

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
Introdução à Computação
Mariana Caravanti de Souza

**Relógio Digital desenvolvido com a utilização de uma placa
Arduino Uno**

RTX-360

João Pedro Huppes Arenales - 202519040640
Kaio Campos Tadeu - 202519040375
Paola Campos da Silva - 202519040057
Valentina Campos Soares - 202519040391

Campo Grande
26 de junho de 2025

1 Introdução

A tecnologia está gradativamente presente na rotina cotidiana, sua utilização possibilita novos meios de comunicação, oportunidades econômicas e geração de renda. Portanto, essa se torna um importante capital social (BAGGIO, 2000). Em meio a importância da tecnologia, conforme seu avanço ocorria, tornou-se objetos cada vez mais digitais, um desses é o relógio utilizado como medida de tempo, no qual desenvolveu-se ao ponto de propor mais precisão e multifuncionalidade.

À luz da história, Whitrow (1972) impôs, em seu livro “O que é tempo?”, a evolução histórica do tempo e sua importância na civilização. Desde tempos antigos, as pessoas continham um instinto que proporcionavam compreender o momento para fazer determinadas funções, isso é nomeado de “Relógio Biológico”. Conforme novos conhecimentos foram surgindo, novos métodos de medição de tempo foram propostos, em sua maioria se relacionavam com as observações astronômicas. Entretanto, a medição exata do tempo era impedida, uma vez que conhecimentos mecânicos, que mais tarde proporcionariam relógios mais precisos, eram desconhecidos. Portanto, as civilizações ficavam restringidas as suas percepções da natureza, levando-as desenvolverem alguns projetos de relógios que foram utilizados durante anos, como o “Relógio de Água”, ou “clepsidra”, e o “Relógio de Sol”, ou “gnómon”.

Além disso, por mais que a importância dada a medida do tempo seja perpetuada ao longo da história da humanidade, como visto anteriormente, Whitrow (1972) expõe que nas últimas décadas, cresceu-se uma necessidade de medida de alta precisão. Sendo assim, Mello e Tonelli (2002) expõem como atualmente o tempo é gradativamente referenciado com palavras sinônimas de rapidez, como se este fosse devagar demais para os avanços da humanidade, transformando-o em um recurso escasso.

Nesse viés, conforme os conhecimentos da humanidade aprimoravam, cada vez mais o desenvolvimento dos relógios buscavam uma melhor precisão, proporcionando, assim, relógios que poderiam impor funcionalidades além das simples de mostrar horas e minutos, como o Relógio Digital.

O Relógio Digital, em geral, exibe o tempo em formato numérico, ou seja, diferente do Relógio Analógico que é preciso compreender os ponteiros, no Relógio Digital basta ler os valores numéricos para ter conhecimento do horário (JÚNIOR et al., 2023).

De acordo com Rodrigues (2024), os relógios digitais oferecem uma solução moderna e eficaz para a disseminação de mensagens comerciais e informações públicas, especialmente quando integrados à infraestrutura urbana. Essa multiplicidade de aplicações torna os relógios digitais um tema interessante e atual para estudos na área de tecnologia.

Além disso, Júnior et al. (2023) elaboraram um Relógio Digital com a utilização do Arduino Uno, estes apontam conhecimentos e componentes necessários para tal elaboração. Ao fim, justificam que a importância do projeto se dá ao possibilitar um melhor gerenciamento do tempo, uma vez que esse pode ser utilizado para diversas tarefas que necessitam de uma medição ou marcação do tempo decorrido, além de contribuir didaticamente, ao passo que envolve desenvolvimento de lógica e conhecimentos de circuitos e componentes elétricos.

Com isso, esse relatório expõe a elaboração de um Relógio Digital por meio da utilização de uma placa Arduino, a fim de compreender e aprimorar os conhecimentos acerca das funcionalidades e componentes conforme o desenvolvimento do tema.

1.1 Objetivo Geral

O objetivo geral do presente trabalho consistiu em desenvolver e documentar a elaboração de um relógio digital utilizando uma placa Arduino, detalhando os componentes, o processo de montagem e a verificação de seu funcionamento.

1.2 Objetivos Específicos

1. Listar e descrever os componentes eletrônicos utilizados na montagem de um relógio digital com Arduino.
2. Apresentar o procedimento de montagem do circuito do Relógio Digital, especificando conexões, estrutura lógica e organização do projeto.
3. Explicar o funcionamento básico do sistema implementado.
4. Testar e validar o funcionamento do código-fonte desenvolvido.

Considerando os apontamentos sobre a evolução dos relógios, este trabalho tem como propósito demonstrar a construção de um Relógio Digital utilizando a placa Arduino. Busca-se, assim, proporcionar novas perspectivas acerca do tema ao utilizar como plataforma o Arduino. De antemão, os capítulos seguintes estão organizados da seguinte forma: a Metodologia, composta pelas seções *Componentes*, *Circuito Elétrico* e *Fluxograma*; em seguida, são apresentados os Resultados e Discussões; e, por fim, a Conclusão, que retoma os objetivos e avalia o desempenho do projeto.

2 Metodologia

A princípio, ressalta-se que esse projeto não foi desenvolvido fisicamente, e sim utilizando o ambiente de simulação *Tinkercad* (TINKERCAD, 2025). Tal escolha se deu, principalmente, pela praticidade de se fazer via *internet*, levando em consideração que sua elaboração foi feita por um grupo formado por 4 estudantes, possibilitando um horário flexível e favorável a todos para que ocorra esse desenvolvimento, sem a necessidade de deslocamento para elaboração presencial, o que poderia restringir certos horários conforme a disponibilidade de cada um.

2.1 Componentes

Nessa seção serão expostos os componentes elétricos utilizados no presente projeto, explicando e justificando brevemente cada um.

2.1.1 Arduino UNO R3

O Arduino é ”um pequeno computador programável utilizado para estabelecer comunicação entre os componentes externos conectados a ele e as entradas e saídas do próprio Arduino” (SILVA, 2022). Além disso, este faz uma leitura de dezenas de sensores, por meio dos dados lidos pelos sensores é possível tratá-los com o auxílio de programações do *software*. No viés da física, há várias aplicações dos conhecimentos da área, uma vez que o

sistema Arduino lê sinais elétricos a partir de suas portas digitais e analógicas (MARTINAZZO et al., 2014). Silva (2022) ainda fala da vantagem da utilização do Arduino devido ao seu baixo custo.

Para a elaboração do projeto, utilizou-se o Arduino UNO R3, que contém 14 portas digitais, com 6 sendo PWM (Pulse Width Modulation), e 6 analógicas (MARTINAZZO et al., 2014).

2.1.2 *Display LCD 16 x 2*

A princípio, a função geral do LCD (*Liquid Crystal Display*, ou *Display* de Cristal Líquido) é a de exibir informações na tela por via eletrônica (SILVA, 2022). Para a tela do relógio digital, foi utilizado o LCD 16x2, que possui 2 linhas e 16 caracteres. Ao utilizar o LCD é preciso que uma biblioteca seja implementada na programação do Arduino de modo a interagir com o *display*, tal comunicação se dá "através de 16 pinos que funcionam para alimentação, comunicação de dados, ativação e controle de luminosidade" (LUCINI; CRUZ, 2022)

Além disso, optou-se por utilizar o display LCD com comunicação paralela no modo de 4 bits, possibilitando utilizar apenas seis conexões digitais ao Arduino. Apesar de existir display com interface I2c, que utiliza apenas dois pinos de dados, além da alimentação, o modelo paralelo foi considerado melhor durante o desenvolvimento, pois permite maior controle sobre o conteúdo exibido.

2.1.3 $1\text{ k}\Omega$ Resistor

O resistor é um componente utilizado para limitar a corrente elétrica em um circuito, sendo sua resistência medida em ohms (Ω). A relação entre a tensão elétrica (V), a corrente elétrica (I) e a resistência (R) é expressa pela Lei de Ohm, dada pela equação:

$$V = R \cdot I \quad (1)$$

Em que:

- V é a tensão em volts (V),
- R é a resistência em ohms (Ω),
- I é a corrente em amperes (A).

Com base nessa relação, é possível determinar a corrente que circulará pelo circuito a partir dos valores de resistência e tensão aplicados (COMMUNITY, 2025). Neste projeto, utilizou-se um resistor de $1\text{ k}\Omega$, com uma margem de tolerância de 5%.

2.1.4 $250\text{ k}\Omega$ Potenciômetro

O potenciômetro é um componente eletrônico cuja função é semelhante à de um resistor, com a diferença de que sua resistência pode ser ajustada manualmente por meio de um eixo rotativo. Assim, o potenciômetro permite controlar a resistência de forma contínua dentro de um intervalo pré-definido. A resistência equivalente (R) fornecida pelo potenciômetro depende da posição do cursor entre os terminais extremos. (BARELA, 2018).

Dessa forma, no desenvolvimento do projeto, utilizou-se o potenciômetro de $250\text{ k}\Omega$.

2.1.5 Botão

Este componente foi utilizado para o usuário conseguir acessar as funções presentes no Relógio Digital ao clicar nos botões, que contêm funcionalidades próprias.

2.1.6 Sensor de Temperatura (TMP36)

Como funcionalidade a mais, optou-se por mostrar a temperatura com a utilização de um sensor de temperatura, o TMP36, que é ”um sensor analógico de temperatura capaz de medir com precisão uma extensa faixa climática através de um diodo que aumenta sua voltagem de acordo com a temperatura”(MEDEIROS et al., 2022).

2.1.7 Piezo

Segundo Geddes (2017) o componente eletrônico Piezo é um tipo de Buzzer que ”é capaz de gerar apenas um som de clique, que é criado aplicando-se uma tensão”, portanto, é possível propor frequências específicas para criar notas reconhecíveis. Este componente foi implementado de forma a proporcionar uma funcionalidade que ecoa músicas.

2.2 Circuito Elétrico

O circuito eletrônico implementado para o funcionamento do Relógio Digital com Arduino Uno é composto por ligações entre os componentes já descritos em subseção anterior. Esta subseção dedica-se a explicar o modo como tais componentes estão conectados fisicamente e os fundamentos elétricos e físicos que justificam seus funcionamentos.

2.2.1 Distribuição e Organização das Ligações

Para a compreensão da organização citada nessa subseção da subseção *Circuito Elétrico*, a figura 1 a seguir demonstra como ocorreu a organização dos componentes no Arduino Uno, sendo esta obtida pelo site de simulação previamente dito.

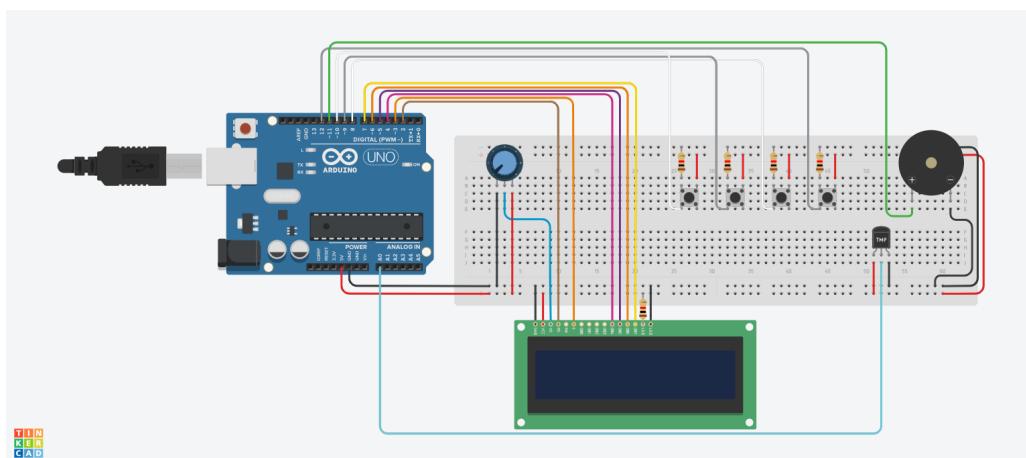


Figura 1: Organização do circuito com Arduino Uno.

Assim, é visto que o Arduino Uno está conectado a todos os componentes por meio de uma protoboard. Além disso, a alimentação elétrica do sistema é fornecida pela porta

USB, com tensão de 5V. O terminal 5V do Arduino distribui energia positiva a todos os componentes, enquanto o terminal GND fornece potencial nulo.

A seguir, destacam-se as principais conexões:

- O display LCD 16x2 encontra-se ligado nos pinos digitais D2-D7;
 - O sensor de temperatura está conectado à entrada analógica A0;
 - O buzzer é ativado por meio do pino digital D11;
 - Os botões de controle estão conectados aos pinos D8, D9, D10 e D12;
 - Um potenciômetro de $250\text{ k}\Omega$ ajusta o contraste do display via o pino VO.

2.2.2 Funcionamento Elétrico do Circuito

Segundo no viés elétrico, a Figura 2 demonstra o Circuito Elétrico também disponibilizado pelo site de simulação utilizado.

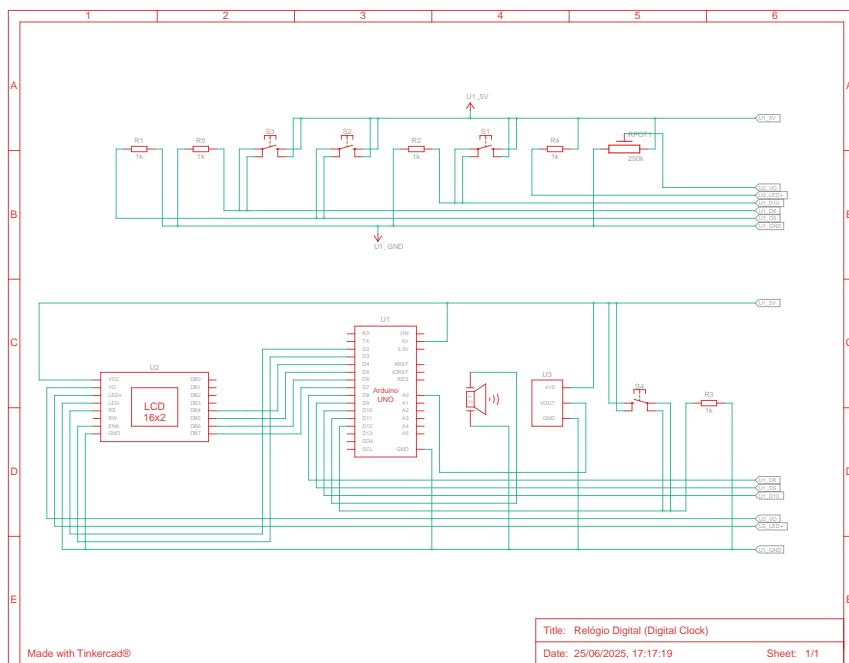


Figura 2: Circuito Elétrico do Relógio Digital.

O circuito opera a 5V, padrão do Arduino Uno. A corrente elétrica flui dos terminais de 5V para os diversos componentes e retorna ao GND, formando os laços de corrente necessários para o funcionamento.

Entretanto, para o funcionamento dos botões do circuito foram empregados resistores de $1\text{ k}\Omega$ em série, nos quais limitam a corrente que circula quando os botões são pressionados. Tal medida previne danos aos pinos digitais do microcontrolador e contribui para a estabilidade do sinal lógico. A corrente elétrica é regida pela Lei de Ohm:

$$I = \frac{V}{R} \quad (2)$$

Assim, em um acionamento com 5V, a corrente fica limitada a aproximadamente 5mA.

Além disso, os botões funcionam como chaves momentâneas, conectando os pinos digitais do Arduino ao GND quando pressionados. Isso permite a leitura de sinais de *nível lógico baixo* (LOW) e, quando abertos, o estado retorna ao *nível alto* (HIGH), assumido por padrão. Esse funcionamento possibilita a navegação entre modos e ajustes de parâmetros no Relógio Digital.

Pensando na divisão de tensão, o potenciômetro, conectado entre os terminais 5V e GND, possui seu cursor intermediário ligado ao terminal VO do LCD. Essa configuração forma um divisor de tensão, cuja tensão de saída é dada por:

$$V_{\text{saída}} = V_{\text{entrada}} \cdot \frac{R_2}{R_1 + R_2} \quad (3)$$

Tal tensão determina o contraste da tela de cristal líquido, sendo ajustável.

Indo para o sensor de temperatura (TMP36), este fornece uma tensão proporcional à temperatura ambiente em seu terminal de saída. Essa tensão é lida pela entrada analógica do Arduino, que realiza a conversão através de seu conversor analógico-digital (ADC). O ADC transforma o sinal analógico (tensão contínua) em um valor digital que pode ser tratado no código. A partir desse valor, calcula-se a temperatura em graus Celsius por meio da seguinte equação:

$$T(^{\circ}\text{C}) = \left(\frac{V_{\text{ADC}}}{1023} \cdot 5 - 0,5 \right) \cdot 100 \quad (4)$$

Nessa expressão, V_{ADC} representa a leitura do conversor ADC, e os coeficientes refletem o comportamento linear típico do sensor TMP36, cuja tensão de repouso a 0°C é de 0.5.

O buzzer, ou Piezo, é comandado por sinais de modulação por largura de pulso (PWM) enviados pelo pino digital D11. A produção sonora ocorre pela vibração interna do buzzer em resposta a um sinal de frequência variável. A relação entre frequência e tempo é dada por:

$$f = \frac{1}{T} \quad (5)$$

em que f representa a frequência sonora (em hertz) e T o período do sinal. Notas musicais específicas são geradas ao aplicar frequências padronizadas.

Através da combinação de entradas digitais (botões), entrada analógica (sensor), saídas visuais (LCD) e saídas sonoras (buzzer), o projeto se apresenta como uma aplicação prática de fundamentos físicos e lógicos.

2.3 Fluxograma

O código do sistema implementado com Arduino para a elaboração do Relógio Digital é composto por três módulos principais: Exibição do Horário; Exibição da Temperatura; e, Execução de Músicas. O fluxo do programa segue uma linha ciclífica, alternando entre esses três modos citados. Sendo assim, a Figura 3 apresenta o Fluxograma Geral de funcionamento do sistema.

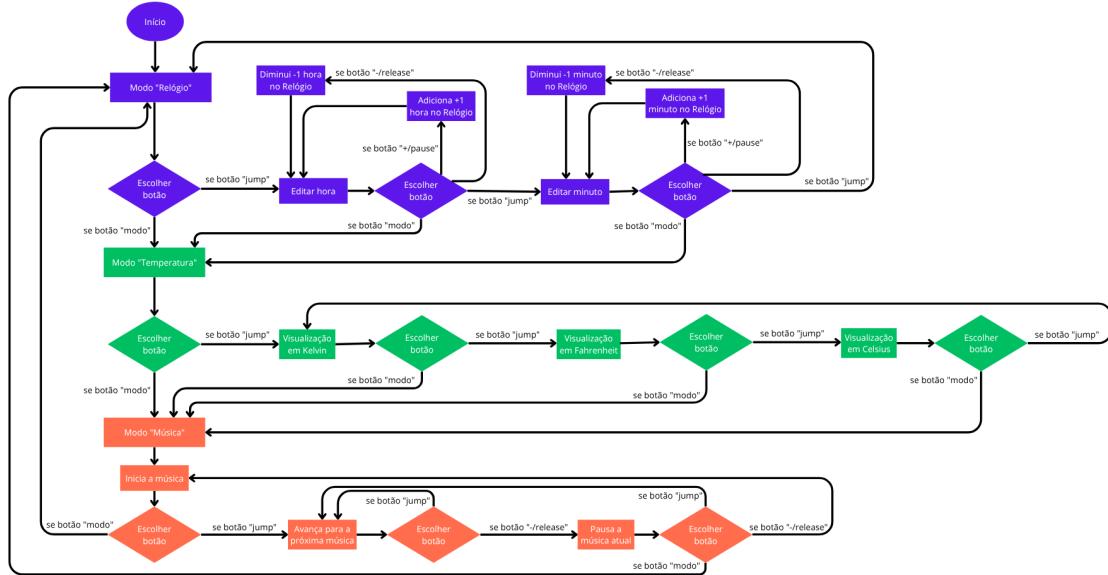


Figura 3: Fluxograma Geral do código do Relógio Digital.

Ao analisar o Fluxograma Geral, observa-se que, assim que ocorre a inicialização do código, o sistema entra em um *loop*, em que o usuário pode alternar entre os modos e operar funcionalidades utilizando outros 3 botões existentes.

A princípio, a Figura 4 apresenta o trecho de código responsável pela variação de modos.



Figura 4: Trecho de código com função de alteração de modo.

Tal código promove a mudança de modo utilizando um contador, seu funcionamento ocorre quando o botão designado "modo" é pressionado, com isso implementando um no contador até que seu resultado chegue no valor três, a partir desse valor os modos entram em um ciclo voltando para o modo "Relógio" e assim por diante.

Para tanto, o código faz a seguinte conta:

$$pag = (pag + 1) \% 3 \quad (6)$$

Ou seja, independente do valor do contador "pag", ele sempre fica entre 0 a 2.

A seguir, as subseções *Modo "Relógio"*, *Modo "Temperatura"* e *Modo "Música"* apresentam os trechos relevantes do código desenvolvido.

2.3.1 Modo "Relógio"

A princípio, o primeiro modo ao iniciar o sistema é o modo "Relógio", seu código está apresentado a seguir na Figura 5.

De antemão, as possibilidades de escolhas dadas ao usuário abordam desde seguir para o próximo modo até fazer mudanças dos valores apresentados, sendo estes: horas e minutos.

```

switch(cnt){ // Começo do tratamento de passagem de tempo
    case 2:
        m = m + dg;
        dg = 0; if(m > 59){
            m = 0;};
        if(m < 0){
            m = 59;};
        break;
    case 1:
        h = h + dg;
        dg = 0;if(h > 23){
            h = h - 24;};
        if(h < 0){
            h = 23;};
        break;
    case 3:
        dt = dt + dg;
        dg = 0;if(dt > 1[m]){
            dt = 1[m];};
        if(dt < 1){
            dt = 1;};
        break;
    case 4:
        mo = mo + dg;
        dg = 0;if(mo > 12){
            mo = 1;};
}
}

if(s < 1)
mo = 12;
if(dt > 1[mo]){
    dt=1;
    mo++;
    mo = (1 + (mo-1) % 12);
}break;
}

if(s > 59){ // tratamento de 60 sec = 1 min | 60 min = 1h | 24h = 0h0m0s
    s = 0;
    m++;
    if(m > 59){
        m = 0;
        m++;
        if(h > 23){
            h = 0;
            dt++;
            if(dt > 1[mo]){
                dt = 1;
                mo++;
            }
            if(mo > 12){
                mo = 1;
            }
        }
    }
}

```

Figura 5: Trecho de código com função do modo Relógio

Este código, em geral, aborda o tratamento da passagem do tempo, realiza a modificação manual feita pelo usuário e trata a passagem de 60 segundos para 1 minuto e os outros casos corelacionados.

Ademais, ainda apresenta atuação independente de qual modo o usuário está, portanto, mesmo se o usuário seguir para os outros modos, o relógio persistirá funcionando como em "segundo plano".

2.3.2 Modo "Temperatura"

Quando no modo Temperatura o sistema passa a expor a temperatura captada pelo sensor TMP36, ao converter a tensão recebida para Celsius, o sistema possibilita que o usuário escolhe entre ver a temperatura em 3 diferentes escalas: Celsius, Kelvin e Fahrenheit.

A Figura 6 expõe o código que possibilita essa alternância.

```

if ((millis() - lastDebounceTime) > debounceDelay) {
    // define se é Celsius, Kelvin ou Fahrenheit
    if (digitalRead(jump) == HIGH) {
        tipo_clima = (tipo_clima + 1)%3;
    }
}

```

Figura 6: Trecho de código com função de alteração de temperatura.

A lógica utilizada nesse trecho é semelhante à forma utilizada para alternar entre os modos no Relógio, apresentada na Figura 4.

Além disso, incrementou-se mensagens diversas que são apresentadas na tela conforme a temperatura registrada pelo sensor, visto a seguir na Figura 7:

```

lcd.setCursor(0,1);
if (temperaturaC < 30 && temperaturaC > 20){
    lcd.print("Esta agradável^^");
} else if (temperaturaC >= 30 && temperaturaC < 40){
    lcd.print("Esta bem calor!");
} else if (temperaturaC <= 20 && temperaturaC > 0){
    lcd.print("Esta friozinho");
}
else if (temperaturaC <= 0){
    lcd.print("ESTA CONGELANDO!!!");
} else if (temperaturaC >= 40){
    lcd.print("EU VOU DERRETER!!!");
}

```

Figura 7: Trecho de código com função de apresentar texto dinâmico conforme a temperatura registrada.

Essa mensagem é apresentada antes da temperatura, ou seja, primeiro o usuário visualiza uma mensagem dinâmica e, em seguida, obtém a temperatura registrada. Para alternar a possibilidade de fazer um código para cada escala de temperatura, esse código é feito apenas para a escala Celsius e depois faz a conversão para a escala escolhida, sem necessitar da conversão em todos os casos.

2.3.3 Modo “Música”

Seguindo para o último modo, quando o usuário está no Modo Música, este já inicia tocando uma das duas músicas existentes. Como visto no Fluxograma, o usuário pode pausar e despausar uma música, quando esta está pausada, ele passa a ter 3 opções para

prosseguir: despausar, trocar de música e ir para o modo ”Relógio”. Tal código está apresentado na Figura 8.

```
while (stop) {
    if (selecao_musica < 0){
        selecao_musica = 0;
    }
    // primeiro caso -> volta a tocar:
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (digitalRead(release) == HIGH) {
            stop = false;
            lastDebounceTime = millis();
            // Não altera selecao_musica aqui, mantém a atual
            break; // Sai do Loop while imediatamente
        }
    }
    // segundo caso -> quer pular a musica
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (digitalRead(jump) == HIGH) {
            selecao_musica = (selecao_musica + 1) % 2;
            Serial.println("Pulando para musica: " + String(selecao_musica));
            stop = false;
            lastDebounceTime = millis();
            break; // Sai do Loop while imediatamente
        }
    }
    // terceiro caso -> voltar pro relogio
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (digitalRead(modo) == HIGH) {
            selecao_musica = -1;
            stop = false;
            lastDebounceTime = millis();
            break; // Sai do Loop while imediatamente
        }
    }
}
```

Figura 8: Trecho de código com função de opções no modo ”Música”.

Seguindo adiante, quando a música não está pausada o usuário perde a possibilidade de alternar de modo, uma vez que a música tocando se encontra dentro de um loop, no qual não finaliza mesmo quando este seleciona a troca de modo. O próximo código presente na Figura 9 expõe um exemplo de uma das músicas, na qual é definida sua duração e as notas presentes nessa.

Figura 9: Trecho de código com função de interrupção da música atual.

Tal código utilizou de base a metodologia criada pelo brasileiro Robson Couto, contendo alterações como a opção de pausar a música (COUTO, 2019).

3 Resultados e Discussões

O presente trabalho tem como objetivo a construção de um relógio digital utilizando a plataforma Arduino e que possua diversas funcionalidades, visto que não existem muitos projetos com uma temática parecida com essa. Os objetivos desse relatório incluem a elaboração do relógio digital, o detalhamento de seus componentes, a apresentação de seu processo de montagem, a verificação de seu funcionamento e a explicação de seu sistema.

Durante o planejamento, existiam diversas expectativas com relação a como o relógio digital funcionaria e como ele seria elaborado, dentre elas, a possibilidade de obtenção do horário de forma automática por meio de bibliotecas específicas do Arduino, o que, infelizmente, não foi possível, visto que várias dessas bibliotecas estavam desatualizadas ou elas não funcionavam no ambiente de simulação *Tinkercad* (TINKERCAD, 2025), bem como métodos que possuíam estruturas muito complexas para o entendimento básico proporcionado pelo curso introdutório ao qual esse relatório é dirigido, então foi necessário fazer com que o horário do relógio fosse ajustado manualmente por meio de botões presentes no próprio Arduino.

Além disso, outra expectativa inicial era a exibição da temperatura atual junto do horário no LCD, a tela de exibição utilizada, o que teve uma melhoria, pois houve o desenvolvimento de um botão no próprio Arduino que permitisse alternar entre diferentes “estados” de tela, ou seja, que fosse possível a exibição de um “estado” da tela em que ela mostrasse o relógio e em outro em que ela mostrasse a temperatura, assim como um comentário personalizado dependendo da temperatura atual do ambiente. Também foi adicionado outro “estado” de tela, no qual existem opções de músicas que podem ser escolhidas pelo usuário por meio de um botão e que são tocadas por um *buzzer*, ambos presentes no próprio Arduino.

Exibição de Horário

A tabela presente abaixo visa demonstrar os valores assumidos pelo visor do LCD ao decorrer dos valores de teste. A contagem de tempo é realizada por incremento manual

da variável de segundos, com passagem automática ao final da contagem. Os valores são atualizados a cada segundo no *display*.

Tabela 1: Valores de tempo observados durante simulação

Unidade	Valor Esperado	Valor Obtido	Observação
Horas	0–23	Corretos	Avanço esperado por transição
Minutos	0–59	Corretos	Comportamento conforme esperado
Segundos	0–59	Corretos	Atualização visual sincronizada

Leitura de Temperatura

O sistema realiza a leitura do sensor analógico, convertendo a tensão lida em temperatura com a seguinte expressão:

$$T(^{\circ}\text{C}) = ((V_{\text{sensor}}/1023) \times 5 - 0,5) \times 100 \quad (7)$$

Essa temperatura pode ser exibida nas escalas Celsius, Kelvin e Fahrenheit, conforme o opção selecionada.

Tabela 2: Respostas do sistema para valores simulados de temperatura

Tensão (V)	Temp. Estimada ($^{\circ}\text{C}$)	Mensagem LCD
0,75	25	”Está agradável ^ ^”
0,90	40	”EU VOU DERRETER!!!”
0,50	0	”ESTÁ CONGELANDO!!”

Execução de Músicas

O *buzzer* realiza a reprodução sonora por meio da função `tone()` do Arduino. Foram implementadas duas trilhas sonoras: ”Tetris OST” e ”Marcha Imperial”. A navegação entre faixas e controle de pausa/reprodução é feita por meio de botões.

O sistema se mostrou operacional e eficiente dentro das expectativas. O *debouncing* dos botões foi implementado, que é o tratamento de um efeito que consiste na remoção de ressaltos, ou *bounces*, que ocorrem por conta de quando os contatos de um interruptor mecânico se chocam, fazendo com que eles ricochetiem um pouco antes de se estabilizarem (JACK G. GANSSLE, 2007). A contagem de tempo mostrou-se funcional durante a simulação, o que reforça a viabilidade do uso da lógica de *software* como alternativa em projetos didáticos.

O presente projeto possuí um grau de inovação considerável, visto que não existem muitos trabalhos acadêmicos que falem sobre e desenvolvam relógios digitais em uma placa Arduino, muito menos com as suas funcionalidades *bônus*, como a exibição da temperatura e a possibilidade de que músicas sejam tocadas, aprimorando as estruturas convencionais de um relógio digital nesses quesitos. Tais motivos fizeram com que a busca por referências acadêmicas e fontes práticas se tornasse um desafio, ao mesmo tempo que possibilitou que o projeto se destacasse, tanto pelo seu caráter inovador quanto pelo fato de ele reforçar o seu valor como experiência prática e como contribuição didática.

Apesar dessas dificuldades, foi possível achar alguns trabalhos que se assemelhassem a esse projeto, como: o trabalho desenvolvido por Júnior et al. (2023), o qual visou a

criação e implementação de um relógio analógico por meio de uma placa de *Arduino Uno* a fim de que ele funcionasse como um temporizador. Além disso, ainda existe o trabalho desenvolvido por Silva (2022), que apesar de possuir menor relevância, também demonstra características relativamente semelhantes ao projeto em questão, visto que ele consiste em um relógio contador para partidas de xadrez.

Existiram diversos desafios e limitações no desenvolvimento desse projeto, como, por exemplo, a pouca flexibilidade proporcionada pela tela LCD, tendo em vista o seu tamanho e número de bytes limitados, bem como a dificuldade para o tratamento do *debouncing* dos botões.

Dentre os dados atípicos, consta-se que durante a elaboração do projeto, tiveram diversos erros complexos com relação a um laço condicional no código que conferia qual seria o "estado" de tela que seria usado, mais especificamente, o de tocar música. Tal erro foi corrigido por meio da mudança no tratamento do *debouncing* e na declaração das variáveis ativadoras fora de qualquer função, ao invés de serem declaradas na função `void setup()`.

Conclui-se que o sistema é capaz de integrar com sucesso três funcionalidades distintas. As principais contribuições incluem a utilização do LCD como *display* para representação dos dígitos, a interpretação de sensores analógicos e o controle musical por meio de botões e *buzzers*. Em vista disso, o projeto mostra-se promissor para aplicações didáticas.

Abaixo se encontra o link da plataforma *Tinkercad* TINKERCAD (2025), na qual foi feita a simulação do projeto.

https://www.tinkercad.com/things/36kzDeIk5x3/edit#?returnTo=%2Fdashboard&sharecode=u51D5q36EGcgMbWyzAUupwF8PEu3wMQ5v-_LddKm60Y

4 Conclusão

No fim, foi possível desenvolver um sistema em Arduino de relógio digital que não só mostrasse o horário, e permitisse ajustá-lo de acordo com as necessidades do momento, mas também exibisse a temperatura atual do ambiente, e comentários personalizados dependendo dela, bem como permitisse que tocasse músicas, e exibisse os seus nomes na tela do LCD.

O projeto em questão possibilitou que os seus contribuintes colocassem em prática os seus conhecimentos, assim como ajudou a ressaltar a relevância do Arduino como um instrumento de aprendizado e como uma interessante estrutura de *hardware*.

Os objetivos desse trabalho foram cumpridos, desde do objetivo geral, o qual evidencia a elaboração de um relógio digital a partir da utilização de placa de Arduino, até os objetivos específicos, os quais pretendiam que os componentes eletrônicos do circuito dessa mesma placa fossem listados e descritos, que a montagem do circuito e toda a sua lógica e seus procedimentos fossem apresentados, que o funcionamento do sistema implementado fosse devidamente explicado, e que código desenvolvido fosse testado e validado de forma correta.

Esse trabalho agrupa amplamente para o aprendizado de seus contribuintes, assim como para a sociedade e para o mundo da tecnologia como um todo, visto que apresenta uma inovação na elaboração e retratação de um relógio digital por conta de suas funcionalidades *bônus*.

O presente projeto pode ter diversas melhorias, dentre elas, pode-se destacar o uso de um método para a obtenção automática do horário, o uso do *display* LCD para a exibição de *emojis* (desenhos simplificados que transmitem informações e sentimentos para os seus

espectadores), o acréscimo de uma funcionalidade de alarme, a qual utilizaria o ”estado” de tela do relógio para selecionar um horário para que o *buzzer* acionasse e avisasse a chegada do horário selecionado, e a possibilidade de simplificação do código.

Referências

- BAGGIO, Rodrigo. A sociedade da informação e a infoexclusão. **Ciência da Informação**, IBICT, v. 29, n. 2, p. 16–21, mai. 2000. Acesso em: 07 mai. 2025. ISSN 0100-1965. Disponível em: [\[https://doi.org/10.1590/S0100-19652000000200003\]](https://doi.org/10.1590/S0100-19652000000200003).
- BARELA, Anne. **Make It Change: Potentiometers**. Acesso em: 25 jun. 2025. Set. 2018. Disponível em:
[\[https://learn.adafruit.com/make-it-change-potentiometers?view=all\]](https://learn.adafruit.com/make-it-change-potentiometers?view=all).
- COMMUNITY, Instructables. **Resistors**. Acesso em: 25 jun. 2025. Disponível em:
[\[https://www.instructables.com/Resistors/\]](https://www.instructables.com/Resistors/). Acesso em: 13 jun. 2025.
- COUTO, Robson. **arduino-songs**. Repositório GitHub. 2019. Disponível em:
[\[https://github.com/robsoncoutho/arduino-songs\]](https://github.com/robsoncoutho/arduino-songs).
- GEDDES, Mark. **Manual de projetos do Arduino: 25 projetos práticos para começar**. [S.l.]: Novatec Editora, 2017. Acesso em: 25 jun. 2025. Disponível em:
[\[https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=ooSKDgAAQBAJ&oi=fnd&pg=PA8&dq=piezo+arduino&ots=iCqN2z1IsT&sig=3jZe1fxEsrww29gb6dDu1DqfkHU#v=onepage&q=piezo%20arduino&f=false\]](https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=ooSKDgAAQBAJ&oi=fnd&pg=PA8&dq=piezo+arduino&ots=iCqN2z1IsT&sig=3jZe1fxEsrww29gb6dDu1DqfkHU#v=onepage&q=piezo%20arduino&f=false).
- JACK G. GANSSLE. **A Guide to Debouncing**. [S.l.: s.n.], abr. 2007. PDF lecture notes, CSE140L, UCSD. Acesso em: 25 jun. 2025. Disponível em:
[\[https://cseweb.ucsd.edu/classes/sp07/cse140L/debounce.pdf\]](https://cseweb.ucsd.edu/classes/sp07/cse140L/debounce.pdf). Acesso em: 25 jun. 2025.
- JÚNIOR, Ângelo et al. **Projecto Executivo de um Relógio Analógico Personalizado com Arduino Uno**. Acesso em: 24 jun. 2025. 2023. Disponível em:
[\[https://pt.scribd.com/document/714407077/relogio0\]](https://pt.scribd.com/document/714407077/relogio0).
- LUCINI, André; CRUZ, Marlon. **Sistema de monitoria de reservatório d’água com acesso remoto utilizando Arduino**. 2022. Trabalho de Conclusão de Curso (Tecnologia em Sistemas de Telecomunicações) – Universidade Tecnológica Federal do Paraná, Curitiba. Acesso em: 25 jun. 2025.
- MARTINAZZO, Claodomir et al. Arduino: Uma tecnologia no ensino de física. **Revista Perspecfiva**, v. 38, n. 143, 2014. Acesso em: 25 jun. 2025. Disponível em:
[\[https://eadcampus.spo.ifsp.edu.br/pluginfile.php/938271/mod_resource/content/1/143_430.pdf\]](https://eadcampus.spo.ifsp.edu.br/pluginfile.php/938271/mod_resource/content/1/143_430.pdf).
- MEDEIROS, Miguel Pereira et al. Arduino e cultivo: projeto de hidroponia residencial. **Pensar Acadêmico**, v. 20, n. 1, p. 49–64, 2022. Acesso em: 25 jun. 2025. Disponível em: [\[https://pensaracademicounifacig.edu.br/index.php/pensaracademicounifacig/article/view/2756/2454\]](https://pensaracademicounifacig.edu.br/index.php/pensaracademicounifacig/article/view/2756/2454).

MELLO, Hivy Damásio Araújo; TONELLI, Maria José. O tempo e as organizações: Concepções do tempo em periódicos de estudos organizacionais. **Anais do II Encontro Nacional de Estudos Organizacionais (ENEÓ)**, 2002. Acesso em: 13 mai. 2025.

Disponível em: [jhttps://gvpesquisa.fgv.br/sites/gvpesquisa.fgv.br/files/arquivos/tonelli_o_tempo_e_as_organizacoes_concepcoes_do_tempo_em_periodicos.pdf](https://gvpesquisa.fgv.br/sites/gvpesquisa.fgv.br/files/arquivos/tonelli_o_tempo_e_as_organizacoes_concepcoes_do_tempo_em_periodicos.pdf).

RODRIGUES, Diego. **PLANO DE NEGÓCIOS: Relevância da Mídia Out Of Home em Era Digital - Um Estudo de Viabilidade Técnico-Econômico para a implantação de Relógios Eletrônicos Digitais em Belo Horizonte**. 2024. Diss. (Mestrado) – Universidade Federal de Minas Gerais, Faculdade de Ciências Econômicas. Acesso em: 10 mai. 2025. Disponível em: [jhttp://hdl.handle.net/1843/77814](http://hdl.handle.net/1843/77814).

SILVA, Gustavo da. **Relógio em Arduino desenvolvido para as diferentes modalidades de tempo do xadrez**. [S.l.: s.n.], 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação). Acesso em: 25 jun. 2025. Disponível em: [jhttps://repositorio.uniceub.br/jspui/handle/prefix/17274](https://repositorio.uniceub.br/jspui/handle/prefix/17274). Acesso em: 24 jun. 2025.

TINKERCAD, AUTODESK. **Tinkercad**. [S.l.: s.n.], 2025. site de simulação online. Acesso em: 25 jun. 2025.

WHITROW, G. J. **What is Time?** [S.l.]: Thames & Hudson, 1972. Acesso em: 13 mai. 2025. Disponível em:
[jhttps://books.google.com.br/books?hl=pt-BR&lr=&id=CSBqghMnvjAC&oi=fnd&pg=PA13&dq=medi%C3%A7%C3%A3o+do+tempo+hist%C3%B3ria&ots=WRnCiuSAj-&sig=ibGA1UPZ8GtbNnYZ2nBBjDXePjM#v=onepage&q&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=CSBqghMnvjAC&oi=fnd&pg=PA13&dq=medi%C3%A7%C3%A3o+do+tempo+hist%C3%B3ria&ots=WRnCiuSAj-&sig=ibGA1UPZ8GtbNnYZ2nBBjDXePjM#v=onepage&q&f=false).