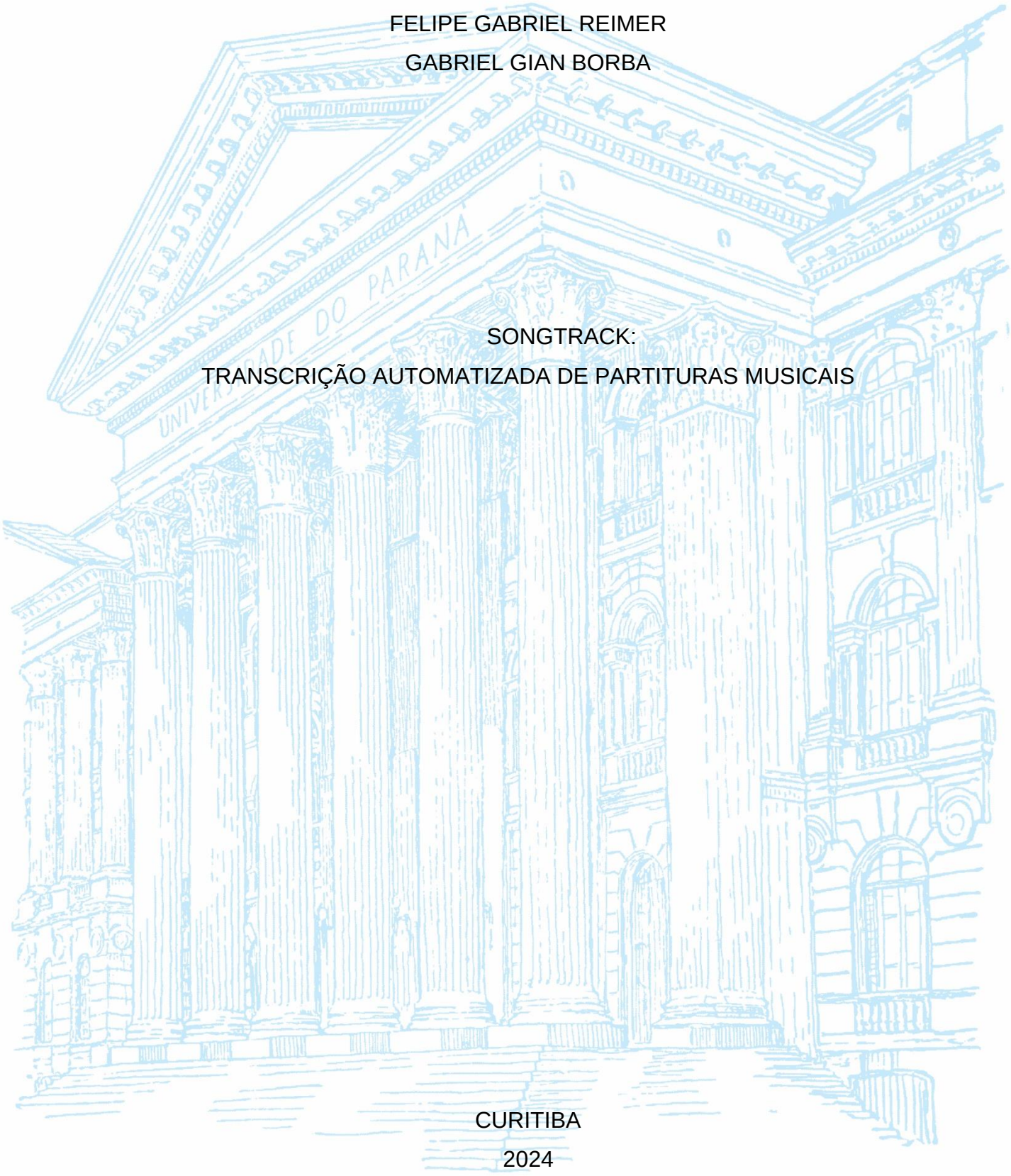
UNIVERSIDADE FEDERAL DO PARANÁ

FELIPE GABRIEL REIMER

GABRIEL GIAN BORBA



SONGTRACK:
TRANSCRIÇÃO AUTOMATIZADA DE PARTITURAS MUSICAIS

CURITIBA

2024

FELIPE GABRIEL REIMER
GABRIEL GIAN BORBA

SONGTRACK:
TRANSCRIÇÃO AUTOMATIZADA DE PARTITURAS MUSICAIS

Trabalho de conclusão de curso apresentado ao curso de Graduação de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador(a): Prof(a). Dr(a). João Eugenio Marynowski

CURITIBA
2024

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR, CEP 81520-260
Telefone: 3360-5000 - http://www.ufpr.br/

Ata de Reunião

TERMO DE APROVAÇÃO

FELIPE GABRIEL REIMER
GABRIEL GIAN BORBA

SONGTRACK: TRANSCRIÇÃO AUTOMATIZADA DE PARTITURAS MUSICAIS

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Dr. João Eugenio Marynowski
Orientador – SEPT/UFPR

Prof. Dr. Alexander Robert Kutzke
SEPT/UFPR

Prof. Dr. Razer Anthom Nizer Rojas Montano
SEPT/UFPR

Curitiba, 15 de agosto de 2024.

Documento assinado eletronicamente por **JOAO EUGENIO MARYNOWSKI**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 15/08/2024, às 11:25, conforme art. 1º, III, "b", da Lei 11.419/2006.

Documento assinado eletronicamente por **RAZER ANTHOM NIZER ROJAS MONTANO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 15/08/2024, às 12:40, conforme art. 1º, III, "b", da Lei 11.419/2006.

Documento assinado eletronicamente por **ALEXANDER ROBERT KUTZKE**, **Usuário Externo**, em 01/10/2024, às 19:18, conforme art. 1º, III, "b", da Lei 11.419/2006.

A autenticidade do documento pode ser conferida aqui informando o código verificador **6929997** e o código CRC **71A56643**.

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação baseada em Python para a transcrição de partituras musicais a partir de arquivos de música. O sistema utiliza bibliotecas como Demucs e Librosa com a finalidade de separar fontes de áudio de uma música, criar partituras musicais, além de processar arquivos MIDI. O sistema também utiliza o Flet como ferramenta para o desenvolvimento de interfaces do usuário. Procuramos ajudar músicos, produtores e pesquisadores a interagir com a música de forma mais eficiente. Para isso, exploramos os fundamentos teóricos e as aplicações práticas do tema, abordando a separação de trilhas de áudio e o processamento de diferentes sinais. A conclusão deste trabalho destaca que a aplicação desenvolvida conseguiu atingir os objetivos propostos, proporcionando uma ferramenta útil e intuitiva para a transcrição de partituras e manipulação de sinais musicais, embora ainda existam desafios e oportunidades para aprimoramento, especialmente no que tange à precisão da classificação das notas musicais e à eficiência no processamento dos sinais de áudio.

Palavras-chave: Análise musical; processamento de áudio; processamento de sinais; Demucs; Librosa; Flet; transcrições de partituras musicais; separação de trilhas de áudio;

ABSTRACT

This work presents the development of a Python-based application for transcribing musical scores from music files. The system uses libraries such as Demucs and Librosa to separate audio sources from a song, create musical scores, and process MIDI files. The system also utilizes Flet as a tool for developing user interfaces. We seek to help musicians, producers, and researchers interact with music more efficiently. To achieve this, we explore the theoretical foundations and practical applications of the subject, addressing the separation of audio tracks and the processing of different signals. The conclusion of this work highlights that the developed application successfully met the proposed objectives, providing a useful and intuitive tool for score transcription and signal manipulation, although there are still challenges and opportunities for improvement, particularly in the accuracy of note classification and the efficiency of signal processing.

Keywords: Musical analysis; audio processing; signal processing; Demucs; Librosa; Flet; musical scores transcriptions; audio track separation;

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS OU SIGLAS

IA	- Inteligência Artificial
GUI	- Interface Gráfica do Usuário
RNR	- Redes Neurais Recorrentes
RNC	- Redes neurais convulacionais
DDD	- Domain Driven Design
MIDI	- Musical Instrument Digital Interface
NMF	- Fatoração de Matrizes Não-Negativas
GRU	- Gated Recurrent Units
LSTM	- Long Short-Term Memory
GB	- Gigabytes
CQT	- Transformada Constante-Q

SUMÁRIO

1 INTRODUÇÃO

A transcrição de partituras musicais é uma tarefa que, quando realizada manualmente, exige não só tempo, mas também um elevado nível de conhecimento musical. Além disso, a manipulação de faixas musicais envolve desafios complexos que requerem técnicas avançadas de processamento de sinais e inteligência artificial. A separação de fontes musicais, por exemplo, é fundamental para diversas aplicações, incluindo softwares de remixagem de músicas e a análise minuciosa de composições musicais (LI et al., 2021).

Neste contexto, o Demucs, que emprega redes neurais para a separação de componentes musicais, tem se mostrado uma ferramenta eficaz, isolando diferentes componentes de uma faixa musical com o uso de arquiteturas avançadas (DÉFOSSEZ et al., 2019). A biblioteca é utilizada primordialmente para a separação de fontes de áudio, facilitando a extração de vocais, bateria, baixo e outros instrumentos de uma música.

O presente projeto visa combinar o Demucs com outras ferramentas de manipulação de sinais de áudio, como o PyDub, para criar uma solução integrada e robusta que atenda a diversas necessidades nos campos de musicologia, teoria musical e desenvolvimento de software musical. Essa abordagem proporciona uma plataforma flexível para experimentação e pesquisa, permitindo análises detalhadas de estruturas musicais e o processamento de arquivos de áudio digital e dados MIDI.

O objetivo central deste trabalho é desenvolver um software eficiente que realize a separação das fontes de um arquivo de áudio, processe arquivos no formato MIDI, e supere as dificuldades técnicas associadas ao uso do Demucs e outras ferramentas. Além disso, o software visa garantir uma excelente experiência ao usuário, oferecendo uma interface acessível e funcional, facilitando a manipulação e análise dos arquivos de áudio. Dessa forma, o projeto pretende fornecer uma ferramenta que auxilie músicos, produtores e pesquisadores a interagir com a música de maneira mais eficiente e produtiva.

1.1 PROBLEMA

A análise de sinais de áudio no contexto musical é uma tarefa complexa, devido à variedade de elementos como ritmo, tom e timbre, que interagem de maneiras intrincadas (CASEY et al., 2008). Desenvolver um sistema capaz de interpretar esses sinais e extrair informações significativas de cada componente musical é um grande desafio. A separação de fontes musicais, como vocais, instrumentos de percussão e harmonia, requer técnicas avançadas de processamento de sinais e inteligência artificial (RISOUD et al., 2018).

A utilização do Demucs apresenta diversos desafios adicionais, embora seja poderoso, sua implementação e uso eficaz requerem um entendimento de modelos de aprendizado profundo e processamento de áudio (DÉFOSSEZ et al., 2019). Configurar o ambiente adequado, ajustar os hiperparâmetros da rede neural convolucional utilizada pelo Demucs, e lidar com grandes volumes de dados de treinamento são tarefas que demandam conhecimento especializado e recursos computacionais significativos (GOODFELLOW; BENGIO; COURVILLE, 2016).

1.2 OBJETIVOS

A seguir, são apresentados os objetivos que direcionam o desenvolvimento do projeto, divididos em objetivo geral e objetivos específicos. Esses objetivos estabelecem a base para as atividades e resultados esperados ao longo do desenvolvimento do sistema.

O objetivo geral consiste em produzir um aplicativo de interface intuitiva e direta, que permite ao usuário inexperiente processar e extrair informações úteis dos sinais de áudio, utilizando um modelo de classificação de notas musicais a partir do reconhecimento de imagens.

Para atingir esse objetivo amplo, foram estabelecidos objetivos específicos, que incluem:

- Desenvolver um modelo de reconhecimento de notas musicais a partir do processamento de imagens utilizando redes neurais convulacionais e integrá-lo no sistema de transcrição automática de músicas.

- Desenvolver um sistema de análise de áudios musicais, integrando diferentes bibliotecas de software.
- Criar uma interface gráfica de usuário (GUI) flexível e intuitiva, utilizando o *framework* Flet.
- Implementar uma separação de fontes de áudio.
- Lidar com a criação e análise de partituras musicais.
- Gerar partituras notadas no formato PDF.

1.3 JUSTIFICATIVA

Este software busca proporcionar uma ferramenta acessível e eficiente para músicos ou profissionais da área, que permite análise e interpretação de músicas através do uso do aplicativo.

Além disso, no ramo da produção musical, este sistema pode transformar gravações em partituras analisáveis, oferecendo uma maneira simples de explorar novas ideias e composições.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está organizado em cinco capítulos principais, além das Referências e Apêndices.

No Capítulo 1 - Introdução, são apresentados o problema de pesquisa, o objetivo geral e os objetivos específicos, bem como a justificativa para o desenvolvimento do estudo.

O Capítulo 2 - Fundamentação Teórica aborda os conceitos fundamentais relacionados à teoria musical, notação musical, transcrição musical, MIDI, redes neurais convolucionais e separação de faixas musicais, divididos em seções que detalham cada um desses tópicos.

O Capítulo 3 - Materiais e Métodos descreve a metodologia adotada para o desenvolvimento do software, incluindo as técnicas de gestão de projetos, como Kanban, a divisão de tarefas, o cronograma, os requisitos funcionais e não funcionais, e os diagramas de especificação.

No Capítulo 4 - Apresentação do Sistema, é apresentada a arquitetura do sistema desenvolvido, incluindo a estrutura, navegação, interação e as integrações realizadas. Além disso, são descritas as telas principais da aplicação, como a tela de geração de partitura e gestão de áudios.

O Capítulo 5 - Considerações Finais oferece um resumo das principais contribuições do trabalho, discute as limitações encontradas e sugere trabalhos futuros que podem ser desenvolvidos a partir deste estudo.

Por fim, são apresentadas as Referências utilizadas ao longo do trabalho, seguidas pelos Apêndices, que contêm diagramas, histórias de usuário e outros documentos auxiliares relevantes para a compreensão do desenvolvimento do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Base teórica que sustenta o desenvolvimento do projeto, abordando conceitos fundamentais de teoria musical e notação musical. A fundamentação teórica oferece o suporte necessário para a compreensão dos desafios técnicos e conceituais envolvidos no processo de transcrição e análise musical.

2.1 TEORIA MUSICAL

Segundo Cook (1998), música é uma forma de arte complexa e diversa, que combina padrões de maneiras criativas e expressivas, possuindo variações em termos de ritmo, melodia, harmonia, timbre, textura e forma. Compreender e interpretar essa complexidade é um dos elementos enfrentados por músicos e profissionais da área.

Teoria musical é o estudo dos elementos e estruturas que compõem a música. Ela serve como base para a análise e a composição de músicas, e sua compreensão é essencial para a transcrição musical (COOK, 1998).

2.2 ELEMENTOS BÁSICOS DA MÚSICA

A música é composta por três elementos fundamentais: melodia, harmonia e ritmo. Esses elementos trabalham em conjunto para criar a estrutura e a expressão de uma peça musical.

A melodia é a sequência linear de notas musicais que são percebidas como uma única entidade. É o elemento da música que geralmente capta a atenção do ouvinte, sendo muitas vezes considerada a "voz" da composição. Segundo Piston (1978), a melodia é composta por uma série de sons que são tocados ou cantados em sucessão, formando frases musicais que possuem início, meio e fim.

A harmonia refere-se ao uso simultâneo de diferentes notas ou acordes para criar uma textura sonora rica e coesa. Ela é responsável por dar profundidade e contexto à melodia. De acordo com Benward e Saker (2009), a harmonia é o suporte que dá sentido à melodia, proporcionando contraste e complementação através de acordes e progressões harmônicas.

O ritmo é o elemento que organiza a música no tempo, determinando a duração e a acentuação das notas e dos silêncios. Conforme Rossing (2007), o ritmo é a base

que sustenta a estrutura temporal de uma peça musical, sendo responsável por criar padrões repetitivos que podem variar em complexidade e velocidade, dando à música seu caráter pulsante e dinâmico.

2.3 NOTAS MUSICAIS

De acordo com Benward e Saker (2009), as notas musicais são os blocos construtivos da música. Cada nota corresponde a uma altura específica, que é determinada por sua frequência sonora. No sistema de notação musical ocidental, as notas são designadas pelas letras A, B, C, D, E, F e G, representando as notas Lá, Si, Dó, Ré, Mi, Fá e Sol, respectivamente. Estas notas podem ser modificadas por sinais de sustenido (#) ou bemol (♭), que aumentam ou diminuem a altura da nota em um semitom, respectivamente.

Como observado por Piston (1978), as notas musicais são a base para a construção de escalas, melodias e harmonias, sendo representadas graficamente em uma pauta, que permite ao músico identificar tanto a altura quanto a duração de cada som. Além das notas naturais, há as notas alteradas, que desempenham um papel crucial na criação de diferentes tonalidades e modos musicais. A combinação e variação dessas notas formam a estrutura essencial para a composição musical.

Na teoria musical o Tom, refere-se à distância entre duas notas em uma escala. Um tom completo corresponde ao intervalo de dois semitons, sendo o semitom a menor distância entre duas notas na música ocidental. Por exemplo, na escala de Dó maior, a distância entre as notas Dó e Ré é de um tom. A compreensão dos intervalos de tom é essencial para a construção de escalas e acordes, influenciando diretamente a harmonia e a melodia de uma composição musical (BENWARD; SAKER, 2009).

Segundo Rossing (2007), a oitava é um dos intervalos mais importantes na teoria musical, definida como a relação entre duas notas em que a frequência da nota superior é o dobro da frequência da nota inferior. Esse intervalo é percebido como o mesmo tom, embora em registros diferentes, fazendo com que uma nota "Dó" em uma oitava soe como um "Dó" em outra oitava, mas com uma diferença de altura.

Conforme explica Kostka e Payne (2009), dentro de uma escala diatônica, uma oitava abrange oito notas, começando e terminando na mesma nota, mas em alturas distintas. Na escala de Dó maior, por exemplo, as notas Dó, Ré, Mi, Fá, Sol, Lá, Si e

Dó compõem uma oitava. A compreensão e organização das notas em oitavas são fundamentais para a construção de acordes, melodia e harmonia na música.

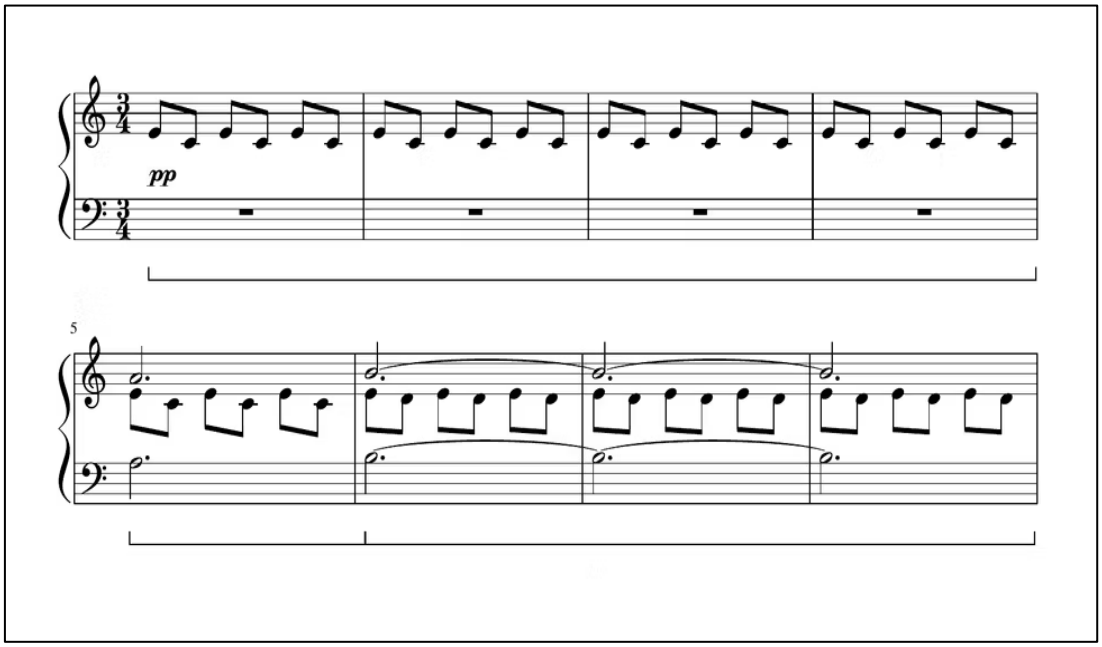
2.4 NOTAÇÃO MUSICAL

Conforme destacado por Mello (2015) a notação musical refere-se a qualquer método de escrita usado para representar visualmente uma composição musical, desde que permita o intérprete a executar conforme as intenções do compositor. Os diferentes sistemas de notação podem ser categorizados em dois tipos principais: símbolos fonéticos, que utilizam principalmente palavras, sílabas, abreviaturas, letras e números, e símbolos gráficos, que utilizam curvas, linhas, pontos e outras figuras abstratas.

O sistema de notação musical mais amplamente utilizado hoje é o sistema gráfico, que emprega símbolos escritos sobre uma pauta de cinco linhas, conhecida como pentagrama ou pauta. A combinação da pauta e outros símbolos musicais para representar uma peça musical é chamada de partitura (Figura 1). Existem também outros sistemas de notação utilizados na música moderna:

- Tablatura: Esta notação mostra como posicionar os dedos em um instrumento em vez de indicar notas musicais. Isso permite que músicos toquem o instrumento sem necessidade de formação especializada. A tablatura tornou-se especialmente popular para compartilhar músicas na Internet, pois pode ser facilmente escrita em formato de texto. Exemplo de tablatura pode ser visto na Figura 2.
- Cifra: Este sistema de notação musical usa símbolos gráficos ou letras para indicar os acordes a serem tocados por um instrumento musical. As cifras são amplamente utilizadas na música popular moderna, geralmente posicionadas acima das letras ou partituras de uma composição, mostrando qual acorde deve ser tocado junto com a melodia principal ou para acompanhar o canto. Exemplo de tablatura pode ser visto na Figura 3.

FIGURA 1 - EXEMPLO PARTITURA



FONTE: Autores (2024)

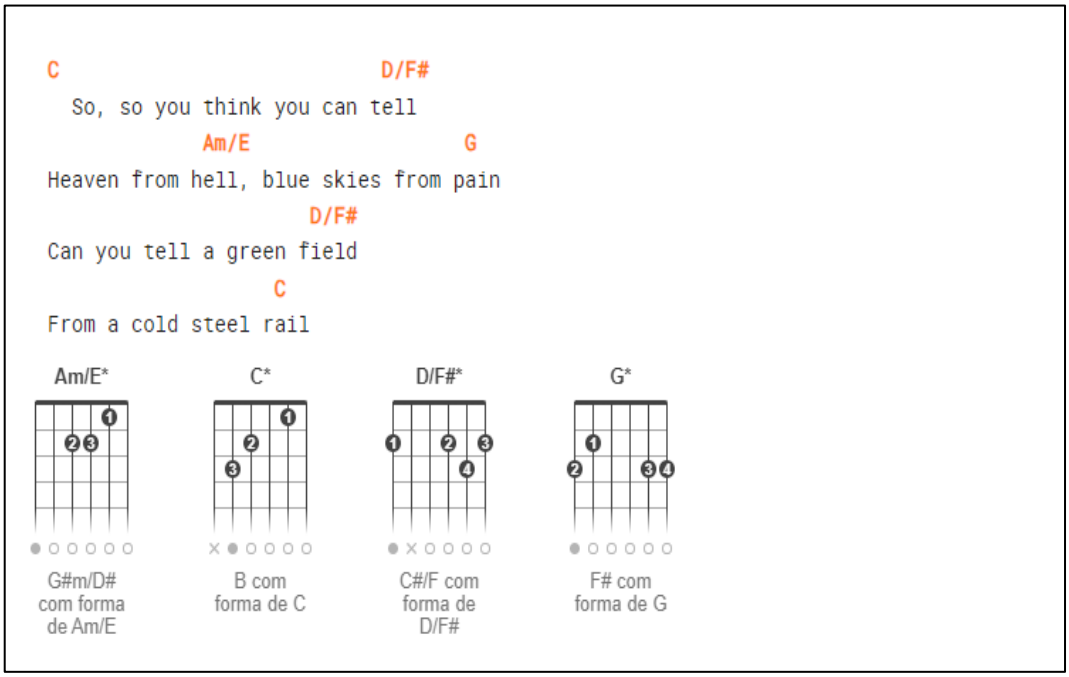
FIGURA 2 - EXEMPLO TABLATURA

```
E |3-1-0---0-1-3-3-3--5-3-1-1-1--3-1-0-0-0----------]
B |------3-------------------------------------------]
G |-------------------------------------------------]
D |-------------------------------------------------]
A |-------------------------------------------------]
E |-------------------------------------------------]

E |--0-5-5-5--7-5-3-3-3--0-1-3--0-1-3--0-1-3--0-1-3-1-0----]
B |3-------------------------------------------------3-1]
G |-------------------------------------------------]
D |-------------------------------------------------]
A |-------------------------------------------------]
E |-------------------------------------------------]
```

FONTE: Autores (2024)

FIGURA 3 - EXEMPLO CIFRA



FONTE: Adaptado de Cifra Club (2024)

2.5 TRANSCRIÇÃO MUSICAL

De acordo com Benetos et al. (2018), converter música acústica em alguma forma de notação musical, é uma tarefa desafiadora em processamento de sinais e inteligência artificial, este processo envolve percepção (analisar cenas auditivas complexas), cognição (reconhecer objetos musicais), representação de conhecimento (formando estruturas musicais) e inferência (testando alternativas hipóteses).

O desafio da transcrição musical decorre, em grande parte, da dificuldade em reunir dados suficientes para o treinamento de um modelo confiável e preciso. Além disso, o processo exige um alto custo computacional, uma vez que é necessário identificar e processar diversos elementos musicais, como acordes, ritmos, tempos e notas, entre outros.

2.6 MIDI

O MIDI (*Musical Instrument Digital Interface*) foi criado em 1983 como resultado de uma colaboração entre diversos fabricantes de instrumentos musicais dos Estados Unidos e do Japão. Esse padrão de comunicação de dados surgiu com o objetivo de

possibilitar a integração e a comunicação entre diferentes instrumentos musicais eletrônicos e computadores, superando as limitações de compatibilidade que existiam anteriormente (KERR, 2009).

Kerr (2009) relata que o MIDI, não armazena áudio, mas sim uma série de instruções que descrevem eventos musicais. Ao invés de capturar o som real de um instrumento, o MIDI registra informações como qual tecla foi pressionada, a intensidade com que foi tocada e a duração da nota. Isso proporciona uma flexibilidade significativa, pois o mesmo arquivo MIDI pode controlar diferentes instrumentos, variando desde sintetizadores até softwares de produção musical.

Além de definir um conjunto de mensagens digitais, o padrão MIDI estabelece uma interface física que permite a conexão entre instrumentos e computadores. As mensagens digitais cobrem uma vasta gama de comandos musicais, incluindo a indicação da nota a ser tocada, a intensidade do toque, e as variações de volume ao longo de uma composição. Esse padrão se tornou essencial na produção musical moderna, uma vez que facilita a composição, edição e reprodução de músicas de forma eficiente e versátil (HUBER, 2013).

2.7 SEPARAÇÃO DE FAIXAS MUSICAIS

A separação de faixas musicais é uma técnica de processamento de áudio cuja função é isolar diferentes componentes de uma música, como vocais, bateria, baixo e outros instrumentos. Esta técnica é utilizada em várias aplicações, incluindo remixagem, remasterização, transcrição musical e análise musicológica. A separação de fontes musicais utiliza diversas abordagens como o processamento de sinais, aprendizado de máquina e análise espectral (VINCENT; FÉVOTTE; GRIBONVAL, 2006).

2.7.1 Abordagens Baseadas em Processamento de Sinais

Uma das abordagens de separação de fontes musicais envolve técnicas de processamento de sinais, como a Transformada de Fourier para análise espectral. Métodos como a Fatoração de Matrizes Não-Negativas (NMF) decompõem o

espectrograma da gravação em componentes aditivos, permitindo a identificação e a separação das diferentes fontes sonoras.

Conforme discutido por Risoud et al. (2018), a NMF é eficaz para separar componentes que possuem padrões espectrais distintos, como vozes e instrumentos.

2.7.2 Abordagens Baseadas em Redes Neurais

Outra abordagem utilizada na separação de fontes musicais envolve o uso de redes neurais convolucionais (RNCs) e redes neurais recorrentes (RNRs), que são amplamente empregadas para aprender representações complexas das fontes sonoras a partir de grandes conjuntos de dados de áudio. JANSSON et al. (2017) descreve o processo de separação usando RNCs da seguinte maneira: os dados de áudio são convertidos em espectrogramas, que são representações visuais da amplitude do sinal em função do tempo e da frequência.

As RNCs aplicam filtros convolucionais ao espectrograma para extrair características relevantes das fontes sonoras, em conjunto operações de *pooling* são usadas para reduzir a dimensionalidade dos dados e destacar as características mais importantes. Após várias camadas convolucionais e *pooling*, os dados são passados por camadas densas (camadas totalmente conectadas) que ajudam a combinar as características extraídas e tomar decisões sobre a separação das fontes.

Grossman e Grossman (2020) empregaram a seguinte abordagem como base para o modelo de Transcrição Automática de Música. Utilizando o conjunto de dados MAESTRO, os autores converteram os arquivos de áudio em espectrogramas a fim de alimentar as Redes Neurais Convolucionais (RNCs) para detectar padrões e características, permitindo que o modelo identificasse quais notas musicais estavam ativas em momentos específicos durante as gravações de áudio. Para gerar os espectrogramas, foi utilizado a biblioteca librosa, baseada em Python. Aplicando a Transformada Constante-Q (CQT), uma variante da transformada de Fourier, projetada para representar sinais musicais. A CQT é particularmente adequada para música pois fornece uma escala de frequência logarítmica, alinhando-se bem com a forma como os seres humanos percebem o tom. Os espectrogramas resultantes mostram a distribuição de potência através de diferentes bandas de frequência ao longo do tempo, servindo como entrada para a rede neural.

A separação por RNRs pode ser resumida como o processamento do áudio de maneira continua através de uma sequência de eventos ao longo do tempo, permitindo que a rede capture dependências temporais. As células de memória das RNRs, como LSTMs (*Long Short-Term Memory*) ou GRUs *(Gated Recurrent Units*), ajudam a preservar informações relevantes sobre o contexto temporal. Dessa forma, as RNRs modelam a evolução temporal das características musicais, facilitando a identificação e a separação de diferentes fontes sonoras que variam ao longo do tempo (CHAN et al. 2015).

Existem também modelos híbridos que utilizam as abordagens descritas em conjunto para obter separações de faixas musicais de maneira mais precisa. Nesse contexto entra o Demucs, que faz a separação de faixas musicais utilizando uma arquitetura que combina convoluções temporais e redes recorrentes, permitindo a modelagem de dependências temporais e a captura de características locais e globais das fontes musicais (DÉFOSSEZ et al., 2022).

Ao analisar o desempenho do Demucs em comparação com outros softwares de separação de faixas musicais, é possível observar que a combinação de convoluções temporais e redes recorrentes permite ao Demucs alcançar resultados superiores em diversos cenários. Softwares tradicionais, como o Spleeter, que utiliza redes neurais convolucionais puras, demonstram bom desempenho em termos de velocidade e qualidade, porém, carecem da mesma precisão em capturar dependências temporais mais complexas, especialmente em faixas com instrumentos que possuem variações dinâmicas significativas (HENNEQUIN; KOPPENBERGER; MAMEI, 2020).

Stöter, Liutkus e Ito (2019) apresentam o Open-Unmix, que também baseado em redes neurais, foca em um espectro de frequência com menos ênfase nas dependências temporais, o que pode resultar em uma separação menos natural em músicas com ritmos complexos. O Demucs, ao incorporar tanto características locais quanto globais, se diferencia ao produzir separações mais suaves e naturais, especialmente em faixas onde a separação clara entre instrumentos é crítica para a qualidade final.

Portanto, embora todos esses softwares utilizem redes neurais em sua abordagem, a arquitetura híbrida do Demucs proporciona uma vantagem significativa ao combinar o melhor de diferentes abordagens, resultando em uma qualidade

superior de separação, conforme demonstrado em diversos benchmarks e estudos comparativos recentes (DÉFOSSEZ et al., 2022).

2.8 REDES NEURAIS CONVULACIONAIS

Utilizada em tarefas de visão computacional e aprendizado de máquina redes neurais convulacionais (RNCs), análogas às redes neurais artificiais tradicionais, na medida em que são compostas por neurônios que se auto otimizam por meio do aprendizado, são primariamente utilizadas nos reconhecimentos de padrões em imagens (O'SHEA; NASH, 2015).

As RNCs funcionam através de uma série de camadas que processam os dados de entrada sequencialmente. O processo inicia com uma imagem que é representada como um tensor de ordem 3 (com dimensões altura, largura e profundidade/canais de cor).

A primeira camada de uma RNC é a camada de convolução, na qual filtros (ou *kernels*) são aplicados sobre a imagem para extrair características fundamentais, como bordas, texturas e padrões simples. Esses filtros deslizam sobre a imagem original, realizando a operação de convolução e gerando mapas de ativação, também chamados de *feature maps*. Esses mapas destacam características específicas da imagem, que são relevantes para a tarefa de classificação ou reconhecimento (GOODFELLOW; BENGIO; COURVILLE, 2016).

Após a convolução, é comum que as RNCs utilizem uma camada de *pooling*, que tem como função reduzir a dimensionalidade dos mapas de ativação, preservando as características mais importantes e eliminando as menos relevantes. O *max pooling*, por exemplo, seleciona o valor máximo dentro de uma região específica do mapa, o que diminui a quantidade de parâmetros e a complexidade computacional, além de reduzir o risco de *overfitting* (LI et al., 2020).

O termo *overfitting* refere-se a uma situação em que o modelo aprende com demasiada precisão os detalhes e ruídos presentes nos dados de treinamento, ao ponto de capturar padrões que não são generalizáveis para novos dados. Em outras palavras, o modelo se ajusta tão bem aos dados de treinamento que perde a capacidade de generalizar para dados desconhecidos, resultando em um desempenho inferior quando exposto à novas informações (GOODFELLOW; BENGIO; COURVILLE, 2016).

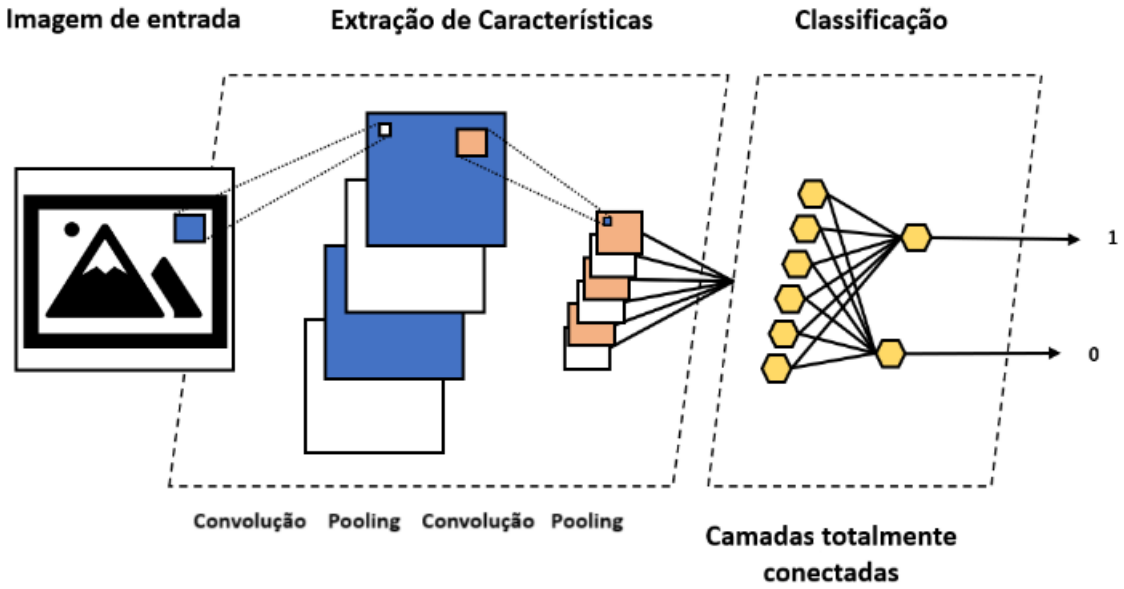
Ainda segundo Goodfellow, Bengio e Courville (2016), o *overfitting* ocorre, geralmente, quando o modelo é excessivamente complexo em relação à quantidade e variabilidade dos dados de treinamento. Por exemplo, se uma RNC possui muitas camadas ou neurônios, ela pode acabar memorando as particularidades dos dados de treinamento, em vez de aprender características relevantes que possam ser aplicadas a novos conjuntos de dados. Para mitigar o *overfitting*, técnicas como a regularização, a utilização de conjuntos de dados maiores e a aplicação de métodos como *dropout* podem ser empregadas, ajudando a manter o equilíbrio entre a capacidade de aprendizado do modelo e sua generalização.

À medida que os dados avançam por múltiplas camadas de convolução e *pooling*, a rede se torna capaz de capturar características cada vez mais complexas da imagem, como formas e objetos completos. Finalmente, após passar por diversas camadas, os dados são encaminhados para camadas totalmente conectadas (*fully connected layers*), que atuam como classificadores, gerando a saída final da rede, seja a identificação de uma classe de objeto ou outra forma de reconhecimento (O'SHEA; NASH, 2015).

As RNCs demonstram grande eficiência em uma variedade de aplicações, incluindo a detecção de objetos, reconhecimento facial e diagnósticos médicos baseados em imagens. A principal vantagem dessas redes está em sua capacidade de aprender automaticamente as características relevantes para uma tarefa, dispensando a necessidade de intervenções manuais na engenharia de características (LECUN; BENGIO; HINTON, 2015).

A FIGURA 4, ilustra o processo de uma Rede Neural Convolucional o qual envolve várias etapas, incluindo a aplicação de camadas de convolução, *pooling*, extração de características, e camadas completamente conectadas para a classificação final.

FIGURA 4 - EXEMPLO REDE NEURAL CONVULACIONAL



FONTE: Autores (2024)

2.9 SOFTWARES SEMELHANTES

O *Audacity* é um software livre e de código aberto para edição e gravação de áudio digital, amplamente utilizado em diversas plataformas, incluindo Windows, macOS e Linux. Desde seu desenvolvimento inicial no ano 2000, destaca-se por oferecer diversas ferramentas para manipulação de áudio (AUDACITY TEAM, 2024).

O software emprega diversos métodos e algoritmos para a análise e processamento de sinais de áudio, permitindo aos usuários realizar tarefas como edição, mixagem e aplicação de efeitos sonoros. Um dos principais métodos de análise utilizados pelo software é a Transformada de Fourier Rápida (FFT), que converte sinais de áudio do domínio do tempo para o domínio da frequência. Essa transformação possibilita visualizar e analisar o espectro de frequências de uma gravação, identificando componentes tonais e harmônicos presentes no áudio. Além disso, a aplicação conta com uma ferramenta de espectrograma, que fornece uma representação visual detalhada, facilitando a identificação de padrões e características específicas no sinal de áudio (AUDACITY TEAM, 2024).

Devido à sua flexibilidade e recursos avançados, o sistema é utilizado em diversas áreas, como produção musical, *podcasting*, radiodifusão, educação e pesquisa. No âmbito educacional, segundo o departamento de desenvolvimento de

professores da Universidade Estadual da Califórnia, o *Audacity* pode ser utilizado como uma ferramenta de ensino e aprendizado, sendo um software flexível para a produção de podcasts e edição de áudio em projetos educacionais (CALIFORNIA STATE UNIVERSITY, 2024). Em *podcasting* e radiodifusão, é utilizado para gravação e edição de podcasts, programas de rádio e conteúdo para transmissão. Em pesquisa e análise de áudio, permite uma análise detalhada de sinais para estudos em áreas como fonética, bioacústica e engenharia de som (THE UNIVERSITY OF TENNESSEE, 2024).

Outro software da mesma categoria é o *Sibelius*, software profissional de notação musical desenvolvido pela Avid Technology. O programa oferece ferramentas para a criação, edição e impressão de partituras, através de múltiplas opções para a inserção de notas, incluindo o uso de teclado MIDI, teclado do computador, mouse ou dispositivos de entrada por toque, que inclui recursos para edição detalhada da partitura, como ajuste de espaçamento, formatação de páginas, estilos de texto e símbolos personalizados (AVID Technology, 2024).

3 MATERIAIS E MÉTODOS

Capítulo onde são descritos os materiais utilizados e os métodos empregados no planejamento e desenvolvimento do sistema. A abordagem metodológica adotada é detalhada para fornecer uma compreensão clara das etapas e ferramentas envolvidas no desenvolvimento do software.

3.1 MÉTODOS

Nesta seção, serão descritos os métodos empregados no planejamento e desenvolvimento do sistema.

3.1.1 Desenvolvimento do Software

O projeto foi concebido com o objetivo de desenvolver um software capaz de transcrever partituras utilizando um modelo baseado em Redes Neurais Convolucionais. Para tanto, foi planejado o uso de bibliotecas como PyTorch e Librosa, em conjunto com o conjunto de dados MAESTRO, disponibilizado pela Google, que resulta de 10 anos de coleta de dados do International Piano-e-Competition, totalizando aproximadamente 200 horas de gravações de áudio e arquivos MIDI, com um tamanho total de 120 GB.

Tomando como referência o trabalho de Grossman e Grossman (2020), foi aplicada e configurada a Transformada Constante-Q a partir de um arquivo de áudio no formato .WAV, gerando, assim, uma sequência de espectrogramas.

Como parâmetros:

- *n_bins*: Definido como 288, esse valor representa o número total de *bins* de frequência no espectrograma, abrangendo várias oitavas. Refere-se ao número total de *bins* de frequência que serão gerados. Cada *bin* corresponde a uma banda de frequência específica. Este parâmetro controla a resolução do espectrograma. Um valor maior de *bins* significa maior resolução;
- *bins_per_octave*: Configurado como 36, este parâmetro define a quantidade de *bins* por oitava. O valor de 36 foi selecionado para proporcionar uma distribuição granular das frequências ao longo das oitavas. Esta escolha

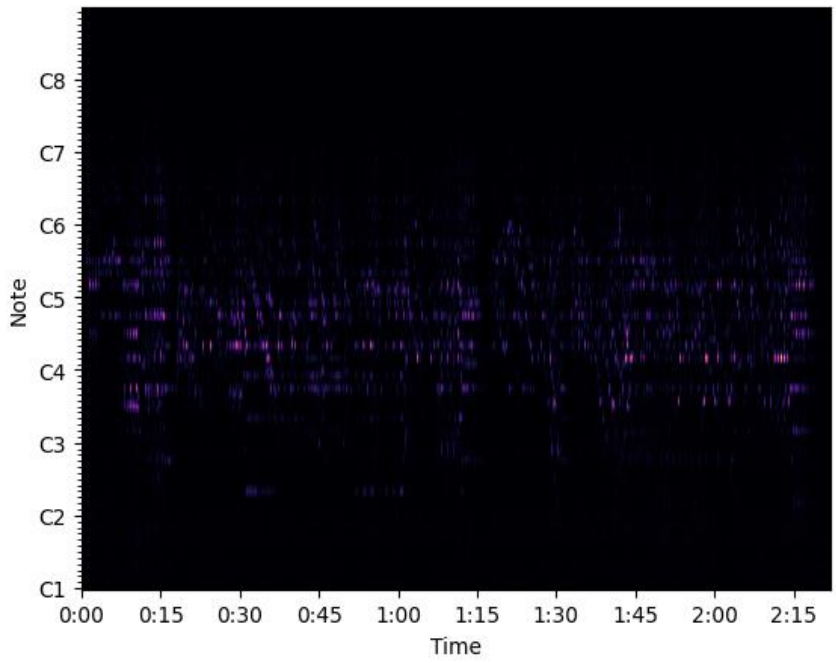
permite que o sistema capture as variações sutis dentro de cada oitava, que são importantes para distinguir notas próximas e melhorar a precisão da transcrição.

- *filter_scale*: Esse parâmetro foi ajustado para 0,5, o que resulta em filtros mais estreitos no domínio da frequência. Um valor menor de *filter_scale* proporciona maior precisão na detecção de frequências, embora com um leve comprometimento da precisão temporal. Esse ajuste foi realizado com o objetivo de priorizar a clareza na identificação de frequências específicas, essenciais para a transcrição de notas musicais, mantendo-se, ainda assim, uma resolução temporal aceitável.

Na análise espectral, um *'bin'* representa uma subdivisão da faixa de frequências analisada, onde cada *bin* corresponde a uma banda específica de frequências (LYONS, 2011)."

A Figura 5 ilustra um espectrograma obtido durante nossos testes, utilizando os parâmetros mencionados. Este espectrograma demonstra como as frequências das notas musicais são representadas ao longo do tempo.

FIGURA 5 - EXEMPLO DO ESPECTROGRAMA



FONTE: Autores (2024)

Em seguida, utilizar o *framework* PyTorch para criar uma rede neural convulacional e fornecer como material de treino os espectrogramas gerados. O modelo então seria treinado para aprender a mapear os padrões de frequência e tempo dos espectrogramas para as notas musicais correspondentes, visando a transcrição automática de partituras. Além disso, consideramos a geração de *piano rolls* dos arquivos MIDI correspondentes, proporcionando uma representação complementar para análise musical e potencial aprimoramento do modelo.

Todavia, a utilização eficaz dos dados usados para o treinamento do modelo exige uma compreensão técnica extensa, especialmente para associar corretamente as notas musicais aos elementos correspondentes nos espectrogramas. Entre os desafios encontrados, cita-se a sincronização temporal entre as notas extraídas dos arquivos de *piano rolls* e as frequências representadas nos espectrogramas, onde pequenas discrepâncias temporais ou de frequência resultaram em transcrições imprecisas, evidenciando a complexidade intrínseca deste processo. Além disso, o processo de correlacionar esses elementos sobrecarregou os recursos computacionais disponíveis, bem como o tempo necessário para realizar as operações.

A análise dos arquivos de áudio e MIDI de uma única música gerava, em média, 150 MB de dados, devido à densidade das informações presentes nos arquivos de *piano rolls* e a quantidade de imagens de espectrogramas geradas. A quantidade de dados e o processamento intensivo necessários para lidar com essas representações evidenciaram os limites computacionais do projeto. A infraestrutura disponível mostrou-se insuficiente para sustentar o treinamento do modelo, conforme o planejado inicialmente.

Diante desses desafios, a decisão foi tomada para redirecionar o foco deste requisito. Em vez de continuar a busca por uma transcrição precisa baseada em espectrogramas e *piano rolls*, optou-se por concentrar a aplicação na transcrição de arquivos MIDI, uma tarefa consideravelmente menos exigente em termos de processamento. Essa reorientação permitiu que o projeto continuasse a progredir sem a necessidade de treinar novos modelos, aproveitando os dados já disponíveis de forma mais eficiente.

Através desta adaptação, não apenas mitigaram-se os problemas associados à sobrecarga dos recursos computacionais, mas também se garantiu a viabilidade do projeto dentro das limitações práticas enfrentadas. O foco na transcrição MIDI, uma

abordagem mais direta e compatível com os recursos à disposição, assegurou que os objetivos principais do projeto fossem mantidos.

3.1.2 Kanban

Baseado em alguns princípios que visam melhorar o fluxo de trabalho e a gestão de tarefas, a metodologia Kanban é uma abordagem amplamente utilizada no desenvolvimento de sistemas, conhecida por sua capacidade de melhorar a eficiência, a comunicação e a flexibilidade das equipes de trabalho (LADAS, 2009).

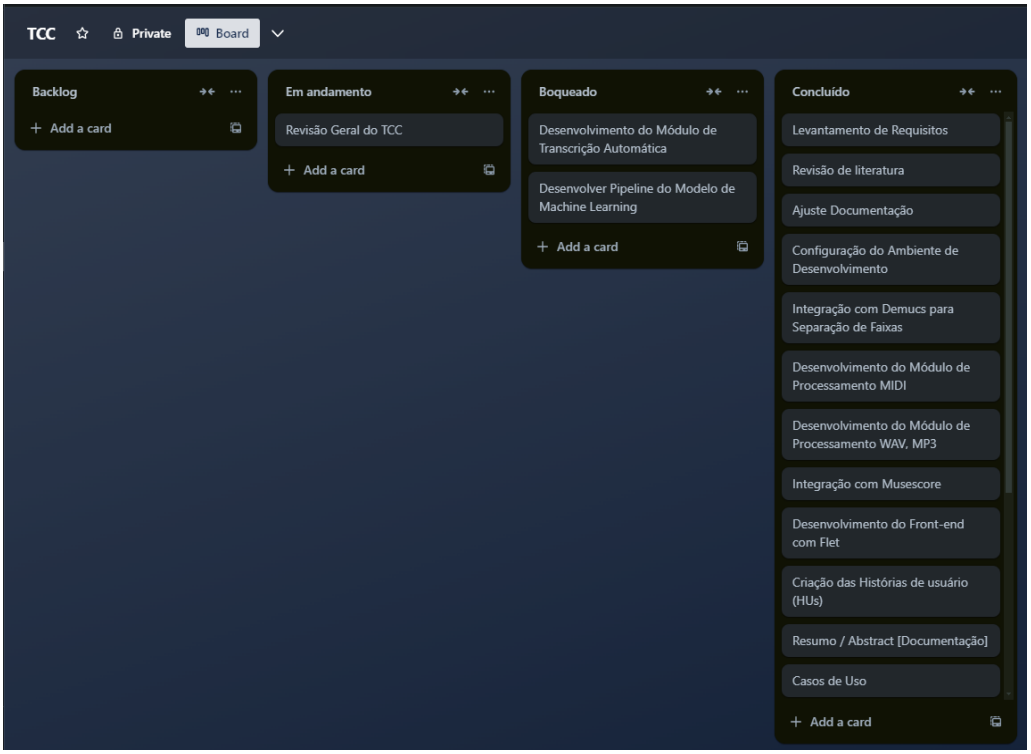
Uma de suas principais características é a visualização do trabalho em andamento. Isso é feito através de um quadro exclusivo, onde as tarefas são representadas por cartões que se movem através de diferentes colunas que representam os estágios do processo de desenvolvimento. Esta visualização facilita a identificação de gargalos e áreas de melhoria (ANDERSON, 2010).

Outra prática crucial da aplicação da metodologia, é a limitação do número de tarefas que podem estar em progresso simultaneamente. Isso ajuda a garantir que as equipes mantenham um foco claro em completar as tarefas antes de iniciar novas (KNIBERG; SKARIN, 2010), evitando sobrecarga e promovendo também a medição e a gestão do fluxo de trabalho. Monitorar o tempo que as tarefas levam para atravessar o processo ajuda a identificar ineficiências e áreas que necessitam de ajustes (LADAS, 2009).

Durante o planejamento do desenvolvimento do sistema, foi adotada a metodologia e práticas do Kanban, considerada uma abordagem simples, intuitiva e altamente eficaz para organizar e visualizar tanto as partes concluídas quanto as em progresso do software.

A prática também provou ser uma excelente ferramenta para apoiar as sprints e facilitar a distribuição de tarefas entre os membros da equipe. Utilizamos a ferramenta Trello para configurar um quadro personalizado e compartilhável, no qual definimos listas de cartões para “Concluído”, “Em andamento”, “*Backlog*” e “Bloqueado” conforme ilustrado na Figura 6.

FIGURA 6 - KANBAN



FONTE: Autores (2024)

3.1.3 Divisão de tarefas

A equipe, composta por dois integrantes, optou por uma abordagem predominantemente colaborativa, realizando a maior parte do projeto em conjunto por meio de programação em par. Essa estratégia permitiu maior sinergia e troca de ideias durante o desenvolvimento. Apenas algumas tarefas específicas foram divididas entre os membros para serem realizadas individualmente. As reuniões remotas periódicas foram conduzidas utilizando o Microsoft Teams, complementado pelo aplicativo Discord, facilitando a comunicação e o alinhamento contínuo das atividades.

3.1.4 Cronograma

O projeto foi originalmente planejado para seguir dois cronogramas, distribuídos ao longo de dois semestres. No entanto, apenas o primeiro cronograma, conforme apresentado no Quadro 1, seguiu como previsto. Durante o segundo semestre, o desenvolvimento do sistema foi interrompido devido a dificuldades previamente mencionadas, o que exigiu uma reavaliação do projeto.

QUADRO 1 - CRONOGRAMA DO PRIMEIRO SEMESTRE

Data de Início	Data de Término	Atividade
03/04/2023	17/04/2023	Revisão de literatura
17/04/2023	01/05/2023	Levantamento de Requisitos
01/05/2023	15/05/2023	• Elaboração de Diagramas de caso de uso • Elaboração Histórias de Usuário • Elaboração diagrama de classes • Elaboração de diagramas de sequência
15/05/2023	29/05/2023	Prototipação de telas
29/05/2023	12/06/2023	Elaboração do documento: • Resumo • Introdução • Fundamentação teórica
12/06/2023	26/06/2023	Elaboração do documento: • Materiais e Métodos • Ajustes

FONTE: Autores (2024)

Como resultado dessa reavaliação, foi elaborado um novo cronograma, no qual os ajustes necessários foram realizados para alinhar as atividades restantes às novas condições do projeto. Esse cronograma ajustado é detalhado no Quadro 2, refletindo as mudanças implementadas para garantir a continuidade e a conclusão do desenvolvimento do sistema.

QUADRO 2 - CRONOGRAMA DO SEGUNDO SEMESTRE

Data de Início	Data de Término	Atividade
23/11/2023	01/11/2023	• Revisão de literatura • Levantamento de Requisitos
01/11/2023	15/11/2023	• Protótipo do aplicativo e prova de conceito
25/03/2024	08/04/2024	Desenvolvimento da seção de configuração: • Implementação de arquivos persistentes de configuração

08/04/2024	22/04/2024	Desenvolvimento da seção de mixagem musical: • Criação da interface gráfica para manipulação de arquivos
22/04/2024	06/05/2024	Desenvolvimento da seção de mixagem musical: • Integração com a API Demucs, implementação da funcionalidade de separação de fontes
06/05/2024	20/05/2024	Desenvolvimento da seção partituras: • Implementação da lógica para manipulação e conversão de arquivos
20/05/2024	03/06/2024	Desenvolvimento da seção partituras: • Implementação da criação de partituras no formato PDF
03/06/2024	13/10/2024	Ajustes na documentação

FONTE: Autores (2024)

3.1.5 Requisitos

Nesta seção, são apresentados os requisitos essenciais para o desenvolvimento do sistema proposto. Os requisitos foram levantados a partir de uma análise sobre as necessidades funcionais e não funcionais do sistema, com o objetivo de assegurar que o sistema atenda às expectativas dos usuários e desempenhe suas funções corretamente.

A seguir são detalhados os requisitos funcionais do sistema:

- RF1: O sistema deve permitir que o usuário faça o upload de um arquivo de áudio.
- RF2: O sistema deve ser capaz de processar o arquivo de áudio e extrair as informações musicais relevantes.
- RF3: O sistema deve ser capaz de gerar uma partitura a partir do arquivo de áudio.
- RF4: O sistema deve permitir que o usuário baixe a partitura gerada.

- RF5: O sistema deve ser capaz de separar os componentes individuais de um arquivo de áudio complexo.
- RF6: O sistema deve fornecer suporte a vários formatos de arquivo de áudio, como MP3 e WAV.
- RF7: O sistema deve ser capaz de processar arquivos de áudio de diferentes durações, desde pequenos clipes até peças musicais mais longas.

A seguir são detalhados os requisitos não-funcionais do sistema:

- RNF1: O sistema deve ter uma interface gráfica de usuário intuitiva.
- RNF2: O sistema deve ser capaz de funcionar com uma ampla variedade de tipos de música e gêneros.
- RNF3: O sistema deve ser capaz de lidar com erros e falhas de processamento de forma amigável, fornecendo mensagens de erro claras e sugestões de como solucioná-las.
- RNF4: O sistema deve ser capaz de funcionar em diferentes sistemas operacionais, como Windows, macOS e Linux.

3.1.6 Especificação dos requisitos

Com base nos requisitos levantados, foi realizado o detalhamento dos requisitos funcionais, conforme apresentado no Quadro 3.

QUADRO 3 - ESPECIFICAÇÃO DOS REQUISITOS FUNCIONAIS

RF1	O usuário deve ser capaz de navegar pelas pastas do sistema operacional e selecionar um arquivo de áudio de sua escolha para upload. Esse processo deve ser fácil e intuitivo, e a interface deve claramente indicar o progresso do upload.
RF2	O sistema deve usar algoritmos avançados de processamento de sinal e inteligência artificial para identificar e extrair as características musicais do áudio, como ritmo, melodia, harmonia, e outros elementos sonoros.
RF3	Baseado nas informações musicais extraídas, o sistema deve converter os dados em uma partitura musical visual, com notas, claves, ritmos, dinâmicas etc.
RF4	O usuário deve ser capaz de baixar a partitura criada no formato PDF.

RF5	O sistema deve usar técnicas de separação de fontes de áudio para descompor uma música em seus componentes individuais (como vocais, bateria, guitarra etc.), se necessário.
RF6	O sistema deve aceitar os formatos de arquivo de áudio mais comuns, como MP3 e WAV, e ser capaz de processar esses arquivos de maneira eficiente e eficaz.
RF7	O sistema deve ser flexível e eficiente o suficiente para processar arquivos de áudio de qualquer duração, desde clipes curtos até composições mais longas.

FONTE: Autores, 2024

No Quadro 4, são detalhados os requisitos não funcionais.

QUADRO 4 - ESPECIFICAÇÃO DOS REQUISITOS NÃO FUNCIONAIS

RNF1	A interface gráfica de usuário (GUI) deve ser simples e direta, com um design limpo e sem elementos confusos ou desnecessários. Deve ser fácil para o usuário realizar todas as operações necessárias.
RNF2	O sistema deve ser treinado em uma ampla variedade de músicas e gêneros para garantir que possa lidar com uma ampla gama de estilos e características musicais.
RNF3	Em caso de erro, o sistema deve fornecer uma mensagem explicando o problema de maneira compreensível, juntamente com sugestões para resolvê-lo ou prevenir erros semelhantes no futuro.
RNF4	O sistema deve ser construído em uma plataforma que permita a execução em diferentes sistemas operacionais. Ele deve ser testado em Windows, macOS e Linux para garantir a compatibilidade e a funcionalidade em todas essas plataformas.

FONTE: Autores (2024)

3.1.7 Diagrama de Caso de Uso e Histórias de Usuário

O diagrama de caso de uso encontra-se disponível no APÊNDICE A. A especificação do caso de uso é feita pelas histórias de usuário que podem ser encontradas no APÊNDICE B.

3.1.8 Diagrama de Classes

Diagrama de classes referente a interação dos objetos. Todos os diagramas de casos de uso podem ser encontrados no APÊNDICE C.

3.1.9 Diagramas de Sequência

Diagramas de sequência das operações realizada pelo usuário. Todos os diagramas de sequência podem ser encontrados no APÊNDICE D.

3.2 MATERIAIS

Nesta seção, serão descritas as ferramentas de hardware e software empregadas no desenvolvimento do sistema.

3.2.1 Linguagem de Programação Python

Devido à complexidade do projeto, tornou-se necessário selecionar uma linguagem de programação que combinasse flexibilidade, confiabilidade e uma vasta gama de recursos, especialmente no que se refere à disponibilidade de bibliotecas. A escolha recaiu sobre o Python, que é amplamente reconhecido por sua versatilidade e aplicabilidade em diferentes áreas do desenvolvimento, como ciência de dados, automação e desenvolvimento web (LUTZ, 2014).

Entre os fatores que motivaram essa decisão, destaca-se a extensa quantidade de bibliotecas e *frameworks* que o Python oferece, proporcionando uma base robusta para atender às diversas demandas do projeto, desde o processamento de dados até a criação de interfaces. Além disso, a sintaxe clara e intuitiva do Python contribui para a facilidade de compreensão do código, minimizando a ocorrência de erros e otimizando o tempo de desenvolvimento (VANDERPLAS, 2016).

3.2.2 Librosa

Ferramenta importante para pré-processar e extrair características de áudio que são utilizadas como entrada em modelos de *machine learning*, além disso também é usada para estudar a estrutura e os padrões musicais, analisar ritmos, e entender características acústicas de gravações de músicas.

A biblioteca oferece diversas ferramentas para realizar transformações no tempo e na frequência, como transformada de Fourier, CQT (Constant-Q Transform), espectrogramas, e outros, dessa forma, sendo capaz de extrair uma ampla variedade

de características de áudio, como MFCCs (Mel Frequency Cepstral Coefficients), zero-crossing rate, tonalidade, ritmo (MCFEE et al., 2015).

3.2.3 Demucs

Demucs (*Deep Extractor for Music Sources*) é uma biblioteca *open-source* desenvolvida pelo Facebook AI Research (FAIR), que utiliza redes neurais profundas para a tarefa de separação de fontes musicais (DÉFOSSEZ, et al. 2019). A abordagem principal do Demucs baseia-se em arquiteturas de redes neurais convolucionais (RNCs) e recorrentes (RNRs), que são treinadas para separar diferentes componentes de uma mistura de áudio, como vocais, baixo, bateria e outros instrumentos (DÉFOSSEZ, 2021).

A arquitetura da biblioteca é composta por várias camadas de convoluções e unidades recorrentes, que permitem ao modelo capturar tanto as características locais quanto as dependências temporais do sinal de áudio (DÉFOSSEZ, et al. 2019). O modelo é treinado com uma grande quantidade de dados rotulados, onde as faixas de áudio são fornecidas com suas respectivas componentes isoladas (DÉFOSSEZ, et al. 2019).

Técnicas utilizadas pelo Demucs:

- Convoluções: As camadas convolucionais são responsáveis por extrair características locais do sinal de áudio, como padrões de frequência e texturas temporais (DÉFOSSEZ, 2022).
- Recorrentes: As camadas recorrentes, como LSTMs (Long Short-Term Memory), são usadas para modelar dependências temporais de longo alcance, capturando a evolução dos componentes ao longo do tempo (DÉFOSSEZ, 2022; BAYSAL; EFE, 2023).
- Estratégias de Treinamento: O treinamento do Demucs envolve o uso de técnicas avançadas de otimização e regularização para garantir que o modelo aprenda a separar os componentes de forma eficaz, minimizando artefatos e preservando a qualidade do áudio (DÉFOSSEZ, 2022; SCHAFFER et al., 2022).

Técnicas utilizadas pelo Demucs e outros softwares semelhantes, contribuíram para a área de separação de fontes musicais, oferecendo uma qualidade de separação significativamente superior às abordagens tradicionais (MITSUFUJI et al.,

2022; JEON; LEE, 2022; MANILOW et al., 2022). Algumas das principais aplicações incluem:

- Produção Musical: Permite a engenheiros de som e produtores musicais isolar e manipular componentes específicos de uma faixa de áudio (JEON; LEE, 2022).
- Remasterização e Restauração: A separação de componentes pode ser utilizada para melhorar a qualidade de gravações antigas ou danificadas (JEON; LEE, 2022).
- Pesquisa e Educação: Fornece ferramentas poderosas para a análise e estudo de músicas, auxiliando pesquisadores e estudantes na compreensão das técnicas de produção e composição (MANILOW et al., 2022; (LIU; KONG; LIU, 2021)).

3.2.4 PyDub

Biblioteca de manipulação de áudio em Python que permite o processamento de arquivos de áudio em diversos formatos, como WAV, MP3 e outros. Utilizada para tarefas de manipulação de áudio, como corte, mesclagem, conversão de formatos e aplicação de efeitos básicos (JIA, 2015).

3.2.5 PyTorch

PyTorch é uma biblioteca de código aberto para aprendizado profundo, desenvolvida originalmente pelo grupo de Pesquisa em Inteligência Artificial do Facebook. Distingue-se por sua execução dinâmica e imediata de operações de tensor, permitindo uma integração natural com o ecossistema Python, ao mesmo tempo que mantém uma performance comparável às demais bibliotecas. (PASZKE et al. 2019).

3.2.6 MuseScore

Software de notação musical gratuito e de código aberto que permite a criação, edição e reprodução de partituras (WERNER et al., 2023).

3.2.7 Flet

O Flet é um framework que se apresenta como uma solução eficiente para aplicações que necessitam de integração entre diferentes pilhas de desenvolvimento de maneira desacoplada e reativa, utilizando uma única linguagem de programação. A ferramenta elimina a necessidade de rotas tradicionais, ao concentrar-se na atualização dinâmica dos componentes dentro da mesma página (FLET TEAM, 2023). Atuando como uma interface para a linguagem Flutter, o Flet oferece uma variedade de componentes reativos que podem ser adicionados à uma página. Esses componentes reagem automaticamente às mudanças de estado, como cliques em botões ou alterações em formulários, assegurando a atualização dinâmica da interface do usuário. Entre os componentes disponíveis, destacam-se botões, tabelas, gráficos e formulários (FLET TEAM, 2023).

4 APRESENTAÇÃO DO SISTEMA

Este capítulo aborda a estrutura planejada para o sistema e suas funcionalidades junto à demonstração de suas respectivas telas.

4.1 ARQUITETURA

A arquitetura do sistema unifica *front-end* e *back-end* utilizando a linguagem Python, com a biblioteca Flet atuando como *framework* de interface do Flutter. A arquitetura segue o modelo monolítico *stateful* e implementa uma aplicação do tipo SPA (*Single-Page Application*). A Figura 7 ilustra o diagrama da arquitetura do sistema.

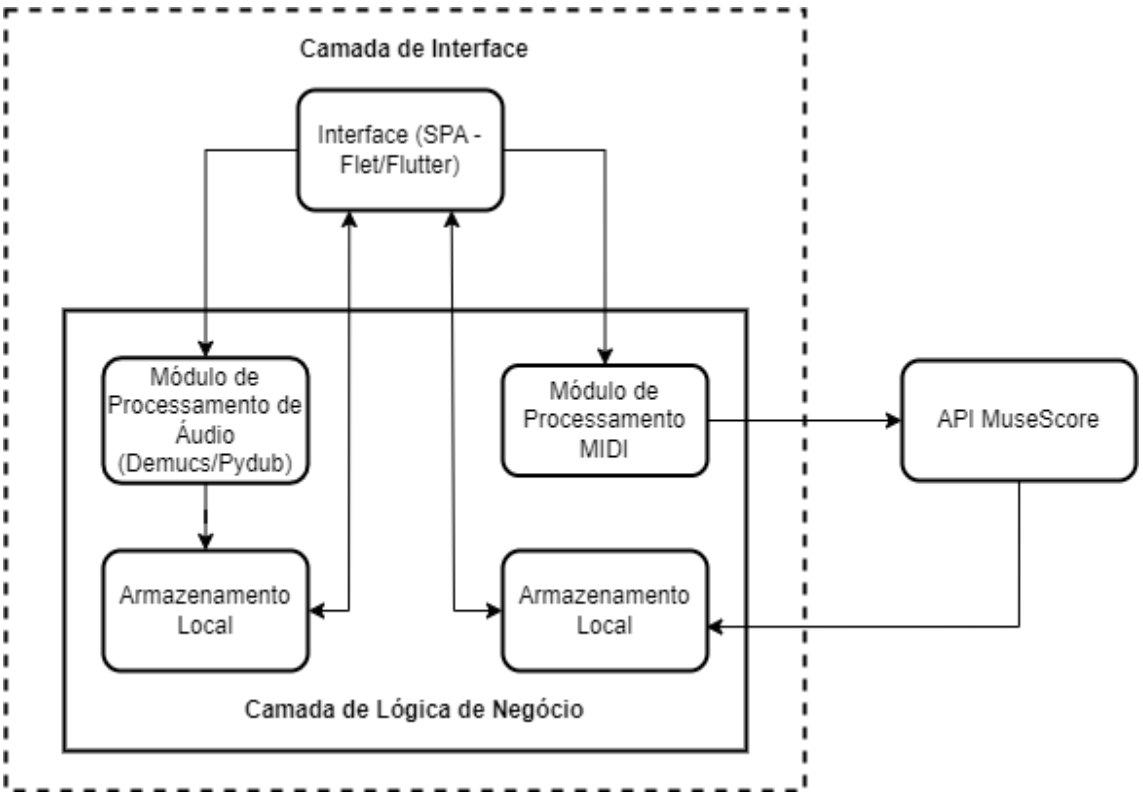
A camada de interface representa a interface direta com o usuário, ela se comunica diretamente com os módulos de processamento na camada de lógica de negócio, sendo responsável por exibir dados ao usuário e receber suas interações, como comandos para o processamento de arquivos de áudio e MIDI.

A camada de lógica de negócio, subdividida em dois módulos principais, realiza o processamento e manipulação dos arquivos de entrada e saída, cada qual com seu respectivo armazenamento local.

O módulo de processamento de áudio utiliza ferramentas como Demucs e Pydub para realizar a manipulação de arquivos de áudio, como separação de trilhas, edição ou conversão de formatos. Ele se conecta à interface para receber comandos e fornecer os resultados do processamento. O armazenamento local associado a esse módulo mantém os dados processados ou intermediários, permitindo o controle de versões dos arquivos ou resultados gerados.

O módulo de processamento de arquivos MIDI interage com o MuseScore por meio de uma API externa, que deve estar disponível no mesmo ambiente da aplicação, para manipulação e conversão de arquivos MIDI em partituras. Essas partituras são então armazenadas no repositório local.

FIGURA 7 - DIGRAMA DA ARQUITETURA



FONTE: Autores (2024)

Em uma aplicação SPA, a navegação entre diferentes seções é gerenciada por meio de *views* que são carregadas dinamicamente na mesma página. Cada *view* representa uma seção específica da aplicação, como a página inicial, *mixer* ou configurações. A página (*Page*) do Flet atua como um contêiner para todas as *views*.

A navegação entre as *views* é gerenciada por um *router*, que altera a *view* atual com base nas interações do usuário.

O sistema utiliza o padrão de publicação/assinatura (Pub/Sub) para a atualização dinâmica de *widgets*, garantindo que a interface do usuário esteja sempre sincronizada com o estado da aplicação. Os *widgets* (*subscribers*) se inscrevem em tópicos de interesse, enquanto as mudanças de estado são publicadas pelos componentes de negócio ou dados. Quando ocorre uma atualização, todos os *widgets* inscritos no tópico são notificados e atualizados automaticamente.

4.2 INTEGRAÇÕES

Para realizar algumas das funcionalidades de manipulação de áudio, foi realizada a integração com algumas bibliotecas junto à aplicação.

Dentre todas, destacamos especialmente a biblioteca Demucs, responsável pela separação de fontes musicais, permitindo a extração de diferentes instrumentos e vozes de uma faixa de áudio. Dessa forma, usuários podem carregar uma música e separar as faixas de voz e instrumentos para visualização ou posterior edição.

A implementação envolve a utilização de modelos pré-treinados do Demucs, que podem ser carregados e executados diretamente no Python. A integração com o Flet permite que os resultados do processamento de áudio sejam apresentados de forma intuitiva e interativa na interface do usuário, através de players de áudio que são utilizados para reproduzir as diferentes faixas separadas, facilitando a edição e análise.

4.3 TELAS

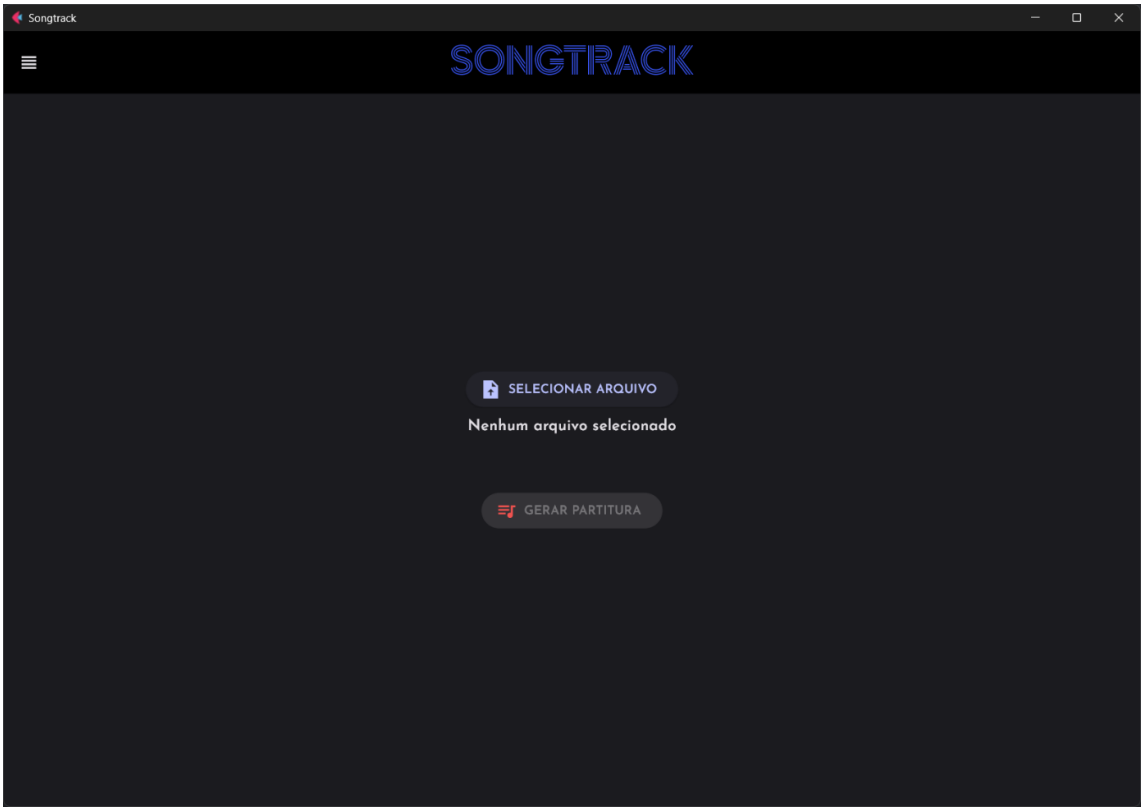
Seção onde serão apresentadas e descritas as telas do sistema.

4.3.1 Tela de Gerar Partitura

A Figura 8 apresenta a tela de gerar partitura do sistema. A tela conta com um menu de navegação que expande e colapsa horizontalmente a partir da ativação do botão no canto superior esquerdo. Exibe também 2 botões centrais ‘Selecionar Arquivo’ e ‘Gerar Partitura’ além de um texto dinâmico que informa ao usuário o arquivo selecionado.

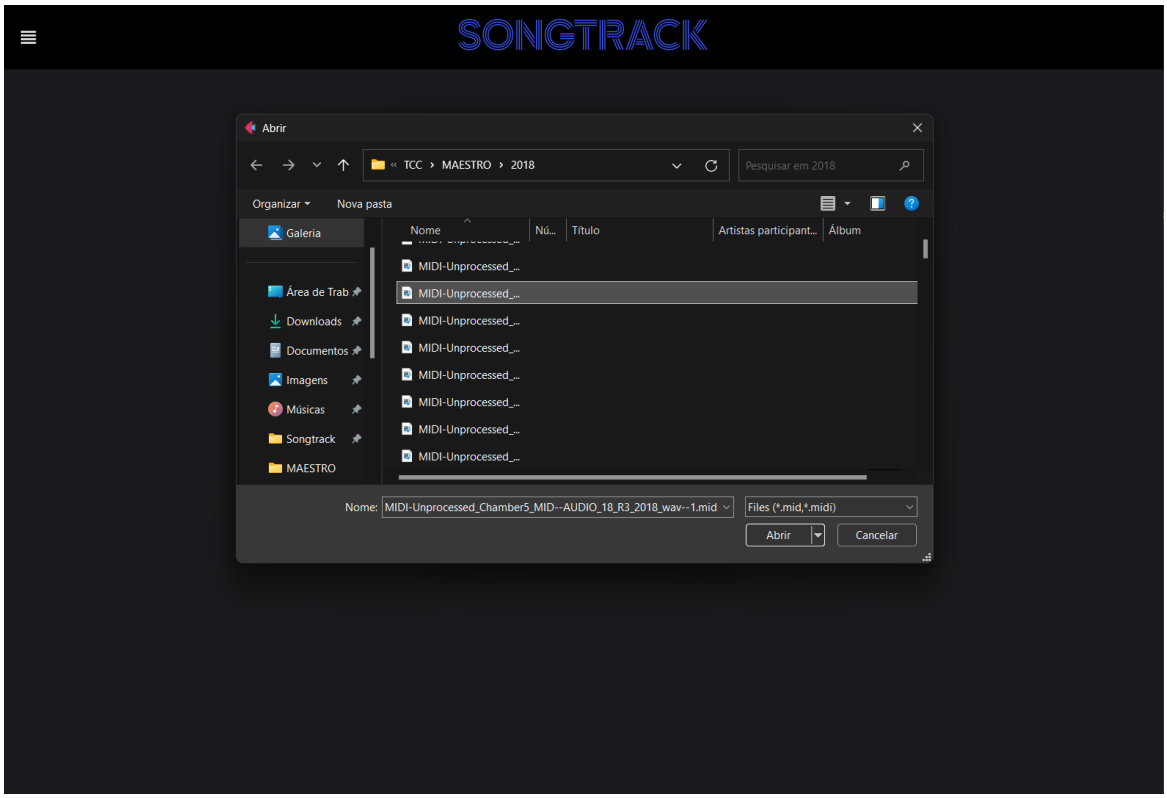
Ao clicar em "Selecionar Arquivo", uma janela é aberta para a seleção do arquivo desejado pelo usuário (Figura 9). Em seguida, ao acionar o botão "Gerar Partitura", o sistema inicia o processo de geração da partitura e exibe um pop-up com o nome do arquivo que está sendo processado e se o processo teve êxito ou não (Figura 10).

FIGURA 8 - TELA GERAR PARTITURA



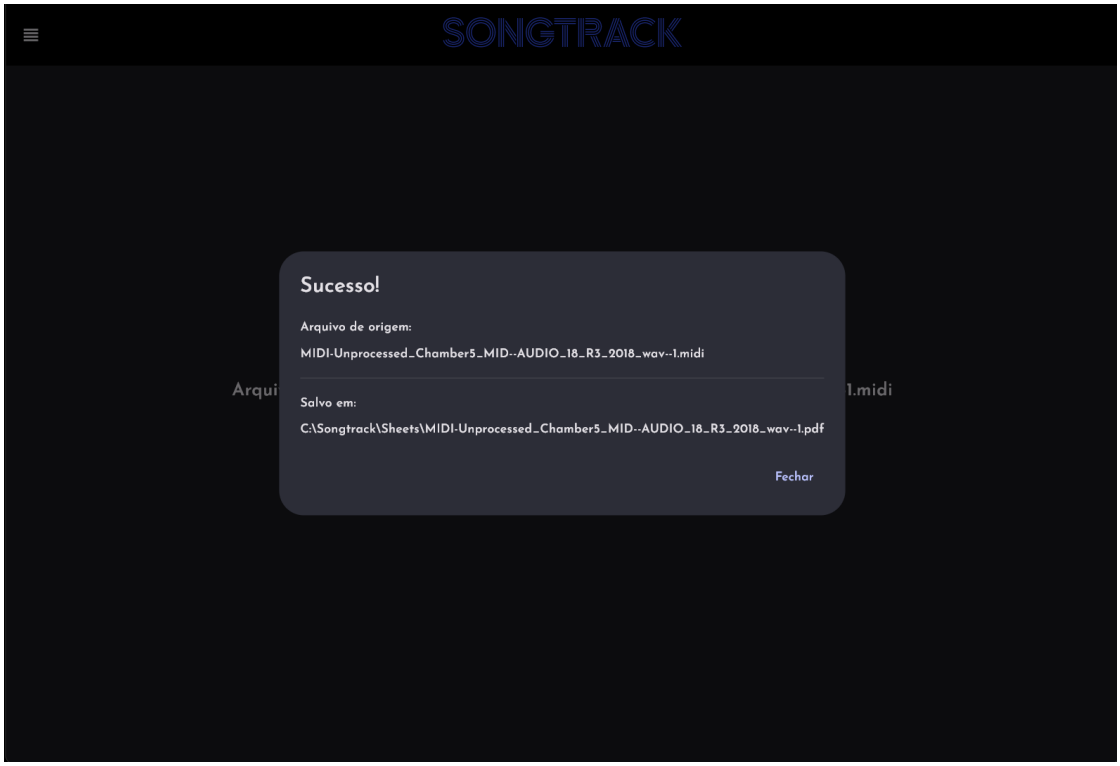
FONTE: Autores (2024)

FIGURA 9: SELECIONAR ARQUIVO



FONTE: Autores (2024)

FIGURA 10: ATIVAR GERAR PARTITURA

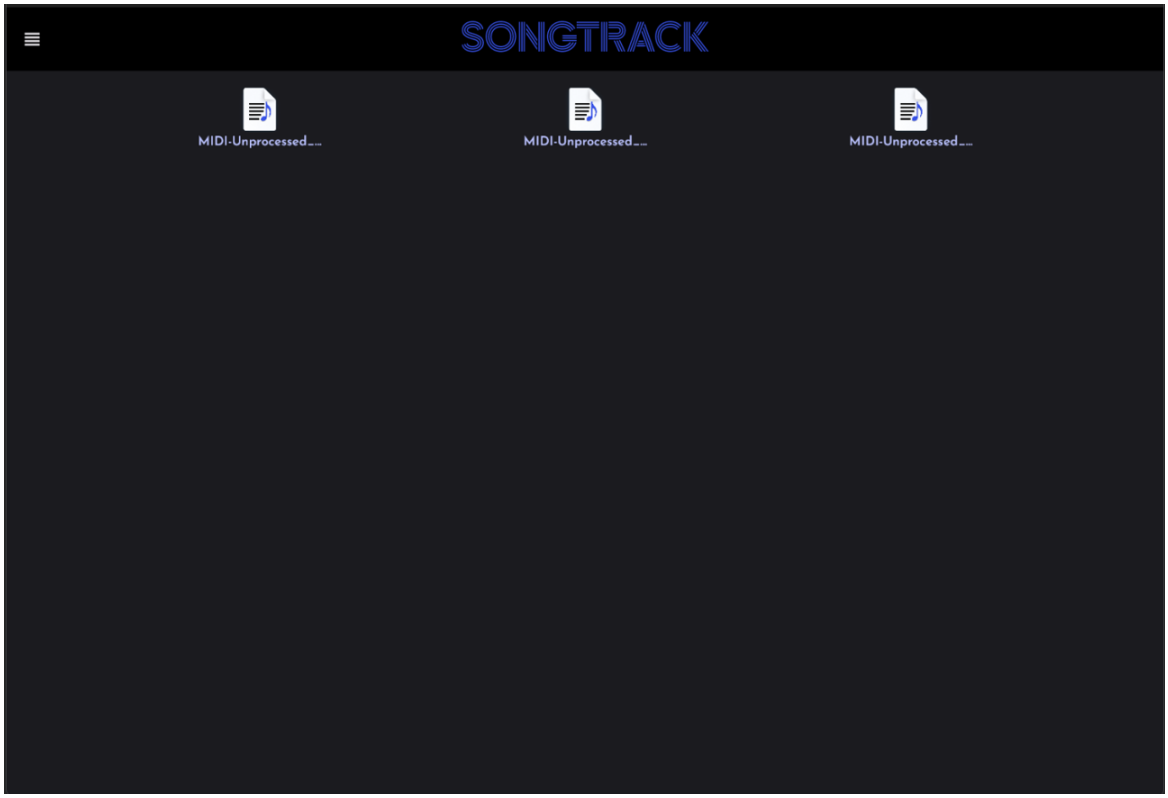


FONTE: Autores (2024)

4.3.2 Tela de Gerenciar Partituras

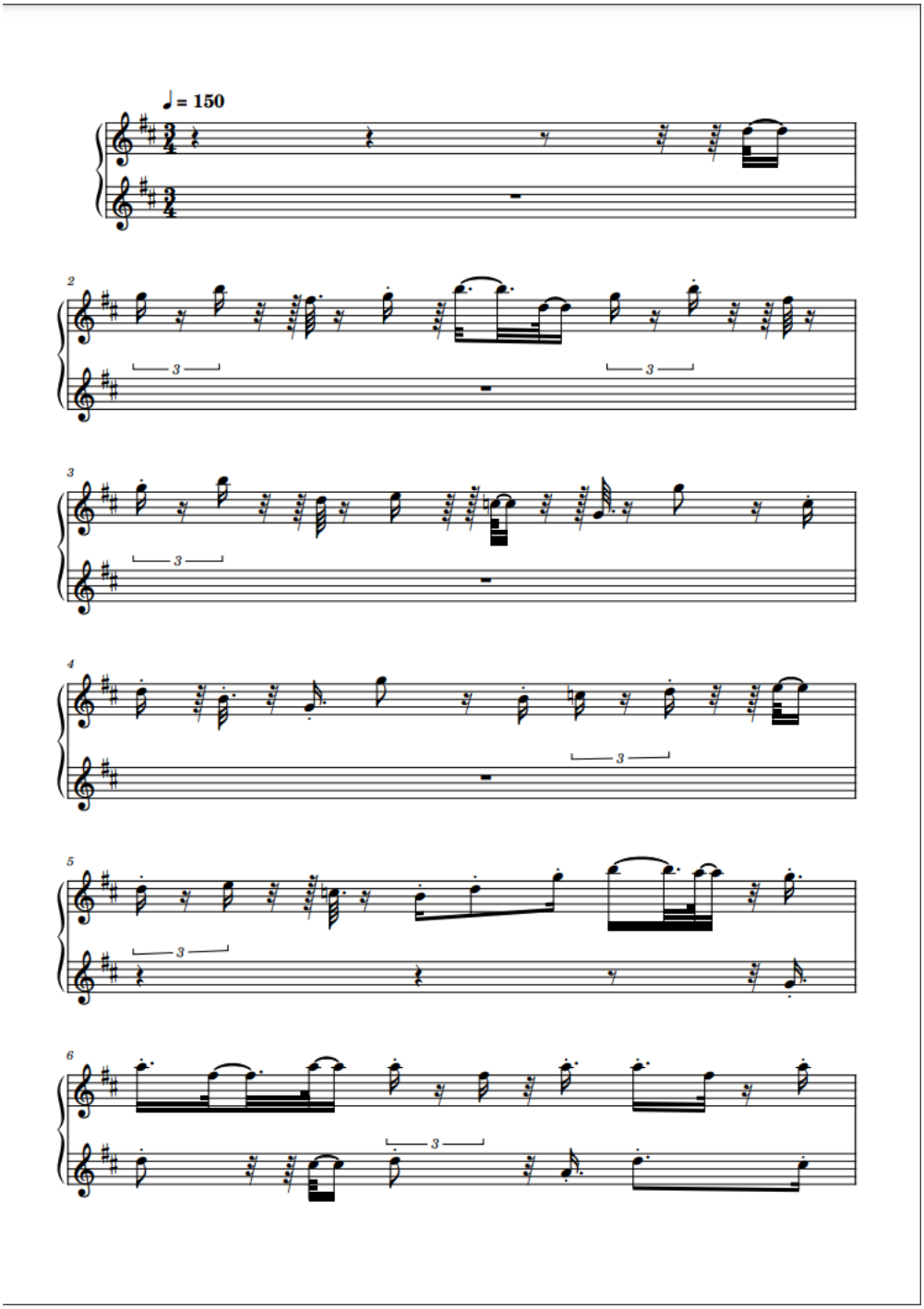
A Figura 11 exibe a tela de gerenciamento de partituras geradas. Da mesma forma como a tela inicial esse e todas as outras possui o menu de navegação, além disso lista todas a partituras geradas pelo sistema. Os objetos da lista servem como links que abrem o arquivo da partitura, ilustrada na figura 12.

FIGURA 11 - TELA GERENCIAR PARTITURAS



FONTE: Autores (2024)

FIGURA 12 – PARTITURA GERADA



FONTE: Autores (2024)

4.3.3 Tela Gerenciar Áudios

A tela de gerenciamento de áudios (Figura 13) possui diversas funcionalidades. O menu de navegação já mencionado e agora um menu de ações fixo disposto na lateral direita da interface. Essa tela é responsável em permitir o usuário gerar e

manipular arquivos MP3 e WAV. Nela é possível importar, criar e deletar arquivos musicais. A funcionalidade de separação e mescla de faixas musicais fica contido aqui também. Todos os áudios podem ser reproduzidos aqui selecionado os ‘cards’ onde estão localizados os arquivos de música.

Nesta tela, o usuário pode importar, criar e deletar arquivos musicais. A funcionalidade de separação e mescla de faixas musicais também está disponível aqui. Todos os áudios podem ser reproduzidos ao selecionar os cards onde os arquivos de música estão localizados.

Os principais botões e controles disponíveis são:

- Selecionar Arquivo: Ao clicar neste botão, uma janela é aberta para a seleção de arquivos que o usuário deseja importar para o sistema.
- Botão Play/Pause: Permite reproduzir ou pausar as músicas selecionadas. Quando uma música está em execução, o ícone do botão muda de "Play" para "Pause".
- Barra de Volume: Possibilita ajustar o volume da reprodução das músicas, aumentando ou diminuindo conforme a preferência do usuário.
- Menu de Manipulação de Músicas: Composto por seis botões que oferecem funcionalidades específicas:
    - Separar: Separa as faixas musicais de cada arquivo selecionado em quatro outros arquivos individuais: vocal, baixo, percussão e outros instrumentos.

      Ao acionar o botão Separar, é exibido um pop-up de confirmação, onde o usuário pode confirmar ou cancelar a operação. Nesse momento, é possível desmarcar, por meio de checkboxes, os arquivos que não se deseja separar (Figura 14). Após a confirmação, outro pop-up indica o progresso do processo e as etapas de separação, oferecendo também a opção de cancelar o processo através do botão Cancelar (Figura 15). Por fim, um pop-up informa a conclusão do processo de separação (Figura 16).
    - Mesclar: Mescla as faixas musicais dos arquivos selecionados em um único arquivo.

      Ao clicar em Mesclar, um pop-up de confirmação é exibido (Figura 17). Após a confirmação, um segundo pop-up mostra o progresso da

mesclagem (Figura 18). Ao término do processo, um pop-up indica a conclusão da operação (Figura 19).

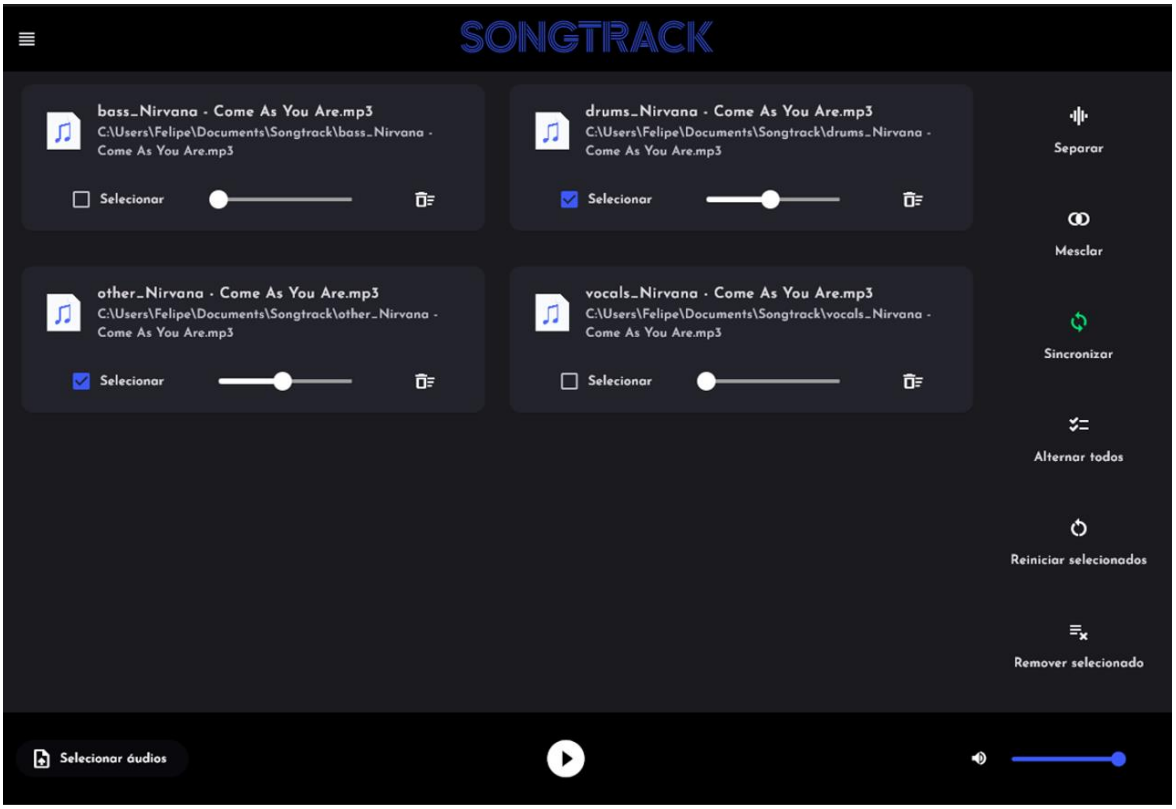
- Sincronizar: Atualiza e sincroniza o tempo dos arquivos musicais selecionados, alinhando-os para reprodução simultânea ou combinada.
- Alternar Todos: Seleciona ou desmarca todos os cards, marcando todos os arquivos como alvo para as operações disponíveis.
- Reiniciar Selecionados: Reinicia a reprodução dos arquivos selecionados, retornando ao início das músicas.
- Remover Selecionado: Remove do sistema os arquivos selecionados.

  Ao acionar o botão Remover Selecionado, um pop-up de confirmação é exibido, solicitando que o usuário confirme ou cancele a remoção dos arquivos selecionados (Figura 20).

Cada card de arquivo musical na tela possui os seguintes elementos e controles:

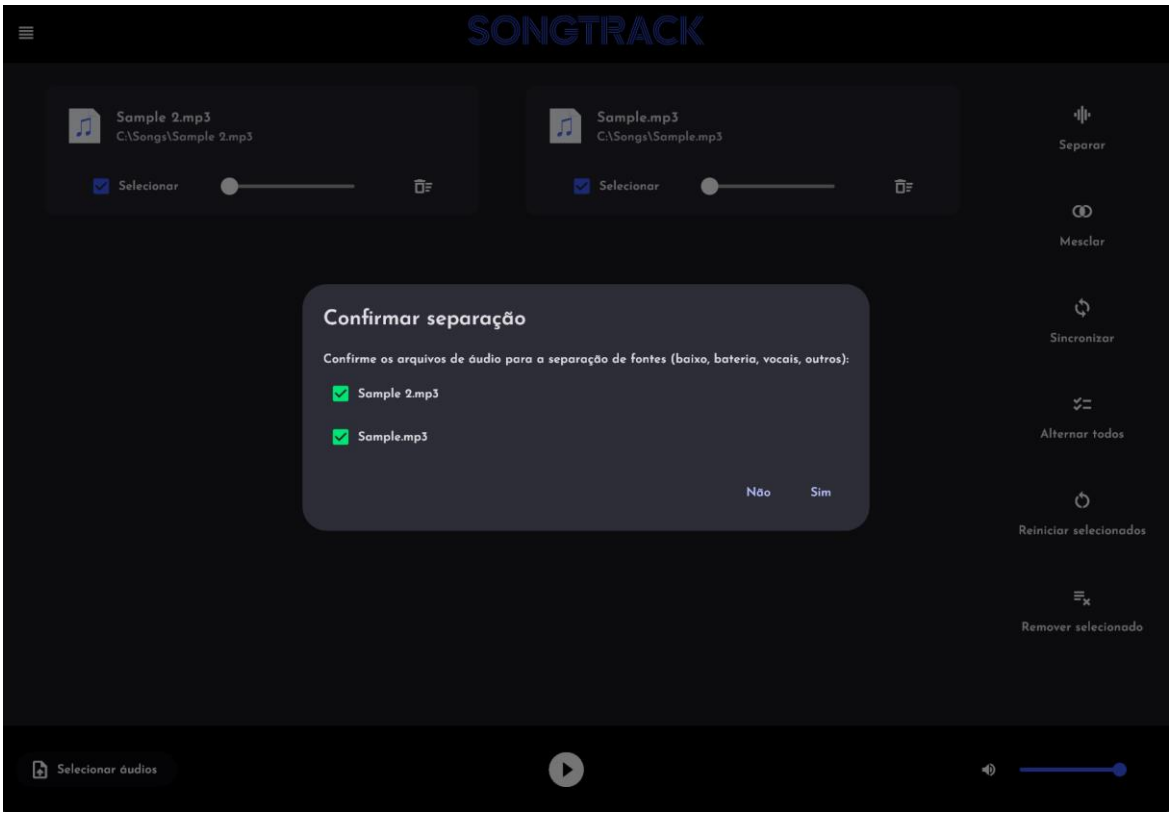
- Ícone de Lixo: Ao clicar neste ícone, o arquivo correspondente é removido da lista.
- Barra de Progresso: Exibe o progresso da reprodução da música e permite alterar o tempo de execução, possibilitando que o usuário avance ou retroceda para um ponto específico da faixa.
- Checkbox de Seleção: Indica se o card está selecionado. Quando marcado, o arquivo musical torna-se alvo para todas as operações disponíveis no menu de manipulação. Se não estiver selecionado, o arquivo é ignorado nos comandos que requerem seleção.

FIGURA 12 - TELA GERENCIAR ÁUDIOS



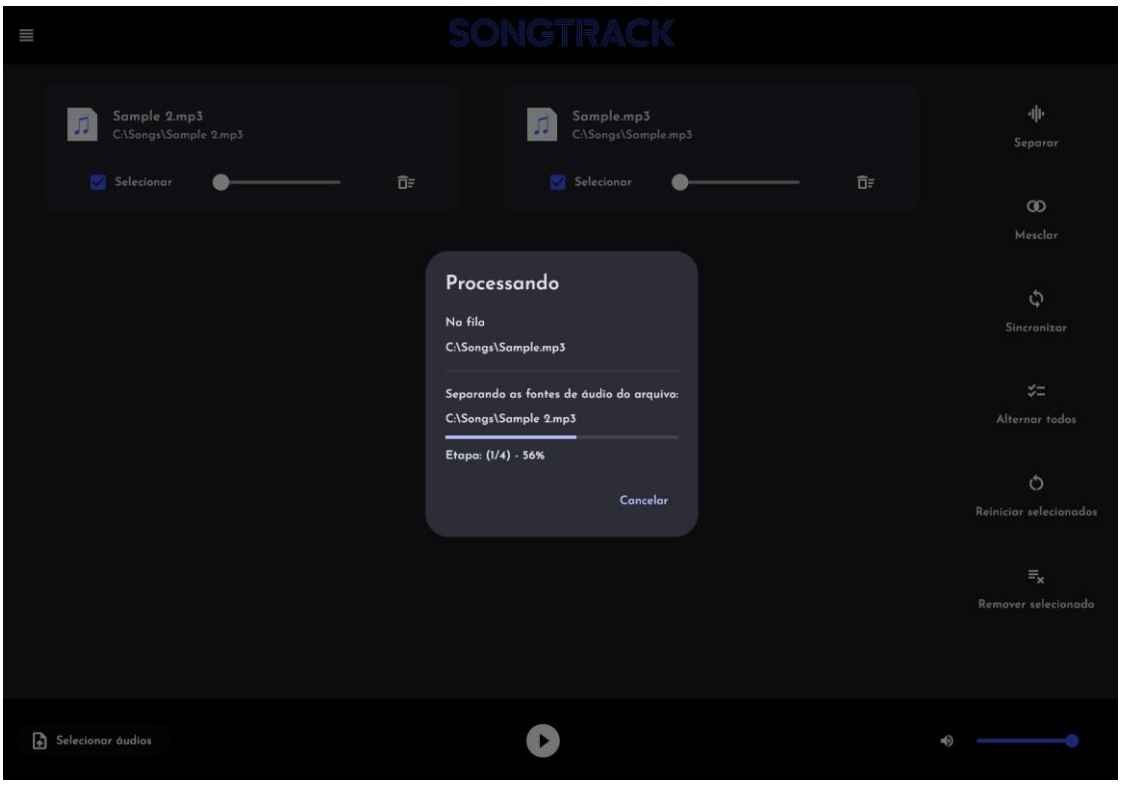
FONTE: Autores (2024)

FIGURA 13: CONFIRMAR SEPARAÇÃO



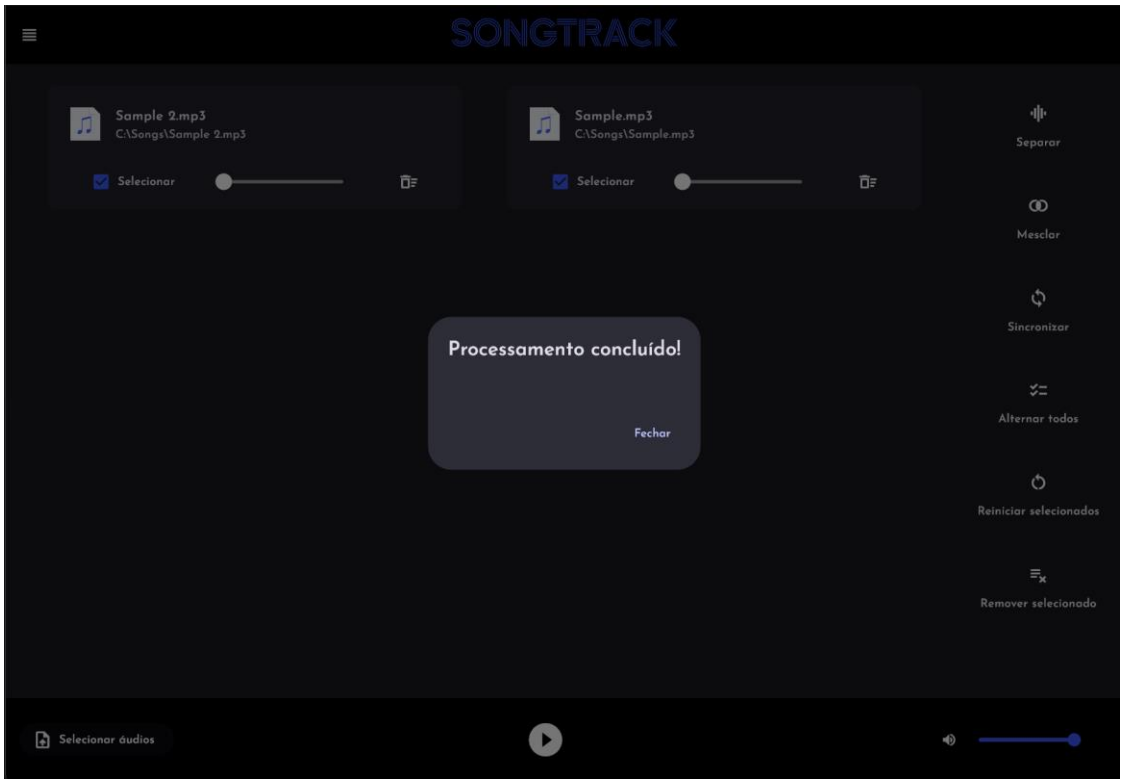
FONTE: Autores (2024)

FIGURA 14: PROCESSANDO SEPARAÇÃO



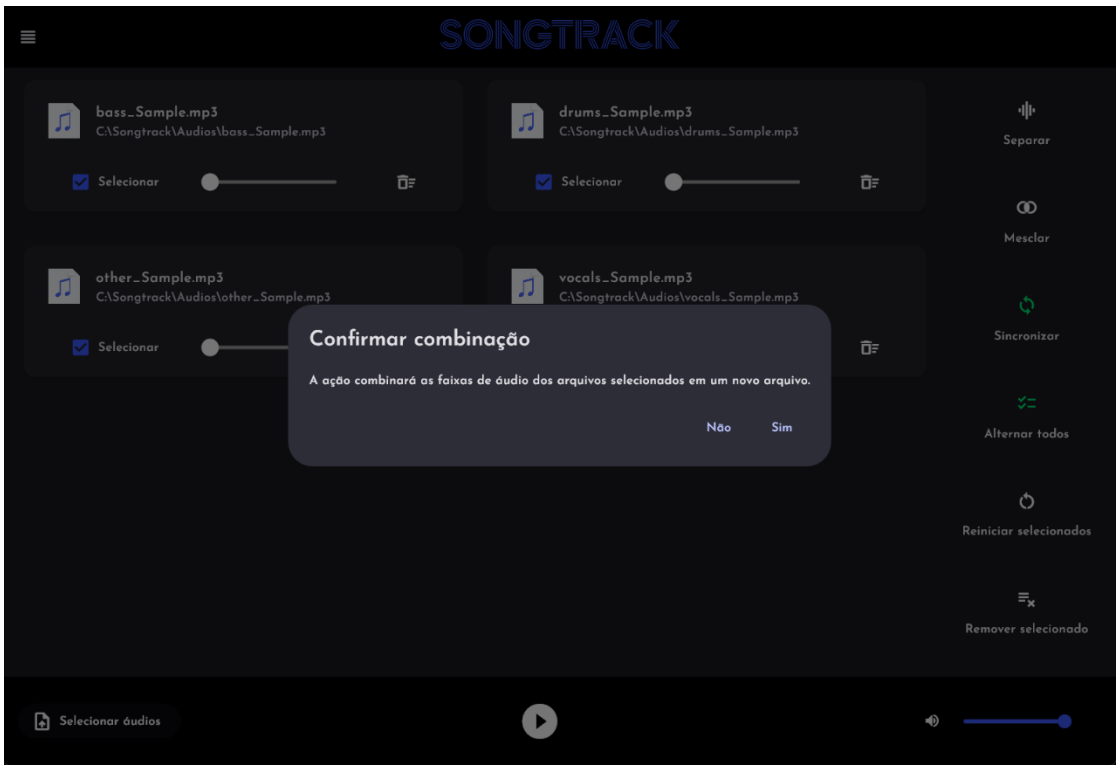
FONTE: Autores (2024)

FIGURA 15: CONCLUSÃO DA SEPARAÇÃO



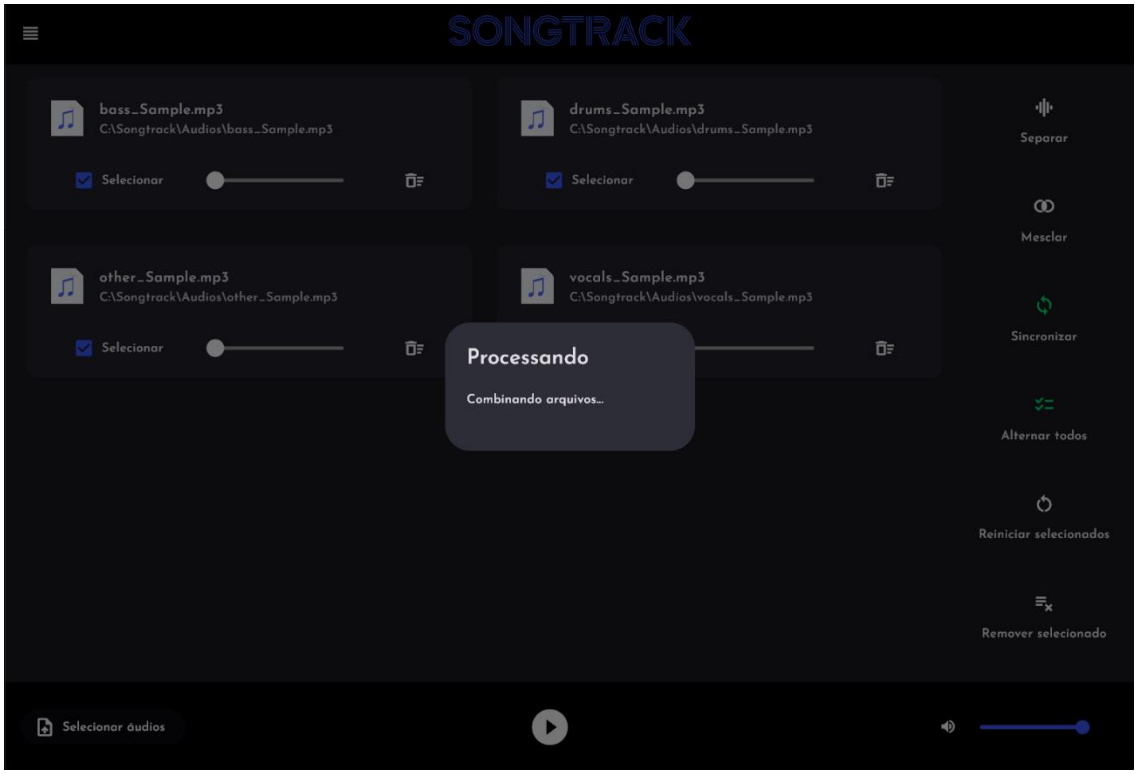
FONTE: Autores (2024)

FIGURA 16: CONFIRMAÇÃO DA COMBINAÇÃO



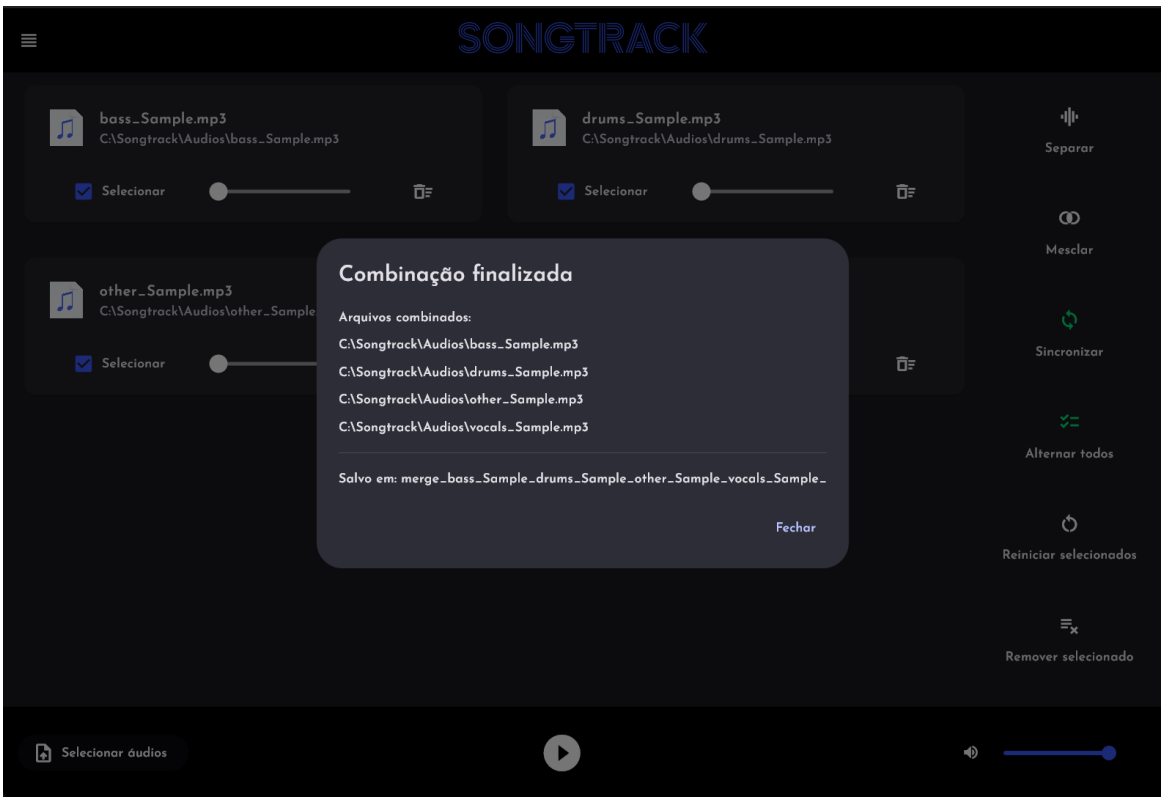
FONTE: Autores (2024)

FIGURA 17: PROCESSANDO COMBINAÇÃO



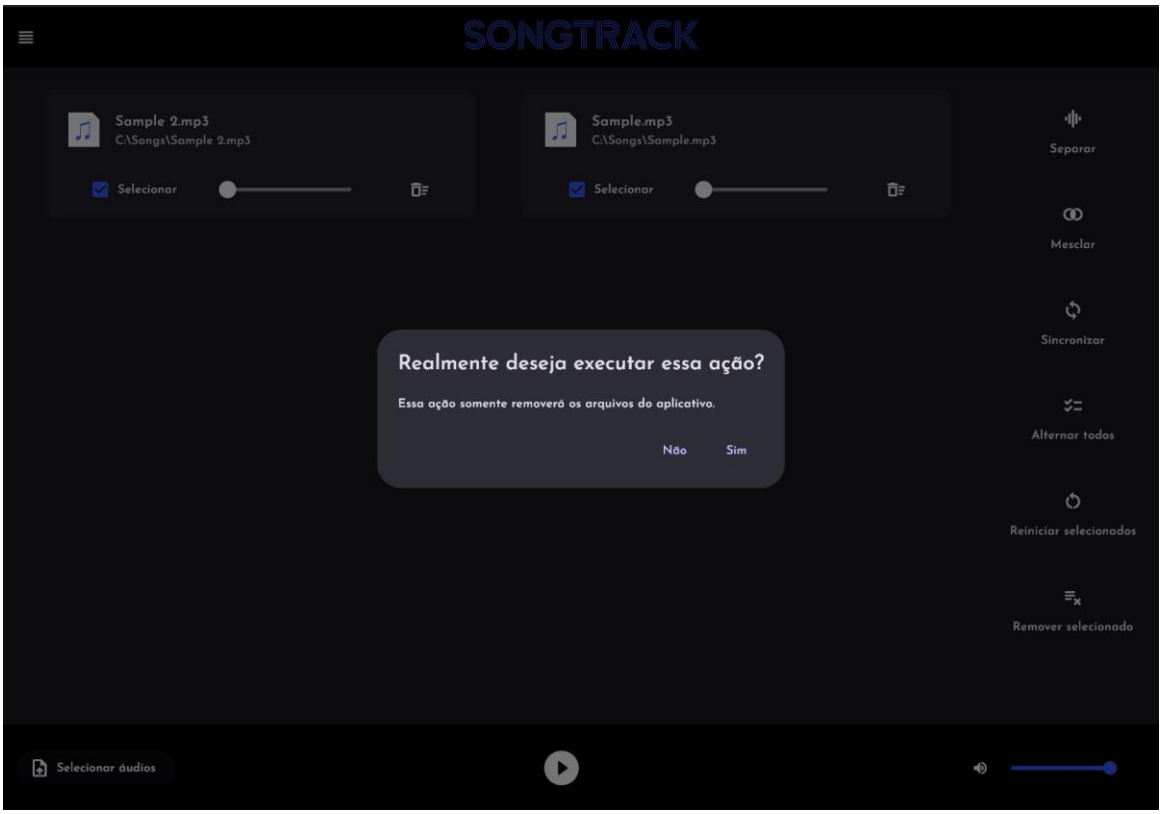
FONTE: Autores (2024)

FIGURA 18: CONCLUSÃO DA COMBINAÇÃO



FONTE: Autores (2024)

FIGURA 19: CONFIRMAÇÃO REMOÇÃO DOS ARQUIVOS SELECIONADOS



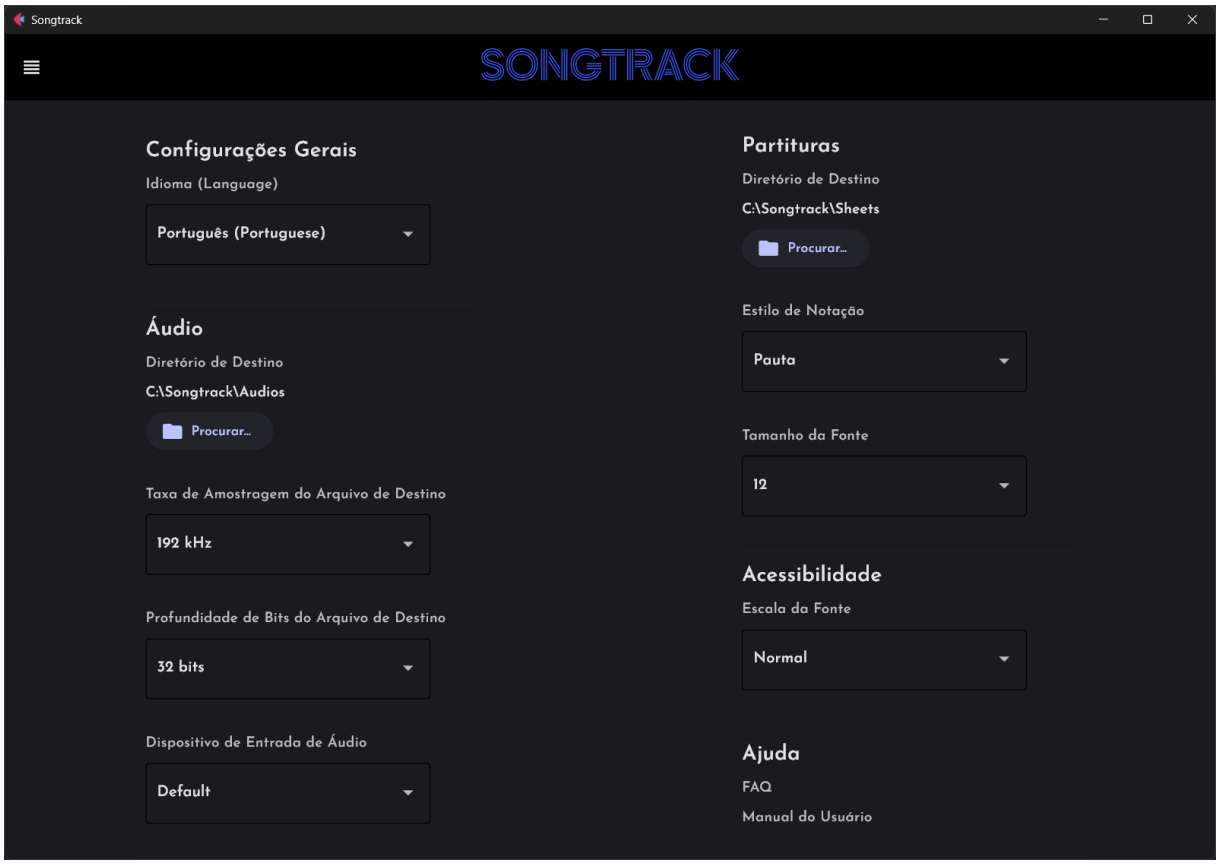
FONTE: Autores (2024)

4.3.4 Tela de Configurar Sistema

Configurações gerais do sistema são exibidas na Figura 21. Essa tela disponibiliza aos usuários as funções de configuração do software. Conforme exposto na figura abaixa as opções existentes são:

- Troca de idioma;
- Troca do diretório de onde os arquivos de áudio serão salvos;
- Taxa de amostragem;
- Profundidade de bits;
- Troca do dispositivo de entrada de áudio;
- Troca do diretório de onde os arquivos de partituras serão salvos;
- Escolha do tipo de Notação utilizada para transcrição;
- Tamanho da fonte;
- Escala da fonte.

FIGURA 20 - TELA DE CONFIGURAR SISTEMA



FONTE: Autores (2024)

5 CONSIDERAÇÕES FINAIS

A transcrição musical automática e a separação de faixas musicais são áreas de crescente interesse e inovação dentro do campo da musicologia computacional e processamento de áudio. Com o avanço das tecnologias de aprendizado de máquina e inteligência artificial, automatizar tarefas que antes eram realizadas manualmente, como a transcrição de gravações de áudio em notação musical e a separação de diferentes componentes de uma música, tonaram-se cada vez mais viáveis.

Nesse contexto, o projeto proposto visou desenvolver um sistema eficiente para a transcrição de partituras e a separação de faixas musicais utilizando ferramentas de processamento de sinais e aprendizado de máquina. O projeto buscou implementar modelos de Redes Neurais Convulacionais com o uso de bibliotecas como PyTorch e Librosa, utilizando o *dataset* MAESTRO como fonte de dados.

Para o desenvolvimento do sistema, foi adotada a biblioteca Demucs para a separação das faixas musicais, enquanto a notação musical foi realizada utilizando o software MuseScore. Essa abordagem permitiu o progresso contínuo do sistema, assegurando a eficácia na transcrição musical e na manipulação de arquivos de áudio.

5.1 TRABALHOS FUTUROS

Embora o sistema desenvolvido tenha alcançado os objetivos propostos dentro das limitações existentes, existe potencial para aprimoramentos em várias áreas. Um dos aperfeiçoamentos seria em otimizar o processo de treinamento para a detecção de notas musicais, com o objetivo de integrar modelos próprios ao software desenvolvido, substituindo o uso de bibliotecas desenvolvida por terceiros. A implementação necessária abrange a configuração e otimização do processamento paralelo do algoritmo, que passará a utilizar placas de vídeo que suportem essas tecnologias.

Outra possível melhoria envolve o desenvolvimento de uma solução interna para a geração e visualização de partituras em formato PDF, eliminando a dependência do software MuseScore. A implementação de um módulo dedicado à renderização de notação musical proporcionará maior flexibilidade e independência ao sistema. Outras áreas de interesse incluem a otimização do desempenho computacional, possibilitando a execução do sistema em ambientes com recursos limitados, e a

expansão das capacidades de exportação para incluir novos formatos de arquivo. Também há possibilidade de integrar o sistema com outros aplicativos e bibliotecas, ampliando seu alcance e funcionalidade no contexto da transcrição e manipulação de música digital.

REFERÊNCIAS

ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business.** Blue Hole Press, 2010.

AUDACITY TEAM. **Audacity Manual**. 2023. Disponível em: https://manual.audacityteam.org/. Acesso em: 07 out. 2024.

AVID Technology. **Experience the power of Sibelius music software**. Disponível em: https://www.avid.com/sibelius. Acesso em: 12 out. 2024.

BAYSAL, B.; EFE, M. Ö. *A comparative study of blind source separation methods.* **Turkish Journal of Electrical Engineering & Computer Sciences,** [S.l.], v. 31, n. 1, p. 1-20, 2023. Disponível em: http://journals.tubitak.gov.tr/elektrik/. Acesso em: 11 abr. 2023.

BENETOS, E.; DIXON, S.; DUAN, Z.; EWERT, S. **Automatic Music Transcription: An Overview**. *IEEE Signal Processing Magazine*, v. 36, n. 1, p. 20-30, jan. 2019. DOI: 10.1109/MSP.2018.2869928.

BENWARD, Bruce; SAKER, Marilyn. **Music in theory and practice**. 8. ed. Boston: McGraw-Hill, 2009.

CALIFORNIA STATE UNIVERSITY, Chico. **Audacity in the classroom**. Chico State Faculty Development. Disponível em: https://www.csuchico.edu/fdev/fdev-teaching-guides/teachingguide-2.shtml. Acesso em: 10 out. 2024.

CASEY, M. A.; VELTKAMP, R.; GOTO, M.; LEMAN, M.; RHODES, C.; SLANEY, M. **Content-Based Music Information Retrieval: Current Directions and Future Challenges.** *Proceedings of the IEEE*, v. 96, n. 4, p. 668-696, 2008.

CHAN, A.; JANSSON, A.; HUMPHREY, E.; RAFAEL, M. **Vocal separation using recurrent neural networks.** International Society for Music Information Retrieval Conference (ISMIR). 2015.

COOK, N. **Music: A Very Short Introduction**. Oxford: Oxford University Press, 1998. 1ª ed. (A Very Short Introduction).

DÉFOSSEZ, A. **Hybrid Spectrogram and Waveform Source Separation**. In: *ISMIR 2021 MDX Workshop*. [S.l.]: arXiv, 2022. Disponível em: https://doi.org/10.48550/arXiv.2111.03600. Acesso em: Acesso em: 20 nov. 2023.

DÉFOSSEZ, A.; USUNIER, N.; BOTTOU, L.; BACH, F. **Demucs: Deep Extractor for Music Sources with Extra Unlabeled Data Remixed**. *arXiv preprint arXiv:1909.01174*, 2019. Disponível em: https://doi.org/10.48550/arXiv.1909.01174. Acesso em: 11 abr. 2023.

DÉFOSSEZ, A.; USUNIER, N.; BOTTOU, L.; BACH, F. **Music Source Separation in the Waveform Domain**. *arXiv preprint arXiv:1911.13254*, 2021. Disponível em: https://doi.org/10.48550/arXiv.1911.13254. Acesso em: 11 abr. 2023.

FLET TEAM. **Flet Documentation**. 2023. Disponível em: https://flet.dev. Acesso em: 18 ago. 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning.** MIT press, 2016. 800 p. Disponível em: https://www.deeplearningbook.org/contents/intro.html. Acesso em: 11 abr. 2023.

GROSSMAN, A.; GROSSMAN, J. **Automatic music transcription: generating MIDI from audio**. *Reports of CS230 Deep Learning, Lecture at Stanford University*, p. 1-6, 2020.

HENNEQUIN, R.; KOPPENBERGER, M.; MAMEI, A. **Spleeter: a fast and efficient music source separation tool with pre-trained models**. *Journal of Open Source Software*, v. 5, n. 50, p. 2154, 2020. DOI: https://doi.org/10.21105/joss.02154. Acesso em: 11 abr. 2023.

HUBER, D. M.; RUNSTEIN, R. E. **Modern Recording Techniques**. 8. ed. New York: Routledge, 2013.

JANSSON, A.; HUMPHREY, E.; MONTECCHIO, N.; BITTNER, R.; KUMAR, A.; WEYDE, T. **Singing voice separation with deep U-Net convolutional networks.** 18th International Society for Music Information Retrieval Conference (ISMIR). 2017.

JEON, C. B.; LEE, K. **A Hybrid Approach for Music Source Separation: Combining Deep Neural Networks with Conventional Signal Processing Techniques**. *Journal of Electrical Engineering & Technology*, 2022.

JIA, S. **PyDub Documentation**. 2015. Disponível em: https://github.com/jiaaro/pydub. Acesso em: 18 ago. 2024.

KERR, D. A. MIDI—**The Musical Instrument Digital Interface.** 2009. Disponível em: http://dougkerr.net/Pumpkin/articles/MIDI.pdf. Acesso em: 12 jun. 2023.

KNIBERG, H.; SKARIN, M. **Kanban and Scrum - Making the Most of Both.** C4Media, 2010. 122 p.

KOSTKA, Stefan; PAYNE, Dorothy. **Tonal harmony**. 6. ed. New York: McGraw-Hill, 2009.

LADAS, C. **Scrumban: Essays on Kanban Systems for Lean Software Development**. [S.l.]: Lulu.com, 2009.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. *Nature*, v. 521, p. 436-444, 2015. Disponível em: https://doi.org/10.1038/nature14539. Acesso em: 18 ago. 2024.

LI, P.; LEE, M.; WANG, Y.; MA, T.; LIU, J. **Deep learning-based music source separation: A comprehensive review**. *IEEE Transactions on Neural Networks and Learning Systems*, v. 32, n. 12, p. 5555-5576, 2021.

LI, Z.; YANG, W.; PENG, S.; LIU, F. **A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects**. *arXiv preprint arXiv:2004.02806*, 2020. Disponível em: https://doi.org/10.48550/arXiv.2004.02806. Acesso em: 18 ago. 2024.

LIU, H.; KONG, Q.; LIU, J. **CWS-PResUNet: Music Source Separation with Channel-wise Subband Phase-aware ResUNet**. In: *MDX Workshop @ ISMIR 2021*. [S.l.]: arXiv, 2021. Disponível em: https://doi.org/10.48550/arXiv.2112.04685. Acesso em: 20 nov. 2023.

LUTZ, M. **Python Pocket Reference**. 5. ed. Sebastopol: O'Reilly Media, 2014.

LYONS, Richard G. **Understanding digital signal processing**. 3. ed. Upper Saddle River: Prentice Hall, 2011.

MANILOW, E.; SEETHARAMAN, P.; PISHDARE, S.; PARIKH, A.; REYNOLDS, T. **Singing Voice Separation Using Deep Learning: Insights from the MUSDB18-HQ Dataset**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

MCFEE, B.; RAFFEL, C.; LIANG, D.; ELLIS, D. P. W.; MCVICAR, M.; BATLE, M.; NIETO, O. **librosa: Audio and music signal analysis in Python**. In: *Proceedings of the 14th Python in Science Conference (SciPy 2015)*. Austin: [s.n.], 2015. p. 18-25. Disponível em: https://doi.org/10.25080/Majora-7b98e3ed-003. Acesso em: 18 ago. 2024.

MELLO, Marcelo. ***Apostila de Teoria Musical***. 2015. Disponível em: <http://marcelomelloweb.net/mmteoria_apostila.htm>. Acesso em: 20 nov. 2023.

MITSUFUJI, Y.; SAITO, S.; TAKAHASHI, N.; SAITO, Y.; KAMADA, M.; KAWAHARA, H. **Music Demixing Challenge 2022: Enhancing Audio Source Separation Techniques with MUSDB18-HQ**. *IEEE Transactions on Audio, Speech, and Language Processing*, 2022.

MUSIC Notation Project. **Sibelius music notation software**. Disponível em: https://musicnotation.org/software/sibelius. Acesso em: 12 jul. 2024.

O'SHEA, K.; NASH, R. **An Introduction to Convolutional Neural Networks.** Aberystwyth: Aberystwyth University, Lancaster: Lancaster University, 2015.

PASZKE, A. et al. PyTorch: **An Imperative Style, High-Performance Deep Learning Library.** In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, 2019.

PISTON, Walter. **Harmony**. 5. ed. New York: W.W. Norton & Company, 1978.

RISSOUD, M.; HANSON, J.-N.; GAUVRIT, F.; RENARD, C.; LEMESRE, P.-E.; BONNE, N.-X.; VINCENT, C. **Sound source localization. European Annals of Otorhinolaryngology, Head and Neck Diseases,** Lille, v. 135, n. 4, p. 259-264, Aug. 2018. Available online: 3 May 2018. Version of Record: 20 Aug. 2018.

ROSSING, Thomas D. **The science of sound**. 3. ed. San Francisco: Addison-Wesley, 2007.

SCHAFFER, N.; COGAN, B.; MANILOW, E.; MORRISON, M.; SEETHARAMAN, P.; PARDO, B. **Music separation enhancement with generative modeling**. *arXiv preprint arXiv:2208.12387*, 2022. Disponível em: https://arxiv.org/abs/2208.12387. Acesso em: 11 abr. 2023.

STÖTER, F. R.; LIUTKUS, A.; ITO, N. **Open-Unmix - A reference implementation for music source separation.** *Journal of Open Source Software*, v. 4, n. 41, p. 1667, 2019. DOI: https://doi.org/10.21105/joss.01667. Acesso em: 18 ago. 2024.

UNIVERSITY OF TENNESSEE. **Audacity**. Office of Information Technology. Disponível em: https://oit.utk.edu/instructional/toolkit/teaching-tools-course-content/audacity/. Acesso em: 11 jul. 2024.

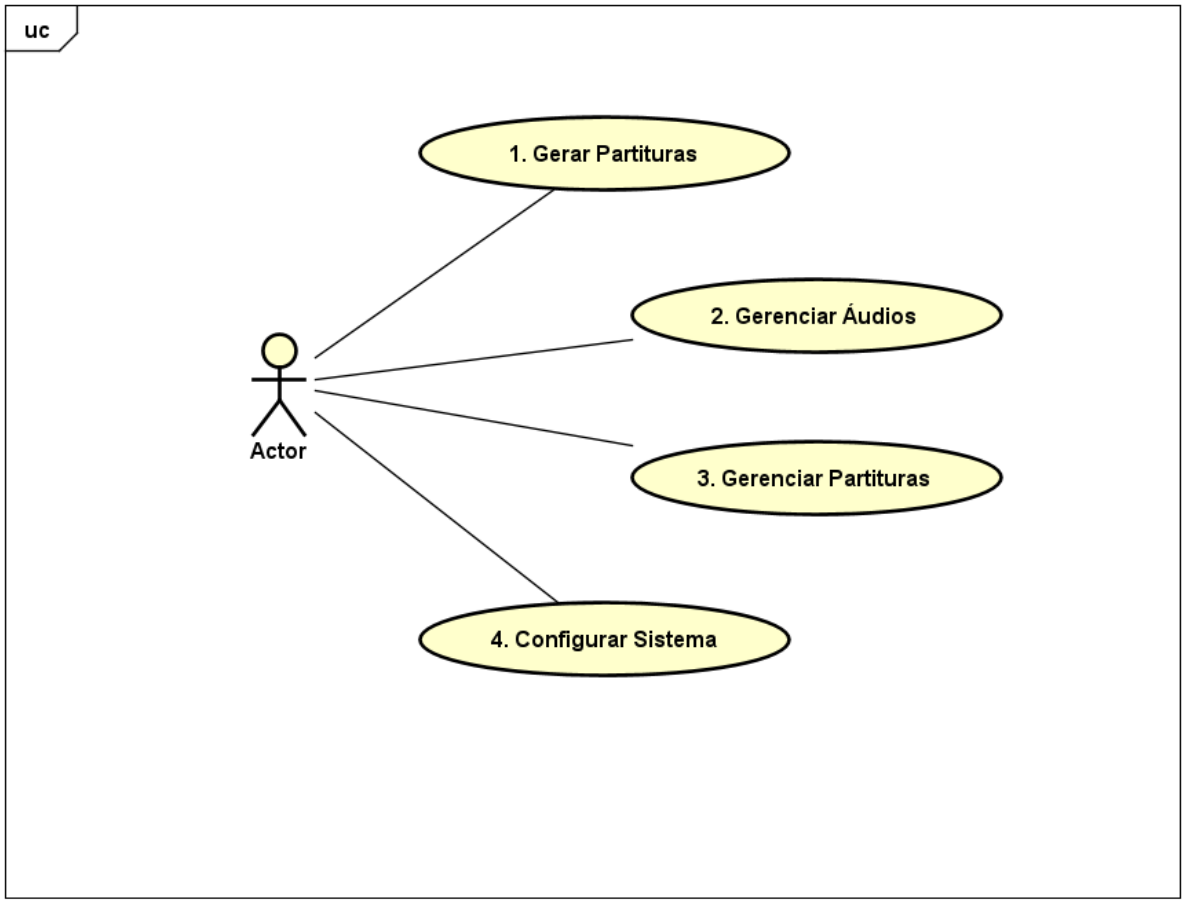
VANDERPLAS, J. **Python Data Science Handbook: Essential Tools for Working with Data**. Sebastopol: O'Reilly Media, 2016.

VINCENT, E.; FÉVOTTE, C.; GRIBONVAL, R. **Performance measurement in blind audio source separation**. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, n. 4, p. 1462-1469, 2006.

WERNER, T.; BONTE, N.; NICHIFOR, V.; SCHREER, J.; SABATINI, A.; PANNETIER, J. **MuseScore: Free music composition & notation software**. 2023. Disponível em: https://musescore.org. Acesso em: 18 ago. 2024.

APÊNDICE A – DIGRAMA DE CASO DE USO

FIGURA 21 - DIAGRAMA DE CASO DE USO

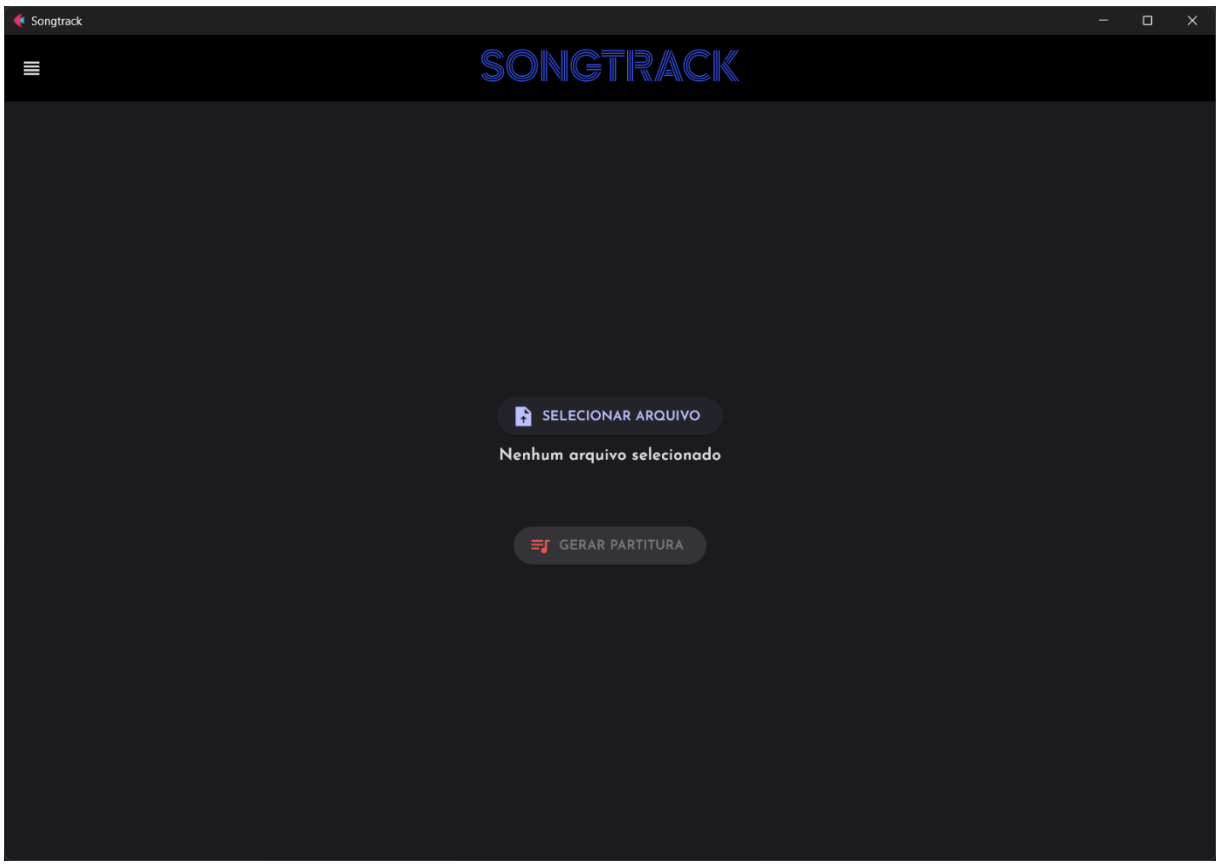


FONTE: Autores (2024)

APÊNDICE B – HISTÓRIAS DE USUÁRIO

HU 001 – Gerar Partitura

FIGURA 22 - TELA - GERAR PARTITURA



FONTE: Os autores (2024)

SENDO Usuário do sistema

QUERO fazer o upload de um arquivo de música no formato mp3

PARA gerar uma partitura a partir do arquivo enviado

CRITÉRIOS DE ACEITAÇÃO

1 - Seleção de arquivo mp3

- DADO QUE o usuário acessa a página "Gerar Partitura"

- QUANDO clica no botão "Selecionar Arquivo"

- ENTÃO deve abrir uma janela para selecionar um arquivo de música no formato mp3

2 - Confirmação de arquivo selecionado

- DADO QUE o usuário selecionou um arquivo
- QUANDO o arquivo é escolhido
- ENTÃO o nome do arquivo deve ser exibido na tela indicando que foi selecionado corretamente

3 - Geração de partitura

- DADO QUE o usuário selecionou um arquivo mp3
- QUANDO clica no botão "Gerar Partitura"
- ENTÃO o sistema deve processar o arquivo e gerar uma partitura correspondente

4 - Exibição de erro ao não selecionar arquivo

- DADO QUE o usuário acessa a página "Gerar Partitura"
- QUANDO clica no botão " Gerar Partitura " sem ter selecionado um arquivo
- ENTÃO deve ser exibida uma mensagem de erro indicando que é necessário selecionar um arquivo mp3

5 - Navegação pelo menu lateral

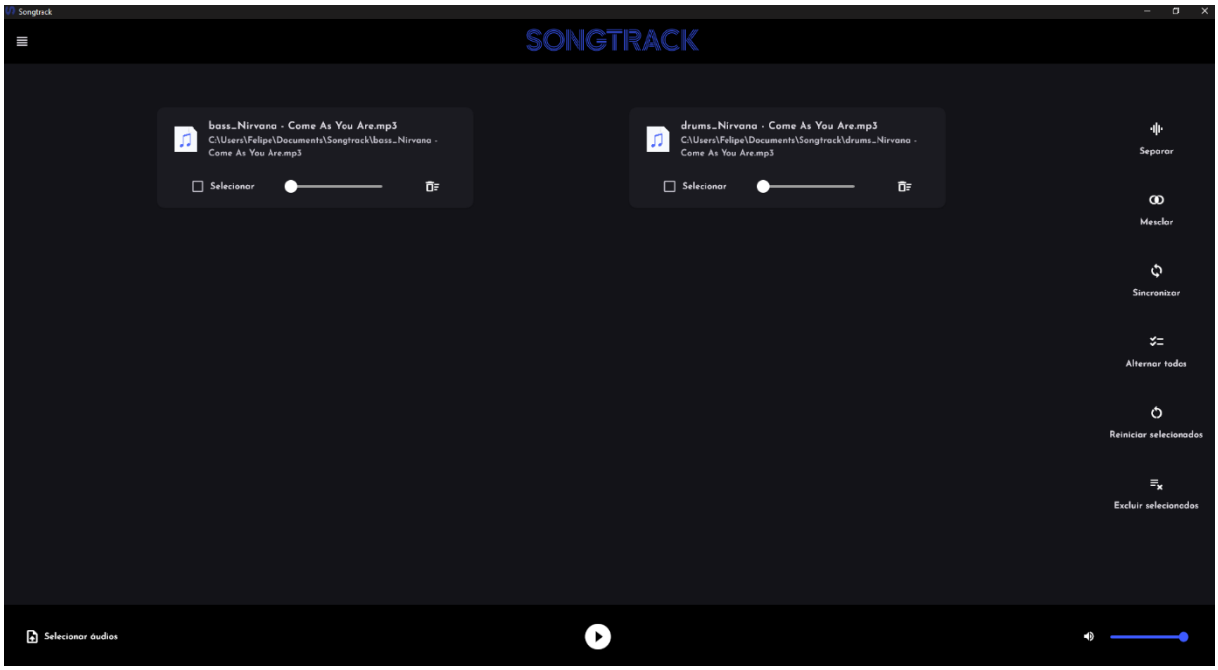
- DADO QUE o usuário acessa qualquer página do sistema
- QUANDO clica no ícone do menu lateral
- ENTÃO o menu deve ser expandido, exibindo as opções de navegação: "Gerar Partitura", "Áudios Salvos", "Partituras Salvas", "Configurações"

REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - Apenas arquivos no formato mp3 devem ser aceitos para upload.
R2 - O sistema deve verificar se o arquivo é um arquivo de música válido antes de tentar gerar a partitura.

HU 002 – Gerenciar Áudios

FIGURA 23 - TELA - GERENCIAR ÁUDIOS



FONTE: Os autores (2024)

SENDO Usuário do sistema

QUERO visualizar e manipular os áudios salvos

PARA gerenciar os arquivos de música utilizados para gerar partituras

CRITÉRIOS DE ACEITAÇÃO

1 - Seleção de arquivos MP3

- DADO QUE o usuário acessa a página "Áudios Salvos"

- QUANDO clica no botão "Selecionar áudios" no canto inferior esquerdo

- ENTÃO deve abrir uma janela para selecionar vários arquivos de música nos formatos MP3 ou WAV

2 - Exibição de arquivos selecionados

- DADO QUE o usuário selecionou arquivos MP3

- QUANDO os arquivos são escolhidos

- ENTÃO os arquivos devem ser exibidos na tela dentro de cards, mostrando o nome do arquivo e permitindo monitorar e manipular o progresso

3 - Manipulação de arquivos

- DADO QUE o usuário acessa a página "Áudios Salvos"
- QUANDO seleciona um ou mais arquivos
- ENTÃO deve ser possível deletar, selecionar/deselecionar, e manipular o progresso dos arquivos

4 - Separação de faixas

- DADO QUE o usuário seleciona um ou mais arquivos
- QUANDO clica no botão "Separar"
- ENTÃO o sistema deve separar a música em diferentes faixas (Bass, Vocal, Drums e Outros)

5 - Mesclagem de faixas

- DADO QUE o usuário seleciona um ou mais arquivos
- QUANDO clica no botão "Mesclar"
- ENTÃO o sistema deve mesclar as faixas separadas em um único arquivo

6 - Sincronização de faixas

- DADO QUE o usuário seleciona um ou mais arquivos
- QUANDO clica no botão "Sincronizar"
- ENTÃO o sistema deve sincronizar as faixas musicais selecionadas

7 - Alternar seleção de arquivos

- DADO QUE o usuário acessa a página "Áudios Salvos"
- QUANDO clica no botão "Alternar todos"

- ENTÃO todos os arquivos devem ser selecionados ou deselecionados de uma vez só

8 - Reiniciar arquivos selecionados

- DADO QUE o usuário seleciona um ou mais arquivos
- QUANDO clica no botão "Reiniciar selecionados"
- ENTÃO o sistema deve reiniciar o progresso dos arquivos selecionados

9 - Reproduzir áudios

- DADO QUE o usuário seleciona um ou mais arquivos
- QUANDO clica no botão "Play"
- ENTÃO todos os arquivos selecionados devem ser reproduzidos ao mesmo tempo

10 - Ajuste de volume

- DADO QUE o usuário acessa a página "Áudios Salvos"
- QUANDO utiliza o manipulador de volume
- ENTÃO o volume de reprodução dos arquivos deve ser ajustado

11 - Navegação pelo menu lateral

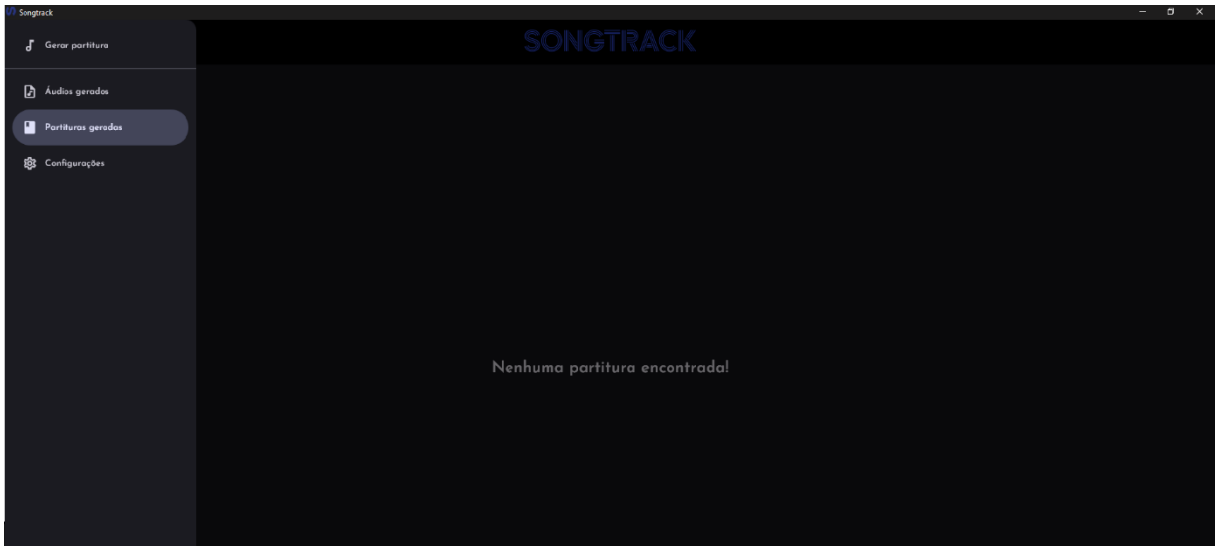
- DADO QUE o usuário acessa qualquer página do sistema
- QUANDO clica no ícone do menu lateral
- ENTÃO o menu deve ser expandido, exibindo as opções de navegação: "Gerar Partitura", "Áudios Gerados", "Partituras Geradas", "Configurações"

REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - Apenas arquivos nos formatos MP3 ou WAV devem ser aceitos para upload.
R2 - O sistema deve permitir manipular várias faixas musicais ao mesmo tempo.

HU 003 – Gerenciar Partitura

FIGURA 24 - TELA - GERENCIAR PARTITURA



FONTE: Os autores (2024)

SENDO Usuário do sistema
QUERO visualizar e selecionar as partituras geradas
PARA gerenciar as partituras processadas pelo sistema

CRITÉRIOS DE ACEITAÇÃO

1 - Exibição de partituras

- DADO QUE o usuário acessa a página "Partituras Geradas"
- QUANDO a página é carregada
- ENTÃO o sistema deve localizar a pasta definida para armazenar os arquivos de partitura, reconhecer os arquivos de partitura e exibi-los na tela

2 - Mensagem de ausência de partituras

- DADO QUE o usuário acessa a página "Partituras Geradas"
- QUANDO não há arquivos de partitura na pasta definida
- ENTÃO o sistema deve exibir a mensagem "Nenhuma partitura encontrada!"

3 - Navegação pelo menu lateral

- DADO QUE o usuário acessa qualquer página do sistema

- QUANDO clica no ícone do menu lateral

- ENTÃO o menu deve ser expandido, exibindo as opções de navegação: "Gerar Partitura", "Áudios Gerados", "Partituras Geradas", "Configurações"

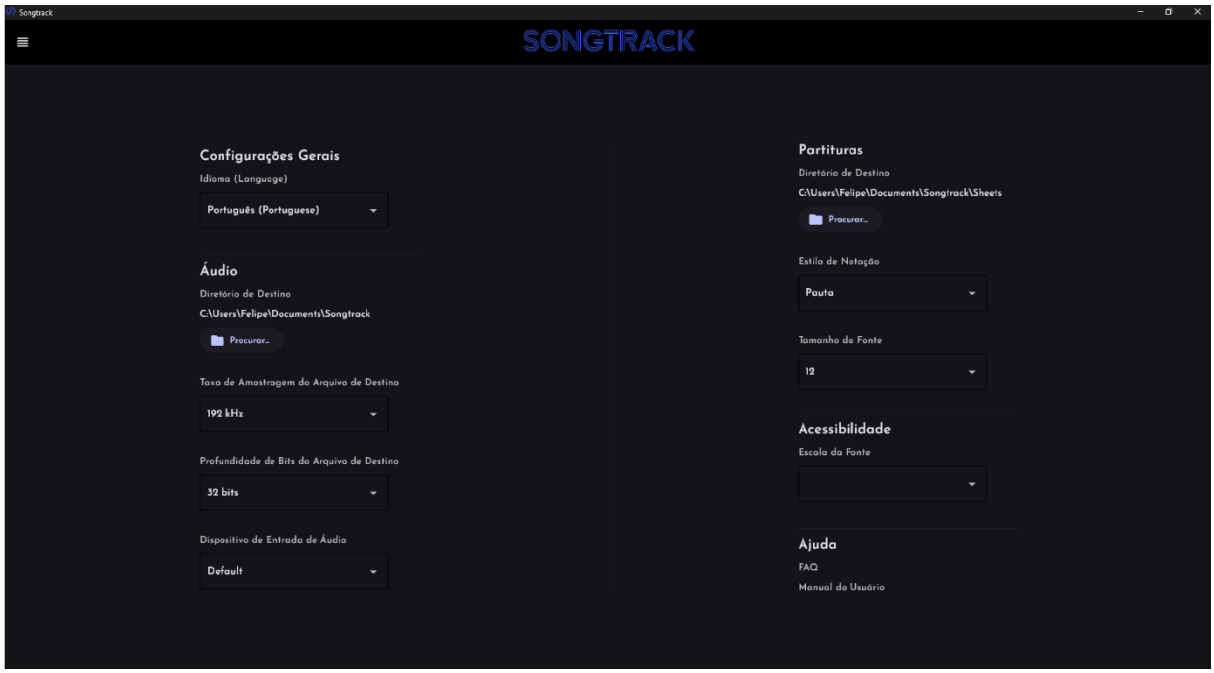
REGRAS DE NEGÓCIO DA HISTÓRIA:

R1 - Apenas arquivos no formato de partitura devem ser exibidos na lista (XML).

R2 - O sistema deve atualizar a lista de partituras sempre que um novo arquivo for adicionado à pasta.

HU 004 – Configurar Sistema

FIGURA 25 - TELA - CONFIGURAR SISTEMA



FONTE: Os autores (2024)

SENDO Usuário do sistema
QUERO manipular as configurações do sistema
PARA personalizar a minha experiência de uso

CRITÉRIOS DE ACEITAÇÃO

1 - Troca de linguagem do sistema

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona uma linguagem diferente
- ENTÃO o sistema deve mudar a interface para a linguagem selecionada - Linguagens disponíveis: inglês, português, japonês, espanhol, russo e chinês simplificado

2 - Alteração do diretório de áudio

- DADO QUE o usuário acessa a página "Configurações Gerais"

- QUANDO clica no botão "Procurar..." no campo "Diretório de Destino" de Áudio
- ENTÃO deve abrir uma janela para selecionar uma nova pasta de destino para os arquivos de áudio

3 - Alteração da taxa de amostragem

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona uma nova taxa de amostragem no campo "Taxa de Amostragem do Arquivo de Destino"
- ENTÃO o sistema deve atualizar a configuração para a taxa selecionada - Taxas disponíveis: 192 kHz, 96 kHz, 48 kHz, 44.1 kHz

4 - Alteração da profundidade de bits

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona uma nova profundidade de bits no campo "Profundidade de Bits do Arquivo de Destino"
- ENTÃO o sistema deve atualizar a configuração para a profundidade selecionada - Profundidades disponíveis: 32 bits, 24 bits

5 - Alteração do dispositivo de entrada de áudio

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona um novo dispositivo no campo "Dispositivo de Entrada de Áudio"
- ENTÃO o sistema deve atualizar a configuração para o dispositivo selecionado

6 - Alteração do diretório de partituras

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO clica no botão "Procurar..." no campo "Diretório de Destino" de Partituras
- ENTÃO deve abrir uma janela para selecionar uma nova pasta de destino para os arquivos de partituras

7 - Alteração do estilo de notação

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona um novo estilo de notação no campo "Estilo de Notação"
- ENTÃO o sistema deve atualizar a configuração para o estilo selecionado - Estilos disponíveis: Pauta, Tablatura

8 - Alteração do tamanho da fonte

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona um novo tamanho de fonte no campo "Tamanho da Fonte"
- ENTÃO o sistema deve atualizar a configuração para o tamanho selecionado

9 - Alteração da escala da fonte

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO seleciona uma nova escala no campo "Escala da Fonte"
- ENTÃO o sistema deve atualizar a configuração para a escala selecionada - Escalas disponíveis: Normal, Grande

10 - Acesso a FAQ e Manual do Usuário

- DADO QUE o usuário acessa a página "Configurações Gerais"
- QUANDO clica no link "FAQ" ou "Manual do Usuário"
- ENTÃO o sistema deve abrir a página correspondente com as informações

11 - Navegação pelo menu lateral

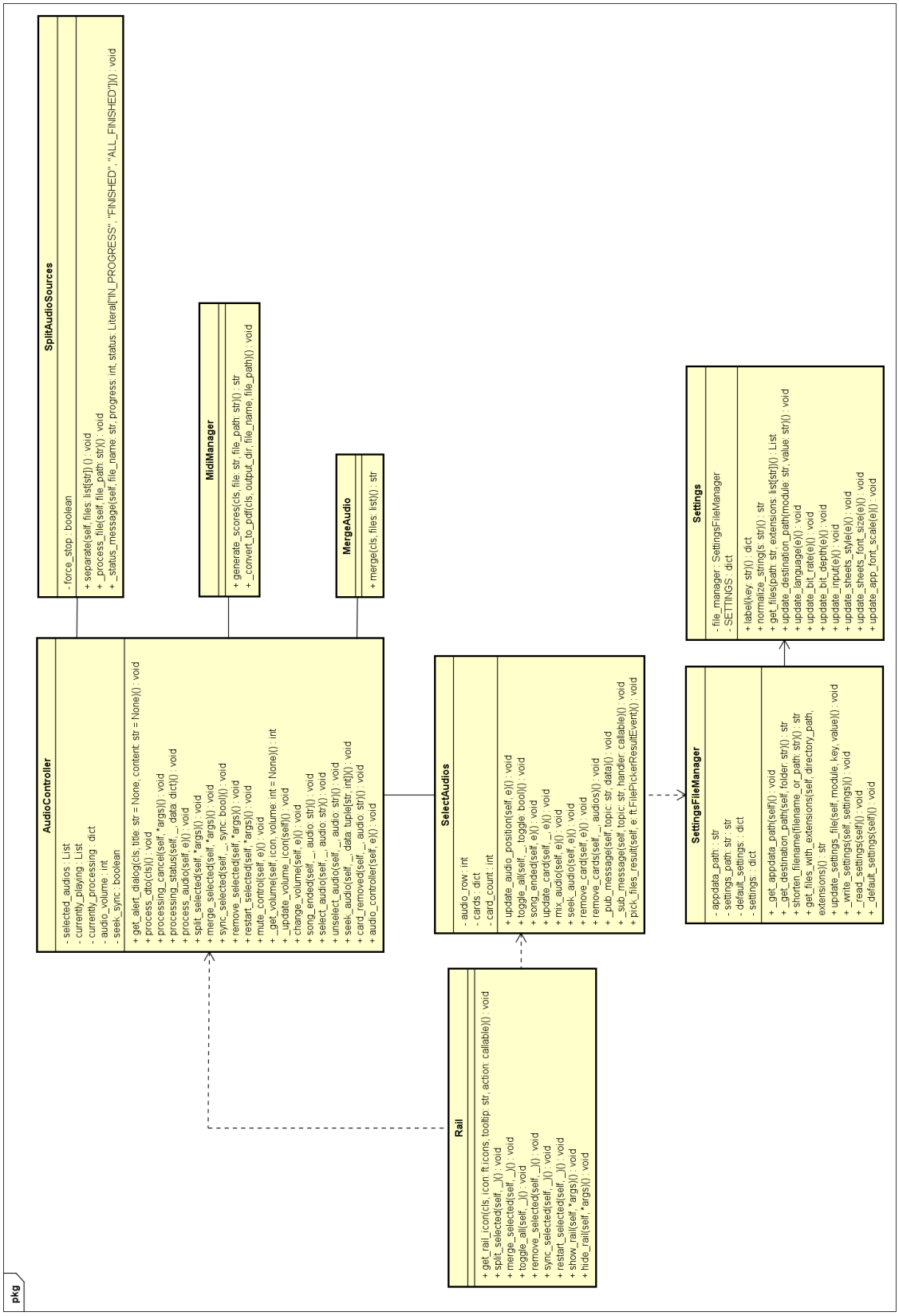
- DADO QUE o usuário acessa qualquer página do sistema
- QUANDO clica no ícone do menu lateral
- ENTÃO o menu deve ser expandido, exibindo as opções de navegação: "Gerar Partitura", "Áudios Gerados", "Partituras Geradas", "Configurações"

REGRAS DE NEGÓCIO DA HISTÓRIA:
R1 - O sistema deve aplicar as mudanças de configuração após reiniciar o sistema.

R2 - O sistema deve persistir as configurações para que sejam mantidas entre as sessões do usuário.

APÊNDICE C – DIGRAMA DE CLASSE

FIGURA 26 - DIAGRAMA DE CLASSE

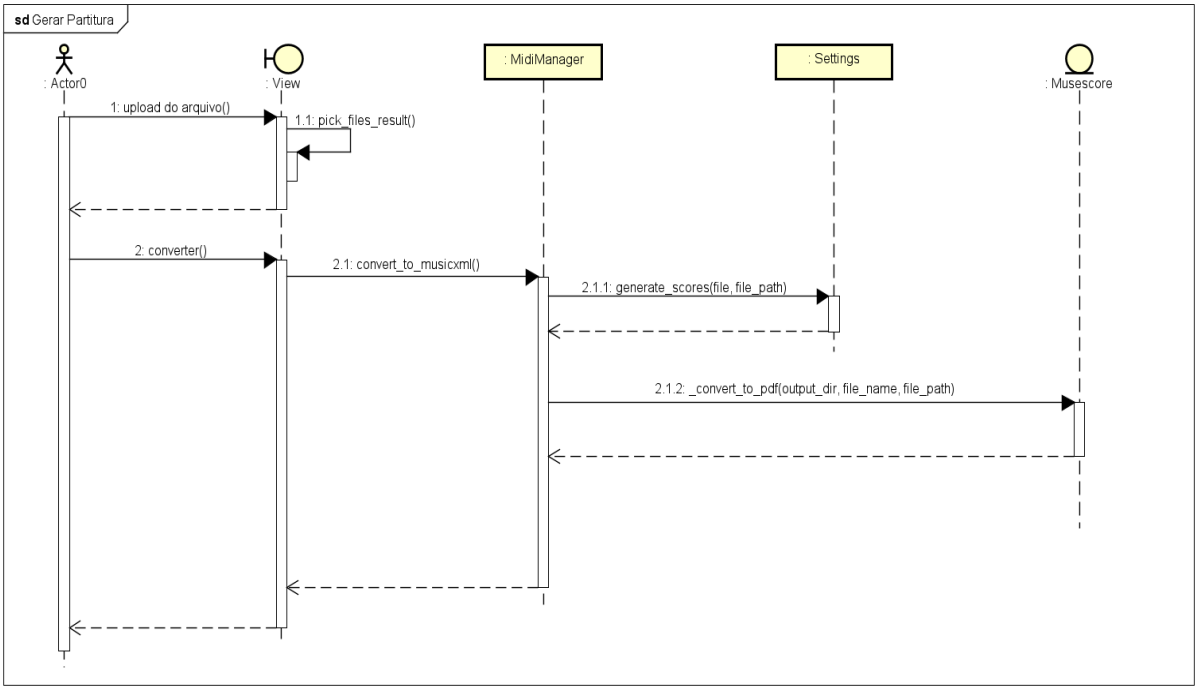


FONTE: Autores (2024)

APÊNDICE D – DIAGRAMAS DE SEQUÊNCIA

Diagrama da tela de gerar partitura:

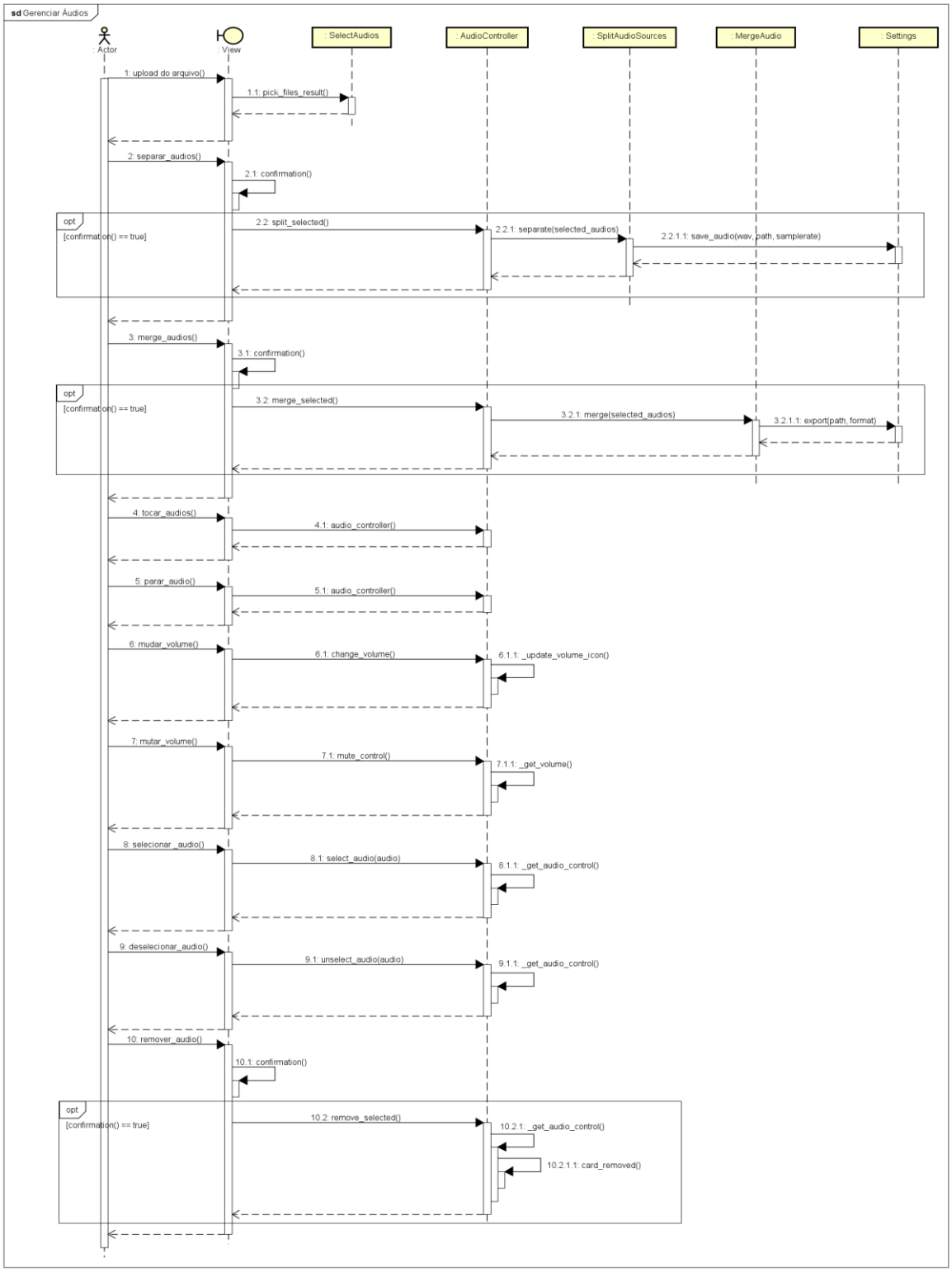
FIGURA 27 - DIAGRAMA DE SEQUÊNCIA - GERAR PARTITURA



FONTE: Os autores (2024)

Diagrama da tela de áudio salvos:

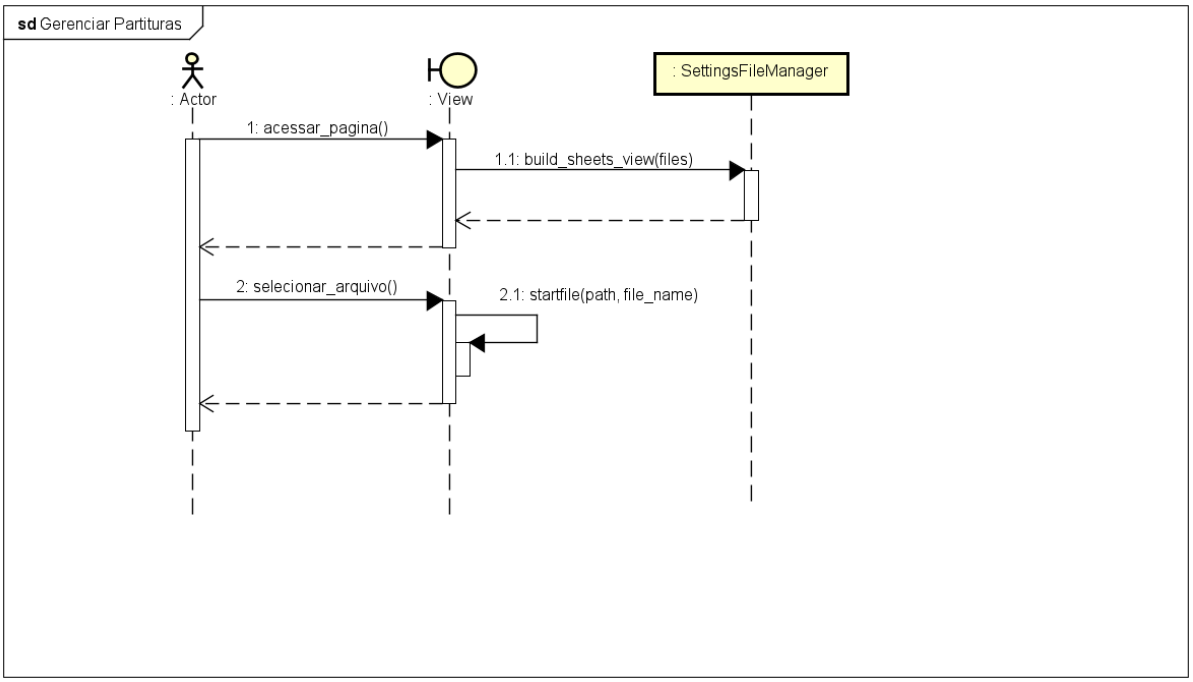
FIGURA 28 - DIAGRAMA DE SEQUÊNCIA - GERENCIAR ÁUDIOS



FONTE: Os autores (2024)

Diagrama da tela de partituras geradas:

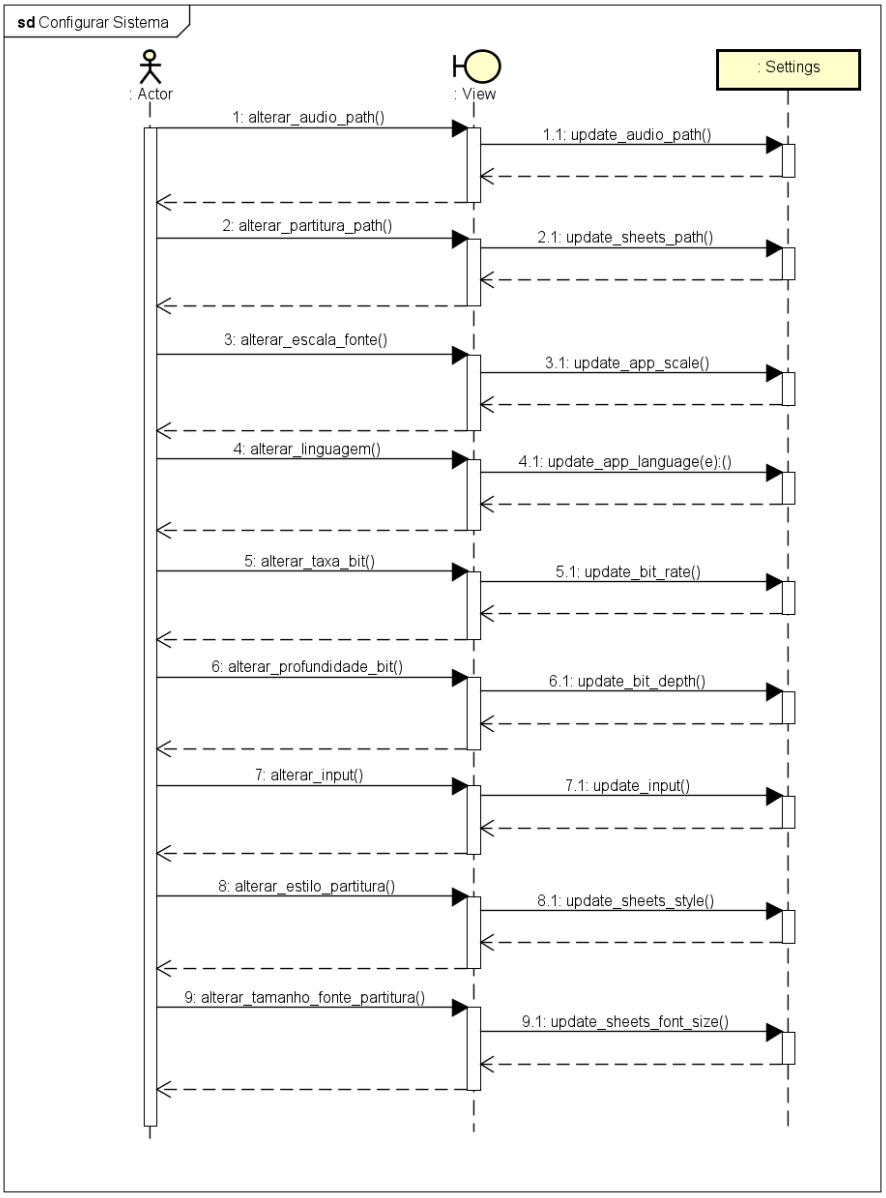
FIGURA 29 - DIAGRAMA DE SEQUÊNCIA - GERENCIAR PARTITURAS



FONTE: Os autores (2024)

Diagrama da tela de configurações gerais:

FIGURA 30 - DIAGRAMA DE SEQUÊNCIA - CONFIGURAR SISTEMA



FONTE: Os autores (2024)