



**UNIVERSIDADE DO ESTADO DA BAHIA**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA**  
**CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

**WENDELL DE FREITAS VALOIS RIOS**

**UM COLETOR DE DADOS AUTÔNOMO, BASEADO EM ARDUINO, PARA  
EXPERIMENTOS DIDÁTICOS**

**SALVADOR**

**2021**

WENDELL DE FREITAS VALOIS RIOS

UM COLETOR DE DADOS AUTÔNOMO, BASEADO EM ARDUINO, PARA  
EXPERIMENTOS DIDÁTICOS

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Orientador: Cláudio Alves de Amorim

SALVADOR

2021

FICHA CATALOGRÁFICA  
Sistema de Bibliotecas da UNEB  
Dados fornecidos pelo autor

R586u

Rios, Wendell de Freitas Valois

Um coletor de dados autônomo, baseado em Arduino, para experimentos didáticos / Wendell de Freitas Valois Rios.-- Salvador, 2021.

48 fls : il.

Orientador(a): Cláudio Alves de Amorim.

Inclui Referências

TCC (Graduação - Sistemas de Informação) - Universidade do Estado da Bahia. Departamento de Ciências Exatas e da Terra. Campus I. 2021.

1.Arduino. 2.Ensino de física. 3.Radiação solar.

CDD: 000

WENDELL DE FREITAS VALOIS RIOS

UM COLETOR DE DADOS AUTÔNOMO, BASEADO EM ARDUINO, PARA  
EXPERIMENTOS DIDÁTICOS

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

---

Cláudio Alves de Amorim (Orientador)  
Universidade do Estado da Bahia – UNEB

---

Leandro Santos Coelho de Souza  
Universidade do Estado da Bahia - UNEB

---

Diego Gervasio Frías Suárez  
Universidade do Estado da Bahia - UNEB

## **AGRADECIMENTOS**

Agradeço a meu irmão Rodrigo que serve de inspiração e exemplo de determinação, sempre me apoiando em tudo que faço e sempre sendo um dos melhores e mais fieis amigos que tenho. Agradeço também a Lucas pela algazarra e risos de manhã cedo e pelos puxões de orelha quando percebia falhas ou partes faltantes em trabalhos da faculdade.

Agradeço imensamente ao meu orientador Cláudio Amorim que me mostrou o caminho das pedras e apontou uma alternativa inusitada de trabalho científico, além de fornecer aulas fantásticas sobre assuntos que nunca irei esquecer. Agradeço a todos os meus amigos da UNEB que sempre me motivaram em todas as etapas de graduação a seguir em frente e não desistir.

"O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado."

(Ariano Suassuna)

## RESUMO

O seguinte estudo busca trazer um ambiente de coleta de dados com baixo custo, usando Arduino, independente de computador. Este estudo se deu devido a necessidade de uma opção mais flexível para coleta de dados. O projeto possui o intuito de prosseguir e fazer parte de uma série de outros trabalhos que visam engajar alunos e professores por meios de ambientes didáticos, com foco no estudo da física, de maneira mais estimulante e significativa através de aparatos modulares e configuráveis. Como prova de conceito, foi realizado a coleta de dados em um experimento para observar o fenômeno da radiação em que latas pintadas de cores diferentes recebem luz solar em um intervalo de tempo superior a duas horas. O objetivo dos testes práticos e empíricos é de constatar a influência das cores na absorção e dissipação de calor. Foi constatado em gráficos a visível diferença entre a temperatura das diferentes latas e a influência do clima em tempos ensolarados e nublados. Com isso foi comprovado a eficiência do modelo proposto, visto que, outros tipos de experimentos diferentes podem ser executados com o mesmo aparato, incluindo diferentes tipos de sensores, facilitando e automatizando medições que com uso de instrumentos convencionais como termômetros analógicos seriam repetitivos, tediosos e por vezes imprecisos.

**Keywords:** Arduino. Física. Temperatura. Radiação. Interface. Coleta de dados.

## **ABSTRACT**

This study brings a low cost data collection environment based in the Arduino platform, to help Physics learning in a more computer independent way. This study was created in the need for a data collection system that is more flexible, allowing to be unplugged from a computer and also placed in outdoor environments. The project aims to follow a series of other works that seeks to engage students and teachers in the learning of physics, by a more practical approach using modular and configurable devices. As a proof of concept, the module collected temperature data in different weather conditions from color painted tin cans placed outdoors, in order to, observe the radiation phenomenon. The purpose of this practical and empirical tests is to verify the influence of colors on heat absorption and dissipation. As a result, the graphics generated by the project showed a visible difference between the temperature from the different cans in sunny and cloudy weather conditions, specially in the white and black tin cans. In conclusion this model proved to work as intended and can be customized to different experiments only alternating sensors, which automates and makes easier data collection, differently from conventional methodologies that uses instruments such as analogical thermometers in ways considered repetitive, tedious and sometimes inaccurate.

**Keywords:** Arduino. Physics. Temperature. Radiation. Interface. Data collect. Learning.

## LISTA DE FIGURAS

Figura 1 – Esquema básico do Arduino . . . . .	17
Figura 2 – Circuitos Integrados e Microcontroladores . . . . .	19
Figura 3 – Tipos de troca de calor . . . . .	21
Figura 4 – Transferência de calor por radiação . . . . .	23
Figura 5 – Reflexão da cor . . . . .	25
Figura 6 – Modelo do projeto . . . . .	27
Figura 7 – Programa <i>Python</i> . . . . .	33
Figura 8 – Tintas acrílicas de artesanato . . . . .	35
Figura 9 – Pintura de lata em fase inicial . . . . .	35
Figura 10 – Experimento conectado a tomada . . . . .	36
Figura 11 – Iniciação da coleta . . . . .	37
Figura 12 – Organização do experimento . . . . .	37
Figura 13 – Dia nublado . . . . .	39
Figura 14 – Gráfico de coleta em dia nublado . . . . .	40
Figura 15 – Dia parcialmente nublado . . . . .	41
Figura 16 – Gráfico em dia parcialmente nublado . . . . .	42
Figura 17 – Dia ensolarado . . . . .	42
Figura 18 – Gráfico de latas coloridas em dia ensolarado . . . . .	43
Figura 19 – Gráfico em dia ensolarado . . . . .	43

## LISTA DE ABREVIATURAS E SIGLAS

CPU	<i>Central Process Unit</i>
EEPROM	<i>Electrical Erasable Programmable Read-Only Memory</i>
EPROM	<i>Electrical Erasable Programmable Read-Only Memory</i>
EPS	Poliestireno Expandido
IC	<i>Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IV	Infra Vermelho
LCD	<i>Liquid Crystal Display</i>
MCU	<i>Microcontroller Unit</i>
RAM	<i>Random-access memory</i>
ROM	<i>Read-Only Memory</i>
SD	<i>Secure Digital</i> - Cartão de memória
USB	<i>Universal Serial Bus</i>

## LISTA DE SÍMBOLOS

$^{\circ}C$	Graus Celsius
$V$	Volt
$mA$	Miliampère

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>ARDUINO E PERIFÉRICOS</b>	<b>16</b>
<b>2.1</b>	<b>Arduino</b>	<b>16</b>
<b>2.2</b>	<b>Microcontroladores</b>	<b>18</b>
<b>2.3</b>	<b>Memórias não voláteis</b>	<b>19</b>
<b>2.4</b>	<b>Sensores de temperatura</b>	<b>20</b>
<b>3</b>	<b>PROCESSOS DE PROPAGAÇÃO DE CALOR</b>	<b>21</b>
<b>3.1</b>	<b>Tipos de propagação de calor</b>	<b>21</b>
<b>3.2</b>	<b>Irradiação ou Radiação</b>	<b>21</b>
3.2.1	Absorção, Reflexão e Transmissão em Superfícies Reais	22
3.2.1.1	Absortividade/Absorbância	22
3.2.1.2	Reflectância/Refletividade	22
3.2.1.3	Transmissividade/Transmitância	22
3.2.2	Frações de energia	23
<b>3.3</b>	<b>Influência da cor na absorção de calor</b>	<b>24</b>
3.3.1	Cor e Calor	24
<b>4</b>	<b>DISPOSITIVO AUTÔNOMO PARA AQUISIÇÃO DE DADOS</b>	<b>26</b>
<b>4.1</b>	<b>Organização dos componentes Arduino</b>	<b>26</b>
4.1.1	LCD 16x2	26
4.1.2	Módulo I2C para LCD	27
4.1.3	Sensor de temperatura DS18B20	27
4.1.4	Módulo de cartão SD	28
<b>4.2</b>	<b>Programação Arduino</b>	<b>28</b>
4.2.1	Mensuração de tempo com função <i>millis()</i>	28
4.2.2	Pausas no Arduino: função <i>delay()</i> e uso de <i>millis()</i> para pausas	29
4.2.3	Evitando <i>overflow</i> da memória SRAM	30
4.2.4	Bibliotecas utilizadas	31
4.2.5	Algoritmo do projeto para Arduino	32

<b>4.3</b>	<b>Programação de interface Python . . . . .</b>	<b>32</b>
<b>5</b>	<b>EXPERIMENTO . . . . .</b>	<b>34</b>
<b>5.1</b>	<b>Latas e Pintura . . . . .</b>	<b>34</b>
<b>5.2</b>	<b>Captura de dados . . . . .</b>	<b>35</b>
<b>6</b>	<b>RESULTADOS . . . . .</b>	<b>39</b>
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>44</b>
<b>7.1</b>	<b>Trabalhos Futuros . . . . .</b>	<b>44</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>46</b>

## 1 INTRODUÇÃO

O ensino de física é uma tarefa bastante difícil, pois demanda muito do professor maneiras criativas de tornar os assuntos atraentes e a prender atenção dos alunos. As causas da falta de atratividade são muitas, incluindo principalmente o despreparo e desmotivação dos professores, a carência de recursos de infraestrutura para as escolas, a ausência de renovação das práticas pedagógicas, a falta de currículos mais atualizados e a escassa elaboração de materiais educacionais consistentes com novas formas de aprender e ensinar (FERNANDES, 2016a).

Além disso, como se pode ver em (SILVEIRA, 2011), matérias de exatas, como a matemática, sofrem com alguns estigmas pela sociedade. Os alunos por fim acabam visando à terminalidade, ou seja, estudam apenas para passar nas provas. Essa interpretação do aprendizado das matérias de exatas como *difícil* torna a relação de aprender e ensinar mais complicada, o papel do educador no entanto é mostrar que o termo correto a ser utilizado é o de *desafiador*, desconstruindo essa visão negativa.

Somando a estes problemas uma das queixas frequentes dos alunos é a de que professores não costumam realizar experimentos em sala de aula e poucas vezes usam laboratórios, sendo que exercícios práticos são fundamentais para a criação de aulas mais dinâmicas (FERNANDES, 2016b). Os professores acabam assim tendendo a dar preferência a aulas teóricas, embora os alunos costumam preferir o artifício de aulas práticas com experimentos diversificados. Não se deve negligenciar e abandonar as aulas práticas, pois estas facilitam na compreensão e fixação dos assuntos abordados teoricamente.

Os cursos de física, tanto no ensino médio quanto no ensino superior, também têm dificuldade de mostrar, na prática, os eventos físicos, já que é difícil achar aparelhos que se possam usar em sala de aula. (CATELAN, 2008). Algumas poucas opções de ferramentas no mercado para o ensino de física são importadas e, além do seu alto custo, são produtos fechados onde não se sabe exatamente, o seu funcionamento, suas regras e lógica interna. Um exemplo disso são os produtos da Pasco e Vernier que a duas décadas ajudam salas de aula a automatizar experimentos para o ensino de física, com sensores de fácil uso, como sensores de temperatura, humidade, espectômetros, força e aceleração, etc. No entanto essas ferramentas são

financeiramente proibitivas para maioria das instituições, pois são produtos importados e de alto custo. (HUANG, 2015)

Um dos artefatos encontrados atualmente com o poder de se tornar uma boa ferramenta para a facilitação do ensino de física é o uso de computadores para simulação, modelização e monitoramento. De acordo com (FIOLHAIS; TRINDADE, 2003) os computadores não trazem uma ameaça de substituição ao professor, assim como, embora pareça ser bastante promissor também não serve como um artefato mágico que resolve os problemas do ensino. Resumidamente a tecnologia por si só não é o suficiente, cabe aos professores um maior esforço em fazer com que as tecnologias sejam bem aplicadas, sendo esta apenas uma ferramenta que serve para uma finalidade que é o ensino.

Com a popularização de computadores e componentes eletrônicos também fica cada vez mais fácil e menos custoso criar ferramentas simples com a utilização de sensores que possam contribuir para a observação e investigação de fenômenos físicos. Com um computador e um microcontrolador Arduino é possível criar diversos tipos de ambientes diferentes para testar e consolidar assuntos vistos na teoria relacionados a fenômenos físicos, podendo também ser utilizado em sala de aula(SOUZA et al., 2011). A simplicidade do Arduino facilita no entendimento de conceitos abstratos e atrelado ao seu uso de sensores, é possível extrair dados sobre o ambiente como temperatura, umidade, luminosidade, pressão atmosférica, gás, som, magnetismo, infravermelho, entre outros. Tais sensores tem tamanho compacto e fácil uso. Embora o tamanho dos componentes atuais sejam consideravelmente pequeno é possível maiores miniaturizações, pois o Arduino serve com maior propósito a prototipação de sistemas, e não a de produtos finais comercializáveis.

Não é uma tarefa difícil encontrar projetos para o aprendizado de física usando Arduino como o seu componente principal. Em meio acadêmico há exemplos do uso de Arduino e microcontroladores na criação ferramentas para a observação de fenômenos físicos, a exemplo de (ROSA; GIACOMELLI; TRENTIN, 2014) que utiliza sensores de temperatura pra observar um experimento de condução térmica entre duas barras metálicas de propriedades diferentes.

O que foi constatado é que maioria dos projetos vistos em meios acadêmicos e na própria Universidade Estadual da Bahia para fins didáticos como: (AMORIM; CONCEIÇÃO, 2019), (ROSA; GIACOMELLI; TRENTIN, 2014), (ROCHA; MARANGHELLO; LUCCHESI, 2014); sempre continham um computador com tela, seja desktop, laptop, ou Raspbery Pi atrelado

ao Arduino servindo de ponte para a coleta de dados. Em experimentos que necessitam a realização de coleta em ambientes externos acaba por dificultar e por vezes invalidar o uso de computadores em conjunto com o Arduino para monitoramento de fenômenos em tempo real.

Com relação a este problema foi elaborado este projeto de pesquisa que destina na criação de um modelo de interface interacional para Arduino, auxiliando na coleta de dados e monitoração de fenômenos físicos para ensino didático, tentando minimizar o uso de um computador em campo. Como prova de conceito foram realizados testes em método empírico observando a influência das cores na absorção de calor através da coleta de temperatura do interior de latas coloridas expostas ao sol.

Este projeto não visa criar um estudo de fenômenos físicos sobre o estudo de radiação de forma completamente inovadora, pois já se encontra em meio acadêmico vários estudos sobre o tema, inclusive utilizando ferramentas e artefatos em nível mais profissional como pode ser visto em (FRAGATA; SENS; SEBRÃO, 2015). Reiterando, o objetivo deste projeto é a de criar o modelo de um sistema coletor de dados com baixo custo utilizando Arduino, que serve para uso didático com possibilidade de ser personalizado para outros tipos de experimentos. Há também a elaboração da parte de algoritmo para coleta de dados e interação do sistema por meio de menus, além de um software para geração de gráficos com os dados obtidos.

## 2 ARDUINO E PERIFÉRICOS

Uma das ferramentas mais populares de prototipação atualmente é o Arduino por causa da velocidade e simplicidade que se consegue desenvolver algo devido sua modularidade e farta documentação, além da diversidade de módulos disponíveis. (MCROBERTS, 2018)

### 2.1 ARDUINO

De acordo com o site *Arduino.cc*, o Arduino é uma plataforma eletrônica baseada em *hardware* e software open-source de fácil utilização (*easy-to-use*) (ARDUINO, 2020b). Deste ecossistema do Arduino o *hardware* foi projetado na Itália em 2005 por 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Eles tinham como objetivo criar um dispositivo fácil de aprender e de programar, barato e funcional. É vendido por meios oficiais como a loja do Arduino, no entanto não é necessário adquirir um *hardware* oficial para criar projetos, já que é facilmente encontrável versões não-oficiais completamente funcionais e com preços ainda mais acessíveis.

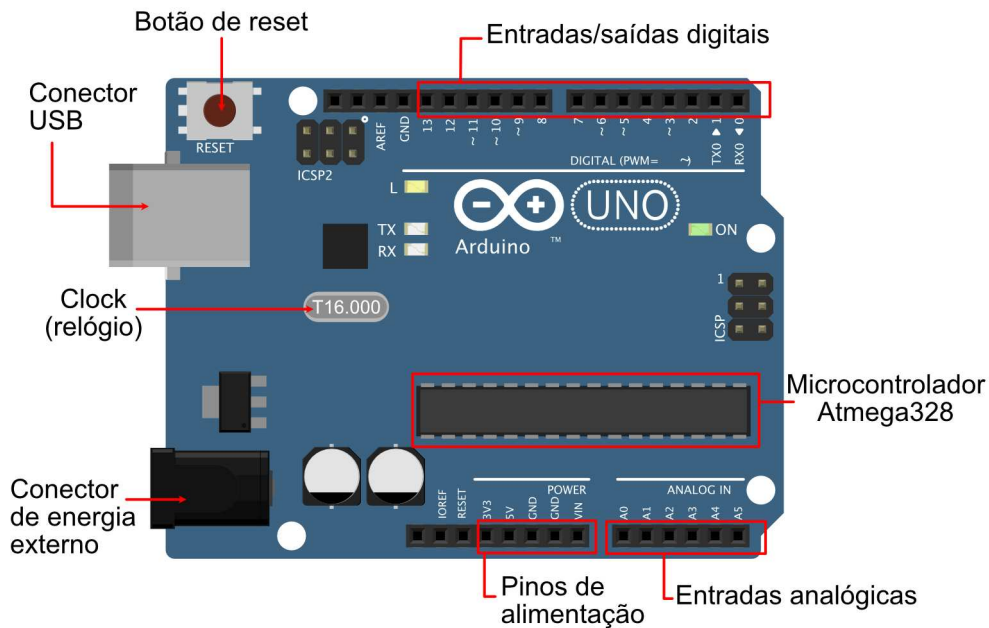
O Arduino não é em si um microcontrolador, no entanto ele contém um microcontrolador Atmel como seu centro de operações. Como o Arduino foi feito pensando em facilidade de uso e didática, o seu *hardware* nada mais consiste do que uma placa com vários componentes que servem para auxiliar o uso do microcontrolador, permitindo assim uma facilitação para prototipagem. O microcontrolador embutido na sua placa a depender do modelo pode ser facilmente extraído, ou fixo.

Para programar o Arduino assim como qualquer dispositivo programável é necessária a utilização de uma *Integrated Development Environment* (IDE) que pode ser encontrada no site oficial, sendo possível programar em linguagem C ou C++ e o *upload* das instruções é dado via *Universal Serial Bus* (USB) através do computador. O site oficial também contém uma série de tutoriais e documentação das bibliotecas utilizadas para programação de componentes do Arduino e há vários exemplos de usos destes componentes e exemplos de como programa-los (OLIVEIRA; ZANETTI, 2015). É possível encontrar no mercado muitas variantes das placas Arduinos para diferentes tipos de projetos, no entanto o mais comumente utilizado é o Arduino

Uno que é o sucessor do *Duemilanove* que foi o utilizado para esta pesquisa.

O Arduino e sensores agora com o seu preço acessível facilitaram a popularização do estudo de conceitos de física, robótica, automação e uma série de outras áreas. (MCROBERTS, 2018) A simplicidade do Arduino facilita o entendimento de conceitos abstratos e atrelado ao seu uso de sensores é possível extrair dados sobre o ambiente como temperatura, umidade, luminosidade, pressão atmosférica, gás, som, magnetismo, infravermelho, entre outros. Tais sensores tem tamanho compacto e são de fácil uso, pois o Arduino tem como maior princípio a prototipação de sistemas, que podem posteriormente receber uma miniaturização dos componentes em um nível maior de sofisticação.

Figura 1 – Esquema básico do Arduino



Fonte: Adaptada de (PHILIPPHENKEL, 2018)

O Arduino UNO Figura 1 contém

- **Conector USB:** É por onde é feito o envio dos códigos, também pode transmitir dados em tempo real para o computador. Serve como fonte de alimentação do Arduino. O tipo de cabo utilizado é USB-A em uma ponta e USB-B da outra, que é um cabo mais comumente visto em impressoras.
- **Botão de reset:** Reinicia o algoritmo em execução
- **Conector de energia externa:** É de uso opcional feita através do conector *Jack*. Com

positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites 6V. a 20V., porém se alimentada com uma tensão abaixo de 7V., a tensão de funcionamento da placa, que no Arduino Uno é 5V, pode ficar instável e quando alimentada com tensão acima de 12V, o regulador de tensão da placa pode sobreaquecer e danificar a placa. Dessa forma, é recomendado para tensões de fonte externa valores de 7V. a 12V. (SOUZA, 2013)

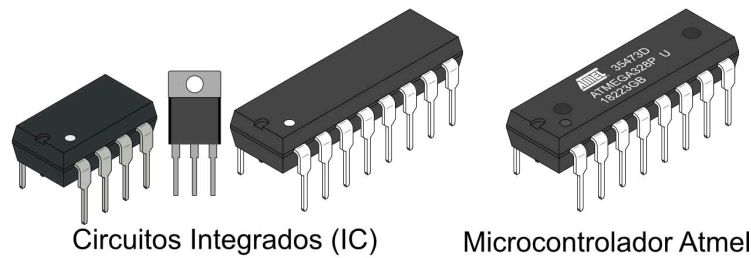
- **Entradas/saídas digitais:** Operam em 5V, onde cada pino pode fornecer ou receber uma corrente máxima de 40 mA. Emitem e recebem energia de forma binária sendo um valor abaixo de 1.5V considerado LOW e acima disso como HIGH. Em placas de 3.3V o valor LOW é medido em valores abaixo de 1V(ARDUINO, 2020).
- **Entradas analógicas:** Muito similares as digitais, porém leem valores entre 0 e 5V.
- **Conectores de energia:** Fornece energia para sensores e equipamentos externos com saídas 5V ou 3.3V, também pinos de referência terra (Ground). Há também pino para reset, entrada de energia para baterias externas (VIN) e o pino IOREF que serve para referência de tensão onde o microcontrolador está operando.
- **Clock:** Utiliza um cristal oscilador que determina a sincronia de tempo.

## 2.2 MICROCONTROLADORES

Microcontrolador (as vezes chamado de *Microcontroller Unit* (MCU)) é um pequeno computador auto contido em um circuito integrado podendo ser utilizado como um sistema embarcado (GUDINO, 2018). Este contém uma Unidade Central de Processamento(*Central Process Unit* (CPU)), memória *Random-access memory* (RAM), comunicação serial, periféricos programáveis de entrada e saída, e em muitos casos pode conter uma memória do tipo *Electrical Erasable Programmable Read-Only Memory* (EEPROM).

Os circuitos integrados (*Integrated Circuit* (IC)) Figura 2 são componentes em um invólucro e que exercem funções determinadas pelas disposição dos componentes no seu interior no momento que são fabricados. No entanto, os microcontroladores a princípio não tem função alguma em específico, pois eles podem ser programados para realizar diferentes tipos de funções a depender de como forem configurados.

Figura 2 – Circuitos Integrados e Microcontroladores



Fonte: O autor

### 2.3 MEMÓRIAS NÃO VOLÁTEIS

O Arduino contém uma memória não volátil EEPROM, ou seja, que não perde os dados assim que desliga, do tipo EEPROM de 1KB para eventual armazenamento de dados. Mas antes, para melhor compreensão do funcionamento desse tipo de memória é necessário uma explicação resumida sobre alguns tipos de memórias não voláteis.

- **Read-Only Memory (ROM):** memória que pode ser escrita apenas uma vez, geralmente em seu processo de fabricação. Após a primeira escrita não é mais permitido alterar ou apagar os dados, só é permitido a leitura.
- **Electrical Erasable Programmable Read-Only Memory (EPROM):** Muito similar a ROM, a diferença é a possibilidade de resetar o seu conteúdo por meio de um aparelho que emite uma forte luz ultravioleta em uma janela do chip habilitando desta forma uma reescrita.
- **EEPROM:** Assim como a EPROM que pode resetar os seus dados para reescrita, a EEPROM faz isso de forma eletrônica sem a necessidade de algo externo como a luz ultravioleta. Embora o seu nome Read-Only, essa classificação é incorreta pois pode ser reescrita diversas vezes.
- **Flash:** As memórias do tipo Flash podem ser consideradas como uma evolução das memórias EEPROM, e tem como principal características processos de gravação e regravação muito mais rápido, sendo também mais duráveis, podendo guardar volumes elevados de dados. É muito comum em cartões de memória flash (*Secure Digital* - Cartão de memória (SD)), Pendrives, SSDs e outros (ALECRIM, 2011).

Embora a memória EEPROM possa ser reescrita diversas vezes, a sua vida útil é limitada. Para se trabalhar com Arduino é possível também o uso de módulos para cartões de

memória a depender da necessidade de dados a serem armazenados.

## 2.4 SENSORES DE TEMPERATURA

Existem diversos tipos de sensores de temperatura, sendo alguns não feitos especificamente para temperatura, mas que podem ser usadas para tal. Sensores de temperatura podem atender a diferentes amplitudes de temperatura, diferentes acurácias e variar entre sensores analógicos e digitais.

- **DHT11:** É um sensor de temperatura e umidade de baixo custo, não muito acurado, perdendo sutis variancias de temperatura. Pode fazer medida de temperatura a cada segundo. Devido sua mensuração de temperatura e umidade, também serve para verificar a sensação térmica e ponto de orvalho. Lê de 0 à 50°C.
- **DHT22:** É uma versão melhorada do DHT11 e com maior acurácia, no entanto com preço mais elevado. Detecta temperaturas em uma amplitude maior. Pode fazer medida de temperatura a cada 2 segundos. Lê de -40 à 80°C.
- **LM35DZ:** Mede somente temperatura e de forma analógica. É bastante simples de ser utilizado e com uma relativa boa precisão, no entanto pode haver algumas oscilações entre medidas. É sensível a perturbações e interferências. -55 à 150°C.
- **DS18B20:** Sensor digital de temperatura. Necessita uso de um resistor 4k7 para alimentação e eliminação de interferência no pino de dados. Para programação precisa das bibliotecas OneWire e DallasTemperature. Tem a vantagem de prover uma comunicação de dados com o Arduino por apenas um fio ocupando menos portas em projetos com múltiplos destes sensores. Comparado com o LM32 é mais preciso e confiável, como pode ser visto em (SCHNEIDER, 2020). Lê de -55 à 150°C.

Para uma melhor descrição e visualização de gráficos e tabelas expondo resultados comparativos entre estes sensores ver (RANDOMNERDTUTORIALS.COM, 2019).

### 3 PROCESSOS DE PROPAGAÇÃO DE CALOR

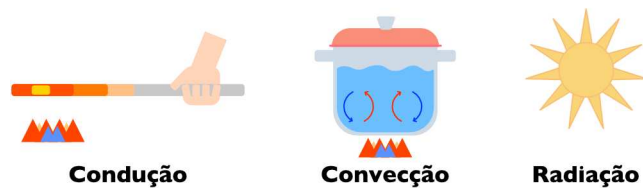
O calor pode ser descrito como uma energia em trânsito com um sentido único, indo do corpo de maior temperatura para o de menor temperatura até chegar em um estado de equilíbrio térmico (BERGMAN; LAVINE; QUEIROZ, 2000).

#### 3.1 TIPOS DE PROPAGAÇÃO DE CALOR

Existem três tipos de transmissão de calor:

- **Condução:** Ocorre quando corpos com temperaturas diferentes são colocados em contato e a energia é conduzida átomo a átomo do corpo mais quente para o corpo mais frio.
- **Convecção:** quando há a troca de calor pela movimentação/deslocamento ocorrido em meio líquido, ou gasoso. Exemplo, a movimentação de massas de ar frio e ar quente.
- **Radiação** Propagação do calor através de ondas eletromagnéticas, por exemplo a luz. Ocorre mesmo no vácuo (YOUNG; FREEDMAN; FÍSICA, 2008).

Figura 3 – Tipos de troca de calor



Fonte: O autor

#### 3.2 IRRADIAÇÃO OU RADIAÇÃO

Este é o tipo de propagação a ser estudado neste projeto. Podemos dizer que a irradiação térmica é o processo mais importante, pois sem ela seria praticamente impossível haver vida na Terra, já que é por irradiação que o calor liberado pelo Sol chega até a Terra (YOUNG; FREEDMAN; FÍSICA, 2008).

Todos os corpos emitem radiação, ou seja, emitem ondas eletromagnéticas, cujas

características e intensidade dependem do material de que é feito o corpo e de sua temperatura. Portanto, o processo de emissão de ondas eletromagnéticas é chamado de irradiação(SILVA, 2020).

### **3.2.1 Absorção, Reflexão e Transmissão em Superfícies Reais**

A Absorção, Reflexão e Transmissão são processos fundamentais do processo de irradiação. Elas estão associadas a outro conceito chamado de emissividade, que é a razão definida entre a radiação emitida pela superfície e a radiação emitida por um corpo negro à mesma temperatura. Já um corpo negro é um absorvedor e emissor perfeito, por exemplo o Sol. (BERGMAN; LAVINE; QUEIROZ, 2000)

#### **3.2.1.1 Absortividade/Absorbância**

Absorbância, é a capacidade intrínseca dos materiais em absorver radiações por uma superfície em uma frequência específica. Usualmente, tal propriedade é empregada na análise de soluções em química analítica.

#### **3.2.1.2 Reflectância/Refletividade**

Reflectância é a proporção entre o fluxo de radiação eletromagnética incidente numa superfície e o fluxo que é refletido. Além de depender da direção da radiação incidente, ela também depende da direção da radiação refletida. Frequentemente a reflectância é apresentada sob a forma de porcentagem.

#### **3.2.1.3 Transmissividade/Transmitância**

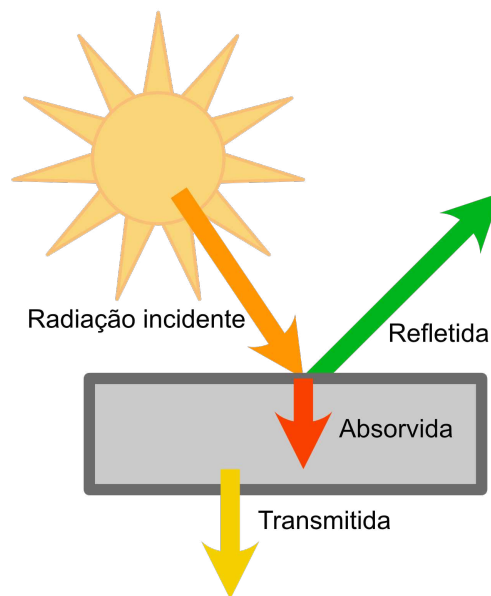
É a fração da radiação incidente sobre um sistema que consegue atravessá-lo. Inclui tanto a transferência de radiação que não sofreu interação com o sistema quanto a produção de radiação difusa para o hemisfério de destino do fluxo incidente. É numericamente igual à soma da transmitância direta com a transmitância difusa (também chamada de especular)(BERGMAN; LAVINE; QUEIROZ, 2000).

### 3.2.2 Frações de energia

Quando uma energia radiante chega a um corpo ela tem 3 destinos diferentes, como se pode observar na Figura 4:

- Parte da energia é absorvida pelo corpo ( $Q_a$ )
- Parte da energia é refletida pelo corpo ( $Q_r$ )
- Parte da energia é transmitida pelo corpo ( $Q_t$ )

Figura 4 – Transferência de calor por radiação



Fonte: O autor

Portanto a quantidade total de energia radiante no corpo ( $Q_i$ ) é igual a soma das energias absorvidas, refletida e transmitida.

$$Q_i = Q_a + Q_r + Q_t$$

A soma da absorvidade ( $a$ ), refletividade( $r$ ) e transmissão é igual a 1

$$a + r + t = 1$$

Se considerarmos que não há transmissividade, então:

$$a + r = 1$$

Em teoria temos o que chamamos por *Corpo Negro* é a de um corpo que absorve

toda a energia que recebe, ou seja  $a=1$  e  $r=0$ . Já o seu inverso, o que chamamos de um *Espelho Ideal*, tem o máximo de refletividade e nenhuma absorção de energia, portanto  $r=1$  e  $a=0$ .

Todo o corpo que é um bom absorvedor de energia também é um bom emissor de energia e do contrário também é correto afirmar.

### 3.3 INFLUÊNCIA DA COR NA ABSORÇÃO DE CALOR

A cor de um objeto interfere na absorção de energia. Artigos e matérias apontam que a absorção pode afetar em setores econômicos e o mal uso da coloração de objetos pode acarretar diversos problemas. A cor de um carro pode afetar na temperatura de um carro e fazer com que o condutor precise utilizar mais o ar condicionado, juntamente com uma ligeiramente maior evaporação e consumo de combustível.

Na edição da revista QUATRO RODAS de fevereiro de 1987, foi publicado um teste que mostrou isso na prática. Dois Chevrolet Monza iguais – com exceção da cor – foram deixados expostos ao sol durante duas horas, com temperatura externa média de 31 °C. Resultado: o interior do carro preto atingiu 64 °C e o do branco, 55 °C (RODAS, 2017). Ou seja, uma diferença de 11 graus em curto espaço de tempo.

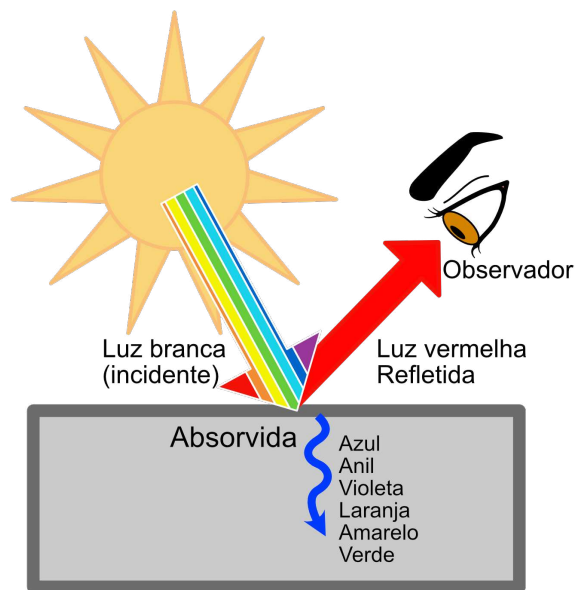
É muito comum encontrar diversos artigos sobre o assunto buscando sobre engenharia e arquitetura, esse conhecimento é muito aplicado em projetos de casa, pois a cor da pintura principalmente do lado externo de uma casa ou edifício de uma casa pode afetar diretamente no conforto, como pode ser visto em (SYUHADA; MAULANA, 2018).

#### 3.3.1 Cor e Calor

A cor se deve à reflexão e à absorção seletivas da porção visível da irradiação oriunda do Sol ou de uma fonte de luz artificial. Uma camisa é “vermelha” porque contém um pigmento que absorve preferencialmente os componentes azul, verde e amarelo da luz incidente. Assim, as contribuições relativas desses componentes na luz refletida, que é vista pelo olho humano, são diminuídas e o componente vermelho é dominante. De modo semelhante, uma folha é “verde” porque suas células contêm clorofila, um pigmento que apresenta uma forte absorção do azul e do vermelho, e uma reflexão preferencial do verde. Uma superfície aparece “preta” quando absorve toda a radiação visível incidente, e “branca” quando reflete essa radiação. Entretanto,

devemos ter cuidado como interpretamos tais efeitos visuais. Para uma dada irradiação, a “cor” de uma superfície pode não indicar sua capacidade global como um absorvedor ou um refletor, uma vez que uma grande parte da irradiação pode estar na região Infra Vermelho (IV). Por exemplo, uma superfície “branca” como a neve é altamente refletiva à radiação visível, porém absorve fortemente radiação IV, aproximando desse modo o comportamento de um corpo negro em maiores comprimentos de onda (BERGMAN; LAVINE; QUEIROZ, 2000). Um exemplo deste evento poderemos ver na figura Figura 5

Figura 5 – Reflexão da cor



Fonte: O autor

## 4 DISPOSITIVO AUTÔNOMO PARA AQUISIÇÃO DE DADOS

Este capítulo contém uma série de informações sobre os códigos fonte, bibliotecas utilizadas, e eventuais problemas e desafios que costumam acontecer no desenvolvimento de algoritmos para as plataformas especificadas. Aqui também é mostrada a organização do projeto em termos de *software* e em *hardware*.

### 4.1 ORGANIZAÇÃO DOS COMPONENTES ARDUINO

Em conjunto com o Arduino sensores e outros componentes foram utilizados para a criação da interface como é vista na Figura 6. A lista de componentes são:

- Arduino Uno
- 4 botões/*switches*
- 4 sensores de temperatura DS18B20
- 1 resistor 4K7 (4.7K)
- *Display* LCD 16x2
- Módulo I2C para LCD
- Módulo de cartão de memória SD

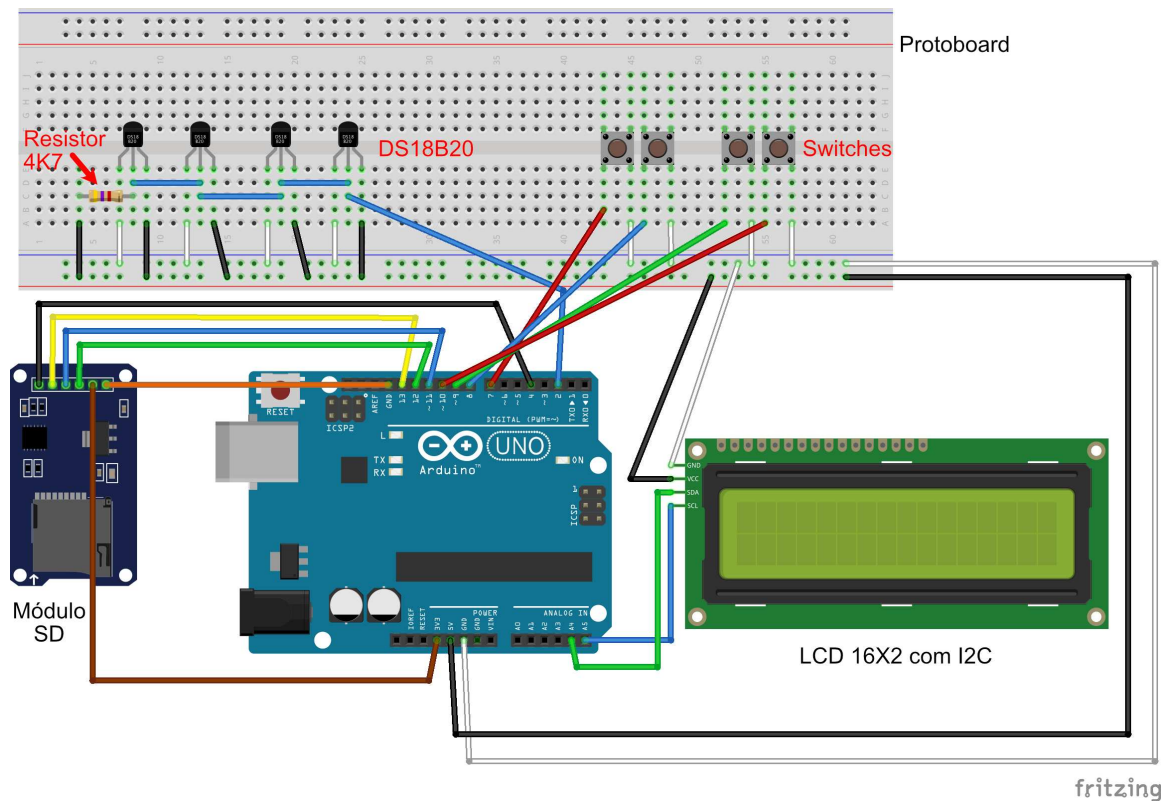
#### 4.1.1 LCD 16x2

O princípio por trás do *Liquid Crystal Display* (LCD) é a de energizar uma série de cristais para aparecer tanto opaco como transparente contra uma luz de fundo. Há dois tipos de *display* LCD, o *display* alfanumérico e o *display* gráfico, no entanto para este projeto foi usado o *display* alfanumérico.

O *display* para Arduino é composto de pequenas matrizes de pontos. Cada matriz contém 5 colunas e 8 linhas que formam letras, números ou símbolos no *display*. O *display* geralmente contém 2 linhas de 16 caracteres cada.

Para maiores informações consultar Datasheet Hitachi HD44780U (LCD-II)(HITACHI, 1998).

Figura 6 – Modelo do projeto



Fonte: O autor

#### 4.1.2 Módulo I2C para LCD

Um dos problemas do *display* LCD para Arduino é a quantidade de fios necessários para seu funcionamento, o que pode ocupar 16 portas do Arduino. O módulo I2C serve como intermediário e reduz este número para apenas 4, além de ter incluído um potenciômetro que é fundamental servindo para ajuste de brilho do *display*. O único contra é que para usar o módulo é necessário instalar bibliotecas para seu funcionamento via software.

#### 4.1.3 Sensor de temperatura DS18B20

O sensor utilizado para o projeto foi o DS18B20 feito pela *Dallas* que tem uma maior acurácia em relação aos outros modelos de sensores encontrados para o experimento. Para seu uso é necessária a instalação e configuração de duas bibliotecas: *OneWire* e *DallasTemperature*; também é ideal o uso de um resistor 4k7 na entrada de dados para uma melhor acurácia de dados. Os sensores DS18B20 enviam a temperatura como um fluxo de dados seriais, transmitidos por uma única linha de transmissão – isso explica por que o protocolo é chamado de *1-Wire* (1

Fio). Cada sensor também tem um número serial único, permitindo que você consulte sensores diferentes utilizando seu número de ID. Isso faz com o DS18B20 seja popular em projetos que envolvem o Arduino, uma vez que podemos encadear uma quantidade praticamente ilimitada de sensores, em apenas um único pino do Arduino. O intervalo de temperatura verificado é de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  (MCROBERTS, 2018).

#### 4.1.4 Módulo de cartão SD

Este projeto foi iniciado com intuito de utilizar a própria memória EEPROM do Arduino, porém essa decisão foi alterada pois limitaria em muito a quantidade de dados para 1KB e estratégias de economia de dados aumentariam consideravelmente a complexidade do código para pouca melhora na quantidade de armazenamento de dados. No mercado existem módulos de memória EEPROM externa assim como módulos de cartões SD. Optou-se por utilizar o módulo SD externo pois seu custo é mais baixo e é de mais fácil utilização, pois suas bibliotecas de programação são mais simples.

## 4.2 PROGRAMAÇÃO ARDUINO

A programação utilizada no projeto foi feita em linguagem C, o ambiente de desenvolvimento contém bibliotecas próprias especificamente para programação do *hardware*. O Arduino foi elaborado para pequenos projetos, mas ao se deparar com a necessidade de criação de projetos maiores algumas armadilhas específicas do ambiente podem aparecer. Alguns conceitos são utilizados para contornar problemas comuns que programadores experientes costumam fazer e iniciantes podem ter dificuldades de descobrir sozinhos. Tais conceitos podem viabilizar coisas como a simulação do processamento simultâneo de múltiplas funções, contagem do tempo e contornar *overflow* de dados.

### 4.2.1 Mensuração de tempo com função *millis()*

O Arduino tem um relógio em *hardware*, que é um circuito elétrico que cria um sinal, como um pulso voltaico, em uma frequência persistente. Em termos de programação na biblioteca padrão do Arduino há uma função própria chamada *millis()*, esta função contabiliza o tempo em milissegundos que ocorreu desde o quando o Arduino foi iniciado e entrado em estado de funcionamento. No entanto há algumas coisas a considerar sobre essa função, pois o

seu contador reinicia assim que muito tempo foi passado desde o início do microcontrolador, ou seja ocorre um overflow do contador. O tempo para que esse contador seja resetado é bastante extenso, funcionando continuamente em uma estimativa de 49 dias. Normalmente projetos de teste com Arduino não costumam durar por tempos tão extensos, mas é necessário se ter isso em mente a depender do tipo de aplicação a ser feita.

Outra informação a tomar como observação é que o número representado pela função `millis()` costuma crescer muito rápido, isso significa que não é uma boa ideia armazenar esse tipo de dado em um inteiro, já que o indicado é armazenar em uma variável do tipo *unsigned long* que vai de 0 até 4,294,967,295. Para se fazer cálculos e operações matemáticas com variáveis do tipo *unsigned long*, deve se prestar atenção para que as outras variáveis que estão na mesma operação sejam também do tipo *unsigned long*, ou erros lógicos irão acontecer.

#### **4.2.2 Pausas no Arduino: função `delay()` e uso de `millis()` para pausas**

Normalmente iniciantes costumam utilizar-se da função `delay()` para criar algum tipo de espera na execução de um código, no entanto isso faz com que o microcontrolador entre em um estado de congelamento do código evitando que todos os outros processos sejam conferidos até que o tempo do delay (atraso) seja encerrado. Uma alternativa eficaz é o uso da função `millis()` de modo estratégico para criação de espera evitando bloqueio de outras funções no código.

Para tal, é necessário o uso da função similarmente a o uso de um cronômetro, onde se contabiliza o início da operação e espera a contagem do "cronômetro" chegar até uma determinada faixa de tempo. O uso da função `millis()` como alternativa ao `delay()` pode simular um tipo de processamento *multithread* ou multiprocessamento na programação, o único contra é que a estratégia diminui a clareza do código. Não obstante, é fundamental que seja introduzido o conceito a iniciantes no estudo do Arduino, já que possibilita que por exemplo, o apertar de um botão não interfira na captura de um sensor como o de temperatura. Existem bibliotecas que também servem para ajudar este processo como a *MsTimer2* para Arduino, mas para simplificação não foi utilizado neste projeto.

O exemplo abaixo mostra o uso da função `millis()` fazendo um papel de espera similar a função `delay()`, porém evitando o bloqueio de execução da linha 13 a cada *loop*, o que não seria possível com a função `delay()`.

### Exemplo de código não bloqueavel com millis()

```

1 int tempo_espera = 5000; // Equivalente a 5 segundos
2 unsigned long tempo_agora = 0;
3
4 void setup() {
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     if(millis() >= tempo_agora + tempo_espera){
10         tempo_agora += tempo_espera;
11         Serial.println("Ola, eu imprimo a cada 5 segundos");
12     }
13     Serial.println("Eu imprimo o tempo inteiro");
14 }

```

### 4.2.3 Evitando *overflow* da memória SRAM

O Arduino UNO contém três locais de memória no seu microcontrolador.

- **Memória Flash (espaço do programa):** Onde a sketch do Arduino é armazenado, ou seja o código fonte.
- **SRAM(memória de acesso aleatório estático):** Onde o sketch cria e manipula variáveis em tempo de execução
- **EEPROM:** Espaço de memória que programadores podem fazer uso para armazenamento de dados.

No chip do Arduino UNO (ATmega328P) temos: Flash 32KB, SRAM 2KB e EEPROM 1KB. Já chips como ATmega2560 e Mega2560 tem maiores espaços de memórias (ARDUINO, 2020a).

Como boa prática é importante economizar no uso de memória SRAM, que é essencial para a criação de variáveis em tempo de execução. Um mal gerenciamento de variáveis pode causar *overflow* da SRAM ocasionando em falhas inesperadas de funcionamento enquanto o Arduino estiver em execução. A IDE oficial do Arduino dá uma previsão aproximada em porcentagem do quanto de SRAM que será utilizado a cada vez que o código é submetido dando

aviso de risco caso a porcentagem esteja perto de 80%.

É muito comum estourar a SRAM quando se insere no código muitas cadeias de caracteres(*strings*), o que costuma ocorrer em códigos com muita impressão de dados no console da IDE Arduino, em interfaces como LCD ou cartão de memória, o que acontece frequentemente no código criado para este projeto. Todas as *strings* do código por padrão são armazenadas na SRAM consumindo o pouco espaço de memória disponível, o que pode parecer algo inconcebível e facilmente causar frustrações, pois não é algo óbvio de descobrir.

#### Exemplo de string na impressão em console

```
1 Serial.println("Esta string ocupa 35 bytes na SRAM");
```

Uma das maneiras mais fáceis de contornar o problema e salvar espaço na SRAM é em código fazer a transferência das cadeias de caracteres que serão fixas para a memória flash. A partir da versão 1.0 da IDE do Aduino foi introduzida a macro *F()* para armazenar dados na memória flash ao invés da SRAM.

#### Exemplo de uso da macro F()

```
1 Serial.println(F("Esta string sera armazenada na memoria flash"));
```

Para mais explicações sobre este problema e um maior explicação sobre a memória RAM do Arduino vide (CAMPOS, 2015).

#### 4.2.4 Bibliotecas utilizadas

Para a utilização de módulos Arduino é necessário encontrar e instalar bibliotecas específicas para cada módulo. A maioria das bibliotecas é facilmente instalável na própria IDE do Arduino, no entanto algumas devem ser encontradas na internet em sites de fabricantes ou comunidades.

- **LiquidCrystal\_I2C.h:** Biblioteca para uso de painel LCD em conjunto com módulo I2C.
- **OneWire.h e DallasTemperature.h:** Bibliotecas exclusivas para o sensor de temperatura DS18B20.
- **SD.h:** Biblioteca para uso de módulo de cartão SD

#### 4.2.5 Algoritmo do projeto para Arduino

O algoritmo produzido para o Arduino é concentrado em seu menu. Ao ligar o dispositivo a coleta não é iniciada, apenas ao selecionar a opção *coletar dados*. Para fazer a parada dos dados é necessário desplugar o Arduino da tomada ou apertar o botão de reset, presente fisicamente na placa.

As diferentes opções do menu são:

- **Temp. Atual:** Mostra a temperatura atual de cada um dos 4 sensores em tempo real.
- **Coletar Dados:** Inicia o processo de captura de dados para o cartão SD, para parar a captura é necessário usar o botão reset, ou desligar o aparelho.
- **Ler dados:** Envia todos os dados de captura que estão armazenados no cartão SD para um computador conectado.
- **Intervalo:** Altera o tempo em minutos a cada captura de dados. Por padrão foi colocado como a cada 1 minuto.
- **Num. Sensores:** Altera a quantidade de sensores ativos para as capturas.
- **Deletar captura:** Remove do cartão SD o arquivo com todas as últimas coletas feitas. Essencial fazer isso antes de uma nova coleta.

#### 4.3 PROGRAMAÇÃO DE INTERFACE PYTHON

Como o Arduino faz a coleta de dados, é necessário então o uso dos dados coletados para algum propósito, no caso mais comum a de geração de gráficos que mostre as mudanças capturadas pelos sensores ao longo do tempo. Para isso é necessário um segundo software que possa gerar gráficos e salva-los, de preferência tendo uma interface gráfica.

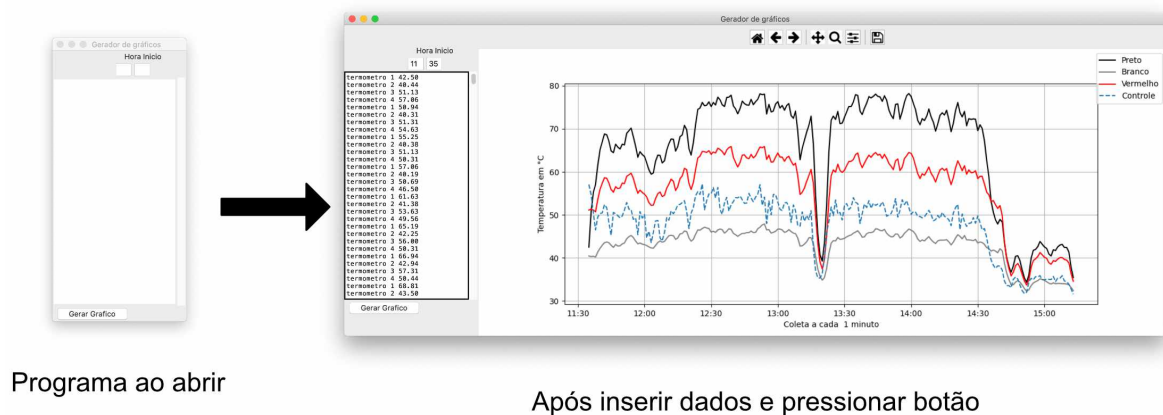
Para a produção deste software para o projeto, foi escolhido a linguagem de programação *Python* versão 3. A escolha do *Python* se deve a sua clareza e facilidade de compreensão por iniciantes e sua capacidade de uso em múltiplas plataformas e sistemas operacionais como Linux, Windows e MacOs (FOUNDATION, 2018). Para a produção da interface gráfica (GUI) que gera os gráficos finais foi utilizada a biblioteca *Tkinter* que já vem incluída nos pacotes mais recentes do Python 3. Foi evitado ao máximo a utilização de ferramentas proprietárias, ou formatos de arquivos proprietários e de difícil acesso visando fácil reprodução e alterações por outras pessoas em trabalhos futuros.

Em conjunto com o *Python* e *Tkinter* para a produção da interface gráfica é necessário a configuração e instalação da biblioteca *Matplotlib*, muito utilizada para plotagem de gráficos. o *Matplotlib* contém ferramentas para exportação de imagens em formatos png, jpeg, svg, além de proporcionar uma análise interativa, o que facilita o trabalho para produções textuais com gravuras.

O Programa produzido para este projeto tem um funcionamento bastante simples. Como podemos observar na Figura 7 apenas deve-se colocar todas as medições de temperatura armazenadas do Arduino na caixa de texto especificada, juntamente deve-se inserir o horário no campo de horário e em seguida apertar o botão para gerar os gráficos.

Com isso é gerado um gráfico e é possível interagir com o cursor do *mouse* observando com maior precisão os valores, além de poder utilizar das ferramentas de configuração, zoom e é possível salvar o gráfico como imagem ou documento, para uso posterior.

Figura 7 – Programa *Python*



Fonte: O autor

## 5 EXPERIMENTO

Este capítulo mostra os passos realizados para fazer uso do módulo de coleta para observação do evento de radiação.

### 5.1 LATAS E PINTURA

Desde o início foi imaginado que seria essencial a pintura de latas na cor preto e branco, mas para uma maior diversidade também foram pintadas uma lata vermelha, azul e amarela, no total de 5 latas. Para a pintura foi mais do que suficiente um pote de 37ml de tinta acrílica de artesanato para cada lata, estas são facilmente encontradas em livrarias. A tinta escolhida, de acordo com a embalagem, são resistentes ao tempo e intempéries. Como alternativa foi cogitado o uso de latas de tinta *spray*, mas tornaria o preço mais elevado e o objetivo era de manter o menor custo possível para a eventual replicação do experimento. A fim de comparação de preço os pequenos potes de tinta custaram em torno 5 a 6 reais cada no período do experimento, enquanto uma única lata de tinta *spray* custava acima de 20 reais. Para a pintura das latas foram usadas tintas brilhantes nas cores vermelho, azul, amarelo e preto, no entanto para a cor branca foi utilizada tinta fosca por falta de tinta brilhante na casa de comércio visitado. Em um outro experimento de pesquisa mais refinado e com instrumentos profissionais (FRAGATA; SENS; SEBRÃO, 2015) foi notado que o brilho das tintas altera minimamente no experimento, mas de forma desprezível, não sendo assim algo digno de preocupação.

O que foi possível observar na fase da pintura é que as tintas brilhantes coloridas tinham uma certa transparência, sendo necessário complementar a pintura com uma série de camadas extras de modo a não aparecer a pintura original da lata. Foi necessária em média 7 camadas de tinta por lata, sendo a preta mais fácil de atingir uma maior opacidade, exigindo de 3 a 4 camadas. Também se nota que a tinta branca por ser de característica fosca tem uma melhor cobertura, sem muita transparência e foi necessária também poucas camadas assim como a lata preta. Em um posterior replicação do experimento seria mais aconselhável o uso apenas de tintas foscas para facilitar a pintura, também seria mais aconselhável utilizar outro tipo de lata que não a de milho, já que esta tem uma pintura própria feita pela fabricante que dificulta a cobertura

Figura 8 – Tintas acrílicas de artesanato



Fonte: O autor

total pela tinta. Uma alternativa mais aconselhada seria a de latas de leite condensado que não tem pintura e sim um rótulo de papel facilmente removível.

Figura 9 – Pintura de lata em fase inicial



Fonte: O autor

## 5.2 CAPTURA DE DADOS

Os experimentos foram realizados seguindo o seguinte método:

1. Os componentes do coletor Arduino são posicionados em cima de uma placa Poliestireno Expandido (EPS) (Poliestireno) de construção civil para minimizar a interferência do calor do solo.
2. São feitos os ajustes de cada um dos sensores dentro das respectivas latas a serem analisadas. O posicionamento deve-se observar que as latas não façam sombra umas nas outras.

3. Um dos sensores se põe fora de qualquer lata para servir como um termômetro de controle. Este geralmente colocado em uma posição afastada de sombras.
4. O Arduino é conectado a uma tomada de energia, veja na Figura 10.

Figura 10 – Experimento conectado a tomada

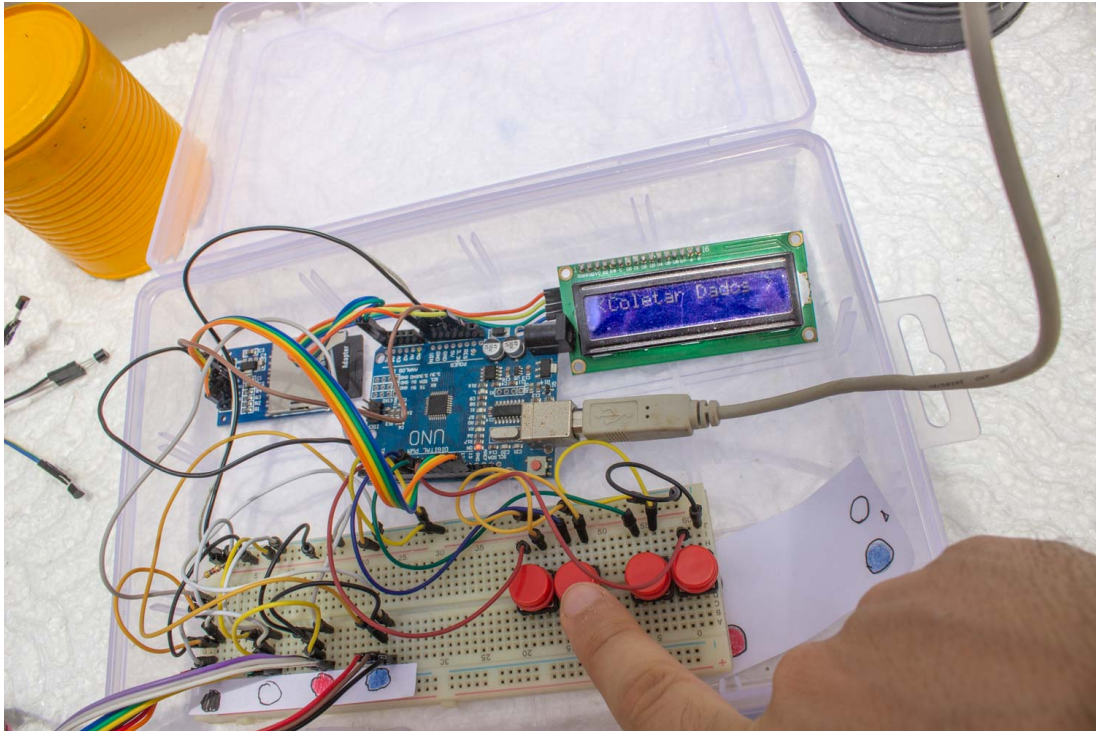


Fonte: O autor

5. É feita a configuração do Arduino pela interface e iniciado a coleta, veja na Figura 11.
6. É coberta a caixa dos componentes Arduino com material reflexivo como um azulejo para evitar superaquecimento desgaste dos componentes, veja na Figura 12.
7. É anotada a data e horário que foi iniciado a coleta.
8. Ao final do horário da coleta desliga-se o Arduino da tomada.
9. As latas e componentes são removidos e coletados para o próximo dia de coleta
10. O Arduino é levado até um computador para a análise dos dados e criação de gráficos. Os dados são transferidos via USB pelo console da IDE do Arduino.

Pelo que foi observado a iluminação no ambiente onde foi realizado o experimento a incidência direta da luz do sol acontecia por volta das 11:20h até às 14:30h. Portanto a grande maioria das coletas foram executadas entre estes periodos.

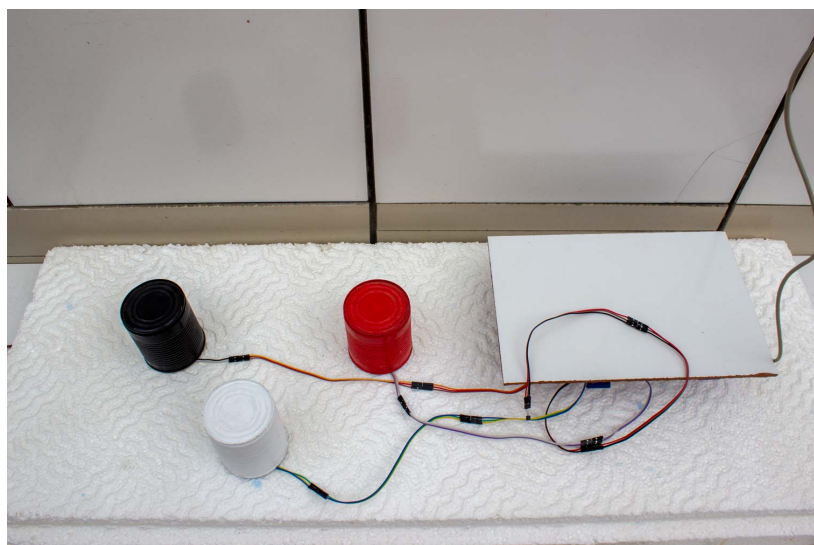
Figura 11 – Iniciação da coleta



Fonte: O autor

Com o intuito de observar o aquecimento interno da lata foi inserido de forma livre o sensor DS18B20 dentro de cada uma das latas. Fazendo uma análise do que aconteceria com a lata e o comportamento esperado seria de que a lata recebe a radiação incidente do sol, parte da radiação seria refletida a depender das características da cor da pintura. O calor transmitido

Figura 12 – Organização do experimento



Fonte: O autor

para dentro da lata não é uma radiação luminosa assim como a recebida, mas sim tanto um calor transmitido por convecção do ar, juntamente com radiação provocada por ondas mais longas como infravermelho, é um evento similar ao que podemos ver no fenômeno de efeito estufa.

O Arduino foi conectado a uma tomada como alimentação de energia, mas o mesmo experimento poderia ser feito usando um módulo de bateria, no entanto deve-se considerar que não é indicado o uso de baterias em ambientes de alta temperatura.

## 6 RESULTADOS

Na primeira coleta realizada no dia 29/10/2020 após um dia de chuvas intensas, o clima era nublado como observado na Figura 13 e ainda havia risco de chuva, porém não ocorreu. Assim como a maioria dos experimentos realizados, para este dia foram utilizados 3 sensores nas latas preta, branca e vermelha, e um quarto sensor serviu como controle, ficando exposto ao sol diretamente. Observando o resultado da Figura 14, as temperaturas tiveram poucas flutuações e apenas três picos principais que podem ser o resultado de uma desanuviação.

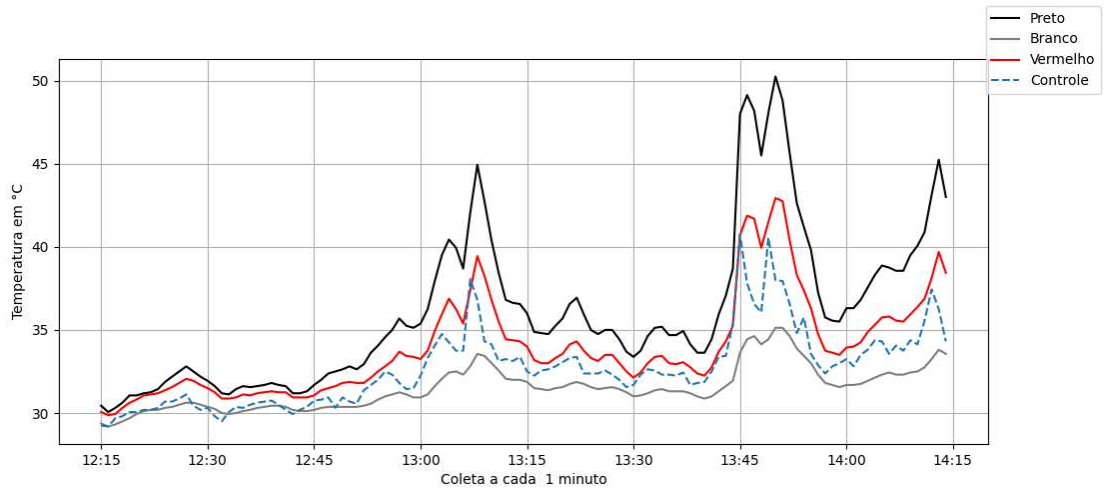
Figura 13 – Dia nublado



Fonte: O autor

Na segunda coleta, dia 30/10/2020, estava parcialmente nublado, com menos nuvens do que no dia anterior. No gráfico da Figura 16 é perceptível um aumento da temperatura chegando a temperaturas próximas a 80°C, enquanto, no dia anterior teve seu máximo pouco acima dos 50°C. Há diversos picos de temperatura em decorrência da quantidade de nuvens

Figura 14 – Gráfico de coleta em dia nublado



Fonte: O autor

que embora não tão carregadas como na coleta anterior, interferiam na emissão direta de raios solares.

Já no quarto dia de coleta, dia 8/11/2020 onde o clima estava ensolarado e com raras nuvens como observado na Figura 17, foi para maior diversidade de tipos de coletas utilizado as latas coloridas azul, vermelho e amarelo. Era esperado que a lata azul atingisse temperaturas maiores que a vermelha em seguida a amarela. Observando os gráficos da Figura 18, o resultado esperado foi exatamente o que foi obtido.

Na quinta coleta, dia 09/11/2020 era também um dia ensolarado similar o dia anterior e retornando as latas preta, branca, e vermelha. Podemos observar na Figura 19 uma temperatura mais constante com uma queda brusca, de uma potencial nuvem passageira. O que ocorre é que por descuido a coleta não foi encerrada às 14:30 e o sol foi parcialmente bloqueado pela casa vizinha bloqueando causando sombra na lata preta e branca, em após alguns minutos a lata vermelha. É possível perceber que a temperatura continua como previsto porém em temperaturas inferiores.

Os resultados na grande maioria do tempo mostrou trazer resultados esperados dados as suas condições. A temperatura entre as latas branco e preto onde era esperado uma maior diferença térmica chegaram em alguns dos gráficos demonstrados a valores acima de 30 graus. Para o experimento observado é possível presenciar que as latas apresentam temperaturas em menos de 5 minutos.

Figura 15 – Dia parcialmente nublado

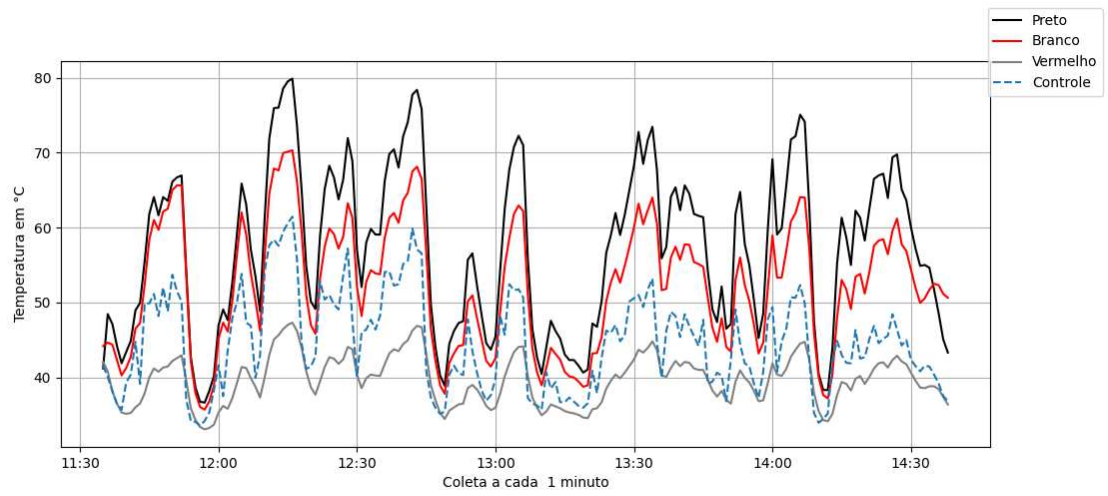


Fonte: O autor

Como um módulo de coleta independente, embora haviam sido criadas várias opções de menu, os que foram mais importante foram o menu de visualização de dados sensores em tempo real pelo monitor LCD, pois por vezes acontecia um mal contato de algum componente e a leitura de temperatura não era ocorrida, necessitando uma checagem de fios. Outros menus como o de limpeza de arquivos antigos no SD e o de coleta eram essenciais. Nem sempre a checagem de fios era eficaz, em algumas coletas os dados ocorreu afrouxamento dos fios de sensores algum tempo depois de iniciado a coleta ocasionando a gravação do valor -127 e estas coletas tiveram de ser invalidadas.

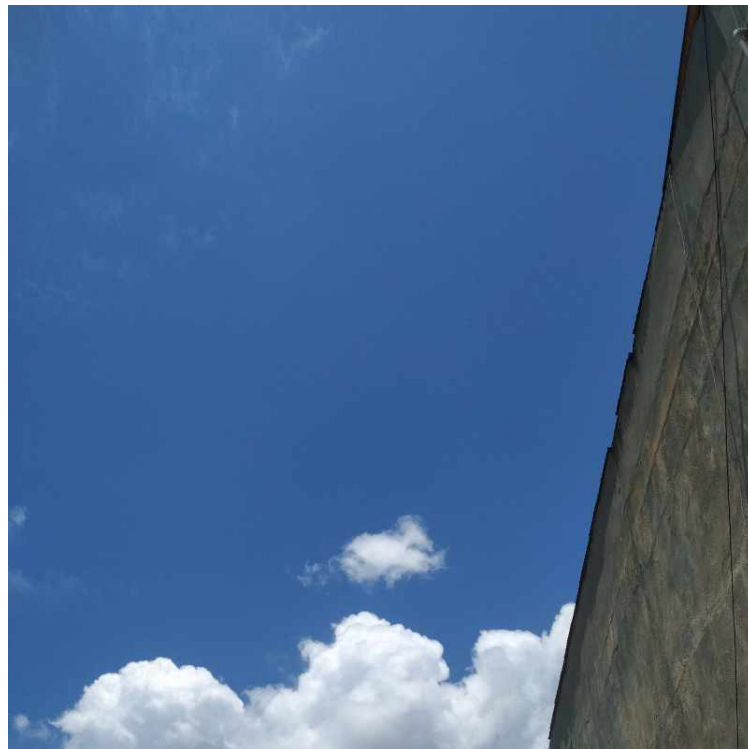
Com isso, podemos afirmar que o módulo de coleta consegue executar o seu papel como a de uma ferramenta que automatiza medições digitalmente e de forma precisa. Foi possível verificar que os dados geravam gráficos condizentes com o que era imaginado.

Figura 16 – Gráfico em dia parcialmente nublado



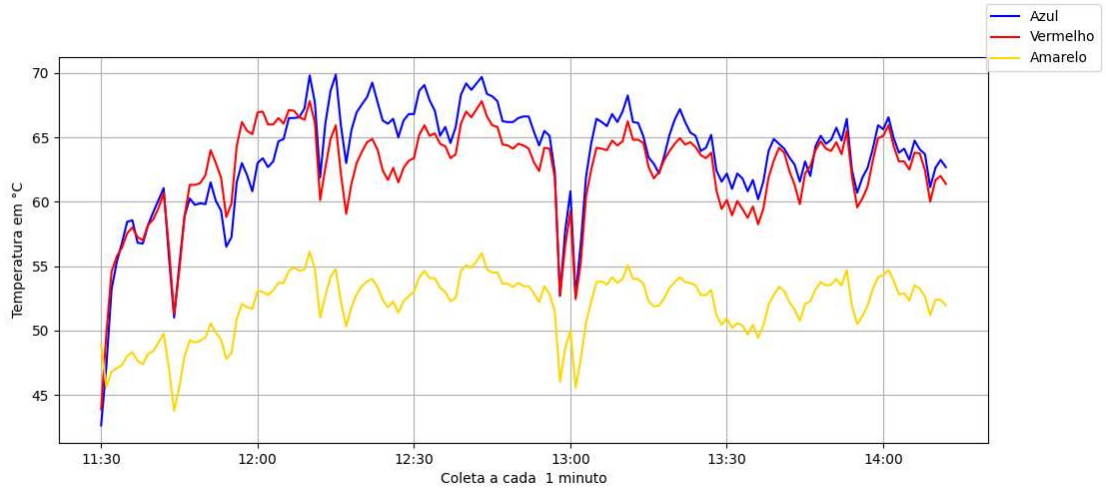
Fonte: O autor

Figura 17 – Dia ensolarado



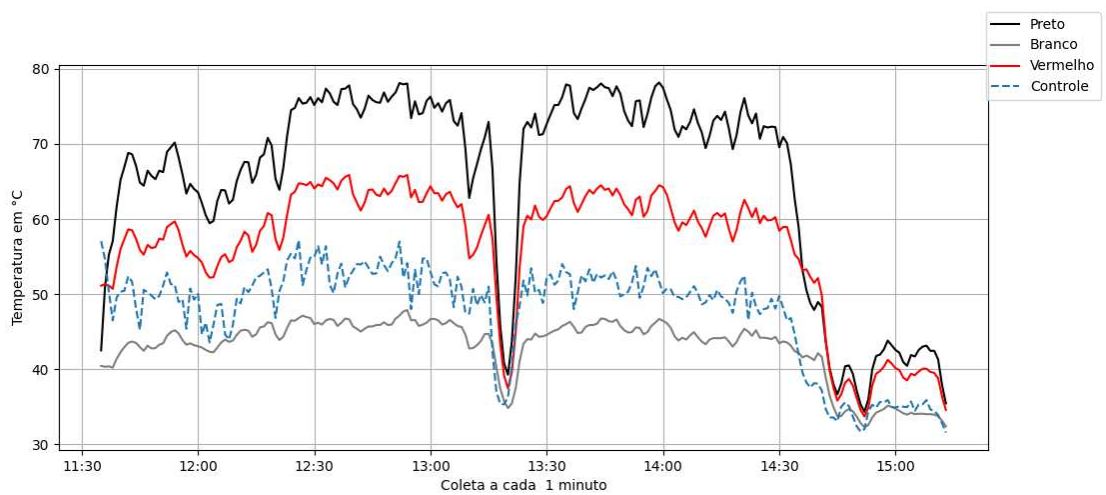
Fonte: O autor

Figura 18 – Gráfico de latas coloridas em dia ensolarado



Fonte: O autor

Figura 19 – Gráfico em dia ensolarado



Fonte: O autor

## 7 CONCLUSÃO E TRABALHOS FUTUROS

O trabalho aqui realizado mostrou que é possível criar sistemas de coleta com baixo custo, servindo de uma alternativa adicional e complementar ao ensino de física em sua forma prática e empírica. Como vantagem proporciona maior liberdade e flexibilidade, pois pode ser personalizado em diferentes tipos de experimentos, basta ligeiras modificações das partes físicas e dos softwares aqui apresentado, diferente do que costuma ocorrer em ferramentas comerciais.

É evidente que o modelo proposto se mostra promissor, atingindo seus objetivos como a automatização de medições, juntamente com um sistema interativo que pode modificar os procedimentos de coleta em campo sem uma necessidade de constante de se ter um computador atrelado.

### 7.1 TRABALHOS FUTUROS

O projeto ainda segue com diversas limitações. Os componentes físicos do Arduino por natureza são frágeis e não foram projetados para resistir a intempéries como insolação, altas temperaturas e água. O objetivo proposto foi para o uso em campo, o que faz isso entrar em conta como fator de risco, sendo o ideal a elaboração e organização dos componentes em invólucros mais fechados, como uma caixa personalizada em impressora 3D, o que serviria para uma maior facilidade de operação e transporte.

O sistema físico produzido neste projeto tinha uma grande quantidade de ligações com fios, o que trazia uma alta probabilidade de falhas em conectores. Por algumas vezes os experimentos de testes trouxeram dados nulos ocasionados por falhas de mal contato e tiveram de ser refeitos, o que significava por vezes a perda de todo um dia de coleta. Uma solução para a melhora nesse quesito seria a de soldagem de componentes em um sistema mais permanente, miniaturizado e menos passível a ocorrência de qualquer mal contato.

Para uma melhor automatização da coleta de dados seria ideal o uso de um módulo de relógio para registrar a data e hora precisa em que a coleta foi iniciada. Neste projeto tentei utilizar um módulo de relógio DS1302, porém não obtive muito sucesso, provavelmente

o módulo adquirido era defeituoso, é comum ver relatos de usuários em fóruns de internet contando sobre experiências de drenagem de bateria e outros problemas associados a este modelo de módulo, visto que certos fabricantes produzem versões com algumas falhas de projetos. Uma melhor recomendação seria utilizar um módulo mais confiável como o DS1307 ou DS3231. Vale resaltar que é necessário cautela ao manusear estes módulos com baterias de 3V, pois erros de configuração como voltagem incorreta pode acarretar em explosão e dano destas baterias. Há duas diferentes versões deste tipo de bateria 3V como a CR2025 (comum) e a CR2032 (recarregável), que pode receber ou não receber alimentação de retorno para recarga. Também vale ressaltar de que a depender da temperatura a velocidade de descarga da bateria costuma ser maior.

Outra funcionalidade que não pôde ser implementada foi a de um menu melhor estruturado para gerenciamento de arquivos de coletas. O sistema apenas usa um arquivo do cartão SD para armazenamento de dados, para nova captura é necessário deletar a última. É plenamente possível criar um simples menu que possa alternar entre arquivos, isso aumentaria a quantidade de dados guardadas sem a necessidade de a cada coleta ter de guardar em um computador, trazendo mais autonomia e independência.

Uma outra idéia alternativa e conjunta ao sistema de gerenciamento de arquivo seria a de inserir um sistema automático de geração de arquivos a cada nova coleta combinando a data e hora trazida pelo módulo de relógio para a nomeação do arquivo, causando uma melhor organização de dados. Por exemplo: *2020-11-10 10\_30\_05.txt*, ou seja, dia 10/11/2020 às 10:30:05.

Ainda outra melhoria é a de uma configuração mais automatizada de coleta, que ao programar um horário específico do dia, o módulo inicie e termine a coleta sem a necessidade de interação manual. No entanto para isso também necessitaria de um módulo de relógio confiável.

## REFERÊNCIAS

- ALECRIM, E. *Memórias RAM e ROM*. 2011. <<https://www.infowester.com/memoria.php>>. Acessado: 2020-07-19.
- AMORIM, C. A. de; CONCEIÇÃO, R. do C. Um ambiente aberto baseado em microcontrolador para experimentos didáticos de física: conversão fotovoltaica. 2019.
- ARDUINO. *Constantes*. 2020. <<https://www.arduino.cc/reference/pt/language/variables/constants/constants/>>. Acessado: 2020-08-07.
- ARDUINO. *Memory*. 2020. <<https://playground.arduino.cc/Learning/Memory/>>. Acessado: 2020-11-01.
- ARDUINO. *What is Arduino?* 2020. <<https://www.arduino.cc/en/Guide/Introduction>>. Acessado: 2020-07-14.
- BERGMAN, T. L.; LAVINE, A. S.; QUEIROZ, E. M. *Fundamentos de Transferência de Calor E de Massa*. [S.l.]: Grupo Gen-LTC, 2000.
- CAMPOS, A. *Mais memória no Arduino: indo além dos 2KB de RAM com a PROGMEM*. 2015. <<https://br-arduino.org/2015/06/arduino-progmem-sram.html>>. Acessado: 2020-11-01.
- CATELAN, S. S. Atividades experimentais como recurso pedagógico ao ensino de ciências naturais e matemática. *Universidade Federal de Mato Grosso*, 2008.
- FERNANDES, E. F. As dificuldades de compreender física dos alunos do ensino médio das escolas públicas de iguatu-ce. *Universidade Estadual do Ceará-UECE, Iguatu-Ceará*, 2016.
- FERNANDES, E. F. As dificuldades de compreender física dos alunos do ensino médio das escolas públicas de iguatu. *Universidade Estadual do Ceará - UECE*, 2016.
- FIOLHAIS, C.; TRINDADE, J. Física no computador: o computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. *Revista Brasileira de Ensino de Física*, scielo, v. 25, p. 259 – 272, 09 2003. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172003000300002&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172003000300002&nrm=iso)>.
- FOUNDATION, T. P. S. *The Python Tutorial*. 2018. <<https://docs.python.org/3/tutorial/index.html>>. Acessado: 2018-11-16.
- FRAGATA, F.; SENS, M.; SEBRÃO, M. Influência da cor de tintas de poliuretano na absorção e na dissipação de calor. *Corrosão e Proteção de Materiais*, scielopt, v. 34, p. 53 – 59, 12 2015. ISSN 0870-1164. Disponível em: <[http://www.scielo.mec.pt/scielo.php?script=sci\\_arttext&pid=S0870-11642015000200003&nrm=iso](http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S0870-11642015000200003&nrm=iso)>.
- GUDINO, M. *Introduction to Microcontrollers*. 2018. <<https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller>>. Acessado: 2020-08-06.

- HITACHI. *HD44780U (LCD-II) - Datasheet*. 1998. <<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>>. Acessado: 2019-11-06.
- HUANG, B. Open-source hardware–microcontrollers and physics education–integrating diy sensors and data acquisition with arduino. In: *the American Society for Engineering Education Annual Conference*. [S.l.: s.n.], 2015. v. 26, p. 1–26.
- MCROBERTS, M. *Arduino básico*. [S.l.]: Novatec Editora, 2018.
- OLIVEIRA, C. L. V.; ZANETTI, H. A. P. *Arduino descomplicado: como elaborar projetos de eletrônica*. [S.l.]: Saraiva Educação SA, 2015.
- PHILIPPHENKEL. *Arduino Uno Microcontroller Board*. 2018. <<https://commons.wikimedia.org/wiki/File:ArduinoUno.svg>>. Acessado: 2020-07-23.
- RANDOMNERDTUTORIALS.COM. *DHT11 vs DHT22 vs LM35 vs DS18B20 vs BME280 vs BMP180*. 2019. <<https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/>>. Acessado: 2020-08-15.
- ROCHA, F. S. da; MARANGHELLO, G. F.; LUCCHESI, M. M. Acelerômetro eletrônico e a placa arduino para ensino de física em tempo real. *Caderno Brasileiro de Ensino de Física*, v. 31, n. 1, p. 98–123, 2014.
- RODAS, R. Q. *Qual a influência das cores na temperatura do carro?* 2017. <<https://quatorrodas.abril.com.br/auto-servico/qual-a-influencia-da-cor-na-temperatura-do-carro/>>. Acessado: 2019-11-07.
- ROSA, Á. B. da; GIACOMELLI, A.; TRENTIN, M. A. S. Utilização de sensores de temperatura e da placa arduino como alternativa para um experimento de condução térmica. 2014.
- SCHNEIDER, F. *Comparação entre sensores de temperatura: LM35 X DS18B20*. 2020. <<https://automacaoifrsrg.wordpress.com/2014/05/11/comparacao-entre-sensores-de-temperatura-lm35-x-ds18b20/>>. Acessado: 2020-08-11.
- SILVA, D. C. M. da. *Processos de propagação de Calor*. 2020. <<https://brasilecola.uol.com.br/fisica/processo-propagacao-calor.htm>>. Acessado: 2019-11-02.
- SILVEIRA, M. R. A. da. A dificuldade da matemática no dizer do aluno: ressonâncias de sentido de um discurso. *Educação & Realidade*, Universidade Federal do Rio Grande do Sul, v. 36, n. 3, p. 761–779, 2011.
- SOUZA, A. R. d. et al. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. *Revista Brasileira de Ensino de Física*, scielo, v. 33, p. 01 – 05, 03 2011. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172011000100026&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172011000100026&nrm=iso)>.
- SOUZA, F. *Arduino UNO*. 2013. <<https://www.embarcados.com.br/arduino-uno/>>. Acessado: 2020-08-06.
- SYUHADA, A.; MAULANA, M. I. Heat transfer capability of solar radiation in colored roof and influence on room thermal comfort. *AIP Conference Proceedings*, v. 1931, n. 1, p. 030054, 2018. Disponível em: <<https://aip.scitation.org/doi/abs/10.1063/1.5024113>>.

YOUNG, H. D.; FREEDMAN, R. A.; FÍSICA, I. Termodinâmica e ondas. *São Paulo*, v. 10, 2008.