



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA**

**CAMPUS CAMPINA GRANDE**

**CURSO SUPERIOR BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**HÉRCULES DE SOUSA SILVA**

**MATHEUS ALVES DA SILVA**

**Easy Melody:**

Uma plataforma para aprendizado e produção musical

Campina Grande – PB

2023

Hércules de Sousa Silva

Matheus Alves da Silva

## **Easy Melody:**

Uma plataforma para aprendizado e produção musical

Este Trabalho de Conclusão de Curso foi submetido ao Curso de Bacharelado em Engenharia de Computação, oferecido pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - *Campus* Campina Grande, como parte do cumprimento dos requisitos parciais necessários para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Igor Barbosa da Costa

Campina Grande – PB

2023

S586p Silva, Hércules de Sousa.

Easy melody: uma plataforma para aprendizado e produção musical / Hércules de Sousa Silva, Matheus Alves da Silva. Campina Grande, 2023.

57 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2023.

Orientador: Prof. Dr. Igor Barbosa da Costa.

1. Desenvolvimento *web* 2. Engenharia de *software* 3. Teoria musical I. Silva, Matheus Alves da II. Costa, Igor Barbosa da III. Título.

CDU 004.4

Hércules de Sousa Silva

Matheus Alves da Silva

## **Easy Melody: Uma plataforma para aprendizado e produção musical**

Este Trabalho de Conclusão de Curso foi submetido ao Curso de Bacharelado em Engenharia de Computação, oferecido pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - *Campus* Campina Grande, como parte do cumprimento dos requisitos parciais necessários para a obtenção do título de Bacharel em Engenharia de Computação.

Aprovada em \_\_\_/\_\_\_/\_\_\_.

### BANCA EXAMINADORA

---

Prof. Dr. Igor Barbosa da Costa

IFPB

---

Prof. Danyllo Wagner Albuquerque

IFPB

---

Prof. Henrique do Nascimento Cunha

IFPB

*Dedicamos este trabalho para nossos familiares e amigos que nos forneceram incentivo e força para persistir na busca desta jornada.*

## **Agradecimentos**

Expressamos aqui a nossa profunda gratidão a todas as pessoas que nos apoiaram nesta trajetória, pois graças a elas, grandes empecilhos se tornaram pequenos e nós conseguimos a força necessária para continuar trilhando essa jornada. Portanto, gostaríamos de deixar registrado neste documento nossos agradecimentos a: primeiramente, Deus, sem o qual nada seria possível e por nos ter fornecido força, coragem, saúde e resiliência para superar os desafios enfrentados. Nossas famílias, por terem se mantido presentes e nos fornecido suporte, principalmente emocional, para continuarmos lutando apesar dos problemas que surgiam. Ao professor Igor Barbosa da Costa, por todo o auxílio, paciência e disposição de embarcar nessa trajetória, graças a isso este trabalho se tornou possível e pôde chegar no nível que alcançou. Ao cliente da aplicação Cauê Beltrame por ter se disposto a nos ajudar, sempre sendo muito paciente e solícito. Aos nossos amigos, que através do convívio, companheirismo, da escuta, das alegrias compartilhadas nos deram ânimo para seguir buscando nosso objetivo. Dentre eles gostaríamos de fazer uma menção honrosa a Filipe Salomão por ter se disposto a testar o produto e ter feito várias contribuições ao trabalho aqui descrito.

*A música exprime a mais alta filosofia numa linguagem que a razão não compreende.*

**Arthur Schopenhauer**

## Resumo

A música é uma das principais formas de expressão humana, ela tem existido e acompanhado a humanidade ao longo da sua história, contudo, apesar de tão presente na vida cotidiana, poucas são as pessoas que de fato sabem tocar um instrumento e compreendem as ideias e os princípios por trás da teoria musical, devido às dificuldades de assimilação dos conceitos teóricos e a sua aplicação em contextos práticos. Pensando nisso, foram levantados requisitos com o objetivo de desenvolver uma plataforma que apresentasse uma interface lúdica voltada para a aprendizagem e a experimentação dos conceitos musicais sem exigir habilidades motoras ou conhecimentos teóricos profundos. Este documento apresenta as etapas do processo de desenvolvimento desta aplicação, que contou com o apoio do produtor e professor de música Cauê Beltrame e foi construída através do método ágil Scrum em conjunto com práticas da Engenharia de Software e de conhecimentos na área de desenvolvimento de sistemas voltados para *Web*. A partir de tudo isso foi elaborada uma plataforma que permite que os seus usuários construam faixas musicais com melodia, harmonia e ritmo, sem exigir os conhecimentos teóricos anteriormente mencionados e que ainda apresenta funcionalidades extras como um pequeno gerador de melodia e a exportação das músicas em MIDI (*Musical Instrument Digital Interface*), um formato amplamente usada na indústria de produção musical para registrar e armazenar músicas. Diante dos resultados obtidos e do *feedback* do cliente, este trabalho pode ser classificado como tendo sido concluído com sucesso por ter atingido todos os objetivos inicialmente propostos.

**Palavras-chave:** Desenvolvimento *web*; Engenharia de *software*; Música; Teoria musical; Ensino.



## **Abstract**

Music is one of the main forms of human expression, it has existed and followed humanity throughout its history, however, despite being so present in everyday life, few people actually know how to play an instrument and understand the ideas and the principles behind musical theory, due to the difficulties in assimilating theoretical concepts and their application in practical contexts. With this in mind, requirements were raised with the aim of developing a platform that presented a playful interface aimed at learning and experimenting with musical concepts without requiring motor skills or in-depth theoretical knowledge. This document presents the stages of the development process of this application, which had the support of the producer and music teacher Cauê Beltrame and was built using the agile Scrum methodology in conjunction with Software Engineering practices and knowledge in the field of systems development for the Web. From all this, a platform was created that allows its users to build musical tracks with melody, harmony and rhythm, without requiring the previously mentioned theoretical knowledge and which also has extra features such as a small melody generator and the ability to export songs in MIDI (Musical Instrument Digital Interface), a format widely used in the music production industry to record and store songs. Given the results obtained and the client's feedback, this work can be classified as having been completed successfully as it achieved all the objectives initially proposed.

**Key words:** Web development; Software Engineering; Music; Music theory; Teaching.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Escala maior de dó . . . . .	18
Figura 2 - Escala cromática . . . . .	19
Figura 3 - Rascunho da aplicação . . . . .	28
Figura 4 - Rascunho do sequenciador de melodia com notas não redimensionáveis . . . . .	29
Figura 5 - Rascunho do sequenciador de melodia com notas redimensionáveis e indicador visual . . . . .	31
Figura 6 - Menu de configuração de ritmo . . . . .	32
Figura 7 - Sequenciador musical com notas descritas por graus . . . . .	34
Figura 8 - Menu de geração . . . . .	35
Figura 9 - Compasso quaternário . . . . .	36
Figura 10 - Compasso ternário . . . . .	37
Figura 11 - Sequenciador de melodia com faixa de exemplo . . . . .	41
Figura 12 - Sequenciador de melodia com notas selecionadas . . . . .	42
Figura 13 - Tonalidade configurada . . . . .	43
Figura 14 - Menu de configuração dos acordes . . . . .	44
Figura 15 - Menu de seleção dos ritmos . . . . .	44
Figura 16 - Botão de geração de melodia . . . . .	45
Figura 17 - Sequenciador de testes . . . . .	46
Figura 18 - Configurações de geração de melodia . . . . .	47
Figura 19 - Opções de formatos de melodia . . . . .	47
Figura 20 - Confirmar sobrescrita . . . . .	48
Figura 21 - Botões de exportação e importação . . . . .	49
Figura 22 - Salvar importação . . . . .	49

## LISTA DE TABELAS

Tabela 1 - Comparação entre o Easy Melody e outras plataformas . . . . .	40
--	----

## LISTA DE ABREVIATURAS E SIGLAS

C	Nota dó
D	Nota ré
E	Nota mi
F	Nota fá
G	Nota sol
A	Nota lá
B	Nota si
API	<i>Application Programming Interface</i>
REST	<i>Representational State Transfer</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
MIDI	<i>Musical Instrument Digital Interface</i>
JSON	<i>JavaScript Object Notation</i>
SPA	<i>Single-Page Application</i>

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>13</b>
1.1 Objetivos.....	14
<b>2. TRABALHOS RELACIONADOS.....</b>	<b>15</b>
2.1 Song Maker.....	15
2.2 ChordChord.....	15
2.3 GarageBand.....	16
<b>3. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>17</b>
3.1 Música.....	17
3.2 Teoria Musical.....	18
3.2.1 Escalas musicais.....	18
3.2.2 Intervalos musicais.....	18
3.2.3 Campo harmônico.....	19
3.3 Tecnologia de Produção Musical: Sequenciadores e MIDI.....	19
3.3.1 Sequenciadores musicais.....	20
3.3.2 MIDI.....	20
3.4 Tecnologias de Desenvolvimento de Sistemas.....	20
3.4.1 JavaScript.....	21
3.4.2 TypeScript.....	21
3.4.3 Node.js.....	21
3.4.4 Tone.js.....	21
3.4.5 Tonal.js.....	22
3.4.6 React.....	22
3.4.7 React Konva.....	22
3.4.8 MobX.....	23
3.4.9 Material UI.....	23
3.5 Arquitetura de Sistemas.....	23
3.5.1 API REST.....	23
3.5.2 SPA (Single-Page Application).....	24
3.5.3 Stores.....	24
3.6 Processo e Organização.....	24
3.6.1 Scrum.....	25
3.6.2 Git.....	25
3.6.3 GitHub.....	26
<b>4. PROCESSO DE DESENVOLVIMENTO.....</b>	<b>27</b>
4.1 Sprint 01: Tocando Sons Através do Navegador.....	27

4.2 Sprint 02: Construindo um Protótipo do Sequenciador de Melodia.....	29
4.3 Sprint 03: Refatorando Protótipos para Construir o Produto.....	30
4.4 Sprint 04: Implementação do Sequenciador de Harmonia.....	31
4.6 Sprint 06: Exportando faixa musical formatado de um JSON e adicionando drag and drop.....	33
4.7 Sprint 07: Realizando pequenos ajustes e implementando o metrônomo.....	33
4.8 Sprint 08: A criação de um sequenciador melódico de teste e a geração de melodia a partir de padrões musicais.....	34
4.9 Sprint 09: Divisão por três.....	36
4.10 Sprint 10: Exportação e importação MIDI.....	37
<b>5. RESULTADOS.....</b>	<b>38</b>
5.1 Comparação com Outras Plataformas.....	39
5.2 Demonstração.....	41
5.2.1 Criando e tocando melodias simples.....	41
5.2.2 Criando e tocando harmonias.....	43
5.2.3 Criando e tocando ritmos.....	44
5.2.4 Utilizando o mini gerador de melodia.....	45
5.2.5 Realizando exportação e importação de JSON e baixando arquivo MIDI..	49
<b>7. CONSIDERAÇÕES FINAIS.....</b>	<b>51</b>
7.1 Conclusão e Trabalhos Futuros.....	51
7.2 Limitações.....	52
<b>8. REFERÊNCIAS.....</b>	<b>54</b>

## 1. INTRODUÇÃO

A música, presente em todas as culturas ao longo da história, é mais do que uma forma de arte: é uma ferramenta poderosa de expressão e comunicação. Desde as primeiras sociedades até a era digital, a música tem moldado e enriquecido a experiência humana, conforme mostram descobertas arqueológicas e estudos (BOAL-PALHEIROS, 2014). O avanço tecnológico, particularmente nas mídias digitais, aumentou significativamente o acesso e a interação com a música, tanto em termos de como os artistas escrevem e gravam seu conteúdo e como nós, como ouvintes, o consumimos (FEYRE, 2020).

A música é também um aspecto distintivo da humanidade, com estudos sugerindo que sua origem pode ser mais antiga que outras grandes inovações humanas, como a linguagem e a agricultura (ROCHA, 2013). Essa perspectiva histórica reforça a importância da música, apesar dos debates sobre seu papel exato na evolução humana. Sua presença em várias culturas e épocas (ROCHA, 2013) sublinha a necessidade de métodos de ensino que honrem sua rica história e acessibilidade universal.

Aprender música, no entanto, apresenta desafios significativos, dentre os quais pode-se citar: a dificuldade enfrentada pelos alunos em assimilar conceitos teóricos e aplicá-los na prática, o uso de partituras, uma prática comum no ensino tradicional, pode ser particularmente intimidadora para iniciantes, criando uma barreira para a entrada no mundo musical e a dificuldade em traduzir conhecimento teórico em habilidades práticas em instrumentos musicais pode ser uma fonte de frustração e desânimo (DARBY, 2018).

Diante dos desafios enfrentados no ensino de música, surge a necessidade de uma abordagem renovada, mais inclusiva, acessível e envolvente. Para atender a essa necessidade, este trabalho apresenta o desenvolvimento de uma plataforma inovadora para produção musical, focando na melodia e na experiência do usuário. O objetivo é tornar o aprendizado musical uma experiência mais intuitiva e gratificante.

O objetivo deste projeto é o desenvolvimento de um sequenciador MIDI intuitivo, que se distingue das ferramentas existentes. Ele será acessível para iniciantes, mas igualmente atraente para músicos mais avançados, visando superar as barreiras comuns no aprendizado musical e promover a criatividade e a experimentação.

Espera-se que este projeto transforme a educação musical, tornando a criação musical mais inclusiva e ampla, melhorando o acesso à educação musical e democratizando a criação artística. O objetivo é contribuir significativamente para a educação musical e a expressão artística, reforçando a música como uma linguagem universal que une e inspira pessoas de todas as idades e origens.

## **1.1 Objetivos**

O objetivo geral deste Trabalho de Conclusão de Curso é desenvolver uma aplicação voltada para a produção, aprendizado e experimentação musical, utilizando uma interface gráfica lúdica e interativa. A aplicação será projetada para facilitar a compreensão de conceitos musicais, tanto em termos de teoria quanto de prática, através de um sequenciador MIDI.

Com o intuito de atingir o objetivo geral, foram determinados os seguintes objetivos específicos:

- Permitir que o usuário consiga realizar experimentações em cada uma das camadas musicais de forma individual: esse objetivo será alcançado através da construção de dois sequenciadores musicais, sendo um para melodia e outro para harmonia, além de uma terceira estrutura para a escolha de ritmos musicais predefinidos;
- Auxiliar no aprendizado e experimentação de melodias: esta meta será alcançada através de um pequeno gerador de melodias no qual o usuário poderá explorar diferentes formatos de um trecho melódico.



## 2. TRABALHOS RELACIONADOS

Neste tópico, são apresentados projetos relacionados que proporcionam *insights* valiosos para o desenvolvimento da plataforma proposta neste trabalho. Todos são ferramentas inovadoras na área de produção musical, cada uma com suas características e limitações.

### 2.1 Song Maker

O Song Maker, uma iniciativa do Google, visa facilitar a construção de músicas de maneira simplificada e interativa, tornando a música mais acessível por meio da tecnologia (Prass, 2018). Este aplicativo possui uma interface intuitiva, que inclui dois sequenciadores musicais distintos: um dedicado à construção de ritmos e outro à composição de melodias e harmonias. Contudo, o Song Maker apresenta algumas limitações significativas: não permite criar notas que se estendam por mais de uma batida, selecionar múltiplas notas simultaneamente, e o processo de criação de ritmos pode se tornar repetitivo com o crescimento da música. Além disso, a utilização de acordes exige que o usuário tenha conhecimento prévio sobre formação de acordes.

### 2.2 ChordChord

ChordChord é uma plataforma focada no ensino e aprendizado musical, com especial atenção à criação de progressões harmônicas (CHORDCHORD, s.d.). Esta plataforma oferece uma interface agradável que possibilita aos usuários criar harmonias e ritmos de maneira bastante completa. Além disso, o ChordChord fornece sugestões de faixas musicais geradas por inteligência artificial. Apesar de sua eficiência em harmonia e ritmo, o ChordChord não oferece recursos para o aprendizado e criação de linhas melódicas personalizadas, que muitas vezes são a parte mais memorável das músicas. Esta limitação representa um desafio significativo para usuários que buscam desenvolver habilidades melódicas.

### **2.3 GarageBand**

O GarageBand é um software desenvolvido pela Apple com objetivo de permitir a produção de música através de meios digitais (Apple, s.d.). Ele possui um enorme conjunto de funcionalidades e recursos, além de contar também com uma interface atraente e agradável que o torna bastante popular no universo de criação musical, contudo o GarageBand não é um software indicado para indivíduos com pouco conhecimento em teoria musical, pois quando visto pelos olhos de uma pessoa leiga este pode se tornar confuso devido ao número de funcionalidades (TECHTUDO, s.d.). Além disso, diferente do produto construído neste trabalho ele não é acessível através de outros dispositivos que não sejam de origem da Apple, o que diminui a sua portabilidade e acessibilidade a diferentes públicos.

### **3. FUNDAMENTAÇÃO TEÓRICA**

Esta seção estabelece uma base teórica para o entendimento do projeto, abordando conceitos e ferramentas essenciais utilizados no desenvolvimento do software. A explanação desses elementos é fundamental para compreender as decisões e o caminho seguido durante o desenvolvimento.

#### **3.1 Música**

A música, sendo uma característica universal da espécie humana e um comportamento social complexo, existe em todas as épocas e culturas, embora com designações e significados diferentes (BOAL-PALHEIROS, 2014). Sendo algo tão fascinante e tão antigo quanto o próprio homem, muitos pesquisadores têm se questionado qual a razão e a importância da música para que esta tenha se perpetuado ao longo do tempo, uma vez que ela não parece ser necessária para a sobrevivência e sua utilidade imediata não é óbvia. Darwin via a capacidade musical como precursora da capacidade linguística, tendo ambas origens comuns e coexistindo como complementos da comunicação humana (BOAL-PALHEIROS, 2014). Porém, para além desta origem e utilidade linguística a música tem importância na formação de interações sociais, na cooperação e coesão de grupos ao fortalecer laços interpessoais e, principalmente, no seu valor emocional graças a sua capacidade de produzir prazer nos seus ouvintes e intérpretes (BOAL-PALHEIROS, 2014).

A definição do que é música tem sido alvo de discussão de diversos autores. Dessa forma, apresentar um conceito sobre ela não é algo tão trivial. Uma das definições mais abrangentes é de que música é a arte dos sons (CHEDIAK, 1986) sendo uma composição harmoniosa e expressiva de melodia, harmonia e ritmo (DESCOMPLICANDO A MÚSICA, s.d.). A melodia é a voz principal, ou seja, é aquilo que pode ser cantado, a harmonia é a sobreposição de notas que servem de base para a melodia, e o ritmo é a marcação do tempo de uma música (DESCOMPLICANDO A MÚSICA, s.d.). Quando organizados de forma coerente, esses elementos geram sons harmoniosos e cativantes.

## 3.2 Teoria Musical

A teoria musical é a base para a composição musical consciente e se constitui em um conjunto de normas e sistemas destinado a analisar, classificar, compreender, compor e se comunicar acerca da música (TECLACENTER, s.d.). Dentre os principais conceitos da teoria musical podemos destacar:

### 3.2.1 Escalas musicais

Escala é uma sequência ordenada de notas (DESCOMPLICANDO A MÚSICA, s.d.). Essa ordenação é estruturada de acordo com intervalos musicais feitos de forma a reproduzir sons harmônicos e agradáveis de serem ouvidos. Como exemplo podemos citar uma escala bastante conhecida, a escala diatônica maior de dó, que está descrita na Figura 1:

Figura 1 – Escala maior de dó

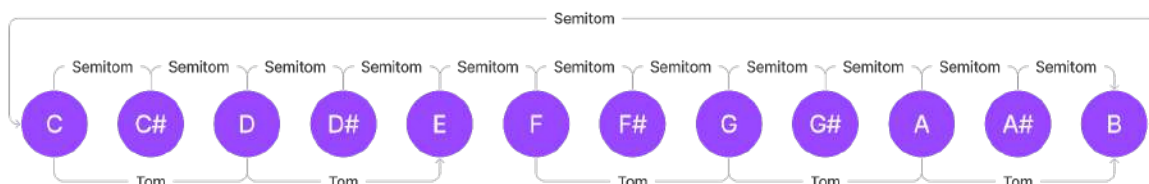


Fonte: autoria própria

### 3.2.2 Intervalos musicais

São as distâncias entre dois sons (DESCOMPLICANDO A MÚSICA, s.d.). Os intervalos são medidos em tons e semitons, conforme a Figura 2, onde semitons são a menor unidade de medida entre duas notas musicais e os tons são a união de dois semitons.

Figura 2 – Escala cromática



Fonte: autoria própria

### 3.2.3 Campo harmônico

Campo harmônico é o conjunto de acordes construídos a partir de uma escala musical. Um acorde é a união de três ou mais notas tocadas simultaneamente (DESCOMPLICANDO A MÚSICA, s.d.), sendo formado por pelo menos o primeiro, o terceiro e o quinto grau da escala em uso. Os graus são uma nomenclatura criada para ajudar o músico na localização dos intervalos (DESCOMPLICANDO A MÚSICA, s.d.), eles são usados para fazer referência a uma nota a partir de outra, para melhor compreensão consideremos a escala de dó maior representada pela Figura 1. Os graus são enumerados de acordo com a sequência apresentada, sendo a primeira nota o primeiro grau, a segunda o segundo, e assim por diante. Portanto, como já mencionado, para formar o acorde de dó maior é necessário haver pelo menos o primeiro, terceiro e o quinto graus, dessa forma é preciso tocar as notas dó, mi e sol. Já para formar o acorde de ré é preciso ter as notas ré, fá e lá, e assim sucessivamente. É importante também ressaltar que o acorde de ré no campo harmônico maior de dó é chamado de acorde menor, pois sua terça é menor (DESCOMPLICANDO A MÚSICA, s.d.), porém para os fins deste documento essa nomenclatura não é tão relevante.

### 3.3 Tecnologia de Produção Musical: Sequenciadores e MIDI

No cenário contemporâneo de produção musical, a tecnologia desempenha um papel fundamental, capacitando músicos e produtores a explorar novas fronteiras

criativas e aprimorar seus processos de composição. Entre as ferramentas cruciais dessa revolução tecnológica estão os sequenciadores musicais e o padrão MIDI.

### **3.3.1 Sequenciadores musicais**

Um sequenciador de música é uma ferramenta que permite programar e reproduzir sequências de notas, ritmos e efeitos automaticamente, em vez de executar ou gravar cada parte em tempo real. Os sequenciadores não geram seus próprios sons. Em vez disso, eles enviam informações MIDI e CV (Control Voltage) para acionar outros instrumentos ou parâmetros de efeitos (MITCHELL, 2022).

### **3.3.2 MIDI**

*Musical Instrument Digital Interface* (MIDI) é um padrão para transmitir e armazenar música, originalmente projetado para sintetizadores digitais. A reprodução de músicas antes da invenção do MIDI era feita por meio da transmissão das ondas sonoras na forma de um sinal analógico para que este fosse reproduzido através de alto-falantes, porém com a criação deste padrão a propagação do som passou a ser feita de uma forma completamente diferente. O MIDI não envia as ondas sonoras produzidas por um instrumento, em vez disso, ele envia informações sobre as notas musicais e o dispositivo receptor usa seus próprios mecanismos internos para gerar os sons (WRIGHT, 2023). Esta inovação foi bastante revolucionária no cenário musical por permitir a comunicação entre diferentes instrumentos musicais, como sintetizadores, teclados digitais e baterias de diferentes marcas, algo que não acontecia antes da concepção deste formato (WRIGHT, 2023).

## **3.4 Tecnologias de Desenvolvimento de Sistemas**

No contexto da Engenharia de Software, a escolha cuidadosa das tecnologias de desenvolvimento desempenha um papel muito importante na eficiência e na robustez dos sistemas criados. Nas seções abaixo serão apresentadas as principais escolhas tecnológicas feitas para o desenvolvimento deste trabalho.

### 3.4.1 JavaScript

JavaScript é uma linguagem de programação leve, interpretada, orientada a objetos, baseada em protótipos e em *first-class functions* (funções de primeira classe), mais conhecida como a linguagem de script da Internet (ROVEDA, 2021). Sua compatibilidade com os navegadores *web* modernos tornou-a uma escolha bastante natural, tendo em vista que um dos requisitos não funcionais do projeto era que ele fosse possível de ser utilizado na forma de um site.

### 3.4.2 TypeScript

O TypeScript é um *superset* do JavaScript, ou seja, um conjunto de ferramentas por meio do qual é possível definir tipagem estática de variáveis, parâmetros e retornos de funções (ROVEDA, 2023). Essa tipagem estática oferece a grande vantagem de permitir que os desenvolvedores identifiquem erros de forma mais rápida e eficiente durante o desenvolvimento. A linguagem é capaz de apontar inconsistências ou erros em tempo de compilação, além de contribuir na melhora da escalabilidade e manutenibilidade do código (ROVEDA, 2023).

### 3.4.3 Node.js

Node.js é um ambiente de execução JavaScript baseado no motor V8 do Chrome que permite executar aplicações desenvolvidas com a linguagem de forma autônoma, sem depender de um navegador. Com ele, é possível criar praticamente qualquer tipo de aplicação *web*, desde servidores para sites estáticos e dinâmicos, até *APIs* e sistemas baseados em microsserviços. O Node.js adota uma arquitetura assíncrona e não-bloqueante baseada em eventos, eliminando a necessidade de filas de processamento e tornando as aplicações mais eficientes e responsivas (MELO, 2021).

### 3.4.4 Tone.js

Tone.js é uma *framework* de áudio da *web* para criar músicas interativas no navegador (TONEJS, s.d.). Ele se baseia na *API* de áudio do JavaScript e proporciona

uma interface de alto nível para simplificar o processo de criação e manipulação de sons e músicas no ambiente do navegador. O Tone.js permite a sintonização e sincronização de eventos, o que garante precisão e coordenação das faixas musicais reproduzidas. Ele também inclui sintetizadores de som embutidos e apresenta recursos poderosos de sequenciamento de trechos musicais.

### 3.4.5 Tonal.js

Tonal.js é uma biblioteca de teoria musical, ela contém funções para manipular elementos tonais da música (notas, intervalos, acordes, escalas, modos, tonalidades). Ela atua apenas como abstração desses conceitos e, por si só, não é capaz de reproduzir som (TONALJS, s.d.), sendo necessária a sua combinação com a biblioteca anterior a fim de produzir faixas musicais harmônicas e agradáveis.

### 3.4.6 React

React é uma biblioteca JavaScript para a criação de interfaces de usuário. Criado em 2011 pelo time do Facebook, o React surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no *feed* de notícias da rede social, bem como em outros pontos do seu produto, tais como no *chat*, *status*, listagem de contatos e outros. Ela permite a divisão da interface em componentes reutilizáveis e independentes, facilitando a construção de uma aplicação padronizada e coesa (ROVEDA, 2023).

### 3.4.7 React Konva

React Konva é uma biblioteca JavaScript para desenhar gráficos complexos em tela usando React (LAVRENOV, 2020). Ele fornece ligações declarativas e reativas ao *Konva Framework* combinando a abordagem componentizada do React e permitindo a criação de ilustrações interativas e de alta complexidade.



### 3.4.8 MobX

MobX é uma biblioteca de gerência de estado, que tem como objetivo facilitar a conexão dos seus dados reativos à sua interface (GALVÃO, 2020). Ela introduz uma abordagem reativa que permite que as atualizações no estado sejam automaticamente refletidas nas partes correspondentes da interface do usuário, além de trazer uma melhor organização do código ao separar o estado do aplicativo em depósitos denominados de *stores*.

### 3.4.9 Material UI

Material UI (MUI) é uma estrutura CSS que fornece componentes React prontos para uso e segue o Material Design do Google lançado em 2014 (LLC, 2022), agilizando o processo de criação de uma aplicação React e facilitando a padronização da interface seguindo as melhores práticas do mercado.

## 3.5 Arquitetura de Sistemas

A escolha cuidadosa do modelo arquitetural dentro do âmbito da arquitetura de sistema representa uma responsabilidade crucial na definição da estrutura e eficiência de uma aplicação. Nas seções abaixo serão exploradas algumas arquiteturas que foram usadas para a construção desta plataforma.

### 3.5.1 API REST

*API REST*, também chamada de *API RESTful*, é uma interface de programação de aplicações em conformidade com as restrições do estilo de arquitetura *REST*, ela permite a interação com serviços *web RESTful* (RED HAT, 2023).

O termo *API* se refere a um conjunto de definições e protocolos usado no desenvolvimento e na integração de aplicações. Já o termo *REST* diz respeito ao conjunto de restrições da arquitetura, onde uma das principais características é o uso de uma representação do estado do recurso solicitado transferido através do protocolo *HTTP* e normalmente descrito através do formato *JSON*, embora também possa ser apresentado como *HTML*, *XLT*, *Python*, *PHP* ou texto sem formatação (RED HAT,

2023). A arquitetura *REST* já vem sendo consolidada no mercado por trazer benefícios como ser independente de linguagem, permitir a separação entre cliente e servidor e facilitar a escalabilidade dos sistemas.

### **3.5.2 SPA (*Single-Page Application*)**

Um SPA (aplicativo de página única) é uma forma de implementação de aplicações que carrega apenas um único documento da *web* e, em seguida, atualiza o conteúdo do corpo desse único documento por meio de APIs JavaScript, como XMLHttpRequest e Fetch, quando um conteúdo diferente deve ser mostrado. Isso permite que os usuários utilizem sites sem carregar páginas totalmente novas do servidor, o que pode resultar em ganhos de desempenho e uma experiência mais dinâmica (DEV MEDIA, s.d.).

### **3.5.3 Stores**

As *stores* podem ser encontradas em qualquer arquitetura Flux e podem ser comparadas um pouco com os controladores no padrão MVC. A principal responsabilidade das *stores* é mover a lógica e o estado de seus componentes para uma unidade autônoma testável que pode ser usada tanto no *front-end* quanto no *back-end* (MobX, s.d.). Elas funcionam como depósitos do estado da aplicação, sendo utilizadas para isolar e centralizar os dados do seu aplicativo de forma a garantir mais performance, flexibilidade e organização do código.

## **3.6 Processo e Organização**

Nas subseções a seguir serão analisados os conceitos fundamentais que compõem o processo e a organização do trabalho descrito neste documento. Além da escolha das tecnologias é crucial escolher de forma correta os princípios e metodologias que irão guiar o trabalho a ser realizado, bem como a maneira com a qual eles serão estruturados a fim de fornecer uma base sólida que irá sustentar o projeto ao longo do tempo.

### 3.6.1 Scrum

O Scrum é uma estrutura ágil de gestão de projetos que ajuda as equipes a estruturar e gerenciar o trabalho por meio do conjunto de valores, princípios e práticas. O Scrum estimula as equipes a aprender com as experiências, a se organizar para resolver um problema e a refletir sobre os êxitos e fracassos para melhorar sempre (ATLASSIAN, s.d.).

O Scrum tem a melhoria contínua como um dos seus conceitos-chave, assim como a valorização do aspecto humano em detrimento da burocracia que muitas outras filosofias possuem. No Scrum, o trabalho é organizado em ciclos de tempo fixo denominados “*sprints*” e envolve reuniões diárias curtas chamadas de “*daily*”.

Esse método se baseia em essência no conhecimento empírico (SCRUM.ORG, s.d.), que reconhece que, no começo de um projeto, a equipe responsável por criar um novo produto ou serviço muitas vezes desconhece os detalhes de como realizar essa tarefa complexa, e que esse conhecimento virá com a experiência. Dessa forma, o Scrum é estruturado de forma a encorajar mudanças decorrentes do aprendizado e observação contínuos. Sempre com naturalidade e com vistas a garantir a melhora constante do time.

### 3.6.2 Git

O Git é um Sistema de Controle de Versões Distribuído. Estes sistemas de controle possuem a função de registrar quaisquer alterações feitas em cima de um código, armazenando essas informações e permitindo que, caso seja necessário, um(a) programador(a) possa regredir a versões anteriores de uma aplicação de modo simples e rápido (KRIGER, 2022). Git é um projeto de código aberto, que se destaca por sua eficiência, segurança, flexibilidade e arquitetura distribuída, onde cada desenvolvedor possui uma cópia completa de todo o repositório no qual está trabalhando. O Git foi desenvolvido em 2005 por Linus Torvalds, criador do *kernel* do sistema operacional Linux, e desde sua criação ele tem se mostrado bastante superior aos seus principais concorrentes em diversas atividades essenciais para a produção de

*software*, como na criação de *commits* e *branches*, mesclagem de diferentes variações de código e a comparação de versões de arquivos (ATLASSIAN, s.d.). Essas atividades são cruciais para um processo de desenvolvimento eficiente.

### 3.6.3 GitHub

O GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso (L., 2019). Além disso, o GitHub disponibiliza uma série de outras funcionalidades para facilitar a evolução dos projetos hospedados em sua plataforma, como a organização de problemas a serem resolvidos em “*issues*”, a criação de solicitações de adição de mudanças ao código principal chamadas de “*pull request*”, ferramentas de integração e entrega contínua, entre outras.

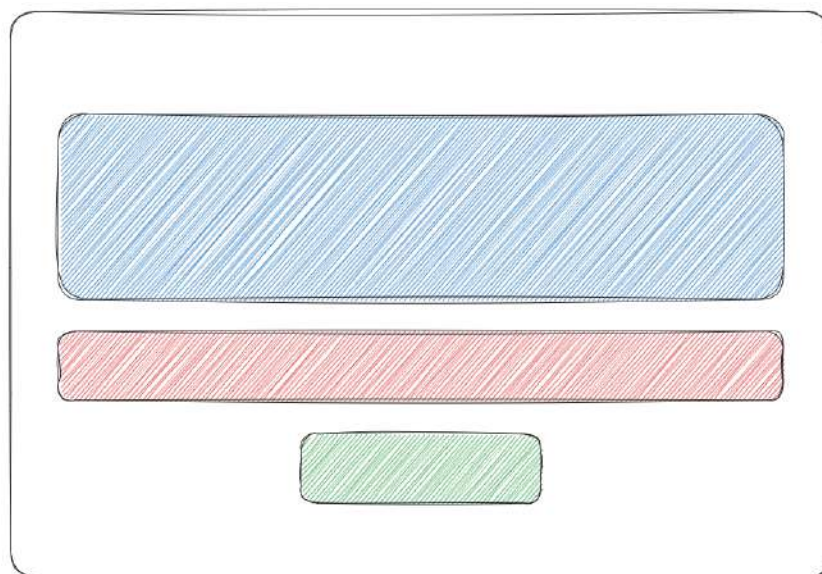
## 4. PROCESSO DE DESENVOLVIMENTO

Esta seção detalha os componentes cruciais na produção da aplicação. O projeto foi gerenciado utilizando a metodologia ágil Scrum. No contexto deste método, os papéis foram distribuídos da seguinte forma: Cauê Beltrame, representando o cliente, atuou como *Product Owner* (PO), focando nos interesses e necessidades dos usuários; o professor Igor Barbosa da Costa assumiu o papel de *Scrum Master*, organizando as *sprints* bissemanais e auxiliando na compreensão dos requisitos do sistema; Hércules de Sousa e Matheus Alves integraram a equipe de desenvolvimento. Além da organização da equipe, outro ator que foi de fundamental importância para a construção deste projeto foi o usuário da aplicação, que pode ser definido como um aluno de música ou pessoas interessadas no assunto, com uma faixa etária entre 12 e 40 anos e acesso à internet através de um computador e que almejam experimentar de forma digital conceitos da teoria musical, realizar testes ou avaliar ideias para suas músicas. Após definido o usuário e a equipe do projeto foi iniciado o processo de desenvolvimento, dessa forma a seguir seguem as descrições de cada uma das *sprints* que deram origem ao trabalho construído.

### 4.1 *Sprint* 01: Tocando Sons Através do Navegador

A primeira etapa do desenvolvimento iniciou-se com uma reunião de alinhamento envolvendo todos os membros da equipe. O propósito era esclarecer a ideia central da aplicação e estabelecer os requisitos necessários. Foi acordado que o projeto visaria a criação de uma plataforma de produção musical para ensino, composta por dois sequenciadores musicais - um para melodia e outro para harmonia - e uma estrutura para seleção de ritmos pré-definidos.

Figura 3 – Rascunho da aplicação



Fonte: autoria própria

A Figura 3, um rascunho da aplicação, ilustra esses componentes. O sequenciador de melodia, representado pelo retângulo azul, permitiria aos usuários definir a voz principal da música. Analogamente, o retângulo vermelho refere-se ao sequenciador de harmonia, destinado à especificação de acordes de acompanhamento. O retângulo verde indicava uma estrutura para a escolha de ritmos, que seriam disponibilizados pela aplicação e tocados em *loop*.

Com os requisitos em mente, a equipe buscou as tecnologias apropriadas, focando inicialmente em como reproduzir sons através do navegador. Considerando o uso do *framework React*, duas bibliotecas foram consideradas: *Reactronica* e *Tone.js*. A escolha recaiu sobre o *Tone.js* pela sua maior flexibilidade e versatilidade, apesar da *Reactronica* ser mais fácil de usar. A decisão foi influenciada pela percepção de que a *Reactronica*, apesar de não apresentar limitações imediatas, poderia restringir o desenvolvimento futuro devido à falta de *delays* nas notas e à forma como organiza as informações musicais.

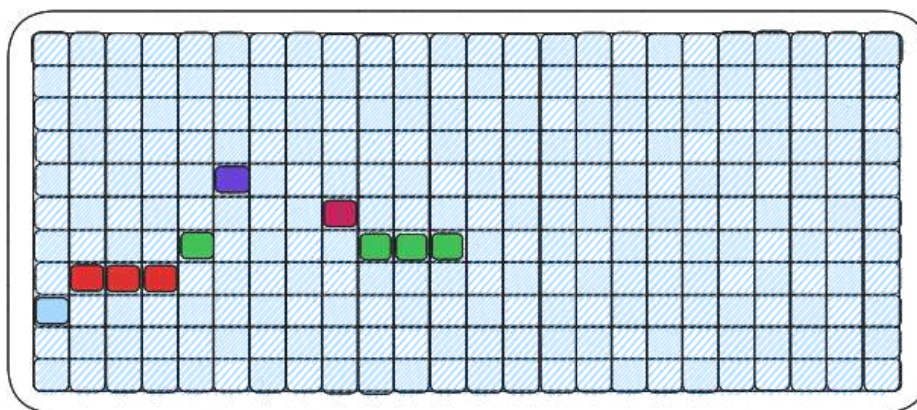
Por fim, após a seleção da biblioteca, um código de teste para reproduzir sons no navegador foi desenvolvido e apresentado na reunião quinzenal seguinte. Sem *Rs*

específicos de melhorias devido ao estágio inicial do projeto, a equipe decidiu avançar na construção de um protótipo para o sequenciador de melodia

#### 4.2 *Sprint 02*: Construindo um Protótipo do Sequenciador de Melodia

Durante a *Sprint 02*, a equipe concentrou esforços na construção de um protótipo para o sequenciador de melodia, conforme planejado anteriormente. O processo começou com um rascunho detalhado, representado na Figura 4, para facilitar a visualização do *design* proposto.

Figura 4 – Rascunho do sequenciador de melodia com notas não redimensionáveis



Fonte: autoria própria

Este protótipo inicial, desenvolvido usando HTML, CSS e JavaScript, era capaz de reproduzir uma sequência fixa de notas. Após a demonstração ao cliente, surgiu a necessidade de expandir a funcionalidade do sequenciador para permitir notas de duração variável. A implementação dessa característica com tecnologias *web* padrão se revelou desafiadora, especialmente para funcionalidades adicionais como seleção e arrasto de notas.

Portanto, foi decidido adotar o *React Konva*, uma biblioteca que oferece uma solução intermediária entre versatilidade e facilidade de uso. Essa escolha foi motivada pela capacidade da biblioteca de criar componentes dentro de uma *canvas* com alta performance, sua integração com o React e o suporte ativo da comunidade. O *React*

*Konva* atendia às necessidades imediatas de retângulos redimensionáveis e arrasto e solta de notas, alinhando-se com os requisitos futuros da aplicação.

Paralelamente, a equipe dedicou-se a aprofundar o conhecimento em conceitos musicais fundamentais, como compasso, figuras musicais, batida e ritmo, para informar adequadamente o *design* e as funcionalidades do sequenciador.

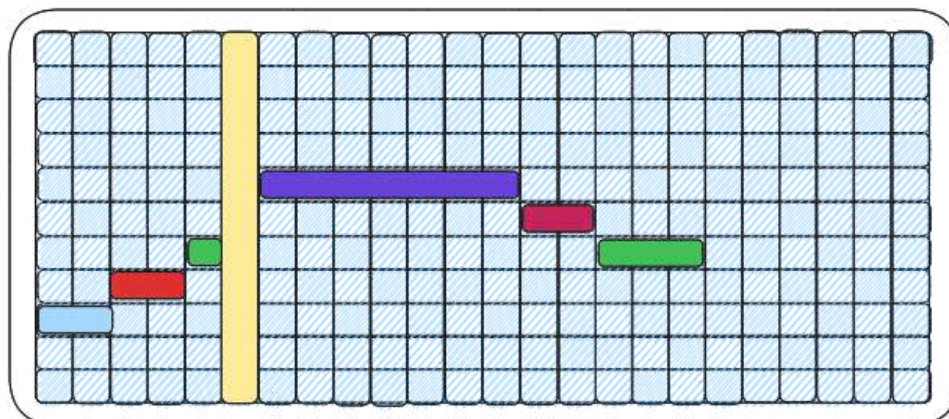
### **4.3 *Sprint* 03: Refatorando Protótipos para Construir o Produto**

No decorrer da *Sprint* 03, a equipe focou no aperfeiçoamento dos protótipos existentes, dando início ao desenvolvimento da versão final do produto. O *TypeScript* foi escolhido para melhorar a confiabilidade do código por meio de sua tipagem estática. O sequenciador de melodia passou por uma reconstrução completa, adotando uma abordagem baseada em componentes do *React*. As configurações foram organizadas em um arquivo separado para maior clareza.

A gestão de estado na aplicação, que crescia em complexidade, foi otimizada com a adoção da biblioteca *MobX*, facilitando a manutenção e a depuração do código. Um desafio específico enfrentado foi a implementação de um indicador visual de batida no sequenciador de melodia, especialmente na sincronização com os eventos musicais do *Tone.js*. Uma solução eficiente foi encontrada, incluindo a automatização do *scroll* do sequenciador para manter o indicador visível e sincronizado com a música, como mostrado na Figura 5.



Figura 5 – Rascunho do sequenciador de melodia com notas redimensionáveis e indicador visual



Fonte: autoria própria

Além disso, a configuração da aplicação foi desenvolvida, permitindo ajustes no BPM, volume e tonalidade. O pan foi removido a pedido do cliente, pois se mostrou mais relevante em contextos com múltiplos instrumentos. Com o término desta *sprint*, a seção de melodia foi concluída e validada em uma reunião, preparando o terreno para o desenvolvimento da seção de harmonia.

#### 4.4 *Sprint* 04: Implementação do Sequenciador de Harmonia

A *Sprint* 04 teve como foco principal a implementação da funcionalidade de harmonia, que possibilita a criação de acordes para acompanhar a melodia. Utilizando o código do sequenciador melódico como base, foi desenvolvido um sistema simplificado de harmonia. Este sistema representa acordes em uma única linha, alinhada com a melodia, para manter a clareza e aprimorar a experiência do usuário.

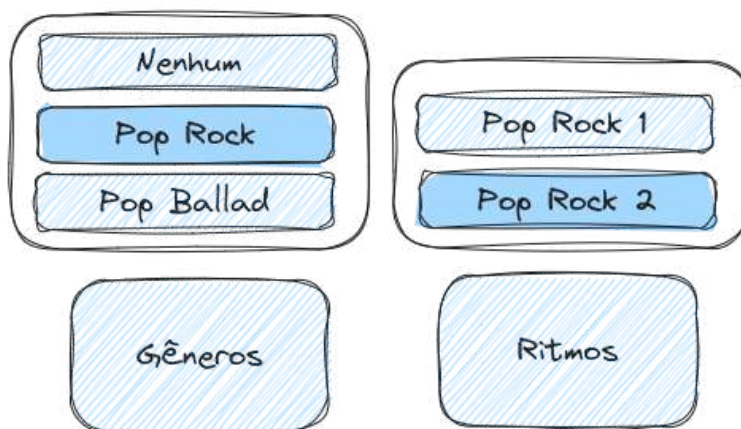
No sequenciador harmônico, os usuários podem gerar acordes clicando nos retângulos correspondentes, com o acorde inicial sendo automaticamente o primeiro grau do campo harmônico da tonalidade selecionada. Além disso, foi adicionada a funcionalidade de editar acordes, permitindo alterações na nota base, seleção entre tríades ou tétrades e escolha de acordes de empréstimo modal.

Após a implementação, uma reunião de revisão foi realizada, onde o cliente aprovou o trabalho com sugestões de ajustes menores. Esta reunião também serviu para coletar os requisitos necessários para a próxima etapa do projeto: a implementação da seleção de ritmos pré-definidos.

#### 4.5 *Sprint 05*: Implementando estrutura para escolha de ritmos e criando um pequeno servidor

Na *Sprint 05*, a implementação do componente de ritmo foi o foco principal, completando a estrutura básica para o estudo de faixas musicais na plataforma. Utilizando amostras de ritmos pré-definidos, organizadas em gêneros como Pop Rock e Pop Ballad, a equipe desenvolveu um menu interativo para a escolha de ritmos, com botões para seleção de gêneros e ritmos específicos, como mostrado na Figura 6.

Figura 6 – Menu de configuração de ritmo



Fonte: autoria própria

Paralelamente, para superar a limitação do Tone.js em reproduzir arquivos de áudio locais, foi desenvolvida uma pequena API REST, que serviu como servidor para disponibilizar os sons de bateria e piano.

Ao final da *sprint*, uma reunião de revisão e *feedback* levantou possíveis melhorias e funcionalidades adicionais. Devido a restrições de tempo, algumas dessas ideias foram reservadas para trabalhos futuros. A *sprint* se encerrou com êxito, e o

próximo passo definido foi focar na implementação de seleção múltipla de notas musicais.

#### **4.6 *Sprint* 06: Exportando faixa musical formatado de um JSON e adicionando *drag and drop***

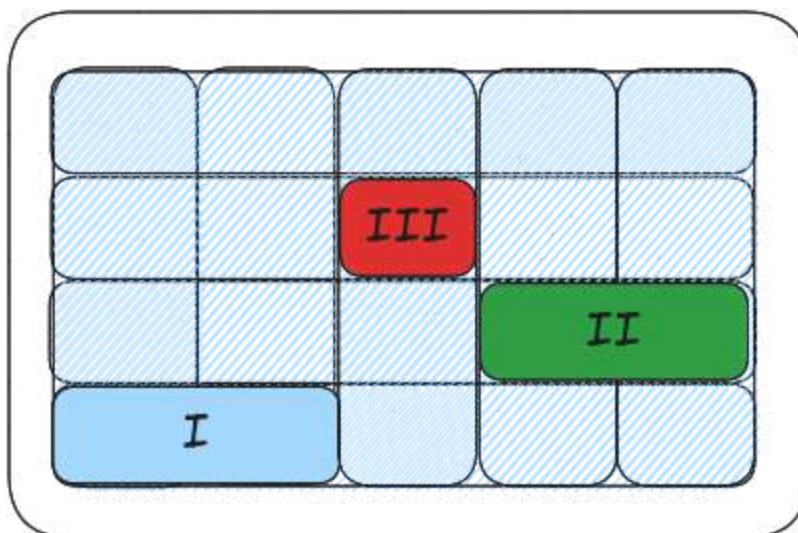
A *Sprint* 06 focou no desenvolvimento de funcionalidades essenciais: a exportação de faixas musicais em formato JSON e a implementação de arrasto e soltura (*drag and drop*). A necessidade de um mecanismo para salvar e posteriormente importar faixas musicais surgiu após uma apresentação beta, levando à implementação da exportação do estado da aplicação em JSON, abrangendo elementos como melodia, harmonia e ritmo.

A funcionalidade de *drag and drop*, apesar de desafiadora, foi efetivada com sucesso, permitindo inicialmente o movimento de uma nota por vez e depois expandindo para a manipulação de múltiplas notas. Essas adições foram bem recebidas durante a revisão de *sprint*, consideradas um acréscimo valioso. Para a próxima *sprint*, planejou-se desenvolver um metrônomo e realizar ajustes na aplicação, como a inclusão de acidentes musicais e alteração na representação das notas no sequenciador.

#### **4.7 *Sprint* 07: Realizando pequenos ajustes e implementando o metrônomo**

A *Sprint* 07 teve como foco principal a implementação de um metrônomo e a realização de ajustes refinados na aplicação, visando aprimorar a interatividade e a experiência educativa dos usuários. Uma mudança significativa foi na representação das notas no sequenciador de melodia, introduzindo a visualização em graus musicais (ver Figura 7), o que facilita o entendimento dos intervalos e ajuda na transposição de músicas.

Figura 7 – Sequenciador musical com notas descritas por graus



Fonte: autoria própria

Adicionalmente, desenvolveu-se um metrônomo na plataforma, um recurso essencial para estudos rítmicos, oferecendo um pulso sonoro constante e ajustável. Outra implementação importante foi a capacidade de adicionar acidentes musicais, permitindo alterações nas notas em meio tom através de um comando intuitivo.

Com a conclusão bem-sucedida desta *sprint*, foi realizada uma reunião de revisão onde as melhorias foram positivamente avaliadas. Motivada pelo sucesso, a equipe planejou focar na próxima *sprint* em adicionar atalhos de teclado para controle da reprodução e na criação de um sequenciador de testes, facilitando a experimentação na geração de melodias.

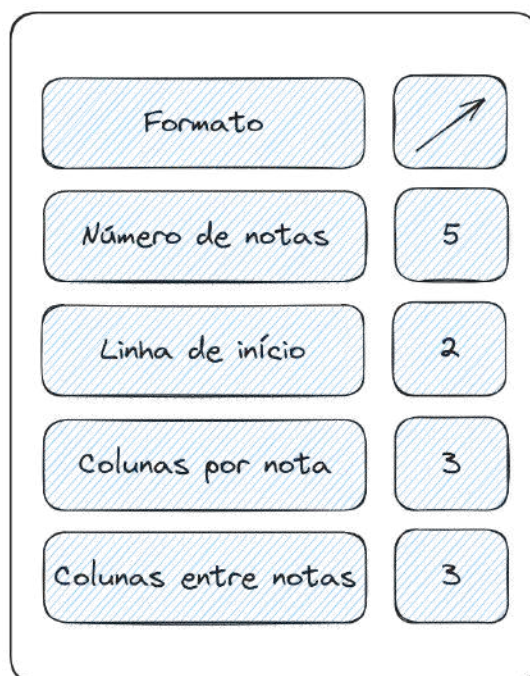
#### **4.8 *Sprint* 08: A criação de um sequenciador melódico de teste e a geração de melodia a partir de padrões musicais**

A *Sprint* 08 foi um marco no projeto, focando na geração de melodias com base em padrões musicais definidos pelo cliente. Um 'laboratório de teste' foi desenvolvido, um sequenciador melódico aprimorado que permitia aos usuários duplicar e modificar melodias de teste, melhorando a flexibilidade na experimentação musical.

Implementou-se um sistema de copiar e colar para as melodias e um indicador visual de inserção no sequenciador padrão.

Um algoritmo para a geração de melodia foi criado, com opções de padrões e configurações de parâmetros acessíveis por um menu intuitivo, ilustrado na Figura 8. A geração de melodias ocorria com um clique, permitindo que as notas fossem adicionadas ao sequenciador de teste antes de serem transferidas para a melodia padrão.

Figura 8 – Menu de geração



Fonte: autoria própria

Após implementar essas funcionalidades, uma reunião de retrospectiva validou as adições, sugerindo apenas pequenos ajustes. O planejamento das próximas etapas incluiu a adição de compassos ternários, exportação de arquivos MIDI e foco na documentação do produto.

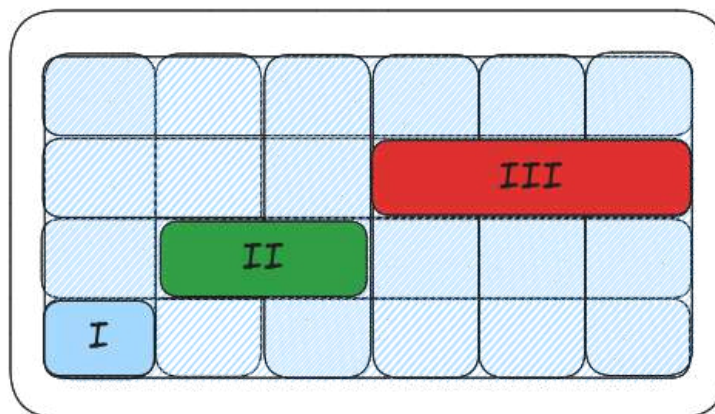
#### 4.9 *Sprint 09*: Divisão por três

A *Sprint 09* foi dedicada ao desafio técnico de implementar compassos ternários na aplicação, uma melhoria complexa devido à estrutura pré-existente baseada em compassos quaternários. A nova implementação requereu alterações significativas na disposição dos retângulos no sequenciador, agora divididos em frações menores para acomodar notas em intervalos de terça, enriquecendo a estrutura rítmica.

As mudanças exigiram adaptações em diversas áreas do código, incluindo a validação das posições das notas, ajustes na duração das notas para o compasso ternário e a sincronização com o Tone.js. Foi desenvolvida uma lógica para identificar retângulos adjacentes e ajustar a duração e a validação dos retângulos.

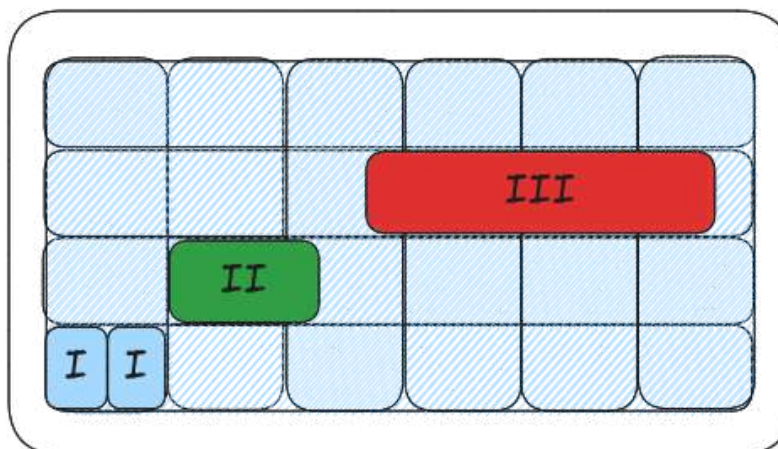
As Figuras 9 e 10 ilustram a diferença entre os compassos quaternários e ternários, demonstrando a flexibilidade rítmica adquirida. A implementação dos compassos ternários foi bem recebida, marcando uma evolução significativa na aplicação.

Figura 9 – Compasso quaternário



Fonte: autoria própria

Figura 10 – Compasso ternário



Fonte: autoria própria

#### 4.10 *Sprint* 10: Exportação e importação MIDI

A *Sprint* 10 concentrou-se no desenvolvimento da funcionalidade de manipulação de arquivos MIDI, permitindo aos usuários importar e exportar músicas. Utilizou-se o submódulo MIDI da biblioteca Tone.js para ler e escrever dados MIDI e implementou-se um botão para ativar a função de exportação, inicialmente focada na melodia.

A funcionalidade de exportação MIDI envolveu a conversão detalhada de notas em um objeto MIDI, com cálculos precisos para cada nota e a criação de um Blob para o download do arquivo. Posteriormente, a funcionalidade foi expandida para incluir a harmonia e as configurações de faixa musical, proporcionando uma representação mais completa da música.

Com a conclusão bem-sucedida da importação e exportação MIDI, a *Sprint* alcançou seus objetivos, levando o projeto para a fase de documentação detalhada, preparando-o para a conclusão e apresentação final.

## 5. RESULTADOS

A partir dos esforços descritos na seção de metodologia foi concebida uma aplicação que possibilita que pessoas sem conhecimento de teoria musical consigam testar suas ideias musicais de forma simples e direta. Como resultados têm-se dois sequenciadores musicais utilizados para a criação de melodia e harmonia, e um menu para a escolha de ritmos criados previamente pelo cliente da aplicação. O sequenciador de melodia assemelha-se a uma tabela na qual o usuário pode criar retângulos para representar notas musicais, esse retângulos são móveis e redimensionáveis para trazer flexibilidade no momento de criação dos sons, além disso é possível manipular mais de um retângulo ao mesmo tempo, seja para apagar ou arrastá-los ao longo da interface. Cada um dos retângulos é rotulado de acordo com sua respectiva linha na tabela e levando-se em consideração qual é o tom ou o grau configurado, de forma que a nota tônica ou primeiro grau seja sempre a nota mais abaixo no sequenciador e as demais notas sigam uma sequência de acordo com os intervalos da escala diatônica maior.

Para a harmonia foi aproveitado o sequenciador de melodia já criado e foram realizadas alterações para atingir o comportamento esperado nesta outra estrutura. Uma das modificações feitas foi de que o sequenciador de harmonia tivesse apenas uma linha e que ao selecionar alguma posição ela deveria ser preenchida com o primeiro grau do campo harmônico configurado a partir da tonalidade escolhida. Para editar o acorde colocado basta que o usuário clique em cima do acorde que um menu de configurações irá aparecer onde ele poderá realizar as alterações almejadas.

Para a parte de ritmos existe um menu na parte inferior da aplicação onde é possível selecionar uma faixa de percussão entre as opções criadas pelo cliente e disponibilizadas para uso. Já no que diz respeito ao gerador de melodia existe a renderização dinâmica de um sequenciador de testes na parte superior da interface quando o usuário clica no botão de gerar melodia, este sequenciador serve de





laboratório para a criação e teste das melodias geradas antes de serem adicionadas no sequenciador de melodia padrão.

Os resultados obtidos foram satisfatórios para o cliente da aplicação e para colegas do curso e amigos que testaram a aplicação de forma informal e despreziosa, porém a forma como a interface está organizada poderia ser melhorada a partir de uma melhor aplicação das regras de *design* para aplicações *web*, o que poderia ter sido aperfeiçoado por um profissional *designer*. Ainda na esfera do *design*, as cores escolhidas não passaram por nenhum tipo de teste para validar se pessoas de baixa visão estariam confortáveis ao utilizar a plataforma, sendo este também mais um ponto de melhoria com vistas à criação de uma aplicação mais inclusiva. Existe também um desbalanceamento de flexibilidade entre o sequenciador de melodia e o de harmonia, sendo o primeiro bastante versátil possibilitando seleção de várias notas e a movimentação destas, enquanto que o segundo não possui tais funcionalidades. Além desses pontos existem ainda outras fragilidades que serão tratadas no capítulo de trabalhos futuros e limitações.

## **5.1 Comparação com Outras Plataformas**

Na Tabela 1 serão listadas as funcionalidades obtidas pelo trabalho descrito neste documento e comparadas com as funções de aplicações semelhantes anteriormente citadas, com o objetivo de tornar mais claro o posicionamento do Easy Melody quando relacionado a outras plataformas já utilizadas no mercado.

Tabela 1 – Comparação entre o Easy Melody e outras plataformas

<b>Funcionalidade</b>	<b>Song Maker</b>	<b>ChordChord</b>	<b>GarageBand</b>	<b>Easy Melody</b>
Criação de melodia				
Criação de harmonia				
Criação de ritmo				
Importação de faixas musicais				
Exportação de faixas musicais				
Redimensionamento de notas e/ou acordes				
Criação de acordes de forma fácil				
Seleção de notas para mover, copiar e apagar				
Criação de acordes de acordo com o campo harmônico				
Geração de melodia				
Geração de harmonia e/ou ritmo				
Compartilhamento de música por link				

Fonte: autoria própria

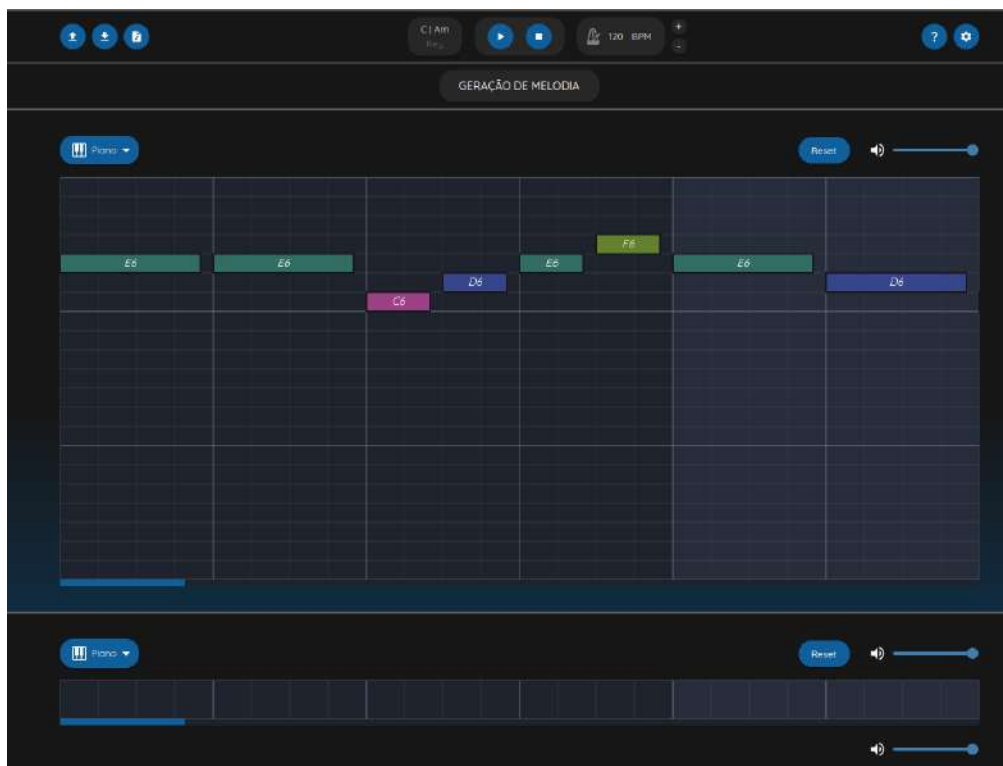
## 5.2 Demonstração

Neste tópico, é oferecida uma demonstração prática da aplicação, destacando seus fluxos de trabalho e funcionalidades-chave. As imagens apresentadas refletem o estado atual da aplicação, mas é importante notar que mudanças na interface podem ocorrer em versões futuras.

### 5.2.1 Criando e tocando melodias simples

Na tela inicial da aplicação que pode ser encontrada no [link](#), o usuário pode interagir com o sequenciador de melodia para compor trechos musicais. Conforme ilustrado na Figura 11, cada retângulo representa uma nota musical, identificável pela notação padrão. Estes retângulos podem ser movidos ou redimensionados dentro do sequenciador para ajustar a melodia.

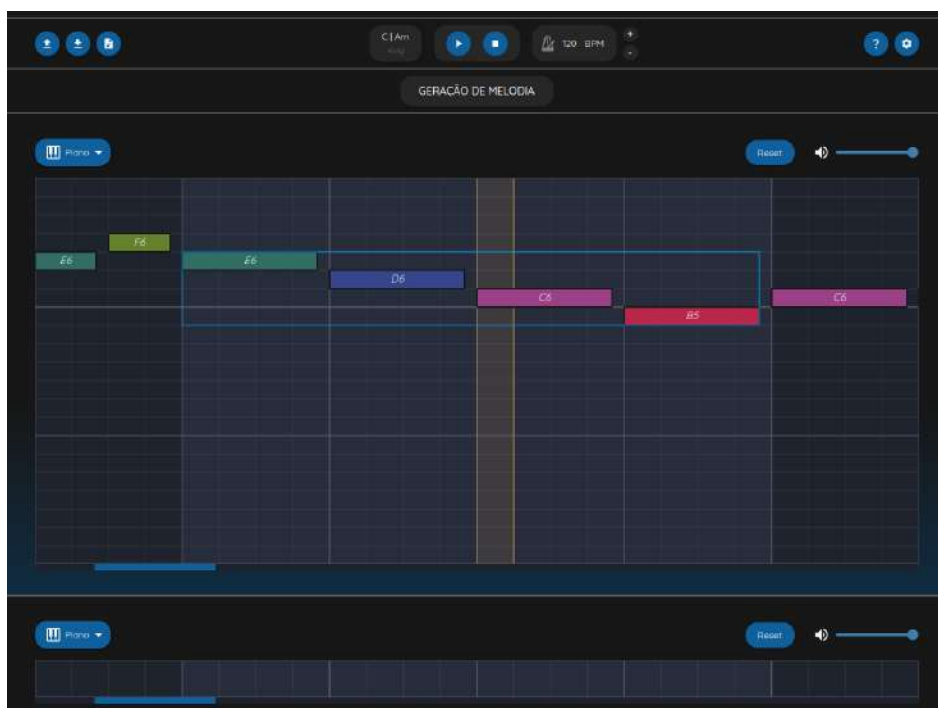
Figura 11 – Sequenciador de melodia com faixa de exemplo



Fonte: autoria própria

Os usuários têm a flexibilidade de editar e interagir com a melodia de várias maneiras. Para remover as notas, basta clicar com o botão direito ou selecionar várias notas usando a tecla Shift e apertando Delete, como mostrado na Figura 12. Além disso, o sequenciador oferece controles de reprodução, incluindo botões de play, pause e stop, e a opção de iniciar a reprodução a partir de um ponto específico da melodia.

Figura 12 – Sequenciador de melodia com notas selecionadas



Fonte: autoria própria

Depois da criação da melodia é possível escutá-la ao clicar no botão de *play* localizado no centro da aplicação na parte superior e melhor ilustrado na Figura 13 do tópico abaixo. Ao clicar no botão, a faixa musical será carregada em um evento que irá produzir os sons das notas na sequência descrita e mover o indicador visual para indicar onde a melodia se encontra. Enquanto a música estiver sendo reproduzida o botão de tocar se torna o de pausar, para caso o usuário deseje interromper a operação e fazer alguma modificação antes de dar prosseguimento. Outra opção de funcionalidade é o botão de parar que irá interromper a música e voltar a sua execução para o início. O usuário também tem a possibilidade de definir explicitamente a partir de

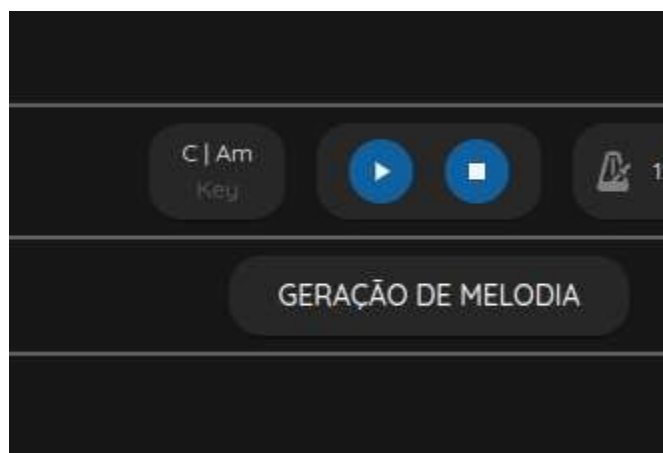
qual ponto da linha do tempo do trecho musical ele deseje começar segurando a tecla Ctrl do teclado e clicando com o botão esquerdo do mouse em uma coluna, ao realizar essa ação o indicador musical será posicionado na coluna anterior à selecionada.

### 5.2.2 Criando e tocando harmonias

Nesta seção explora-se a criação de harmonias, uma funcionalidade central da aplicação. O sequenciador de harmonia, alinhado com o sequenciador melódico, permite ao usuário compor acordes que complementam a melodia.

Para adicionar um acorde, o usuário clica na posição desejada no sequenciador. O acorde criado inicialmente corresponde à tríade do primeiro grau do campo harmônico da tonalidade selecionada, que pode ser ajustada pelo botão 'key', conforme mostrado na Figura 13.

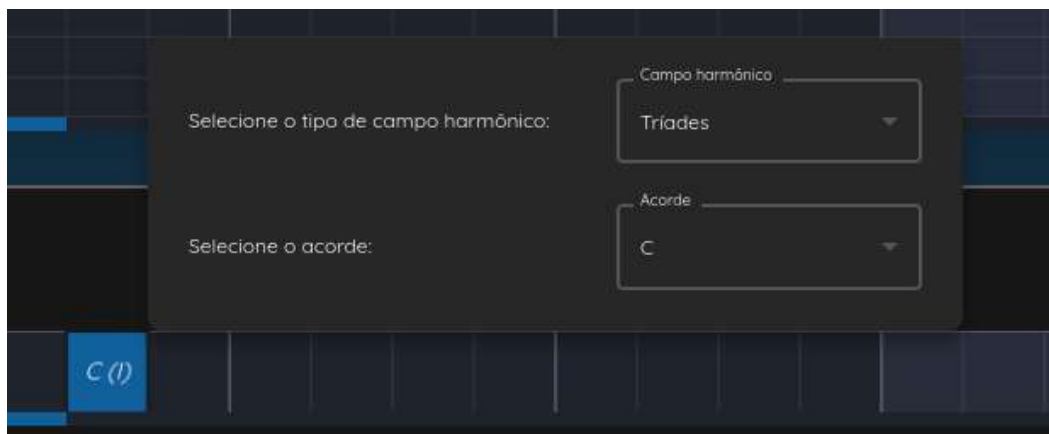
Figura 13 – Tonalidade configurada



Fonte: autoria própria

A edição de acordes é feita de forma simplificada, permitindo alterações na nota tônica e na estrutura do acorde, incluindo a escolha entre tríades, tétrades ou acordes de empréstimo modal. Essas opções, acessíveis por um menu de edição (Figura 14), ampliam as possibilidades criativas do usuário.

Figura 14 – Menu de configuração dos acordes



Fonte: autoria própria

Após configurar a harmonia, o usuário pode ouvir o resultado clicando no botão de play, o mesmo utilizado para a melodia, garantindo uma experiência de composição integrada e sem exigir do usuário o conhecimento da formação de acordes para testar suas ideias.

### 5.2.3 Criando e tocando ritmos

Nesta seção, será abordado a criação e a reprodução de ritmos. Logo abaixo do sequenciador de harmonia, a aplicação oferece uma interface simples para selecionar faixas de percussivas, melhorando a dimensão rítmica das músicas criadas.

Figura 15 – Menu de seleção dos ritmos



Fonte: autoria própria

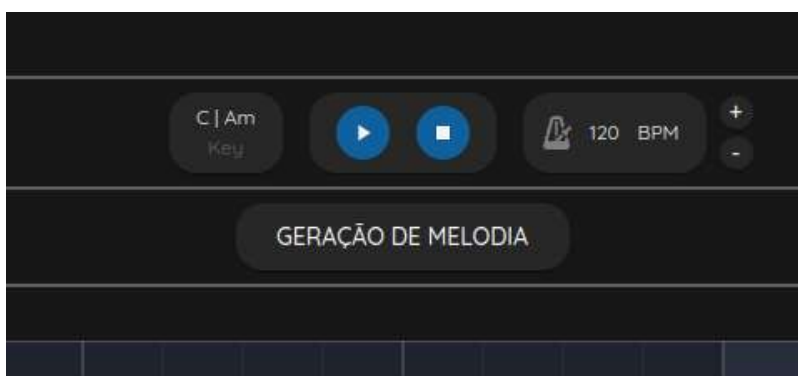
Os usuários podem escolher entre diferentes ritmos pré-definidos, divididos em gêneros como Pop Rock e Pop Ballad, cada um com duas variantes rítmicas. Ao

selecionar um ritmo (Figura 15), ele automaticamente se integra à faixa musical, geralmente com uma base de sons de bateria.

#### 5.2.4 Utilizando o mini gerador de melodia

Na subseção 4.1.4, é apresentado o mini gerador de melodia, uma ferramenta projetada para auxiliar no processo criativo, oferecendo sugestões de trechos melódicos com base em parâmetros definidos pelo usuário.

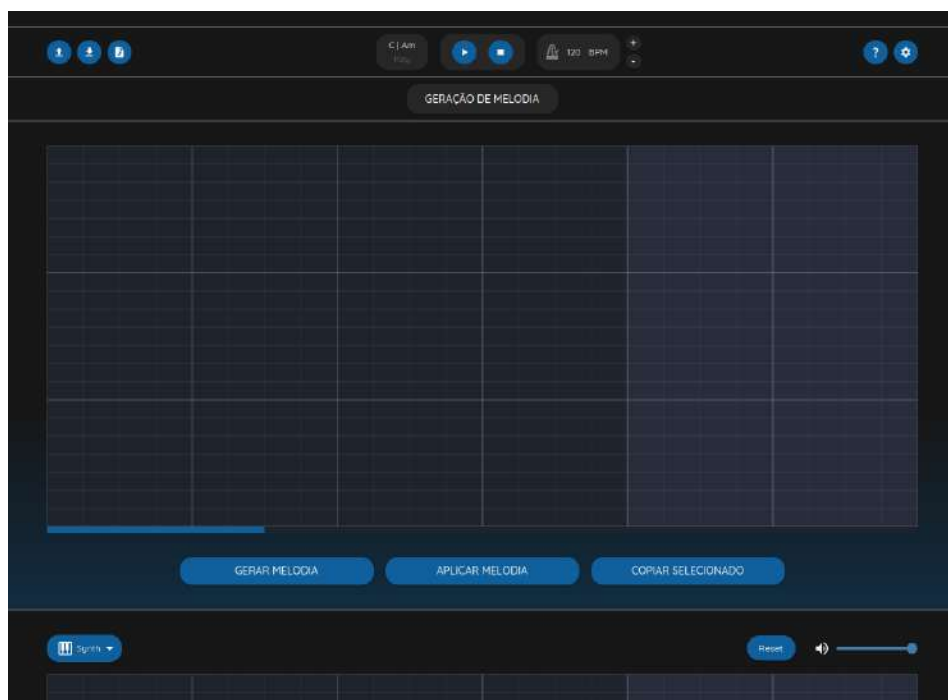
Figura 16 – Botão de geração de melodia



Fonte: autoria própria

O gerador é acessível através de um botão dedicado na parte central da interface da aplicação (Figura 16), nele é permitido realizar experimentações independentes da melodia principal (Figura 17).

Figura 17 – Sequenciador de testes

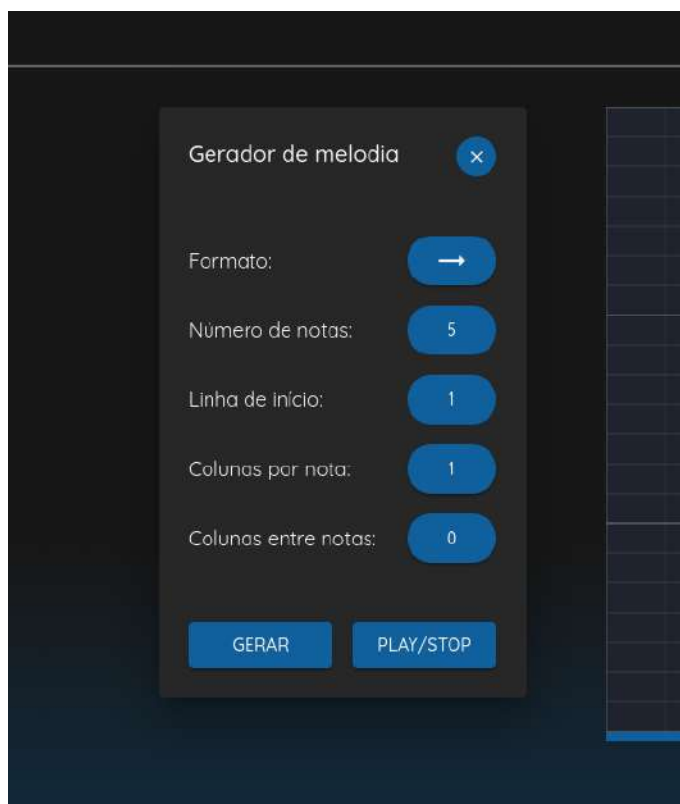


Fonte: autoria própria

Para utilizar a melodia de teste criada no sequenciador padrão basta que o usuário clique no botão de aplicar melodia que está localizado no meio dos três botões inferiores, conforme aparece na Figura 17. A aplicação da melodia no sequenciador padrão pode ocasionar perda de dados caso existam notas nos sequenciadores que se sobreponham, dessa forma para evitar este contratempo foi desenvolvido a funcionalidade copiar um trecho selecionado que é ativada ao clicar no terceiro botão da Figura 17. Por fim, a opção de geração, acionada pelo botão de gerar melodia (Figura 17), irá exibir um menu suspenso para realizar a configuração dos parâmetros de geração. A Figura 18 demonstra as opções disponibilizadas.



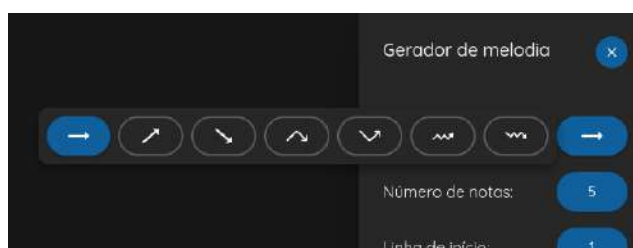
Figura 18 – Configurações de geração de melodia



Fonte: autoria própria

O formato diz respeito à disposição das notas ao longo do sequenciador, isto é, a forma como elas serão organizadas. Neste ponto existem várias alternativas disponíveis como ilustrado na Figura 19:

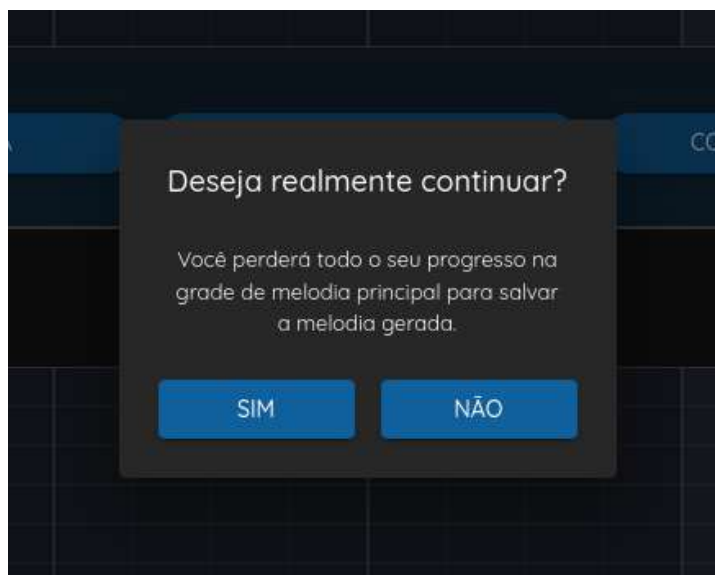
Figura 19 – Opções de formatos de melodia



Fonte: autoria própria

No formato em linha reta a melodia gerada irá caminhar sempre na mesma nota, um exemplo de música que usa desse padrão é o Samba de Uma Nota Só do Tom Jobim, existem ainda outros desenhos como o formato ascendente, descendente e em zigue-zague. Após essa configuração o usuário também poderá definir o número total de notas a ser usado na geração, a linha de início, a quantidade de colunas por nota, que define a duração de cada uma delas, e por fim a quantidade de colunas entre as notas usada para definir os silêncios da melodia. Finalmente, após realizar todas as configurações, basta clicar no botão de gerar que ocorrerá a produção das notas conforme especificado pelas opções. Depois de gerada, também é possível escutar a melodia através do botão “Play/Stop” também mostrado pela Figura 18. Caso o usuário escolha aplicar a melodia gerada ao sequenciador principal, um botão específico realiza essa ação. Se a integração implicar na sobrescrita de dados existentes, um aviso de confirmação é exibido (Figura 20), garantindo que o usuário tenha controle total sobre as mudanças na composição.

Figura 20 – Confirmar sobrescrita

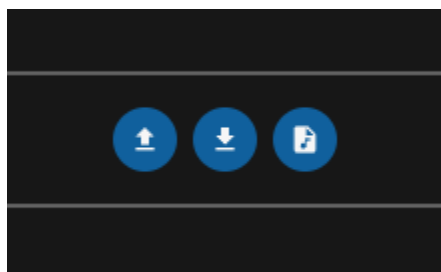


Fonte: autoria própria

### 5.2.5 Realizando exportação e importação de JSON e baixando arquivo MIDI

Na subseção 4.1.5, é detalhado como exportar e importar faixas musicais em formato JSON. Os botões para essas operações estão localizados na parte superior esquerda da aplicação (Figura 21). A importação inicia um processo de seleção de arquivo, enquanto a exportação salva automaticamente a faixa musical no formato JSON na pasta de Downloads do usuário.

Figura 21 – Botões de exportação e importação



Fonte: autoria própria

O primeiro botão da esquerda para direita é o de importação, caso o usuário clique nele será aberto um menu para que ele possa abrir o explorador de arquivos e selecionar o JSON com a faixa musical a ser importada. Após selecionar o arquivo correto basta que o usuário clique no botão de salvar, como ilustrado na Figura 22:

Figura 22 – Salvar importação



Fonte: autoria própria

Para realizar a exportação basta clicar no botão do meio conforme apresentado na Figura 21 e um arquivo no formato JSON será baixado na pasta de Downloads do computador do usuário.

Além do JSON, os usuários também podem baixar suas composições como arquivos MIDI, utilizando o terceiro botão mostrado na Figura 21. Esse recurso facilita a utilização das faixas em outros softwares ou dispositivos compatíveis com MIDI.

## **7. CONSIDERAÇÕES FINAIS**

Nas seções abaixo serão examinadas de perto as conquistas alcançadas por meio do Easy Melody, ressaltando suas características fundamentais que o distinguem de outros produtos semelhantes e explorando oportunidades futuras de melhoria do projeto descrito, a fim de aprimorar ainda mais a solução construída.

### **7.1 Conclusão e Trabalhos Futuros**

O Easy Melody, resultante deste projeto, se estabelece como uma aplicação inovadora para o estudo e a produção de faixas musicais, projetada para ser acessível mesmo a usuários sem conhecimento prévio de teoria musical. Diferenciando-se de outros produtos como Song Maker e ChordChord, destaca-se pelo equilíbrio entre recursos avançados e uma interface simplificada, incentivando a criatividade dos usuários.

A aplicação já publicada oferece sequenciadores musicais para melodia e harmonia, além de um menu para ritmos pré-definidos. Recursos como seleção múltipla de notas, arrasto de retângulos representativos de melodias, criação intuitiva de acordes e manipulação de arquivos MIDI, juntamente com um gerador musical para padrões de melodias, são algumas de suas funcionalidades inovadoras.

Os desafios técnicos enfrentados ao longo do desenvolvimento foram superados, contribuindo para o aprimoramento da ferramenta. O feedback positivo do cliente reflete o sucesso do projeto, indicando sua eficácia e valor. Embora ainda existam funcionalidades a serem implementadas, o Easy Melody já se estabelece como uma ferramenta valiosa e versátil para a criação musical, com planos futuros para expansão e melhorias contínuas, promovendo um impacto significativo na educação musical e na expressão artística.

Para trabalhos futuros no Easy Melody, algumas ideias inovadoras podem ser consideradas:

- Desfazer Comandos (Ctrl + Z): Desenvolver uma funcionalidade de desfazer, permitindo aos usuários reverter ações na ordem inversa em que foram realizadas, aumentando a eficiência e a flexibilidade na edição musical.
- Zoom nos Sequenciadores Musicais: Implementar recursos de zoom nos sequenciadores, proporcionando aos usuários a capacidade de visualizar suas criações musicalmente tanto em detalhes (micro) quanto de forma mais ampla (macro).
- Delimitação de Região de Loop: Adicionar a opção de definir regiões específicas nos sequenciadores para repetição em loop, permitindo aos usuários focar e trabalhar em seções específicas de suas composições.
- Viradas de Bateria Personalizadas: Criar um mecanismo para definir padrões de bateria personalizados, permitindo variações e criatividade rítmica fora dos padrões convencionais.

Além do acréscimo dessas funções citadas acima também é importante haver a validação das que já foram adicionadas a fim de consolidar o que já foi construído e abrir novos horizontes para debilidades que não haviam sido antes percebidas. Uma das maneiras de realizar essa validação seria através de testes de usuários onde seria passada uma tarefa para um usuário da plataforma e ele teria que cumprir o que lhe foi proposto em uma certa faixa de tempo. Após a realização de vários testes com diferentes usuários de variados níveis de conhecimento seria possível definir com clareza quais pontos do trabalho precisam de mais atenção e quais foram atingidos com sucesso.

## **7.2 Limitações**

Assim como em qualquer outro tipo de projeto, é fundamental reconhecer que, embora as aplicações tecnológicas tenham o poder de revolucionar processos e facilitar a vida cotidiana, elas não estão isentas de limitações. Esta seção apresentará algumas das debilidades do projeto descrito neste documento a fim de trazer uma visão mais realista das capacidades e fragilidades desta plataforma. Dentre as vulnerabilidades encontradas pode citar:

- Criação de faixas musicais muito longas: quando ocorre a criação de uma faixa musical muito longa é perceptível que acontecem gargalos na atualização dos estados da interface, o que gera pequenos atrasos, principalmente no indicador visual que serve para mostrar em qual ponto a melodia se encontra.
- Exploração de ritmos: da maneira como a plataforma se encontra hoje não é possível explorar viradas percussivas personalizadas, o que limita o usuário no seu aprendizado rítmico.
- Falta de flexibilidade no sequenciador de harmonia: quando comparado com o sequenciador de melodia é notável que o sequenciador de harmonia não é tão flexível, pois o primeiro apresenta um conjunto maior de funcionalidades que poderiam também ser úteis na criação de harmonias, como selecionar vários acordes, mover eles de posição, dentre outras;
- Importação de MIDI: até a elaboração deste documento a plataforma apenas realiza a exportação da faixa musical em formato MIDI, faltando assim a capacidade de importar músicas nesse formato de arquivo que é largamente utilizado na indústria de produção musical.

## 8. REFERÊNCIAS

DESCOMPLICANDO A MÚSICA. **O que é música?** Teoria Musical |. Disponível em: <<https://www.descomplicandoamusica.com/o-que-e-musica/>>. Acesso em: 26 Aug.

2023.

FEYRE, R. **HOW THE DEMOCRATIZATION OF MUSIC CHANGED THE INDUSTRY**. 2020.

BOAL-PALHEIROS, G. A importância da música no desenvolvimento e na educação das crianças. 2014.

TECLACENTER. **O que é teoria musical? - Blog TeclaCenter o que é a teoria musical e como estudá-la?** Blog TeclaCenter. Disponível em:

<<https://www.teclacenter.com.br/blog/o-que-e-teoria-musical/>>. Acesso em: 26 Aug.

2023.

DESCOMPLICANDO A MÚSICA. **Escalas musicais**. Teoria Musical |. Disponível em: <<https://www.descomplicandoamusica.com/escalas-musicais/>>. Acesso em: 26 Aug.

2023.

DESCOMPLICANDO A MÚSICA. **Intervalo musical**. Teoria Musical |. Disponível em: <<https://www.descomplicandoamusica.com/intervalo-musical/>>. Acesso em: 26 Aug.

2023.

DESCOMPLICANDO A MÚSICA. **Acorde**. Descomplicando a Música |. Disponível em: <<https://www.descomplicandoamusica.com/acorde/>>. Acesso em: 12 Nov. 2023.

ROVEDA, Ugo. **JavaScript: o que é, para que serve e como funciona o JS?** Blog | Kenzie Academy Brasil. Disponível em: <<https://kenzie.com.br/blog/javascript/>>. Acesso em: 26 Aug. 2023.

ROVEDA, Ugo. **TypeScript: o que é, como começar e quais são as vantagens?**

Blog | Kenzie Academy Brasil. Disponível em: <<https://kenzie.com.br/blog/typescript/>>.

Acesso em: 27 Aug. 2023.

MELO, Diego Diego Melo. **O que é Node.js? [Guia para iniciantes] – Tecnoblog**.

Tecnoblog. Disponível em:

<<https://tecnoblog.net/responde/o-que-e-node-js-guia-para-iniciantes/>>. Acesso em: 27

Aug. 2023.



TONEJS. **GitHub - Tonejs/Tone.js: A Web Audio framework for making interactive music in the browser.** GitHub. Disponível em: <<https://github.com/Tonejs/Tone.js>>.

Acesso em: 27 Aug. 2023.

TONALJS. **GitHub - tonaljs/tonal: A functional music theory library for Javascript.**

GitHub. Disponível em: <<https://github.com/tonaljs/tonal>>. Acesso em: 27 Aug. 2023.

ROVEDA, Ugo. **React: o que é, como funciona e porque usar e como aprender.**

Blog | Kenzie Academy Brasil. Disponível em: <<https://kenzie.com.br/blog/react/>>.

Acesso em: 27 Aug. 2023.

LAVRENOV, Anton. **Getting started with react and canvas via Konva.** Konva.js - JavaScript 2d canvas library. Disponível em:

<<https://konvajs.org/docs/react/index.html>>. Acesso em: 27 Aug. 2023.

GALVÃO, Kelven. Guia completo do MobX. **Flutterando**, 2020. Disponível em:

<<https://blog.flutterando.com.br/guia-completo-do-mobx-11d20391428e>>. Acesso em:

27 Aug. 2023.

LLC., Flatlogic. **What is Material UI?** Flatlogic Blog. Disponível em:

<<https://flatlogic.com/blog/what-is-material-ui/>>. Acesso em: 27 Aug. 2023.

MITCHELL, Liam. **What is a sequencer in music? How to use it in your productions.** Native Instruments Blog. Disponível em:

<<https://blog.native-instruments.com/what-is-a-sequencer-in-music/>>. Acesso em: 27

Aug. 2023.

RED HAT. **O que é API REST?** RedHat. Disponível em:

<<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 28 Aug. 2023.

DIO. 3 vantagens de usar REST. **DIO**, 2023. Disponível em:

<<https://www.dio.me/articles/3-vantagens-de-usar-rest>>. Acesso em: 28 Aug. 2023.

DEVMEDIA. Já ouviu falar em Single Page Applications?. DevMedia. Disponível em:

<<https://www.devmedia.com.br/ja-ouviu-falar-em-single-page-applications/39009>>.

Acesso em: 28 Aug. 2023.

MOBX. **Defining data stores · MobX** 🇧🇷. MobX. Disponível em:

<<https://mobx.js.org/defining-data-stores.html>>. Acesso em: 29 Aug. 2023.

ATLASSIAN. **Scrum: o que é, como funciona e como começar.** Atlassian. Disponível

em: <<https://www.atlassian.com/br/agile/scrum>>. Acesso em: 29 Aug. 2023.

SCRUM.ORG. **Scrum Guide**. Scrum Guides. Disponível em:

<<https://scrumguides.org/scrum-guide.html>>. Acesso em: 29 Aug. 2023.

KRIGER, Brunno. **O que é Git e GitHub, para que serve e quais as vantagens?** Blog | Kenzie Academy Brasil. Disponível em: <<https://kenzie.com.br/blog/o-que-e-git/>>.

Acesso em: 30 Aug. 2023.

ATLASSIAN. **O que é Git**. Atlassian. Disponível em:

<<https://www.atlassian.com/br/git/tutorials/what-is-git>>. Acesso em: 30 Aug. 2023.

L., Andrei. O Que é GitHub, Para Que Serve e Como Usar. **Hostinger**, 2019.

Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-github>>. Acesso em: 30 Aug. 2023.

WRIGHT, Gavin. MIDI (Musical Instrument Digital Interface). **TechTarget**, 2023.

Disponível em:

<<https://www.techtarget.com/whatis/definition/MIDI-Musical-Instrument-Digital-Interface>>. Acesso em: 7 Oct. 2023.

LEVINOWITZ, Lili M. The importance of music in early childhood. **General music today**, v. 12, n. 1, p. 4-7, 1998.

ROCHA, Viviane Cristina da; BOGGIO, Paulo Sérgio. A música por uma óptica neurocientífica. **Per musi**, p. 132-140, 2013.

DESCOMPLICANDO A MÚSICA. **Harmonia funcional**. Descomplicando a Música |.

Disponível em: <<https://www.descomplicandoamusica.com/harmonia-funcional/>>.

Acesso em: 9 Nov. 2023.

CHEDIAK, Almir. **Harmonia & Improvisação-Vol. 1**. Irmãos Vitale, 1986.

PRASS, Ronaldo. **Song Maker: saiba criar sua própria música usando o**

**navegador de internet**. Tira-dúvidas de tecnologia, 2018. Disponível em:

<<https://g1.globo.com/tecnologia/blog/tira-duvidas-de-tecnologia/post/song-maker-saiba-criar-sua-propria-musica-usando-o-navegador-de-internet.html>>. Acesso em: 15 Nov. 2023.

DARBY, Megan E. **Challenges to student success in an introductory music theory I course**. 2018. Tese de Doutorado. Walden University.

CHORDCHORD. **ChordChord: Chord Progression Generator & Music Maker**.


Disponível em: <<https://chordchord.com/>>. Acesso em: 15 Nov. 2023.

APPLE. **GarageBand para iOS**. Apple. Disponível em:

<<https://www.apple.com/br/ios/garageband/>>. Acesso em: 16 Dec. 2023.

TECHTUDO. GarageBand. **TechTudo**, Disponível em:

<<https://www.techtudo.com.br/tudo-sobre/garageband/>>. Acesso em: 16 Dec. 2023.

	<b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA</b>
	Campus Campina Grande
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

## Documento Digitalizado Ostensivo (Público)

### TCC

<b>Assunto:</b>	TCC
<b>Assinado por:</b>	Matheus Alves
<b>Tipo do Documento:</b>	Projeto
<b>Situação:</b>	Finalizado
<b>Nível de Acesso:</b>	Ostensivo (Público)
<b>Tipo do Conferência:</b>	Cópia Simples

Documento assinado eletronicamente por:

- Matheus Alves da Silva, ALUNO (201911250021) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 27/01/2024 17:16:30.

Este documento foi armazenado no SUAP em 27/01/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1064190

Código de Autenticação: 04f26e7086

