Cifre.ME:
Gerador de Cifras automáticas.

Denilson Pedro Coutinho da Silva

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

Denilson Pedro Coutinho da Silva

Cifre.ME:

Gerador de Cifras automáticas.

Relatório Técnico apresentado ao curso
Ciência de dados e Inteligência Artificial
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel

Orientador: Yuri de Almeida Malheiros Barbosa

Abril de 2024

Catalogação na publicação
Seção de Catalogação e Classificação

S586c Silva, Denilson Pedro Coutinho da.
Cifre.ME: gerador de cifras automáticas. / Denilson Pedro Coutinho da Silva. - João Pessoa, 2024.
56 f. : il.

Orientação: Yuri Barbosa.
TCC (Graduação) - UFPB/CI.

1. Cifre.ME. 2. Cifras. 3. Acordes. 4. Transcrição de letras. 5. Plataforma web. I. Barbosa, Yuri. II. Título.

UFPB/CI CDU 004.52

Elaborado por Michelle de Kássia Fonseca Barbosa - CRB-738

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência de Dados e Inteligência Artificialintitulado ***Cifre.ME: Gerador de Cifras automáticas.*** de autoria de Denilson Pedro Coutinho da Silva, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba (UFPB)

Prof. Dra. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba (UFPB)

Prof. Dr. Carlos Eduardo Coelho Freire Batista
Universidade Federal da Paraíba (UFPB)

João Pessoa, 26 de abril de 2024

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

Benevolência: é como se chama à disposição de desejar o bem aos outros, demonstrando bondade, generosidade e compaixão. É um comportamento caracterizado pela prática de atos altruístas e pela preocupação genuína com o bem-estar e felicidade dos outros, muitas vezes sem esperar nada em troca.

AGRADECIMENTOS

Gostaria de expressar minha gratidão, em primeiro lugar, a Deus, por ter-me concedido sabedoria e saúde. Esses dons têm sido fundamentais para enfrentar e superar, de forma resiliente, os desafios que a vida me apresenta. Com a cabeça erguida, tenho buscado constantemente alcançar uma melhor qualidade de vida e desenvolver-me como um profissional qualificado. Em seguida, expresso minha gratidão à minha família, cujo apoio e incentivo têm sido constantes ao longo de minha vida e jornada universitária. Eles têm me fornecido não apenas carinho e conselhos, mas também auxílio financeiro e um lar seguro. Expresso também minha profunda gratidão aos meus amigos, os quais considero como irmãos sem exceção. Eles desempenharam um papel fundamental em minha jornada, fornecendo apoio emocional, conselhos valiosos e momentos de descontração, incluindo nossas gameplays avançadas no League of Legends. Sem a presença deles, não teria alcançado os patamares que alcancei. Sua constante motivação e amizade foram essenciais para que eu concluísse o curso da melhor maneira possível. Estou verdadeiramente realizado por ter encontrado essas pessoas incríveis, cuja amizade levarei para toda a vida.

Gostaria também de expressar um agradecimento especial à música, pela sua existência e pelo impacto positivo que trouxe à minha vida. Desde o momento em que comecei a me dedicar mais profundamente à música, tenho percebido melhorias significativas em minha saúde emocional e no meu desempenho nos instrumentos musicais que tanto aprecio, como o violão e a guitarra. A música possui um poder único de renovar as esperanças e proporcionar um alívio para as dificuldades do ser humano. Recomendo a todos aqueles que buscam fugir um pouco da rotina e desejam embarcar em uma jornada de aprendizado e autodescoberta.

Por último, mas definitivamente não menos importante, expresso minha profunda gratidão aos meus Orientadores, Yuri Malheiros e Thaís Gaudêncio. Sua paciência e dedicação nas correções do meu texto foram inestimáveis, e sou grato por terem acreditado no potencial do meu trabalho. Quero que saibam que dediquei meu máximo esforço para corresponder às suas expectativas e para que vocês se orgulhem do resultado final.

E assim, conclui-se uma jornada de seis longos anos, repletos de momentos de alegria, tristeza e um inestimável aprendizado. Ao longo desse período, contei com a companhia de grandes amigos e adquiri conhecimentos significativos, proporcionando-me a base necessária para me tornar um profissional qualificado e preparado para ingressar no mercado de trabalho.

RESUMO

Em uma jornada musical, é comum que pessoas iniciantes enfrentem dificuldades ao tentar tocar suas músicas favoritas. Isso ocorre principalmente devido à complexidade dos acordes, que podem ser desafiadores de executar. Além disso, muitas vezes esses iniciantes se deparam com a frustração de não encontrar a cifra de sua música favorita em plataformas online. Isso acontece porque a cifra em questão ainda não foi submetida por nenhum usuário nessas plataformas, tornando-se assim indisponível para consulta. Neste contexto, o trabalho aborda a implementação do *Cifre.ME*, uma plataforma web cujo principal objetivo é automatizar a geração de cifras musicais, além de auxiliar músicos iniciantes a tocarem suas músicas de maneira simples e intuitiva, por meio de cifras simplificadas. A aplicação utiliza uma API especializada em extrair informações de arquivos de áudio, permitindo a identificação de acordes, a transcrição de letras e até mesmo a geração de *timestamps* para alinhar os acordes conforme a letra da música. Assim, o *Cifre.ME* apresenta três funcionalidades principais. A primeira delas é o *upload de arquivo*, na qual o usuário seleciona uma música de sua preferência nos formatos ".mp3" ou ".webm" para realizar os processamentos necessários a fim de gerar a cifra. Além disso, há a opção de adicionar uma URL do Youtube que contenha a música desejada. Por fim, o usuário também tem a possibilidade de realizar o download da cifra no formato ".txt". Durante o processo de desenvolvimento da aplicação, foram empregadas diversas tecnologias de ponta no campo do desenvolvimento web. Inicialmente, utilizou-se o Figma para criar o protótipo da aplicação, permitindo uma visualização prévia e detalhada de sua estrutura e funcionalidades. Em seguida, para a construção da interface do usuário, optou-se pelo ReactJS, uma ferramenta poderosa que possibilita a criação de interfaces dinâmicas e responsivas. Para a parte servidor da aplicação, a escolha recaiu sobre o FastAPI, uma estrutura ágil e eficiente que permite o desenvolvimento rápido de APIs robustas. Além disso, foi integrada à plataforma a API da Music.AI, responsável por fornecer os dados essenciais para o funcionamento da aplicação, como informações sobre acordes, transcrições de letras e *timestamps* para alinhar os acordes com a letra da música. No que diz respeito à avaliação do produto, elaborou-se e executou-se um plano abrangente de testes internos, com o objetivo de validar o funcionamento de todas as funcionalidades da aplicação.

Palavras-chave: <Cifre.ME>, <Cifras>, <Acordes>, <Transcrição de letras>, <Plataforma Web>.

ABSTRACT

On a musical journey, it is common for beginners to face certain difficulties when trying to play their favorite songs. This mainly occurs due to the complexity of the chords, which can be challenging to execute. Additionally, beginners often encounter the frustration of not finding the chord of their favorite song on online platforms. This happens because the specific chord has not yet been submitted by any user on these platforms, thus becoming unavailable for consultation. In this context, the project addresses the implementation of *Cifre.ME*, a web platform whose main objective is to automate the generation of musical chords, as well as assist beginner musicians in playing their songs in a simple and intuitive way through simplified chords. The application uses an API specialized in extracting information from audio files, allowing for chord identification, transcription of lyrics, and even generating timestamps to align the chords with the song lyrics. Therefore, *Cifre.ME* presents three main functionalities. The first one is the file upload, where the user selects a preferred song in ".mp3" or ".webm" formats for the necessary processing to generate the chord. Additionally, there is an option to add a YouTube URL containing the desired song. Finally, the user also has the possibility to download the chord in ".txt" format. During the application development process, various cutting-edge web development technologies were employed. Initially, Figma was used to create the application prototype, allowing for a preview and detailed visualization of its structure and functionalities. Then, for the user interface construction, ReactJS was chosen, a powerful tool enabling the creation of dynamic and responsive interfaces. For the server-side of the application, FastAPI was chosen, an agile and efficient framework enabling the rapid development of robust APIs. Furthermore, the Music.AI API was integrated into the platform, responsible for providing essential data for the application's operation, such as chord information, lyric transcriptions, and timestamps to align the chords with the song lyrics. Regarding the product evaluation, a comprehensive internal testing plan was devised and executed to validate the functioning of all application functionalities.

Keywords: <Cifre.ME>, <Cifras>, <Chords>, <Transcription of lyrics>, <Web Platform>.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

- API - *Application Programming Interface*
- BPM - Batidas por minuto
- CSS - *Cascading Style Sheets*
- DOM - *Document Object Model*
- HTML - *HyperText Markup Language*
- JSON - *Javascript Object Notation*
- UI - *User Interface*
- URL - *Uniform Resource Locator*
- UX - *User Experience*

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da tecnologia, especialmente no campo da música, foram desenvolvidas e aperfeiçoadas várias técnicas que fazem uso de modelos de inteligência artificial. Essas técnicas têm a capacidade de extrair uma gama diversificada de informações de uma música, incluindo seus acordes, tonalidade e BPM, além de identificar sons de instrumentos específicos. Isso tem causado uma verdadeira revolução na maneira como os músicos interagem com a música, oferecendo-lhes valiosos recursos para suas composições. Plataformas como o *Music.Ai*[1] têm se destacado nesse cenário, fornecendo ferramentas que facilitam todo o processo criativo. No entanto, é importante destacar que essas ferramentas não substituem a criatividade humana.

O sistema *Music.Ai* utiliza algoritmos e modelos de aprendizagem de máquina para extrair informações de músicas. Os dados extraídos são disponibilizados através da API *Music.Ai*, que contém uma variedade de módulos especializados em processamentos de áudio. A mesma foi fundamental no desenvolvimento da solução proposta neste trabalho, o *Cifre.ME*, pois fornece todas as informações necessárias para a criação das cifras da música.

O avanço da tecnologia também impulsionou a busca por plataformas online de ensino musical. Nesse cenário, o "Cifra Club[2]" se destaca, oferecendo uma variedade de cursos que abrangem desde violão para iniciantes, até teoria musical avançada. Essas plataformas proporcionam facilidade de acesso, permitindo que os indivíduos explorem seus interesses musicais no conforto de suas casas, ajustando seus horários de estudo de acordo com sua disponibilidade.

Entretanto, é importante reconhecer que algumas limitações podem existir nessas plataformas. Por exemplo, as cifras das músicas disponibilizadas muitas vezes são criadas manualmente, exigindo a contribuição direta dos usuários para adicionar letras e acordes. Nesse contexto, surge o *Cifre.ME* como uma alternativa inovadora. Diferentemente das plataformas convencionais, no *Cifre.ME* tudo acontece de forma automática, tornando o processo de criação de cifras mais ágil e eficiente.

Dessa forma, o *Cifre.ME* se destaca pela sua eficiência na automação do processo de criação das cifras, oferecendo cifras automáticas de maneira simplificada. Essa abordagem visa auxiliar especialmente pessoas iniciantes a aprenderem suas músicas favoritas de forma mais acessível e prática. Essa característica o torna uma ferramenta valiosa para aqueles que estão dando os primeiros passos na jornada musical, oferecendo um recurso eficaz para aprimorar suas habilidades e desfrutar da experiência de tocar suas músicas preferidas.

[1] https://music.ai/
[2] https://www.cifraclub.com.br/

1.1 Objetivo geral

Desenvolver o *Cifre.ME*, uma plataforma web responsiva com o intuito de reconhecer os acordes e a letra de uma música selecionada pelo usuário. Com essa informação, o sistema é capaz de gerar uma cifra simplificada, visando o aprendizado musical, especialmente para iniciantes.

1.2 Objetivos específicos

- Gerar cifras de maneira automática a partir de uma música.
- Comparar os resultados obtidos com as cifras do "Cifra Club".
- Garantir que a plataforma seja compatível com diversos dispositivos, permitindo que o usuário acesse o produto a partir do seu computador e smartphone.

1.3 Estrutura do documento técnico

O trabalho está dividido em cinco capítulos. Na primeira é apresentada uma introdução ao tema, assim como o problema e seus objetivos geral e específicos. O segundo capítulo apresenta os conceitos gerais para compreensão do trabalho realizado. O terceiro capítulo apresenta detalhes de como a aplicação *Cifre.ME* foi desenvolvida, bem como a metodologia utilizada para o desenvolvimento do mesmo. No quarto capítulo serão apresentados os testes e análises dos resultados obtidos. No quinto capítulo vem a conclusão e futuras melhorias que poderão ser incorporadas no *Cifre.ME*.

2 CONCEITOS GERAIS

Neste capítulo serão detalhados os principais conceitos e ferramentas usadas que servem de base para construção do *Cifre.ME*

2.1 HTML (HyperText Markup Language)

O HTML é uma linguagem de marcação que tem facilitado a construção de páginas Web desde 1991, mantendo-se relevante até hoje na sua versão 5. Com uma variedade de recursos e funcionalidades, o HTML 5 traz melhorias significativas para o desenvolvimento web (MOURA; HENRIQUE, 2024).

Sua principal função é demarcar a estrutura de uma página web por meio de *tags* específicas. Estas *tags* são utilizadas para o navegador identificar e definir diversos elementos, tais como imagens, títulos, parágrafos, links, entre outros.(MOURA; HENRIQUE, 2024) Assim, para garantir que um documento seja interpretado corretamente pelo navegador, é essencial que o arquivo tenha a extensão ".html". A partir dessa extensão, o conteúdo poderá ser visualizado em qualquer navegador web. No entanto, o HTML apenas demarca os elementos na interface. Assim para desenvolver uma página web por completo é necessário o uso de outras tecnologias para aplicar os estilos nas páginas e adicionar interatividade. Essas tecnologias são o CSS (*Cascading Style Sheets*), detalhado na Seção 2.2 e o *Javascript*, descrito na Seção 2.3.

2.2 CSS

O CSS desempenha um papel crucial na definição dos estilos das páginas Web, atribuindo estilos visuais essenciais, como cores, tamanhos de fontes e disposição dos elementos na tela e até mesmo responsividade, ou seja a adaptação da página em tablets, smartphones e computadores. Essa capacidade é fundamental para criar interfaces que cativem a atenção do usuário e proporcionem uma experiência agradável de navegação em qualquer tipo de dispositivo seja ele *desktop* ou móvel (NEVES, 2023a).

Por meio de seletores, classes, propriedades e pseudo-classes, o CSS permite referenciar as *tags* HTML Seção 2.1, aplicando estilos específicos a cada elemento. Essa abordagem garante que a interface esteja alinhada com o design estabelecido para o produto, mantendo uma consistência visual em toda a aplicação (NEVES, 2023a).

Além disso, o CSS oferece a possibilidade de adicionar efeitos interativos, como animações e mudanças visuais em resposta à interação do usuário com o mouse. Esses recursos podem tornar a experiência de usuário ainda mais envolvente e dinâmica. Com o *Javascript* Seção 2.3, por sua vez, é possível criar mudanças visuais dinâmicas dentro

das tags HTML, enriquecendo as capacidades de interatividade proporcionadas pelo *Javascript* junto ao CSS. Dessa forma, a combinação dessas tecnologias possibilita a criação de interfaces mais ricas e funcionais.

2.3 Javascript

O *Javascript* (JS) é uma linguagem de programação bastante utilizada na Web (FLANAGAN, 2012). Possui uma grande presença em sites modernos, já que todos os navegadores atuais têm a capacidade de interpretá-lo (SILVA, 2010). Isso resulta em uma relevância significativa no cenário da Web.

Além disso, o *Javascript* faz parte da denominada tríade da Web, que são tecnologias essenciais para o desenvolvimento Web. Enquanto o HTML conceituado na Seção 2.1 é utilizado para especificar o conteúdo das páginas Web e o CSS (Seção 2.2) formata sua apresentação visual, o *Javascript* (Seção 2.3) aparece para definir o comportamento dinâmico dessas páginas.

JS também é considerada uma linguagem de programação de alto nível, dinâmica, interpretada, não tipada e multiparadigma (FLANAGAN, 2012). Foi desenvolvida em uma colaboração entre a Netscape e a Sun Microsystems, no ano de 1995, tendo como objetivo fornecer interatividade a uma página web (SILVA, 2010).

A viabilização da interatividade em uma página Web é essencialmente mediada pelo DOM (*Document Object Model*), um conjunto de objetos que se organiza em uma estrutura em forma de árvore, herdando toda a estrutura do HTML da página (PIMENTEL, 2022). Ao ser carregada, o navegador constrói integralmente essa DOM, e é por meio dela que o *JavaScript* se conecta ao HTML, possibilitando a inclusão de elementos interativos nas páginas Web (PIMENTEL, 2022).

Nesse cenário, surgiram diversos *frameworks* e bibliotecas, como é o caso do *ReactJS* (Seção 2.4). Fundamentados em *JavaScript*, esses *frameworks* e bibliotecas empregam o conceito da DOM para efetuar manipulações de forma eficaz e otimizada, impulsionando assim a performance das aplicações web (PIMENTEL, 2022).

2.4 ReactJS

O React é um *framework* Javascript criado pelo Facebook (atualmente Meta), amplamente adotato para construção de interfaces de usuário (UI), ou seja, o desenvolvimento do *frontend*(vide Seção 2.6) (NEVES, 2023b). Sua popularidade no cenário do desenvolvimento web se justifica pela gama de recursos e ferramentas que oferece, simplificando e tornando mais eficiente o trabalho no *frontend* (NEVES, 2023b). Entre esses recursos, destacam-se as bibliotecas de componentes prontos para uso, desenvolvidas por terceiros

e disponibilizados em gerenciadores de pacotes como o NPM (*Node Package Manager*) ou *Yarn*.

Ao empregar o conceito de componentes, que são pequenos blocos de código destinados a representar partes específicas da interface do usuário em páginas web, o React oferece um recurso altamente performático e bem visto no ponto de vista do *Clean code* (Código Limpo): a reutilização de código. Isso significa que a interface é fragmentada em vários componentes, os quais podem ser utilizados em qualquer parte da interface sem a necessidade de reescrever o código correspondente (NEVES, 2023b). Essa abordagem não apenas promove a modularidade e a organização do código-fonte, mas também simplifica o processo de desenvolvimento, aumentando a eficiência e facilitando a manutenção do sistema.

Quando o usuário interage com o sistema, realizando ações como clicar em um botão o estado dos componentes é atualizado e as mudanças são refletidas na (UI) (Seção 2.5). Isso é feito através de funções de *callbacks*, as quais são chamadas quando uma ação é executada por um usuário.

Além disso, o React faz uso do chamado Virtual DOM (*Document Object Model Virtual*), uma representação em memória da interface do usuário que se assemelha à DOM, como explicado na Seção 2.3. Quando ocorrem atualizações nos estados dos componentes, o Virtual DOM também é atualizado, comparando-se com a DOM para identificar as alterações necessárias na UI alterando apenas a parte da UI que mudou (NEVES, 2023b). Esse processo demonstra uma eficiência notável em relação à atualização direta da DOM, como era feito anteriormente à existência dos *frameworks* Javascript.

No entanto, o React é especializado no desenvolvimento de interfaces de software. Porém, para alcançar uma interface consistente, é fundamental contar com um protótipo bem elaborado da aplicação. Nesse sentido, é recomendável utilizar softwares voltados para design, como o Figma, que será abordado com mais detalhes na próxima seção.

2.5 Figma

A criação de produtos digitais tem experimentado um crescimento significativo nos últimos anos, impulsionado pela crescente popularização da área de *User Experience* (UX), conhecida pelo acrônimo “Experiência do Usuário”(VILLAIN; SILVEIRA, 2023). A mesma busca identifica problemas que afetam a experiência do usuário, por meio de análises, pesquisas e testes realizados diretamente com os mesmos. O objetivo é proporcionar uma experiência confortável e intuitiva, durante o uso do sistema para o público-alvo do produto, garantindo assim sua aceitação e eficácia no mercado digital.

Simultaneamente, complementando a área de UX, temos a *User Interface* (UI), também conhecida pelo acrônimo “Interface do Usuário”. Esta vertente é responsável

por converter os resultados obtidos nos testes realizados com os usuários em interfaces digitais tangíveis. Dessa forma, fornece um meio para que esses usuários interajam de forma eficiente com o produto, facilitando a conclusão de seus objetivos de forma intuitiva e satisfatória.(VILLAIN; SILVEIRA, 2023)

Neste contexto, destaca-se o Figma, uma ferramenta que possui a capacidade de criar interfaces para diversos tipos de produtos digitais, como aplicativos e sites. O Figma permite que os designers desenvolvam todo o fluxo, estrutura e composição do projeto, de forma integrada e colaborativa, possibilitando uma abordagem mais eficiente e dinâmica na criação de interfaces digitais, que atendam às necessidades do usuário e aos objetivos do produto (VILLAIN; SILVEIRA, 2023). Além de oferecer recursos como o *Dev Mode*[3] recentemente lançada essa ferramenta é capaz de apontar algumas propriedades CSS (vide Seção 2.2) no design da interface facilitando assim o desenvolvimento das mesmas.

No entanto, as interfaces desenvolvidas em softwares dedicados ao design como é o caso do Figma, desempenham um papel crucial no desenvolvimento de produtos. Elas são implementadas no *Frontend* (Seção 2.6) dos produtos tornando realidade tudo que foi planejado anteriormente. A integração entre design e *frontend* é essencial para garantir que a experiência do usuário atenda as expectativas e necessidades estabelecidas durante o projeto.

2.6 *Frontend*

De forma geral, o *frontend* representa a interface gráfica de aplicações (OMAUROSOUTO, 2023). Essa parte essencial do sistema é responsável pela interação direta entre o usuário e o sistema, permitindo que este envie requisições para o *backend* (Seção 3.3.2) utilizando suas funcionalidades. Ela é formada pela junção do HTML (vide Seção 2.1), CSS (vide Seção 2.2) e Javascript (vide Seção 2.3), tecnologias consideradas bases da Web, especialmente os dois primeiros (OMAUROSOUTO, 2023).

Para garantir o pleno funcionamento de uma aplicação web, é preciso estabelecer uma integração entre o *frontend* e o *backend*. Como mencionado anteriormente, o *frontend* constitui a interface gráfica, enquanto o *backend*, que será abordado em detalhes na seção subsequente, desempenha o papel de servidor da aplicação. Nessa dinâmica, a interface se conecta ao servidor por meio de requisições e o servidor responde a interface geralmente no formato de *JSON*.

[3]https://www.figma.com/blog/introducing-dev-mode/

2.7 *Backend*

O *backend* é o servidor responsável pela gestão de todas as requisições da aplicação Web (OMAUROSOUTO, 2023). Quando um cliente interage com a aplicação através do *frontend* (vide Seção 2.6), todas as solicitações são encaminhadas para o *backend*. Este último é encarregado de processar todas as informações recebidas e fornecer respostas à aplicação, geralmente no formato *JSON*, amplamente utilizado para comunicação entre aplicações.

No contexto do *Cifre.ME* o *backend* utilizou uma API externa para fornecer os dados necessários para o *Frontend* (vide Seção 2.6). Através da integração com a Music.AI que será detalhada na seção seguinte.

2.8 Music.AI

A Music.AI[4] é uma API (*Application Programming Interface*) criada pela empresa *moises.ai* na qual é especializada em extrair informações de áudios, tais como acordes (vide Seção 2.11), letras, *timestamps* de início e término dos acordes, entre outros. Utilizando algoritmos e inteligência artificial ela consegue identificar informações relevantes contidas em um arquivo de áudio.

A Music.AI emprega o conceito de *Módulos*, *Workflows* e *Jobs* para lidar com o processamento de áudio. Os Módulos são blocos de construção individuais que oferecem funcionalidades ou processamentos específicos. Em outras palavras, cada módulo desempenha uma tarefa claramente definida, como, por exemplo, a transcrição de áudio. Os *Workflows*, por sua vez, constituem-se como sequências de módulos conectados que são executados em uma ordem predefinida, formando um *pipeline* de processamento. Esses *Workflows* são altamente flexíveis, pois permitem encadear sequências exclusivas de módulos para alcançar o resultado desejado, de forma eficaz e personalizada. Por fim, os *Jobs* representam as solicitações para processar um ou mais arquivos de áudio utilizando um *Workflow* específico, consolidando assim o processo de manipulação de áudio dentro da API.

Para uma compreensão mais aprofundada das técnicas empregadas pela Music.AI na captura de informações de uma música, é fundamental possuir um conhecimento básico sobre conceitos como Notas Musicais (conferir Seção 2.9), Escalas Musicais (consultar Seção 2.10) e Acordes (verificar Seção 2.11). Todas essas informações serão explicadas nas próximas seções, proporcionando uma compreensão mais abrangente do funcionamento da Music.AI e de seu papel no contexto do *Cifre.ME*

[4]https://music.ai/docsgetting-started/introduction/

2.9 Notas musicais

Uma nota musical é caracterizada por um som singular, que ressoa de maneira constante em uma determinada afinação (DANTAS; CRUZ, 2019). O conceito de afinação refere-se à calibração do som em uma frequência específica. No contexto da música ocidental, existem sete notas naturais, que são: DÓ, RÉ, MI, FÁ, SOL, LA e SI, cada uma delas possuindo frequências distintas.

Essas notas naturais podem ser modificadas por meio de alterações de tom, conhecidas como acidentes musicais. Existem dois tipos principais de acidentes: o sustenido, representado pelo símbolo #, e o bemol, representado pelo símbolo **b**. O sustenido indica um **aumento de meio tom** na nota original, enquanto o bemol indica uma **diminuição de meio tom** (NETO, 2021).

Por exemplo, se tomarmos a nota DÓ e a elevamos meio tom, ela se torna DÓ#. Por outro lado, se baixarmos a nota RÉ meio tom, ela se torna RÉb. Na Figura 1 segue a representação das notas naturais e acidentadas.

Figura 1: Notas musicais naturais e acidentadas.
Fonte:
<https://polimatadigital.medium.com/notas-musicais-introdução-à-teoria-2dcd6b0bb8d0>

As notas musicais desempenham um papel fundamental na construção das escalas musicais e melodias sendo abordada com detalhes na seção subsequente. A partir dessas notas, é possível criar uma variedade de escalas musicais e melodias enriquecendo a harmonia das composições musicais.

2.10 Escala musical

Ao conhecer as notas musicais (ver Seção 2.9), é possível compreender a base para a formação das escalas musicais, as quais desempenham um papel fundamental na criação de harmonias e melodias, em uma ampla variedade de gêneros musicais (ZALANDAUSKAS, 2023). Uma escala, por sua vez, consiste em um conjunto de notas dispostas em uma ordem particular, cuja estrutura é definida pela tonalidade específica da composição em questão.(ZALANDAUSKAS, 2023) Existem diversas escalas musicais no mundo da música, porém vamos abordar aqui as mais simples que são a escala maior e a menor.

A escala maior, composta por sete notas distintas, segue uma ordem precisa, determinada pela fórmula específica da escala maior, que utiliza a abordagem de tons e

semitons para sua construção (ZALANDAUSKAS, 2023). Tanto o tom, quanto o semitom, são medidas fundamentais para classificar a distância entre duas notas musicais. O tom, sendo a distância mais comum, ocorre entre a maioria das notas, enquanto o semitom corresponde à metade de um tom, representando a menor distância entre dois sons considerados como notas musicais (NETO, 2021). A fórmula para a construção da escala maior é representada por **T - T - sT - T - T - T - sT**, na qual "T" denota um tom e "sT" um semitom, com a progressão iniciando sempre na nota desejada para a montagem da escala (ZALANDAUSKAS, 2023).

Na Figura 1 segue a tabela completa da escala maior natural.

Escala	**T**	**T**	**ST**	**T**	**T**	**T**	**ST**
C	**D**	**E**	**F**	**G**	**A**	**B**	**C**
C#	**D#**	**F**	**F#**	**G#**	**A#**	**C**	**C#**
D	**E**	**F#**	**G**	**A**	**B**	**C#**	**D**
Eb	**F**	**G**	**Ab**	**Bb**	**C**	**D**	**Eb**
E	**F#**	**G#**	**A**	**B**	**C#**	**D#**	**E**
F	**G**	**A**	**Bb**	**C**	**D**	**E**	**F**
F#	**G#**	**A#**	**B**	**C#**	**D#**	**F**	**F#**
G	**A**	**B**	**C**	**D**	**E**	**F#**	**G**
Ab	**Bb**	**C**	**Db**	**Eb**	**F**	**G**	**Ab**
A	**B**	**C#**	**D**	**E**	**F#**	**G#**	**A**
Bb	**C**	**D**	**Eb**	**F**	**G**	**A**	**Bb**
B	**C#**	**D#**	**E**	**F**	**G**	**A**	**Bb**

Tabela 1: Escala maior

De forma análoga, a escala menor é formada por sete notas musicais distintas e segue o conceito de tons e semitons, embora sua fórmula apresente uma pequena variação. A fórmula da escala menor é representada por **T - sT - T - T - sT - T - T**.(ZALANDAUSKAS, 2023)

Na Tabela 2 segue a tabela completa da escala menor natural.

As sete notas musicais contidas, tanto na escala maior, quanto na menor, correspondem a cada um dos graus da escala, indo do primeiro ao sétimo grau. Após o sétimo grau, completa-se uma **oitava**, ou seja, a nota retorna a si mesma, porém com frequências diferentes, podendo ser mais **aguda** ou mais **grave**. Quando a nota é mais aguda, podemos afirmar que está uma oitava acima; de maneira análoga, quando a nota é mais grave, dizemos que está uma oitava abaixo (CONCEITO.DE, 2024).

Tomando como base as escalas musicais maior e menor, torna-se possível a construção de uma ampla variedade de acordes, sejam eles do tipo Tríade ou Tétrade. A seguir, será apresentada uma explicação detalhada sobre o que é um acorde e como ele é formado.

Escala	T	ST	T	T	ST	T	T
C	**D**	**Eb**	**F**	**G**	**Ab**	**Bb**	**C**
C#	**D#**	**E**	**F#**	**G#**	**A**	**B**	**C#**
D	**E**	**F**	**G**	**A**	**Bb**	**C**	**D**
Eb	**F**	**Gb**	**Ab**	**Bb**	**B**	**Db**	**Eb**
E	**F#**	**G**	**A**	**B**	**C**	**D**	**E**
F	**G**	**Ab**	**Bb**	**C**	**Db**	**Eb**	**F**
F#	**G#**	**A**	**B**	**C#**	**D**	**E**	**F#**
G	**A**	**Bb**	**C**	**D**	**Eb**	**F**	**G**
Ab	**A#**	**B**	**Db**	**Eb**	**E**	**F#**	**Ab**
A	**B**	**C**	**D**	**E**	**F**	**G**	**A**
Bb	**C**	**Db**	**Eb**	**F**	**Gb**	**Ab**	**Bb**
B	**C#**	**D**	**E**	**F#**	**G**	**A**	**B**

Tabela 2: Escala menor

2.11 Acordes

Um acorde é o conjunto de três ou mais notas tocando simultaneamente, ou seja, as notam ressoam em conjunto.(NETO, 2021) Podem ser classificados de duas formas distintas: **Tríades** e **Tétrades**, detalhados nas próximas seções.

Para a classificação dos acordes, são utilizadas as notas das escalas musicais maior e menor (ver Seção 2.10), bem como os intervalos, que representam as distâncias entre as notas musicais de tons e semitons e a posição da nota em relação à escala. Estes intervalos são classificados como: **M** (maior), **m** (menor), **J** (justo), **A** (aumentado) que também pode ser simbolizado como (+) e **D** (diminuto).(FROZZA, 2019)

A Tabela 3 apresenta a classificação de todos os possíveis intervalos entre duas notas musicais mostrando sua nomenclatura, representação e a distância em tons e semitons.

Intervalos	Símbolo	Tons
Segunda menor	**2^{a}m**	**meio tom**
Segunda maior	**2^{a}M**	**1 tom**
Terça menor	**3^{a}m**	**1 tom e meio**
Terceira maior	**3^{a}M**	**2 tons**
Quarta justa	**4^{a}J**	**2 tons e meio**
Quarta aumentada	**4^{a}A**	**3 tons**
Quinta diminuta	**5^{a}D**	**3 tons**
Quinta justa	**5^{a}J**	**3 tons e meio**
Sexta menor	**6^{a}m**	**4 tons**
Sexta maior	**6^{a}M**	**4 tons e meio**
Sétima menor	**7^{a}m**	**5 tons**
Sétima maior	**7^{a}M**	**5 tons e meio**
Oitava	**8^{a}**	**6 tons**

Tabela 3: Intervalos harmônicos
Fonte: Adaptado da tabela de (NETO, 2021).

Com os intervalos definidos e conhecidos, conforme ilustrado na Tabela 3, e as escalas musicais apresentadas nas Tabelas 1 e 2, torna-se possível formar os acordes. Para as tríades, utilizamos as notas do primeiro, terceiro e quinto grau da escala. Já para as tétrades, basta adicionar o sétimo grau da escala.(FROZZA, 2019)

A Tabela 4 será tomada como referência para formar os acordes, de acordo com a classificação do seu intervalo.

T	**2^{a}m**	**2^{a}M**	**3^{a}m**	**3^{a}M**	**4^{a}j**	**5^{a}m**	**5^{a}J**	**$+5^{a}$**	**6^{a}M**	**7^{a}m**	**7^{a}M**
C	C#	D	D#	E	F	F#	G	G#	A	A#	B
D	D#	E	F	F#	G	G#	A	A#	B	C	C#
E	F	F#	G	G#	A	A#	B	C	C#	D	D#
F	F#	G	G#	A	A#	B	C	C#	D	D#	E
G	G#	A	A#	B	C	C#	D	D#	E	F	F#
A	A#	B	C	C#	D	D#	E	F	F#	G	G#
B	C	C#	D	D#	E	F	F#	G	G#	A	A#

Tabela 4: Acordes e intervalos de suas notas.
Fonte: Adaptado da tabela de (FROZZA, 2019).

2.11.1 Acordes Tríades

Uma tríade é formada pela sobreposição de duas terças, o que resulta em um acorde composto por três sons distintos (NETO, 2021). A nota fundamental, também conhecida como tônica, desempenha um papel crucial nessa estrutura, não apenas nomeando o acorde, mas também sendo a nota mais grave. As terças podem ser de dois tipos: maior

ou menor, e essa distinção determina se o acorde será classificado como maior ou menor. Além disso, a quinta é adicionada à estrutura, formando assim uma tríade (NETO, 2021).

As tríades, essenciais na teoria musical, são classificadas de acordo com suas características distintas, o que resulta em várias formas diferentes: **Acorde maior**, **Acorde menor**, **Acorde diminuto** e **Acorde aumentado** (FROZZA, 2019). Essas classificações refletem variações nas combinações de notas e intervalos, influenciando diretamente na sonoridade e na emoção transmitida pela música. A Tabela 5 ilustra as diferentes formas de tríades, oferecendo uma representação visual para uma compreensão mais clara da estrutura da sua estrutura.

Nome do acorde	Estrutura
Acorde Maior	T + 3^{a}M + 5^{a}J
Acorde Menor	T + 3^{a}m + 5^{a}J
Acorde Diminuto	T + 3^{a}m + 5^{a}D
Acorde Aumentado	T + 3^{a}M + 5^{a}A

Tabela 5: Acordes Tríades.
Fonte: Adaptado da tabela de (NETO, 2021).

O Acorde maior é uma das formas mais comuns de tríade e é caracterizado pela presença da **tônica**, **terça maior** e **quinta justa**. Por exemplo, ao formar o acorde de DÓ maior (C), precisamos das notas C (tônica), E (terça maior) e G (quinta justa), que juntas compõem o acorde de DÓ maior (FROZZA, 2019).

O acorde menor é caracterizado pela presença da **tônica**, **terça menor** e **quinta justa**. Por exemplo para formar o acorde de DÓ menor (Cm), precisamos das notas C (tônica), D# (terça menor) e G (quinta justa), que juntas compõem o acorde de DÓ menor (FROZZA, 2019).

O acorde aumentado é formado pela **tônica**, **terça maior** e **quinta aumentada**. Por exemplo, para montar o acorde de C aumentado (Caug ou C+), precisamos das notas C (tônica) , E (terça maior) e G# (quinta aumentada), que juntas compõem o acorde de DÓ aumentado (FROZZA, 2019).

Por último, o acorde diminuto é formado pela **tônica**, **terça menor** e **quinta menor**, ou **quinta diminuta**. Por exemplo, para montar o acorde de DÓ diminuto (C^{o} ou Cdim), precisamos das notas C (tônica), D# (terça menor) e F# (quinta diminuta), que juntas compõem o acorde de DÓ diminuto (FROZZA, 2019).

2.11.2 Acordes Tétrades

As tétrades possuem uma sonoridade mais encorpada e brilhante devido ao acréscimo da sétima nota da escala musical (ver Seção 2.10). São bastante utilizadas no Rock

progressivo, Jazz, Blues, Bossa Nova e são formados pela superposição de terças, ou seja, formando um acorde de quatro sons: uma nota fundamental (**tônica**) e terças sobrepostas sobre elas (NETO, 2021). São classificadas de várias formas distintas, sendo elas: **Acorde Maior com Sétima Maior**, **Acorde Maior com Sétima Menor**, **Acorde Menor com Sétima Menor**, **Acorde Meio Diminuto**, **Acorde Suspenso com Sétima Menor**, **Acorde Maior com Sexta Maior**, **Acorde Menor com Sexta Maior** e **Acorde Aumentado Tétrade**.(NETO, 2021)

O **Acorde Maior com Sétima Maior** é geralmente representado por **X7M ou X7+**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça maior**, **quinta justa** e **sétima maior** (NETO, 2021).

O **Acorde Maior com Sétima Menor** é geralmente representado por **X7**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça maior**, **quinta justa** e **sétima menor** (NETO, 2021).

O **Acorde Menor com Sétima Menor** é geralmente representado por **Xm7**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça menor**, **quinta justa** e **sétima menor** (NETO, 2021).

O **Acorde Meio Diminuto** é geralmente representado por **Xm7(b5)**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça menor**, **quinta diminuta** e **sétima menor** (NETO, 2021).

O **Acorde Suspenso com Sétima Menor** é representado por **Xsus7**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **quarta justa**, **quinta justa** e **sétima menor** (NETO, 2021).

O **Acorde Maior com Sexta Maior** é representado por **X6** onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça maior**, **quinta justa** e **sexta maior** (NETO, 2021).

O **Acorde Menor com Sexta Maior** é representado por **Xm6**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça menor**, **quinta justa** e **sexta maior** (NETO, 2021).

E, por último, o **Acorde Aumentado** é representado por **X7M(#5)**, onde "X" é a tônica que dá o nome ao acorde. É formado pela **tônica**, **terça maior**, **quinta aumentada** e **sétima maior** (NETO, 2021).

Todos esses acordes podem ser formados seguindo a Tabela 6 no qual apresenta estrutura de um Acorde Tétrade.

Nome do acorde	Estrutura
Acorde Maior com Sétima Maior	T + 3^{a}M + 5^{a}J + 7^{a}M
Acorde Maior com Sétima Menor	T + 3^{a}M + 5^{a}J + 7^{a}m
Acorde Menor com Sétima Menor	T + 3^{a}m + 5^{a}J + 7^{a}m
Acorde Meio Diminuto	T + 3^{a}m + 5^{a}D + 7^{a}m
Acorde Suspenso com Sétima Menor	T + 4^{a}J + 5^{a}J + 7^{a}m
Acorde Maior com Sexta Maior	T + 3^{a}M + 5^{a}J + 6^{a}M
Acorde Menor com Sexta Maior	T + 3^{a}m + 5^{a}J + 6^{a}M
Acorde Aumentado Tétrade	T + 3^{a}M + 5^{a}A + 7^{a}M

Tabela 6: Acordes Tétrades
Fonte: Adaptado da tabela de (NETO, 2021).

Os acordes são essenciais para o alcance do objetivo principal do *Cifre.ME*, que consiste na geração automatizada de cifras. Para atingir esse propósito, são necessários acordes no formato de Tríades (Seção 2.11.1) e Tétrades (Seção 2.11.2), essenciais para a formação das cifras detalhadas na Seção 2.12.

2.12 Cifra

A cifra é uma forma de notação musical amplamente utilizada para representar os acordes em uma composição musical. Essa notação consiste no nome dos acordes, podendo ser visualizada acima das letras das músicas (JESUS, 2017). Além de simplesmente destacar os acordes, as cifras são posicionadas em lugares específicos ao longo da letra da música, indicando o momento exato em que esses acordes devem ser executados. Essa forma de notação musical, através de cifras, é especialmente benéfica para músicos iniciantes, pois oferece uma abordagem mais acessível e direta para a compreensão e execução das músicas. Comparada à tablatura, que pode ser mais complexa e requer um entendimento mais profundo da teoria musical para ser compreendida, as cifras fornecem uma maneira mais simples de visualizar e tocar os acordes. Isso é especialmente útil para aqueles que estão começando a aprender música e ainda estão se familiarizando com os conceitos básicos da teoria musical (JESUS, 2017).

Na Figura 2, é apresentado um exemplo da cifra da música “Tempo Perdido” da banda Legião Urbana. Estão detalhados todos os acordes utilizados na música, bem como a letra correspondente e a disposição dos acordes em relação à letra. Essa representação facilita a compreensão e a execução da música, permitindo que músicos interpretem corretamente os acordes no contexto das letras.

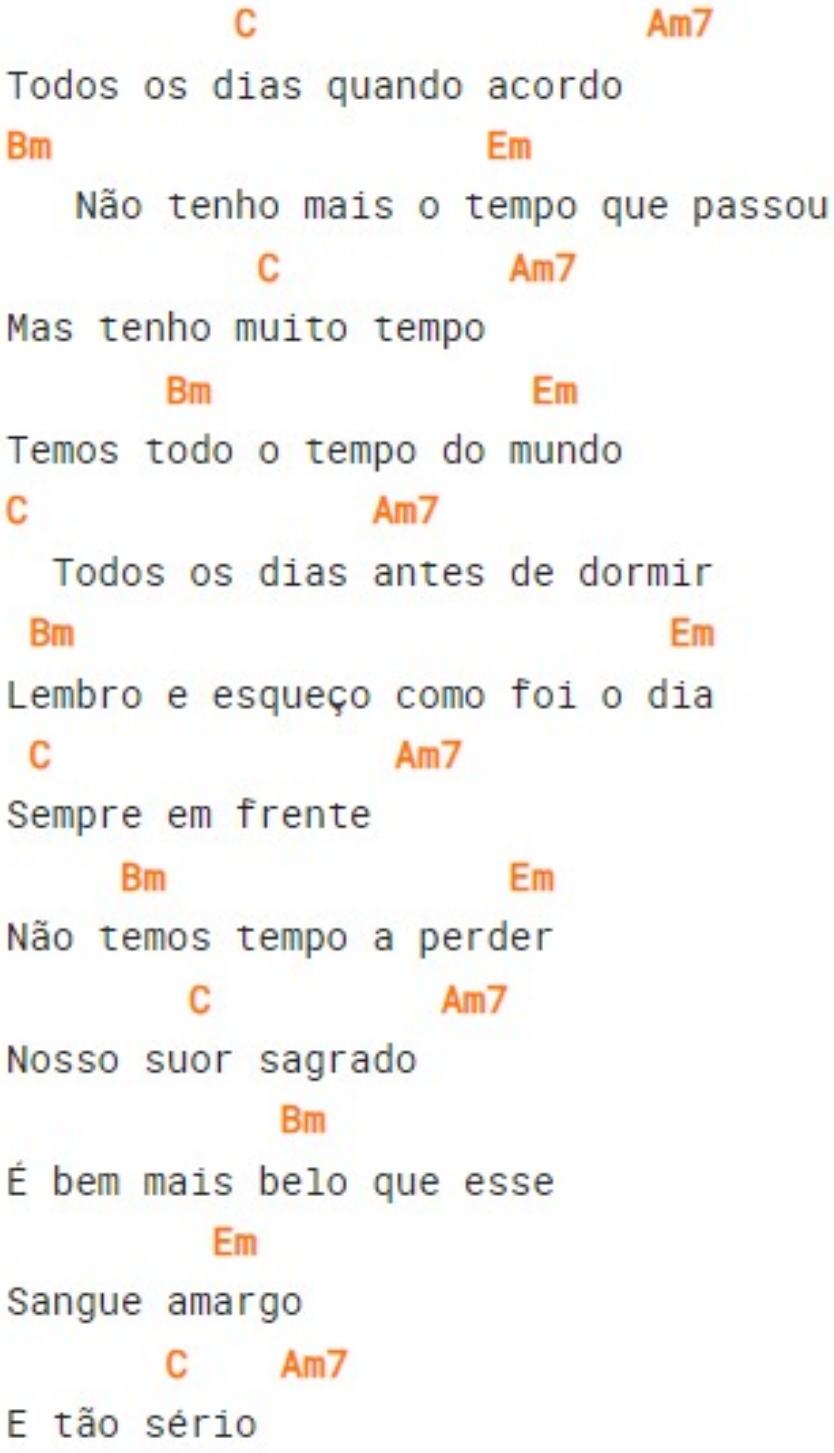

Figura 2: Cifra da música Tempo perdido
Fonte: Cifra Club

3 METODOLOGIA

Neste capítulo, apresentaremos em detalhes a metodologia adotada para a criação da plataforma web denominada *Cifre.ME*. Essa plataforma foi desenvolvida com o propósito de proporcionar suporte a músicos iniciantes, auxiliando-os no reconhecimento dos acordes em uma composição musical.

No âmbito deste estudo, serão elucidadas todas as fases do desenvolvimento da plataforma, desde a concepção inicial, até a conclusão de sua implementação. Dessa forma, buscamos oferecer uma compreensão abrangente de todo o processo de criação, destacando as escolhas metodológicas, os desafios enfrentados e as soluções aplicadas em cada etapa do desenvolvimento do *Cifre.ME*.

A metodologia teve início com uma análise detalhada dos módulos da API, com o objetivo de compreender seu funcionamento e identificar quais seriam aplicáveis ao *Cifre.ME*. Posteriormente, foi iniciada a fase de prototipação da aplicação, durante a qual todas as funcionalidades foram mapeadas e o *layout* foi desenvolvido. Após essa etapa, o *frontend* da aplicação foi implementado, utilizando como base o *layout* desenvolvido anteriormente.

Com o *frontend* concluído, deu-se início ao desenvolvimento do *backend*. Na próxima etapa, realizou-se o processamento das cifras para posicionar os acordes corretamente em relação às palavras. Após a conclusão de todas essas etapas, o *Cifre.ME* foi construído.

O fluxograma apresentado na Figura 3 apresenta as etapas seguidas na metodologia.

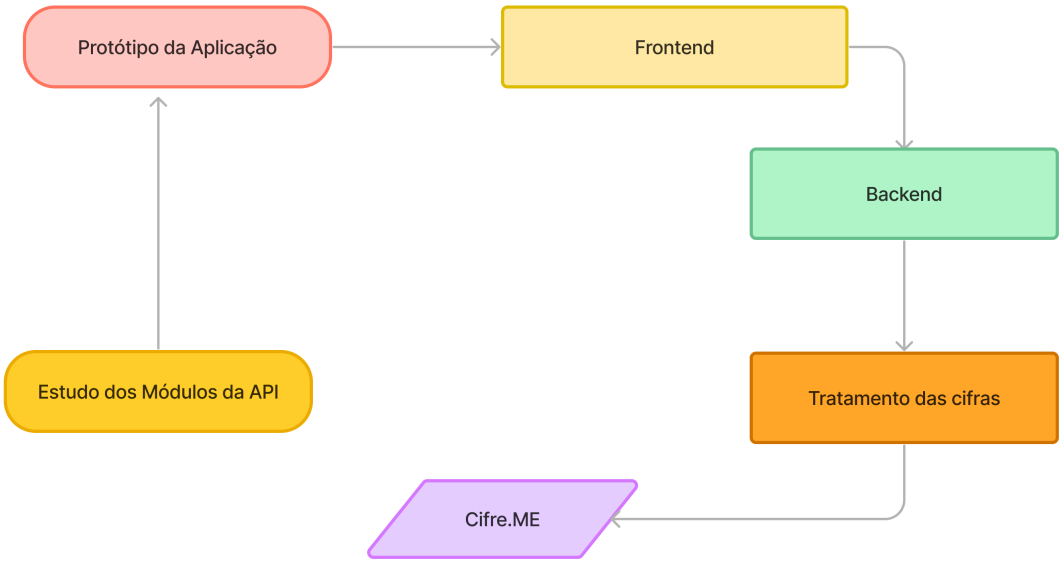

Figura 3: Fluxograma que apresenta a Metodologia
Fonte: Autoria Própria

3.1 Módulos da API

A Music.AI é uma API composta por diversos módulos desenvolvidos para extrair informações de áudio. Dentre esses módulos, destacam-se a separação de faixas, permitindo a extração de áudios de instrumentos específicos em uma determinada música. Além disso, o módulo de mixagem possibilita a combinação de vários canais de áudio, proporcionando a personalização dos volumes de cada canal individualmente e a masterização, outro componente essencial, personaliza a profundidade dos bits da música, aprimorando sua qualidade final.

Antes de iniciar efetivamente o processo de desenvolvimento da aplicação, uma análise detalhada de todos esses módulos foi conduzida, utilizando a documentação da Music.AI como guia. O objetivo desse estudo era decidir quais desses módulos seriam incorporados na construção do produto final.

Após uma avaliação, a escolha recaiu sobre o módulo de transcrição, uma vez que ele englobava integralmente a visão inicial da aplicação. Essa decisão foi respaldada pelo fato de que todas as funcionalidades desejadas, como o reconhecimento de acordes, a transcrição da letra da música e a detecção da linguagem, estavam encapsuladas nesse módulo específico.

3.1.1 *Transcription*

Conforme descrito anteriormente, foi utilizado o módulo de transcrição da API. Esse módulo, por sua vez, é composto por diversos submódulos, cada um encarregado de funções específicas. Estes incluem o reconhecimento de acordes na música, a sincronização dos tempos de início e término de cada acorde e de cada frase, bem como a transcrição precisa da letra da música, descritas nas próximas subseções.

3.1.2 *Lyrics*

Dentro do contexto do módulo de transcrição, é importante destacar que este submódulo específico desempenha um papel crucial ao transcrever e alinhar as letras de qualquer arquivo de áudio. Sua função principal é converter o conteúdo cantado para uma forma textual compreensível. Além disso, tem a capacidade de identificar a língua à qual a música pertence.

Para compreender mais detalhadamente o funcionamento técnico desse submódulo, ele opera retornando um *array* de objetos. Cada objeto contém a representação de uma frase da música, e dentro de cada frase, são listadas as palavras individuais. Cada palavra é acompanhada de seus respectivos tempos de início e término, todos sincronizados de

acordo com o tempo total da música. Este processo é realizado durante o processamento do áudio pela API, assegurando uma transcrição precisa e alinhada com a temporalidade da obra musical. Essa abordagem técnica reforça a eficiência do submódulo, ao fornecer informações detalhadas e estruturadas sobre o conteúdo do áudio.

Na Figura 4, é apresentada a representação do retorno deste submódulo para a aplicação. Na imagem, destacam-se as características do *array* de objetos que compõem as frases da música. No contexto desse *array*, o atributo *start* indica o momento de início da frase, *end* representa o tempo final, *text* corresponde à frase completa, e a estrutura *words* engloba cada palavra da frase, incluindo seus tempos de início e fim, além da pontuação de confiança *score*.


```
{
  "start": 15.183,
  "end": 17.225,
  "text": "Pra que falar",
  "words": [
    {
      "word": "Pra",
      "start": 15.183,
      "end": 15.304,
      "score": 0.485
    },
    {
      "word": "que",
      "start": 15.384,
      "end": 15.544,
      "score": 0.841
    },
    {
      "word": "falar",
      "start": 15.564,
      "end": 17.225,
      "score": 0.771
    }
  ],
  "language": "portuguese"
},
```


Figura 4: Retorno do Módulo Lyrics
Fonte: Music.AI

3.1.3 *Chords*

O módulo *Chords* é encarregado de extrair todos os acordes presentes na música, juntamente com seus respectivos *timestamps*, que indicam o momento em que cada acorde é tocado.

Analisando sob uma perspectiva mais técnica, o módulo retorna um *array* de objetos que compreende não apenas os acordes em sua forma mais complexa, mas também inclui informações sobre variações específicas de cada acorde. Esta abordagem abrangente oferece uma representação detalhada da estrutura harmônica da música.

É importante ressaltar que, considerando o propósito primário do *Cifre.ME* em auxiliar músicos iniciantes, foi feita a escolha de fornecer os acordes na forma simplificada. Esta decisão visa facilitar o processo de aprendizado dos usuários, uma vez que acordes simplificados são mais acessíveis e amigáveis para aqueles que estão começando a explorar o universo musical. Assim, ao optar pela forma simplificada, o produto busca garantir uma experiência de aprendizado mais suave e eficiente para seus usuários iniciantes.

Abaixo, na Figura 6, é representada a estrutura do *Chords*, no qual o *Chords Map* retorna os acordes da música e o *Root Key* representa a tonalidade daquela determinada música.

Figura 5: Módulo Chords
Fonte: Music.AI

Ao Considerar a Figura 6, deparamo-nos com a representação dos dados provenientes do submódulo de Acordes. Nessa visualização, o *Chords map* apresenta os acordes da música em diversas formas, acompanhados de seus respectivos tempos de início e fim. Adicionalmente, o *Root key* destaca a tonalidade predominante da composição musical.

```
{
  "start": 0.78,
  "end": 2.4,
  "chord_majmin": "C#:maj",
  "chord_complex_jazz": "C#",
  "chord_simple_jazz": "C#",
  "chord_complex_pop": "C#",
  "chord_simple_pop": "C#",
  "bass": null
},
{
  "start": 2.4,
  "end": 4.02,
  "chord_majmin": "D#:maj",
  "chord_complex_jazz": "D#",
  "chord_simple_jazz": "D#",
  "chord_complex_pop": "D#",
  "chord_simple_pop": "D#",
  "bass": null
},
```


Figura 6: Módulo Chords
Fonte: Music.AI

3.2 Protótipo da Aplicação

O protótipo da aplicação foi desenvolvido no Figma pelo designer Vinicius Santos. Reconhecendo a crescente preferência dos usuários por dispositivos móveis, a responsividade do site foi identificada como um requisito crucial. Nesse sentido, todo o layout da aplicação foi concebido de maneira altamente flexível, assegurando um desempenho eficaz, tanto na Web, quanto em dispositivos móveis. Essa abordagem visa proporcionar uma experiência consistente e otimizada, independentemente do dispositivo utilizado pelo usuário.

A etapa de prototipação se revela de extrema importância, pois é nela que as funcionalidades do sistema são mapeadas, considerando as particularidades de cada uma, assim como questões de usabilidade e acessibilidade. Em outras palavras, uma prototipação bem elaborada desempenha um papel crucial na oferta de conforto e facilidade para o usuário final ao utilizar as funcionalidades do sistema. Sendo assim, ao abordar bem esses elementos na fase de prototipação, é possível criar uma experiência satisfatória para o usuário.

3.3 Cifre.ME

O *Cifre.ME* é uma plataforma web desenvolvida para auxiliar músicos iniciantes a reconhecer os acordes de uma música, representando-os em cifras, um sistema de notação musical utilizado para indicar os acordes a serem executados por um instrumento musical. Além disso, é mais intuitiva do que a partitura, sendo, portanto, mais indicada para iniciantes na música.

A plataforma proporciona ao usuário duas funcionalidades. A primeira opção permite realizar o *upload* de arquivos de áudio com extensões .mp3 ou .webm, por meio de um *dropzone*, componente bastante utilizado em *uploads* de arquivo devido à sua intuitividade. Já a segunda alternativa possibilita adicionar uma URL do Youtube contendo a música cujos acordes se deseja reconhecer.

Ao concluir o processo, o usuário terá à sua disposição uma cifra completa. Nessa cifra, será possível visualizar, tanto os acordes, quanto a letra da música. Além disso, o usuário terá a opção de executar a música em seu instrumento musical, proporcionando uma experiência prática e imersiva. Caso prefira, há também a possibilidade de baixar a cifra no formato .txt, oferecendo praticidade e flexibilidade para acessar e utilizar as informações da música, conforme suas necessidades.

3.3.1 Frontend

Após a conclusão das fases de estudo dos módulos da API, conforme detalhado na Seção 3.1, e da prototipação, conforme discutido na Seção 3.2, deu-se inicio à fase de desenvolvimento do Frontend. Nessa etapa, o foco é a construção da interface da aplicação, na qual tudo que foi concebido durante a prototipação começou a ser efetivamente implementado na tela do usuário.

Durante esse processo, todos os componentes da página foram cuidadosamente desenvolvidos, estabelecendo a base para a representação visual dos elementos previstos na fase de prototipação. Subsequentemente, foram criadas as páginas, mesmo que ainda não houvesse a integração completa com o Backend que será citado na seção seguinte. Essa abordagem sequencial permitiu uma progressão estruturada do projeto, concentrando-se inicialmente no design e na estrutura visual da interface.

Somente após a conclusão dessa etapa de construção da interface é que se avançou para a integração funcional com o Backend. Essa metodologia adotada proporciona uma vantagem significativa, pois permite a identificação e resolução de possíveis problemas de forma mais eficiente, garantindo um desenvolvimento organizado.

O Frontend foi desenvolvido utilizando a bliblioteca *ReactJS*, amplamente empregada em sites e compatível com a maioria dos navegadores. Além disso, é importante

destacar que essa tecnologia oferece flexibilidade, pois dispõe de diversos pacotes com componentes previamente desenvolvidos. Esses componentes, por sua vez, podem ser facilmente integrados à interface de maneira intuitiva. Para isso, basta empregar um gerenciador de pacotes de escolha, como o *Yarn* ou *Npm*, e realizar a instalação do pacote desejado. Essa abordagem não apenas agiliza o processo de desenvolvimento, mas também amplia consideravelmente as opções disponíveis, permitindo uma personalização eficiente e adaptável da interface, conforme as necessidades específicas do projeto.

3.3.2 Backend

O Backend foi implementado utilizando o FastAPI, um *framework* do Python utilizado para desenvolvimento de APIs RestFull. Destaca-se sua capacidade de gerar automaticamente a documentação da API, bastando apenas adicionar algumas *flags* durante a instanciação do mesmo. A documentação gerada pelo *framework* é facilitada pelo Swagger, uma ferramenta amplamente empregada na documentação de APIs RestFull, sendo especialmente intuitiva para que os usuários compreendam, de maneira clara, os requisitos da API.

A comunicação entre o Backend e o Frontend é estabelecida por meio de *endpoints*, que são URLs que abrigam uma API capaz de receber e processar consultas direcionadas ao banco de dados. No contexto específico do *Cifre.ME*, um dos *endpoints* disponíveis apresenta-se com a URL *https://gregorylira-ytb-dlp-muses.hf.space/send_audio?url*. Esse *endpoint* desempenha uma função crucial, convertendo a URL recebida como parâmetro em um arquivo .mp3.

Em termos mais claros, este *endpoint* é responsável por realizar o *download* do vídeo vinculado à URL fornecida pelo usuário e, posteriormente, converter o conteúdo para o formato .mp3. Este *endpoint* é acionado apenas caso o usuário opte pela funcionalidade associada à URL.

O segundo *endpoint*, *https://gregorylira-ytb-dlp-muses.hf.space/send_audio*, equivale ao anterior, contudo, a distinção fundamental reside no fato de que, neste caso, a API aguarda a inclusão de um arquivo no formato *FormData* no corpo da requisição. O *FormData* é uma estrutura de dados versátil que permite o armazenamento eficaz de pares chave-valor. Esta abordagem é especialmente conveniente ao lidar com arquivos no contexto do JavaScript, sendo uma prática comumente adotada.

Em ambas as situações, o Backend, após a conclusão bem-sucedida de todas as etapas, iniciará uma sequência ordenada de chamadas. Inicialmente, uma requisição do tipo GET será efetuada na seguinte URL: *https://api.music.ai/api/upload*. O propósito dessa solicitação reside na obtenção da URL de *upload*, imprescindível para direcionar o arquivo previamente recebido do usuário pelo Backend até a API da Music.AI. Após essa

etapa, torna-se essencial a criação de um "workflow" diretamente no console da Music.AI. Vale ressaltar que "workflow" é um acrônimo que representa "Fluxo de trabalho" , indicando que todos os processamentos nos arquivos de áudio serão conduzidos dentro desse contexto estruturado. Este "workflow" , por sua vez, engloba *jobs* exclusivos, ou seja, cada processamento realizado dentro dele constituirá um *job* distinto. Suponhamos que seja realizado o processamento de três músicas distintas, cada uma dessas operações será identificada como um *job* único.

Após a conclusão da etapa de criação do *workflow*, o Backend realizará um *POST* na seguinte URL: *https://api.music.ai/api/job*. Esse *endpoint* específico é projetado para receber diversos parâmetros no corpo da requisição. Entre esses parâmetros, incluem-se o nome do *job*, ao qual são atribuídos identificadores (IDs) únicos, o *workflow*, que representa o nome do *workflow* anteriormente criado no console da Music.AI, e um objeto chamado "params".

Dentro desse objeto "params", destaca-se a presença do *inputUrl*. Neste campo específico, enviamos a URL obtida antes no *endpoint https://api.music.ai/api/upload*. Após a conclusão dos processamentos, é fornecido um ID que referencia aquele *job* específico. Para acessar os dados obtidos durante o processamento do áudio, é essencial realizar uma última consulta no *endpoint https://api.music.ai/api/job/:id*. Neste caso, o ID, que é retornado após o término do *job*, será passado como parâmetro na rota. Essa abordagem permite obter os dados relacionados ao mesmo, proporcionando uma visão abrangente e detalhada do resultado do processamento de áudio.

3.4 Tratamento das cifras

Foram realizadas diversas manipulações nos dados, com o propósito de alinhar precisamente o acorde com a palavra em que é tocado, levando em consideração o tempo da música. O módulo *Lyrics* oferece informações, incluindo o tempo inicial e final da frase, cada palavra dentro da frase com seus respectivos tempos de início e término, além do *score*, que representa a confiança na precisão daquela palavra. Já o módulo *Chords* fornece o tempo de início e fim associado a cada acorde. Diante dessas relações estabelecidas por cada módulo, torna-se possível calcular o exato momento em que o acorde é executado em relação à palavra, permitindo definir o intervalo específico em que o acorde é tocado, respeitando os limites temporais da frase.

O processo de cálculo do intervalo do acorde é executado da seguinte maneira: primeiramente, percorremos o *array* de objetos retornado pelo módulo *Lyrics* e, simultaneamente, iteramos sobre o *array* de frases, que é retornado dentro de cada objeto desse módulo. Durante esse percurso, realizamos uma busca (*find*) no objeto dos acordes, o qual é retornado pelo módulo *Chords*. Este procedimento envolve duas condições essenciais.

A primeira condição verifica se o tempo de início da palavra está dentro do intervalo de tempo do acorde. Em outras palavras, essa parte da condição é avaliada como verdadeira, se o início da palavra coincide com o momento de início, ou ocorre após o início do acorde, mas antes do seu término.

A segunda condição verifica se o tempo de término da palavra está dentro do intervalo de tempo do acorde. Nesse caso, a condição é verdadeira se o término da palavra ocorrer após o início do acorde e, simultaneamente, antes ou no mesmo instante do término do acorde.

Em resumo, verificamos se há sobreposição no tempo entre o acorde e a palavra. A condição será verdadeira, em geral, se parte ou todo o período da palavra coincidir com o período do acorde, seja no início, no final ou em ambos. A seguir, na Figura 7 é apresentado o resultado obtido após a aplicação dos tratamentos na cifra. Nesse contexto, o *array* inclui tanto a palavra, quanto o acorde, sendo estes tocados durante a execução da respectiva palavra.

Figura 7: Dados após os tratamentos realizados na cifra
Fonte: Autoria Própria

3.5 Requisitos funcionais

Nesta seção serão detalhados todos os requisitos funcionais do sistema, ou seja, todas as funcionalidades que o usuário poderá utilizar durante o uso da plataforma.

3.5.1 [RF01] Upload de arquivo

A funcionalidade permite que o usuário faça o *upload* de um arquivo de áudio, com extensão .mp3 ou .webm. O Backend é responsável por receber esse arquivo e encaminhá-lo para a API do Music.AI.

3.5.2 [RF02] Inserção de URL

A funcionalidade oferece ao usuário a opção de adicionar um URL do YouTube que contenha o vídeo de uma música de sua preferência. O Backend é encarregado de converter esse vídeo para os formatos .mp3 ou .webm e enviar o resultado para a API do Music.AI.

3.5.3 [RF03] Exibição da Cifra

As cifras geradas serão exibidas de acordo com a música selecionada pelo usuário. Nessa exibição, serão apresentados a tonalidade da música, os acordes e a letra da canção.

3.5.4 [RF04] *Download* da Cifra

Ao clicar no botão “Baixar Cifra”, o usuário receberá um arquivo .txt contendo os acordes e a letra da música de seu interesse.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste capítulo serão detalhadas todas as telas do *Cifre.ME*, acompanhadas dos respectivos planos de testes elaborados para validar suas funcionalidades. Além disso, será apresentada uma descrição dos casos de testes aplicados, visando assegurar a robustez e eficácia do produto. Em seguida, será realizada uma análise comparativa com plataformas similares, destacando os pontos fortes e áreas de melhoria. Por fim, será conduzida uma discussão aprofundada dos resultados obtidos ao longo do desenvolvimento do produto, consolidando as descobertas e lições aprendidas durante o processo.

4.1 Apresentação das telas

A seguir serão apresentadas todas as telas da aplicação que foram desenvolvidas de forma simples Figuras (8, 9, 10) , para que o usuário consiga utilizar as funcionalidades da maneira mais intuitiva possível.

Figura 8: Página Inicial
Fonte: Autoria Própria

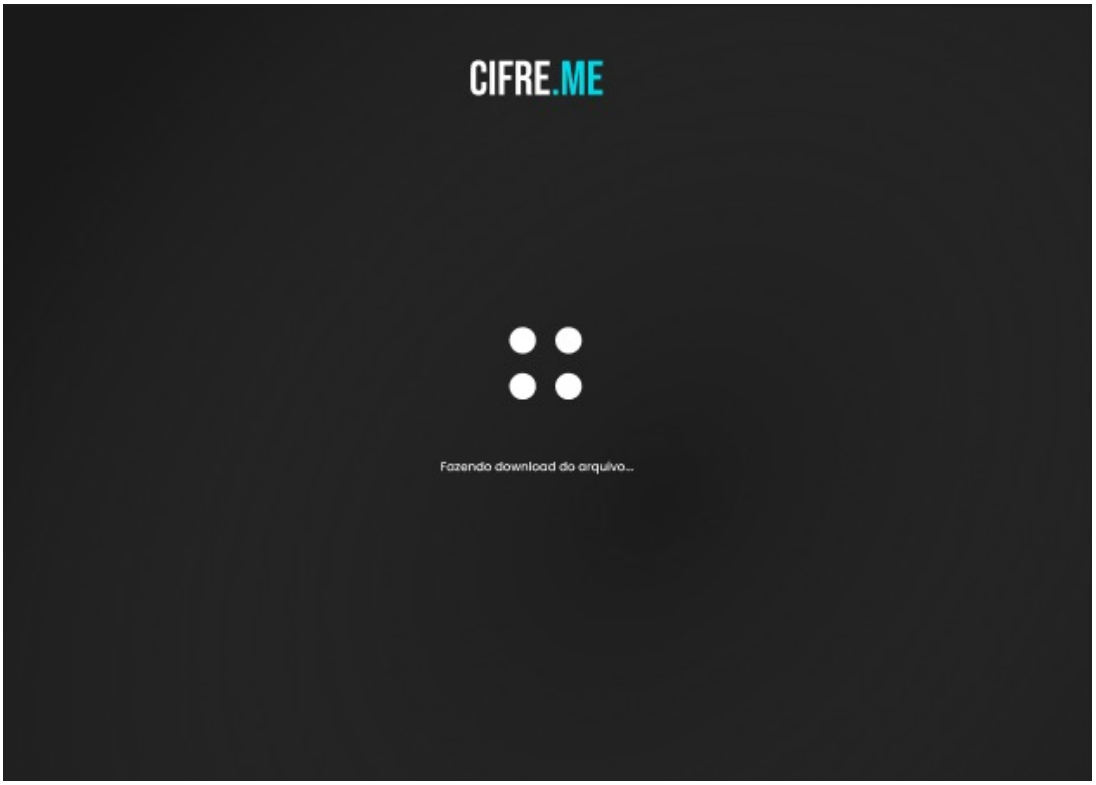

Figura 9: Página de Loading
Fonte: Autoria Própria

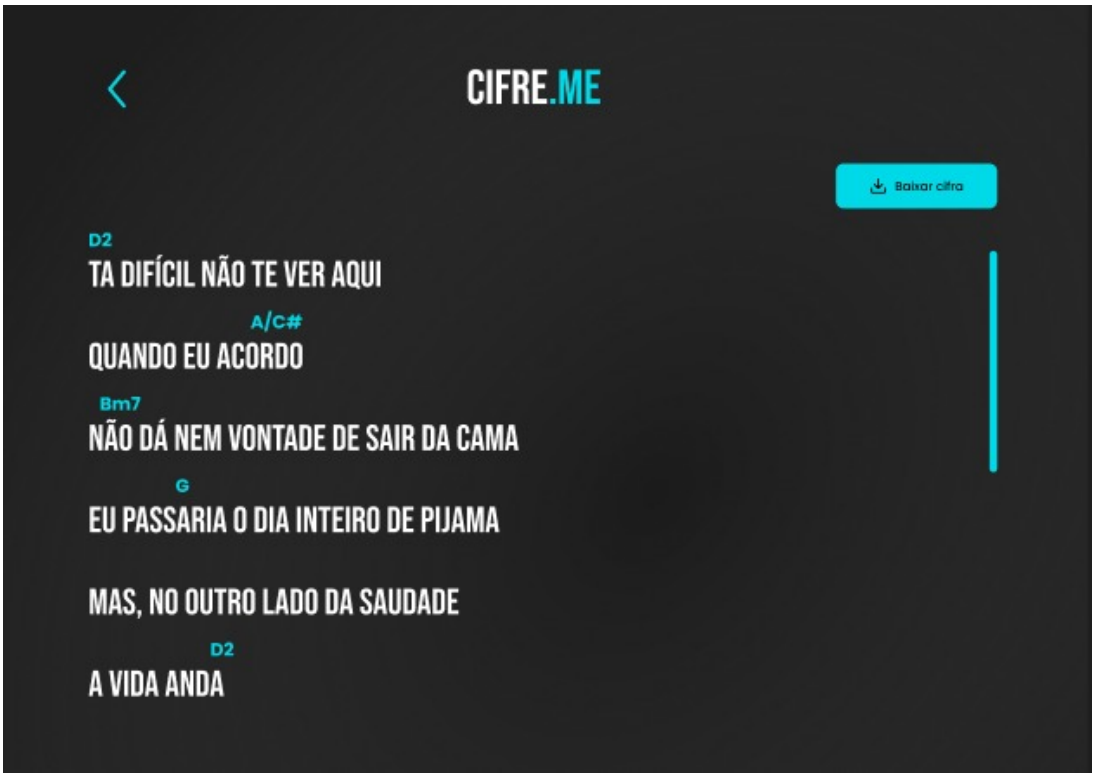

Figura 10: Página da Cifra
Fonte: Autoria Própria

Na figura 8, apresenta-se a *homepage* da aplicação que corresponde à página prin-

cipal do *Cifre.ME*. Nesta tela, o usuário tem a possibilidade de realizar o *upload* de um arquivo de áudio ou inserir uma URL do Youtube, com o objetivo de gerar a cifra da música selecionada cumprindo os requisitos funcionais (3.5.1 e 3.5.2).

Na figura 9, é exibida a tela de carregamento da aplicação, a qual desempenha a função de *fallback* quando o usuário submete um arquivo de áudio ou a URL do Youtube para que os processamentos necessários sejam executados.

Por fim, na figura 10, é exibida a cifra gerada após o envio do arquivo de áudio ou da URL. Nessa tela, o usuário tem a possibilidade de visualizar a cifra da música e também de efetuar o seu download, bastando clicar no botão "Baixar cifra". E com isso cumpre-se os requisitos funcionais (3.5.3 e 3.5.4).

4.2 Plano de testes

A seguir, serão apresentados o plano de testes da plataforma web *Cifre.ME*, acompanhado de seu planejamento. Essas estratégias de teste foram elaboradas visando a detecção de eventuais problemas na plataforma que possam impactar diretamente em seu funcionamento. Com essa abordagem, será possível identificar e resolver questões de falhas, usabilidade e desempenho, garantindo assim uma experiência consistente e satisfatória para os usuários.

Os testes têm como principal objetivo explorar todas as funcionalidades disponibilizadas pela plataforma, reproduzindo potenciais interações dos usuários. A abordagem adotada acompanhou o fluxo natural da aplicação, iniciando com o processo de *upload* de um arquivo contendo a música desejada para obter a cifra. Posteriormente, foram conduzidos testes no fluxo alternativo de inserção de uma URL do YouTube, também visando a obtenção da cifra correspondente. Por fim, foi realizado o teste de download da cifra gerada, completando assim uma avaliação abrangente das principais funcionalidades da plataforma em diferentes cenários de uso. Todos os testes foram elaborados pelo autor desse trabalho e as ferramentas empregadas incluíram um computador pessoal com o sistema operacional *Windows* 10, além de um dispositivo móvel equipado com o sistema operacional *Android*, ambos acessando a plataforma por meio do navegador *Google Chrome*.

4.3 Casos de teste

Nesta seção são apresentados os casos de teste de maior relevância para o produto. Cada caso de teste é detalhado com informações específicas para uma avaliação precisa da funcionalidade em questão.

- **Descrição:** Esta linha explica o caso de teste e seu objetivo.

- **Entrada:** Apresenta os requisitos necessários para a realização do teste.
- **Resultado Esperado:** Define as expectativas de resultado para o teste em questão.
- **Resultado Obtido:** Esta linha oferece variações, de acordo com o desempenho do teste. Os resultados podem ser classificados como:
  - **Bem-sucedido:** Quando a funcionalidade testada apresenta 100% do resultado esperado.
  - **Parcialmente Bem-sucedido:** Se a funcionalidade testada apresentar apenas 50% do resultado esperado.
  - **Falha:** Caso a funcionalidade apresente menos de 50% do resultado esperado.
  - **Análise:** Descreve o resultado do teste, indicando se a funcionalidade funcionou corretamente, ou se houve falha.
  - **Observações:** Esta linha oferece observações adicionais sobre cada caso de teste, proporcionando *insights* adicionais sobre o desempenho e comportamento da funcionalidade em avaliação.

As tabelas (7, 8, 9 e 10) exibem os casos de teste que foram identificados como as funcionalidades mais cruciais da aplicação. A seguir, serão detalhadas essas funcionalidades por meio de descrições.

Na tabela 7, será conduzido o teste de *Upload de arquivo*, que tem como propósito avaliar o procedimento de envio de um arquivo para a API. Nesse teste, o usuário submete um arquivo de áudio e espera-se que o mesmo seja enviado para a API.

Na tabela 8, será conduzido o teste da URL do Youtube, que tem como propósito avaliar o processo de envio de uma URL contendo uma música para o *backend*. Neste teste, o usuário submete uma URL do Youtube para o *backend* e espera-se que o mesmo realize o *download* do vídeo no formato ".mp3" ou ".webm" e após isso envie para API da Music.AI para serem efetuadas as devidas manipulações.

Na tabela 9, será conduzido o teste de Renderização das cifras, com o objetivo de avaliar se a cifra gerada será corretamente exibida na tela do usuário após o término da tela de carregamento. Durante este teste, espera-se que, após a conclusão da tela de carregamento, a cifra da música e sua tonalidade sejam apresentadas conforme o esperado.

Por fim, na tabela 10, será realizado o teste de *Download* da cifra, visando avaliar se a cifra gerada poderá ser baixada com sucesso. Durante esse procedimento, espera-se que, após a geração da cifra, o usuário consiga efetuar o *download* do arquivo sem qualquer dificuldade.

Caso de Teste 01	Upload de arquivo
Descrição	Esse caso de teste tem como objetivo avaliar o processo de *upload* de um arquivo.
Entradas	O usuário envia um arquivo de áudio com extensão ".mp3" ou ".webm".
Resultado esperado	Após o *upload* o arquivo deve ser enviado para o *backend* para que o mesmo envie para API da Music.AI para serem efetuadas as devidas manipulações.
Resultado obtido	Bem-sucedido
Análise do resultado	Funcionou corretamente
Observações	Nenhuma

Tabela 7: Caso de teste 01

Caso de Teste 02	URL do Youtube
Descrição	Esse caso de teste tem como objetivo avaliar o processo de envio de uma URL do Youtube contendo uma música para o *backend*.
Entradas	O usuário envia uma URL do Youtube contendo o vídeo música que deseja conhecer a cifra.
Resultado esperado	Após o envio da URL o *backend* deve realizar o *download* do vídeo no formato ".mp3" ou ".webm" e enviar para API da Music.AI para serem efetuadas as devidas manipulações.
Resultado obtido	Funcionou corretamente
Análise do resultado	Bem-sucedido
Observações	Nenhuma

Tabela 8: Caso de teste 02

Caso de Teste 03	Apresentação das cifras na tela
Descrição	Esse caso de teste tem como objetivo avaliar se as cifras foram apresentadas corretamente na tela após a tela de *loading*.
Entradas	Arquivo de áudio ou URL do Youtube
Resultado esperado	Após a tela de *loading* deve ser apresentada a cifra da música e sua tonalidade.
Resultado obtido	Funcionou corretamente
Análise do resultado	Bem-sucedido
Observações	Nenhuma

Tabela 9: Caso de teste 03

Caso de Teste 04	Download da cifra
Descrição	Esse caso de teste tem como objetivo avaliar o processo de *download* da cifra após o usuário clicar no botão "Baixar cifra".
Entradas	Arquivo de áudio ou URL do Youtube
Resultado esperado	Após o usuário clicar no botão "Baixar cifra" é esperado que o mesmo receba um arquivo com extensão ".txt" contendo a cifra da música.
Resultado obtido	Funcionou corretamente
Análise do resultado	Bem-sucedido
Observações	Nenhuma

Tabela 10: Caso de teste 04

Após a realização dos testes nas principais funcionalidades do *Cifre.ME*, não foi identificado nenhum problema significativo. Todos os testes foram bem-sucedidos, demonstrando que o produto está pronto para ser utilizado pelos usuários.

4.4 Comparando com plataformas similares

Existem diversas plataformas que disponibilizam cifras musicais e podem servir como pontos de referência para avaliar o *Cifre.ME*. Uma dessas plataformas é o Cifra Club, especializado no ensino musical e além disso oferece uma ampla variedade de cifras. No Cifra Club, as cifras são elaboradas de forma manual, ou seja, são submetidas pelos próprios usuários. Por outro lado, no caso do *Cifre.ME*, as cifras serão geradas automaticamente.

Para realizar comparações com outras cifras, primeiramente optou-se por selecionar a cifra da música "Velha Infância - Tribalistas". Esta escolha foi feita devido ao seu gênero musical, MPB, que é bastante presente na música nacional. Nas Figuras 11 e 12, apresenta-se um trecho da cifra desta música a fim de realizar uma comparação entre as plataformas mencionadas.

A análise comparativa das figuras revela que tanto o *Cifre.ME*, quanto o Cifra Club [5], apresentam a mesma tonalidade, sendo ela o **F#m** (Fá sustenido menor), e os mesmos acordes, uma vez que ambos incluem o **F#m** (Fá sustenido menor), **Bm** (Si menor) e **E** (Mi maior). No entanto, o *Cifre.ME* exibe algumas discrepâncias na posição dos acordes e na letra da música, incluindo palavras errôneas quando comparadas à composição original.

[5]https://www.cifraclub.com.br/

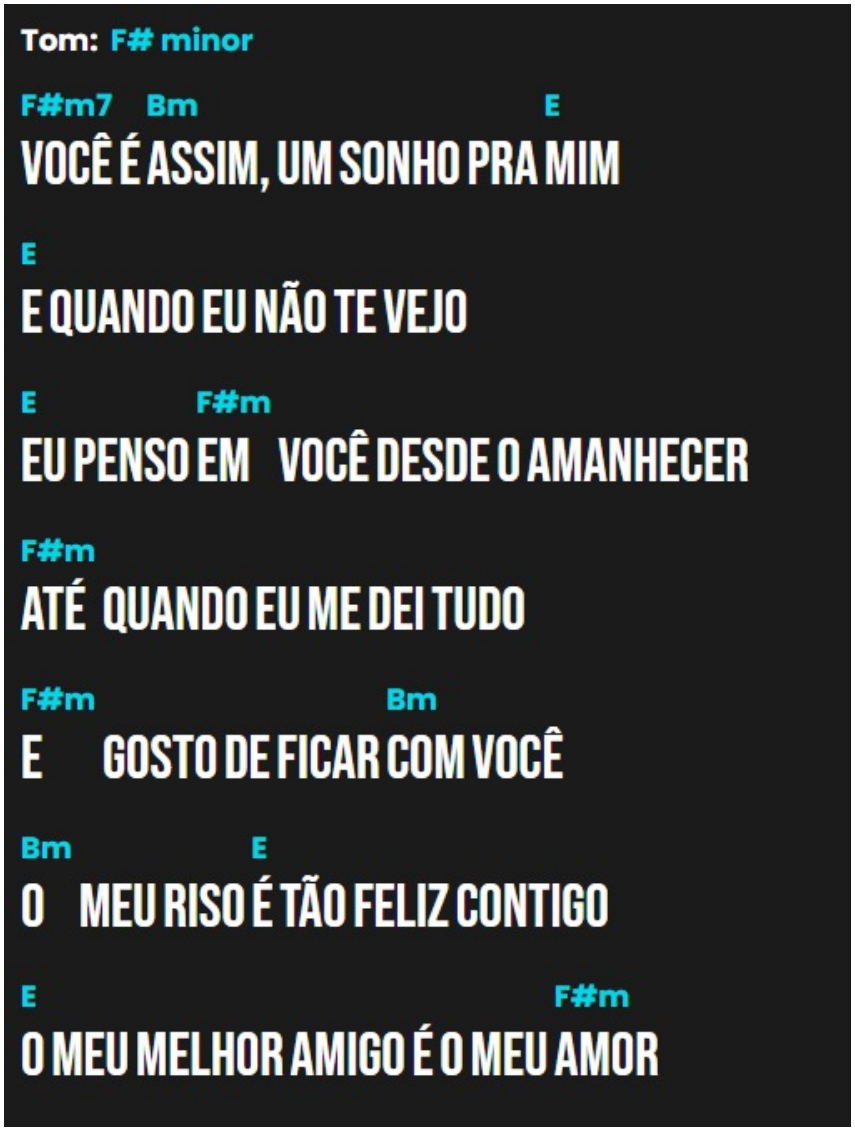

Figura 11: Cifra da Música Velha infância gerada pelo Cifre.ME
Fonte: Autoria própria

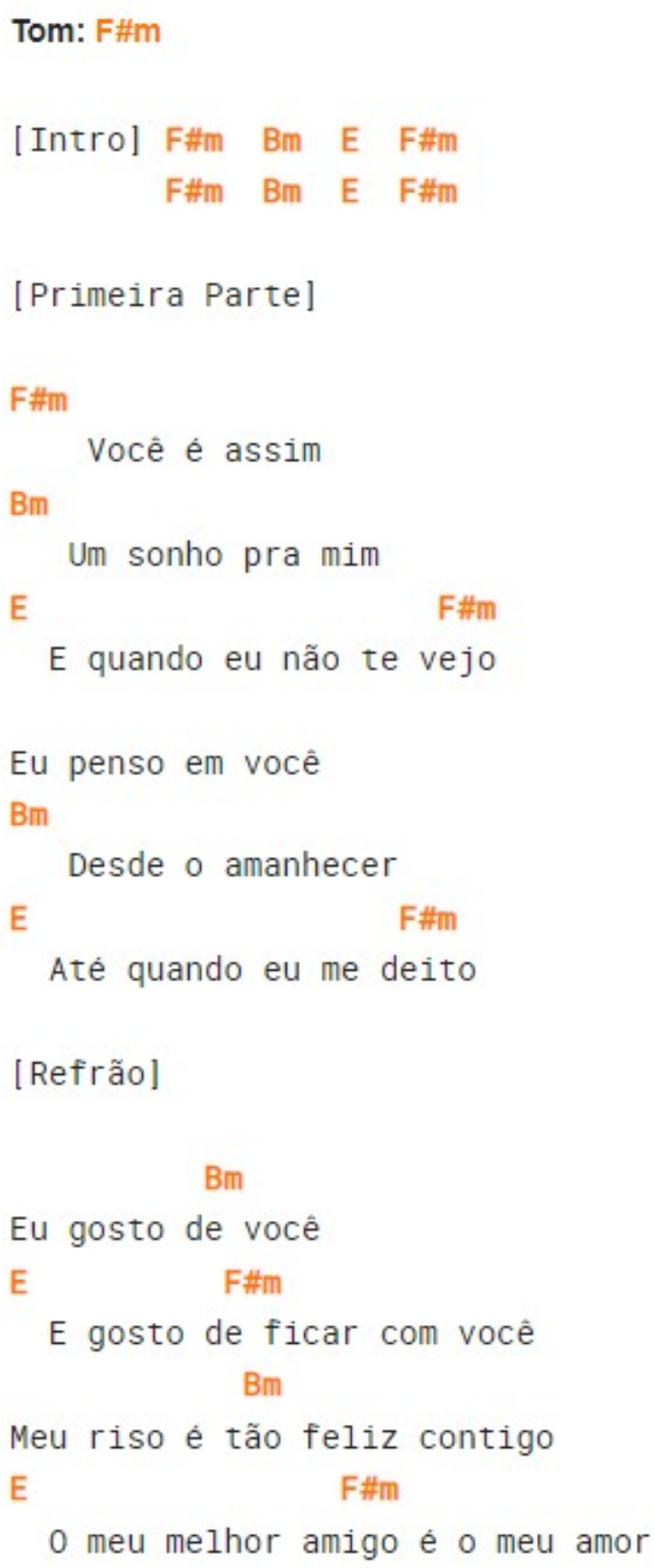

Figura 12: Cifra da Música Velha infância gerada pelo Cifra club
Fonte: Cifra club

Com o propósito de realizar mais comparações para obter conclusões mais exatas, desta vez foi escolhida a música **Hotel California - Eagles** escolhida por apresentar um gênero musical bastante ouvido pelos amantes da música sendo ele o rock internacional. Nas Figuras 13 e 14, é apresentado um trecho da cifra dessa música para análise comparativa.

Ao compararmos as Figuras (13 e 14), é evidente a semelhança entre os acordes apresentados em ambas as cifras. Tanto o **Bm** (Si menor), **F#7** (Fá sustenido com Sétima Maior), **A** (Lá maior), **Em** (Mi menor), **G** (Sol maior) e **D** (Ré maior) são comuns em ambas as representações. No entanto, há uma discrepância no acorde **E** (Mi maior), que no *Cifre.ME* aparece de forma simplificada porque o *Cifre.ME* utiliza os acordes na sua forma simples, enquanto na cifra do Cifra Club é representado como **E7** (Mi com Sétima maior), em sua forma mais complexa.

Apesar da questão de posicionamento dos acordes ao longo da cifra, é notável que ambas as plataformas conseguem identificar corretamente os acordes e a tonalidade da música em questão. Assim, podemos concluir que o *Cifre.ME* está desempenhando sua função de forma eficaz ao reconhecer os acordes e a tonalidade da música analisada.

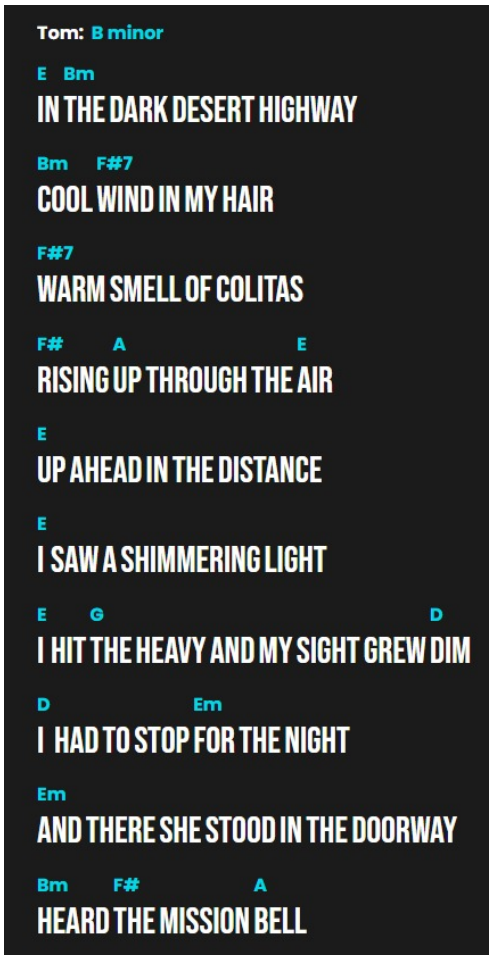

Figura 13: Cifra da Música Hotel Califórnia gerada pelo Cifre.ME
Fonte: Autoria própria

Figura 14: Cifra da Música Hotel Califórnia gerada pelo Cifra club
Fonte: Cifra club

Agora, será abordada outra música com um gênero musical distinto das anteriores: a música **Malandragem - Cássia Eller**. Esta canção foi escolhida por ser um clássico do Pop Rock nacional na qual muitas pessoas ainda escutam esta composição apesar de ser bem antiga. Nas figuras (15 e 16), é apresentada um trecho da cifra dessa música para possibilitar análises comparativas.

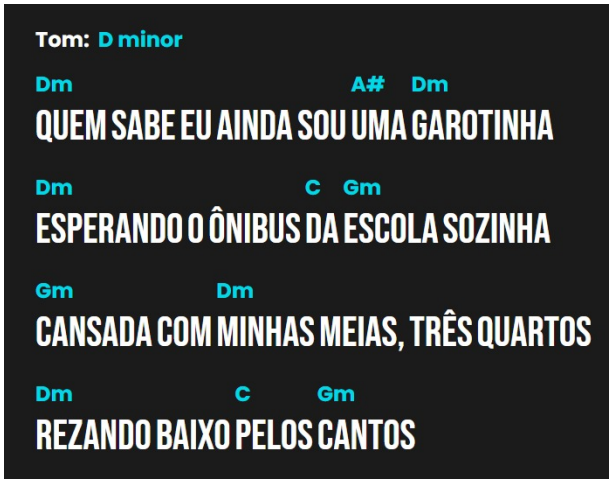

Figura 15: Cifra da Música Malandragem gerada pelo Cifre.ME
Fonte: Autoria própria

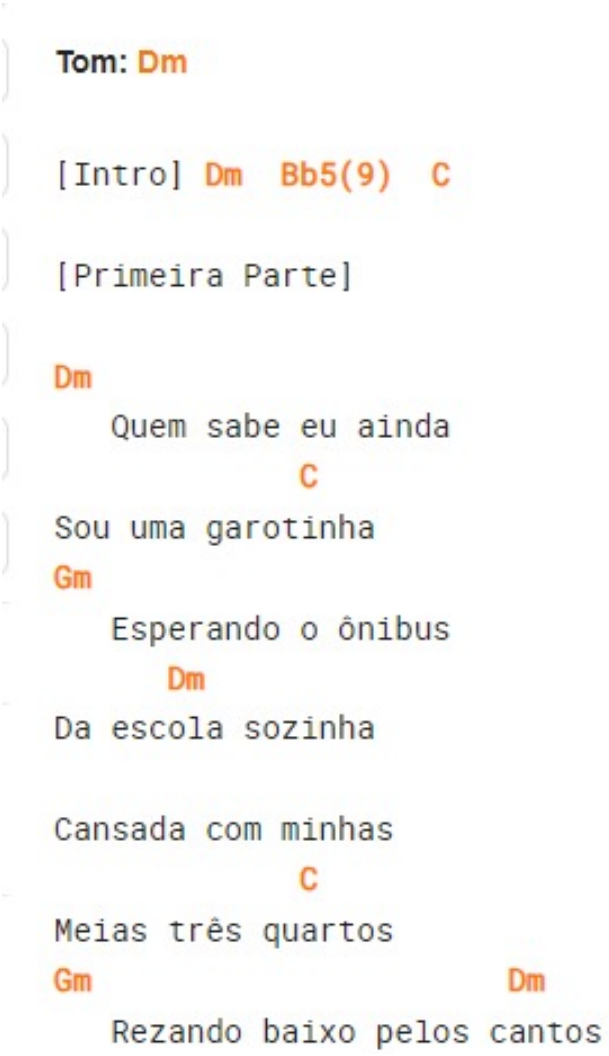

Figura 16: Cifra da Música Malandragem gerada pelo Cifra club
Fonte: Cifra club

Observamos que a tonalidade da música é consistente em ambas as cifras, sendo ela o **Dm** (Ré menor). Além disso, os acordes **Dm** (Ré menor), **C** (Dó maior) e **Gm** (Sol menor) estão presentes em ambas as representações. No entanto, no *Cifre.ME*, é apresentado o acorde **A#** (Lá sustenido ou Si Bemol), enquanto na cifra do Cifra Club é indicado o acorde **C**. Apesar da diferença, é importante ressaltar que o acorde **A#** também faz parte da tonalidade de **Dm**, portanto, há a possibilidade de não estar incorreto, uma vez que pertence à tonalidade da música.

É relevante destacar que a escala musical apresentada na Tabela 2 corrobora com a presença do acorde **A#** na tonalidade de **Dm**, reforçando a coerência da representação utilizada pelo *Cifre.ME* neste contexto específico.

Com o objetivo de confirmar a eficiência do *Cifre.ME* em relação a outras plataformas similares foi realizado mais um teste para comprovar se de fato pelo menos a tonalidade e os acordes eram similares em ambas as cifras. Dessa vez, a música escolhida foi **Skank - Acima do Sol** escolhida por ser uma música bem peculiar excelente para músicos iniciantes aprenderem. Nas figuras (18 e 17) é apresentada um trecho da música que serve de exemplo para serem feitas as devidas comparações.

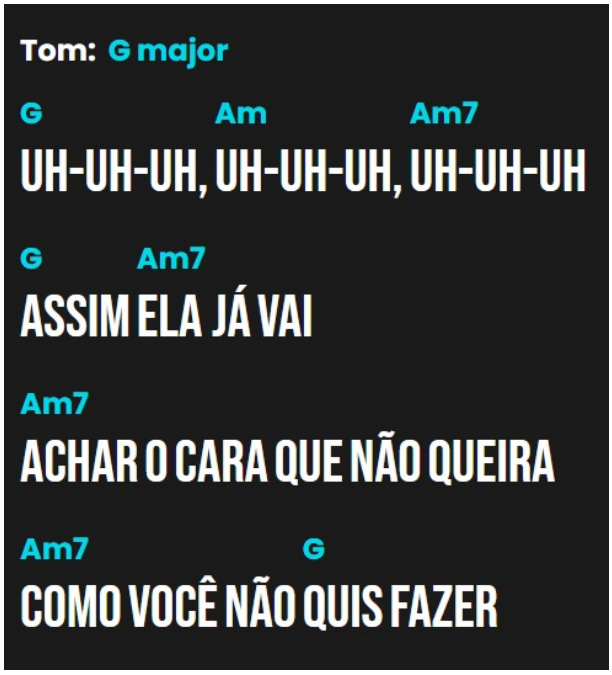

Figura 17: Cifra da Música Acima do Sol gerada pelo Cifre.ME
Fonte: Autoria própria

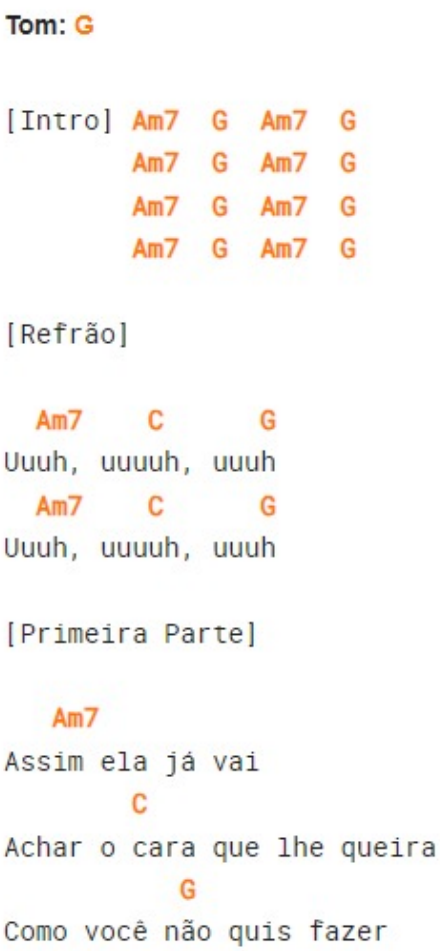

Figura 18: Cifra da Música Acima do sol gerada pelo Cifra club
Fonte: Cifra club

Ao observar as figuras acima, é evidente que, mais uma vez, a tonalidade da música é idêntica em ambas as cifras, sendo ela o **G** (Sol Maior). No entanto, há diferenças nos acordes e na letra da música. O *Cifre.ME* identificou três acordes na música: **G** (Sol maior), **Am** (Lá menor) e **Am7** (Lá menor com Sétima menor). Por outro lado, o Cifra Club também apresentou três acordes, mas o **Am** não está presente, sendo substituído por um **C** (Dó maior). Neste caso, é possível que tenha ocorrido um equívoco por parte do *Cifre.ME*.

É relevante notar que tais divergências podem ocorrer devido a diferentes interpretações ou versões da música, bem como variações na transcrição das cifras. Assim, é importante considerar essas nuances ao comparar cifras de diferentes fontes.

4.5 Discussão dos resultados

Os resultados obtidos sugerem que o *Cifre.ME* demonstrou eficácia em grande parte de suas funcionalidades. Sua capacidade de resposta em dispositivos móveis foi

considerada satisfatória, e todos os testes conduzidos alcançaram êxito. No contexto da análise das tonalidades e acordes das músicas, o *Cifre.ME* obteve sucesso ao atingir a meta de reconhecimento, tanto do tom, quanto dos acordes das composições. No entanto, em relação ao posicionamento dos acordes sobre as palavras correspondentes, foi conduzido um estudo sobre os dados retornados pela API da Music.AI a fim de investigar por que o posicionamento dos acordes difere da cifra do Cifra Club. Após essa análise, concluiu-se que o *Cifre.ME* está adicionando os acordes nos *timestamps* corretos, de acordo com o retorno da API. No entanto, ao comparar as cifras do *Cifre.ME* com as do Cifra Club, plataforma similar, observou-se que os acordes e as tonalidades são iguais. No entanto, é importante notar que o posicionamento dos acordes apresenta algumas diferenças devido aos *timestamps* retornados pela API. Essa disparidade pode gerar dificuldades para o usuário, especialmente quando ele segue a cifra da música à risca. Embora os acordes e as tonalidades estejam corretos, pode ocorrer que a mudança de acorde esteja sincronizada incorretamente com o tempo da música.

5 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho delineou uma proposta concreta: desenvolver uma aplicação web responsiva, denominada *Cifre.ME*, com o objetivo específico de automatizar o processo de geração de cifras. Essa iniciativa visa proporcionar aos usuários uma forma simplificada e acessível de tocar músicas, utilizando cifras simplificadas, além de oferecer a comodidade da geração automática das cifras. Esse último aspecto é particularmente relevante para simplificar a vida daqueles que se dedicam à elaboração dessas cifras e desejam disponibilizá-las em plataformas online.

Para alcançar esse propósito, foi conduzida uma análise aprofundada nos módulos da Music.AI (Seção 2.8), os quais fornecem os dados necessários para o funcionamento do *Cifre.ME*. A integração desses módulos revelou-se crucial para alcançar o objetivo maior, que consistia em extrair informações essenciais das músicas, viabilizando, assim, a geração automática das cifras. Além disso, todas as fases de desenvolvimento da aplicação foram detalhadas bem como as ferramentas utilizadas para o seu desenvolvimento e até mesmo um planejamento de testes que por sua vez, foram executados internamente para validar as funcionalidades e os possíveis problemas além de comparações com plataformas similares.

No entanto, embora o *Cifre.ME* ainda apresente algumas falhas, é inegável que a ferramenta demonstrou, durante os testes realizados, um desempenho adequado em suas funcionalidades principais. Embora tenha sido identificado problemas no posicionamento dos acordes e na transcrição da música apresentando algumas palavras erradas (dificuldade apresentada pela API), especialmente em comparação com outras plataformas, o projeto tem como expectativa tornar-se uma ferramenta para músicos, permitindo a geração automática de cifras. Portanto, é interessante realizar correções, modificações e acréscimos de novos requisitos para atender cada vez melhor às necessidades dos usuários.

Com base nesse contexto, algumas melhorias e correções são sugeridas para trabalhos futuros:

- Aprimoramento no posicionamento dos acordes nas cifras, visando uma maior precisão e alinhamento.
- Implementação da funcionalidade para baixar as cifras em formato PDF.
- Criação de listas personalizadas com cifras favoritas, facilitando o acesso e organização das músicas preferidas pelos usuários.
- Divisão das cifras em seções distintas, como Introdução, Refrão e Final, para uma melhor compreensão e interpretação da música.

- Adição do nome da música e do compositor, proporcionando informações adicionais e enriquecendo a experiência do usuário.
- Inclusão da opção de alterar a tonalidade das cifras, permitindo que os usuários adaptem as músicas de acordo com suas preferências e habilidades musicais.

Essas sugestões visam aprimorar a usabilidade e a funcionalidade do *Cifre.ME*, garantindo que se torne uma ferramenta ainda mais completa e satisfatória para os músicos em seu uso cotidiano.

REFERÊNCIAS

CONCEITO.DE. **Oitava**. 2024. Disponível em: <https://conceito.de/oitava>.

DANTAS, J. D.; CRUZ, S. d. S. Um olhar físico sobre a teoria musical. **Revista Brasileira de Ensino de Física**, Sociedade Brasileira de Física, v. 41, n. 1, 2019. ISSN 1806-1117. Disponível em: <https://doi.org/10.1590/1806-9126-RBEF-2018-0099>.

FLANAGAN, D. **JavaScript: o guia definitivo**. [S.l.]: Bookman Editora, 2012.

FROZZA, A. C. DetecÇÃo de acordes musicais por meio de informaÇÕes espectrais. **Universidade Tecnológica Federal do Paraná**, 2019. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/6019/1/acordesmusicaisinformacoesespectrais.pdf>.

JESUS, T. S. d. Chordix: um aplicativo para auto-acompanhamento musical através de cifras. Departamento de Ciência da Computação, 2017.

MOURA, B.; HENRIQUE, M. O que é html: suas tags - parte 1 - estrutura básica. **Alura**, 2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-html-suas-tags-parte-1-estrutura-basica>.

NETO, M. Q. N. Estudo sobre extração de características e reconhecimento de padrões aplicados em cifragem de músicas. **INSTITUTO LATINO AMERICANO DE CIÊNCIAS DA VIDA E NATUREZA (ILACVN)**, 2021. Disponível em: <https://dspace.unila.edu.br/bitstream/handle/123456789/6309/Tcc_Mauro_Nooblath_2021.pdf?sequence=1&isAllowed=y>.

NEVES, V. Css: o que é, como usar no html e um guia para iniciar. **Alura**, 2023. Disponível em: <https://www.alura.com.br/artigos/css>.

_____. React: o que é, como funciona e um guia dessa popular ferramenta js. **Alura**, 2023. Disponível em: <https://www.alura.com.br/artigos/react-js>.

OMAUROSOUTO. O que é front-end back-end e full stack - aprenda as diferenças entre essas áreas. **Alura**, 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>.

PIMENTEL, E. O que é o dom. **Alura**, 2022. Acesso em: 22.3.2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-o-dom?utm_term=&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=20987928442&hsa_grp=157916200306&hsa_ad=689395782879&hsa_src=g&hsa_tgt=dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQjw2PSvBhDjARIsAKc2cgM1iKY5lId2lpP95SqpWCMLc4BjMhMROzVE1_tLVnzX_irWptPP6lcaAoBcEALw_wcB>.

SILVA, M. S. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. [S.l.]: Novatec Editora, 2010.

VILLAIN, M.; SILVEIRA, I. Figmadesign: A ferramenta de design definitiva. **Alura**, 2023. Acesso em: 19.3.2024. Disponível em: <https://www.alura.com.br/artigos/figma>.

ZALANDAUSKAS, B. I. Desvendando as escalas musicais: um jeito simples de aprender! **JAM Music**, 2023. Disponível em: <https://www.jam.mus.br/desvendando-as-escalas-musicais/>.