

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JOÃO VICTOR GALVÃO DO PRADO SOUZA**

**SISTEMA DE AQUISIÇÃO DE DADOS DE BAIXO CUSTO COM ARDUINO:  
TERMÔMETRO DE CINCO PONTOS**

**LONDRINA**

**2022**

**JOÃO VICTOR GALVÃO DO PRADO SOUZA**

**SISTEMA DE AQUISIÇÃO DE DADOS DE BAIXO CUSTO COM ARDUINO:  
TERMÔMETRO DE CINCO PONTOS**

**LOW-COST DATA ACQUISITION SYSTEM WITH ARDUINO: FIVE-POINT  
THERMOMETER**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Engenharia Mecânica, do Departamento Acadêmico de Engenharia Mecânica, da Universidade Tecnológica Federal do Paraná.  
Orientador: Prof. Dr. Roger Nabeyama Michels.

**LONDRINA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JOÃO VICTOR GALVÃO DO PRADO SOUZA**

**SISTEMA DE AQUISIÇÃO DE DADOS DE BAIXO CUSTO COM ARDUINO:  
TERMÔMETRO DE CINCO PONTOS**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Mecânica da Universidade  
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 17/junho/2022

---

Roger Nabeyama Michels  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Cláudia Santos Fiuza Lima  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Ismael de Marchi Neto  
Doutorado  
Universidade Tecnológica Federal do Paraná

**LONDRINA**

**2022**

Esse trabalho é dedicado a todas as pessoas das quais tive que me distanciar durante meu período de graduação. Mesmo distante, meu coração sempre esteve com vocês.

## **AGRADECIMENTOS**

A Deus, por me conceder saúde para que eu supere meus desafios.

Aos meus pais, Aerosvaldo e Mariana, pelo imenso incentivo para que eu busque meus sonhos, mesmo que isso custe os sonhos deles.

Aos meus irmãos, Pedro e Ana Clara, por iluminarem meus dias.

À minha família, em especial a minha Avó Neide, pelo carinho, atenção e cuidado.

À Julia, por ser minha parceira em todos os momentos e agir como meu porto seguro.

Aos meus amigos, pelo grande companheirismo, em especial ao Lucas, por me acompanhar e compartilhar forças nessa jornada universitária.

Ao professor Roger Nabeyama Michels, o qual além de me orientar com maestria nesse trabalho, sempre foi fonte de motivação e inspiração durante minha graduação.

Ao Guilherme Quereza e João Paulo Bachega, pelo ótimo suporte nos conhecimentos técnicos.

## RESUMO

No século XXI o uso da termometria se mostra essencial para diversos setores e atividades, como no controle e análise de dados, porém trata-se de uma tecnologia muitas vezes inviável devido ao valor expressivo dos termômetros do mercado, como por exemplo o termômetro Penta III, utilizado na refrigeração, além de requisitado em pesquisas universitárias. Desta forma, o objetivo deste projeto é desenvolver um termômetro de cinco pontos utilizando a plataforma Arduino. Para a montagem do protótipo foi utilizada uma placa Arduino Mega 2560, *shields* RTC e SD, um *display* e sensores de temperatura DS18B20. Devido a flexibilidade deste sistema, foi possível, além de mostrar a temperatura dos sensores em tempo real via *display*, armazenar os dados em cartão Micro SD com frequência variável de acordo com a necessidade. Com a conclusão deste trabalho, foi obtido um produto alternativo ao encontrado no mercado, com maior potencial tecnológico, custo inferior e com a filosofia *Open Source*.

Palavras-chave: termometria; código aberto; automatização.

## **ABSTRACT**

In the 21st century, the use of thermometry is essential for various sectors and activities, such as data control and analysis, but it is often an unfeasible technology due to the significant value of market thermometers, such as the Penta III thermometer, used in refrigeration, in addition to being requested in university research. Thus, the objective of this project is to develop a five-point thermometer using the Arduino platform. To assemble the prototype, an Arduino Mega 2560 board, RTC and SD shields, a display and DS18B20 temperature sensors will be used. Due to the flexibility of this system, it will be possible, in addition to showing the temperature of the sensors in real time via display, to store the data on a Micro SD card with variable frequency according to need. With the conclusion of this work, it was able to obtain a market alternative product with great technological potential, low cost and with an open source philosophy.

Keywords: thermometry; open source; automation.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1 - Esquema de um Termoscópio .....</b>	<b>18</b>
<b>Figura 2 - Termômetro Clínico Eletrônico .....</b>	<b>19</b>
<b>Figura 3 - Termômetro Penta III FullGauge .....</b>	<b>20</b>
<b>Figura 4 - Registrador de Dados Portátil de Seis Canais.....</b>	<b>21</b>
<b>Figura 5 - Placa Arduino Mega 2560 R3.....</b>	<b>22</b>
<b>Figura 6 - Sensor DS18B20 Encapsulado .....</b>	<b>24</b>
<b>Figura 7 - Fluxograma da metodologia.....</b>	<b>25</b>
<b>Figura 8 - Esquema do plano inicial da montagem do sistema. ....</b>	<b>26</b>
<b>Figura 9 - Protótipo Finalizado.....</b>	<b>30</b>
<b>Figura 10 - Sistema visto de outro ângulo .....</b>	<b>31</b>
<b>Figura 11 - Curva de temperatura com dados brutos .....</b>	<b>32</b>
<b>Figura 12 - Dashboard interativo com gráfico de medições do ambiente.....</b>	<b>33</b>
<b>Figura 13 – Esquema do experimento com copos .....</b>	<b>34</b>
<b>Figura 14 - Dashboard interativo com gráfico de medições dos copos.....</b>	<b>35</b>



## LISTA DE TABELAS

Tabela 1 - Preço Médio dos Componentes .....	29
Tabela 2 - Trecho das medições de repetibilidade.....	36

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
<b>2</b>	<b>OBJETIVOS</b> .....	<b>15</b>
<b>2.1</b>	<b>Objetivo geral</b> .....	<b>15</b>
<b>2.2</b>	<b>Objetivos específicos</b> .....	<b>15</b>
<b>3</b>	<b>JUSTIFICATIVA</b> .....	<b>16</b>
<b>4</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>17</b>
<b>4.1</b>	<b>Evolução da termometria</b> .....	<b>17</b>
4.1.1	Conceituação e necessidade da termometria .....	17
4.1.2	Desenvolvimento das escalas e instrumentos de medição .....	17
4.1.3	Padronização .....	19
4.1.4	Advento dos termômetros eletrônicos .....	19
<b>4.2</b>	<b>Termômetro penta</b> .....	<b>20</b>
<b>4.3</b>	<b>Dispositivos de medição ômega</b> .....	<b>20</b>
<b>4.4</b>	<b>Plataforma arduino</b> .....	<b>21</b>
4.4.1	Shields e módulos .....	22
<b>4.5</b>	<b>Sensores</b> .....	<b>23</b>
4.5.1	Tecnologia onewire e o sensor ds18b20 .....	23
<b>5</b>	<b>METODOLOGIA</b> .....	<b>25</b>
<b>5.1</b>	<b>Panorama geral</b> .....	<b>25</b>
<b>5.2</b>	<b>Confecção do sistema de aquisição de dados</b> .....	<b>26</b>
5.2.1	Endereçamento dos sensores .....	26
5.2.2	Desenvolvimento do código .....	27
5.2.3	Bibliotecas utilizadas .....	27
5.2.4	Montagem final .....	28
5.2.5	Funcionamento e obtenção de dados .....	28
<b>5.3</b>	<b>Análise de custos</b> .....	<b>28</b>
<b>6</b>	<b>RESULTADOS E DISCUSSÃO</b> .....	<b>30</b>
<b>6.1</b>	<b>Protótipo finalizado</b> .....	<b>30</b>
6.1.1	Primeiro teste e seus desafios .....	31
6.1.2	Tratamento e visualização de dados .....	32
6.1.3	Segundo teste .....	33
6.1.4	Terceiro teste.....	35
6.1.5	Visualização via LCD.....	36

<b>6.2</b>	<b>Oportunidade de melhoria.....</b>	<b>37</b>
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>38</b>
	<b>REFERÊNCIAS.....</b>	<b>39</b>
	<b>ANEXO A - Código do sistema .....</b>	<b>42</b>

## 1 INTRODUÇÃO

A medição sempre foi um processo intrínseco à vida humana, consciente ou inconscientemente sempre foi preciso medir para assim compreender.

A medição de temperatura, também conhecida como termometria, não é diferente, conceitos mais abstratos de compreensão da temperatura, como calor e frio, existem desde séculos a.C., fato esse que demonstra a curiosidade humana quanto a esse fenômeno. Entretanto, ao contrário da metrologia, a termometria não é intuitiva, dado que, por exemplo ao unir duas peças de 10 cm, tem-se uma peça de 20 cm, porém, ao unir duas peças com 10 °C, a peça resultante também terá 10 °C. Além desse fato, o ser humano é ineficaz para medir temperatura por si próprio, pois apenas consegue diferenciar se o objeto está mais quente ou mais frio que si mesmo.

Apenas quando começou a existir um consenso quanto às escalas termométricas foi possível obter grandes avanços tanto na termometria, quanto na termodinâmica. Em decorrência dessa disparidade de desenvolvimento, foi incentivada e realizada a padronização das escalas termométricas por meio do SI (Sistema Internacional de Unidades).

Na segunda metade do século XIX, ao acompanhar a digitalização crescente em todas as áreas, surgiram os termômetros eletrônicos, uma solução ecológica e posteriormente mais econômica em relação aos termômetros convencionais. Apesar dessas características chamarem atenção, as qualidades dos termômetros eletrônicos vão muito além disso, porque englobam principalmente a flexibilidade dos sistemas eletrônicos e sua grande capacidade de aperfeiçoamento. Decorrente disso, diversos tipos de termômetros surgiram, cada um com suas peculiaridades e com aplicações diferentes, o que agregou muito à termometria.

Mesmo com o advento e ascensão da era digital, composta por eletrônicos e sensores, essa tecnologia ainda não é acessível financeiramente para muitos que dela necessitam para fins de aquisição, controle e análise de dados, como pequenos empreendedores, professores e alunos (AZÚA-BARRÓN, 2017).

Um exemplo dessa realidade é com um dos termômetros mais utilizados no setor de refrigeração, o termômetro Penta III, cujo preço é, em média, 580 reais. Isso cria uma lacuna de mercado reservada para alternativas de sistemas de aquisição de dados a baixo custo.

Com posição dominante nesse mercado encontra-se o Arduino, graças ao seu custo acessível, expressiva comunidade em fóruns, facilidade de utilização, entre outros fatores (DE ALMEIDA, 2021).

O grande trunfo do Arduino é sua adaptabilidade, dado que essa plataforma pode ser usada para uma imensa gama de aplicações, desde experimentação amadora até dispositivos para chão de fábrica. Ao utilizar diversos periféricos e possibilitar a reprogramação inúmeras vezes, o Arduino se torna uma plataforma extremamente competente.

Graças ao grande número de programadores que colaboram com a comunidade Arduino, a maior parcela dos códigos a serem programados são simplificados, isso se dá pelo uso das bibliotecas que esses programadores criam, as quais acrescentam ao desenvolvimento diversas nuances sem necessidade de aprofundamento técnico.

Ao juntar todos os benefícios técnicos e financeiros do Arduino, fica explicitado o porquê dessa plataforma ser a líder nesse mercado. O que fica ainda mais claro ao notar que, atualmente, a grande maioria dos trabalhos acadêmicos que envolvem eletrônica utilizam ou fazem referência ao Arduino. Com esse destaque, surgiu a tendência de criar alternativas com melhor custo-benefício para diversos dispositivos eletrônicos.

Neste trabalho é apresentada uma alternativa ao termômetro de cinco pontos, para tal, foi utilizada a plataforma Arduino em conjunto com os sensores DS18B20, que já se provaram ser uma combinação válida para um sistema de aquisição de dados a um preço acessível (KOESTER, 2019).

## **2 OBJETIVOS**

### **2.1 Objetivo geral**

Desenvolver um termômetro automático de 5 pontos de baixo custo com a utilização da plataforma *Open Source* Arduino.

### **2.2 Objetivos específicos**

- Levantar bibliografia relacionada;
- Desenvolver programação em Arduino;
- Pesquisar e comprar componentes;
- Elaborar a montagem eletrônica;
- Montar protótipo;
- Analisar os dados coletados;
- Testar e ajustar o protótipo;

### 3 JUSTIFICATIVA

O sistema de aquisição de dados possui grande flexibilidade em comparação com o Termômetro Penta comercial, dado que o caráter reprogramável do Arduino permite alterações em diversos âmbitos, como no intervalo de medição, que pode ser adaptado para qualquer frequência desejada e com isso é possível obter uma base de dados rica e adaptável ao estudo pretendido. Outra vantagem significativa é o custo, muito inferior ao do termômetro Penta da FullGauge e ainda mais econômico se comparado aos dispositivos de aquisição de dados termométricos da Omega. Embasado em pesquisas de preço dos componentes em sites de e-commerce brasileiros, é estimado uma redução de, no mínimo, 40% em relação ao modelo comercial da FullGauge.

Em conjunto com a flexibilidade, vem a imensa gama de possíveis melhorias, as quais podem levar o SAD (Sistema de Aquisição de Dados) para um patamar completamente diferente. Essas melhorias são possíveis por meio dos diversos *shields* disponíveis para acoplamento na placa Arduino, como tela LCD, placa de Bluetooth, sensores, entre inúmeros outros.

Dessa forma, ao implementar um *shield* e realizar as adaptações necessárias no código, é possível acrescentar ao sistema uma função totalmente nova.

A automatização que o sistema confeccionado fornece é um de seus principais pontos fortes, dado que o Termômetro Penta III não possui essa característica e os dispositivos Omega, que possuem, contam também com um custo altíssimo ao comparar.

Essas melhorias se tornam ainda mais viáveis com o auxílio da enorme comunidade que ensina, testa e melhora as mais variadas tecnologias que envolvem Arduino.

## **4 REFERENCIAL TEÓRICO**

Nessa seção do trabalho é conceituada e embasada toda a teoria que sustenta o mesmo, a seguir a linha do tempo até o presente momento.

### **4.1 Evolução da termometria**

#### **4.1.1 Conceituação e necessidade da termometria**

A termometria é o estudo dos processos de medição da temperatura dos corpos, ciência essa que compreende a temperatura como medida do grau de agitação das moléculas (PIRES, 2006).

Segundo Pires (2006), o corpo humano consegue apenas distinguir o diferencial de temperatura em relação a sua própria, ou seja, aferir por meio do tato se algo está mais quente ou frio, porém sequer essa análise pode ser feita com exatidão, pois existe uma alta influência da condutividade térmica de uma superfície para com a sensação humana.

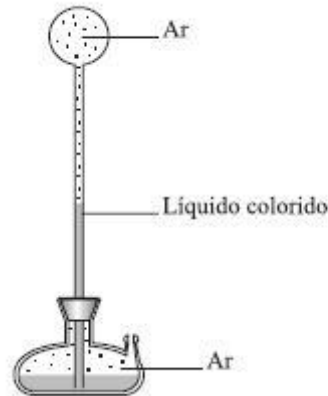
A partir dessa imprecisão é necessário que seja estabelecido um padrão de medição por meio de um instrumento, o qual é apelidado de termômetro e utiliza da variação das propriedades das substâncias conforme a temperatura muda para assim fazer a aferição (PIRES, 2006).

#### **4.1.2 Desenvolvimento das escalas e instrumentos de medição**

De acordo com Hay (2020), as primeiras escalas graduadas de temperatura surgiram com os termoscópios (Figura 1), com sua invenção atribuída a Galileo ou Santorio de Padua dependendo da fonte, esses equipamentos utilizavam ar e se mostraram os precursores dos termômetros de expansão de líquido, que surgiram em 1654 com Ferdinand II, desde então ocorreram inúmeras melhorias com o passar dos anos, em especial a troca do álcool pelo mercúrio como líquido termométrico por Daniel Gabriel Fahrenheit em 1714.



**Figura 1 - Esquema de um Termoscópio**



**Fonte: Timetoast**

Conforme os avanços ocorriam na tecnologia dos termômetros, mais escalas eram desenvolvidas para quantificar e comparar as medidas, no mesmo século da invenção desse instrumento houve o surgimento de mais de 30 escalas diferentes de temperatura (HAY, 2020).

Dentre essa vasta gama de escalas, três foram predominantemente utilizadas para trabalhos científicos no século XVIII, as escalas termométricas de Fahrenheit, Réaumur e Celsius (PIRES, 2006).

Daniel Gabriel Fahrenheit estipulou sua escala termométrica ao fixar 0°F no ponto de fusão de uma mistura de água com cloreto de amônia e o outro extremo foi fixado na temperatura do corpo humano, esse intervalo foi subdividido em 96 partes iguais (HAY, 2020).

René-Antoine Ferchault de Réaumur propôs uma escala que parte de 0 na temperatura em que o gelo derrete à 80 no ponto da água fervente (HAY, 2020).

Anders Celsius desenvolveu em 1742 um termômetro de mercúrio com escala de 0 a 100, que traz como pontos fixos os pontos de fusão e ebulição da água (HAY, 2020).

William Thomson, posteriormente Lord Kelvin, estabeleceu a escala absoluta em 1848, a qual tem como referência o zero absoluto, temperatura onde a agitação térmica cessa e as moléculas se encontram em repouso. Para estipular a graduação, Kelvin confirmou que a cada grau Celsius diminuído a partir do ponto triplo da água, a pressão do gás abaixava em  $1/273$ , o que o fez concluir que numa temperatura de aproximadamente  $-273^{\circ}\text{C}$  as moléculas estariam em repouso (HAY, 2020).

#### 4.1.3 Padronização

O Sistema Internacional de Unidades define como unidade de temperatura termodinâmica o Kelvin (K), entretanto, o SI reconhece que para situações cotidianas a temperatura em Kelvins não é ideal, portanto, fica estipulada a expressão (INMETRO, 2021)

$$273,15 + \frac{t}{^{\circ}\text{C}} = \frac{T}{\text{K}} \quad (1)$$

Onde o termo  $t$  é a temperatura em graus Celsius e o termo  $T$  representa a temperatura em Kelvin.

A partir dessa conversão, também é permitido o uso do grau Celsius.

#### 4.1.4 Advento dos termômetros eletrônicos

Para Vilar (2015), desde os anos 70 os termômetros eletrônicos têm uma presença cada vez mais dominante no mercado, isso se dá devido ao barateamento da tecnologia utilizada e na gama de vantagens existentes em sua implementação, como praticidade, remoção do erro humano, baixo peso, rápida resposta, boa resistência mecânica e ampla faixa de temperatura de trabalho.

Um termômetro eletrônico, apresentado na Figura 2, tem como base sensores elétricos, como termopares, termômetros de resistência e termístores, além de sensores de radiação infravermelho (PIRES, 2006).

**Figura 2 - Termômetro Clínico Eletrônico**



**Fonte: Hospinet**

## 4.2 Termômetro penta

O termômetro Penta é um tipo específico de termômetro portátil que tem como função monitorar e apresentar a temperatura em 5 pontos distintos, por meio do uso de termopares para medição, sua faixa de medição é de -50 a 105 °C e a precisão é +-0,3 f.e., além disso, o comprimento dos cabos dos sensores é de 1,5 metros e o equipamento conta com um *display* para mostrar as aferições.

O funcionamento do termômetro Penta III, explícito na Figura 3, gira em torno da exibição de uma temperatura no *display*, essa temperatura pode, por meio do botão SEL, ser a temperatura de um dos sensores (a escolha), a temperatura média entre uma quantidade X de sensores (1 e 2; 1, 2 e 3; 1, 2, 3, 4; 1, 2, 3, 4 e 5) e a diferença de temperatura entre dois sensores, existem também as funções *HOLD*, Máximo e Mínimo, a primeira trata de travar a leitura no sensor, já a segunda mostra a temperatura máxima dentre os cinco sensores, enquanto a última mostra a temperatura mínima nas mesmas condições. De acordo com FullGauge (2021), o Termômetro Penta III “[...] é excelente para medições de temperatura em aparelhos de ar-condicionado central, automotivo ou de janela; freezers; no balanceamento frigorífico de evaporadores, entre outros”.

Figura 3 - Termômetro Penta III FullGauge



Fonte: FullGauge

## 4.3 Dispositivos de medição ômega

A Omega é uma empresa de 1962 que trabalha com instrumentação industrial para controle de temperatura, pressão, vazão, pH, dentre outros fatores, o braço brasileiro dessa empresa conta com milhares de itens que vão desde produtos para medição e controle, aquisição de dados, entre outras opções.

Ao focar no escopo de medição de temperatura, é notável que a Omega com vários dispositivos portáteis de aquisição de dados de temperatura que envolvem tecnologia de ponta, com possibilidade de uso de sondas RTD e até mesmo termopares de diversos tipos.

Um exemplo disposto no site da Omega seria o *Registrador de Dados Portátil de Seis Canais com Tela Sensível ao Toque*, vendido pelo valor de R\$5660 e ilustrado na figura 4, o qual conta com funções como entradas para termopares e RTD, análise simultânea entre canais, possibilidade de registro manual e programado, 4 GB de armazenamento de dados, exibição de valores de máximo, mínimo, média e desvio padrão, pode ser alimentado a bateria ou com fonte, tela sensível ao toque e gráfico de temperatura embutido.

**Figura 4 - Registrador de Dados Portátil de Seis Canais**



Fonte: Omega

#### 4.4 Plataforma arduino

Com sua origem em 2005, a plataforma Arduino, criada por Massimo Banzi, surgiu a partir da necessidade de um sistema barato e fácil aplicado ao ensino de eletrônica (SEVERANCE, 2014).

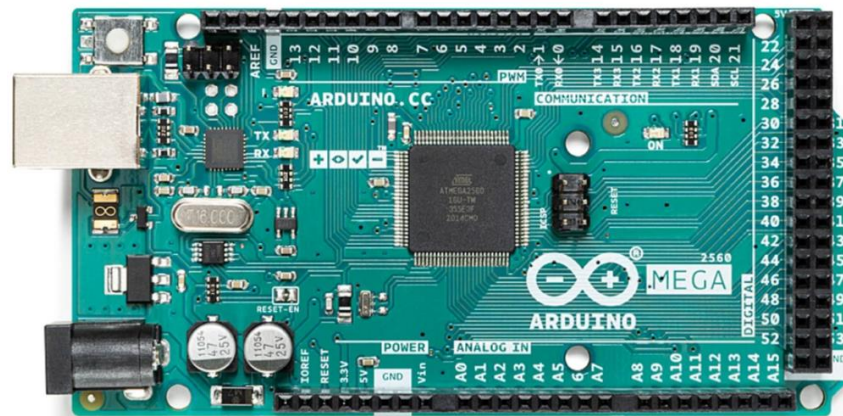
De acordo com Severance (2014), Banzi decidiu lançar o *hardware* como código aberto para que outros pudessem construí-lo. Conforme os usuários confeccionaram, documentaram e fizeram *upload* de projetos, a popularidade da

plataforma cresceu exponencialmente. Em Conjunto com a popularidade crescente, houve o aumento da demanda, o que fomentou a expansão da oferta.

Conforme Badamasi (2014), Arduino é uma placa de circuito com chip, como mostrado na Figura 5, no qual é possível programar, apagar e reprogramar com os mais variados objetivos, isso ao utilizar dispositivos de entrada como sensores e dispositivos de saída como telas.

“Arduino é uma plataforma eletrônica de código aberto baseada em *hardware* e *software* fáceis de usar” (ARDUINO, 2018).

**Figura 5 - Placa Arduino Mega 2560 R3**



Fonte: Arduino

Quanto ao *software*, é utilizado *The Arduino IDE*, que utiliza de uma versão simplificada da linguagem C++, o que possibilita um ambiente de programação amigável, além disso, os códigos produzidos podem conter funcionalidades diferentes ou podem ser simplificados, ambos através de inúmeras bibliotecas que estão disponíveis para a comunidade (BADAMASI, 2014 e ARDUINO, 2018).

Com isso, é evidenciado que o Arduino se estabeleceu no mercado por ser uma alternativa de baixo custo e dispor de um procedimento automatizado, sem interferência humana e graças a isso, com menor taxa de erro (REGIS-SILVA ET AL, 2017).

#### 4.4.1 Shields e módulos

Uns dos responsáveis pela flexibilidade do Arduino são os *shields* e módulos que agem como extensão do hardware principal. Os *shields* são placas de circuito que

contêm outros dispositivos, tais placas podem ser acopladas à placa principal Arduino diretamente e assim, atribui-se novas funcionalidades. Já os módulos são um pouco diferentes, pois apesar de ter a mesma finalidade dos *shields*, sua conexão é feita por fios (SANTOS apud MCROBERTS, 2018).

#### 4.5 Sensores

Segundo Santos *apud* Banzi (2018), “Sensores são componentes que convertem medidas do mundo real em sinais elétricos.”

A definição de sensor aponta para “aquilo que sente”, dessa forma, ele pode detectar sinais de diversos fenômenos físicos, como luminosidade, temperatura, radiação, entre outros. Um sensor pode ser analógico ou digital, o segundo é muito utilizado na eletrônica, o seu funcionamento é baseado em níveis de tensão pré-estabelecidos LOW e HIGH, ou 0 e 1. Atualmente existe uma vasta gama de sensores com diferentes precisões, escalas e custos, dentre esses, os mais comuns são os termopares, termístores e RTDs (*Resistance Temperature Detector*), essa última opção funciona a partir do princípio físico da relação de proporcionalidade entre o aumento da temperatura e o aumento da resistência elétrica, uma vez que esse sistema é monitorado, é possível quantificar a variação e definir a temperatura (MARTINAZZO e ORLANDO, 2016).

##### 4.5.1 Tecnologia onewire e o sensor ds18b20

A recente geração de sensores digitais com única via de comunicação, denominado *OneWire Bus*, se caracteriza por utilizar apenas um fio na comunicação com o microcontrolador e possui baixíssimo consumo de energia, o que permite que os sensores utilizem a própria tensão da placa. A tecnologia *OneWire* demanda que cada sensor tenha seu número de identificação, para que seja possível diferenciá-los utilizando um programa, dado que a comunicação é feita apenas por uma via (AMORIM, 2015).

Ainda segundo Amorim (2015), o Sensor DS18B20, explicitado na Figura 6, é um sensor digital da Maxim e é encapsulado, ou seja, à prova d’água, vale salientar que esse sensor possui faixa de operação entre -55 a 125 °C, acurácia de 0,5 °C entre

-10 °C e 85 °C e resolução programável de 9 até 12 bits. Dentre as vantagens desse sensor pode-se apontar versatilidade, custo acessível e calibração de fábrica.

**Figura 6 - Sensor DS18B20 Encapsulado**

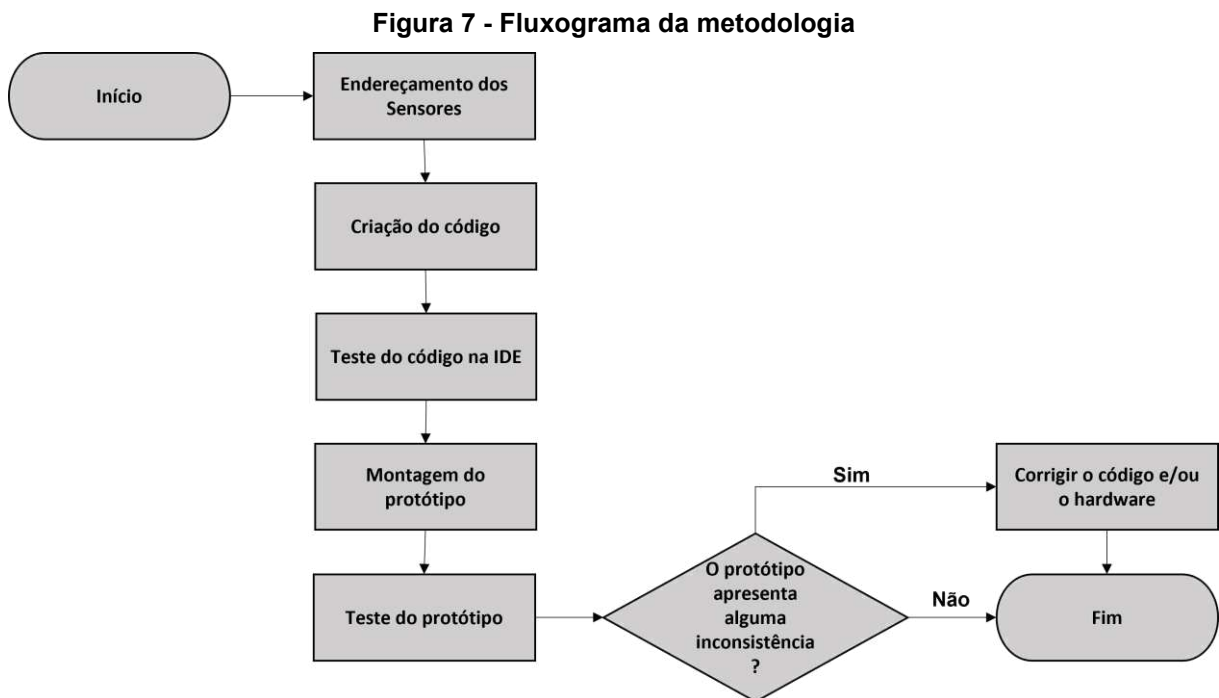


**Fonte: Eletrogate**

## 5 METODOLOGIA

### 5.1 Panorama geral

O sistema de aquisição de dados a ser desenvolvido é um termômetro de 5 pontos e é composto pela placa Arduino Mega, o módulo RTC/SD, o módulo RTC DS1307, o cartão micro SD, cinco sensores DS18B20 e o *shield* LCD. A Figura 7 aponta o fluxograma que norteará a metodologia desse projeto.

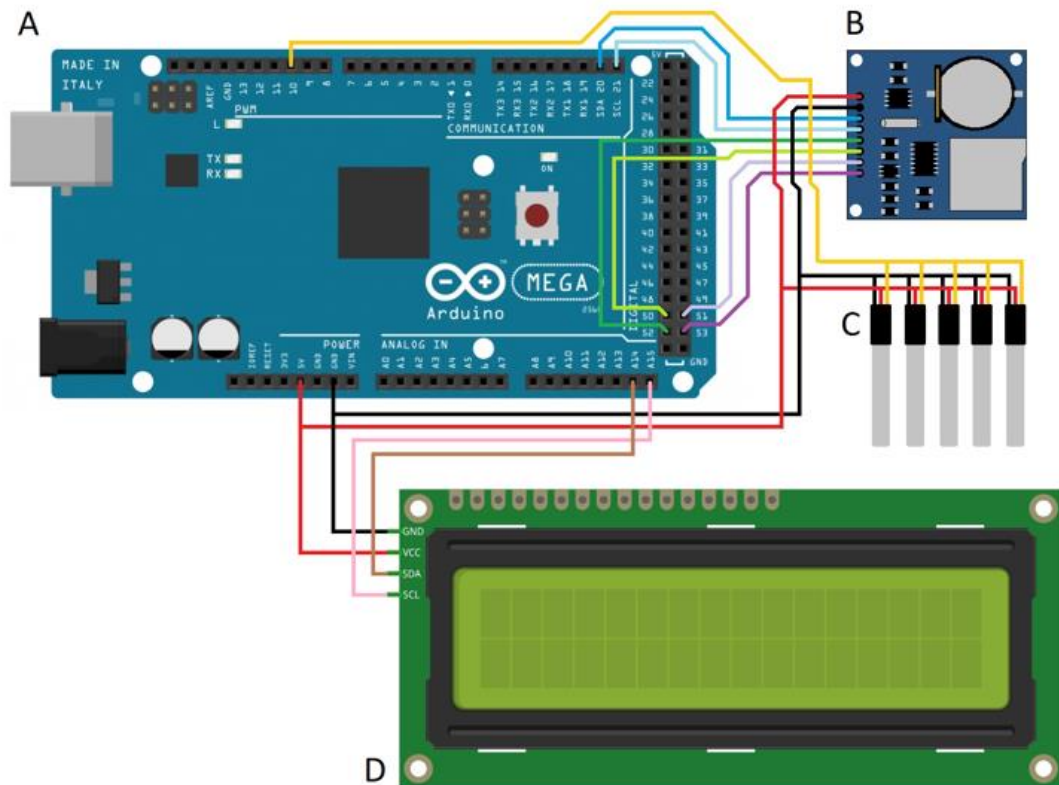


Fonte: Autoria própria

Já a Figura 8 aponta qual foi o formato final do projeto do sistema, com todos os componentes necessários já acoplados corretamente. Entretanto na execução do projeto o *shield* datalogger não foi compatível com a placa Arduino Mega 2560, o que obrigou a compra de dois módulos separados que fazem essa mesma função, são eles os módulos micro SD e RTC DS1307.



**Figura 8 - Esquema do plano inicial da montagem do sistema.**  
**A) Placa Arduino MEGA; B) Datalogger SD/RTC; C) Sensores DS18B20; D) Tela LCD.**



Fonte: Autoria própria

## 5.2 Confeção do sistema de aquisição de dados

### 5.2.1 Endereçamento dos sensores

Para a confecção desse sistema, o primeiro passo necessário foi o endereçamento dos sensores, procedimento que serve para identificar os sensores com a finalidade de fazê-los serem referenciados e acionados pelo código.

Para a obtenção do endereço do sensor, foi necessário conectá-lo em uma *protoboard* por meio de três fios, o negativo (cor preta), o positivo (cor vermelha) e o *data* (cor amarela). O negativo é conectado ao pino GND do Arduino, o positivo no 5V do mesmo e é ligado ao fio *data* com um resistor de 4.7k Ohms ou de 10k Ohms. O *Data* é conectado ao pino de número 10 do Arduino por ele apresentar a função PWM. Feita a ligação corretamente, o próximo passo é a programação.

O código usado para definir os endereços dos sensores utiliza o protocolo *OneWire*, mas primeiramente é necessário conceituar alguns passos.

O endereço do sensor é composto de 8 módulos, cada um formado pela sequência "0x" e um valor no sistema hexadecimal. A biblioteca específica para este protocolo (*OneWire.h*) já apresenta a função *search()* que se encarrega de encontrar o valor hexadecimal.

Desta forma, o código utiliza um *loop* cuja condição de parada é a função *search()* deixar de encontrar novos endereços. Para cada endereço, o programa entra em um novo *loop* para preenchê-lo de forma correta, iniciando cada módulo com a sequência "0x", compara se o valor encontrado é menor que 16, condição na qual um 0 (zero) deve ser adicionado para manter a cessão do módulo, e em seguida apresenta o valor encontrado pela função na forma hexadecimal. Por último, o código utiliza outra função da biblioteca, *crc8()*, cuja função é verificar a validade do endereço encontrado.

### 5.2.2 Desenvolvimento do código

Após o endereçamento é essencial começar a programação na própria IDE, pois oferece suporte para um teste vertical, o qual aponta se a placa escolhida é compatível com a tarefa desejada. Isso evita a possibilidade de comprar uma placa incompatível erroneamente.

A programação é realizada em linguagem C++ simplificada e é o cerne do dispositivo, para o funcionamento adequado é preciso utilizar bibliotecas complementares, que contam com recursos necessários para a execução de certas tarefas.

Por fim, o código fica conforme o exposto no Anexo A, o qual contempla recursos como registro de dados, leitura multicanais e impressão de dados no LCD.

### 5.2.3 Bibliotecas utilizadas

Ao aprofundar na função principal de cada biblioteca utilizada, a *OneWire* permite acessar dispositivos de 1 fio, ou seja, os sensores de temperatura. A *DallasTemperature* é uma biblioteca para circuitos integrados e é responsável pela interpretação das leituras de temperatura. Já a *SD* é a biblioteca responsável por realizar a leitura e gravação em cartão SD e possui suporte para FAT16 e FAT23. A *SPI* tem como função orquestrar a comunicação entre o microcontrolador e outros componentes que utilizam a interface de comunicação série síncrona (SPI). A *Wire*

permite a comunicação com dispositivos I2C/TWI. Por fim, a *RTCLib* tem a função para tornar a interface com o módulo DS1307 mais simples (ARDUINO, 2018).

#### 5.2.4 Montagem final

Para montar o *Hardware*, ou seja, inserir na placa Arduino Mega os dispositivos necessários para as funções de medição, são esses: módulo LCD, módulo I2C, módulo RTC DS1307, módulo mini SD e os cinco sensores DS18B20.

Toda a comunicação é realizada em SPI (Cartão SD) e I2C (RTC e LCD), no segundo caso, como existem dois dispositivos na mesma porta, é necessário discriminá-los por meio de um código de identificação, assim como os utilizados nos sensores. Para a comunicação SPI são utilizadas as portas MISO, MOSI e SCK, já para a comunicação I2C tem-se as portas SDA e SCL.

Vale salientar que todos os dispositivos estarão conectados ao 5V e ao GND e com isso são alimentados pela mesma porta.

O passo final é a conexão da placa via USB para transferir a programação para o Arduino.

#### 5.2.5 Funcionamento e obtenção de dados

Feito isso, o dispositivo está finalizado e pode ser usado para medições, as quais serão registradas no cartão micro SD. A partir do momento em que o cartão micro SD é retirado e conectado no PC, nele haverá um arquivo *.txt* com as medições feitas em cada ponto.

### 5.3 Análise de custos

O custo médio ao encontrar esses produtos na internet é o calculado na Tabela 1. Vale salientar que a confecção do protótipo é custeada pelo próprio aluno responsável.

**Tabela 1 - Preço Médio dos Componentes**

<b>Componente</b>	<b>Preço</b>
Placa Arduino MEGA 2560	R\$ 80,00
5 sensores DS18B20	R\$ 80,00
Tela LCD	R\$ 38,00
Módulo RTC DS1307	R\$ 21,00
Caixa De Passagem Sobrepor Elétrica	R\$ 28,00
Fonte de alimentação para Arduino Mega	R\$ 30,00
Módulo mini SD	R\$ 15,00
Cartão mini SD	R\$ 25,00
Módulo I2C	R\$ 12,00
<b>Total</b>	<b>R\$ 329,00</b>

**Fonte: Autoria própria**

Dentre os gastos, vale salientar que houve um dispositivo, o *shield Datalogger* RTC DS1307, no valor de R\$ 35,00, que não se mostrou compatível com o Arduino Mega 2560 e não consta na Tabela 1, com isso foi necessário realizar a compra dos módulos Micro SD e RTC DS1307 para assim fazer a substituição.

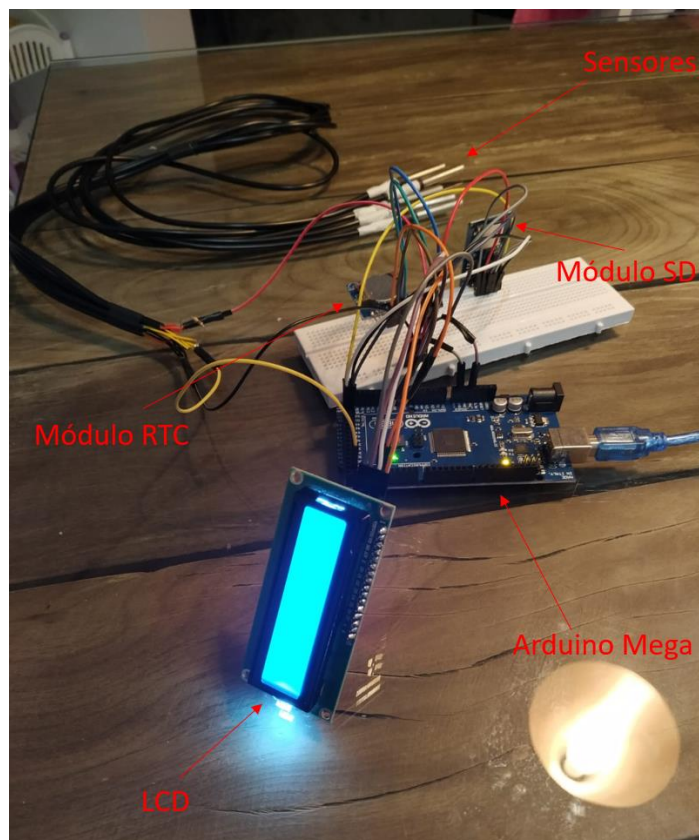
O valor apresentado prova um custo-benefício extremamente relevante, dado que é 43,3% menor do que o valor do termômetro de cinco pontos comercial da FullGauge. Se for comparado com um registrador de dados multicanais da Omega, a diferença de preço chega a 94,2%.

## 6 RESULTADOS E DISCUSSÃO

### 6.1 Protótipo finalizado

A partir da metodologia explicitada foi confeccionado o protótipo do produto com as devidas configurações, na Figura 9 encontra-se a imagem do protótipo. O próximo passo para satisfazer os objetivos do trabalho foi realizar uma bateria de testes em situações diversas para comprovar a eficiência e precisão do dispositivo.

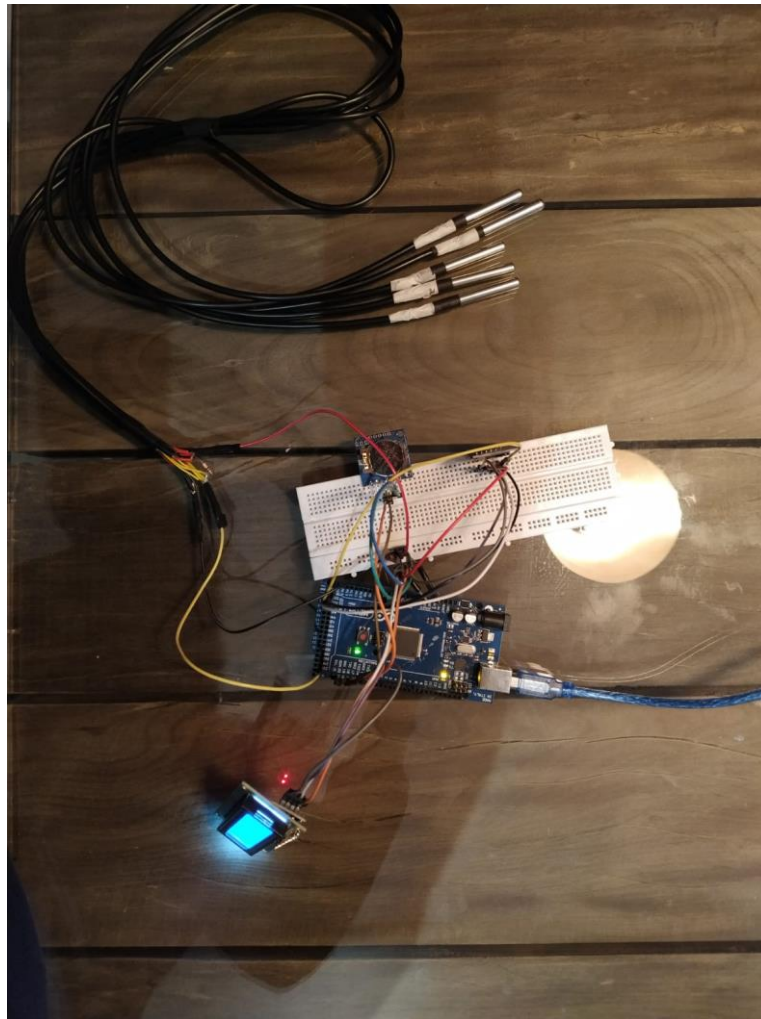
Figura 9 - Protótipo Finalizado



Fonte: Autoria propria

Ao visualizar de outro ângulo (Fig. 10) é possível observar os componentes vitais do sistema, como os módulos micro SD, RTC, além dos sensores.

**Figura 10 - Sistema visto de outro ângulo**



**Fonte: Autoria propria**

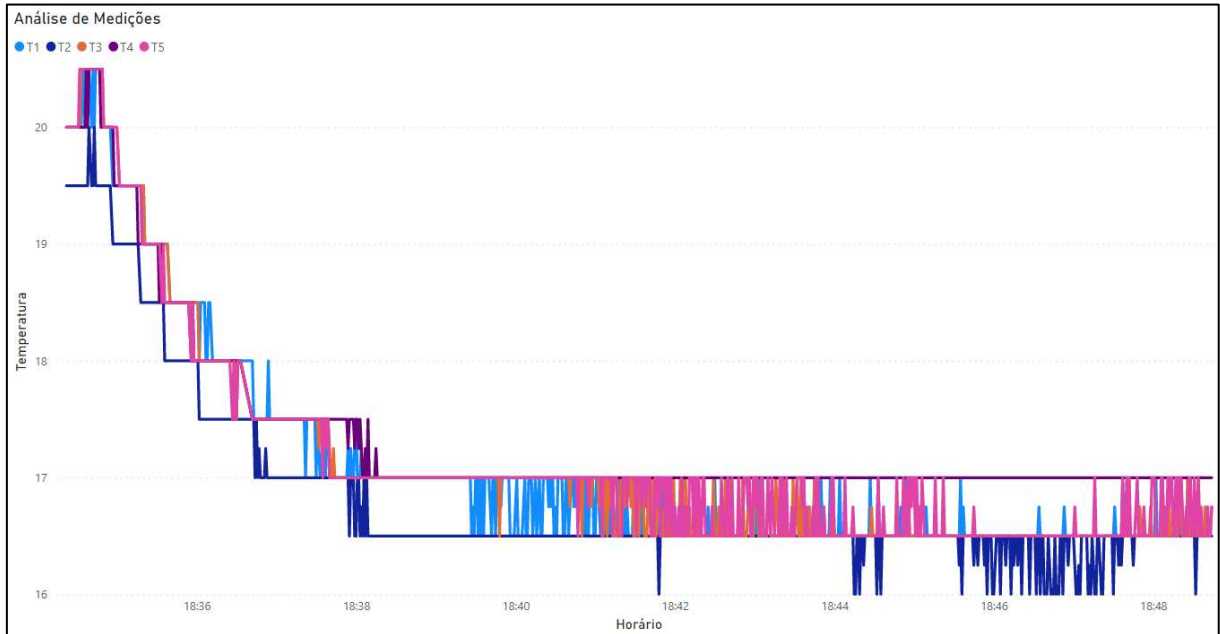
O primeiro ponto de destaque do sistema confeccionado a ser notado já na execução é sua possibilidade de definir qualquer intervalo para realização das medições, para checar as repercussões disso foram configuradas para todos os testes intervalos inferiores a 1 segundo.

#### 6.1.1 Primeiro teste e seus desafios

A priori foi realizado um teste que consiste em 10 minutos de medição com intervalo de 0,5 segundos entre cada uma, o intuito desse teste é checar a repetibilidade e consistência das medições, dado que o meio medido era o ambiente e a variação de temperatura em 10 minutos é causada pela mudança de ambiente dos sensores, pois eles estavam com a temperatura superficial compatível com o

interior da residência e foram realocados para o exterior. Com isso, obteve-se exatas 1092 medições que apontaram a seguinte dispersão apresentada na Figura 11.

**Figura 11 - Curva de temperatura com dados brutos**



**Fonte: Autoria própria**

Como pode-se observar, o gráfico gerado é extremamente denso e de difícil leitura, isso ocorre devido ao grande volume de dados por segundo, dado que a frequência de medição estava abaixo de 1000 milissegundos. Para considerar a alta taxa de medições, mas ainda assim remover a poluição visual deixada por esse fato, foi realizado um tratamento nos dados que consistia em sintetizar as medições por minuto, mas sem deixar de evidenciar diversos fatores importantes entre os sensores, como o desvio padrão, a medição máxima, mínima, média e a variação.

### 6.1.2 Tratamento e visualização de dados

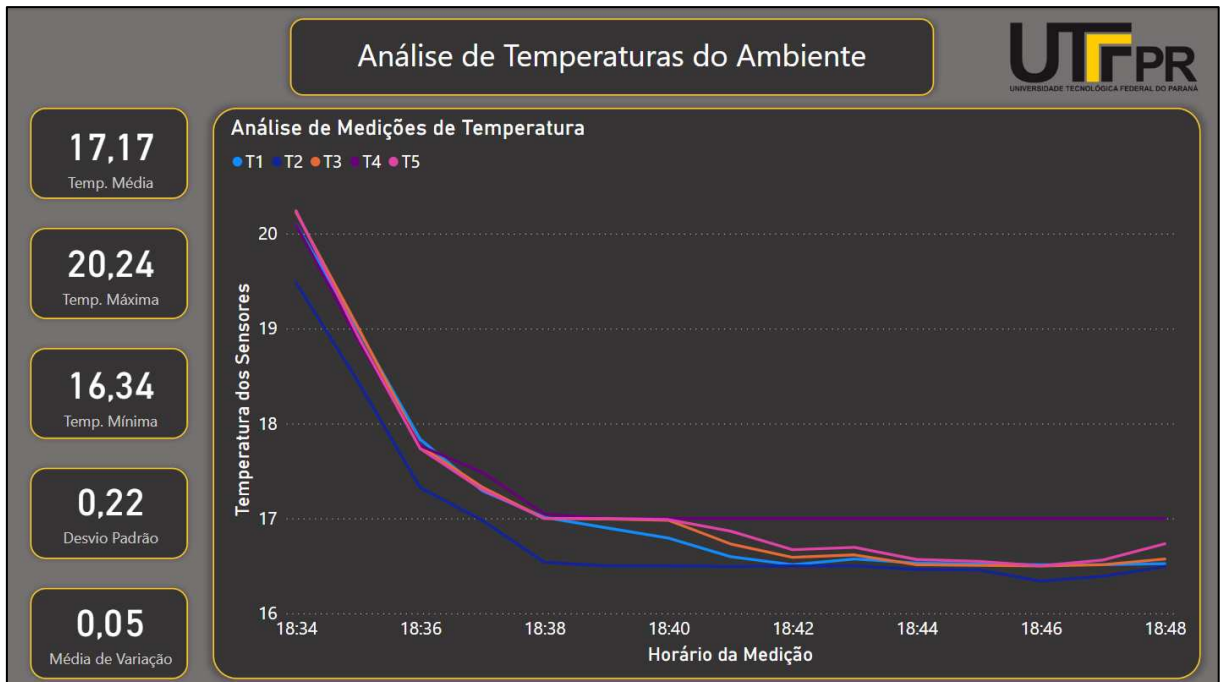
Para a interpretação, disposição e interação de dados, foi escolhido o software *Power BI*, dado que, com o computador, se torna *Hub* de informações no qual seja possível analisar diversas características (temperatura média, por exemplo) com apenas alguns cliques, o que lembra sistemas como os da Omega e outras empresas de instrumentação industrial, as quais disponibilizam softwares robustos de análise.

Com isso, é necessário realizar um tratamento dos dados coletados e realizar o relacionamento correto da tabela para que os itens corretos sejam interligados na consulta.



Na Figura 12 é apresentado o gráfico com os valores já tratados.

Figura 12 - Dashboard interativo com gráfico de medições do ambiente



Fonte: Autoria propria

Isso evidencia que a variação de temperatura se encontra dentro do *range* de imprecisão estipulado pelo fornecedor do sensor DS18B20 (0,5°C), contudo vale salientar que também foi observado que, apesar de se encontrarem no mesmo ambiente e nas mesmas condições, as temperaturas dos sensores raramente entravam em consenso absoluto (cinco medições idênticas), porém apontavam sempre diferenças dentro da imprecisão informada anteriormente.

### 6.1.3 Segundo teste

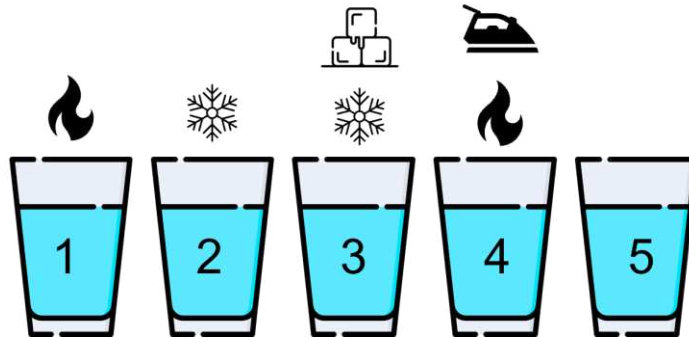
Já para analisar a possibilidade de medições em cinco pontos com condições distintas foi realizado um experimento com cinco copos de água, como apresentado na Figura 13, cada um com uma condição e temperatura inicial, da seguinte forma:

- Copo 1 (um) com água fervida e exposto ao ambiente;
- Copo 2 (dois) com água gelada e exposto ao ambiente;
- Copo 3 (três) com água gelada e um cubo de gelo (inserido após 2 minutos de teste), esse copo está exposto ao ambiente;



- Copo 4 (quatro) com água fervida e ao lado de um ferro de passar (desligado após 1 hora e 50 minutos de teste), ambos expostos ao ambiente;
- Copo 5 (cinco) com água natural exposto ao ambiente;

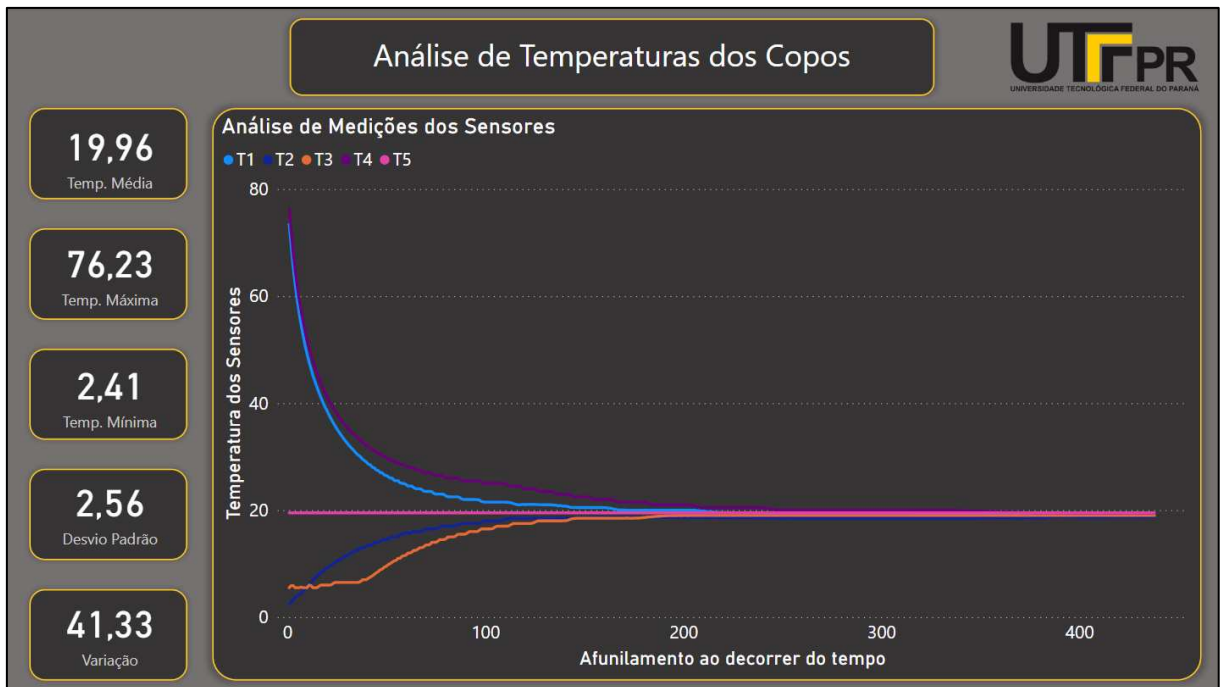
**Figura 13 – Esquema do experimento com copos**



Fonte: Autoria própria

As medições duram até que a temperatura em cada um desses copos atinja um equilíbrio. A curva de temperatura em cada um dos sensores fora plotada no *Power BI* e obteve-se o seguinte comportamento evidenciado na Figura 14.

Figura 14 - Dashboard interativo com gráfico de medições dos copos



Fonte: Autoria Própria

Com isso, pode-se notar que o termômetro de cinco pontos possui a capacidade de realizar medições simultâneas onde cada uma delas apresenta temperaturas significativamente diferentes. Isso traz à tona a usabilidade desse termômetro em análises de refrigeração, como no teste comparativo de um ar-condicionado, por exemplo.

#### 6.1.4 Terceiro teste

Para testar a robustez da quantidade de medições, foi realizada uma bateria de medidas de 8 horas e 40 minutos. O que gerou uma tabela de 42109 linhas, da qual tem-se o trecho referenciado na Tabela 3.

Tabela 2 - Trecho das medições de repetibilidade

Horário	T1 (°C)	T2 (°C)	T3 (°C)	T4 (°C)	T5 (°C)
04:40:29	19	19	19	19,5	19,5
04:40:30	19	19	19	19,5	19,5
04:40:30	19	19	19	19,5	19,5
04:40:31	19	19	19	19,5	19,5
04:40:32	19	19	19	19,5	19,5
04:40:33	19	19	19	19,5	19,5
04:40:33	19	19	19	19,5	19,5
04:40:34	19	19	19	19,5	19,5
04:40:35	19	19	19	19,5	19,5
04:40:35	19	19	19	19,5	19,5
04:40:36	19	19	19	19,5	19,5
04:40:37	19	19	19	19,5	19,5
04:40:38	19	19	19	19,5	19,5
04:40:38	19	19	19	19,5	19,5
04:40:39	19	19	19	19,5	19,5
04:40:40	19	19	19	19,5	19,5
04:40:41	19	19	19	19,5	19,5
04:40:41	19	19	19	19,5	19,5
04:40:42	19	19	19	19,5	19,5
04:40:43	19	19	19	19,5	19,5
04:40:44	19	19	19	19,5	19,5
04:40:44	19	19	19	19,5	19,5
04:40:45	19	19	19	19,5	19,5
04:40:46	19	19	19	19,5	19,5

Fonte: Autoria própria

A partir disso é possível observar a facilidade, praticidade e objetividade do sistema ao realizar uma bateria longa de medições. A partir de uma regra de três é possível descobrir quantas medições o cartão micro SD suporta por *Gigabyte* de capacidade de armazenamento.

$$\frac{42109 \text{ medições}}{2,445 \text{ MB}} = \frac{x \text{ medições}}{1 \text{ GB}} \quad (2)$$

$$x = 17.222.494 \text{ medições} \quad (3)$$

### 6.1.5 Visualização via LCD

Outro ponto importante a ser resolvido é a praticidade na tomada de decisões, isso pode ser facilitado por meio de um visor LCD que imprime a temperatura do

momento. Para utilizar menos portas (redução de 16 para 4), em conjunto com o *shield* LCD, foi acoplado um módulo I2C.

## **6.2 Oportunidade de melhoria**

Ainda sobre a experiência com o sistema, houve um *insight* de melhoria futura no projeto. A partir dos trâmites com o registro e tratamento dos dados das medições, surgiu a ideia de, em um momento oportuno, realizar a adição de um *shield Ethernet* no sistema e através da Internet das Coisas realizar um upload e acompanhamento simultâneo dos dados, como nas portas lógicas utilizadas na indústria. Isso pode acrescentar complexidade e capacidade de resposta a toda e qualquer aplicação do sistema.

## 7 CONCLUSÃO

Pela observação dos aspectos analisados, é cabível afirmar que o termômetro *open source* de cinco pontos é uma alternativa totalmente viável para as mais diversas aplicações (de pesquisas a análises técnicas), o qual contempla tanto situações que demandam menos complexidade (graças ao LCD), como em análises pontuais, quanto situações que necessitam de uma base de dados bem estruturada (por meio de análises do *Power BI* aliado aos registros por tempo) e com o devido tratamento de dados.

Observa-se o leque de oportunidades de melhorias e adaptações que o sistema Arduino possibilita para o dispositivo, o que o permite navegar entre os mais variados tipos de termômetros e sistemas de aquisição, basta definir um equipamento comercial como parâmetro e realizar alterações direcionadas.

Por isso tudo, é de extremo valor o estudo focado em otimizar a instrumentação de baixo custo, dado que isso proporciona a realização de projetos com complexidade média ou elevada ao usar um equipamento de custo relativamente baixo, rendimento adequado à demanda e oportunidade de melhoria contínua.

## REFERÊNCIAS

AMORIM, Helio Salim do, Dias, Marco Adriano e Soares, Vitorvani. Sensores digitais de temperatura com tecnologia one-wire: Um exemplo de aplicação didática na área de condução térmica. **Revista Brasileira de Ensino de Física** [online]. 2015, v. 37, n. 4, p. 4310-1-4310-9. Disponível em: <<https://doi.org/10.1590/S1806-11173742009>>. Acesso em: 22 de novembro de 2021.

ARDUINO. **Arduino**, 2021. Disponível em: <<https://www.arduino.cc>>. Acesso em: 24 de outubro de 2021.

AZUA-BARRON, Martín et al. Sistema de adquisición de datos de bajo costo con la plataforma arduino. **Rev. Mex. Cienc. Agríc**, Texcoco. 2017, v. 8, n. 1, p. 1-12. Disponível em <<https://doi.org/10.29312/remexca.v8i1.67>>. Acesso em: 9 de novembro de 2021.

BADAMASI, Y. A. "The working principle of an Arduino". **2014 11th International Conference on Electronics, Computer and Computation (ICECCO)**, 2014, p. 1-4, Disponível em: <<https://doi.org/10.1109/ICECCO.2014.6997578>>. Acesso em: 9 de outubro de 2021.

CORDOVA, Hercílio P. et al. Audietermômetro: um termômetro para a inclusão de estudantes com deficiência visual. **Revista Brasileira de Ensino de Física** [online]. 2018, v. 40, n. 2, e2505. Disponível em: <<https://doi.org/10.1590/1806-9126-RBEF-2017-0299>>. Acesso em: 19 de novembro de 2021.

D'AUSILIO, A. Arduino: A low-cost multipurpose lab equipment. **Behavior Research Methods**. 2012, v. 44, p. 305–313. Disponível em: <<https://doi.org/10.3758/s13428-011-0163-z>>. Acesso em: 7 de novembro de 2021.

DE ALMEIDA, Marcos M. Experimento de baixo custo para medição da dissipação da energia em um pêndulo. **Revista Brasileira de Ensino de Física** [online]. 2021, v. 43, e20200360. Disponível em: <<https://doi.org/10.1590/1806-9126-RBEF-2020-0360>>. Acesso em: 10 de novembro de 2021.

DOS SANTOS, Rodrigo Vieira. Uso Do Arduino E Shield Ethernet Para Monitoramento De Luminosidade, Controle De Temperatura E Dispositivos. **Revista Científica Semana Acadêmica**. Fortaleza, 2018, n. 000129, 13/08/2018. Disponível em: <<https://semanaacademica.org.br/artigo/uso-do-arduino-e-shield-ethernet-para-monitoramento-de-luminosidade-controle-de-temperatura-e>>. Acesso em: 7 de novembro de 2021.

HAY, Bruno. A brief history of the thermal properties metrology, **Measurement**. 2020, v. 155, 107556. Disponível em: <<https://doi.org/10.1016/j.measurement.2020.107556>>. Acesso em: 2 de novembro de 2021.

INSTITUTO NACIONAL DE METROLOGIA NORMALIZAÇÃO E QUALIDADE INDUSTRIAL – INMETRO. **Sistema Internacional de Unidades**. Rio de Janeiro. 2012. 94p. Disponível em: <[http://www.inmetro.gov.br/inovacao/publicacoes/si\\_versao\\_final.pdf](http://www.inmetro.gov.br/inovacao/publicacoes/si_versao_final.pdf)>. Acesso em: 10 de novembro de 2021.

INVENÇÕES da Climatologia. **Timetoast**, c2007. Disponível em: <<https://www.timetoast.com/timelines/invencoes-da-climatologia>>. Acesso em: 20 de novembro de 2021.

KOESTER, R.A.; Saleh, Y.A.; Roihan, I.; Harinaldi. A simple method for calibration of temperature sensor DS18B20 waterproof in oil bath based on Arduino data acquisition system. **AIP Conf. Proc.** 2019, v. 2032, 020006. Disponível em: <<https://doi.org/10.1063/1.5086553>>. Acesso em: 13 de novembro de 2021.

MARTINAZZO CA, ORLANDO T. Comparação Entre Três Tipos de Sensores de Temperatura em Associação Com Arduino. **Perspectiva**. 2016, v. 40, n. 151, p. 93-104. Disponível em: <[https://www.uricer.edu.br/site/pdfs/perspectiva/151\\_587.pdf](https://www.uricer.edu.br/site/pdfs/perspectiva/151_587.pdf)>. Acesso em: 23 de novembro de 2021.

MCROBERTS, M. **Arduino Básico**. Rafael Zanolli. São Paulo: Novatec Editora, 2011. 456 p.

PENTA iii. **Fullgauge**, c2021. Disponível em: <<https://www.fullgauge.com.br/produto-penta-iii>>. Acesso em: 1 de novembro de 2021.

PIRES, Denise Prazeres Lopes, Afonso, Júlio Carlos e Chaves, Francisco Artur Braun. Do termoscópio ao termômetro digital: quatro séculos de termometria. **Química Nova** [online]. 2006, v. 29, n. 6, p. 1393-1400. Disponível em: <<https://doi.org/10.1590/S0100-40422006000600041>>. Acesso em: 27 de outubro de 2021.

REGIS-SILVA et al. Comparação entre um termômetro construído com arduino e termômetros usados no mercado. **Rev. Bras. Eng. Pesca** .2017, v.10, n.2, p. 140-147. Disponível em: <<https://ppg.revistas.uema.br/index.php/REPESCA/article/view/1366/1131>>. Acesso em: 25 de outubro de 2021.

SENSOR de Temperatura DS18B20 a Prova D'água. **Eletrogate**, 2021. Disponível em: <<https://www.eletrogate.com/sensor-de-temperatura-ds18b20-a-prova-dagua>>. Acesso em: 16 de outubro de 2021.

SEVERANCE, Charles. "Massimo Banzi: Building Arduino," in **Computer**. 2014, vol. 47, no. 1, p. 11-12. Disponível em: <<https://doi.org/10.1109/MC.2014.19>> Acesso em: 29 de outubro de 2021.

OMEGA. **Omega Engineering Inc**, c2015. Disponível em: <<https://br.omega.com>>. Acesso em 11 de junho de 2022.

SHIELDS para Arduino. **Usinainfo**, 2021. Disponível em: <<https://www.usinainfo.com.br/8-nossa-qualidade>>. Acesso em: 30 de outubro de 2021.

TERMÔMETRO clínico digital incoterm a prova d'agua. **Hospinet**, 2021. Disponível em: <<https://www.hospinet.com.br/termometro-clinico-digital-incoterm-a-prova-de-agua/p>>. Acesso em: 28 de outubro de 2021.

VILAR, A.B. et al. Medição de temperatura: O saber comum ignorado nas aulas experimentais. **Revista Brasileira de Ensino de Física** [online]. 2015, v. 37, n. 2, p. 2507-1-2507-5. Disponível em: <<https://doi.org/10.1590/S1806-11173721770>>. Acesso em: 9 de novembro de 2021.



## ANEXO A - Código do sistema

```
// bibliotecas e classes necessárias

#include <OneWire.h>

#include <SD.h>

#include <SPI.h>

#include <DallasTemperature.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <RTCLib.h>

#include "RTCLib.h"

RTC_Millis rtc;

RTC_DS1307 RTC;

byte Rele1 = 2;

File logfile;

const int ONE_WIRE_BUS = 22; // porta onde conectarei os sensores

OneWire ourWire(ONE_WIRE_BUS);

DallasTemperature sensors(&ourWire);

// DEFINIÇÕES

#define LOG_INTERVAL 2000 //define o intervalo entre o registro de duas medidas,
neste caso 20000 ms, ou 20s.

#define SYNC_INTERVAL 10000 // milisegundos entre a chamada para limpar e
para gravar dados no cartão

uint32_t syncTime = 0; // tempo da ultima sincronização

#define ECHO_TO_SERIAL 1 // echo data to serial port

#define WAIT_TO_START 0 // Wait for serial input in s
```

```

#define endereco 0x27 // Endereços comuns: 0x27, 0x3F

#define colunas 16

#define linhas 2

LiquidCrystal_I2C lcd(endereco, colunas, linhas); //definindo comunicação com LCD

// identificação dos sensores

DeviceAddress T1 = {0x28, 0xD7, 0xEB, 0x01, 0x00, 0x00, 0x00, 0x7A};
DeviceAddress T2 = {0x28, 0xD7, 0xB9, 0x01, 0x00, 0x00, 0x00, 0x6C};
DeviceAddress T3 = {0x28, 0x2C, 0x94, 0x01, 0x00, 0x00, 0x00, 0xF2};
DeviceAddress T4 = {0x28, 0x93, 0x95, 0x01, 0x00, 0x00, 0x00, 0x1C};
DeviceAddress T5 = {0x28, 0x66, 0xD1, 0x01, 0x00, 0x00, 0x00, 0x73};

const int chipSelect = 53; // pino principal relacionado ao cartao SD no arduino

int incomingByte;

float temp; // variaveis q vao receber temperaturas

float tp[50]; //criação de vetor pra armazenar temps

float tempC; // temp em Celsius

void setup(void) //executar uma vez somente ao ligar
{
  Serial.begin(9600);

  Serial.println();

  pinMode (Rele1, OUTPUT);

  File logfile;

```

```
#if WAIT_TO_START // teste de funcionamento

  Serial.println("Type any character to start");

  while (!Serial.available());

#endif //WAIT_TO_START

// inicializando cartao SD

Serial.print("Initializing SD card...");

pinMode(chipSelect, OUTPUT);

// Checkup do cartão SD

if (!SD.begin(chipSelect)) {

  Serial.println("Card failed, or not present");

  return;

}

Serial.println("card initialized.");

// criação do arquivo CSV no cartão SD

char filename[] = "DADOS00.CSV";

for (uint8_t i = 0; i < 100; i++) {

  filename[6] = i / 10 + '0';

  filename[7] = i % 10 + '0';

  if (!SD.exists(filename)) {

    logfile = SD.open(filename, FILE_WRITE);

    break;

  }

}

Serial.begin(9600);
```

```
Serial.println();

#if WAIT_TO_START

Serial.println("Type any character to start");

while (!Serial.available());

#endif //WAIT_TO_START

// Se o RTC não funcionar, para setar o tempo:

// if (!RTC.isrunning()) {

//   Serial.println("RTC is running!");

//   A seguinte linha seta o RTC para a data de compilação do Sketch

//   RTC.adjust(DateTime(2022, 6, 5, 19, 20, 0));

//   RTC.begin(DateTime(2022, 5, 26, 21, 27, 15));

// }

//inicialização dos dispositivos

Wire.begin();

RTC.begin();

sensors.begin();

lcd.begin(16, 2);           // inicializa LCD

lcd.backlight();          // ativa led de backlight

sensors.setResolution(T1, 9);

sensors.setResolution(T2, 9);

sensors.setResolution(T3, 9);

sensors.setResolution(T4, 9);

sensors.setResolution(T5, 9);
```

```
}  
  
void displaytemperatura(int indicesensor)  
{  
    lcd.clear();  
  
    float temp_C = sensors.getTempCByIndex(indicesensor); // Envie o comando para  
    obter leituras de temperatura  
  
    lcd.setCursor(0, 0);  
  
    lcd.print(temp_C, 1);  
  
    lcd.write(B11011111); // imprime o simbolo de grau  
  
    lcd.print("C");  
  
    lcd.setCursor(0, 1);  
  
    float temp_F = DallasTemperature::toFahrenheit(temp_C); // Envie o comando para  
    obter leituras de temperatura  
  
    lcd.print(temp_F, 1);  
  
    lcd.write(B11011111); // imprime o simbolo de grau  
  
    lcd.print("F");  
}  
  
//teste de funcionamento dos sensores  
  
void printTemperature(DeviceAddress deviceAddress)  
{  
    tempC = sensors.getTempC(deviceAddress);  
  
    delay (10);  
  
    if (tempC == -127.00) {  
        Serial.print("Erro ao ler temperatura !");  
    } else {
```

```
    }  
}  
  
//atividades realizadas periodicamente  
void loop(void)  
{  
    DateTime now = RTC.now();  
  
    sensors.requestTemperatures(); // puxar dados do Sensor de temperatura  
  
    temp = sensors.getTempCByIndex(0);  
  
    sensors.requestTemperatures();  
  
    Serial.print("\n\r");  
  
    Serial.print(" ");  
  
    Serial.print("Temperatura Sensor 1: ");  
  
    printTemperature(T1);  
  
    tp[1] = tempC;  
  
    Serial.print(tp[1]);  
  
    Serial.print("\n\r");  
  
    Serial.print(" ");  
  
    Serial.print("Temperatura Sensor 2: ");  
  
    printTemperature(T2);  
  
    tp[2] = tempC;  
  
    Serial.print(tp[2]);  
  
    Serial.print("\n\r");  
  
    Serial.print(" ");  
  
    Serial.print("Temperatura Sensor 3: ");
```

```
printTemperature(T3);

tp[3] = tempC;

Serial.print(tp[3]);

Serial.print("\n\r");

Serial.print(" ");

Serial.print("Temperatura Sensor 4: ");

printTemperature(T4);

tp[4] = tempC;

Serial.print(tp[4]);

Serial.print("\n\r");

Serial.print(" ");

Serial.print("Temperatura Sensor 5: ");

printTemperature(T5);

tp[5] = tempC;

Serial.print(tp[5]);

Serial.print("\n\r");

Serial.print(" ");

// registro de horario serial

Serial.print("\n\r\n\r");

Serial.print(now.year(), DEC);

Serial.print('/');

Serial.print(now.month(), DEC);

Serial.print('/');

Serial.print(now.day(), DEC);
```

```
Serial.print(' ');  
Serial.print(now.hour(), DEC);  
Serial.print(':');  
Serial.print(now.minute(), DEC);  
Serial.print(':');  
Serial.print(now.second(), DEC);  
Serial.println();  
File dataFile = SD.open("DADOS00.txt", FILE_WRITE);  
  
// registro de horario  
if (dataFile)  
    dataFile.print("\n\r");  
dataFile.print(' ');  
dataFile.print("\n\r\n\r");  
dataFile.print (now.year(), DEC);  
dataFile.print('/');  
dataFile.print(now.month(), DEC);  
dataFile.print('/');  
dataFile.print(now.day(), DEC);  
dataFile.print(' ');  
dataFile.print(now.hour(), DEC);  
dataFile.print(':');  
dataFile.print(now.minute(), DEC);  
dataFile.print(':');
```



```
dataFile.print(now.second(), DEC);  
  
//registro de temperatura no arquivo  
  
dataFile.print(" ");  
  
dataFile.print(tp[1]);  
  
dataFile.print(" ");  
  
dataFile.print(tp[2]);  
  
dataFile.print(" ");  
  
dataFile.print(tp[3]);  
  
dataFile.print(" ");  
  
dataFile.print(tp[4]);  
  
dataFile.print(" ");  
  
dataFile.print(tp[5]);  
  
dataFile.print(" ");  
  
dataFile.close();  
  
  displaytemperatura(0); //chama a função que lê o sensor e exibe a mensagem na  
  tela do LCD  
  
  // TEMPO DE COLETA DE DADOS  
  
  delay (2000); //ms  
  
}
```