

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
ENGENHARIA ELETRÔNICA

ADIL DE OLIVEIRA COELHO NETO  
DANIEL GRAZZIOTIN BONGIOLO

**AMANHECER: SISTEMA DE DESPERTADOR CONECTADO À  
INTERNET, COM APLICATIVO DE CONFIGURAÇÃO E  
SIMULAÇÃO DE NASCER DO SOL**

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

CURITIBA  
2021

ADIL DE OLIVEIRA COELHO NETO  
DANIEL GRAZZIOTIN BONGIOLO

**AMANHECER: SISTEMA DE DESPERTADOR CONECTADO À  
INTERNET, COM APLICATIVO DE CONFIGURAÇÃO E  
SIMULAÇÃO DE NASCER DO SOL**

Relatório de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Curitiba, como requisito parcial para obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Dr. Rubens Alexandre de Faria  
UTFPR

CURITIBA  
2021

**ADIL DE OLIVEIRA COELHO NETO  
DANIEL GRAZZIOTIN BONGIOLO**

**AMANHECER: SISTEMA DE DESPERTADOR CONECTADO À INTERNET, COM  
APLICATIVO DE CONFIGURAÇÃO E SIMULAÇÃO DE NASCER DO SOL**

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

Data de aprovação: 18/Novembro/2021

---

Prof. Dr. Rubens Alexandre de Faria  
Orientador  
Universidade Tecnológica Federal do Paraná

---

Prof. Dr. Elder Oroski  
Membro da banca  
Universidade Tecnológica Federal do Paraná

---

Prof. Dr. Guilherme de Santi Peron  
Membro da banca  
Universidade Tecnológica Federal do Paraná

**CURITIBA**

**2021**

Dedicamos esse trabalho ao professor Hugo Vieira Neto, um homem cuja inspiração por ele causada será sempre lembrada.

## **AGRADECIMENTOS**

Agradecemos à professora Simone Crocetti pela oportunidade de desenvolver a ideia no tempo em que lecionou Engenharia do Produto a nós.

Agradecemos também ao professor Rubens Alexandre de Faria pela orientação e incentivo.

Por fim, agradecemos aos nossos familiares, cujo apoio foi indispensável.

*“Um homem forte não precisa saber o futuro, ele cria o seu próprio.”*

Frase atribuída ao personagem Solid Snake  
(Jogo eletrônico Metal Gear Solid, 1998)

## RESUMO

BONGIOLO, Daniel. COELHO, Adil. Amanhecer: sistema de despertador conectado à internet, com aplicativo de configuração e simulação de nascer do Sol. 2021. 42 f. Trabalho de Conclusão de Curso de graduação – Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

A vida corrida de algumas pessoas leva à diminuição da qualidade do sono. Isso causa alguns problemas como sonolência, humor instável e baixa produtividade. Segundo estudos, como o de Virginie Gabel et al, acordar exposto a um sistema de iluminação que simule o nascer do sol pode diminuir esses efeitos. Dado isso, o projeto visa a elaboração de um despertador constituído um sistema de iluminação e um sistema de som, cujo objetivo é realizar a simulação de uma iluminação de um amanhecer natural para o usuário antes e no momento do despertar. O sistema, como um todo, foi implementado através de um *system-on-a-chip* ESP32. Com a sua *framework* oficial, a ESP-IDF, foi desenvolvido o *back-end*, que utiliza a linguagem de programação C e sistema operacional FreeRTOS. Para o usuário poder programar horários de despertar ao longo da semana, foi desenvolvida uma aplicação *web* utilizando o *framework* Angular, a qual possui o TypeScript como linguagem de programação nativa. Tudo está inserido em uma caixa de madeira com saída difusa para a luz. A iluminação exigiu uma calibração das cores para corrigir o balanço de luminosidade dos LEDs. O despertador implementado reproduz som, ilumina o ambiente com as cores e tempos definidos e permite a configuração. Detalhes como a qualidade do som e aquecimento são pontos de possível melhoria futura.

**Palavras-chave:** Servidor. FreeRTOS. ESP32. ESP-IDF. Angular.

## ABSTRACT

BONGIOLO, Daniel. COELHO, Adil. Amanhecer: Internet connected and application powered alarm clock system that simulates dawn . 2021. 42 f. Trabalho de Conclusão de Curso de graduação – Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

People's busy life can cause lower sleep quality and this brings issues such as sleepiness, unstable mood and low productivity. According to some studies like Virginie Gabel's, exposing an individual to a dawn simulating device can mitigate these effects. Based on it, the project aims to develop an alarm clock made out of a lighting system and a sound system, whose goals are to aid the wake up time with a dawn-simulating light. The systems is built upon the system-on-a-chip ESP32. With its official framework, ESP-IDF, the back-end was developed using the C language and FreeRTOS as an operating system. For setting times, configuring lighting and selecting musics, a web application was developed using the Angular framework, which uses TypeScript as native programming language. All of this is cased in a wood box with a diffuse opening for the light. The lighting scheme demanded calibration for balancing the LEDs color luminosities. The alarm clock reproduces sound, lights up the environment with the set colors and allows for configuration. The sound quality and heat of the system are future improvement points.

**Keywords:** Server. FreeRTOS. ESP32. ESP-IDF. Angular.



## LISTA DE FIGURAS

Figura 1 – Diagrama de blocos do projeto. . . . .	15
Figura 2 – Diferentes sequências de iluminação do estudo de Gabel et al. . . . .	18
Figura 3 – Foto da fonte de tensão chaveada. . . . .	19
Figura 4 – Foto do componente regulador de tensão 7805 5V. . . . .	20
Figura 5 – Circuito do regulador de tensão. . . . .	20
Figura 6 – Foto da fita de LEDs RGB 5050. . . . .	21
Figura 7 – Circuito equivalente dos LEDs. . . . .	21
Figura 8 – Foto de uma câmera modelo T6i da Canon. . . . .	22
Figura 9 – Fotografia e histograma da parede iluminada por um dia nublado. . . . .	22
Figura 10 – Fotografia e histograma da parede iluminada pelos LEDs não calibrados, ligados em branco. . . . .	23
Figura 11 – Fotografia e histograma da parede iluminada pelos LEDs já calibrados, ligados em branco. . . . .	23
Figura 12 – Foto do módulo amplificador Max98357 Classe D I <sup>2</sup> S. . . . .	24
Figura 13 – Foto do alto-falante. . . . .	24
Figura 14 – Foto do diagrama de blocos simplificado do amplificador de som. . . . .	25
Figura 15 – Arquitetura interna do ESP32. . . . .	25
Figura 16 – Organização dos elementos na caixa de madeira com tampa de vidro. . . . .	26
Figura 17 – Tela inicial e de configuração de horário. . . . .	27
Figura 18 – Relógio para seleção de horário. . . . .	28
Figura 19 – Capturas da tela de configuração do aplicativo em ambas configurações disponíveis. . . . .	29
Figura 20 – Captura da tela de configuração do aplicativo com menu de seleção de músicas aberto. . . . .	30
Figura 21 – Diagrama de organização dos componentes do aplicativo. . . . .	30
Figura 22 – Divisão do sistema. . . . .	31
Figura 23 – Sequência de inicialização na tarefa inicial. . . . .	31
Figura 24 – Diagrama da tarefa de controle de tempo. . . . .	33
Figura 25 – Diagrama da tarefa da geração de som. . . . .	34
Figura 26 – Diagrama da tarefa de controle da iluminação. . . . .	37

## LISTA DE TABELAS

Tabela 1 – Resultados de conversão entre temperatura de cor em Kelvin para o sistema RGB com 8 bits cada . . . . .	35
Tabela 2 – Valores de porcentagem de potência para os tempos da sequência de simulação de nascer do Sol . . . . .	35
Tabela 3 – Resultados de luminosidades do artigo e medidas . . . . .	38

## LISTA DE ABREVIATURAS E SIGLAS

SoC	<i>System-on-a-chip</i> , sigla que em português significa "sistema-em-um-chip"
LED	<i>Light-emitting diode</i> , sigla que em português significa "diodo emissor de luz"
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i> , sigla que em português significa "Transistor de Efeito de Campo de Óxido de Metal Semicondutor"
MIT	<i>Massachusetts Institute of Technology</i> , sigla que em português significa "Instituto de Tecnologia de Massachusetts"
IoT	<i>Internet of Things</i> , sigla que em português significa "Internet das Coisas"
DSL	<i>Dawn Simulation Light</i> , sigla que em português significa "Luz de Simulação do Nascer do Sol"
PWM	<i>Pulse Width Modulation</i> , sigla que em português significa "Modulação por Largura de Pulso"
HTTP	<i>Hypertext Transfer Protocol</i> , sigla que em português significa "Protocolo de Transferência de Hipertexto"
CMOS	<i>Complementary Metal Oxide Semiconductor</i> , sigla que em português significa "Metal-óxido Semicondutor Complementar"
RGB	<i>Red, Green and Blue</i> , em português Vermelho, Verde e Azul, referencia o conjunto das três cores utilizadas nos LEDs do projeto
SM	<i>Standard Model</i> , sigla que em português significa "Modelo Padrão"
I <sup>2</sup> S	<i>Inter-IC Sound</i> , sigla que em português significa Som Entre Circuitos Integrados
RTC	<i>Real time clock</i> , sigla que em português significa "Relógio de Tempo Real"
RAM	<i>Random Access Memory</i>
NVS	<i>Non-volatile storage</i> , sigla que em português significa armazenamento não-volátil
DNS	<i>Domain Name Server</i>
SNTP	<i>Simple Network Time Protocol</i> , sigla que em português significa protocolo de horário na rede simples

WPS

*Wi-Fi Protected Setup*, sigla que em português significa "Configuração Protegida de Wi-Fi"

## SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>14</b>
1.1 Sistema luminoso auxiliar	14
1.2 Objetivos	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos	15
1.3 Metodologia Resumida	15
<b>2 – REVISÃO DE LITERATURA</b>	<b>17</b>
2.1 Luz que simula o nascer do Sol	17
2.2 Web	18
<b>3 – METODOLOGIA</b>	<b>19</b>
3.1 Equipamentos e dispositivos	19
3.1.1 Fonte de tensão chaveada de 12V	19
3.1.2 Regulador de tensão LM7805 de 5V	19
3.1.3 Iluminação	19
3.1.3.1 Calibração	21
3.1.4 Som	24
3.1.5 ESP32 DevKit V1	24
3.1.6 Caixa	26
3.2 <i>Software</i>	26
3.2.1 <i>Front-end</i>	27
3.2.2 <i>Back-end</i>	28
3.2.2.1 Inicialização	29
3.2.2.2 Servidor	32
3.2.2.3 Horário de sistema	32
3.2.2.4 Controle de tempo	33
3.2.2.5 Geração de sinal I <sup>2</sup> S	34
3.2.2.6 Controle de animação de iluminação	34
3.2.3 Medição de luminosidade	37
<b>4 – ANÁLISE E DISCUSSÃO DOS RESULTADOS</b>	<b>38</b>
4.1 Funcionamento	38
4.2 Iluminação	38
4.3 Som	39
4.4 Cor	39

<b>5 – CONCLUSÃO</b> . . . . .	<b>40</b>
<b>Referências</b> . . . . .	<b>41</b>

# 1 INTRODUÇÃO

## 1.1 Sistema luminoso auxiliar

O ato de acordar não é uma tarefa fácil, pois é preciso combater o estado de inércia do sono. Para o auxílio dessa atividade, normalmente é usado um despertador, o qual é um utensílio cujo objetivo é emitir algum som ou música, em um horário especificado pelo usuário, para que este desperte. Esse despertador, considerado comum, apesar de ser eficaz, não reflete benefícios para a pessoa acordando (o bem-estar, a vitalidade e a estabilidade do humor do indivíduo não se mantêm em níveis adequados), caso haja alguma privação de sono, como de uma pessoa que durma cerca de 6 horas no dia (GABEL et al., 2013).

Um estudo em particular concluiu que simular o nascer do sol traz diversas vantagens nesses casos. Logo, uma luz que acende aos poucos, chegando até num determinado nível de intensidade com certa cor, ajuda o usuário a acordar mais desperto, alerta, e menos sonolento como no método convencional (GABEL et al., 2013).

Baseando-se nesses estudos, foi idealizado construir um sistema luminoso que simula esse nascer do Sol. Esse sistema desenvolvido possui conectividade Wi-Fi, com a intenção de facilitar a configuração dos horários de despertar durante a semana, e, caso seja desejado, fazer a própria personalização, podendo mudar a potência e cor da luz e até mesmo trocar as músicas. Tudo isso foi possível devido a construção de um servidor *restful* dentro do próprio sistema. A possibilidade de mudança das configurações é fornecida através de uma aplicação *web*, a qual só irá funcionar caso faça uma chamada para o servidor mencionado. As mudanças são feitas através de chamadas ao servidor e as opções de configuração presentes também são recebidos por chamadas.

O controle desse sistema foi feito pelo uso de um *system-on-a-chip* (SoC) o ESP32. A conexão realizada entre ele e a aplicação *web* também é feita pela conectividade Wi-Fi.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Elaborar um dispositivo de iluminação com servidor *restful* que auxilie o usuário ao despertar. O sistema deve ser capaz de começar a acender LEDs 30 minutos antes do horário programado pelo usuário e que, de tempos em tempos específicos, aumenta a intensidade da luz. Além disso, será concebida uma interface com o usuário, a partir de uma aplicação *web*, onde se tem acesso à todas as configurações, como programar o horário de despertar para qualquer dia da semana, alterar intensidade e cor da luz e/ou escolher alguma música predefinida.

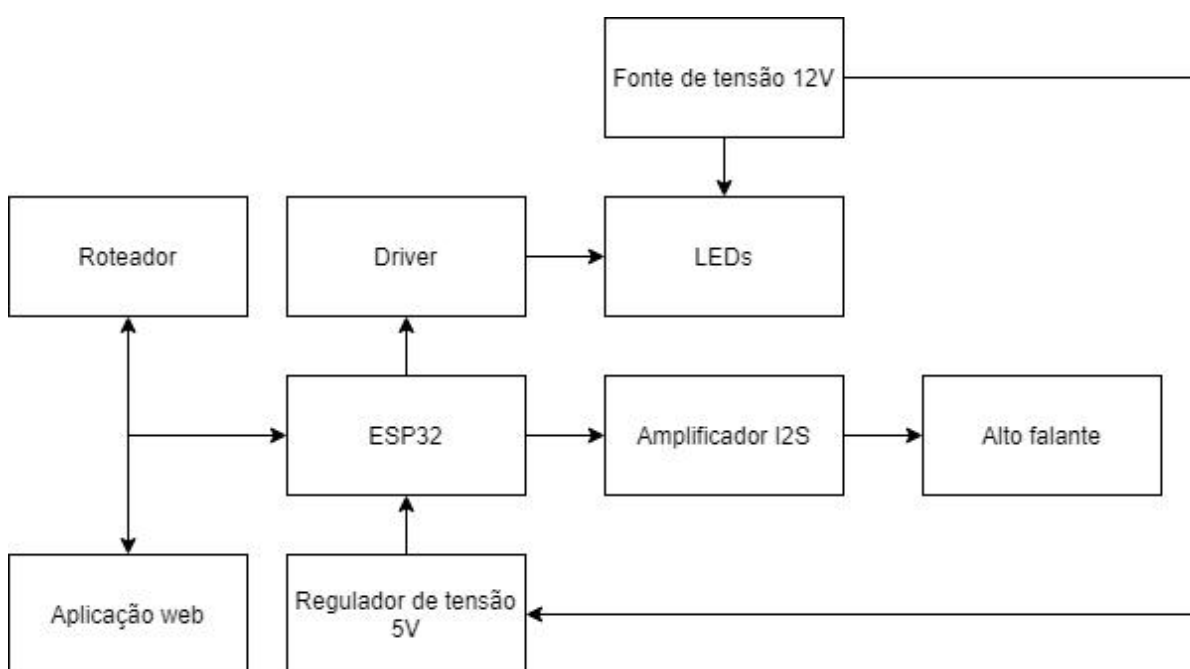
### 1.2.2 Objetivos Específicos

- Elaboração de um código para o controle da iluminação e músicas,
- Conexão do microcontrolador à internet,
- Elaboração de uma aplicação *web* para o usuário poder definir as configurações desejadas,
- Elaboração do produto físico (confeção de protótipo).

### 1.3 Metodologia Resumida

A Figura 1 apresenta o diagrama em blocos do projeto.

Figura 1 – Diagrama de blocos do projeto.



Fonte: Autoria própria.

O SoC ESP32 possui periféricos de conexão Wi-Fi, de transmissão de sinal de som pelo protocolo *Inter-IC Sound* (I<sup>2</sup>S), *Real Time Clock* (RTC) e de geração de sinal de *Pulse-Width Modulation* (PWM). Esses periféricos são a principal razão para sua escolha. Sua alimentação, aqui, é realizada através de um regulador de tensão de 5V, conforme indicado na Figura 1. A fonte do sistema, de 12V com potência de 60W, alimenta também uma fita de LEDs, cujo brilho é controlado por um *driver* de transistores MOSFET. Além da iluminação, há a parte sonora, representada por um alto falante que recebe o sinal de um amplificador cuja entrada é fornecida pela saída de I<sup>2</sup>S do ESP32.

Para que o sistema como um todo funcione, há o estabelecimento de conectividade Wi-Fi, proporcionada pelo uso de um roteador, que interage com a placa mandando e recebendo informações através de chamadas. Essas chamadas só são possíveis graças à criação de um servidor *restful*. Além disso, tem a aplicação *web*, usada para definição das configurações do



---

sistema, a qual também tem seu funcionamento baseado na conexão à internet. A interação dessa aplicação com o microcontrolador é feita, tal como no roteador, realizando chamadas ao servidor.

## 2 REVISÃO DE LITERATURA

Nesta parte será apresentada a teoria usada para desenvolvimento desse projeto. Também será realizada uma breve explicação sobre a estrutura de programação.

### 2.1 Luz que simula o nascer do Sol

Uma pesquisa feita e publicada no *The Journal of Biological and Medical Rhythm Research*, demonstrou resultados com três diferentes tipos de luz, as quais tinham o mesmo objetivo em comum: aumentar o bem-estar, vitalidade e estabilidade do humor da pessoa ao acordar (GABEL et al., 2013).

Três sessões foram conduzidas para comparar os três tipos de iluminação. Os candidatos foram orientados a evitar consumir produtos com cafeína, chocolate e álcool nos dias anteriores a cada sessão. Em todas, o tempo de sono do sujeito foi de 6 horas, para realizar o teste com privação de sono.

Os três tipos de iluminação eram a *Dim Light* (DL), o que significa Luz Fraca. Consistia na exposição do sujeito a uma luminosidade de até 8 lux por 2 horas após a hora de acordar; *Blue Light* (BL), o que significa Luz Azul. Consistia em expor o sujeito à luminosidade de uma lâmpada azul (470 nm) e 100 lux por 20 minutos a 50 cm de distância. Na sessão de BL, era aplicada após as 2 horas de DL; e *Dawn-simulating Light* (DsL), o que significa Luz de Simulação do Nascer do Sol. Esta consistia em uma sequência de iluminação anterior ao despertar do usuário, seguido de 20 minutos da luminosidade máxima do esquema. A sequência de passos de iluminação de DsL, com luminosidade medida a uma distância de 45 cm da fonte de luz é:

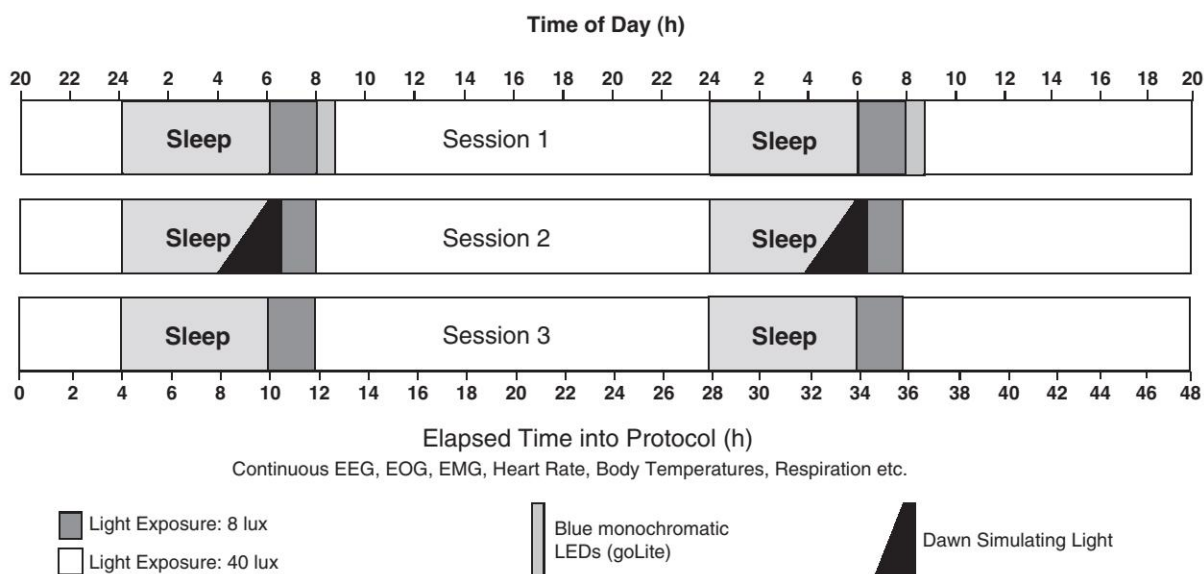
- 30 min antes do despertar: 0 lux;
- 25 min antes do despertar: 1,2 lux, 1090K;
- 15 min antes do despertar: 13 lux, 1500K;
- 6 min antes do despertar: 78 lux, 2200K;
- no momento do despertar: 250 lux, 2750K;
- 20 min seguintes do despertar: 250 lux, 2750K.

Após esses 20 minutos, os sujeitos voltavam a ser iluminados pela DL até o fim das 2 horas pós-despertar.

Todos os esquemas de iluminação podem ser observados na Figura 2.

Após todos os testes, constatou-se que a luz que obteve os melhores resultados subjetivos (de acordo com cada sujeito) com relação ao bem-estar, de tirar o usuário da inércia do sono e de mantê-lo mais alerta, foi a DsL. Logo, baseando-se nessa pesquisa, foi escolhida essa sequência de iluminação para a implementação do projeto.

Figura 2 – Diferentes seqüências de iluminação do estudo de Gabel et al.



Fonte: (GABEL et al., 2013).

## 2.2 Web

Para o usuário poder acessar as configurações do sistema e configurá-las conforme necessidade, foi desenvolvida uma aplicação *web*, servindo como interface. A implementação dessa aplicação foi implementada usando os programas: *Visual Studio Code* como editor de texto, o *framework Angular* para o *front-end* e o *framework* de desenvolvimento ESP-IDF para o SoC ESP32, sendo usado para o *back-end*.

### 3 METODOLOGIA

Neste capítulo são abordadas as diversas etapas de desenvolvimento do projeto.

#### 3.1 Equipamentos e dispositivos

##### 3.1.1 Fonte de tensão chaveada de 12V

A fonte de tensão possui entrada bivolt, potência de 60W e sua tensão de saída é de 12V. Ela é tanto utilizada para a alimentação da fita de LEDs, quanto para alimentar o regulador de tensão de 5V. A figura 3 ilustra a foto do equipamento.

Figura 3 – Foto da fonte de tensão chaveada.



Fonte: Manual do equipamento (Mean Well, 2012).

##### 3.1.2 Regulador de tensão LM7805 de 5V

Para a alimentação do SoC ESP32, foi utilizado um regulador de tensão LM7805 de 5V, o qual recebe um valor de tensão na entrada de 12V, fornecida pela fonte de tensão chaveada. A Figura 4 apresenta uma foto do componente.

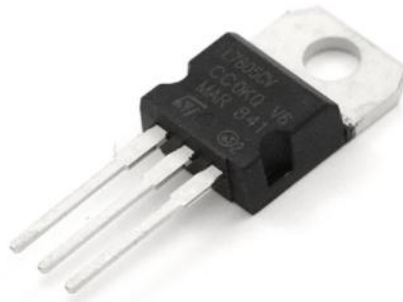
A garantia de estabilidade e a supressão de oscilação do regulador de tensão são concretizadas com a inserção de capacitores na entrada e saída do componente, cujos valores são de 68 $\mu$ F e 10 $\mu$ F, respectivamente. Este circuito está sendo mostrado na Figura 5.

##### 3.1.3 Iluminação

A iluminação é toda feita a partir de pedaços de uma fita de LEDs RGB 5050, alimentados nos 12V fornecidos pela fonte de alimentação. Uma fita semelhante está ilustrada na Figura 6.

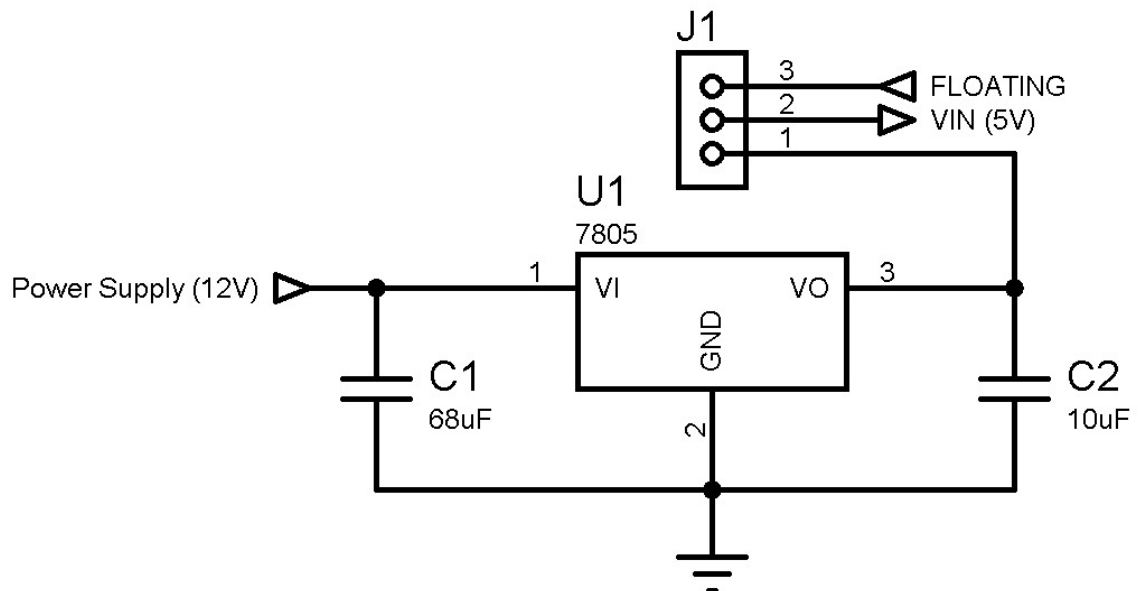
Suas intensidades são controladas por um sinal PWM, que permite diminuir o brilho do LED sem alterar a tensão de alimentação. Aqui, como existem vários módulos em série de

Figura 4 – Foto do componente regulador de tensão 7805 5V.



Fonte: (Filipeflop, 2021).

Figura 5 – Circuito do regulador de tensão.



Fonte: Autoria própria.

três LEDs em paralelo, o controle deve ser realizado por três chaves distintas, a fim de permitir a mudança individual de intensidade para cada cor. A intensidade de cada uma é definida pelo código RGB e por um multiplicador de valor máximo, discutido, mais à frente, no Capítulo 3.1.3.1.

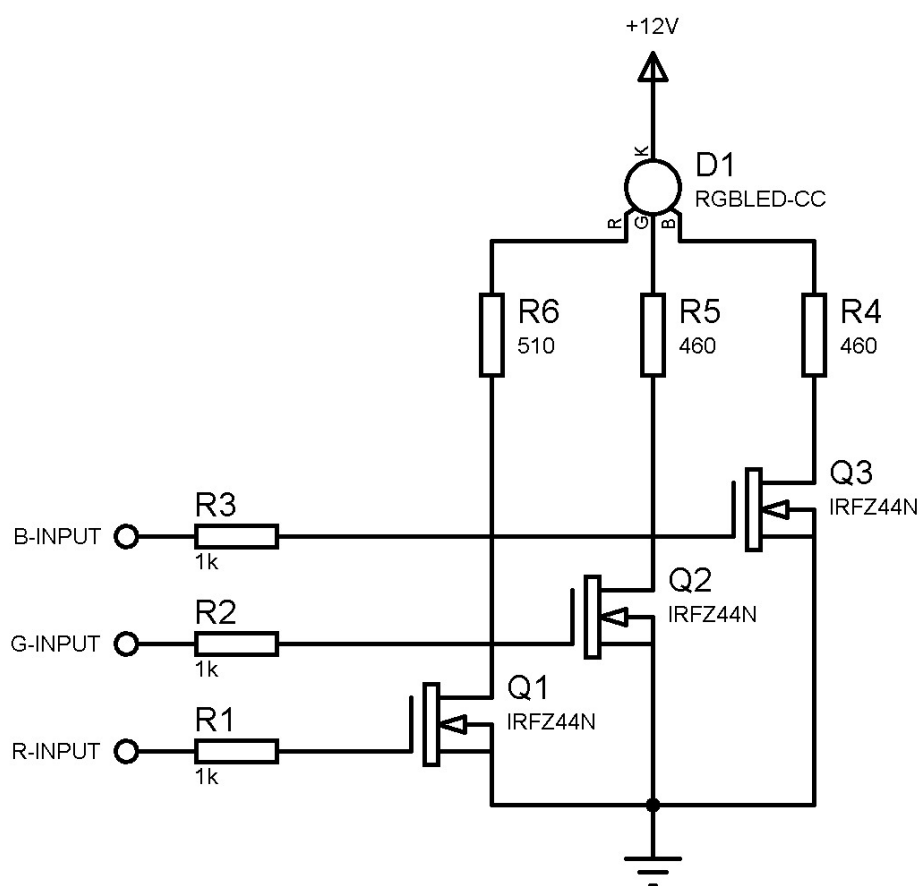
O chaveamento dos LEDs é realizado com transistores MOSFET IRFZ44N. A escolha de MOSFET, ao invés de transistores de junção, se deve ao fato de não haver uma queda significativa de tensão entre os terminais *Drain* e *Source* do transistor, por conta de sua baixíssima resistência equivalente de dreno em saturação, fazendo com que os 12V sejam fornecidos de maneira quase integral aos LEDs (International Rectifier, 2010). O circuito utilizado está na Figura 7.

Figura 6 – Foto da fita de LEDs RGB 5050.



Fonte: (Mercado Livre, 2021).

Figura 7 – Circuito equivalente dos LEDs.



Fonte: Autoria própria.

### 3.1.3.1 Calibração

Os LEDs RGB escolhidos, quando ligados para gerar a cor branca, isto é, com todas cores no seu valor máximo (100%), possuem coloração com tendência para o ciano. Essa

tendência indica um desvio das cores azul e verde com relação à cor vermelha (MOLLICA, 2013).

Para demonstrar e realizar a tentativa de compensar tal desvio, o sensor de pixel ativo (APS, do inglês *active-pixel sensor*) RGB de uma câmera fotográfica foi utilizado para a captura de imagens e breve análise do histograma de cada cor. A câmera é do modelo T6i da fabricante japonesa Canon, a Figura 8 ilustra o modelo.

Figura 8 – Foto de uma câmera modelo T6i da Canon.



Fonte: (Canon, 2021).

O processo escolhido foi o de fotografar uma parede branca iluminada somente pela janela, em um dia nublado, com balanço de branco configurado como “Luz de dia”. A imagem resultante e seu histograma estão na Figura 9. Em outro horário, em que a luz da janela não fosse significativa para a captura, porém com a mesma configuração de balanço de branco, outra fotografia fora registrada da mesma parede, com os LEDs já sofrendo difusão pelo papel manteiga e acesos em branco (todas cores em 100%). A imagem resultante e seu respectivo histograma estão na Figura 10.

Figura 9 – Fotografia e histograma da parede iluminada por um dia nublado.



Fonte: Autoria própria.

Percebe-se que, na captura iluminada naturalmente, os níveis de vermelho, verde e de azul são praticamente paralelos. Isso significa que a imagem, com esta configuração de

Figura 10 – Fotografia e histograma da parede iluminada pelos LEDs não calibrados, ligados em branco.



Fonte: Autoria própria.

balanço de branco, é inteiramente formada por tons de cinza (sem coloração evidente). No caso da captura iluminada pelos LEDs não calibrados, percebe-se que as cores verde e azul acendem em intensidades maiores com a mesma corrente e *duty-cycle* do PWM aplicados na cor vermelha, resultando na cor tendendo para o ciano.

Para realizar a calibração, valores escalares  $\gamma_R$ ,  $\gamma_G$  e  $\gamma_B$  foram multiplicados nos termos do vetor de cores original  $(R_o, G_o, B_o)$  que resultam no vetor de valores a serem enviados ao PWM  $(R_f, G_f, B_f)$ , conforme a Equação 1.

$$(R_f, G_f, B_f) = (\gamma_R R_o, \gamma_G G_o, \gamma_B B_o). \quad (1)$$

Esses escalares deverão ser tais que geram o histograma com a característica mais semelhante possível daquele obtido na captura da parede iluminada pelo dia nublado. Como a câmera fotográfica não fornece valores numéricos no histograma de cada cor, a diminuição dos valores máximos foi realizada de maneira empírica, baseada na comparação visual. Os resultados foram os valores  $\gamma_R = 1$ ,  $\gamma_G = 0,5$ ,  $\gamma_B = 0,18$ . Outra fotografia fora então registrada, cujo histograma e a própria foto podem ser visualizados na Figura 11.

Figura 11 – Fotografia e histograma da parede iluminada pelos LEDs já calibrados, ligados em branco.



Fonte: Autoria própria.



### 3.1.4 Som

O sistema de som é composto por dois elementos: um alto-falante e um amplificador I<sup>2</sup>S. O SoC ESP32 possui dois periféricos I<sup>2</sup>S, que constituem o protocolo de comunicação síncrono e serial, para a transmissão de dados de áudio digital (Espressif Systems, 2021c). Esses dados são enviados para a entrada I<sup>2</sup>S de um módulo amplificador classe D. Através dele, podem ser feitas alterações no ganho, por meio de um pino de I/O. Os ganhos passíveis de serem realizados são de 3dB, 6dB, 9dB, 12dB ou 15dB (Maxim Integrated, 2018). No projeto foi ajustado para um ganho de 9 dB. Na saída, há a conversão desse áudio em um sinal de saída analógico e a amplificação do mesmo, para depois ser conectado diretamente a um alto-falante. As Figura 12 e Figura 13 estão representando o amplificador e o alto-falante utilizados, respectivamente. Além disso, na 14 é mostrado o diagrama de blocos simplificado do amplificador, com a saída já conectada à um alto-falante.

Figura 12 – Foto do módulo amplificador Max98357 Classe D I<sup>2</sup>S.



Fonte: (Shopee, 2021).

Figura 13 – Foto do alto-falante.

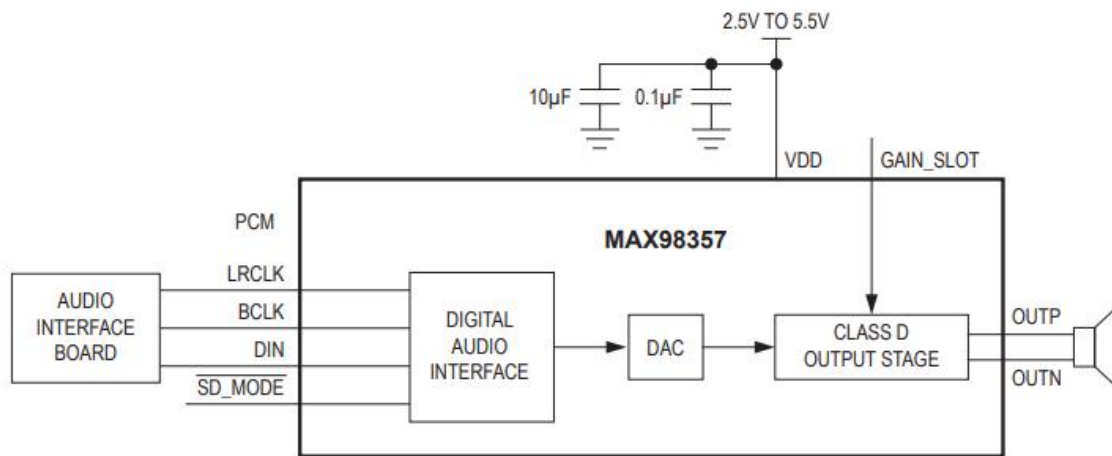


Fonte: (Dallee Electroniccs, 2021).

### 3.1.5 ESP32 DevKit V1

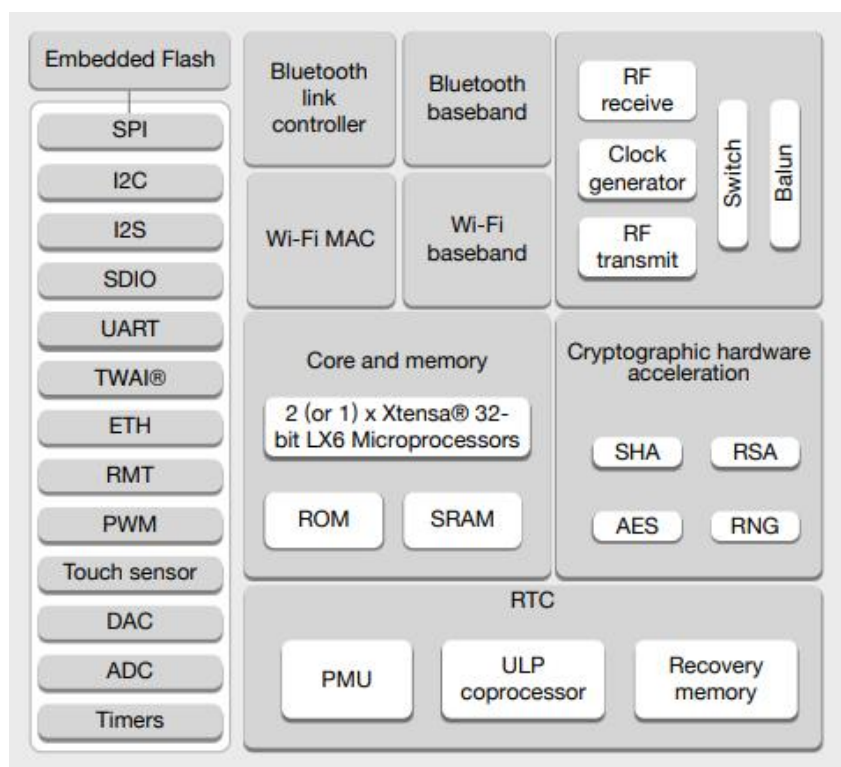
O ESP32 é um SoC que possui 32 bits, 11 pinos de GPIO de uso geral, memória RAM de 520 kB e memória *flash* de 4 MB, configuração de *clock* de até 240 MHz, tensão de operação entre 4,5 V e 9 V e um RTC interno. Além disso, há vários módulos que podem

Figura 14 – Foto do diagrama de blocos simplificado do amplificador de som.



Fonte: Manual do componente (Maxim Integrated, 2018).

Figura 15 – Arquitetura interna do ESP32.



Fonte: Manual do componente (Espressif Systems, 2021).

ser trabalhados, como a conexão Wi-fi, Bluetooth, conversor analógico digital, etc. (Espressif Systems, 2021). A arquitetura interna do mesmo está sendo mostrada na Figura 15.

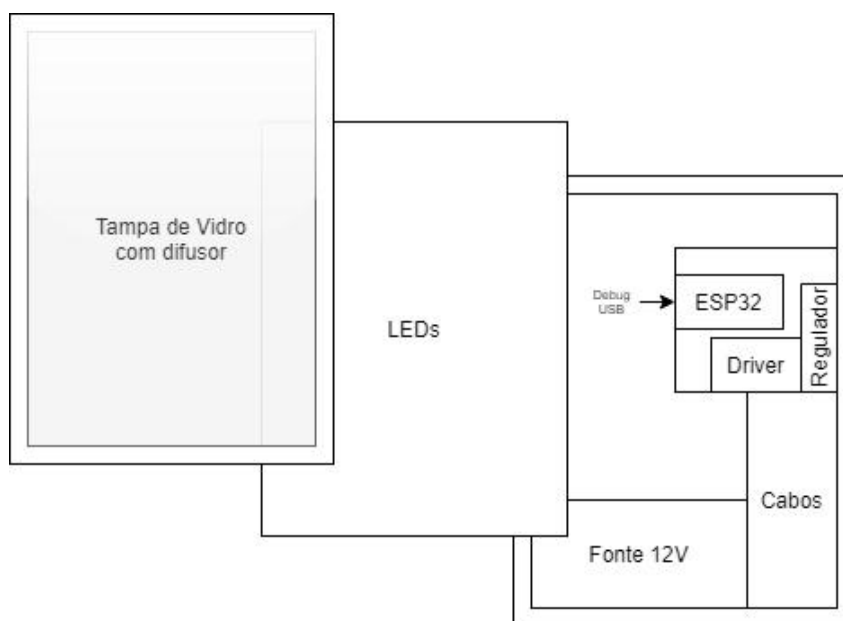
Os motivos da escolha desse SoC se devem ao fato dele possuir conectividade Wi-Fi, permitindo assim o acesso à rede local; ter um bloco de controle de luminosidade de LEDs, com a finalidade de simular o despertar natural; dispor de memória *flash* suficiente, para que

não fosse necessária a utilização de leitor de SD, nem de um cartão SD; a existência de um RTC interno, o qual é usado para sempre manter a data atualizada e realizar comparações com o horário atual; haver um módulo de I2S automático, o qual é empregado na parte de som com a intenção de realizar a transferência de dados de áudio e por último, mesmo tendo todos esses recursos disponíveis, o preço ainda é mais acessível do que outras alternativas disponíveis.

### 3.1.6 Caixa

Toda a estrutura de *hardware* foi organizada dentro de uma caixa de madeira. Sua parte frontal contém um vidro, para a passagem de luz proveniente da fita de LEDs, localizada internamente. Como difusor, foi usado um corte de papel manteiga. Ademais, o SoC foi parafusado para evitar mal contato ou até mesmo de se mover. A disposição dos elementos dentro dessa caixa está mostrado na Figura 16.

Figura 16 – Organização dos elementos na caixa de madeira com tampa de vidro.



Fonte: Autoria própria.

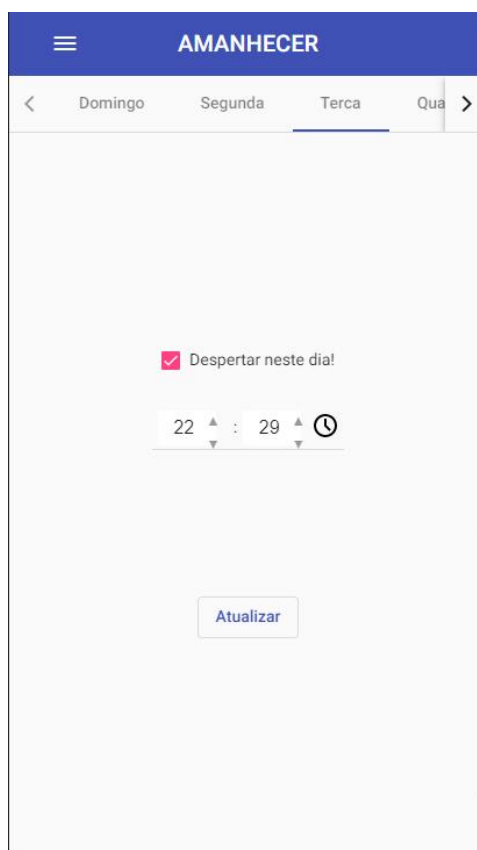
## 3.2 Software

O despertador conta com duas áreas de *software* principais, o *front-end*, responsável pela interação entre humano e máquina e o *back-end*, responsável pelo fornecimento dos arquivos do *front-end* para o navegador e pelo controle do sistema como um todo.

### 3.2.1 Front-end

O *front-end* do Amanhecer é uma aplicação *web* cuja função é a de promover a interação entre o usuário e o produto utilizando um computador ou celular através da conexão *Wi-Fi*. Este permite a configuração de horários e ativação do despertar em todos os dias da semana. As Figuras 17 e 18 tratam de capturas de tela da configuração de horário mencionada.

Figura 17 – Tela inicial e de configuração de horário.



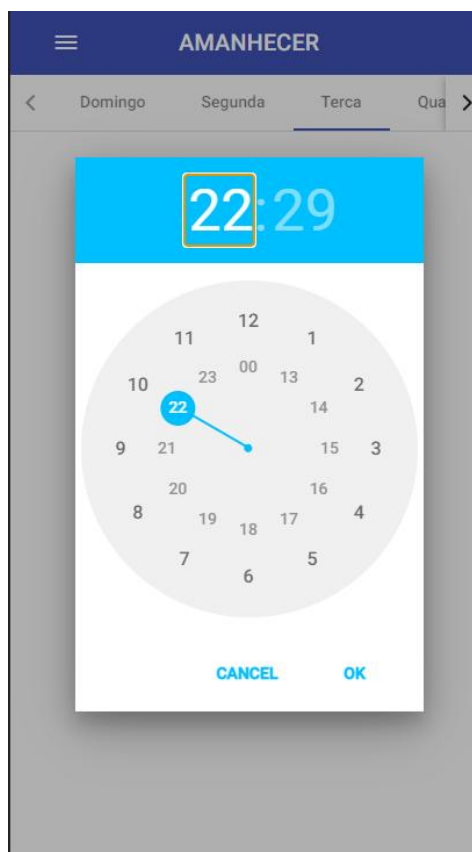
Fonte: Autoria própria.

Outra função do aplicativo acompanhante é a de configuração da animação de iluminação e de, se alguma, qual música utilizada no alarme. A iluminação pode ser escolhida entre a luz de simulação de amanhecer, com tempo e cores pré-definidos, ou uma cor personalizada, em um tempo e potência também configuráveis em *sliders*. A Figura 19 traz a tela nas configurações “Amanhecer” e “Personalizado” de iluminação.

As músicas disponíveis foram obtidas do repositório do *You Tube*, são livres de *royalties* (Google LLC, 2021c) e são nomeadas conforme mostrado na Figura 20, cuja seleção permite também deixar sem música alguma.

Para o aplicativo, foi utilizada a *framework* Angular, que se baseia em componentes e serviços (Google LLC, 2021a). Os componentes são os menores pedaços de uma aplicação Angular e podem ser conectados através de seus *templates*, que são as descrições da parte visual, escritos em HTML e SCSS. Suas partes lógicas são escritas na linguagem TypeScript. Os

Figura 18 – Relógio para seleção de horário.



Fonte: Autoria própria.

serviços são objetos que, na organização do Angular, são usados para gerenciar a comunicação entre componentes ou entre componentes e APIs, este último usado no caso do Amanhecer. Dois serviços de comunicação com a API do ESP32 são utilizados para controle de horário (*TimeService*) e para controle de configuração (*SettingsService*).

Componentes de uma biblioteca chamada Angular-Material foram utilizados como auxílio no desenvolvimento de uma interface responsiva e com aparência de aplicativo nativo. Esta é baseada inteiramente no guia de design gráfico para componentes chamado Material.

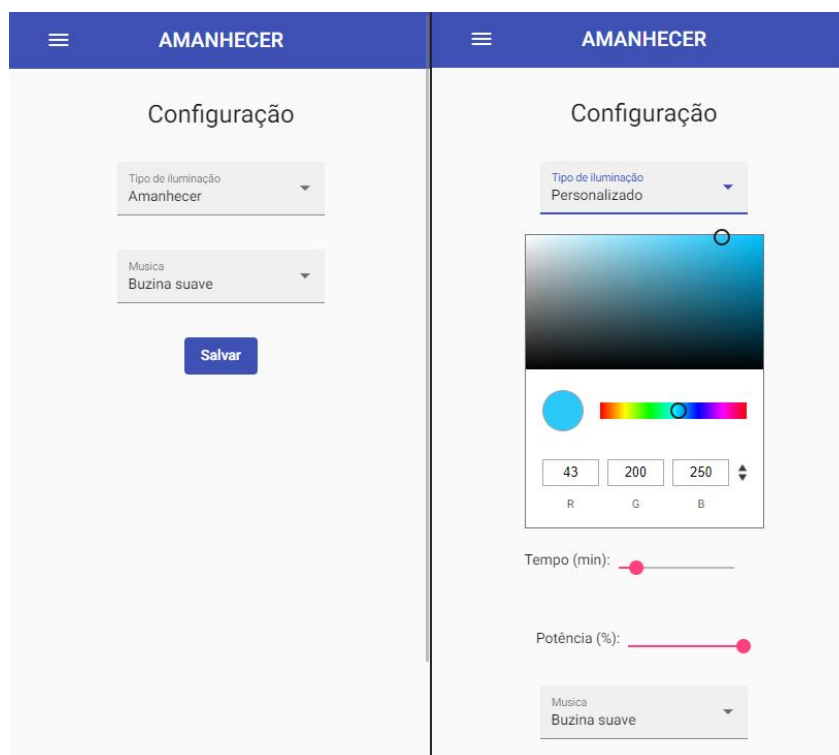
Todo o software da *framework* Angular e dos componentes fornecidos pela sua biblioteca Angular-Material foram produzidos e são mantidos pela companhia Google, que os disponibiliza na licença de software do MIT, requerendo somente que o texto de *Copyright* seja inserido em partes significativas do código (Google LLC, 2021b).

Um diagrama da organização dos componentes e serviços presentes no aplicativo está presente na Figura 21.

### 3.2.2 Back-end

O *back-end* fica inteiramente no SoC escolhido. Nele, o sistema operacional fica responsável pelas tarefas de gerenciamento da conexão *Wi-Fi*, gerenciamento da sincronia

Figura 19 – Capturas da tela de configuração do aplicativo em ambas configurações disponíveis.



Fonte: Autoria própria.

do tempo, comparação de tempo atual, gerenciamento das chamadas HTTP para o servidor, controle da animação da iluminação dos LEDs e geração de sinal I<sup>2</sup>S para o dispositivo que gera a saída de som. Assim, o sistema pode ser dividido e organizado conforme o diagrama da Figura 22.

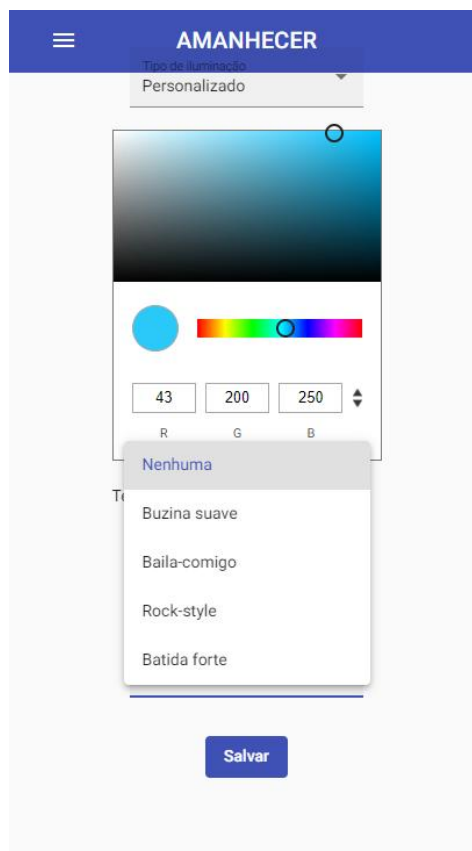
### 3.2.2.1 Inicialização

O sistema operacional inicia sua operação com uma sequência de inicialização definida para configuração dos módulos de operação e das tarefas e filas necessárias para o seu funcionamento. A Figura 23 mostra a tarefa inicial e quais os passos de inicialização.

O primeiro passo foca na inicialização de módulos que permitirão a posterior inicialização do Wi-Fi. Para tal, são inicializados os módulos de NVS, para armazenamento de pares valor-chave (Espressif Systems, 2021d) na memória *flash*, e de NETIF, que é uma camada de abstração utilizada pelo ESP-IDF em cima do protocolo TCP/IP. Este último fornece também APIs que são completamente *thread-safe*. Isto é, garantem a segurança de todas as tarefas que as utilizam (Espressif Systems, 2021a).

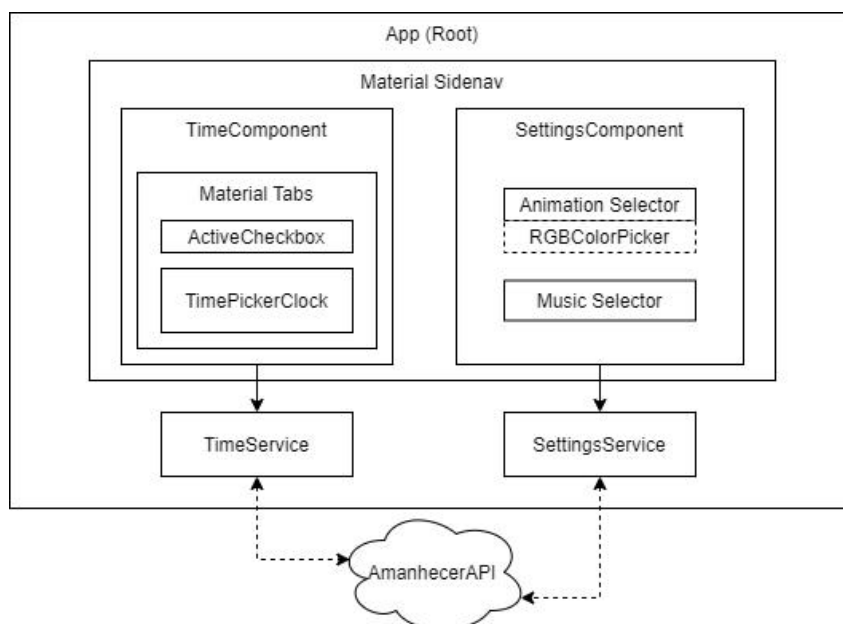
Após, os módulos de sincronia e controle de horário são ativados. Neste passo é configurado o fuso-horário do Brasil e a tarefa de comparação de horário é iniciada, apesar de não realizar trabalho por conta do horário ainda não estar sincronizado. Esta sincronia deve

Figura 20 – Captura da tela de configuração do aplicativo com menu de seleção de músicas aberto.



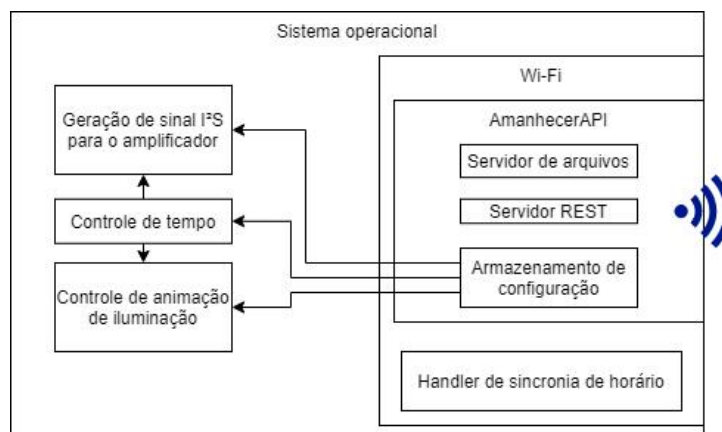
Fonte: Autoria própria.

Figura 21 – Diagrama de organização dos componentes do aplicativo.



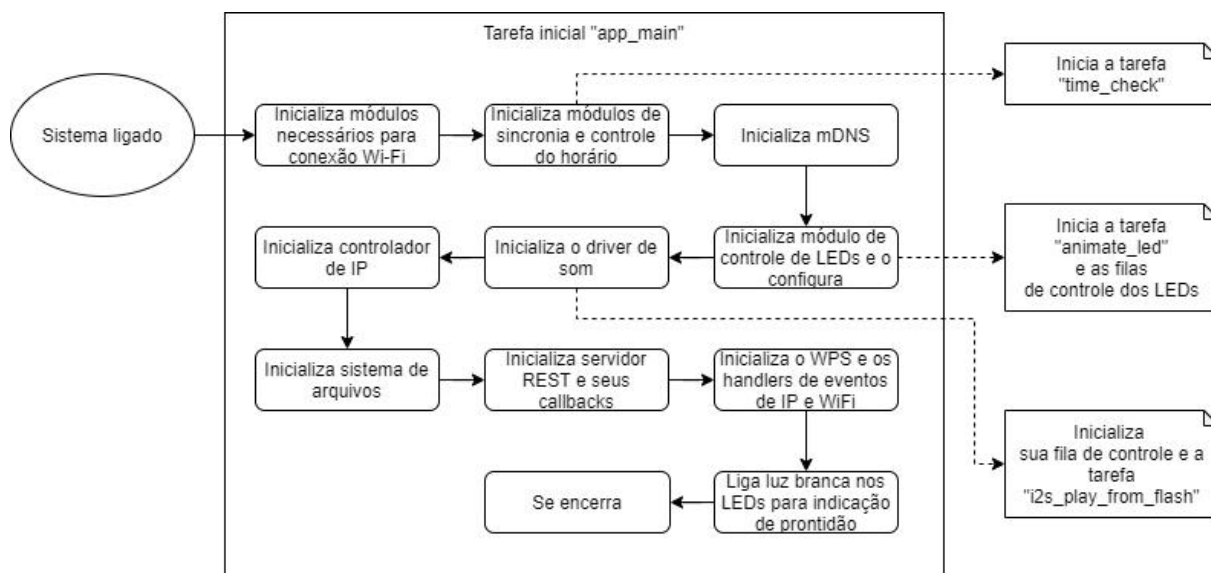
Fonte: Autoria própria.

Figura 22 – Divisão do sistema.



Fonte: Autoria própria.

Figura 23 – Sequência de inicialização na tarefa inicial.



Fonte: Autoria própria.

ocorrer no momento em que o SoC obtiver um endereço IP.

Então, o módulo de mDNS é inicializado. Ele serve para que, na rede local, o usuário não precise saber o IP do dispositivo para se conectar a ele e acessar ao aplicativo do *front-end*. O mDNS responde como se fosse um servidor DNS local para os nomes configurados. Neste caso, ele é configurado com o nome “amanhecer”, em referência ao produto. Assim, o usuário pode digitar “amanhecer” na barra de endereços do navegador preferido para obter conexão com o dispositivo, após conectado à rede Wi-Fi corretamente.

O módulo de controle dos LEDs é então configurado com a inicialização do módulo de *hardware* LEDC, que gera os três sinais PWM desejados, e com a inicialização da tarefa de animação dos LEDs, que também não realiza trabalho por esperar uma mensagem na fila de animação dos LEDs. Esta, quando ativa, controlará a intensidade do brilho dos LEDs conforme



a animação configurada.

Depois disso vem a inicialização do *driver* de som. Onde são ativados o módulo de *hardware* do I<sup>2</sup>S do ESP 32, a fila de controle de música e a tarefa que irá gerar os dados para o módulo do I<sup>2</sup>S enviar ao amplificador para reprodução da música escolhida.

Restam então a inicialização do sistema de arquivos, que permitirá o acesso de leitura e escrita à arquivos armazenados na memória *flash*, permitindo o acesso posterior do servidor de arquivos para fornecer o aplicativo ao requerente; e do *Wi-Fi Protected Setup* (WPS), tecnologia que facilita a conexão Wi-Fi, dispensando o uso de busca por SSIDs e digitação de senha, requerendo somente o pressionamento do botão WPS no roteador.

Com o WPS iniciado, o sistema irá acender os LEDs em cor branca para indicar a possibilidade de conexão.

### 3.2.2.2 Servidor

O servidor tem como base os protocolos fornecidos pelo ESP-IDF. Sua inicialização se dá a partir do componente *HTTP Server* disponível no *framework*. Para tal, fornece-se, para cada *endpoint*, isto é, para cada “ponto de conexão” da API:

- seu URI de gatilho;
- qual método HTTP esperado;
- um ponteiro de função para fazer o *handling* da requisição;
- e um ponteiro para o contexto ao qual será inserido.

Após todos *endpoints* registrados, o servidor pode ser iniciado e uma conexão já pode ser aberta. A API se encarrega de chamar as funções de *handling* quando um URI conhecido é ativado (Espressif Systems, 2021b).

O Amanhecer tem três principais URIs: de horário “amanhecer/api/v1/times”, de configuração “amanhecer/api/v1/config” e de acesso ao sistema de arquivos “amanhecer/\*”. O primeiro dá a possibilidade de ler quais horários estão setados atualmente através do método GET e, quando recebe um POST, altera os horários e ativação diária do despertador. O de configuração, muito semelhantemente, permite que o *caller* de um GET obtenha as configurações atuais e mude as configurações através do método POST. O acesso ao sistema de arquivos, por outro lado, só responde ao método GET e fornece o arquivo requisitado, caso exista. Esse último nos dá a possibilidade de fornecer ao usuário atrás de um computador ou celular o aplicativo, que fica salvo na memória *flash*.

### 3.2.2.3 Horário de sistema

Como discutido no Capítulo 3.2.2.1, a inicialização do sistema também conta com uma operação de sincronia de horário com um serviço *online* que fornece, através do chamado *Simple Network Time Protocol* (SNTP), a data e hora atuais com precisão pouco acima dos milissegundos (USSOLI; PRYTZ, 2013). O software do ESP-IDF permite que esse horário seja

então alimentado no sistema operacional, para uso com funções padrão do FreeRTOS e com a contagem do tempo feita pelo RTC interno do ESP32 (Espressif Systems, 2021e).

### 3.2.2.4 Controle de tempo

Como abordado no Capítulo 3.2.2.3, o ESP-IDF alimenta o sistema operacional com o horário obtido e diversas funções de horário de sistema do FreeRTOS ficam disponíveis para uso. Logo, todo o controle de tempo foi realizado aproveitando-se das mesmas.

A tarefa de controle inicia pela obtenção do horário atual, incluindo também a data. Inicialmente, para que seja possível de conseguir o horário atual com o fuso horário de São Paulo, deve ser feita uma modificação no fuso horário do sistema.

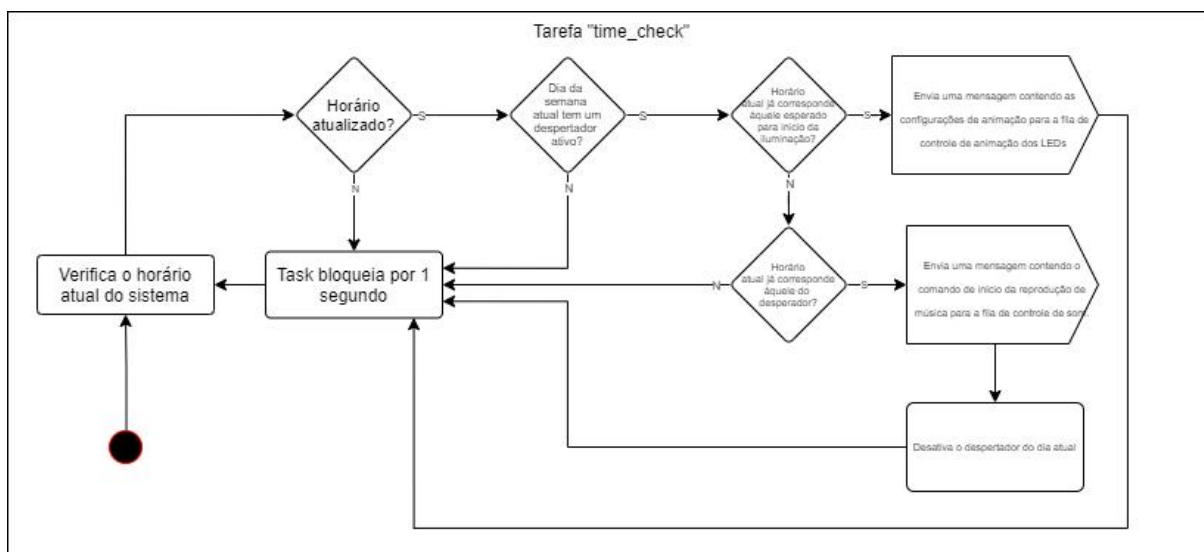
Após, há a verificação de sincronia do horário, através do processo descrito no Capítulo 3.2.2.3. Caso negativo, a tarefa bloqueia por 1 segundo e faz uma nova tentativa na busca do tempo atual até que a sincronia seja realizada.

Quando o horário estiver sincronizado, pega-se o dia da semana atual e é analisado se nesse dia tem ou não um despertador ativo. Caso positivo, é realizada uma verificação de segundo em segundo do horário atual em relação ao horário esperado para o início da iluminação. Quando chega esse momento, é enviada uma mensagem para a fila de controle de animação de LEDs com todas as configurações de animação.

O mesmo é realizado para o horário previsto para o horário definido para despertar, quando inicia-se a música configurada. Então, quando chega o momento, envia-se uma mensagem para a fila de controle de som, notificando-a para começar a tocar a música.

Após isso, é desativado o despertador do dia atual e recomeça o processo. O diagrama dessa tarefa pode ser visualizado na Figura 24

Figura 24 – Diagrama da tarefa de controle de tempo.



Fonte: Autoria própria.

### 3.2.2.5 Geração de sinal I<sup>2</sup>S

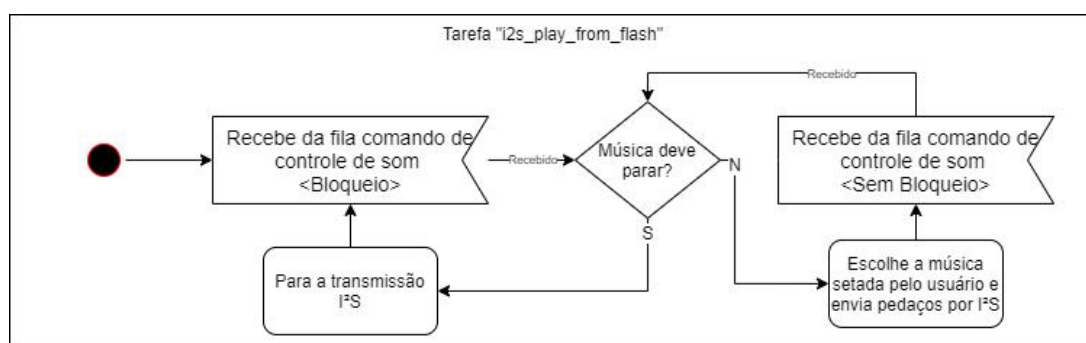
Como já dito anteriormente na seção de *hardware*, o ESP32 possui dois periféricos I<sup>2</sup>S responsáveis pela geração e transmissão de sinal I<sup>2</sup>S, sendo que este sinal é captado na entrada do amplificador.

Essa tarefa se baseia, primordialmente, na existência de uma fila, que recebe um comando de controle de som (*shouldStart*) para que a música possa começar a tocar. A música tocada em questão, é a escolhida pelo usuário na hora de fazer a configuração do sistema.

O envio dos dados da música é executado em etapas, ou seja, são mandados trechos do arquivo por I<sup>2</sup>S. Esses trechos são, primeiramente, convertidos de 8 bits para 16 bits. Depois, são armazenados em um *buffer* até chegar ao limite do tamanho do arquivo da música. Após isso, recomeça o processo e continua a ser armazenado nesse mesmo *buffer*. Caso a pessoa opte por parar a música, onde é efetivado pressionando um botão presente na caixa, o comando *shouldStop* é enviado resultando na parada completa da mesma. Todo esse processo está ilustrado na Figura 25.

O arquivo "songs\_file.h" contém todos os arquivos de músicas disponíveis para escolha do usuário. Todas estão em 8 bits para economia do espaço em flash utilizado.

Figura 25 – Diagrama da tarefa da geração de som.



Fonte: Autoria própria.

### 3.2.2.6 Controle de animação de iluminação

Um dos recursos mais importantes para o Amanhecer é o de iluminação. Para obter o efeito de melhora no despertar do usuário, deve-se usar da DsL (GABEL et al., 2013). A fim de obter uma animação de iluminação entre os passos descritos pelo artigo, fez-se um planejamento de transição linear entre cores e intensidades (0 a 100%), chamados aqui de “passos” e citados no Capítulo 2. Os passos definem as cores como valores de temperatura de cor. Para a conversão destes aos valores correspondentes em RGB, há uma tabela que permite converter a cor de um corpo negro em certa temperatura em Kelvin para RGB (limitado no espaço sRGB) (CHARITY, 2021). Esses valores foram usados para adaptar as cores do DsL em RGB para utilização nesse sistema, resultando na Tabela 1. Por proximidade, o valor de 1090 K foi considerado como 1100 K.

Tabela 1 – Resultados de conversão entre temperatura de cor em Kelvin para o sistema RGB com 8 bits cada.

	Temperatura no fim do passo (K)	Vermelho	Verde	Azul
Passo 1	1090	255	71	0
Passo 2	1500	255	109	0
Passo 3	2200	255	147	44
Passo 4	2750	255	169	87

Fonte: Autoria própria

Deste ponto, considerando que a intensidade máxima de 250 lux descrita no Capítulo 2 representa os LEDs acesos em 100%, foram calculados os multiplicadores de potência  $P$  para cada passo, podendo ser vistos na Tabela 2.

Tabela 2 – Valores de porcentagem de potência para os tempos da sequência de simulação de nascer do Sol.

Tempo (min)	Cor (R,G,B)	Intensidade (lux)	P (%)
5	(255, 71,00)	1,2	0,5
15	(255,109,00)	13	5
24	(255,147,44)	78	31
30	(255,169,87)	250	100

Fonte: Autoria própria

Assim, um vetor de cor  $\vec{C}$  é multiplicado pelo escalar de potência  $P$ , como pode ser visto na Equação 2. Lembrando que os valores de calibração do Capítulo 3.1.3.1 são aplicados ao vetor de cores resultante antes do envio ao PWM.

$$\vec{C}_f = \vec{C} \cdot P_{atual} \quad (2)$$

O ESP32 tem, em seu módulo de controle de intensidade de LEDs, um conjunto de funções para deixar o *hardware* criar transições suaves e contínuas entre níveis de PWM, para cada um de seus módulos. Entretanto, conforme testes realizados no desenvolvimento do Amanhecer, foi percebido que esses entre esses módulos existe um atraso de atuação aleatório, podendo causar um efeito indesejado de coloração sem previsibilidade. Um bom exemplo está no segundo passo, que vai de vermelho a um laranja forte. Nesse momento, o PWM cujo LED associado é verde comumente ganha intensidade mais rapidamente que o LED vermelho, deixando a cor muito mais amarelada num primeiro momento e depois retornando ao laranja, no encerramento da transição, o que é indesejado.

Para conter o efeito citado, a transição de iluminação entre dois passos acontece discretamente entre dois níveis de intensidade dos LEDs de cada cor, tudo controlado por *software*, na tarefa de controle de iluminação, chamada de *animate\_led*. A transição discreta é

realizada em iterações cujo tempo de iteração  $t_i$  é resultado da divisão do tempo total do passo  $t_{total}$  por uma constante arbitrária  $d = 30$  que representa o numero de iterações, conforme a Equação 3. O valor de  $t_i$  é usado no atraso da tarefa de transição.

$$t_i = \frac{t_{total}}{d} \quad (3)$$

Para a variação de cor de cada iteração, um vetor de incremento de cor  $\vec{I}$  é calculado a partir da divisão da diferença entre cor destino  $\vec{C}_{destino}$  e cor origem  $\vec{C}_{origem}$  por  $d$ , como mostrado na Equação 4.

$$\vec{I} = (\vec{C}_{destino} - \vec{C}_{origem}) \cdot \frac{1}{d} \quad (4)$$

Assim, o valor da cor atual  $C_{atual}$  em uma dada iteração  $j$  (de 0 a  $D - 1$ ) é mostrado na Equação 5.

$$\vec{C}_{atual} = \vec{C}_{origem} + j \cdot \vec{I} \quad (5)$$

O cálculo da potência  $P$  a ser aplicada ao vetor de cor em um dada iteração é também linear e muito parecido ao anterior, substituindo os vetores  $\vec{I}$ ,  $\vec{C}_{origem}$ ,  $\vec{C}_{destino}$  e  $\vec{C}_{atual}$  presentes nas equações 4 e 5 para, respectivamente, os valores escalares de incremento de potência  $i$ , potência de origem  $P_{origem}$ , potência de destino  $P_{destino}$  e potência da iteração atual  $P_{atual}$ . A iteração atual continua sendo representada por  $j$ . Essas substituições levam às Equações 6 e 7.

$$i = (P_{destino} - P_{origem}) \cdot \frac{1}{d} \quad (6)$$

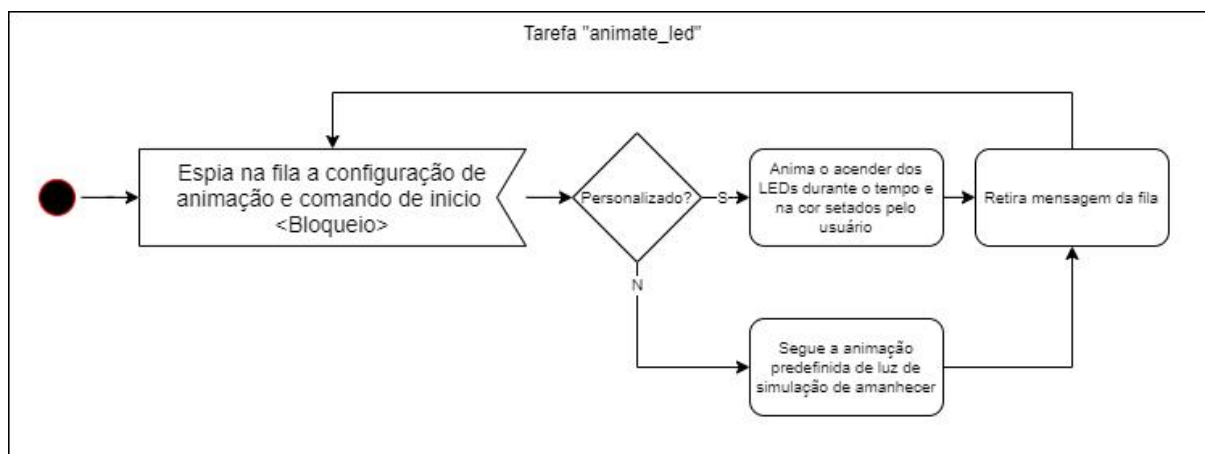
$$P_{atual} = P_{origem} + j \cdot i \quad (7)$$

Para a animação personalizada, é considerada somente uma dessas transições, considerando  $H$  o valor máximo de potência selecionado pelo usuário, de 0 a  $H\%$ , do preto à cor definida pelo usuário no aplicativo *web*. Todo o resto segue o mesmo esquema de transição incremental dividida em 30 períodos.

A tarefa de iluminação inicia a animação quando uma mensagem na fila de animação é recebida. Esta mensagem só é retirada da fila quando a animação se encerra, para evitar que um segundo comando seja considerado depois do início da animação atual.

Um fluxograma da tarefa de controle está ilustrado na Figura 26.

Figura 26 – Diagrama da tarefa de controle da iluminação.



Fonte: Autoria própria.

### 3.2.3 Medição de luminosidade

Foram realizadas medições da luminosidade do equipamento, através de um aplicativo disponível para *Android* e *iOS* chamado *Lux Light Meter Free*. A distância utilizada, para medição com relação ao equipamento, foi de 45 cm, com o intuito de fazer comparações entre as medições feitas e as originais (GABEL et al., 2013).

## 4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

O sistema foi finalizado com uma alteração de *hardware* em relação ao projeto inicial, o qual estabelecia a utilização do microcontrolador Tiva-C como plataforma principal. Porém após uma pesquisa mais aprofundada, avaliando qual opção seria mais viável de se trabalhar e que atendesse a todas as necessidades com um preço baixo, optou-se pelo ESP32.

### 4.1 Funcionamento

O sistema é inicializado, primeiramente, pondo-o na tomada, gerando a alimentação para o ESP32 e à fita de LEDs. Com o sistema ligado, deve-se estabelecer a conexão entre ele e a rede local. Optou-se pelo uso do WPS como recurso para evitar o uso de senhas e realizar uma conexão rápida. Assim, o usuário deve, somente, esperar a luz branca de prontidão do sistema e pressionar o botão WPS no seu roteador.

Após isso, e com o sistema devidamente conectado à rede, pode-se fazer o acesso à aplicação *web* por meio da URL “/amanhecer”. Dentro da aplicação, tem-se as configurações relacionadas à escolha dos dias e horários em que se deseja que o sistema desperte, se deve ou não despertar com alguma música, que pode ser selecionada dentre algumas opções disponíveis, e até mesmo mudar o tipo de iluminação.

### 4.2 Iluminação

A principal escolha do usuário envolve qual tipo de iluminação que se deseja. Tanto a opção Amanhecer, quanto a opção Personalizado (ambas mencionadas na Metodologia), realizam seus objetivos (acender os LEDs nos tempos e cores corretos). Os testes realizados mostraram que deveria haver algumas adaptações em relação aos testes originais mencionados no trabalho de Gabel para obter valores mais similares, porque a fita de LEDs escolhida para iluminação não possui uma intensidade luminosa tão alta quando está exposta a uma distância razoável do usuário. Usando os mesmo tempos estabelecidos no artigo, obtiveram-se os seguintes valores:

Tabela 3 – Resultados de luminosidades do artigo e medidas.

Tempo após o início (min)	Luminosidade original (lux)	Luminosidade Amanhecer(lux)
5	1,2	0,8
15	13	13,06
24	78	51,61
30	250	135

Fonte: Autoria própria

Pode-se notar que quanto maior é o tempo e, conseqüentemente, a intensidade dos LEDs, a discrepância de valores fica maior em relação aos valores encontrados no artigo. Portanto, algumas alternativas para a melhora da solução seriam: a troca por LEDs mais eficientes com a mesma energia aplicada; aumento da densidade de LEDs (por consequência aumentaria também o consumo, o que acabaria tornando uma solução inviável); redução da distância entre o usuário e o despertador, tendo como valor ideal 30,67 cm.

#### 4.3 Som

O dispositivo permite a escolha de 4 músicas diferentes, com características e intensidades sonoras distintas.

A compressão dos arquivos de áudio fez com que a qualidade dos mesmos fosse afetada significativamente, tendo menor nitidez de notas. Contudo, como o sistema não foi construído com essa ênfase, a reprodução de som teve êxito.

#### 4.4 Cor

A cor é um aspecto que poderia ser considerado para melhora em uma possível continuidade ou nova iteração do projeto. Somente com a iluminação RGB convertida dos valores Kelvin, não se pode garantir a mesma composição espectral a que os sujeitos do trabalho de Gabel foram expostos. A utilização de sistemas de iluminação baseados em lâmpadas brancas com diferentes temperaturas de cor poderia ser considerada para melhora desse aspecto (SPEIER; SALSBURY, 2006).



## 5 CONCLUSÃO

Neste trabalho foi concebido um protótipo de sistema de despertador, com conexão à internet, contendo um aplicativo de configuração e que simula o nascer do Sol. A utilização da plataforma *Angular* como aplicação *web* permite que sempre possa haver melhorias no sistema.

O objetivo é o de, utilizando uma sequência de iluminação no horário de despertar, baseado no estudo de GABEL, V. et al, melhorar o bem estar, humor e a produtividade do usuário. Porém, além da funcionalidade padrão seguindo o estudo, fora implementada a opção personalizada, em que o próprio usuário pode escolher livremente a cor desejada e, até mesmo selecionar uma música.

Devido aos LEDs escolhidos, que não possuem a intensidade esperada, a distância recomendada de uso fica reduzida em relação àquela do trabalho de GABEL, V. et al, passando de 45 cm para aproximadamente 31 cm.

Outro ponto a ser levado em consideração é o de aquecimento. Pela utilização de LEDs sem dissipação e de regulador linear para ajuste da tensão, o interior da caixa pode aquecer consideravelmente, se mantido no modo de conexão Wi-Fi e com os LEDs ligados em potência total. Um sistema de arrefecimento seria bem vindo, apesar desse aquecimento não interferir no funcionamento proposto. Isso porque a conexão Wi-Fi inicial que consome mais energia só é realizada uma vez e os LEDs são mantidos normalmente acesos em baixa potência, exceto a poucos minutos do horário de despertar.

Considerados todos os pontos aqui discutidos, o projeto e execução resultaram num produto que atende aquilo a que se propõe, apesar da limitação de luminosidade comentada.

Como continuidade deste projeto, será feita uma análise de mercado e de viabilidade financeira para uma potencial transformação em produto e também melhorias, como otimização do código, melhora na qualidade sonora das músicas, criação de um protótipo mais robusto, uso de LEDs mais potentes ou em uma maior densidade.

## Referências

Canon. **EOS Rebel T6i**. 2021. Disponível em: <<https://www.canon.com.br/produtos/produtos-para-voce---cameras---linha-eos---dslr/eos-rebel-t6i>>. Acesso em: 17 de agosto de 2021. Citado na página 22.

CHARITY, M. **What color is a blackbody?** 2021. Disponível em: <<http://www.vendian.org/mncharity/dir3/blackbody/>>. Acesso em: 12 de agosto de 2021. Citado na página 34.

Dallee Electronics. **Foto do alto-falante**. 2021. Disponível em: <<https://www.dallee.com/product-image/1-1-Round-8-Ohm-Speaker-5-Watt-230>>. Acesso em: 12 de agosto de 2021. Citado na página 24.

Espressif Systems. **ESP-NETIF**. 2021. Disponível em: <[https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/network/esp\\_netif.html](https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/network/esp_netif.html)>. Acesso em: 16 de agosto de 2021. Citado na página 29.

Espressif Systems. **HTTP Server**. 2021. Disponível em: <[https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/protocols/esp\\_http\\_server.html](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/protocols/esp_http_server.html)>. Acesso em: 16 de agosto de 2021. Citado na página 32.

Espressif Systems. **Inter-IC Sound (I<sup>2</sup>S)**. 2021. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/i2s.html>>. Acesso em: 22 de agosto de 2021. Citado na página 24.

Espressif Systems. **Non-volatile storage library**. 2021. Disponível em: <[https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/storage/nvs\\_flash.html](https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/storage/nvs_flash.html)>. Acesso em: 16 de agosto de 2021. Citado na página 29.

Espressif Systems. **System Time**. 2021. Disponível em: <[https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/system/system\\_time.html](https://docs.espressif.com/projects/esp-idf/en/v4.2.1/esp32/api-reference/system/system_time.html)>. Acesso em: 16 de agosto de 2021. Citado na página 33.

Espressif Systems. **ESP32 Series Datasheet**. [S.l.], 2021. 65 p. Citado na página 25.

Filipeflop. **Foto do componente regulador de tensão 7805 5V**. 2021. Disponível em: <<https://uploads.filipeflop.com/2017/07/00107-1.jpg>>. Acesso em: 11 de agosto de 2021. Citado na página 20.

GABEL, V. et al. Effects of artificial dawn and morning blue light on daytime cognitive performance, well-being, cortisol and melatonin levels. p. 11, 2013. Citado 5 vezes nas páginas 14, 17, 18, 34 e 37.

Google LLC. **Introduction to Angular concepts**. 2021. Disponível em: <<https://angular.io/guide/architecture>>. Acesso em: 16 de agosto de 2021. Citado na página 27.

Google LLC. **The MIT License**. 2021. Disponível em: <<https://angular.io/license>>. Acesso em: 16 de agosto de 2021. Citado na página 28.

Google LLC. **Use music and sound effects from the Audio Library**. 2021. Disponível em: <<https://support.google.com/youtube/answer/3376882?hl=en>>. Acesso em: 22 de agosto de 2021. Citado na página 27.

International Rectifier. **Datasheet do transistor MOSFET IRFZ44N**. [S.l.], 2010. 9 p. Disponível em: <<https://www.infineon.com/dgdl/irfz44npbf.pdf?fileId=5546d462533600a40153563b3a9f220d>>. Acesso em: 25 de agosto de 2021. Citado na página 20.

Maxim Integrated. **MAX98357 Development Board Evaluation System (WLP)**. [S.l.], 2018. 9 p. Citado 2 vezes nas páginas 24 e 25.

Mean Well. **Datasheet fonte de tensão S-60**. [S.l.], 2012. 3 p. Citado na página 19.

Mercado Livre. **Foto da fita de LEDs RGB 5050**. 2021. Disponível em: <[https://http2.mlstatic.com/D\\_NQ\\_NP\\_636529-MLB31153799514\\_062019-O.webp](https://http2.mlstatic.com/D_NQ_NP_636529-MLB31153799514_062019-O.webp)>. Acesso em: 11 de agosto de 2021. Citado na página 21.

MOLLICA, P. **Color Theory: An Essential Guide to Color-from Basic Principles to Practical Applications**. 1. ed. [S.l.]: Walter Foster Publishing, 2013. Citado na página 22.

Shopee. **Foto do módulo amplificador Max98357 Classe D I2S**. 2021. Disponível em: <<https://cf.shopee.com.br/file/5d71c922b353b730e0d26028de89e096>>. Acesso em: 12 de agosto de 2021. Citado na página 24.

SPEIER, I.; SALSURY, M. Color temperature tunable white light led system. SPIE, setembro 2006. Disponível em: <<https://doi.org/10.1117/12.679688>>. Citado na página 39.

USSOLI, M.; PRYTZ, G. SNTP time synchronization accuracy measurements. p. 4, 2013. Citado na página 32.