

TRABALHO DE CONCLUSÃO DE CURSO

ENGENHARIA AGRÍCOLA E AMBIENTAL

**ESTIMATIVA DA EVAPOTRANSPIRAÇÃO COM ARDUINO PELO MÉTODO
DE HARGREAVES-SAMANI**

WARLEY MENEZES BARROS



Warley Menezes Barros

**ESTIMATIVA DA EVAPOTRANSPIRAÇÃO COM ARDUINO PELO MÉTODO
DE HARGREAVES-SAMANI**

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciências Agrárias da Universidade Federal de Minas Gerais, como requisito parcial, para obtenção do título de Bacharel em Engenharia Agrícola e Ambiental.

Orientador: Prof. Dr. Sidney Pereira

Montes Claros
2025

Warley Menezes Barros. ESTIMATIVA DA EVAPOTRANSPIRAÇÃO COM ARDUINO
PELO MÉTODO DE HARGREAVES-SAMANI

Aprovado pela banca examinadora constituída por:

Dra. Júlia Ferreira da Silva ICA/UFMG

Dr. Rodolpho Cesar dos Reis Tinini ICA/UFMG

Prof. Dr. Sidney Pereira - Orientador ICA/UFMG

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder força, perseverança e sabedoria ao longo desta jornada acadêmica.

Agradeço à minha mãe, Jaqueline, por sempre acreditar no meu potencial e me apoiar em toda essa jornada. Sem ela, esse sonho não seria possível. Seu amor incondicional e incentivo constante foram essenciais para que eu chegasse até aqui.

Agradeço à minha família, por estar ao meu lado em todos os momentos e apoiar cada um dos meus sonhos, me dando forças para seguir em frente.

Agradeço à UFMG por todo o suporte oferecido ao longo da minha formação, proporcionando um ambiente acadêmico inspirador e repleto de oportunidades para meu crescimento pessoal e profissional. O conhecimento adquirido e as experiências vividas foram essenciais para minha evolução e amadurecimento.

Manifesto minha profunda gratidão ao meu orientador, Sidney Pereira, por sua dedicação, paciência e valiosas orientações durante o desenvolvimento deste trabalho. Seu conhecimento, incentivo e apoio contínuo foram determinantes para a concretização deste projeto, tornando cada desafio uma oportunidade de aprendizado.

Aos meus professores, que sempre estiveram acessíveis e dispostos a compartilhar seus conhecimentos com entusiasmo e comprometimento, deixo meu sincero agradecimento. Cada ensinamento transmitido ao longo da minha trajetória acadêmica contribuiu significativamente para minha evolução e preparação para os desafios da vida profissional.

Meus sinceros agradecimentos aos meus amigos de Brumadinho, Gordo (Francienio) e Xoviz (Rafael), que, mesmo à distância, estiveram comigo em todos os momentos, oferecendo apoio, amizade e palavras de incentivo, tornando essa caminhada muito mais leve e especial.

Aos meus amigos de faculdade, Emilly, Pedro e Tais, sou imensamente grato por toda a ajuda e companheirismo ao longo dessa trajetória. Desde a adaptação em uma nova cidade até as longas sessões de estudo em grupo, vocês tornaram essa experiência muito mais enriquecedora e inesquecível.

A todos que, de alguma forma, contribuíram para a realização deste trabalho, meu sincero muito obrigado!

RESUMO

Este estudo apresenta o desenvolvimento e validação de uma estação meteorológica de baixo custo baseada na plataforma Arduino, projetada para estimar a evapotranspiração de referência (ET_o) pelo método de Hargreaves-Samani (HS). O experimento foi realizado com sensores acessíveis, como o sensor de temperatura DS18B20, e contou com recursos como *display* LCD, armazenamento em SD e comunicação via *Bluetooth*. A coleta de dados ocorreu entre os dias 5 e 15 de dezembro de 2024, e os resultados foram comparados com os obtidos por uma estação meteorológica automática, utilizando o método de Penman-Monteith-FAO (PM) como referência. A validação foi realizada por meio dos índices de concordância de Willmott, correlação de Pearson e índice “c”, mostrando uma boa correlação entre os métodos, destacando o potencial do sistema proposto como uma alternativa acessível e eficiente para a estimativa da ET_o, especialmente em regiões com acesso limitado a dados meteorológicos. Este sistema pode facilitar o manejo hídrico eficiente e o planejamento agrícola sustentável.

Palavras-chaves:

Estações meteorológicas de baixo custo; automação na agricultura; monitoramento ambiental; Penman-Monteith-FAO; dados meteorológicos.

ABSTRACT

This study presents the development and validation of a low-cost weather station based on the Arduino platform, designed to estimate reference evapotranspiration (ET_o) using the Hargreaves-Samani method. The experiment was conducted with accessible sensors, such as the DS18B20 temperature sensor, and features like an LCD display, SD memory, and Bluetooth communication. Data collection took place between December 5 and 15, 2024, and the results were compared with those obtained by an automatic weather station using the Penman-Monteith-FAO method as a reference. Validation was carried out using Willmott's concordance index, Pearson's correlation, and the "c" index, which showed a good correlation between the methods. This highlights the potential of the proposed system as an accessible and efficient alternative for estimating ET_o, especially in regions with limited access to meteorological data. This system can facilitate efficient water management and sustainable agricultural planning.

Keywords:

Low-cost weather stations; automation in agriculture; environmental monitoring; Penman-Monteith-FAO; meteorological data.

LISTA DE ILUSTRAÇÕES

Figura 1 -	Arduino MEGA 2560	14
Figura 2 -	Placa <i>protoboard</i>	15
Figura 3-	<i>Display</i> LCD	15
Figura 4 -	Módulo <i>Shield data Logger SD card</i> com bateria	16
Figura 5 -	Módulo sensor de temperatura DS18B20	16
Figura 6 -	Cartão SD 2GB	17
Figura 7 -	Módulo <i>Bluetooth</i> HC-06	17
Figura 8 -	<i>Layout</i> do aplicativo Arduino <i>Bluetooth Controller</i>	18
Figura 9 -	Processo de incorporação do projeto	19
Figura 10 -	Experimento montado	22
Gráfico 1 -	Relação entre os resultados de ETo dos modelos de HS e PM	24

LISTA DE ABREVIATURAS E SIGLAS

ETo –	Evapotranspiração de referência
HS –	Hargreaves-Samani
PM –	Penman-Monteith-FAO
LCD –	<i>Liquid Crystal Display</i>
LED –	<i>Light-Emitting Diode</i>
FEHAN –	Fazenda Experimental Professor Hamilton de Abreu Navarro
ICA –	Instituto de Ciências Agrárias
UFMG –	Universidade Federal de Minas Gerais
INMET –	Instituto Nacional de Meteorologia
EEPROM –	<i>Electrically Erasable Programmable Read-Only Memory</i>

SUMÁRIO

1 INTRODUÇÃO	9
2 REFERENCIAL TEÓRICO	11
_2.1 Evapotranspiração de referência	11
_2.2 Método de Hargreaves-Samani	11
_2.3 Estações meteorológicas	12
_2.4 Automação através de Arduino	12
3 MATERIAL E MÉTODOS	14
_3.1 Área de estudo	14
_3.2 Instrumentos utilizados	14
_3.3 Logica computacional e aquisição dos dados	18
_3.4 Analise e processamento de dados	20
4 RESULTADOS E DISCUSSÃO	22
5 CONCLUSÃO.....	26
REFERÊNCIAS	27

1 INTRODUÇÃO

A evapotranspiração desempenha papel vital no ciclo hidrológico, ajudando a redistribuir a umidade na atmosfera e contribuindo para a formação da precipitação. Esse processo combina a evaporação da água do solo e a transpiração das plantas, sendo essencial para entender quanta água é perdida para a atmosfera em uma área específica. Compreender esse fenômeno é fundamental para o planejamento e a gestão de recursos hídricos, para a agricultura, para a previsão climática e os estudos ambientais.

Na agricultura, a evapotranspiração é essencial para determinar as necessidades de irrigação das culturas, sendo uma ferramenta indispensável na gestão e no planejamento sustentável dos recursos hídricos e ambientais. Compreender a quantidade de água perdida em determinada área permite, através da estimativa de evapotranspiração, fazer a reposição da água de maneira eficaz, possibilitando o uso eficiente de recursos hídricos e práticas de conservação.

Em ecossistemas naturais, a evapotranspiração desempenha um papel crucial na regulação da disponibilidade de água para a vegetação e outros organismos, contribuindo para a preservação ambiental e o equilíbrio dos recursos hídricos. Além disso, influencia diretamente o balanço hídrico, ajudando a moderar as temperaturas locais e regionais, além de ser um fator indispensável para a previsão do tempo e do clima.

Segundo Lima Junior *et al.* (2016) a abordagem do método de Hargreaves-Samani (HS) é uma das ferramentas empregadas na determinação da evapotranspiração de referência (ET_o), principalmente em áreas onde os dados meteorológicos necessários para cálculos mais complexos podem ser escassos ou indisponíveis. Esse método se fundamenta em diversos parâmetros e, por meio da inserção de dados como radiação solar e temperaturas máximas e mínimas do ar, oferece uma maneira prática e relativamente simples para estimar a ET_o (Hargreaves; Samani, 1985).

O elevado custo das estações meteorológicas precisas dificulta sua adoção generalizada, ressaltando a necessidade de desenvolver dispositivos mais acessíveis e portáteis que permitam monitorar o clima de forma eficiente. A mobilidade desses equipamentos permitiria a coleta de dados em diversos ambientes, ampliando a capacidade de análise meteorológica. Entretanto, encontrar sensores de baixo custo que ofereçam precisão e confiabilidade ainda representa um desafio significativo. Sensores de alta precisão

tem seu preço consideravelmente alto, o que faz com que alternativas mais econômicas comprometam a qualidade dos dados coletados.

Para obter dados meteorológicos de forma acessível, a utilização do Arduino surge como uma alternativa eficiente e prática. De acordo com Frizzarin (2016), o Arduino é uma plataforma de prototipagem eletrônica e uma base acessível para desenvolver projetos eletrônicos controlados por microcontroladores, através de sua combinação de sensores, os quais são capazes de coletar dados ambientais e fornecê-los para o Arduino. Ele representa uma poderosa ferramenta para a automação moderna, sua combinação abre portas para a criação de diversos tipos de projetos, desde sistemas simples até os mais complexos (McRoberts, 2011).

Devido às limitações dos métodos tradicionais, como o método de Penman-Monteith-FAO (PM), que exige grande número de variáveis meteorológicas, a utilização do Arduino, em conjunto com o método HS, é uma solução prática e acessível para coletar dados climáticos e estimar a ETo. Portanto, este estudo propõe a utilização do Arduino para desenvolver um sistema capaz de calcular e estimar a ETo. Além de oferecer uma solução de baixo custo para pequenas propriedades agrícolas ou regiões com dados meteorológicos limitados, essa abordagem visa demonstrar a viabilidade e a precisão dessa metodologia.

Portanto, esse trabalho tem como objetivo desenvolver uma estação meteorológica automática, funcional e de baixo custo para estimar ETo. Além disso, busca estimar a ETo utilizando o método de HS e validar sua precisão por meio da correlação com o método de PM.

2 REFERENCIAL TEÓRICO

2.1 Evapotranspiração de referência

Segundo Rodrigues *et al.* (2013) em regiões áridas e semiáridas, onde a distribuição de água da chuva é irregular, a determinação da ETo é essencial para entender a necessidade hídrica e o momento adequado de se aplicar água na cultura. A ETo varia de região para região, depende unicamente das condições climáticas e representa a demanda hídrica do local (Mantovani *et al.*, 2007).

Para Souza *et al.* (2014) a compreensão da ETo é fundamental para a gestão hídrica, necessitando de estudos, ajustes e avaliações para o melhor manejo hídrico, visando a utilização correta dos recursos hídricos. Métodos de estimativas de ETo, mesmo os mais simples, se demonstram eficazes na elaboração de gestão hídrica.

A ETo desempenha papel essencial em diversas áreas, sua utilização é importante nos estudos sobre a classificação climática, disponibilidade e manejo hídrico de determinada região (Souza *et al.*, 2011). Portanto, compreender a ETo é essencial para tomar decisões sobre o uso sustentável da água e definir uma gestão eficiente para utilização dos recursos hídricos.

2.2 Método de Hargreaves-Samani

Segundo Rodrigues *et al.* (2013), o método considerado padrão para estimar a ETo é o método de PM, sua complexidade se dá pelo fato de exigir muitas variáveis meteorológicas, o que nem sempre é viabilizado, dependendo da região. Devido a essa dificuldade, podem ser utilizados métodos alternativos de estimativa de ETo, que requerem menos dados e são mais simples de se aplicar.

De acordo com Hargreaves e Samani (1985), ao se considerar os problemas associados à disponibilidade e confiabilidade dos dados climatológicos no mundo, o método de análise das temperaturas é sugerido como o método mais simples e prático para se estimar a ETo das culturas.

Por conta disso, o método de HS se torna uma ferramenta acessível pela sua simplicidade. Este método é aplicado para a obtenção de ETo e sua metodologia simplificada

disponibiliza uma equação em que são necessários apenas os dados meteorológicos de temperaturas mínimas e máximas do ar (Carvalho; Delgado, 2016).

2.3 Estações meteorológicas

Segundo Mantovani *et al.* (2007), existem diversos modelos de estações meteorológicas, que podem ser manuais ou automáticas, sendo que as automáticas se sobressaem pela variedade de números de medidas e facilidade operacional. Isso permite uma análise mais precisa e eficiente das variáveis meteorológicas necessárias para a estimativa de ETo. Ao aplicar essas estações em nível de fazenda, é importante considerar a facilidade de instalação, operação e manutenção, além de contar com profissionais qualificados para a instalação e treinamento dos usuários.

Segundo Gonçalves *et al.* (2009), quando se trata do cálculo de ETo, o método padrão da FAO utiliza dados como temperatura (T), umidade relativa do ar (UR), radiação solar (Rs) e velocidade do vento (V). No entanto, muitos produtores rurais não têm acesso a esses dados, o que torna necessário o uso de métodos mais simples, como o de HS.

O custo elevado de equipamentos meteorológicos pode ser um obstáculo significativo para muitos produtores rurais (Elias *et al.*, 2014). A necessidade de baratear esses equipamentos é crucial para tornar a tecnologia mais acessível e permitir que mais agricultores possam se beneficiar dos dados meteorológicos precisos.

2.4 Automação através de Arduino

A automação é uma tecnologia que vem para abordar a redução da intervenção humana na execução de processos, por meio de circuitos eletrônicos, algoritmos ou motores para conceder autonomia a dispositivos manuais. Essa tecnologia está cada vez mais presente, com equipamentos cotidianos integrando tecnologias para agilizar o uso e aumentar a segurança dos usuários (Silva *et al.*, 2022).

De acordo com Frizzarin (2016), o Arduino permite a integração de sensores que coletam dados ambientais e os fornecem ao sistema Arduino. Essa plataforma se destaca por sua abordagem acessível e interativa, combinando software e hardware para criar sistemas que podem detectar e responder ao ambiente ao seu redor. Além disso, o Arduino se alinha com a filosofia *Makers* ou “faça-você-mesmo”, incentivando a criatividade e a inovação ao

possibilitar o desenvolvimento de soluções personalizadas e acessíveis para diversos tipos de projetos.

O Arduino é uma plataforma bastante utilizada hoje em dia, podendo coletar dados por meio de sensores externos, medir variáveis do ambiente e realizar cálculos numéricos, apresentando os resultados por meio de uma interface de programação (Cardozo; Ferreira, 2023).

De acordo com McRoberts (2011), o Arduino é um pequeno computador que possibilita ao usuário programar para lidar com entradas e saídas entre o dispositivo e componentes externos conectados a ele. Ele é uma plataforma de computação física ou embarcada, o que significa que pode interagir com o ambiente por meio de *hardware* e *software*. Com ele, é possível desenvolver objetos interativos independentes ou conectá-lo a um computador, rede ou até mesmo à Internet.

A estimativa da evapotranspiração é essencial para a gestão hídrica, especialmente em regiões onde há uma irregularidade das chuvas. Embora o método de PM seja o padrão para a estimar a ETo, sua complexidade de dados limita sua aplicação. Diante disso, o método de HS surge como uma alternativa acessível, pela sua necessidade reduzida de dados. No entanto, a obtenção de dados meteorológicos ainda é um desafio, devido ao alto custo das estações e sensores, dificultando sua adoção por pequenos agricultores.

Uma solução viável é a automação por Arduino, a qual permite desenvolver uma estação meteorológica de baixo custo, sendo capaz de coletar, armazenar e processar dados meteorológicos de forma eficiente. Com isso, o presente estudo busca implementar um sistema automatizado utilizando o método de HS para estimar a ETo, validando os resultados e viabilizando uma alternativa acessível e funcional.

3 MATERIAL E MÉTODOS

3.1 Área de estudo

A pesquisa foi realizada na Fazenda Experimental Professor Hamilton de Abreu Navarro (FEHAN), do Instituto de Ciências Agrárias (ICA) da Universidade Federal de Minas Gerais (UFMG), no Campus Regional de Montes Claros - MG. A FEHAN abrange uma área física de 232 hectares. O município está localizado na região norte de Minas Gerais e seu clima é classificado como tropical, com uma precipitação média anual de 869 mm (*Climate Data*, 2021).

3.2 Instrumentos utilizados

Para a realização do projeto foram necessários os seguintes equipamentos:

Arduino MEGA 2560 (Figura 1), com as seguintes especificações:

Modelo: Mega 2560;

Microcontrolador: ATmega2560;

Tensão de operação: 5;

Capacidade de memória *flash*: 256 KB;

Capacidade SRAM: 8 KB;

Capacidade EEPROM: 4 KB;

Dimensões: 110 x 50 x 10 mm.

Figura 1 – Arduino MEGA 2560



Fonte: Acervo pessoal do autor, 2024

Placa *proto*board (Figura 2), com as seguintes especificações:

Furos: 830 com barras de distribuição demarcadas;

Faixa de temperatura: -20 a 80°C;

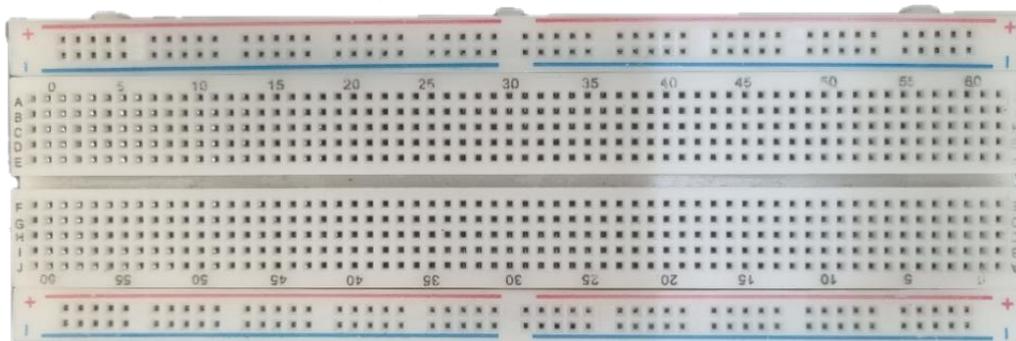
Terminais e condutores: 0,3 a 0,8 mm;

Resistência de isolamento: 100 M Ω min;

Tensão máxima: 500v AC por min;

Dimensões: 165 x 57 x 10 mm.

Figura 2 – Placa *proto*board



Fonte: Acervo pessoal do autor, 2024

Display LCD 16x2 com I2C integrado (Figura 3), com as seguintes especificações:

Display: LCD de 2 linhas e 16 colunas;

Módulo: I2C permite conexão rápida e fácil;

Tensão de operação: 5V;

Controlador: HD44780 ou compatível;

Interface: I2C;

Dimensões do *display*: 84mm x 44mm.

Figura 3 – *Display* LCD



Fonte: Acervo pessoal do autor, 2024

Módulo *Shield Data Logger SD card* com bateria (Figura 4), com as seguintes especificações:

Modelo: XD-05;

Rtc ds1307 integrado (relógio de tempo real);

Slot para cartão SD de memória (cartão SD);

Slot para bateria (cr1220);

Circuito de nível de transmissão: 3,3v;

Área disponível para prototipagem;

Dimensões: 68 x 53 x 10 mm (ignorando-se os pinos).

Figura 4 – Módulo *Shield Data Logger SD card* com bateria



Fonte: Acervo pessoal do autor, 2024

Módulo sensor de temperatura DS18B20 (Figura 5), com as seguintes especificações:

Modelo: DS18B20;

Tensão de operação: 3 e 5,5 V;

Faixa de medição: -55 a 125°C;

Precisão: $\pm 0.5^\circ\text{C}$ de -10°C a $+85^\circ\text{C}$;

Dimensão da ponta: 6 x 50 mm;

Ponta de aço inoxidável.

Figura 5 – Módulo sensor de temperatura DS18B20



Fonte: Acervo pessoal do autor, 2024

Cartão SD 2GB (Figura 6), com especificação:

Modelo: Micro SD

Capacidade: 2 GB.

Figura 6 – Cartão SD de 2GB



Fonte: Acervo pessoal do autor, 2024

Módulo *Bluetooth* HC-06 (Figura 7), com especificações:

Modelo: *Bluetooth* HC-06;

Protocolo *Bluetooth*: v2.0+EDR;

Frequência: 2,4GHz Banda ISM;

Tensão de alimentação: 3,6 - 6 Vdc;

Tensão de comunicação: 3,3 Vdc;

Corrente: antes do pareamento 25 mA; depois do pareamento 8 mA;

Dimensões: 15 x 5 x 38 mm.

Figura 7 – Módulo *Bluetooth* HC-06



Fonte: Acervo pessoal do autor, 2024

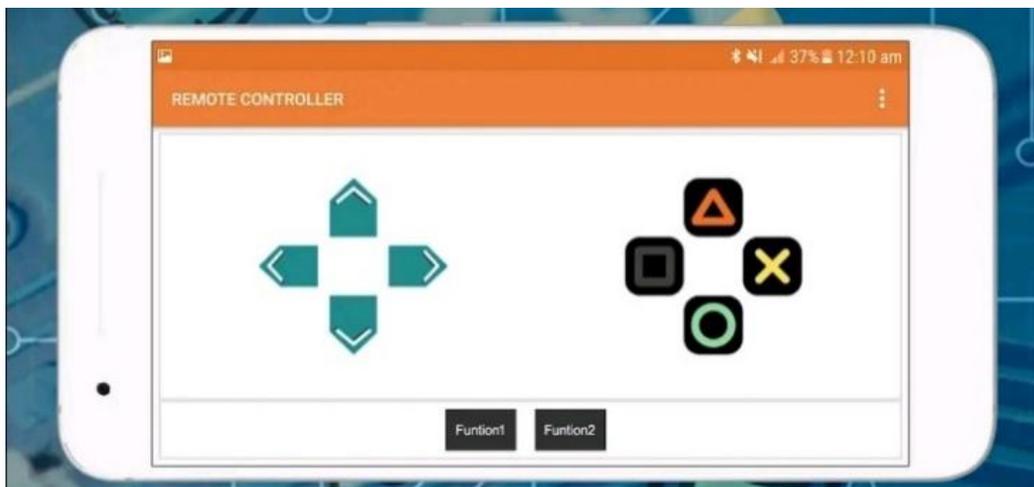
E o aplicativo de celular para se conectar com equipamento *Arduino Bluetooth Controller* (Figura 8), com especificações:

Nome: *Arduino Bluetooth Controller*;

Fornecedor: AppsValley™;

Onde encontrar: Play store.

Figura 8 - *Layout* utilizado do aplicativo *Arduino Bluetooth Controller*



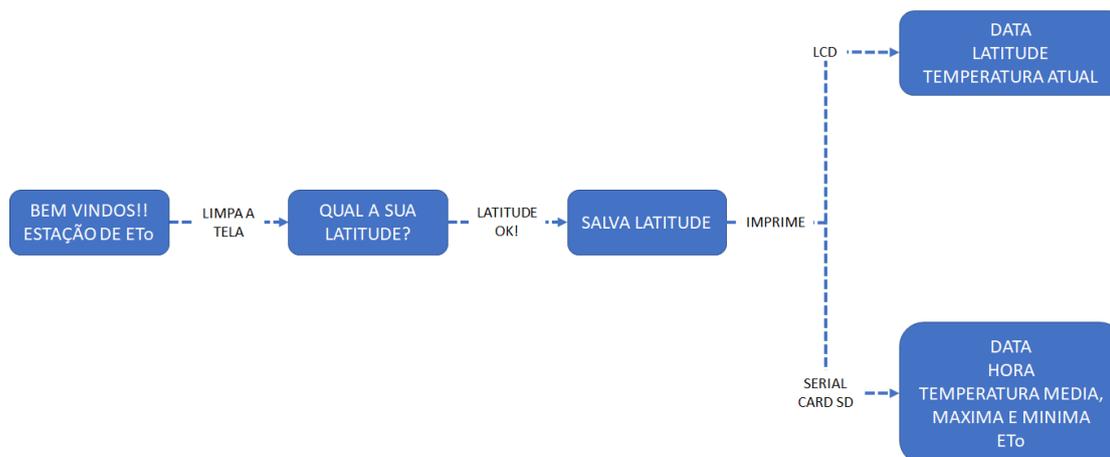
Fonte: MyValley Apps, 2024

3.3 Lógica computacional e aquisição dos dados

Inicialmente, realizou-se a montagem do equipamento para a obtenção dos dados de evapotranspiração, com a utilização de um microcontrolador Arduino MEGA 2560, auxiliado por sensor de temperatura, *protoboard*, *display* e módulos. Depois, foi montada a estrutura do equipamento para mensurar a ETo em campo.

O equipamento foi programado para que, ao ser iniciado, o *display* LCD apresentasse uma mensagem “BEM VINDO!! ESTAÇÃO DE ETo” e, em seguida, a tela é limpa e vem a mensagem de “QUAL A SUA LATITUDE?”, e por meio do aplicativo de celular “*Arduino Bluetooth Controller*” é possível inserir a latitude desejada, sendo esse o único dado manual necessário para o funcionamento do equipamento, já que os dados variam de região para região. O programa foi desenvolvido para garantir o funcionamento contínuo, mesmo em casos de quedas de energia, caso ocorra, o código retornará com a última latitude inserida. O fluxograma apresentado Figura 9 descreve o processo de incorporação do projeto.

Figura 9 – Processo de incorporação do projeto



Fonte: Acervo pessoal do autor, 2024

O módulo *Bluetooth* foi utilizado para enviar comandos para o Arduino. Os botões de seta para cima aumentam o valor da latitude, enquanto os de seta para baixo diminuem. O botão quadrado marrom confirma a latitude selecionada, que será armazenada para cálculo da ETo. O botão círculo verde liga o *display*, enquanto o botão 'X' amarelo desliga. O botão triangular vermelho reinicia o programa, permitindo ajuste em caso de alteração de local ou latitude.

O LCD é responsável por exibir as mensagens iniciais, confirmar a latitude inserida e facilitar a consulta de informações importantes durante o uso do sistema. Entre os dados exibidos estão a data, a latitude configurada e a temperatura atual. Essa funcionalidade garante praticidade e uma interface amigável para o usuário, centralizando as informações essenciais de forma clara e acessível.

O sensor de temperatura DS18B20 foi o responsável por capturar as temperaturas mínimas e máximas diárias com precisão. Para assegurar a confiabilidade dos dados, o sensor foi instalado a uma altura de 2 metros, em local sombreado, garantindo a coleta adequada dos dados de temperatura.

O Módulo *Shield Data Logger SD card* com bateria foi utilizado para garantir a coleta e o armazenamento de dados, mesmo em caso de queda de energia ou se o equipamento não estiver conectado à tomada. Isso evitará a perda de dados e permitirá que seja gerado um checklist que imprimirá as informações diárias de data, hora, temperaturas mínimas, máximas e média e ETo.

A programação do projeto foi realizada no aplicativo Arduino IDE utilizando a linguagem de programação C/C++. O código é fundamentado no método de HS, que se

baseia em uma equação simplificada para obtenção de ETo utilizada em regiões áridas e semiáridas, adequando-se perfeitamente à área de execução do experimento. Os dados foram coletados diariamente.

O método de HS é dado pela Equação 1 (Hargreaves; Samani, 1985)

$$E_{To} = 0,0023 * Ra * (T_{MED} + 17,8) * (T_{MAX} - T_{MIN})^{0,5} \quad \text{Equação 1}$$

Em que:

Ra – Radiação solar no topo da atmosfera (mm/dia), valor tabelado o qual é determinado em função da latitude e da época do ano (Anexo 1);

T – Temperatura média do ar (°C);

T_{MAX} – Temperatura máxima do ar (°C);

T_{MIN} - Temperatura mínima do ar (°C).

3.4 Análise e processamento de dados

Para validar a pesquisa, os dados coletados foram comparados com os dados meteorológicos obtidos da estação automática do Instituto Nacional de Meteorologia (INMET), código A506, localizada em Montes Claros, nas coordenadas de Latitude -16,686389, Longitude -43,843889 e altitude de 645,87 metros, instalada em 18/12/2002 e pela estação manual código 83437, localizada em Montes Claros, nas coordenadas de Latitude -16.686389, Longitude -43.843889 e Altitude de 645.87 metros (INMET).

O método de PM foi utilizado como referência para comparação dos resultados do método de HS. A análise considerou os dados meteorológicos coletados nos mesmos dias do experimento, incluindo insolação total (h), temperaturas máxima, mínima e média (°C), velocidade média do vento (m/s), umidade relativa (%) e pressão atmosférica (mB). Com base nessas variáveis, foi estimada a ETo diária pelo método de PM, amplamente reconhecido como padrão-ouro na estimativa de evapotranspiração (ALLEN *et al.*, 1998), devido à sua alta precisão em diferentes condições ambientais.

Para a análise comparativa utilizou-se o Índice de Concordância de Willmott (Willmott *et al.*, 1985) (Equação 2), o coeficiente de correlação de Pearson (Equação 3) e o índice “c” para avaliar o desempenho do método, conforme proposto por Camargo e

Sentelhas (1997) (Equação 4). A interpretação do desempenho das médias diárias de ETo seguiu os critérios definidos por Camargo e Sentelhas (1997), detalhados no Anexo 2.

$$d = 1 - \frac{\sum_{i=1}^n (P_i - O_i)^2}{\sum_{i=1}^n [(P_i - O) + (O_i - O)]^2} \quad \text{Equação 2}$$

Em que:

d – Índice de concordância ou ajuste;

P_i – ETo obtida pelo método de HS (mm/dia);

O_i – ETo obtida pelo método de PM (mm/dia);

O – Média dos valores de ETo obtido pelo método de PM (mm/dia);

n – Número de observações.

$$r = \frac{\sum_{i=1}^n (X_i - X_m) * (Y_i - Y_m)}{\sqrt{\sum_{i=1}^n (X_i - X_m)^2} * \sqrt{\sum_{i=1}^n (Y_i - Y_m)^2}} \quad \text{Equação 3}$$

Em que:

r – Coeficiente de correlação de Pearson;

X_i – ETo obtida pelo método de HS (mm/dia);

X_m – Média da ETo obtida pelo método de HS (mm/dia);

y_i – ETo obtida pelo método de PM (mm/dia);

y_m – Média ETo obtida pelo método de PM (mm/dia).

$$c = r * d \quad \text{Equação 4}$$

Em que:

c – Coeficiente de confiança ou desempenho;

r – Coeficiente de correlação de Pearson;

d – Índice de concordância de Willmott.

4 RESULTADOS E DISCUSSÃO

O experimento foi desenvolvido atendendo plenamente aos objetivos inicialmente estabelecidos. A programação (Apêndice 1), em conjunto com os sensores, funcionou perfeitamente. Destacou-se o sensor de temperatura DS18B20, que se mostrou altamente eficiente ao fornecer de forma consistente os valores diários de temperaturas máxima, mínima e média.

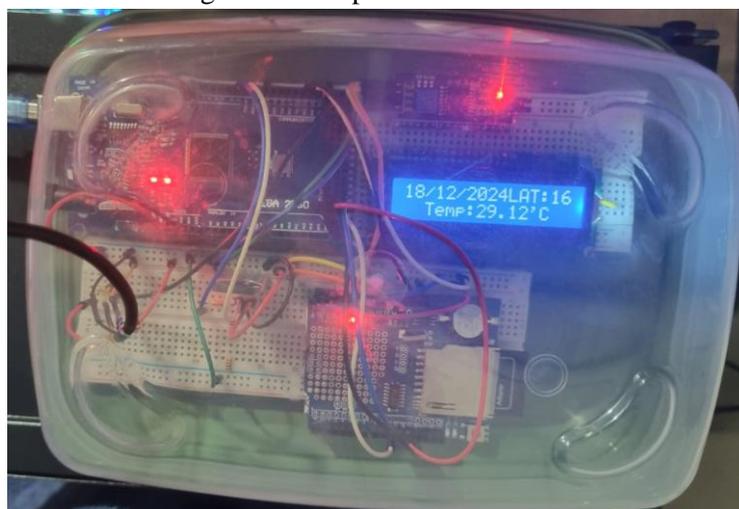
A comunicação via *Bluetooth* mostrou-se estável e confiável, permitindo a execução de comandos remotos com exatidão. Entre as funcionalidades acessadas por meio do *Bluetooth* destacaram-se: seleção de latitude, controle do LED do *display* LCD (acender e apagar) e reinicialização do experimento.

A funcionalidade do armazenamento na EEPROM (*Electrically Erasable Programmable Read-Only Memory*) garantiu que os dados essenciais, como a latitude, fossem preservados em caso de perda de energia. Esse recurso permitiu que o sistema retomasse a operação de onde parou, sem perda de informações críticas, o que foi comprovado pelos resultados obtidos nos testes.

O módulo *Shield Data Logger SD card* com bateria forneceu informações precisas de data e hora em tempo real, sem qualquer variação ou erro identificado durante as coletas. Esses dados foram gravados no cartão SD e no monitor serial de forma confiável, complementando o registro das medições realizadas pelo sistema.

De maneira geral, os componentes integrados ao projeto atenderam às expectativas, comprovando a eficácia do experimento, tanto no monitoramento de dados ambientais quanto na interação remota e no armazenamento seguro das informações coletadas.

Figura 10 – Experimento montado



Fonte: Acervo pessoal do autor, 2024

Os valores obtidos indicaram uma boa concordância e correlação entre os modelos de HS e PM. O índice de concordância de Willmott (d) apresentou um valor de 0,71, evidenciando uma concordância moderada, o que reflete uma concordância moderada, mas ainda aquém do ideal para uma modelagem perfeitamente ajustada, em que o valor de 'd' se aproximaria de 1 (Camargo; Sentelhas, 1997).

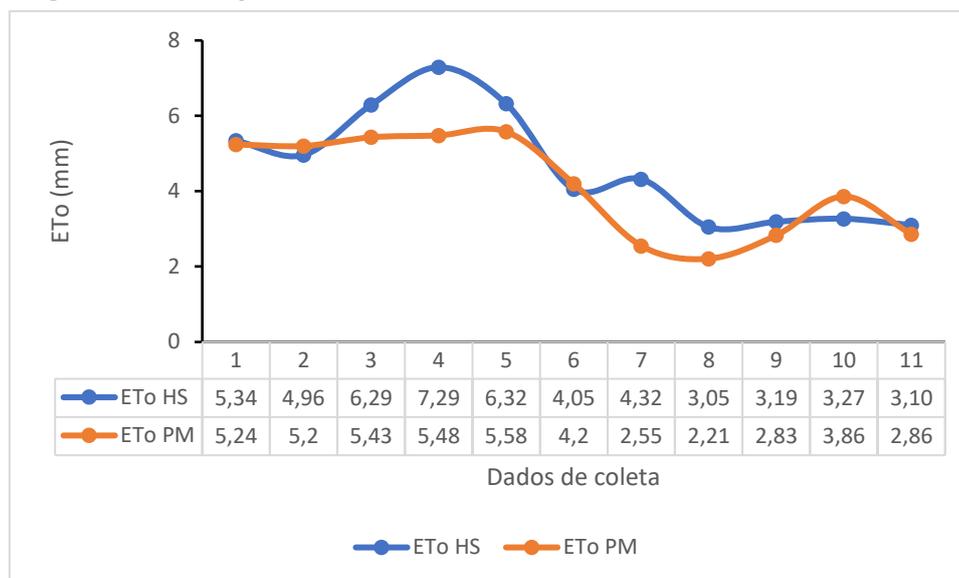
O coeficiente de correlação de Pearson (r) foi de 0,86, indicando uma forte correlação linear entre as estimativas de ETo obtidas pelo modelo HS, utilizando o experimento desenvolvido e os valores de referência do modelo PM, calculado com dados meteorológicos do INMET. Esse alto valor de 'r' indica que as tendências de variação entre os dois modelos estão bem alinhadas (Paranhos *et al.*, 2014).

Por sua vez, o índice "c", que combina os valores de 'd' e 'r', resultou em 0,61, apontando para a necessidade de cautela na aplicação do modelo de HS com os dados obtidos pelo experimento. Esse índice classifica a confiabilidade geral da modelagem como mediana, sugerindo que o modelo HS, embora funcional, pode exigir ajustes adicionais para alcançar maior precisão em determinadas situações (Camargo; Sentelhas, 1997).

Os resultados do experimento, conduzido entre os dias 05 e 15 de dezembro de 2024, demonstram a confiabilidade dos dados coletados ao longo de 11 dias consecutivos. A comparação com os registros da estação automática confirmou a precisão das medições, assegurando a validade das informações obtidas. Os dados foram coletados a uma taxa de uma amostra por segundo. A cada hora, calculava-se a média horária, e, ao final do dia, obtinha-se a média diária.

O Gráfico 1 apresenta a relação gráfica entre os valores de ETo estimados pelo experimento desenvolvidos com base no modelo de HS (Apêndice 2) e os valores calculados pelo modelo de PM.

Figura 11 – Relação entre os resultados de ETo dos modelos de HS e PM



Fonte: Acervo pessoal do autor, 2024

Pode-se observar que muitos pontos estão próximos à linha de concordância 1:1, indicando boa correspondência geral entre os modelos. Entretanto, em condições extremas, como dias de altas amplitudes térmicas ou baixos níveis de radiação solar devido à cobertura de nuvens, os desvios tornam-se mais evidentes.

Algumas interferências externas podem causar erros nas leituras do sensor de temperatura. A interferência eletromagnética, proveniente de motores, rádios, celulares e outras fontes de sinais eletromagnéticos, pode afetar o funcionamento do sensor. Além disso, a influência de superfícies próximas pode distorcer as medições, uma vez que o sensor pode captar o calor irradiado por objetos ao redor, alterando a leitura da temperatura ambiente.

Outro ponto a se considerar, são as limitações do modelo HS, que utiliza um conjunto reduzido de variáveis (temperaturas máxima, mínima, média e radiação solar estimada) em comparação ao modelo PM, que incorpora variáveis adicionais, como umidade relativa do ar e velocidade do vento (Allen et al., 1998).

Entretanto, o experimento demonstrou ser uma solução aplicável e acessível para a estimativa de ETo em locais com acesso limitado a dados meteorológicos. A utilização de componentes simples estabeleceu o experimento como uma alternativa viável para regiões onde estações meteorológicas convencionais são economicamente inviáveis.

Por fim, como recomendação para trabalhos futuros, é preciso validar o experimento em diferentes condições climáticas e locais, o que permitirá consolidar sua robustez como ferramenta para gestão hídrica e planejamento agrícola. Essa solução apresenta grande

potencial para impactar positivamente pequenos agricultores e iniciativas de manejo sustentável da água.

5 CONCLUSÃO

O experimento desenvolvido demonstrou ser uma solução prática e acessível para a estimativa da ETo em regiões com acesso limitado a dados meteorológicos. A utilização da plataforma Arduino, combinada com sensores de fácil aquisição e manuseio, mostrou-se eficaz na coleta de dados de temperatura, na interação remota por *Bluetooth* e no armazenamento seguro das informações. Apesar das limitações inerentes ao modelo de HS, como o uso de um conjunto reduzido de variáveis, os resultados indicaram boa concordância com o método padrão PM, confirmando a viabilidade do experimento.

Os índices de desempenho obtidos, como o coeficiente de correlação de Pearson e o índice de concordância de Willmott evidenciaram a capacidade do experimento em capturar tendências gerais na variação da ETo.

REFERÊNCIAS

- ALLEN, R. G. *et al.* **Crop evapotranspiration: Guidelines for computing crop water requirements.** Rome: FAO Irrigation and drainage paper 56, 1998.
- CAMARGO, A. P. de; SENTELHAS, P. C. Avaliação do desempenho de diferentes métodos de estimativa da evapotranspiração potencial no Estado de São Paulo, Brasil. **Revista Brasileira de Agrometeorologia**, v. 5, n. 1, p. 89-97, 1997. Disponível em: [http://www.leb.esalq.usp.br/agmfacil/artigos/artigos_sentelhas_1997/1997_RBAGro_5\(1\)_89-97_ETPM%E9todosSP.pdf](http://www.leb.esalq.usp.br/agmfacil/artigos/artigos_sentelhas_1997/1997_RBAGro_5(1)_89-97_ETPM%E9todosSP.pdf). Acesso em: 25 jun. 2024.
- CAMARGO, A. P. de; CAMARGO, M. B. P. Teste de uma equação simples para estimativa da evapotranspiração potencial baseada na radiação solar extraterrestre e na temperatura do ar. In: Sociedade Brasileira de Agrometeorologia (org.), **Anais... 3º Congresso Brasileiro de Agrometeorologia**, (p. 229-244). Apresentado no 3º Congresso Brasileiro de Agrometeorologia, Campinas, SBAgro, 1983. Disponível em: <https://sbagro.org/files/biblioteca/3927.pdf>. Acesso em: 19 jun. 2024.
- CARDOZO, L. F.; FERREIRA, A. F. Automação residencial com Arduino, **Revista Ibero-Americana de Humanidades, Ciências e Educação – REASE**, v. 9, n. 2, P. 622-634, 2023. Disponível em: <https://periodicorease.pro.br/rease/article/view/8522/3347>. Acesso em: 21 jun. 2024.
- CARVALHO, R. L. da S.; DELGADO, A. R. S. Estimativas da evapotranspiração de referência do município de Ariquemes (RO) utilizando os métodos Penman-Monteith-FAO e Hargreaves-Samani, **Revista Brasileira de Agricultura Irrigada - RBAI**, v. 10, n. 6, p. 1038-1048, 2016. Disponível em: https://www.inovagri.org.br/revista/index.php/rbai/article/view/497/pdf_311. Acesso em: 14 jun. 2024.
- CLIMA MONTES CLAROS. **Climate Data**. Disponível em: <https://pt.climate-data.org/america-do-sul/brasil/minas-gerais/montes-claros-2886/>. Acesso em: 25 jun. 2024.
- Elias, A. A. de A. *et al.* ARDWEATHER: UMA ESTAÇÃO METEOROLÓGICA BASEADA NO ARDUINO E EM WEB SERVICES RESTFUL. **Health and Environment World Congress**, (p. 44-48). Apresentado no XIV Safety, Health and Environment World Congress, Cubatão, 2014. Disponível em: https://www.researchgate.net/publication/281525465_ARDWEATHER_UMA_ESTACAO_METEOROLOGICA_BASEADA_NO_ARDUINO_E_EM_WEB_SERVICES_RESTFUL/citation/download, Acesso em: 01 fev. 2025.
- FRIZZARIN, F. B. **Arduino: guia para colocar suas ideias em prática.** São Paulo: Casa do Código, 2016.
- GONÇALVES, F. M. *et al.* Comparação de métodos da estimativa da evapotranspiração de referência para o município de Sobral - CE, **Revista Brasileira de Agricultura Irrigada - RBAI**, v. 3, n. 2, p. 71-77, 2009. Disponível em: https://www.inovagri.org.br/revista/index.php/rbai/article/view/26/pdf_19. Acesso em: 14 jun. 2024.

HARGREAVES, G. H.; SAMANI, Z. A. Reference crop evapotranspiration from ambient air temperature. **American society of agricultural engineers**, v. 1, n. 2, p. 96-99, 1985.

Disponível em:

https://www.researchgate.net/publication/247373660_Reference_Crop_Evapotranspiration_From_Temperature. Acesso em: 15 jun. 2024.

INMET. INSTITUTO NACIONAL DE METEOROLOGIA. Disponível em:

<https://portal.inmet.gov.br/paginas/catalogoaut>. Acesso em: 25 jun. 2024.

LIMA JUNIOR, J. C *et al.* Parametrização da equação de Hargreaves e Samani para estimativa da evapotranspiração de referência no Estado do Ceará, Brasil. **Revista Ciência Agronômica**, v. 47, n. 3, p. 447-454, 2016. Disponível em:

<https://www.scielo.br/j/rca/a/smv7gdjXKKmwHT8fsJ8GSYb/?lang=pt#>. Acesso em: 17 jun. 2024.

MANTOVANI, E. C. *et al.* **Irrigação**: princípios e métodos. Viçosa: Editora UFV, 2007.

MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec Editora Ltda, 2011.

MyValley Apps. **Arduino Bluetooth Controller**. 2019. Google Play. Disponível em:

<https://play.google.com/store/apps/details?id=com.appsvalley.bluetooth.arduinocontroller>. Acesso em: 24 jun. 2024.

PARANHOS, R *et al.* Desvendando os Mistérios do Coeficiente de Correlação de Pearson: o Retorno. **Leviathan (São Paulo)**, São Paulo, Brasil, n. 8, p. 66-95, 2014. Disponível em:

<https://www.revistas.usp.br/leviathan/article/view/132346>. Acesso em: 3 jan. 2025.

RODRIGUES, R. R *et al.* Estimativa da demanda hídrica do cafeeiro conilon, variedade robusta tropical, em comparação com a evapotranspiração obtida pelo método de Hargreaves & Samani. **Revista Enciclopédia Biosfera**, V. 09, n. 16, 1055-1066, 2013.

Disponível em:

<https://www.conhecer.org.br/enciclop/2013a/agrarias/ESTIMATIVA%20DA%20DEMANDA.pdf>. Acesso em: 12 jun. 2024.

SILVA, G. F. T. L *et al.* O uso da automação por Arduino no controle da proliferação de microrganismos. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, v. 5, n. 12, p. 160-185, 2022. Disponível em:

<https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/proliferao-de-microrganismos>. Acesso em: 6 jun. 2024.

SOUZA, J. M. de *et al.* Comparação entre métodos de estimativa da evapotranspiração de referência em duas localidades do espírito santo. **Revista Brasileira de Agricultura Irrigada - RBAI**, v. 8, n. 2, p. 114-126, 2014. Disponível em:

https://inovagri.org.br/revista/index.php/rbai/article/view/225/pdf_165. Acesso em: 14 jun. 2024.

SOUZA, L. P. de *et al.* Comparação de métodos da estimativa da evapotranspiração de referência para o município de Cruzeiro do Sul - Acre. **Revista Enciclopédia Biosfera**, V. 07, n. 12, 2011. Disponível em:

<https://www.conhecer.org.br/enciclop/2011a/agrarias/Comparacao%20de%20metodos.pdf>. Acesso em: 12 jun. 2024.

WILLMOTT, C. J. *et. al.* Statistics for the Evaluation and Comparison of Models. **Journal of Geophysical Research Atmospheres**, v. 90, n. C5, p. 8995-9005, 1985. Disponível em: https://www.researchgate.net/publication/259475985_Statistics_for_the_Evaluation_and_Comparison_of_Models. Acesso em: 25 jun. 2024.

APÊNDICE 1 – Código do experimento

```

//Projeto TCC
//Estação de evapotranspiração
//Warley Menezes Barros

//Bibliotecas
#include <OneWire.h> //Sensor temperatura
#include <DallasTemperature.h> //Sensor DS18B20
#include <LiquidCrystal_I2C.h> //Display LCD I2C
#include <SD.h> //SD
#include <SoftwareSerial.h> //comunicação do serial
#include <Wire.h> //I2C, usada pelo RTC e LCD
#include <RTCLib.h> //Data e hora
#include <avr/wdt.h> //Reset o sistema
#include <EEPROM.h> //Salva latitude

//Configurações de Temperatura
#define ONE_WIRE_BUS 22 //Define o pino digital
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

float tempMax = -127.0; //Variável para armazenar a temperatura máxima
float tempMin = 127.0; //Variável para armazenar a temperatura mínima
unsigned long lastReadingTime = 0; //Variável para o tempo da última
leitura de temperatura
const unsigned long readingInterval = 1000; //Intervalo entre leituras em
milissegundos (24 segundos)
const int totalReadingsPerHour = 3600; //Número total de leituras por hora
float tempTotal = 0.0; //Soma total das leituras de temperatura para cálculo da média
int readingsCount = 0; //Contador de leituras realizadas

```

```

float tempMed = 0.0; //Variável para armazenar a temperatura média

//LCD
LiquidCrystal_I2C lcd(0x27, 16, 2); //configura o LCD com endereço I2C

//Cartão SD
const int chipSelect = 53; //Define o pino 53 para seleção do cartão SD
File arquivo; //Variável para manipulação de arquivos no SD

int state = 0; //Comunicação Bluetooth

//Latitude
int latitude = 0; //Variável para armazenar a latitude
bool latitudeConfirmed = false; //Confirmação se a latitude foi configurada
const int LATITUDE_ADDRESS = 0; //Endereço de memória EEPROM onde a latitude
é salva

//Data e hora
RTC_DS3231 rtc; //Instancia o módulo RTC DS3231
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"}; //Array com nomes dos dias da semana

// Tabela de Ra (radiação solar) para latitudes de 0 a 70
const float Ra[36][12] PROGMEM = { /* valores da tabela Ra */
{14.5, 15.0, 15.2, 14.7, 13.9, 13.4, 13.5, 14.2, 14.9, 14.9, 14.6, 14.3}, //Latitude 0
{14.8, 15.2, 15.2, 14.5, 13.6, 13.0, 13.2, 14.0, 14.8, 15.0, 14.8, 14.6}, //Latitude 2
{15.0, 15.3, 15.1, 14.3, 13.3, 12.7, 12.8, 13.7, 14.7, 15.1, 15.0, 14.9}, //Latitude 4
{15.3, 15.4, 15.1, 14.1, 13.0, 12.6, 12.5, 13.5, 14.6, 15.1, 15.2, 15.1}, //Latitude 6
{15.6, 15.6, 15.0, 14.0, 12.7, 12.0, 12.2, 13.2, 14.5, 15.2, 15.4, 15.4}, //Latitude 8
{15.9, 15.7, 15.0, 13.8, 12.4, 11.6, 11.9, 13.0, 14.4, 15.3, 15.7, 15.7}, //Latitude 10

```

{16.1, 15.8, 14.9, 13.5, 12.0, 11.2, 11.5, 12.7, 14.2, 15.3, 15.8, 16.0}, //Latitude 12
 {16.3, 15.8, 14.9, 13.2, 11.6, 10.8, 11.1, 12.4, 14.0, 15.3, 15.9, 16.2}, //Latitude 14
 {16.5, 15.9, 14.8, 13.0, 11.3, 10.4, 10.8, 12.1, 13.8, 15.3, 16.1, 16.4}, //Latitude 16
 {16.7, 15.9, 14.7, 12.7, 10.9, 10.0, 10.4, 11.8, 13.7, 15.3, 16.2, 16.6}, //Latitude 18
 {16.7, 16.0, 14.5, 12.4, 10.6, 9.6, 10.0, 11.5, 13.5, 15.3, 16.4, 16.8}, //Latitude 20
 {16.9, 16.0, 14.3, 12.0, 10.2, 9.1, 9.6, 11.1, 13.1, 15.2, 16.5, 17.0}, //Latitude 22
 {16.9, 15.9, 14.1, 11.7, 9.8, 8.6, 9.1, 10.7, 13.1, 15.1, 16.5, 17.1}, //Latitude 24
 {17.0, 15.9, 13.9, 11.4, 9.4, 8.1, 8.7, 10.4, 12.8, 15.0, 16.6, 17.3}, //Latitude 26
 {17.1, 15.8, 13.7, 11.1, 9.0, 7.8, 8.3, 10.0, 12.6, 14.9, 16.6, 17.5}, //Latitude 28
 {17.2, 15.7, 13.5, 10.8, 8.5, 7.4, 7.8, 9.6, 12.2, 14.7, 16.7, 17.6}, //Latitude 30
 {17.2, 15.6, 13.3, 10.5, 8.1, 7.0, 7.4, 9.2, 11.9, 14.6, 16.7, 17.6}, //Latitude 32
 {17.2, 15.5, 13.0, 10.1, 7.6, 6.6, 7.0, 8.8, 11.6, 14.4, 16.7, 17.6}, //Latitude 34
 {17.3, 15.4, 12.7, 9.7, 7.2, 6.1, 6.5, 8.3, 11.3, 14.3, 16.6, 17.7}, //Latitude 36
 {17.3, 15.2, 12.4, 9.3, 6.7, 5.6, 6.0, 7.9, 11.0, 14.1, 16.6, 17.7}, //Latitude 38
 {17.3, 15.1, 12.1, 8.9, 6.3, 5.1, 5.5, 7.5, 10.6, 13.9, 16.5, 17.8}, //Latitude 40
 {17.2, 14.9, 11.7, 8.4, 5.9, 4.6, 5.1, 7.1, 10.2, 13.7, 16.4, 17.8}, //Latitude 42
 {17.2, 14.7, 11.4, 8.0, 5.5, 4.2, 4.7, 6.7, 9.8, 13.4, 16.3, 17.8}, //Latitude 44
 {17.1, 14.5, 11.0, 7.5, 5.0, 3.8, 4.2, 6.2, 9.4, 13.1, 16.2, 17.8}, //Latitude 46
 {17.0, 14.3, 10.6, 7.0, 4.5, 3.3, 3.7, 5.7, 9.0, 12.8, 16.1, 17.8}, //Latitude 48
 {16.9, 14.1, 10.3, 6.6, 4.0, 2.8, 3.2, 5.3, 8.7, 12.5, 15.9, 17.8}, //Latitude 50
 {16.8, 13.8, 9.9, 6.1, 3.5, 2.4, 2.8, 4.9, 8.2, 12.2, 15.7, 17.7}, //Latitude 52
 {16.7, 13.5, 9.5, 5.7, 3.0, 2.0, 2.4, 4.5, 7.8, 11.8, 15.5, 17.7}, //Latitude 54
 {16.6, 13.2, 9.0, 5.2, 2.5, 1.6, 2.0, 4.0, 7.3, 11.4, 15.4, 17.6}, //Latitude 56
 {16.5, 12.9, 8.6, 4.7, 2.1, 1.2, 1.6, 3.5, 6.9, 11.1, 15.2, 17.6}, //Latitude 58
 {16.4, 12.7, 8.2, 4.2, 1.7, 0.8, 1.1, 3.0, 6.4, 10.8, 15.1, 17.5}, //Latitude 60
 {16.4, 12.4, 7.7, 3.7, 1.4, 0.6, 0.9, 2.5, 5.9, 10.4, 15.0, 17.6}, //Latitude 62
 {16.4, 12.1, 7.2, 3.2, 1.0, 0.4, 0.7, 2.1, 5.4, 10.0, 14.9, 17.7}, //Latitude 64
 {16.4, 11.8, 6.7, 2.8, 0.6, 0.3, 0.5, 1.6, 4.9, 9.8, 14.8, 17.8}, //Latitude 66
 {16.4, 11.5, 6.2, 2.4, 0.3, 0.1, 0.3, 1.2, 4.4, 9.4, 14.6, 17.9}, //Latitude 68
 {16.4, 11.2, 5.9, 2.0, 0.1, 0.0, 0.0, 0.8, 4.0, 9.0, 14.5, 18.1} //Latitude 70
 };

```

// Função para obter o valor de Ra baseado no mês e latitude
float getRa(int month, int latitude) {
    int latIndex = latitude / 2;    //Divide a latitude por 2 para encontrar o índice correspondente
    return pgm_read_float(&(Ra[latIndex][month - 1])); // Retorna o valor armazenado na
    tabela Ra
}

void setup() {
    Wire.begin();                //inicia o I2C
    Serial.begin(9600);          //inicia a comunicação serial
    sensors.begin();             //inicia o sensor de temperatura (DS18B20)

    // Inicialização do LCD
    lcd.init();
    lcd.backlight();
    lcd.setCursor(3, 0);
    lcd.print("BEM VINDO!");
    lcd.setCursor(1, 1);
    lcd.print("Estacao de ETo");
    delay(5000);
    lcd.clear();

    //Carrega latitude salva na EEPROM
    latitude = EEPROM.read(LATITUDE_ADDRESS);
    if (latitude == 255) {        //Valor padrão da EEPROM (não configurado)
        latitude = 0;
        latitudeConfirmed = false;
    } else {
        latitudeConfirmed = true;
        lcd.clear();
        lcd.setCursor(2, 0);
    }
}

```

```

    lcd.print("Latitude ok!");
    delay(2500);
    lcd.clear();
}

// Inicialização do Cartão SD
if (!SD.begin(chipSelect)) {                               //Verifica se o cartão SD está funcionando
    Serial.println("Erro ao iniciar cartao SD!");
    lcd.print("ERRO CARTAO SD");
    delay(5000);
    lcd.clear();
    return;
}
Serial.println("Cartao SD funcionando perfeitamente!");
lcd.print("CARTAO SD OK!!");
delay(5000);
lcd.clear();

// Inicialização do RTC
if (!rtc.begin()) {
    lcd.print("Erro no RTC");
    Serial.println("Nao foi possível iniciar data");
    while (1);                                             //trava o código caso o RTC não seja encontrado
}

// Sincroniza o RTC com o horário atual (apenas uma vez)
//rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));         //Use essa linha se precisar
ajustar a hora

displayLatitude();
}

```

```

void loop() {
  unsigned long currentMillis = millis();          //Captura o tempo atual em milissegundos

  //Lógica para configurar a latitude se ainda não foi confirmada
  if (!latitudeConfirmed) {
    if (Serial.available()) {                    //Verifica se há dados disponíveis na serial
      char command = Serial.read();
      if (command == 'U') {                      //Comando para aumentar a latitude
        latitude = min(latitude + 2, 70);        //Limita a latitude máxima a 70
        displayLatitude();
      } else if (command == 'D') {               //Comando para diminuir a latitude
        latitude = max(latitude - 2, 0);         //Limita a latitude mínima a 0
        displayLatitude();
      } else if (command == 'C') {              //Comando para confirmar a latitude
        latitudeConfirmed = true;
        EEPROM.write(LATITUDE_ADDRESS, latitude); //Salva a latitude na EEPROM
        lcd.clear();
        lcd.setCursor(2, 0);
        lcd.print("Latitude ok!");
        delay(2500);
        lcd.clear();
      }
    }
  } else {
    //Realiza leituras periódicas de temperatura
    if (currentMillis - lastReadingTime >= readingInterval) {
      lastReadingTime = currentMillis;

      sensors.requestTemperatures();
      float tempAtual = sensors.getTempCByIndex(0);
    }
  }
}

```

```

//Validação da temperatura lida
if (tempAtual != DEVICE_DISCONNECTED_C && tempAtual != -127.0 &&
tempAtual >= -50 && tempAtual <= 50) {
    tempMax = max(tempMax, tempAtual);
    tempMin = min(tempMin, tempAtual);
    tempTotal += tempAtual;
    readingsCount++;

// Calcula a temperatura média a cada hora
if (readingsCount >= totalReadingsPerHour) {
    tempMed = tempTotal / readingsCount;
    tempTotal = 0.0;
    readingsCount = 0;
}
}

//Verifica se é 23:59:59 para salvar dados
DateTime now = rtc.now();
if (now.hour() == 23 && now.minute() == 59 && now.second() == 59){
    float Ra_value = pgm_read_float(&(Ra[latitude / 2][now.month() -1]));
    float etoSensor = 0.0023 * Ra_value * (tempMed + 17.8) * sqrt(tempMax - tempMin);
    float etoCal = 0.0023 * Ra_value * (((tempMax+tempMin)/2) + 17.8) * sqrt(tempMax
- tempMin);
    //Salva dados no SD, imprime no serial e reseta variáveis
    saveTemperatureToSD(getData(now), getHora(now), tempMax, tempMin, tempMed,
etoSensor, etoCal); //salva no SD

//imprime no serial
Serial.println();
Serial.print("Data/Hora: ");
Serial.print(getData(now));

```

```
Serial.print(" / ");
Serial.print(getHora(now));
Serial.print(", Tmax: ");
Serial.print(tempMax);
Serial.print(" C, Tmin: ");
Serial.print(tempMin);
Serial.print(" C, Tmed: ");
Serial.print(tempMed);
Serial.print(" C, ETo sensor:");
Serial.print(etoSensor);
Serial.print(" mm, ETo calculado:");
Serial.print(etoCal);
Serial.print(" mm.");

//reseta temperatura
tempMax = -127.0;
tempMin = 127.0;
tempTotal = 0.0;
readingsCount = 0;
}

//apresentado no LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(getData(now));
lcd.setCursor(2, 1);
lcd.print("Temp:");
lcd.print(tempAtual);
lcd.print("C");
lcd.setCursor(10, 0);
lcd.print("LAT:");
```

```

    lcd.print(latitude);
}
ligaDesliga();
}
}

//Função para exibir a latitude no LCD
void displayLatitude() {
    lcd.clear();                //Limpa o display LCD
    lcd.setCursor(3, 0);        //Define o cursor na posição (3, 0)
    lcd.print("QUAL A SUA");    //Exibe a mensagem "QUAL A SUA"
    lcd.setCursor(0, 1);        //Define o cursor para a segunda linha
    lcd.print("LATITUDE?");     //Exibe "LATITUDE?" na segunda linha
    lcd.setCursor(10, 1);       //Define o cursor para o final da segunda linha
    lcd.print(latitude);        //Exibe o valor da latitude
}

//Função para ligar/desligar o backlight do LCD e reiniciar o sistema
void ligaDesliga() {
    if (Serial.available()) {   //Verifica se há dados disponíveis no monitor serial
        char command = Serial.read(); //Lê o comando do serial
        if (command == 'Q') {     //comando para ligar o backlight
            lcd.backlight();
            Serial.println("Backlight ON");
        } else if (command == 'T') { //comando para desligar o backlight
            lcd.noBacklight();
            Serial.println("Backlight OFF");
        } else if (command == 'R') { //comando para reiniciar o sistema
            Serial.println("Reiniciando o equipamento...");
            delay(500);
            resetSystem();        //Chama a função para reiniciar o sistema
        }
    }
}

```

```

    }
}
}

//Função para reiniciar o sistema
void resetSystem() {
    EEPROM.write(LATITUDE_ADDRESS, 255); //Reseta o valor de latitude na EEPROM
    wdt_enable(WDTO_15MS); //Habilita o Watchdog Timer para reiniciar
    rapidamente o sistema
    while (true) { //Entra em um loop infinito para reiniciar o sistema
    }

// Função para formatar a data como String
String getData(DateTime now) {
    String data = "";
    if (now.day() < 10) data += "0"; //Adiciona zero à esquerda, se necessário
    data += String(now.day()) + "/";
    if (now.month() < 10) data += "0"; //Adiciona zero à esquerda, se necessário
    data += String(now.month()) + "/";
    data += String(now.year()); //adiciona o ano
    return data;
}

// Função para formatar a hora como String
String getHora(DateTime now) {
    String hora = "";
    if (now.hour() < 10) hora += "0"; //Adiciona zero à esquerda, se necessário
    hora += String(now.hour()) + ":";
    if (now.minute() < 10) hora += "0"; //Adiciona zero à esquerda, se necessário
    hora += String(now.minute()) + ":";
    if (now.second() < 10) hora += "0"; //Adiciona zero à esquerda, se necessário

```

```

hora += String(now.second());           //adiciona segundos
return hora;
}

// Função para salvar os dados de temperatura no cartão SD
void saveTemperatureToSD(String data, String hora, float tempMax, float tempMin, float
tempMed, float etoSensor, float etoCal) {
  if (!SD.begin(chipSelect)) {         //Verifica se o SD card foi iniciado corretamente
    Serial.print("Erro ao iniciar o SD card. ");
    return;
  }
  File arquivo = SD.open("ETo.txt", FILE_WRITE); //Abre o arquivo "ETo.txt" no modo
de escrita
  if (arquivo){                        //Se o arquivo foi aberto com sucesso
    arquivo.print("Data/Hora: ");
    arquivo.print(data);               //Salva a data
    arquivo.print(" / ");
    arquivo.print(hora);               //Salva a hora
    arquivo.print(",");
    arquivo.print(tempMax);            //Salva a temperatura máxima
    arquivo.print(",");
    arquivo.print(tempMin);            //Salva a temperatura mínima
    arquivo.print(",");
    arquivo.print(tempMed);            //Salva a temperatura média
    arquivo.print(",");
    arquivo.print(etoSensor);          //Salva o valor de ETo do sensor
    arquivo.print("mm,");
    arquivo.print(etoCal);             //Salva o valor de ETo calculado
    arquivo.print("mm");
    arquivo.println();                 //Salva os dados em uma nova linha
    arquivo.close();                   //Fecha o arquivo
  }
}

```

```

Serial.print(" Dados gravados corretamente.");
} else {
//Se houve erro ao abrir o arquivo
Serial.println(" Erro ao abrir o arquivo no SD.");
}
}
}

```

APÊNDICE 2 – Dados obtidos pelo experimento

Dados coletados do arduino				
data	Tmax	Tmin	Tmed (Calculado)	Eto (calculado)
05/12/2024	32,25	22,39	27,32	5,34
06/12/2024	32,13	23,87	28,00	4,96
07/12/2024	34,63	21,37	28,00	6,29
08/12/2024	38,63	22,75	30,69	7,29
09/12/2024	37,75	26,5	32,13	6,32
10/12/2024	31,85	26,63	29,24	4,05
11/12/2024	30,75	24,37	27,56	4,32
12/12/2024	30,5	27,51	29,01	3,05
13/12/2024	30,33	27,01	28,67	3,19
14/12/2024	32,09	28,87	30,48	3,27
15/12/2024	30,97	27,94	29,46	3,10

ANEXO 1 – Tabela de radiação solar Ra (Camargo; Camargo, 1983)

Latitude Sul - Graus	Jan.	Fev.	Mar.	Abr.	Mai.	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
0	14,5	15,0	15,2	14,7	13,9	13,4	13,5	14,2	14,9	14,9	14,6	14,3
2	14,8	15,2	15,2	14,5	13,6	13,0	13,2	14,0	14,8	15,0	14,8	14,6
4	15,0	15,3	15,1	14,3	13,3	12,7	12,8	13,7	14,7	15,1	15,0	14,9
6	15,3	15,4	15,1	14,1	13,0	12,6	12,5	13,5	14,6	15,1	15,2	15,1
8	15,6	15,6	15,0	14,0	12,7	12,0	12,2	13,2	14,5	15,2	15,4	15,4
10	15,9	15,7	15,0	13,8	12,4	11,6	11,9	13,0	14,4	15,3	15,7	15,7
12	16,1	15,8	14,9	13,5	12,0	11,2	11,5	12,7	14,2	15,3	15,8	16,0
14	16,3	15,8	14,9	13,2	11,6	10,8	11,1	12,4	14,0	15,3	15,9	16,2
16	16,5	15,9	14,8	13,0	11,3	10,4	10,8	12,1	13,8	15,3	16,1	16,4
18	16,7	15,9	14,7	12,7	10,9	10,0	10,4	11,8	13,7	15,3	16,2	16,6
20	16,7	16,0	14,5	12,4	10,6	9,6	10,0	11,5	13,5	15,3	16,4	16,8
22	16,9	16,0	14,3	12,0	10,2	9,1	9,6	11,1	13,1	15,2	16,5	17,0
24	16,9	15,9	14,1	11,7	9,8	8,6	9,1	10,7	13,1	15,1	16,5	17,1
26	17,0	15,9	13,9	11,4	9,4	8,1	8,7	10,4	12,8	15,0	16,6	17,3
28	17,1	15,8	13,7	11,1	9,0	7,8	8,3	10,0	12,6	14,9	16,6	17,5
30	17,2	15,7	13,5	10,8	8,5	7,4	7,8	9,6	12,2	14,7	16,7	17,6
32	17,2	15,6	13,3	10,5	8,1	7,0	7,4	9,2	11,9	14,6	16,7	17,6
34	17,2	15,5	13,0	10,1	7,6	6,6	7,0	8,8	11,6	14,4	16,7	17,6
36	17,3	15,4	12,7	9,7	7,2	6,1	6,5	8,3	11,3	14,3	16,6	17,7
38	17,3	15,2	12,4	9,3	6,7	5,6	6,0	7,9	11,0	14,1	16,6	17,7
40	17,3	15,1	12,1	8,9	6,3	5,1	5,5	7,5	10,6	13,9	16,5	17,8
42	17,2	14,9	11,7	8,4	5,9	4,6	5,1	7,1	10,2	13,7	16,4	17,8
44	17,2	14,7	11,4	8,0	5,5	4,2	4,7	6,7	9,8	13,4	16,3	17,8
46	17,1	14,5	11,0	7,5	5,0	3,8	4,2	6,2	9,4	13,1	16,2	17,8
48	17,0	14,3	10,6	7,0	4,5	3,3	3,7	5,7	9,0	12,8	16,1	17,8
50	16,9	14,1	10,3	6,6	4,0	2,8	3,2	5,3	8,7	12,5	15,9	17,8
52	16,8	13,8	9,9	6,1	3,5	2,4	2,8	4,9	8,2	12,2	15,7	17,7
54	16,7	13,5	9,5	5,7	3,0	2,0	2,4	4,5	7,8	11,8	15,5	17,7
56	16,7	13,2	9,0	5,2	2,5	1,6	2,0	4,0	7,3	11,4	15,4	17,6
58	16,5	12,9	8,6	4,7	2,1	1,2	1,6	3,5	6,9	11,1	15,2	17,6
60	16,4	12,7	8,2	4,2	1,7	0,8	1,1	3,0	6,4	10,8	15,1	17,5
62	16,4	12,4	7,7	3,7	1,4	0,6	0,9	2,5	5,9	10,4	15,0	17,6
64	16,4	12,1	7,2	3,2	1,0	0,4	0,7	2,1	5,4	10,0	14,9	17,7
66	16,4	11,8	6,7	2,8	0,6	0,3	0,5	1,6	4,9	9,8	14,8	17,8
68	16,4	11,5	6,2	2,4	0,3	0,1	0,3	1,2	4,4	9,4	14,6	17,9
70	16,4	11,2	5,9	2,0	0,1	0,0	0,0	0,8	4,0	9,0	14,5	18,1

ANEXO 2 – Critério de interpretação do desempenho dos métodos de estimativa da

ETo, pelo índice “C”

Valor de “c”	Desempenho
>0,85	Ótimo
0,76 a 0,85	Muito bom
0,66 a 0,75	Bom
0,61 a 0,65	Mediano
0,51 a 0,60	Sufrível
0,41 a 0,50	Mau
≤0,40	Péssimo