

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

AUGUSTO CARVALHO DE FREITAS
GUILHERME LIBÉRIO DA SOUZA E SILVA
HENRIQUE NELSON DA SILVA VANIN

MLV - MUSIC LIGHT VISUALIZER

OFICINA DE INTEGRAÇÃO 1 – RELATÓRIO FINAL

CURITIBA

2019

AUGUSTO CARVALHO DE FREITAS
GUILHERME LIBÉRIO DA SOUZA E SILVA
HENRIQUE NELSON DA SILVA VANIN

MLV - MUSIC LIGHT VISUALIZER

Relatório Final da disciplina Oficina de Integração 1, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Dr. Gustavo Benvenutti Borba
Prof. Ronnier Frates Rohrich

CURITIBA

2019

AGRADECIMENTOS

A ideia deste projeto nasceu devido a um vídeo do *Youtuber* Devon Crawford. Seu vídeo está referenciado neste documento.

RESUMO

. MLV - MUSIC LIGHT VISUALIZER. 20 f. Oficina de Integração 1 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2019.

Esse documento descreve o processo de concepção, desenvolvimento e integração do projeto de uma fita LED capaz de responder à músicas de forma visual, correlacionando som com luz e cores. Estabelece os componentes necessários para composição do *hardware*, bem como explica o funcionamento do *software* utilizado para atingir o resultado esperado. Detalha os cálculos para o refinamento de dados e o mapeamento de cores baseado nas informações obtidas de uma amostra e determina as ferramentas usadas para controle e criação de algoritmos capazes de controlar os LEDs. Explicita parâmetros logísticos como cronograma, custos, dificuldades e possibilidade de melhorias, além de relatar os resultados obtidos ao final do projeto.

Palavras-chave: Projeto de Integração, Arduino, Fita de LED, *Python*, Som, Cor, Luz, Conversão de dados, Frequência, Intensidade Sonora, Análise de dados

ABSTRACT

. MLV - MUSIC LIGHT VISUALIZER. 20 f. Oficina de Integração 1 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2019.

This document describes the process of conception, development and integration of the project with a goal to make a LED strip capable of responding visually to music, correlating the sound with colored lights. Establishes the necessary components for the hardware composition and explains how the software used to obtain the expected result works. It details the calculations for the refinement of data and color mapping based on the information obtained in a sample and determines the tools used for the control and creation of algorithms capable of controlling the LEDs. The logistic parameters like schedule, cost, difficulties and possible improvements are also present in the document, as well as the final results obtained with the project.

Keywords: Integration Project, Arduino, LED strip, Python, Sound, Color, Light, Data Conversion, Frequency, Sound Intensity, Data Analysis

SUMÁRIO

1	INTRODUÇÃO	7
1.1	MOTIVAÇÃO	8
1.2	OBJETIVOS	8
1.2.1	Objetivo geral	8
1.2.2	Objetivos específicos	8
2	SOFTWARE	9
2.1	PROGRAMA EM <i>PYTHON</i>	9
2.1.1	Transformada Rápida de Fourier	10
2.2	PROGRAMA PARA O ARDUINO	11
2.2.1	Mapeamento de LEDs	13
2.2.1.1	Algoritmos	13
2.2.1.2	Mapeamento de Cor	13
3	HARDWARE	15
4	CRONOGRAMA E CUSTOS DO PROJETO	16
4.1	CRONOGRAMA	16
4.2	CUSTOS	17
5	DIFICULDADES E RESULTADOS	18
6	CONCLUSÕES	19
6.1	CONCLUSÕES	19
6.2	TRABALHOS FUTUROS	19
	REFERÊNCIAS	20

1 INTRODUÇÃO

Este trabalho destina-se a documentar a produção de um tradutor de frequências e intensidades sonoras em luz e cores. Serão descritos de forma objetiva os marcos a serem alcançados ao longo de seu desenvolvimento, bem como aspectos logísticos como cronograma, orçamento, planejamento e divisão de trabalho.

O projeto consiste na construção de um visualizador de frequência e intensidade sonora através de fita de LED RGB. Utilizando um microcontrolador Arduino, foi feito um sistema capaz de associar uma cor específica do espectro de luz à uma frequência e intensidade sonora da música captada pelo microfone do computador.

Por meio de um software desenvolvido em *Python*, utilizando a biblioteca *pyaudio*, o sinal digital da música será captado pelo microfone do computador e extraído sua frequência mais dominante por meio da Transformada Discreta de Fourier (NAKAGOME, 2018) . Posteriormente um programa feito na IDE do Arduino (AASVIK, 2017) irá mapear as diferentes informações recebidas do computador em diferentes tensões.



Figura 1: Diagrama Geral do Projeto

1.1 MOTIVAÇÃO

Desenvolver um projeto audiovisual capaz de expressar música por meio de luz e cores.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Integrar de forma eficaz o hardware e software necessários para atender a correta, e visual, correlação de dados dos sinais digitais provenientes do som para com a resposta da fita led.

1.2.2 OBJETIVOS ESPECÍFICOS

Marcos do Projeto:

- Desenvolver um programa em *Python* responsável por interpretar sinais digitais para abstrair o pico da frequência mais relevante da música em um determinado intervalo de tempo e adquirir o hardware necessário para o projeto.
- Desenvolver um programa na IDE do Arduino para interpretação de frequência e emissão de cor na fita de LED;
- Realizar a integração de hardware e software de forma efetiva e possuir um software capaz de tratar e interpretar dados de forma visualmente perceptível.

2 SOFTWARE

O *software* é composto por um programa em *Python* responsável por extrair os dados de frequência e intensidade sonora e o programa para o Arduino, responsável por receber o *input* de dados, processá-los e emitir o sinal de tensão para a resposta da fita de LED.

2.1 PROGRAMA EM *PYTHON*

O *software* em *Python* utiliza da biblioteca *pyaudio* para armazenar dados de frequência e da biblioteca *audioop* para a intensidade sonora ao longo do tempo, esses dados são obtidos a partir do dispositivo de gravação do computador, sendo possível o uso do *mixagem stereo*, referente ao som do próprio computador, ou um microfone externo, possibilitando o uso de um som exterior ser captado. Contudo, a amostragem de dados por intervalo de tempo pode não ser capaz de representar de forma fidedigna. Por isso foram necessárias funções capazes de condensar essa informação.

Para os dados referentes à intensidade sonora, foi utilizada uma conta simples da raiz da média do quadrados das amostras de volume extraídas num determinado espaço amostral, também conhecido como "*rms*" (*root mean square*). Para este cálculo foi utilizada a biblioteca *audioop.py*.

Para a frequência foi utilizada a transformada rápida de Fourier, por meio da biblioteca *numpy*: uma função definida a partir da série de Fourier, capaz de retirar a frequência dominante a partir de uma combinação de ondas contidas num sinal sonoro. (SANTOS, 2019)

Assim, por meio da biblioteca *serial*, utiliza-se da função *write* para enviar os dados para o arduino, por uma variável do tipo *char*, ou seja, possibilitando valores apenas entre 0 e 255. Por esse motivo, é necessário uma restrição das faixas de frequência e intensidade sonora e para isso são feitos cálculos a fim de limitar os valores absolutos obtidos. Por meio de análises de amostras de música foram determinados os limites superiores e inferiores de ambas informações, buscando homogenizar a distribuição de valores, pois notou-se uma densidade de

dados não uniforme, possuindo picos de variação que destoavam do padrão da amostra. Os gráficos abaixo exemplificam a análise feita:

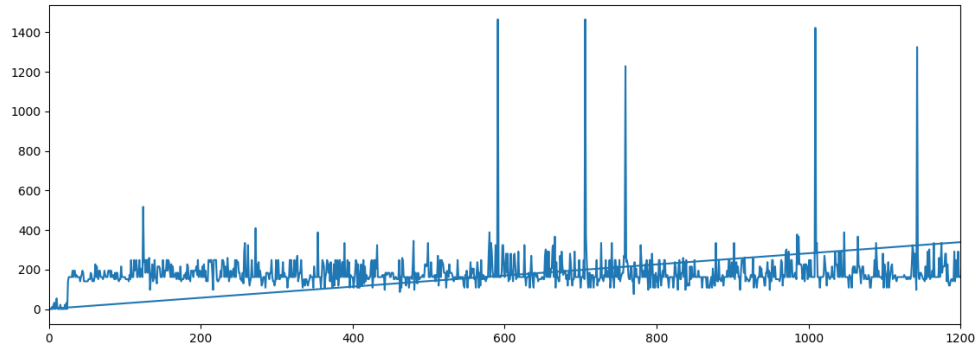


Figura 2: Gráfico de Frequência x Tempo, Música : America - A Horse With No Name

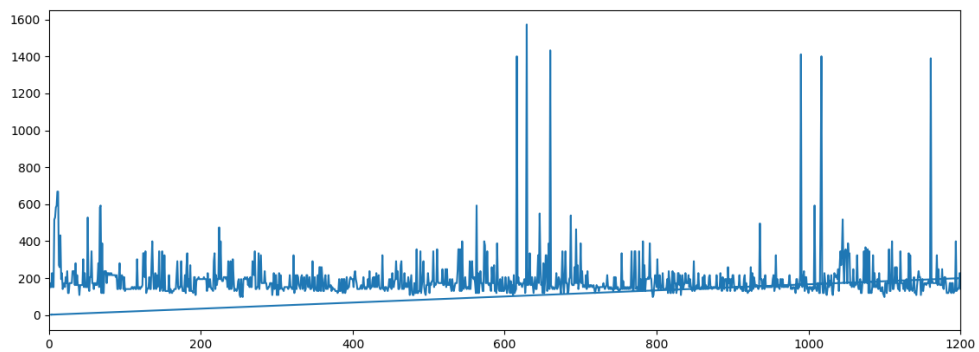


Figura 3: Gráfico de Frequência x Tempo, Música : Angra - Nova Era

Notam-se poucos valores que superam a casa de milhar, portanto não representam de forma adequada o conjunto de dados. Para a intensidade sonora, esta análise buscou apenas determinar o limiar superior utilizando o volume máximo possível do computador, visto que dependendo desse parâmetro, a faixa de valores pode variar, portanto definir um limite inferior pode ser prejudicial ao resultado esperado.

2.1.1 TRANSFORMADA RÁPIDA DE FOURIER

A transformada discreta de Fourier é uma ferramenta utilizada neste projeto de forma exclusivamente prática, por meio da biblioteca *numpy*, entretanto, é relevante abordá-la de forma a elucidar seu propósito. Porém, é importante reforçar a simplicidade desta explicação e seu foco para este trabalho.

”A transformada de Fourier é uma função originada da série de Fourier que explica exatamente quais frequências estão no sinal original de uma forma de onda qualquer.”(SANTOS, 2019, p. 1)

A transformada de Fourier tem como propósito a representação de uma complexa forma de onda em uma combinação de senoides geradoras da onda originária, sendo capaz de ser selecionada uma frequência específica. A exponencial representa os senoides e a integral permite escolher individualmente uma destas para melhor representar o sinal.(SANTOS, 2019)

Para este trabalho foi utilizada a FFT(*Fast Fourier Transform* ou Transformada Rápida de Fourier), um algoritmo que calcula a Transformada Discreta de Fourier(DFT), basicamente a Transformada padrão porém no campo discreto, neste caso, devido ao tratamento de dados computacionais. A DFT possui como finalidade a reconstrução com a maior representatividade possível de um conjunto de sinais com diferentes amplitudes e frequências. Assim, a FFT permite verificar quais sinais são mais dominantes e separá-los, podendo serem tratados como dados capazes de representar a amostra obtida.

Este algoritmo é dividido em três partes:

- 1 - Decompor um ponto N no domínio do tempo em vários sinais, contendo cada um valor.
- 2 - Achar o espectro de cada ponto de N.
- 3 - Sintetizar o espectro de N em um valor de frequência

(SMITH, 2011)

A ideia do algoritmo de compactação é analisar cada pequeno espaço do áudio, separar os componentes de frequência quase imperceptíveis para economizar espaço, bem como alguns dos que estão na extremidade superior do nosso alcance auditivo, determinando os componentes de frequência importantes, eliminando os que não são importantes, até que o resultado seja um arquivo de tamanho chegando a 10% do tamanho original, com a qualidade preservada de forma íntegra aos ouvidos humanos. (SANTOS, 2019, p. 1)

2.2 PROGRAMA PARA O ARDUINO

Programado em *C like*, este *software* recebe os dados processados pelo programa em *Python* e interpreta-os. Devido às limitações da biblioteca serial, usada para a transferência das informações, os valores de entrada variam de 0 a 255, restringindo certos parâmetros, como a precisão de frequência. No entanto, ainda assim é possível atingir uma gama de aproximadamente 256 matizes, abrangendo todo o espectro de luz de forma discreta.

Para controle da fita de led foram testadas duas bibliotecas específicas para este fim: a *Adafruit* e a *FastLED*. Logo a *FastLED* demonstrou-se mais eficaz e simples de ser

utilizada, sendo a opção escolhida neste projeto. Esta biblioteca permite controle de cada LED individualmente através de um vetor, no qual cada posição se refere a uma posição do LED, bem como determinar outros parâmetros, como brilho e cor. Ela possui duas estruturas específicas de padrões para gerenciamento de cores: o RGB (*Red, Green, Blue*) e o HSV (*Hue, Saturation, Valor*). Inicialmente foi trabalhado com o RGB, o qual controla individualmente o valor de cada canal fundamental de cor luz, demonstrando-se mais adequado para um tratamento de cores mais preciso e livre. Já o HSV revelou-se mais prático para este controle, além de ser capaz de definir o brilho, por meio do parâmetro *valor*, de cada LED, uma ferramenta que não se demonstrou possível utilizando a outra estrutura.

Inicialmente é definido o modelo da fita e criado um vetor de *structs* contendo três variáveis de valores inteiros, que dependem do tipo de padrão de cor: RGB ou HSV utilizado, no entanto, todos variam de 0 a 255, responsável por referenciar-se aos leds. Posteriormente, para o uso do controle por RGB, é feito um mapeamento de cores linear, utilizando a forma de *Spectrum*, a mesma usada, e já implementada, pela estrutura HSV com o parâmetro *hue*.(KRIEGSMAN, 2014)

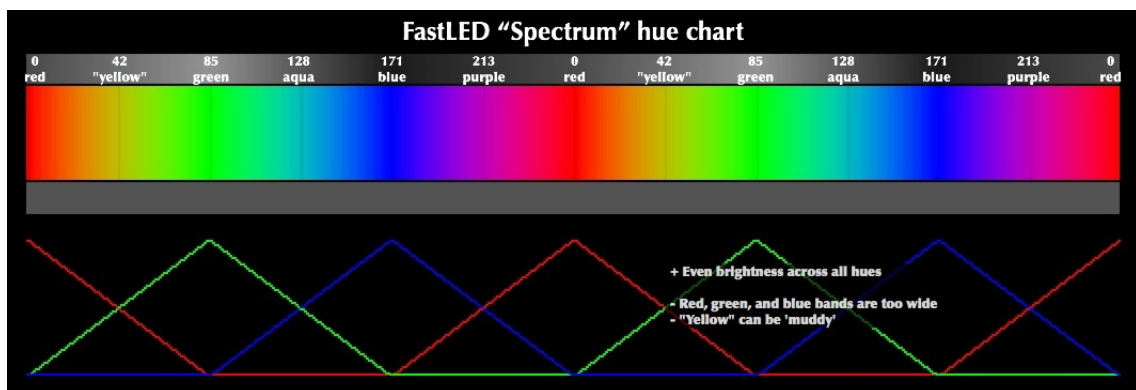


Figura 4: *Spectrum* - Mapeamento de Cores pelo padrão HSV

Este algoritmo permite controlar a densidade de faixas de frequência sonora associadas a diferentes faixas de frequência de cor. Por fim, são implementados os algoritmos de resposta da fita de LED. As funções e estruturas utilizadas foram retiradas primordialmente do site dos próprios desenvolvedores da biblioteca, referenciado neste documento. (GARCIA; KRIEGSMAN, 2016)

2.2.1 MAPEAMENTO DE LEDS

2.2.1.1 ALGORITMOS

Alguns algoritmos de resposta dos leds aos dados foram desenvolvidos a fim de testes e exploração das capacidades do projeto. Os algoritmos descritos abaixo seguem um padrão de sempre definir a cor do led baseado na frequência :

- Algoritmo de Barras : Uma lógica simples de correlação de cores com a frequência e iluminação de um número de leds referente ao valor da intensidade sonora de forma sequencial. Dentro deste algoritmo existem possibilidades de mudar o início da barra, como no início da fita, ou, partindo de seu centro, duas barras simétricas;
- Algoritmo de Pulsos : Através de um parâmetro de intensidade sonora, um led é acendido ao superar este parâmetro e posteriormente, por meio de uma iteração, todas as cores salvas em cada posição do vetor de leds são passadas para frente, criando uma noção de um pulso contínuo através da fita.
- Algoritmo Piano : Determinando um centro de duas linhas de LEDs a serem acendidas baseadas na frequência, o número de LEDs acendidos dessa linha é determinado por uma fração da intensidade sonora. Esse mapeamento é feito de forma espelhada em relação ao centro da fita, ao qual se assemelha à disposição de notas de um piano.

2.2.1.2 MAPEAMENTO DE COR

A fim de criar uma correspondência visual satisfatória com som e cores, foram feitas pesquisas a respeito de como essa relação acontece de acordo com os estímulos de cada elemento. Existe um fenômeno capaz de descrever e estudar este vínculo chamado de *Chromesthesia*, uma forma de sinestesia associada à cores e emoções. Sinestesia envolve a forma como outros sentidos são evocados a partir de um diferente, por exemplo, um cheiro associado a um sabor. Neste caso a *Chromesthesia* é associada as emoções fortalecidas a partir de um matiz de cor, como um sentimento de tristeza ser comumente ligado ao azul. Assim como as cores, a música também evoca emoções dependendo de seu gênero e variações de frequência e instrumentos. Em outras palavras, é possível ver cores ao ouvir uma música. (KIM, 2012)

Apesar de cores e som serem relacionados à emoções que cada um provoca, essas emoções são diferentes para cada pessoa, dependendo de seu passado e ligações íntimas a

estes dois tipos de estímulos. Assim, apesar de criar uma relação som-cor, ela pode não ser intuitiva para todas as pessoas, variando de acordo com sua própria percepção de músicas e cores.(MAKHLIN, 2015)

A correspondência de matizes é feita relacionando frequências baixas ao azul e altas frequências novamente ao mesmo azul, completando o círculo cromático, com a faixa de frequência crescendo ao longo do espectro de cores. Entretanto, é necessário definir uma faixa finita de frequência, pois a densidade de frequência numa amostra musical é diferente, portanto é preciso truncar o limite superior a fim de determinar um escopo frequencial.

O mapeamento de cores desenvolvido segue a lógica do círculo cromático, o qual possui uma variação padronizada entre os três canais de cores. Contudo, devido a densidade variada de frequência observada nas músicas, o mapeamento não foi dividido de maneira igualitária entre as faixas. Cada faixa possui um canal de valor máximo, 255, outro de valor mínimo, 0, e outro canal que varia de forma crescente ou decrescente de acordo com seu valor na faixa anterior.

Neste projeto foi optado a estrutura HSV, possuindo os seguintes parâmetros: Matiz (*Hue*), responsável por definir a identidade da cor, a Saturação (*Saturation*), relacionada a quantidade de cor e o Brilho (*Valor*), a qual controla a intensidade da luz do LED. A matiz é parametrizada pela frequência, a qual realiza o mapeamento de cores do *spectrum*, anteriormente descrito. A saturação foi relacionada diferentemente para cada algoritmo desenvolvido, mas normalmente para reforçar sua cor em momentos de altas intensidades. E o brilho, parametrizado diretamente pela intensidade sonora, ou seja, quanto mais alto o som, mais intenso é o brilho dos leds.(KIM, 2012)

3 HARDWARE

O *hardware* do projeto é constituído por três componentes: uma fita de LED endereçável (WS2812B), um microcontrolador Arduino UNO e uma fonte externa (5 *volts* e 2 amperes).

A fita contém trezentos leds de dimensão 50 mm x 50 mm em sua extensão , com cada led possuindo um microcontrolador WS2812B, o qual é responsável por fazer a iluminação ter a característica de endereçabilidade. Assim, existe uma vasta possibilidade no controle luminoso de cada led, bem como a disposição de cores ao longo da fita, estas divididas em cada canal fundamental do sistema RGB (*Red, Green, Blue*), o usado por este componente.

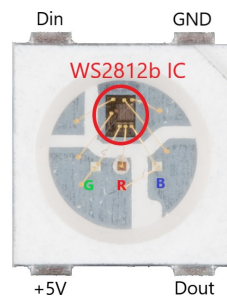


Figura 5: Controlador da LED

A fonte externa foi um componente necessário para o projeto, pelo fato de que o Arduino UNO não tinha capacidade de fornecer a corrente necessária para todos os leds funcionarem corretamente. Então foi escolhida uma fonte com essas especificações para manter a iluminação estável durante todo o comprimento da fita.

Por fim, o Arduino é conectado ao computador via cabo USB AB para transmissão de dados / alimentação da placa. Por meio de conectores, a porta digital 7 do Arduino é ligada com a porta de entrada de dados da fita, e a entrada GND (*ground*) da placa à GND (*ground*) da fita.

4 CRONOGRAMA E CUSTOS DO PROJETO

4.1 CRONOGRAMA

O cronograma foi desenvolvido em conjunto com os professores da disciplina buscando esclarecer as etapas desenvolvidas. Para a proposta inicial foi construído um cronograma prévio com o intuito de organizar o andamento do projeto, contudo, seus tópicos se modificaram ao longo do desenvolvimento do projeto, resultando em um cronograma final, correspondendo as etapas as quais de fato aconteceram. Estas diferenças evidenciam o caráter mutável de um projeto.

SEMANAS	CRONOGRAMA PROPOSTO
19 ago.	Definição de equipe e ideia de projeto
26 ago.	Concepção de projeto e pesquisas necessárias
02 set.	Diagrama completo e proposta inicial
09 set.	Aquisição de Componentes e Introdução ao Software e à teoria de Processamento Digital de Sinais (TDF)
16 set.	Testes de funcionamento e integração do Hardware e familiarização inicial com Software em Python (Biblioteca pyaudio) e Linguagem C like voltada ao Arduino
23 set.	MARCO I - Hardware funcional: PC, Arduino e Fita de LED integrados
30 set.	Familiarização com numpy (Biblioteca referente ao TDF) e C like / FastLED (Biblioteca referente ao gerenciamento da fita de LED) finalizada
14 out.	Input e output de dados pelo Python efetivado
21 out.	MARCO II - Softwares funcionais: Programa em Python finalizado / Programa para Mapeamento de Cor e Controle de LEDs pelo Arduino em Desenvolvimento
4 nov.	Avaliação da viabilidade do uso de Bluetooth e Auto-Falante
11 nov.	Algoritmo de Mapeamento de Cores finalizado
18 nov.	MARCO III - Projeto finalizado: hardware e software completamente integrados e funcionais
25 nov.	Testes finais de hardware e software
2 dez.	Verificações finais, Documentação e Apresentação

SEMANAS	CRONOGRAMA FINAL
19 ago.	Definição de equipe e ideia de projeto
26 ago.	Concepção de projeto e começo de pesquisas iniciais
02 set.	Diagrama completo e proposta inicial; Hardware funcional
09 set.	Familiarização inicial com Software em Python (Biblioteca pyaudio)
16 set.	Testes de funcionamento e integração do Hardware
23 set.	MARCO I - Hardware funcional: PC, Arduino e Fita de LED integrados
30 set.	Input e output de dados pelo Python efetivado, aquisição do hardware mais adequado (Fita de LED endereçável)
14 out.	Estudo da biblioteca FastLED (Controle de LED), correção de correspondências incorretas da fita
21 out.	MARCO II - Softwares funcionais: Programa em Python finalizado / Programa para Mapeamento de Cor e Controle de LEDs pelo Arduino em Desenvolvimento
4 nov.	Análise das possibilidades do uso de cores e brilho da fita e ajustes de algoritmos; Busca de fonte externa para alimentação da fita
11 nov.	Algoritmo de Mapeamento de Cores básico finalizado; Fonte externa funcional
18 nov.	MARCO III - Hardware e software completamente integrados e funcionais de forma unificada; Mapeamento de cores e algoritmos de controle parcialmente finalizados
25 nov.	Testes com o uso de outras estruturas de cores da biblioteca FastLED e desenvolvimento de algoritmos de controle de LEDs mais complexos
2 dez.	Verificações finais, Documentação, Apresentação e Vídeo

Figura 6: Cronograma Proposto e Cronograma Final do Projeto

Os componentes e ferramentas necessários para este projeto estão listados abaixo:

- Microcontrolador Arduino UNO
- Cabo USB AB
- Conectores
- Fita de LED endereçável RGB (WS2812B)
- *Software* na IDE do Arduino
- *Software* em *Python*
- Fonte externa de 5 Volts / 2 Amperes

4.2 CUSTOS

Componentes	Custo
Arduino UNO + Cabo USB	25 Reais
Fita de LED Endereçável	180 Reais
Fonte Externa	20 Reais
Total	225 Reais

5 DIFICULDADES E RESULTADOS

O projeto desmonstrou-se sem muitas dificuldades para seu processo de integração devido à linha direta de relação entre *hardware* e *software*. Contudo, foram enfrentados alguns empecilhos durante seu desenvolvimento.

Primeiramente, foi adquirida uma fita de LED não adequada ao projeto, sendo necessário uma nova aquisição, adiando levemente o desenvolvimento do software para o Arduino. Grande parte das dificuldades foram centradas na extração de dados por meio do programa em *python* e as limitações de leitura do programa do Arduino. Entretanto, foi desenvolvido uma forma eficaz de transferir os dados necessários entre os *softwares*.

No início da idealização do projeto, era visado apenas o uso da frequência sonora do computador, porém, apenas isso não demonstrava um resultado intuitivamente agradável. Assim, posteriormente foi capaz de extrair o volume referente ao som, melhorando o resultado significativamente.

Os testes realizados foram feitos utilizando primeiramente um gerador de frequência pura para verificar a correta correlação entre ela e as cores mostradas pela fita. O volume além de testado com valores absolutos, buscou-se alcançar uma resposta rápida entre o *hardware* e o *input* do som.

De maneira geral, o resultado obtido superou as expectativas dos autores. Apesar de sua linha direta de integração e simplicidade do *hardware*, o projeto ainda pode ser bastante explorado no quesito do *software* responsável por mapear e criar a dinâmica de luz e cores.

Foi produzido um vídeo de apresentação do projeto disponibilizado na plataforma do *Youtube*: youtu.be/i0ZS1kjcP40

6 CONCLUSÕES

6.1 CONCLUSÕES

O projeto não demonstrou grande dificuldade de ser realizado no quesito do *hardware*, o maior obstáculo em relação a isso foi o de determinar a corrente necessária para obtenção de um resultado final desejado pela equipe. O grande desafio visto pelo grupo foi na parte da realização dos *softwares*, sendo necessário um grande tempo para seu desenvolvimento, principalmente no início, para se efetuar a comunicação serial entre ambos e administrar o mapeamento de cores de forma adequada.

O projeto final atende aos requisitos propostos pela disciplina, com um uso pertinente de um microcontrolador, neste caso, escolhido o Arduino UNO. A equipe soube trabalhar bem em equipe, sem atritos e com boa distribuição de tarefas. Por fim, o resultado final alcançado foi satisfatório, conseguindo entregar uma boa experiência audiovisual.

6.2 TRABALHOS FUTUROS

Este projeto pode ser aprimorado essencialmente em seu *software*, visto que os algoritmos desenvolvidos durante o trabalho ainda são simples e diretos, não envolvendo uma análise complexa de dados ao longo do tempo. Além disso é possível uma integração de módulos ao Arduino como *bluetooth* e alto-falantes, com o objetivo de generalizar suas funcionalidades e aprimorar a experiência do usuário.

REFERÊNCIAS

- AASVIK, M. **What Is FFT and How Can You Implement It on an Arduino?** 2017. Disponível em: <<https://www.norwegiancreations.com/2017/08/what-is-fft-and-how-can-you-implement-it-on-an-arduino>>. Acesso em: 1 set. 2019.
- GARCIA, D.; KRIEGSMAN, M. **Color utility functions**. 2016. Disponível em: <http://fastled.io/docs/3.1/group_colorutils.html>. Acesso em: 20 set. 2019.
- KIM, S.-I. A Basic Study on the Conversion of Sound into Color Image using both Pitch and Energy. (Korea)]. **International Journal of Fuzzy Logic and Intelligent Systems**, v. 12, n. 2, p. 101–107, 2012.
- KRIEGSMAN, M. **FastLED HSV Colors**. 2014. Disponível em: <<https://github.com/FastLED/FastLED/wiki/FastLED-HSV-Colors>>. Acesso em: 2 nov. 2019.
- MAKHLIN, J. **Chromesthesia as Phenomenon: Emotional Colors**. 2015. Disponível em: <https://digitalcommons.lmu.edu/cgi/viewcontent.cgi?article=1010context=arc_wp>. Acesso em: 1 out. 2019.
- NAKAGOME, S. **Fourier Transform 101 — Part 4: Discrete Fourier Transform**. 2018. Disponível em: <<https://medium.com/sho-jp/fourier-transform-101-part-4-discrete-fourier-transform-8fc3fbb763f3>>. Acesso em: 1 set. 2019.
- SANTOS, C. S. dos. **Usando a transformada de Fourier para construir um visualizador de música com Python e Arduino**. 2019. Disponível em: <<https://medium.com/@cafelouco/usando-a-transformada-de-fourier-para-construir-um-visualizador-de-mC3BAsica-com-python-e-arduC3ADno-57c98e723cdc>>. Acesso em: 1 set. 2019.
- SMITH, S. W. **Chapter 12: The Fast Fourier Transform How the FFT works**. 2011. Disponível em: <<https://www.dspguide.com/ch12/2.htm>>. Acesso em: 27 out. 2019.