadas em paralelo, de forma que funcionem como uma única célula de 5200mAh.

Para proteger e gerenciar a bateria adequadamente, incluiu-se um sistema de gerenciamento de bateria (BMS) genérico de 1S, regulando a tensão nominal em 3.7V, com limites de 2.9V e 4.2V para a tensão mínima e máxima, respectivamente.

### 3.4.1 Sistema de Medição de Carga por circuito RC

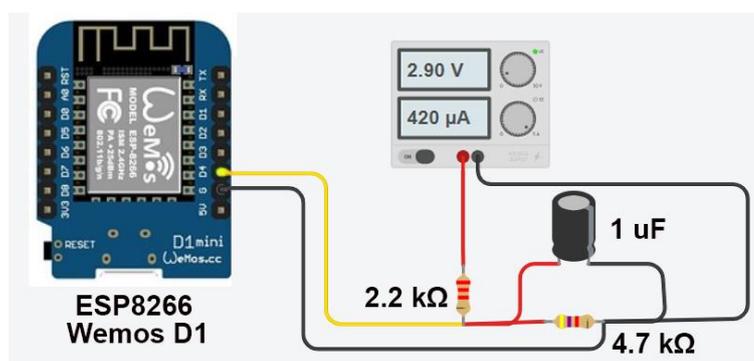
A questão da medição da tensão da bateria foi abordada com um enfoque inovador devido à limitação de apenas uma porta analógica disponível no microcontrolador utilizado. Para contornar essa limitação e proporcionar uma estimativa confiável do nível da bateria, desenvolveu-se um método alternativo baseado no uso de um circuito RC (Resistor-Capacitor).

Este circuito não tinha por finalidade realizar medições precisas de tensão, mas sim fornecer uma estimativa aproximada do nível da bateria. O princípio por trás desse método era estimar a tensão da bateria pelo tempo necessário para que uma porta digital atingisse uma leitura 'HIGH', uma vez conectada ao terminal positivo de um capacitor.

A escolha criteriosa dos valores de resistores e capacitores foi fundamental para assegurar que as medições fossem realizadas na região aproximadamente linear da função de tensão do capacitor, que é bem aparente na figura 3, que mostra o comportamento da tensão sobre um capacitor em carregamento.

Com isso, ao medir o tempo até que a porta digital GPIO2 indicasse um sinal 'HIGH', no circuito da figura 10, sob diferentes níveis de tensão (entre 2.9V e 4.2V), foi possível realizar uma análise de regressão linear utilizando o Excel. Essa análise resultou em uma função que estima o nível de tensão na bateria com base no tempo de saturação da porta GPIO2(D4), oferecendo uma solução eficiente e prática para o monitoramento do estado da bateria, essencial para a gestão de energia do módulo desenvolvido.

Figura 10 - Sistema de medição de carga por circuito RC



Fonte: Elaboração própria (2023)

### 3.4.2 Construção de PCB para suporte ao circuito

Para o projeto, como o circuito é complexo e sensível a mau contato, a construção de uma Placa de Circuito Impresso (PCB) é essencial por diversos motivos. Uma PCB proporciona uma base física robusta e confiável para o circuito, assegurando que todos os componentes estejam seguramente fixados e protegidos contra movimentos e vibrações que o equipamento pode sofrer em campo que podem causar mau contato. Isso é crucial em circuitos complexos, onde a interconexão precisa de múltiplos componentes precisa ser gerenciada de forma eficaz.

O design de uma PCB permite uma organização precisa e uma distribuição otimizada dos componentes e trilhas de conexão, reduzindo a possibilidade de curtos-circuitos ou falhas devido a contatos acidentais. Isso é fundamental para o desempenho e a confiabilidade do circuito, especialmente em aplicações onde a precisão e a estabilidade são vitais.

Além disso, a capacidade de replicar o design de forma consistente é uma vantagem significativa. Em circuitos complexos, qualquer variação ou erro na montagem pode levar a falhas.

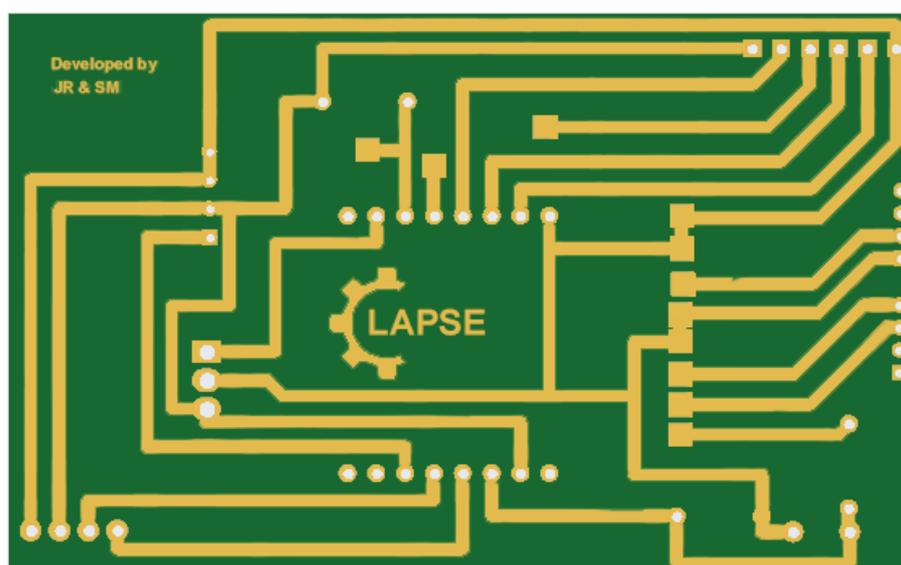
### 3.4.3 Fabricação da PCB

A construção da PCB seguiu os seguintes passos e sua sketch está representada na figura 11:

- a) Seleção e Preparação da Placa de Fenolite:
  - Escolha de uma placa de fenolite com as dimensões e características adequadas para o circuito.
  - Limpeza cuidadosa da superfície da placa para remover impurezas ou oleosidade.
- b) Design do Circuito:

- Utilização de um software especializado para criar o layout do circuito, detalhando a localização de todos os componentes e trilhas.
- c) Impressão por Sublimação:
- Envio do design do circuito para uma empresa especializada em impressão por sublimação.
  - Transferência precisa do layout para a placa de fenolite, onde a tinta protege as áreas que se tornarão as trilhas.
- d) Corrosão com Percloroeto de Ferro:
- Aplicação de percloroeto de ferro sobre a placa para corroer o cobre exposto, isto é, as áreas não protegidas pela tinta.
  - Formação das trilhas do circuito conforme o layout projetado.
- e) Limpeza Pós-Corrosão:
- Limpeza da placa para remover resíduos do processo de corrosão e tinta remanescente, revelando as trilhas de cobre.
- f) Perfuração para Componentes:
- Identificação dos pontos de perfuração conforme o layout do circuito.
  - Perfuração utilizando uma broca de precisão para inserção dos componentes eletrônicos para soldagem.

Figura 11 - Sketch do circuito da PCB



Fonte: Elaboração própria (2023)

### 3.5 SENSORES

#### 1) DHT22

Para compreender a funcionalidade e limitações do sensor digital DHT22, que é um sensor para a aferição de umidade relativa do ar e temperatura com precisão, foram realizados testes preliminares. Estes testes consistiram na análise da correta comunicação entre o sensor e o ESP8266 e funcionamento nos registros das leituras

É importante ressaltar que o DHT22 é considerado uma opção do usuário, não integrando diretamente o escopo deste trabalho. Embora tenham sido realizados testes preliminares para fornecer informações adicionais, a inclusão ou exclusão deste sensor dependerá das necessidades específicas do usuário e das metas do projeto em questão. Portanto, qualquer discussão subsequente sobre o DHT22 deve ser interpretada dentro desse contexto, reconhecendo sua natureza opcional no âmbito da presente pesquisa.

#### 2) Sensor Analógico

O sistema foi enriquecido com um recurso adicional, uma porta analógica, que não é essencial para o funcionamento principal do sistema, mas oferece ao usuário maior versatilidade. Esta porta permite a configuração e calibração de sensores analógicos conforme as necessidades específicas do usuário. Durante os testes preliminares, a funcionalidade adicional foi explorada através da avaliação de um sensor capacitivo de umidade do solo, especificamente o modelo Moisture V1.2. O objetivo desses testes era verificar a precisão das leituras do sensor em variadas condições ambientais, destacando assim a utilidade prática deste recurso adicional no sistema.

### 3.6 ESTRATÉGIA COMUNICAÇÃO DE DADOS E TESTES DE CONECTIVIDADE

A comunicação de dados deve ser realizada por meio de uma página HTML criada pelo ESP, que se encontra configurado como um Access Point. Essa configuração permite ao ESP estabelecer uma rede WiFi dedicada para a transmissão de dados.

Ao conectar-se a essa rede, deve ser possível o acesso à página HTML, que desempenha o papel de interface principal para a comunicação de dados. Nessa interface, as informações são apresentadas de maneira organizada, permitindo ao usuário interagir de forma intuitiva com o sistema para configurar, coletar e visualizar os dados pertinentes.

Na condução dos testes de conectividade, foi avaliado a estabilidade e intensidade do sinal em diversas circunstâncias. A escolha deliberada de locais diversos, abrangendo ambientes internos e externos, com presença ou ausência de obstáculos como paredes e janelas, visa proporcionar uma análise abrangente do desempenho da rede.

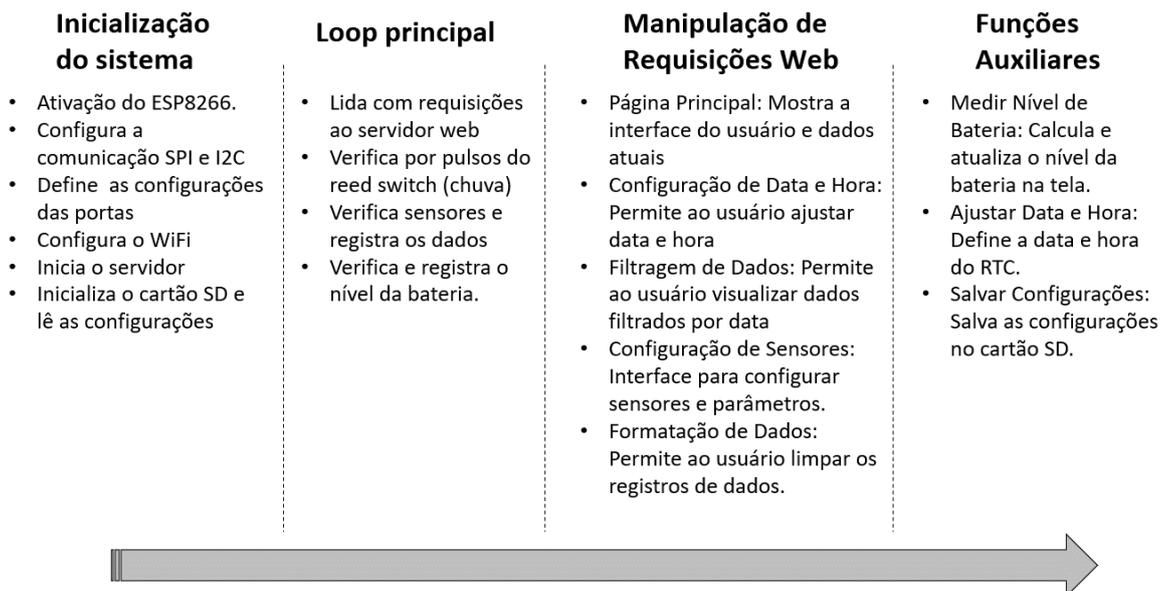
Durante a execução dos testes, foram registradas leituras do sinal em diferentes distâncias entre o dispositivo e o ponto de acesso WiFi, fornecendo insights sobre a variação da qualidade da conexão em função da proximidade.

Na avaliação do desempenho do equipamento, definiu-se que um resultado satisfatório seria caracterizado pela capacidade do usuário de manter uma conexão estável e confiável com a rede Wi-Fi, mesmo sob condições pouco favoráveis, como a presença de obstáculos físicos ou distância do ponto de acesso. Esta estabilidade na conexão é considerada essencial para assegurar a eficácia na coleta e no acesso ao relatório dos dados registrados pelo equipamento. Em contraste, problemas como interrupções frequentes na conexão ou um sinal fraco de Wi-Fi foram estabelecidos como indicativos de um desempenho inadequado. Estes parâmetros fornecem um critério objetivo e claro para aferir a qualidade da conectividade do equipamento em diferentes cenários de uso.

### 3.7 PROGRAMAÇÃO DO SISTEMA

A programação segue o algoritmo desenvolvido na figura 12 e encontra-se no anexo B.

Figura 12 -Algoritmo do código



Fonte: Elaboração própria (2023)

## 4 RESULTADOS E DISCUSSÕES

Com o objetivo de aprofundar a análise do desempenho do sistema desenvolvido e assegurar sua eficácia, procedemos, antes de realizar testes no sistema como um todo, a avaliações individuais e específicas em cada módulo desenvolvido. Os subitens a seguir abordam os testes relacionados à: Construção da balança, calibração do pluviômetro, sistema de comunicação de dados, testes de conectividade, sistema de baterias, testes de autonomia do sistema, sistema adicional - DHT22, testes com o DHT22, sistema adicional - entrada analógica disponível, teste da entrada analógica. Após verificar e garantir o adequado desempenho de cada módulo, conduzimos o teste em campo com o pluviômetro.

### 4.1 BÂSCULA

#### 4.1.1 Análise Crítica dos Problemas com a Balança Original

Diante da balança original, houve uma tentativa de calibração do dispositivo. No entanto, durante o processo de calibração, foram observados problemas na estrutura da balança que impediram uma calibração satisfatória. Diante dessa situação, julgou-se necessário realizar modificações na estrutura da balança para garantir a precisão e a eficácia do pluviômetro. Essas alterações são fundamentais para assegurar que o pluviômetro funcione corretamente e fornecer leituras precisas da precipitação.

##### 4.1.1.1 Deterioração da Superfície Interna e Complicações na Retenção de Água

A inspeção da balança original, construída em plástico PET, revelou uma deterioração significativa em sua superfície interna, marcada por arranhões profundos e extensos. Esta condição, enfatizada na edição da figura 13, era não

apenas um indicativo de desgaste, mas também uma causa direta para o comportamento irregular do escoamento da água. As áreas riscadas criaram zonas de aderência variável, onde a água, em vez de escoar uniformemente, apresentava retenção parcial. Durante o processo de inversão da bscula, crucial para as medies, a gua frequentemente no escoava por completo, ocasionando variaes aleatrias e indesejadas nos resultados.

Figura 13 - Pluvimetro aberto evidenciando reteno parcial da gua durante o escoamento



Fonte: Elaborao prpria (2023)

#### 4.1.1.2 Instabilidade Estrutural e Problemas de Calibrao

Alm da superfcie interna danificada, a leveza da bscula original, trazia desafios adicionais. Quando a gua se aproximava do volume necessrio para a inverso da bscula, sua estrutura leve era sensvel at as mais leves trepidaes, o que causava instabilidade. Esta sensibilidade excessiva tornava os ajustes de calibrao extremamente difceis, comprometendo a preciso da medio.

#### 4.1.1.3 Inconsistncias dos Parafusos de Calibrao

Os parafusos de ajuste, localizados nas bordas da bscula e ilustrados na figura 14, foram inicialmente projetados para auxiliar no processo de calibrao do dispositivo. Contudo, durante a utilizao, identificou-se uma falha tcnica significativa nesses componentes. Os parafusos demonstraram uma tendncia a acumular gua em suas cabeas. Esta reteno hdrica entre o parafuso e a superfcie externa da bscula gerou um efeito de aderncia no intencional,

afetando diretamente a dinâmica operacional do dispositivo. Especificamente, a tensão superficial da água atuava como uma força de retenção, atrapalhando o movimento oscilatório da balança e, conseqüentemente, comprometendo a precisão da medição. Este comportamento inconsistente e imprevisível dos parafusos introduziu uma variável aleatória e indesejada no sistema de medição, desafiando a confiabilidade e a precisão do processo de calibração.

Figura 14 - Pluviômetro aberto destacando parafusos de calibração



Fonte: Elaboração própria (2023)

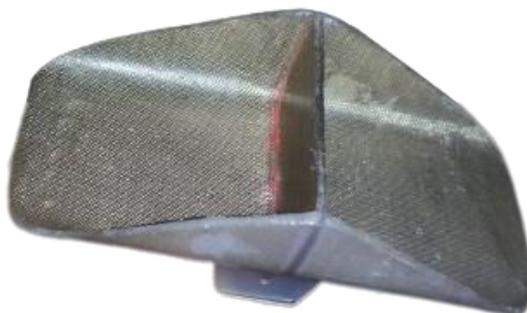
#### **4.1.2 Tentativas de Melhoria e Decisão pela Construção de uma Nova Balança**

##### **4.1.2.1 Tentativa de revestimento com teflon**

Como uma primeira medida para superar os problemas identificados na balança original, a equipe optou por revestir a superfície interna com uma manta de Teflon. Esse material foi escolhido por suas reconhecidas propriedades hidrorrepelentes, com a expectativa de que ele minimizaria a retenção de água na superfície. A fixação da manta de Teflon foi realizada com cola à base de cianoacrilato, um adesivo conhecido por sua forte adesão e rápida polimerização.

Na figura 15, a balança original é mostrada com o revestimento de Teflon aplicado. O revestimento tinha como objetivo proporcionar um escoamento de água mais uniforme, compensando as irregularidades da superfície interna da balança.

Figura 15 - Balança original com revestimento de teflon



Fonte: Elaboração própria (2023)

#### 4.1.2.2 Problemas de Aderência da manta de Teflon e Decisão pela Nova Construção

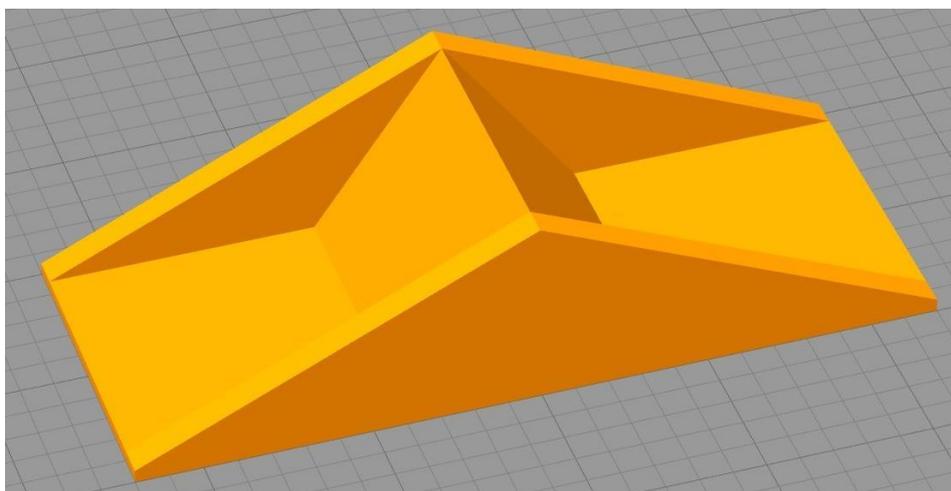
Apesar da aplicação bem-sucedida da manta de Teflon, testes subsequentes revelaram um problema significativo de aderência. Ao puxar a manta, constatou-se que ela se descolava facilmente da superfície da balança, demonstrando uma baixa adesão entre os materiais. Esse resultado evidenciou que a solução com Teflon não seria viável a longo prazo, levando à decisão de descartar a balança original e construir uma nova utilizando a tecnologia de impressão 3D.

### 4.1.3 Desenvolvimento e Aperfeiçoamento da Nova Báscula em Impressão 3D

#### 4.1.3.1 Design e Prototipagem Inicial em PLA

O redesign da báscula no Tinkercad não foi apenas uma questão de ajuste dimensional, mas uma resposta reflexiva às falhas operacionais da versão anterior. Ao aumentar o comprimento da báscula, como mostrado na figura 16, visando um fluxo de água mais eficiente e uma redução significativa nos problemas associados aos parafusos de calibração.

Figura 16 - Projeto 3D da nova báscula no Tinkercad



Fonte: Elaboração própria (2023)

A figura destaca o modelo 3D da báscula no Tinkercad, onde as alterações dimensionais visam superar desafios específicos identificados na versão anterior.

A confecção inicial do protótipo da báscula foi realizada empregando o polímero Ácido Polilático (PLA) e o tratamento com clorofórmio, o resultado é apresentado na figura 17.

Figura 17 - Protótipo inicial da balança em PLA



Fonte: Elaboração própria (2023)

Contudo, durante a fase de teste, emergiram desafios específicos relacionados à sua interação com a água. Durante ensaios foi observado absorção de umidade pelo material, que poderiam levar a degradação em ambiente operacional, esses são fatores que podem impactar significativamente a durabilidade do componente impresso quando exposto a condições úmidas.

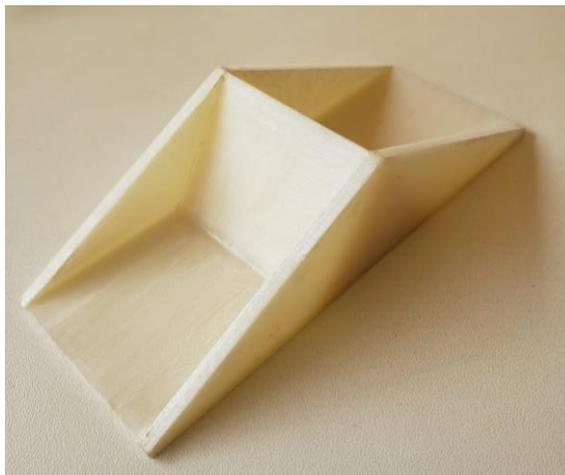
#### 4.1.3.2 Transição para ABS e Tratamento com Acetona

Os testes realizados na balança de PLA revelaram sua robustez e precisão. No entanto, ao submetê-la a condições de temperaturas um pouco mais elevadas, o material apresentou certa deformação devido às características físicas do PLA. Diante desse cenário, uma adaptação foi necessária, resultando na fabricação da balança em ABS.

#### 4.1.3.3 Prototipagem da balança em ABS

A decisão de transicionar para uma balança fabricada em Acrilonitrila Butadieno Estireno (ABS), evidenciada na figura 18, representou um ponto crítico no desenvolvimento do projeto, motivada pelas observações e limitações identificadas no uso do PLA.

Figura 18 - Bscula impressa em ABS



Fonte: Elaborao prpria (2023)

A figura da bscula em ABS, simboliza uma mudana significativa na abordagem do projeto, com foco na durabilidade e estabilidade do material.

#### 4.1.3.4 Tratamento da bscula de ABS com acetona

O resultado deste tratamento em ABS  apresentado na figura 19. A superfcie tratada da bscula mostra uma melhoria significativa, destacando a eficcia do processo de acetona em criar uma superfcie mais lisa e uniforme, crucial para a eficincia operacional da bscula, reduzindo os erros na medio.

Figura 19 - Bscula em ABS aps o tratamento com acetona



Fonte: Elaborao prpria (2023)

## 4.2 CALIBRAÇÃO DO PLUVIÔMETRO

### 4.2.1 Calibração da Nova Bâscula em ABS

#### 4.2.1.1 Processo de Calibração para Medição de Chuva

A calibração da nova bscula em ABS foi realizada atravs do processo descrito na seo 3.2. Utilizando uma pipeta de 20 mL, a gua foi adicionada gota a gota, ajustando cada lado da bscula para suportar precisamente 9,7 mL e o ajuste fino da bscula conseguiu ser alcanado, com a bscula suportando quase que exatamente, e nada a mais, que o valor definido.

### 4.2.2 Avaliao do Erro do Instrumento

#### 4.2.2.1 Avaliao de Erro em Diferentes Intensidades de Chuva

Aps a calibrao da bscula em ABS, foi essencial avaliar o erro do instrumento, especialmente considerando as alteraes na estrutura do pluvimetro.

Diante das trs diferentes intensidades de chuva: 30 mm/h, 60 mm/h e 90 mm/h, os dados coletados podem ser vistos na Figura 20, gerada pelo prprio sistema do pluvimetro.

Figura 20 - Dados coletados pela interface do sistema

**Leituras para 11/2023**

20/11/2023 às 1h foi registrado 28.8000mm de chuva  
 20/11/2023 às 2h foi registrado 28.5000mm de chuva  
 20/11/2023 às 3h foi registrado 28.5000mm de chuva  
 20/11/2023 às 4h foi registrado 28.5000mm de chuva  
 20/11/2023 às 5h foi registrado 28.5000mm de chuva  
 20/11/2023 às 1h foi registrado 56.4000mm de chuva  
 21/11/2023 às 2h foi registrado 56.4000mm de chuva  
 21/11/2023 às 3h foi registrado 56.7000mm de chuva  
 21/11/2023 às 4h foi registrado 56.7000mm de chuva  
 21/11/2023 às 5h foi registrado 56.4000mm de chuva  
 22/11/2023 às 1h foi registrado 81.9000mm de chuva  
 22/11/2023 às 2h foi registrado 81.9000mm de chuva  
 22/11/2023 às 3h foi registrado 81.9000mm de chuva  
 22/11/2023 às 4h foi registrado 81.9000mm de chuva  
 22/11/2023 às 5h foi registrado 80.7000mm de chuva

Fonte: Elaboração própria (2023)

Agora com as leituras, através da abordagem estatística seguindo a equação podemos fazer a análise dos erros percentuais do instrumento, que pode ser visualizada na tabela 1.

Tabela 1 - Análise dos erros percentuais do instrumento

	Média das leituras (mm/h)	Limite de erro aleatório confiança de 95%	Erro sistemático	Erro total do instrumento	Erro relativo total
Vazão de 30mm/h	28,6	0,29	1,4	1,5	5%
Vazão de 60mm/h	56,5	0,33	3,5	3,5	6%
Vazão de 90mm/h	81,9	0,18	8,1	8,1	9%

Fonte: Elaboração própria (2023)

Conforme evidenciado na tabela 1, a análise quantitativa revela que a precisão das medições varia conforme a intensidade da chuva. Para uma intensidade de 30mm/h, o erro estimado foi de aproximadamente 5%. Já para a intensidade de 60mm/h, o erro observado aumentou para cerca de 6%, e para a intensidade de 90mm/h, o erro estimado ascendeu a 9%. Estes dados indicam uma correlação entre o aumento da intensidade da chuva e a elevação do percentual de erro nas medições.

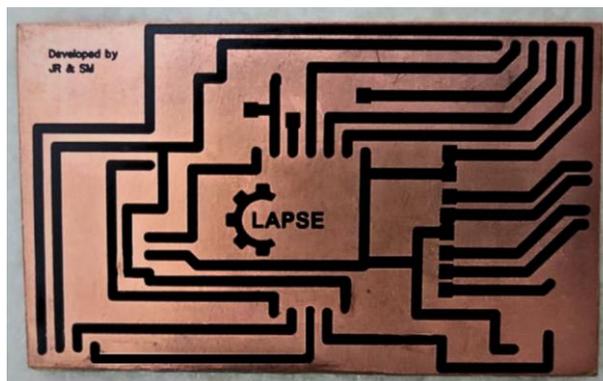
#### 4.3 CIRCUITO E SISTEMA DE REGISTRO DE DADOS

##### 4.3.1 Placa De Circuito Impresso

Durante o desenvolvimento do projeto, uma etapa crucial foi a produção das placas de fenolite, essenciais para o circuito do dispositivo. As placas de fenolite foram inicialmente limpas e, em seguida, submetidas a um processo de estampa por sublimação em uma empresa especializada. O custo para este processo foi de R\$ 5,00 por placa.

A qualidade da pintura nas trilhas, que pode ser observada na figura 21, das placas estampadas demonstrou ser de alto padrão. Essa característica de qualidade foi fundamental, pois assegurou a integridade das trilhas após o processo de corrosão com percloroato de ferro. A presença de uma pintura de qualidade nas trilhas é um aspecto vital para evitar falhas, garantindo a eficiência e a confiabilidade dos circuitos nas condições operacionais pretendidas.

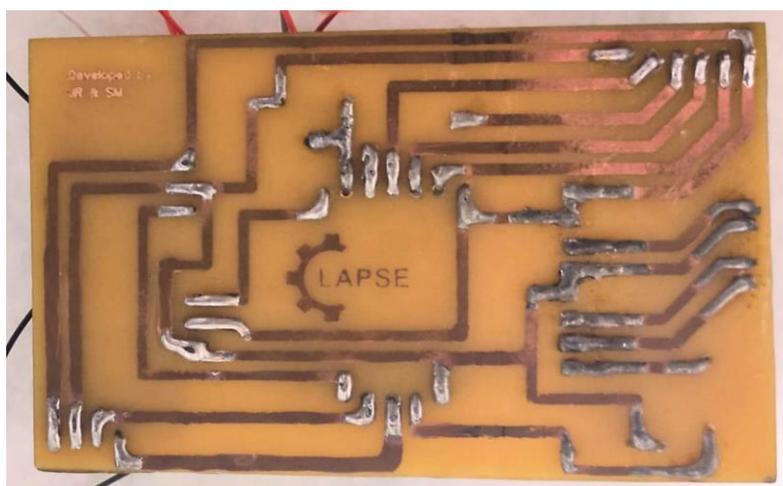
Figura 21 - PCB com o desenho das trilhas antes da corrosão



Fonte: Elaboração própria (2023)

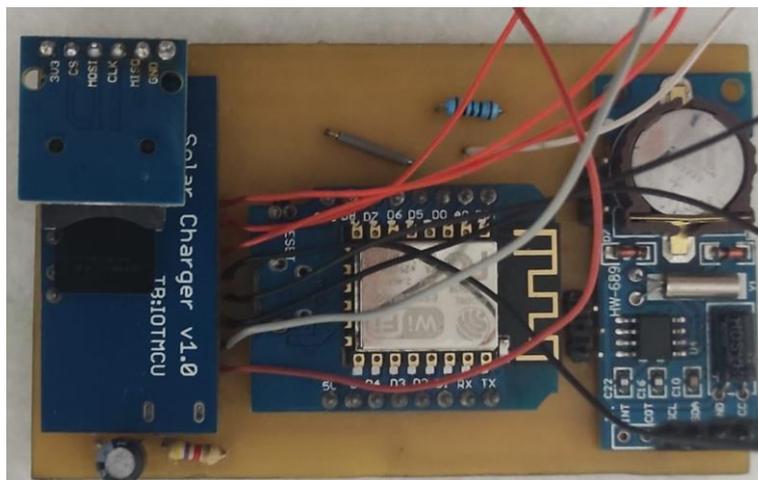
Após a impressão das trilhas nas placas de fenolite, o próximo passo crucial foi a corrosão com perclorato de ferro. Com as trilhas corretamente definidas, prosseguimos com a etapa de perfuração da placa. Esta etapa foi realizada com precisão, criando orifícios nos locais apropriados para a inserção dos componentes eletrônicos. E finalmente, os componentes eletrônicos foram cuidadosamente soldados na placa que pode ser visualizada na figura 22 com a vista das trilhas e na figura 23 com a vistas da dos componentes.

Figura 22 - Vista inferior da PCB após a corrosão com os componentes soldados



Fonte: Elaboração própria (2023)

Figura 23 - Vista dos componentes soldados na PCB



Fonte: Elaboração própria (2023)

#### 4.3.2 Sistema de Baterias

O módulo de bateria é composto por duas células ICR18650 da Samsung (2600mah), monitoradas pelo BMS 1S. O módulo da bateria construído, incluindo o BMS 1S, é ilustrado na figura 24.

Figura 24 - Módulo de bateria construído com BMS 1S



Fonte: Elaboração própria (2023)

A figura mostra o módulo de bateria montado, incluindo as células ICR18650 e o bms1S, uma configuração essencial para garantir uma alimentação estável e segura do sistema.

Para carregamento das baterias, é possível o uso de um painel solar ou fonte de até 6v em um conector do tipo jack logo abaixo do instrumento, como pode ser visto em destaque na figura 25.

Figura 25 - Conector Jack fixado embaixo do equipamento para entrada de energia



Fonte: Elaboração própria (2023)

#### 4.3.3 Sistema de Medição de Carga por circuito RC

Um aspecto inovador do circuito é o uso de um circuito resistor-capacitor para medir a carga da bateria. Este sistema mede o tempo necessário para que a tensão na base do capacitor atinja um nível que sature uma porta digital do ESP8266, enviando um sinal "HIGH" por GPIO2. Através de testes em laboratório, com tensões variando entre 2.9V e 4.2V em 10 intervalos iguais. Posteriormente uma regressão linear foi utilizada, como pode ser observado na tabela 2, para estabelecer uma função que relaciona a média dos tempos de 5 ensaios com a tensão regulada na

fonte. Essa relação permite uma estimativa aproximada da porcentagem de carga da bateria, uma solução prática para monitorar o estado da bateria sem a necessidade de componentes adicionais.

Tabela 2 - Dados experimentais do tempo de detecção de sinal high pelo ESP8266 com base na tensão de alimentação do circuito RC

tempo (ms)	Tensão da fonte (V)
1457,2	2,90
1356,4	3,03
1263,4	3,16
1185	3,29
1119,2	3,42
1061,2	3,55
1008	3,68
958,6	3,81
913	3,94
874,2	4,07
837,4	4,20

Fonte: Elaboração própria (2023)

Dessa forma, através de uma regressão linear feita utilizando o software Excel, foi obtida uma função que retorna o nível da bateria, em porcentagem, em função do tempo, a função é dada pela equação 3:

$$\%bateria = -181,1 * \ln(t) + 1314,4 \quad (3)$$

Embora o sistema de medição da carga da bateria não seja o foco principal do projeto, ele oferece uma visão valiosa sobre o comportamento da bateria ao longo do tempo.

Ao usar um multímetro, foi possível verificar que o sistema forneceu uma boa aproximação da carga da bateria, destacando a eficácia do design do circuito em fornecer informações úteis sobre o estado da bateria, mesmo com uma abordagem simplificada.

#### 4.3.4 Incrementação do DHT22

Para facilitar a integração deste sensor, foram providenciados conectores fêmeas localizados estrategicamente logo abaixo do pluviômetro, que podem ser vistos em destaque na figura 26 permitindo que o sensor seja conectado de maneira simples e eficiente.

Figura 26 – Conector fêmea para conexão do DHT22



Fonte: Elaboração própria (2023)

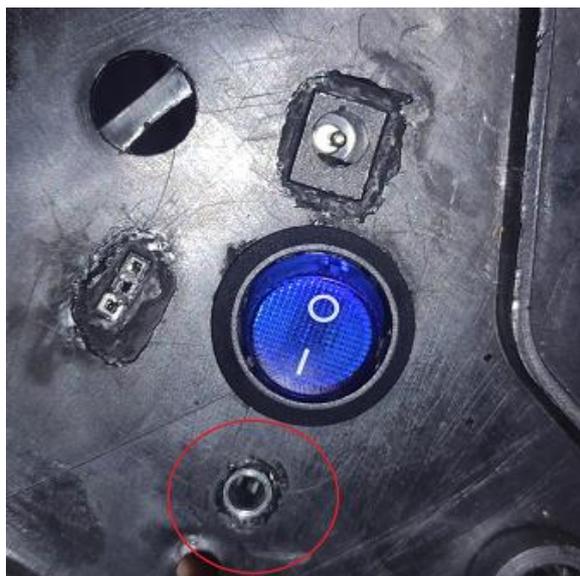
A decisão de tornar a inclusão do sensor opcional e de oferecer uma interface de conexão fácil de usar reflete um compromisso com a adaptabilidade e a personalização do equipamento, permitindo aos usuários configurar o pluviômetro de acordo com as necessidades específicas de seus projetos ou pesquisas.

#### 4.3.5 Sistema adicional – Entrada Analógica

Aqui destacamos a bem-sucedida implementação de um conector P2 fêmea na parte inferior do pluviômetro, como pode ser visto em destaque na figura 27. Essa adição possibilitou a disponibilização da porta analógica do ESP8266, juntamente

com um VCC de 3V e um GND, ao usuário. A facilidade de acesso a esses pontos de conexão permite uma integração mais simples e eficaz de componentes externos, como sensores e outros dispositivos, ampliando as capacidades de monitoramento e pesquisa do pluviômetro.

Figura 27 - Entrada P2 fêmea para adição de um sensor analógico



Fonte: Elaboração própria (2023)

Para os testes foi utilizado o sensor analógico Moisture V1.2, e foi observado que as leituras e gravações ocorreram da maneira esperada.

#### 4.4 SISTEMA DO PLUVIÔMETRO

##### 4.4.1 Sistema de registro e comunicação de dados

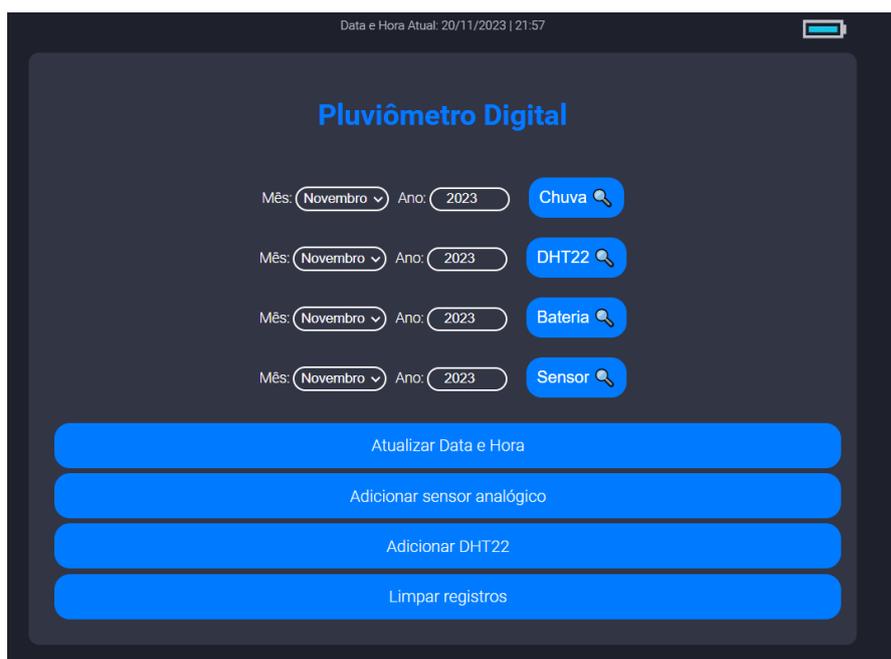
O sistema de monitoramento climático desenvolvido apresenta uma interface de usuário acessível e intuitiva, que pode ser alcançada através de um simples procedimento de navegação web. Para acessar a página principal do sistema, o usuário deve estar conectado à rede Wi-Fi denominada "Pluviômetro". Uma vez estabelecida a conexão, a página pode ser acessada digitando o endereço

pluviometro.local em qualquer navegador de preferência do usuário. Este método de acesso foi projetado para garantir facilidade e flexibilidade, permitindo aos usuários interagir com o sistema em uma variedade de dispositivos, desde computadores pessoais a smartphones e tablets.

#### 4.4.1.1 Página Principal

A página principal da interface, que pode ser visualizada na figura 28, exibe um design intuitivo e informativo. No topo, a data e hora atual são mostradas, juntamente com um ícone indicativo do nível de carga da bateria. Abaixo, encontram-se formulários de filtragem que permitem ao usuário acessar dados específicos por mês e ano para chuvas, leituras do sensor DHT22, status da bateria e informações do sensor analógico opcional. Estes filtros funcionam buscando registros dentro dos arquivos .txt no cartão SDHC, exibindo apenas as leituras correspondentes ao mês especificado. Cada seleção de filtro redireciona o usuário para uma página detalhada, onde é possível visualizar todos os registros do mês escolhido e salvar os dados em formato PDF.

Figura 28 – Interface: Página principal



Fonte: Elaboração própria (2023)

#### 4.4.1.2 Funcionalidades Específicas

A página “Atualizar Data e Hora”, mostrada na figura 29 (Acessada pelo botão "Atualizar Data e Hora" na página principal):

- Interface de Configuração: Ao clicar no botão "Atualizar Data e Hora" na página principal, o usuário é direcionado para uma página onde pode ajustar a data e a hora do sistema.
- Sincronização com Relógio RTC: Uma vez inseridas e confirmadas as informações, o sistema sincroniza automaticamente a hora com o relógio RTC do circuito, garantindo precisão temporal contínua.

Figura 29 - Interface: Página "atualizar data e hora"

A imagem mostra a interface de usuário para ajustar a data e a hora. O título "Ajustar Data e Hora" está em azul. Abaixo dele, há campos de entrada para "Data" (formato dd/mm/aaaa) e "Hora" (formato --:--). Abaixo dos campos, há dois botões azuis: "Atualizar Data e Hora" e "Retornar à página principal".

Fonte: Elaboração própria (2023)

A página “Adicionar Sensor Analógico” mostrada na figura 30 (Acessada pelo botão "Adicionar Sensor Analógico" na página principal):

- Controle de Estado: Clicando neste botão, o usuário é levado para uma página que permite ativar ou desativar o sensor analógico, controlando assim sua operação.
- Configuração de Coleta e Calibração: Nesta mesma página, também é possível definir o intervalo entre as coletas e realizar a calibração do sensor, ajustando a leitura analógica com um coeficiente específico.

Figura 30 - Interface: Página "configurar sensor analógico"

**Configurar Sensor Analógico**

Estado atual do sensor: Ativado  
Função de Calibração:  $y = 1.00x + 0.00$

Ativar Sensor:

Parâmetro A:

Parâmetro B:

Intervalo entre coletas (horas):

Fonte: Elaboração própria (2023)

A página "Configurar DHT22" mostrada na figura 31 (Acessada pelo botão "Configurar DHT22" na página principal):

- Controle do Estado do Sensor: O botão "Configurar DHT22" na página principal conduz a uma página onde o usuário pode ligar ou desligar o sensor DHT22 e ajustar o intervalo entre as leituras, otimizando a coleta de dados.

Figura 31 - Interface: Página "configurar DHT"



**Configurar Sensor DHT**

Estado atual do sensor: Ativado  
Intervalo entre coletas: 1

Ativar Sensor:

Intervalo entre coletas (horas):

Fonte: Elaboração própria (2023)

A página "Limpar Registros" mostrada na figura 32 (Acessada pelo botão "Limpar Registros" na página principal):

- **Página de Confirmação:** Ao selecionar o botão "Limpar Registros", o usuário é redirecionado para uma página de confirmação, onde deve confirmar a intenção de excluir todos os registros históricos.
- **Exclusão de Dados:** Esta funcionalidade é crucial para manter a eficiência do armazenamento de dados e a integridade do sistema, permitindo uma limpeza completa do histórico de dados quando necessário.

Figura 32 - Interface: Página "limpar registros"



Fonte: Elaboração própria (2023)

## 4.5 TESTES DE CONECTIVIDADE

### 4.5.1 Avaliação da Comunicação sem Fio (Wi-Fi)

Durante os testes de comunicação sem fio do sistema utilizando o módulo ESP8266, foram observados resultados significativos em relação à estabilidade da conexão, autonomia da bateria e alcance efetivo do sinal Wi-Fi.

#### 4.5.1.1 Estabilidade da Conexão em Diferentes Condições Ambientais

O módulo ESP8266 demonstrou uma notável estabilidade de conexão. Mesmo testado na presença de obstáculos como paredes e janelas, a conectividade permaneceu robusta, permitindo a geração do relatório dos dados conectados. Isso sugere uma eficiente capacidade do sinal Wi-Fi em condições operacionais, o que é essencial para um sistema de monitoramento meteorológico operando em ambientes diversos.

#### 4.5.1.2 Consumo Vs Potência do Wi-Fi

Conforme apresentado na tabela 3, fica evidente o impacto da potência do Wi-Fi sobre o consumo de energia e a autonomia da bateria em sistemas que utilizam o módulo ESP. A tabela ilustra o aumento significativo no consumo de energia (atingindo 240mA) quando a potência do Wi-Fi está configurada no seu máximo de 19,5 dBm. Em contrapartida, a decisão de limitar a potência para 8.0 dBm demonstra uma redução considerável no consumo, resultando na melhora substancial da autonomia da bateria, mantendo um alcance efetivo de conexão testada em 12 metros.

Tabela 3 - Consumo de energia por potência da rede wifi

Parameters	Min	Typical	Max	Unit
TX802.11 b, CCK 11 Mbps, P <sub>OUT</sub> = +17 dBm	-	170	-	mA
TX802.11 g, OFDM 54Mbps, P <sub>OUT</sub> = +15 dBm	-	140	-	mA
TX802.11 n, MCS7, P <sub>OUT</sub> = +13 dBm	-	120	-	mA
Rx802.11 b, 1024 bytes packet length, -80 dBm	-	50	-	mA
Rx802.11 g, 1024 bytes packet length, -70 dBm	-	56	-	mA
Rx802.11 n, 1024 bytes packet length, -65 dBm	-	56	-	mA

Fonte: Espressif Systems (2015).

Medindo com multímetro na escala de corrente, foi aferido que o consumo de 80 mA de todo o conjunto em funcionamento.

## 4.6 PROTÓTIPO

Após a construção do circuito, todos os componentes foram alocados para uma região tubular que a carcaça do equipamento já possuía e fechado com um cap de 60 mm de diâmetro, como pode ser visto na figura 34.

Figura 33 - Protótipo finalizado



Fonte: Elaboração própria (2023)

## 5 CONCLUSÃO

O presente trabalho teve como objetivo principal desenvolver um sistema eficiente, preciso e de baixo custo para o acondicionamento de pluviômetros antigos, incluindo a capacidade de integrar e dar suporte a sensores adicionais e conexão sem fio para transferência de dados. Este objetivo foi alcançado com sucesso, como evidenciam os resultados obtidos em cada etapa do projeto.

O sistema de acondicionamento do pluviômetro foi projetado com sucesso, integrando o microcontrolador e garantindo a interpretação correta dos pulsos do reed switch, resultando em medições precisas de precipitação. A capacidade de integrar sensores adicionais ao sistema foi uma consideração fundamental, permitindo a expansão e a versatilidade do sistema acondicionado.

Um servidor web foi desenvolvido e hospedado no ESP8266, facilitando a visualização e exportação dos dados coletados, além de oferecer suporte aos dados provenientes dos sensores adicionais. Este sistema tornou as informações de precipitação e de outros parâmetros ambientais facilmente acessíveis e manipuláveis, contribuindo para a eficácia do projeto.

A calibração e os testes do pluviômetro acondicionado, juntamente com os sensores adicionais, foram realizados em um ambiente controlado. Estes testes validaram a precisão do sistema de acondicionamento e a funcionalidade dos sensores adicionais, revelando um erro associado ao instrumento de 5% para chuvas de 30mm/h, 6% para 60mm/h e 9% para 90mm/h, todos dentro de limites aceitáveis.

O sucesso deste projeto é evidenciado pela realização de todos os objetivos específicos estabelecidos, incluindo a integração eficiente de sensores adicionais. O sistema de acondicionamento desenvolvido melhora a precisão dos pluviômetros antigos de maneira econômica e versátil, ressaltando a viabilidade do projeto. Este trabalho estabelece um precedente valioso para futuras iniciativas de modernização de equipamentos meteorológicos antigos, demonstrando que é possível alcançar alta precisão, eficiência e versatilidade a um custo reduzido.

## 6 SUGESTÃO DE TRABALHOS CORRELACIONADOS

Diversos estudos complementam e enriquecem o entendimento sobre pluviômetros e suas aplicações tecnológicas. O trabalho de Quinde Pisco e Junco Vaca (2021) se destaca pelo desenvolvimento de um pluviômetro com sistema de pesagem e integração de sensores de umidade e temperatura. A utilização de tecnologia IoT e a apresentação dos dados em uma página web são aspectos inovadores deste estudo.

Outra pesquisa significativa é a de Sanches et al. (2017), que foca no desenvolvimento de um pluviômetro de baixo custo usando Arduino. Este estudo demonstra a eficiência e confiabilidade de equipamentos econômicos por meio de métodos estatísticos de correlação, comparando-os com estação convencional e automática.

Por fim, De Oliveira, Noronha e Guimarães (2019) exploram a criação de um pluviômetro automatizado de código aberto e baixo custo, também baseado na plataforma Arduino. Este trabalho é relevante para a gestão urbana e monitoramento de índices pluviométricos, destacando-se pela sua capacidade de integração com sistemas de alerta de desastres naturais.

Cada um desses estudos contribui de maneira significativa para o campo da instrumentação meteorológica, oferecendo abordagens inovadoras e soluções práticas para desafios atuais no monitoramento de chuvas e análise de dados climáticos.

## REFERÊNCIAS

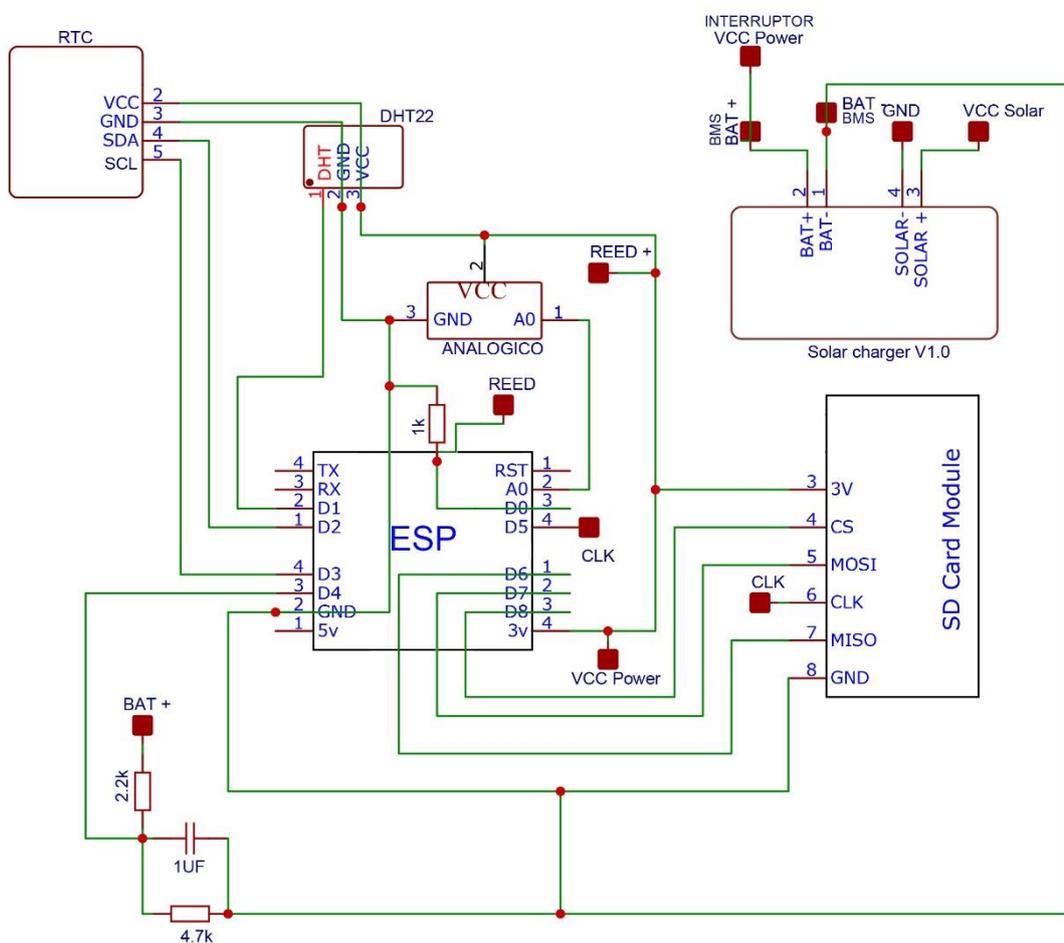
1. ALEXANDER, Charles K. **Fundamentos de circuitos elétricos**. Porto Alegre: Amgh, 2013.
2. ALENA, T. R.; APPLETON, J. S.; SERSTAD, W. H. **Measurement accuracy enhancement of tipping bucket rain gauges at high rainfall rates**. Preprints, Conf. on Operational Precipitation Estimation and Prediction, Anaheim, CA, Amer. Meteor. Soc., 16–111, 1990.
3. AYOADE, J. O. **Introdução à Climatologia para os Trópicos**. 4. ed. Rio de Janeiro: Editora Bertrand Brasil, 1996. 332 p.
4. CANEVAROLO, S. V., Jr. **Ciência dos polímeros: um texto básico para tecnólogos e engenheiros** (1st rep.). São Paulo: Artliber Editora, 2004.
5. DE OLIVEIRA, Samira Borba; NORONHA, Tiago Baptista; GUIMARAES, Vinícius Tavares. Pluviômetro Automatizado de Código Aberto e Baixo Custo Utilizando a Plataforma Arduino. **Revista Educar Mais**, v. 3, n. 3, p. 70-76, 2019.
6. ESPRESSIF SYSTEMS. **ESP8266: ESP8266EX**. Shanghai: Espressif Systems, 2015. Disponível em: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). Acesso em: 13 jul. 2023.
7. GRISON, F. et al. **Desempenho de um pluviômetro digital em campo**. In: Simpósio Brasileiro de Recursos Hídricos, 22., 2017, Florianópolis. Anais... Porto Alegre: ABRH, 2017. p. 1-7.
8. HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentos de Física: Eletromagnetismo**. 9. ed. Rio de Janeiro: LTC, [Ano de publicação]. vol. 3.
9. HUMPHREY, M. D.; ISTOK, J. D.; LEE, J. Y.; HEVESI, J. A.; FLINT, A. L. **A New Method for Automated Dynamic Calibration of Tipping-Bucket Rain Gauges**. American Meteorological Society, 1997.
10. KARVINEN, Kimmo. **Primeiros passos com sensores**. São Paulo: Novatec, 2014. 160 p.
11. KIEHNE, Heinz Albert. **Battery technology handbook**. CRC Press, 2003.
12. MANSANO, V. M. et al. **Calibração de pluviômetros de báscula: metodologia para análise e correção de dados de campo. Calibration of**

- tipping bucket rainfall gauges: a methodology for field data analysis and correction.** In: XV Safety, Health and Environment World Congress. 2015. p. 19-22.
13. MARSELEK, J. **Calibration of the tipping-bucket raingage.** J. Hydrol., 53, 343–354, 1981.
14. MEHTA, M. **ESP8266: A Breakthrough in wireless sensor networks and internet of things.** International Journal of Electronics and Communication Engineering & Technology, v. 6, n. 8, p. 7-11, 2015.
15. QUINDE PISCO, Germán Kevin; JUNCO VACA, Steven Israel. **Desarrollo de un prototipo para la predicción de lluvias por medio de un pluviómetro con sistema de pesaje.** 2021. Trabalho de Conclusão de Curso. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales.
16. SANCHES, Rafael Grecco et al. Proposta de pluviômetro de baixo custo utilizando a plataforma de prototipagem Arduino. **Os desafios da geografia física na fronteira do conhecimento**, v. 1, p. 2024-2035, 2017.
17. SEVRUK, Boris; KLEMM, S. **Catalogue of national standard precipitation gauges.** World Meteorological Organization, 1989.
18. SILVA, Marcel; FRUETT, Fabiano. **Monitoramento Ambiental através de Rede de Sensores Sem Fio de Baixo Custo.** XVI Simpósio Brasileiro de Sensoriamento Remoto-SBSR, INPE, 2013.
19. SODERGARD, A. **Properties of lactic acid based polymers and their correlation with composition.** Progress in polymer Science, p. 1123-1163, 2000.
20. THOMAZINI, Daniel. **Sensores Industriais: fundamentos e aplicações.** 5. ed. São Paulo: Érica, 2005. 222 p.
21. TINKERCAD. **Tinkercad | From mind to design in minutes.** Disponível em: <https://www.tinkercad.com/dashboard>.
22. VAREJÃO-SILVA, M. A. **Meteorologia e Climatologia, Versão Digital 2.** Recife, PE: INMET, 2006. 463 p.
23. Vuolo, J. H. **Teoria dos Erros.** [Local de publicação]: [Editora], 1996.
24. WMO - World Meteorological Organization. **Guide to Meteorological Instruments and Methods of Observation.** Genebra, Suíça: Wmo, 2008. 681 p.

25. WMO - World Meteorological Organization. **Annual report of the world meteorological organization**. Geneva, Suíça: Wmo, 1985. 230 p.
26. World Meteorological Organization. **Guide to meteorological instruments and methods of observation**. 5th ed. Geneva, Switzerland, 1983.

|

## ANEXO A – Guia de Conexões



## ANEXO B – Código Desenvolvido

```
#include <ESP8266WebServer.h>
#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <RTClib.h>
#include <Wire.h>
#include <DHT.h>
#include <SPI.h>
#include <SD.h>

#define DHTPIN 05 // Define a porta à qual o sensor DHT22 está conectado
#define DHTTYPE DHT22 // Define o tipo de sensor (DHT22)

DHT dht(DHTPIN, DHTTYPE);

RTC_PCF8563 rtc;

ESP8266WebServer server(80); // Define o servidor na porta 80

const int chipSelect = 15;

const int reedPin = 16; // Reed switch conectado ao pino D0

int bateria = 100;

float pulseCount = 0; // contador de pulsos
float waterAmount = 0; // quantidade de agua

float a = 0.3; //relação bascula/mm

int Ultima leitura_dht = (-1), Ultimahoraregistrochuva = (-1), Ultima leitura_bat = (-1), Ultima leitura_sensor = (-1);
float Leitura_analogica;
float intervaloColeta chuva = 1;
int intervaloColetaSensor;
int intervalo_dht;
int horaConfig_dht;
int horaConfig_sensor;

double Ax = 0.0; // Parâmetro A da função de calibração
double Bx = 0.0; // Parâmetro B da função de calibração

bool estadoSensor = 0;
```

```
bool estado_dht = 0;

int carga, intervaloColeta;

char buffer[50];

const char Style[] PROGMEM = R"(
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
  body {
    color: #bfc1c8;
    font-family: "Roboto", "Open Sans", sans-serif;
    font-size: 14px;
    font-weight: 300;
    line-height: 1.5;
    background: #1e202b;
    margin: 0;
    padding: 20px 0;
    display: flex;
    flex-direction: column;
    align-items: center;
  }

  #container {
    max-width: 800px;
    width: 100%;
    background: #323544;
    border-radius: 10px;
    margin-top: 20px;
    padding: 20px;
    text-align: center;
    box-sizing: border-box;
  }

  h1 {
    color: #007bff;
    margin: 0;
    padding: 20px;
    text-align: center;
  }
)
```

```
p {
  font-size: 18px;
  margin-bottom: 10px;
  text-align: center;
}

#month, #year, #date, #time, #config{
  background-color: #323544;
  color: #ffffff;
  border: 2px solid #fff;
  border-radius: 15px;
  text-align: center;
  padding: 2px;
  margin: 2px 5px 5px 0px;
}

#config{
  background-color: #323544;
  color: #ffffff;
  border: 1px solid #fff;
  border-radius: 15px;
  text-align: center;
  padding: 2px;
  margin: 2px 5px 5px 0px;
}

#btn {
  display: block;
  width: 100%;
  background-color: #007bff;
  color: #ffffff;
  text-align: center;
  padding: 10px 0;
  margin: 5px;
  border-radius: 15px;
  text-decoration: none;
  font-size: 16px;
  outline: none;
}

#btn:hover {
  background-color: #0056b3;
}

.btn_menu {
```

```
display: inline;
width: 100%;
background-color: #007BFF;
color: #ffffff;
text-align: center;
justify-content: space-between;
padding: 10px 10px;
margin: 10px 0px 10px 10px;
border-radius: 15px;
text-decoration: none;
font-size: 16px;
outline: none;
border: none;
transition: background-color 0.3s;
}
.btn_menu-svg {
display: inline;
width: 30px;
height: 20px;
fill: white;
margin: 0px;
}
.btn_menu:hover {
background-color: #0056b3;
}
.date-time-top {
display: inline;
text-align: center;
font-size: 12px;
position: fixed;
top: 0;
left: 0;
right: 0;
padding: 5px 0;
}
.planner-header {
background-color: #323544;
color: white;
text-align: center;
padding: 10px;
}
.planner-header a {
display: block;
padding: 10px 20px;
width: 90%;
```

```
background-color: #007BFF;
color: #FFFFFF;
text-decoration: none;
border-radius: 20px;
margin: 5px;
transition: background-color 0.3s;
font-size: 16px;
}
.menu a {
display: inline-block;
padding: 10px 0px;
width: auto;
background-color: #007BFF;
color: #FFFFFF;
text-decoration: none;
border-radius: 15px;
margin: 5px;
transition: background-color 0.3s;
font-size: 16px;
}
}

.menu a:hover {
background-color: #555;
}
}

.form-section {
display: none;
}
}

.active {
display: block;
}
}

input[type="submit"] {
display: inline;
width: auto;
background-color: transparent;
color: #ffffff;
text-align: center;
justify-content: space-between;
padding: 10px 0px;
margin: 10px 0px;
border-radius: 15px;
text-decoration: none;
font-size: 16px;
outline: none;
border: none;
```

```
    transition: background-color 0.3s;
}

i.battery-icon {
  display: inline;
  position: absolute;
  top: 10px;
  right: 10px;
  height: 8px;
  width: 22px;
  background-color: #1e202b;
  border: 2px solid #bfc1c8;
  display: inline-block;
  vertical-align: middle;
  &:after,
  &:before {
    background: #bfc1c8;
    content: "";
    position: absolute;
  }
  &:before {
    width: 2px;
    bottom: 1px;
    right: -4px;
    top: 1px;
  }
  &:after {
    bottom: 2px;
    left: 2px;
    top: 2px;
  }
  &[data-percent="100"]:after {
    right: 2px;
    background-color: rgb(14, 200, 233);
  }
  &[data-percent="90"]:after {
    right: 2px;
  }
  &[data-percent="80"]:after {
    right: 4px;
  }
  &[data-percent="70"]:after {
    right: 6px;
  }
}
```

```

&[data-percent="60"]:after {
    right: 8px;
}
&[data-percent="50"]:after {
    right: 10px;
}
&[data-percent="40"]:after {
    right: 12px;
}
&[data-percent="30"]:after {
    right: 14px;
}
&[data-percent="20"]:after {
    right: 16px;
}
&[data-percent="10"]:after {
    right: 18px;
    background-color: red;
    border-color: red;
}
&[data-percent="0"]:after {
    content: "~";
    top: -12px;
    left: 0;
    right: 0;
    bottom: 0;
    color: red;
    background: none;
    font-size: 22px;
    text-align: center;
}
}
</style>
</head>
<body>
)";

const char htmlFooter[] PROGMEM = R"(
    <a href="#" id="btn" ONCLICK='JS1()>Imprimir</a>
    <script>
        function JS1() {
            window.print();
        }
    </script>
)";

```

```

const char setDateTimePage[] PROGMEM = R"(
<html>
<body>
  <div id="container">
    <h1>Ajustar Data e Hora</h1>
    <div class="form-section">
      <form action="/setdatetime" method="POST" class="centered-form">
        <label for="date">Data:</label>
        <input type="date" id="date" name="date" required><br>
        <label for="time">Hora:</label>
        <input type="time" id="time" name="time" required><br>
        <input type="submit" value="Atualizar Data e Hora">
      </form>
    </div>
    <div class="form-section" id="Atualizar Data e Hora">
      <form action="/setdatetime" method="GET">
        <label for="date">Data:</label>
        <input type="date" id="date" name="date" value="{userYear}-{userMonth:02d}-{userDay:02d}"
required><br>
        <label for="time">Hora:</label>
        <input type="time" id="time" name="time" value="{userHour:02d}:{userMinute:02d}" required><br>
        <input type="submit" value="Atualizar Data e Hora">
      </form>
    </div>
  </div>
</body>
</html>
)";

char htmlHeader[] PROGMEM = R"(
</head>
<body>

  <div id="container">
    <h1>Pluviômetro Digital</h1>
    <div class='planner-header'>
  )";

// Função para gerar opções de mês
String gerarOpcoesMes(int mesAtual) {
  const char* meses[] = {"Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho", "Agosto",
"Setembro", "Outubro", "Novembro", "Dezembro"};
  String opcoes = "";
  for (int i = 0; i < 12; i++) {

```

```

    if (i+1 == mesAtual) {
        opcoes += "<option value=\"" + String(i+1) + "\" selected>" + meses[i] + "</option>";
    } else {
        opcoes += "<option value=\"" + String(i+1) + "\">" + meses[i] + "</option>";
    }
}
return opcoes;
}

```

```
DateTime now;
```

```

void setup() {
    Serial.begin(9600); // inicia a comunicação serial
    Wire.begin(4,0); // Inicialize a comunicação I2C
    if (!rtc.begin()) {
        Serial.println("RTC não encontrado!");
    }
}

```

```
DateTime now = rtc.now();
```

```

pinMode(reedPin, INPUT_PULLUP);
pinMode(A0, INPUT);

```

```

WiFi.softAP("Pluviômetro", "dnmdx_rb675"); // cria uma rede WiFi chamada "Pluviômetro" com senha
"d[nmdx=_rb"
Serial.println("Rede WiFi criada.");

```

```
server.begin();
```

```

server.on("/", handleRoot);
server.on("/setdatetime", handleSetDateTime);
server.on("/filtrar_chuva", filtrado_chuva);
server.on("/filtrar_dht", filtrado_dht);
server.on("/filtrar_bat", filtrado_bat);
server.on("/filtrar_sensor", filtrado_sensor);
server.on("/configurardht", handleConfigureDHT);
server.on("/configurarsensor", handleConfigureSensor);
server.on("/formatar", handleFormatar);
server.on("/confirmaformatar", handleConfirmaFormatar);

```

```
Serial.println("Servidor iniciado.");
```

```

if (!SD.begin(chipSelect)) {
    Serial.println("Falha na inicialização do cartão SD.");
}

```

```

}

delay(500);
Ultimahoraregistrochuva = now.hour();

File dataFile = SD.open("dados_bat.txt", FILE_WRITE);
if (dataFile) {
  dataFile.println(String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" + String(now.year(),
DEC) + " às " + String(now.hour(), DEC)+ "h o sistema foi reiniciado");
  dataFile.close();
} else {
  Serial.println("Erro ao abrir o arquivo dados_bat.txt");
}

dht.begin();

WiFi.setOutputPower(5.0);

if (MDNS.begin("pluviometro")) {
  MDNS.addService("http", "tcp", 80);
}

if (SD.exists("config.txt")) {

  File dataFile = SD.open("config.txt", FILE_READ);

  if (dataFile) {
    intervalo_dht = int(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    horaConfig_dht = int(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    estado_dht = bool(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    intervaloColetaSensor = int(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    horaConfig_sensor = int(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    estadoSensor = bool(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    Ax= float(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    Bx = float(dataFile.parseInt());
    dataFile.read(); // pula o delimitador
    dataFile.close();
  }
}

```

```

} else{
    intervalo_dht = 1;
    horaConfig_dht = now.hour();
    estado_dht = 0;
    intervaloColetaSensor = 1;
    horaConfig_sensor = now.hour();
    estadoSensor = 0;
    Ax = 0;
    Bx = 0;
}
}
}

```

```

void enviarFormularioComValoresRTC(const char* acao, const char* valorSubmit, int mes, int ano) {
    char buffer[1000]; // Buffer para formatação dinâmica.
    sprintf(buffer, R"(
<form action="%s" method="GET">
    <label for='month'>Mês:<label>
    <select id='month' name='month'>
        %s
    </select>
    <label for='year'>Ano:<label>
    <input type='number' id='year' name='year' min='2000' max='2100' value='%d'>
    <label class='btn_menu'>
        <input type='submit' value='%s' />
        <span>#128269;</span>
    </label>

</form>
)", acao, gerarOpcoesMes(mes).c_str(), ano, valorSubmit);
    server.setContent(buffer);
}

```

```

void handleRoot() {
    medirNivelBateria();

    DateTime now = rtc.now();
    int dia = now.day();
    int mes = now.month();
    int ano = now.year();
    int hora = now.hour();
    int minutos = now.minute();

```

```

// Formatação dos minutos com zero à esquerda

```

```

char minutosFormatados[3];
sprintf(minutosFormatados, sizeof(minutosFormatados), "%02d", minutos);

server.sendContent_P(Style);
server.sendContent("<div class='date-time-top'>Data e Hora Atual: " + String(dia) + "/" + String(mes) +
"/" + String(ano) + " | " + String(hora) + ":" + String(minutosFormatados) + "</div>");

server.sendContent_P(htmlHeader);

// Enviando cada formulário com os valores do RTC
enviarFormularioComValoresRTC("/filtrar_chuva", "Chuva", mes, ano);
enviarFormularioComValoresRTC("/filtrar_dht", "DHT22", mes, ano);
enviarFormularioComValoresRTC("/filtrar_bat", "Bateria", mes, ano);
enviarFormularioComValoresRTC("/filtrar_sensor", "Sensor", mes, ano);

const char* htmlFinal = R"(
</div>
<div class="menu">
  <a href="/setdatetime" id="btn">Atualizar Data e Hora</a>
  <a href="/configurarsensor" id="btn">Adicionar sensor analógico</a>
  <a href="/configurardht" id="btn">Adicionar DHT22</a>
  <a href="/formatar" id="btn">Limpar registros</a>
</div>
<i id='battery-icon' class='battery-icon' data-charging=" data-percent='%d'></i>
</div>
</body>
)";
char bufferFinal[1000];
sprintf(bufferFinal, htmlFinal, bateria);
server.sendContent(bufferFinal);

Serial.println("Página web atualizada.");
}

void filtrado_chuva() {
int mesFiltro = server.arg("month").toInt();
int anoFiltro = server.arg("year").toInt();

server.sendContent_P(Style);
server.sendContent_P(R("<h1>Leituras para )");
server.sendContent(String(mesFiltro));
server.sendContent("/");
server.sendContent(String(anoFiltro));
server.sendContent_P(R(
</h1>

```

```

    <div id="container">
  );

  // ENTRADA DE DADOS

  File dataFile = SD.open("dados_chuva.txt", FILE_READ);
  if (dataFile) {
    String line;
    while (dataFile.available()) {
      line = dataFile.readStringUntil('\n');
      int indexBarra = line.indexOf('/');
      int indexBarra1 = line.indexOf('/', indexBarra + 1);
      int indexEspaco = line.indexOf(' ', indexBarra1);
      int mesString = line.substring(indexBarra + 1, indexBarra1).toInt();
      int anoString = line.substring(indexBarra1 + 1, indexEspaco).toInt();

      if (int(mesString) == int(mesFiltro) && int(anoString) == int(anoFiltro)) {
        server.sendContent(line + "<br>");
      }
    }
    dataFile.close();
  }

  server.sendContent_P(htmlFooter);
  server.sendContent_P(R"(
    <a href="/" id="btn" >Retornar à página principal</a>
  </div>
  </body>
  </html>
  )");
}

void filtrado_sensor() {
  int mesFiltro = server.arg("month").toInt();
  int anoFiltro = server.arg("year").toInt();

  server.sendContent_P(Style);
  server.sendContent_P(R("<h1>Leituras para )");
  server.sendContent(String(mesFiltro));
  server.sendContent("/");
  server.sendContent(String(anoFiltro));
  server.sendContent_P(R"(
    </h1>
    <div id="container">

```

```

    });

    // ENTRADA DE DADOS

    File dataFile = SD.open("dados_sensor.txt", FILE_READ);
    if (dataFile) {
        String line;
        while (dataFile.available()) {
            line = dataFile.readStringUntil('\n');
            int indexBarra = line.indexOf('/');
            int indexBarra1 = line.indexOf('/', indexBarra + 1);
            int indexEspaco = line.indexOf(' ', indexBarra1);
            int mesString = line.substring(indexBarra + 1, indexBarra1).toInt();
            int anoString = line.substring(indexBarra1 + 1, indexEspaco).toInt();

            if (int(mesString) == int(mesFiltro) && int(anoString) == int(anoFiltro)) {
                server.sendContent(line + "<br>");
            }
        }
        dataFile.close();
    }
    server.sendContent_P(htmlFooter);
    server.sendContent_P(R"(
        <a href="/" id="btn" >Retornar à página principal</a>
    </div></body></html>
    )");
}

void medirNivelBateria() {
    // Descarrega o capacitor
    pinMode(2, OUTPUT);
    digitalWrite(2, LOW);
    delay(500); // Tempo para garantir a descarga completa

    unsigned long startTime = micros();
    pinMode(2, INPUT);

    // Aguarda até que o pino digital leia HIGH
    while (digitalRead(2) == LOW){
        if(digitalRead(2) == HIGH) {break;}
    }

    // Calcula o tempo decorrido
    unsigned long elapsedTime = micros() - startTime;

```

```

// Calcula a carga da bateria de acordo com o tempo de carregamento do capacitor
carga = (-181.1) * log(elapsedTime) + 1314.4;

Serial.print("carga:");
Serial.println(carga);

if(carga > 100){
  carga = 100;
}
if(carga < 0){
  carga = 0;
}

bateria = carga - (carga %10);

}

void handleSetDateTime() {
  if (server.method() == HTTP_POST) {
    if (server.hasArg("date") && server.hasArg("time")) {
      String date = server.arg("date");
      String time = server.arg("time");

      int day = date.substring(8, 10).toInt();
      int month = date.substring(5, 7).toInt();
      int year = date.substring(0, 4).toInt();
      int hour = time.substring(0, 2).toInt();
      int minute = time.substring(3, 5).toInt();

      AdjustDateAndTime(day, month, year, hour, minute);

      String response = String(Style);
      response += "<div id='container'><h1>Ajustar Data e Hora</h1><p>Data e hora atualizadas com
sucesso!</p></div></body></html>";
      response += "<script>setTimeout(function() { window.location.href = '/'; },
1500);</script></body></html>";
      server.send(200, "text/html", response);
    } else {
      server.send(400, "text/plain", "Parâmetros inválidos");
    }
  } else {
    String htmlForm = String(Style) + R"(
<div id="container">
  <h1>Ajustar Data e Hora</h1>
  <form action="/setdatetime" method="POST">

```

```

        <label for="date">Data:</label>
        <input type="date" id="date" name="date" required><br>
        <label for="time">Hora:</label>
        <input type="time" id="time" name="time" required><br>
        <br>
        <input type="submit" id="btn" value="Atualizar Data e Hora">
        <a href="/" id="btn" >Retornar à página principal</a>
    </form>
</div>
</body>
</html>
)";
server.send(200, "text/html", htmlForm);
}
}

void AdjustDateAndTime(int day, int month, int year, int hour, int minute) {
    DateTime dt(year, month, day, hour, minute, 0);
    rtc.adjust(dt);
}

void mudarHorarios_sensor(){
    SD.begin(chipSelect);
    SD.remove("config.txt");

    File dataFile = SD.open("config.txt", FILE_WRITE);

    if (dataFile) {
        dataFile.print(intervalo_dht);
        dataFile.print(",");
        dataFile.print(horaConfig_dht);
        dataFile.print(",");
        dataFile.print(estados_dht);
        dataFile.print(",");
        dataFile.print(intervaloColetaSensor);
        dataFile.print(",");
        dataFile.print(horaConfig_sensor);
        dataFile.print(",");
        dataFile.print(estadosSensor);
        dataFile.print(",");
        dataFile.print(Ax);
        dataFile.print(",");
        dataFile.print(Bx);
        dataFile.close();
    }
}

```

```

}
}

void handleConfigureSensor() {
    if (server.method() == HTTP_POST ) {
        if (server.hasArg("estadoSensor") && server.hasArg("Ax") && server.hasArg("Bx") &&
server.hasArg("intervaloColetaSensor")) {
            estadoSensor = server.arg("estadoSensor").toInt();
            Ax = server.arg("Ax").toFloat();
            Bx = server.arg("Bx").toFloat();
            intervaloColetaSensor = server.arg("intervaloColetaSensor").toInt();

            Serial.println("Intervalo de coleta definido para " + String(intervaloColetaSensor) + " horas");
            Serial.println("Parametros alterados: A = " + String(Ax) + " e B = " + String(Bx));
            Serial.println("Estado:" + String(estadoSensor));
        }
    }

    server.setContent_P(Style);
    server.setContent_P(R"(
<div id="container">
    <h1>Configurar Sensor Analógico</h1>
    <p>Estado atual do sensor: )");
    server.setContent(estadoSensor ? "Ativado" : "Desativado");
    server.setContent_P(R"(<br>Função de Calibração:  $y =$  )");
    server.setContent(String(Ax));
    server.setContent_P("x + ");
    server.setContent(String(Bx));
    server.setContent_P(R"(<br><br>
    <form action="/configurarsensor" method="POST">
        <label>Ativar Sensor: </label>
        <select id="config" name="estadoSensor">
            <option value="1" )");
    server.setContent(estadoSensor ? "selected" : "");
    server.setContent_P(R"(>Ativado</option>
        <option id="config" value="0" )");
    server.setContent(!estadoSensor ? "selected" : "");
    server.setContent_P(R"(>Desativado</option>
    </select><br><br>
    <label for="Ax">Parâmetro A: </label>
    <input type="text" id="config" name="Ax" value=")");
    server.setContent(String(Ax));
    server.setContent_P(R)(" required><br><br>
    <label for="Bx">Parâmetro B: </label>
    <input type="text" id="config" name="Bx" value=")");

```

```

server.sendContent(String(Bx));
server.sendContent_P(R(" required<br><br>
    <label for="intervaloColetaSensor">Intervalo entre coletas (horas): </label>
    <select id="config" name="intervaloColetaSensor">
        )");
for (int i = 1; i <= 4; i++) {
    char optionValue[2];
    itoa(i, optionValue, 10);
    server.sendContent_P("<option value=\"");
    server.sendContent(optionValue);
    server.sendContent_P("\");
    if (i ==intervaloColetaSensor) {
        server.sendContent_P(" selected");
    }
    server.sendContent_P(">");
    server.sendContent(optionValue);
    server.sendContent_P("</option>");
}
server.sendContent_P(R(
    </select><br>
    <input type="submit" id="btn" value="Atualizar">
</form>
));

server.sendContent_P(R(
    <a href="/" id="btn" >Retornar à página principal</a>
</div>
));

mudarHorarios_sensor();
}

```

```

void mudarHorarios_dht(){
    SD.begin(chipSelect);
    SD.remove("config.txt");

    File dataFile = SD.open("config.txt", FILE_WRITE);

    if (dataFile) {
        dataFile.print(intervalo_dht);
        dataFile.print(",");
        dataFile.print(horaConfig_dht);
        dataFile.print(",");
    }
}

```

```

dataFile.print(estado_dht);
dataFile.print(",");
dataFile.print(intervaloColetaSensor);
dataFile.print(",");
dataFile.print(horaConfig_sensor);
dataFile.print(",");
dataFile.print(estadoSensor);
dataFile.print(",");
dataFile.print(Ax);
dataFile.print(",");
dataFile.print(Bx);
dataFile.close();
}
}

void handleConfigureDHT() {
  if (server.method() == HTTP_POST ) {
    if (server.hasArg("estado_dht") && server.hasArg("intervalo_dht")) {
      estado_dht = server.arg("estado_dht").toInt();
      intervalo_dht = server.arg("intervalo_dht").toInt();

      Serial.println("Intervalo de coleta definido para " + String(intervalo_dht) + " horas");
      Serial.println("Estado:" + String(estado_dht));
    }
  }

  server.setContent_P(Style);
  server.setContent_P(R"(
  <div id="container">
    <h1>Configurar Sensor DHT</h1>
    <p>Estado atual do sensor: )");
  server.setContent(estado_dht ? "Ativado" : "Desativado");
  server.setContent_P(R("<br>Intervalo entre coletas: )");
  server.setContent(String(intervalo_dht));
  server.setContent_P(R("<br><br>
  <form action="/configurardht" method="POST">
    <label>Ativar Sensor: </label>
    <select id="config" name="estado_dht">
      <option value="1" )");
  server.setContent(estado_dht ? "selected" : "");
  server.setContent_P(R(">Ativado</option>
  <option value="0" )");
  server.setContent(!estado_dht ? "selected" : "");
  server.setContent_P(R(">Desativado</option>
  </select><br><br>

```

```

        <label for="intervalo_dht">Intervalo entre coletas (horas): </label>
        <select id="config" name="intervalo_dht">
            )");
for (int i = 1; i <= 4; i++) {
    char optionValue[2];
    itoa(i, optionValue, 10);
    server.sendContent_P("<option value=\");
    server.sendContent(optionValue);
    server.sendContent_P("\");
    if (i == intervalo_dht) {
        server.sendContent_P(" selected");
    }
    server.sendContent_P(">");
    server.sendContent(optionValue);
    server.sendContent_P("</option>");
}
server.sendContent_P(R"(
        </select><br>
        <input type="submit" id="btn" value="Atualizar">
        </form>
    )");

server.sendContent_P(R"(
    <a href="/" id="btn" >Retornar à página principal</a>
    </div>
    )");

mudarHorarios_dht();

}

void filtrado_dht() {
    int mesFiltro = server.arg("month").toInt();
    int anoFiltro = server.arg("year").toInt();

    server.sendContent_P(Style);
    server.sendContent_P(R("<h1>Leituras para )");
    server.sendContent(String(mesFiltro));
    server.sendContent("/");
    server.sendContent(String(anoFiltro));
    server.sendContent_P(R"(
        </h1>
        <div id="container">
    )");
}

```

```
// ENTRADA DE DADOS
```

```
File dataFile = SD.open("dados_dht.txt", FILE_READ);
if (dataFile) {
  String line;
  while (dataFile.available()) {
    line = dataFile.readStringUntil('\n');
    int indexBarra = line.indexOf('/');
    int indexBarra1 = line.indexOf('/', indexBarra + 1);
    int indexEspaco = line.indexOf(' ', indexBarra1);
    int mesString = line.substring(indexBarra + 1, indexBarra1).toInt();
    int anoString = line.substring(indexBarra1 + 1, indexEspaco).toInt();

    if (int(mesString) == int(mesFiltro) && int(anoString) == int(anoFiltro)) {
      server.sendContent(line + "<br>");
    }
  }
  dataFile.close();
}
```

```
server.sendContent_P(htmlFooter);
server.sendContent_P(R"(
  <a href="/" id="btn" >Retornar à página principal</a>
</div></body></html>
)");
}
```

```
void filtrado_bat() {
  int mesFiltro = server.arg("month").toInt();
  int anoFiltro = server.arg("year").toInt();

  server.sendContent_P(Style);
  server.sendContent_P(R("<h1>Leituras para )");
  server.sendContent(String(mesFiltro));
  server.sendContent("/");
  server.sendContent(String(anoFiltro));
  server.sendContent_P(R"(
    </h1>
    <div id="container">
  )");
}
```

```
// ENTRADA DE DADOS
```

```
File dataFile = SD.open("dados_bat.txt", FILE_READ);
if (dataFile) {
```

```

String line;
while (dataFile.available()) {
    line = dataFile.readStringUntil('\n');
    int indexBarra = line.indexOf('/');
    int indexBarra1 = line.indexOf('/', indexBarra + 1);
    int indexEspaco = line.indexOf(' ', indexBarra1);
    int mesString = line.substring(indexBarra + 1, indexBarra1).toInt();
    int anoString = line.substring(indexBarra1 + 1, indexEspaco).toInt();

    if (int(mesString) == int(mesFiltro) && int(anoString) == int(anoFiltro)) {
        server.setContent(line + "<br>");
    }
}
dataFile.close();
}

server.setContent_P(htmlFooter);
server.setContent_P(R"(
    <a href="/" id="btn" >Retornar à página principal</a>
</div></body></html>
)");
}

void handleFormatar() {
    String response = String(Style);
    response += "<div id='container'><h1>Limpar registros</h1>";
    response += "<p>Tem certeza que deseja apagar todos os dados ?</p>";
    response += "<a href='/confirmaformatar' id='btn'>Sim</a><br>";
    response += "<a href='/' id='btn'>Não</a></div></body></html>";
    server.send(200, "text/html", response);
}

void handleConfirmaFormatar() {
    SD.begin(chipSelect); // Inicializa o cartão SD novamente
    SD.remove("dados_chuva.txt"); // Formata o cartão SD
    SD.remove("dados_bat.txt");
    SD.remove("dados_sensor.txt");
    SD.remove("dados_dht.txt");

    String response = String(Style);
    response += "<div id='container'><h1>Dados apagados com sucesso!</h1>";
    response += "<a href='/' id='btn'>Retornar à página principal</a></div></body></html>";
    server.send(200, "text/html", response);
}

```

```

void loop() {
  DateTime now = rtc.now();

  server.handleClient();
  MDNS.update();

  if (digitalRead(reedPin) == HIGH) {
    pulseCount++;

    waterAmount = a * pulseCount;

    Serial.println("Pulso detectado.");
    Serial.println("Total de pulsos: " + String(pulseCount));
    Serial.println("Quantidade de água: " + String(waterAmount, 4) + " mm");

    while(digitalRead(reedPin) == HIGH) {
      if(digitalRead(reedPin) == LOW) {
        break;
      }
      delay(10);
    }
  }

  if (((now.hour() - Ultimahoraregistrochuva) == (intervaloColetachuva)) || ((now.hour() -
  Ultimahoraregistrochuva) == (-23))) {

    Ultimahoraregistrochuva = now.hour();

    String Dado_chuva_salvar;

    if(waterAmount > 0) {
      Dado_chuva_salvar= String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" +
      String(now.year(), DEC) + " às " + String(now.hour(), DEC)+ "h foi registrado " + String(waterAmount, 4)
      + "mm de chuva";
      waterAmount = 0;
      pulseCount = 0;

      File dataFile = SD.open("dados_chuva.txt", FILE_WRITE);

      if (dataFile) {
        dataFile.println(Dado_chuva_salvar);
        dataFile.close();
        Serial.println("Dados de chuva gravados.");
      } else {

```

```

    Serial.println("Erro ao abrir o arquivo dados_chuva.txt");
  }
}

if(estado_dht){

  if (((now.hour() % intervalo_dht) == (horaConfig_dht % intervalo_dht)) && (now.hour() !=
Ultimaleitura_dht)) {
    Ultimaleitura_dht=now.hour();

    float humidity;
    float temperature;

    String Dado_dht_salvar;

    if ((!isnan(humidity)) && (!isnan(temperature))) {
      humidity = dht.readHumidity();
      temperature = dht.readTemperature();

      Dado_dht_salvar= String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" +
String(now.year(), DEC) + " às " + String(now.hour(), DEC)+ "h " + "Umidade: " + String(humidity) + "%
Temperatura: " + String(temperature) + "°C";

      File dataFile = SD.open("dados_dht.txt", FILE_WRITE);

      if (dataFile) {
        dataFile.println(Dado_dht_salvar);
        dataFile.close();
        Serial.println("Dados gravados.");

      } else {
        Serial.println("Erro ao abrir o arquivo dados_dht.txt");
      }

    } else {
      Dado_dht_salvar = String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" +
String(now.year(), DEC) + " às " + String(now.hour(), DEC)+ "h " + " Erro ao ler o sensor DHT22";

      File dataFile = SD.open("dados_dht.txt", FILE_WRITE);

      if (dataFile) {
        dataFile.println(Dado_dht_salvar);
        dataFile.close();

```

```

Serial.println("Dados gravados.");

} else {
  Serial.println("Erro ao abrir o arquivo dados_dht.txt");
}
}
}
}

if (((now.hour() == 2) || (now.hour() == 6) || (now.hour() == 10) || (now.hour() == 14) || (now.hour() ==
18) || (now.hour() == 22)) && (now.hour() != Ultimizeitura_bat)) {

  Ultimizeitura_bat= now.hour();
  medirNivelBateria();
  String Dado_bat_salvar;

  Dado_bat_salvar= String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" + String(now.year(),
DEC) + " às " + String(now.hour(), DEC)+ "h " + " Bateria: " + String(carga) + "% ";

  File dataFile = SD.open("dados_bat.txt", FILE_WRITE);

  if (dataFile) {
    dataFile.println(Dado_bat_salvar);
    dataFile.close();
    Serial.println("Dados gravados.");

  } else {
    Serial.println("Erro ao abrir o arquivo dados_bat.txt");
  }
}

if(estadosensor) {

  if(((now.hour() % intervaloColetaSensor) == horaConfig_sensor % intervaloColetaSensor) &&
(now.hour() != Ultimizeitura_sensor)) {
    Ultimizeitura_sensor=now.hour();
    Leitura_analogica = ((analogRead(A0) * Ax) + Bx);
    String Dado_sensor_salvar;

    Dado_sensor_salvar= String(now.day(), DEC) + "/" + String(now.month(), DEC) + "/" +
String(now.year(), DEC) + " às " + String(now.hour(), DEC)+ "h " + "Leitura_analógica: " +
String(Leitura_analogica);

    File dataFile = SD.open("dados_sensor.txt", FILE_WRITE);

```

```
if (dataFile) {  
    dataFile.println(Dado_sensor_salvar);  
    dataFile.close();  
    Serial.println("Dados gravados.");  
  
} else {  
    Serial.println("Erro ao abrir o arquivo dados_sensor.txt");  
}  
}  
}
```