

**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
CAMPUS DE BAURU – SP
SISTEMAS DE INFORMAÇÃO**

**ERICK SCHMOLLER PICHELI
KEVIN HIDEKI UENO**

USO DE INTELIGÊNCIA ARTIFICIAL PARA COMPOSIÇÃO MUSICAL

Bauru
2023

**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
CAMPUS DE BAURU – SP
SISTEMAS DE INFORMAÇÃO**

**ERICK SCHMOLLER PICHELI
KEVIN HIDEKI UENO**

USO DE INTELIGÊNCIA ARTIFICIAL PARA COMPOSIÇÃO MUSICAL

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru, como Trabalho de Conclusão de Curso.

Orientador: Prof. Douglas Rodrigues

Bauru
2023

P592u

PICHELI, Erick; UENO, Kevin Hideki. Uso de inteligência artificial para composição musical -- Bauru, 2023. 27 p. : il.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru

Orientador: Douglas Rodrigues

Coorientador: Kelton Augusto Costa

1. Utilização de IA para composição musical. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Resumo

PICHELI, Erick, S.; UENO, Kevin, H. **Uso de Inteligência Artificial para Composição Musical**. 2023. Monografia - Departamento de Computação. Universidade Estadual de São Paulo "Júlio de Mesquita Filho", Bauru, 2023.

Este trabalho apresenta uma proposta para o uso de inteligência artificial em composição musical, utilizando um modelo treinado de *Transformer* para a previsão e criação de notas musicais que auxiliam no processo de composição de melodias e acordes. Foi implementada a utilização de um arquivo .mid como uma “semente” que auxilia no direcionamento das melodias e composições geradas pela Inteligência Artificial (IA), gerando áudios condizentes com a composição musical original utilizada como inspiração.

Palavras-chave: Inteligência Artificial. *Transformer*. Composição musical. Música.

Abstract

This paper contains a proposal for the use of Artificial Intelligence in musical composition, using a trained Transformer model for the prediction and creation of musical notes that assists in the melody and chord creation process. The usage of a .mid file as a “seed” was implemented, which helps in the melody behavior and the AI-generated music compositions, creating audios fitting the original composition used as base.

Keywords: Artificial Intelligence. Transformer. Musical Composition. Music.

LISTA DE FIGURAS

Figura 1 - Estrutura de um Transformer	13
Figura 2 - Tela principal	17
Figura 3 - Tela de composição, inserção de parâmetros	18
Figura 4 - Tela de composição, resultado da geração	18
Figura 5 - Fluxograma das funções	19

Sumário

1. INTRODUÇÃO	8
2. SOLUÇÕES EXISTENTES E OBJETIVO DA PROPOSTA	10
3. DESCRIÇÃO DO PROJETO PROPOSTO	11
4. TECNOLOGIAS PESQUISADAS E ESCOLHIDAS	12
4.1. Inteligência Artificial - Transformer	12
4.2. Linguagem Python	14
4.3. Linguagem Javascript	14
4.4. React Framework	15
4.5. Git	15
5. ESPECIFICAÇÃO DA FERRAMENTA	16
5.1. Especificação Funcional	16
5.1.1. Descritiva	16
5.1.2. Use Cases	16
5.1.3. Interface	17
5.1.4. Fluxo principal	19
5.2. Especificação técnica	19
5.2.1. Tela principal	19
5.2.2. Tela de composição	20
5.2.2.1. Parâmetros	20
5.2.2.2. Composição	21
5.2.2.3. Reprodução e download	22
6. CASOS DE USO	23
6.1. Caso de uso 1 - Arranjo original	23
6.2. Caso de uso 2 - Composição original	23
7. ANÁLISE DOS RESULTADOS	25
8. CONCLUSÃO	26
Referências	27

1. INTRODUÇÃO

O uso e popularidade da inteligência artificial tem sido cada vez maior nos últimos anos. As IAs podem ser utilizadas para processamento de linguagem natural, análises de vídeo e imagem, sistemas autônomos, detecção de fraude, auxílio em diversas áreas de marketing, saúde, administração, indústria, análises comportamentais, preditivas, entre outros. Com isso, possui enorme relevância e aplicabilidade em diversos setores, incluindo o de produção musical.

Conforme esse avanço da inteligência artificial, principalmente nos campos artísticos, novas possibilidades, técnicas e ferramentas podem ser exploradas, gerando maneiras de auxiliar criativamente os artistas, sendo atualmente muito usadas no campo de desenhos digitais. Analogamente, compositores musicais poderiam se beneficiar de ferramentas com o auxílio de inteligência artificial para criar composições de maneira mais prática e eficiente.

Dessa forma, este trabalho tem como justificativa a necessidade crescente de compositores em produzir novas melodias e harmonias de maneira original e consistente. Um problema evidenciado no ramo artístico é conhecido como bloqueio artístico, em que o artista fica preso em um estado onde não consegue criar ou desenvolver ideias, e uma possível solução para esse problema é o uso da inteligência artificial para gerar conteúdos inéditos de criação musical. Além disso, até o momento, não há ferramentas capazes de efetivamente cumprir esse papel, com este trabalho tendo a intenção de permitir uma melhora considerável nesse ramo pouco explorado.

Assim, como solução, uma aplicação foi desenvolvida para auxiliar na criação de músicas, utilizando inteligência artificial para gerar conteúdo completamente original a partir de uma música proposta previamente, sendo continuada pela inteligência artificial. Tal aspecto da aplicação se encaixa no “modo interativo” (Briot, 2019) de geração de música, em que há interferência ou controle humano para ocorrer a geração.

Conforme apresentado anteriormente, o problema principal gira em torno da estimulação da criatividade do compositor por meio do uso de uma inteligência artificial para auxiliar no processo de composição musical.

A importância do problema é justificada pela ocasional limitação da expressão artística humana durante o processo repetitivo e árduo de composição musical, em que a automatização na criação de músicas facilita e promove uma experiência mais eficiente para o trabalho, que muitas vezes possui prazos apertados e uma necessidade crescente de otimizar o tempo de serviço.

Como uma ferramenta de auxílio à produção musical, um dos impactos esperados é o aumento do rendimento e rapidez nos serviços dos compositores, sendo benéfico para empresas em que a demanda por músicas é muito alta, e possuem prazos bem definidos em relação à entrega delas, como empresas que produzem jogos eletrônicos que são constantemente atualizados, tendo grande demanda por novas músicas de maneira periódica em curtos períodos de tempo. Outro impacto é a criação de músicas em um estilo consistente, ao poder alinhar suas características de composição musical por meio da música inserida na ferramenta. Com o uso de arquivos de música com estilos específicos, é possível simular perfeitamente padrões de composição de determinado gênero musical, ritmo ou até mesmo a criação das frases melódicas, gerando algo completamente novo com certo grau de familiaridade atribuído às composições, sendo consistente com o conteúdo original.

2. SOLUÇÕES EXISTENTES E OBJETIVO DA PROPOSTA

Uma solução similar existente é a encontrada no site musicautobot.com, em que um modelo de deep learning foi utilizado para completar composições musicais, treinado com MIDIs encontrados na internet. Entretanto, possui diversas limitações ao possuir uma seleção pequena de músicas e estilos, sendo em sua maioria músicas clássicas e pop, contendo pouco repertório sobre outros estilos musicais e seus funcionamentos, além de possuir resultados em geral muito parecidos com o material original, tendo muita pouca variabilidade em suas criações, não sendo ideal para novas composições.

Outra solução implementada anteriormente foi utilizada pelo MidiNet, que usa um modelo de CNN-GAN, *Convolutional Neural Network* usando *Generative Adversarial Networks*, e obteve bons resultados em criação de música, porém, foi observado que: “os profissionais apontaram que as melodias geradas pelo modelo MelodyRNN são, às vezes, muito repetitivas, ou fáceis de prever, tornando-as artificiais e menos interessantes” (Yang, 2017), ou seja, foi observado que os resultados eram muito previsíveis, similares, fáceis de ser inferidos, por também se repetirem muito, e tendo estruturas pouco criativas, sendo uma solução não ideal.

O objetivo da proposta deste trabalho é a utilização da inteligência artificial a fim de obter resultados criativos e novos, que contrastem bem com o material alimentado para o procedimento, combinando sequências melódicas e harmônicas de maneira a ter consistência na qualidade e não criar melodias genéricas ou com viés muito forte obtido a partir dos modelos de treinamento, como foi o caso das soluções propostas e citadas acima.

Um dos maiores problemas das soluções existentes é a falta de variedade nos resultados finais, criando melodias muito similares às músicas originais. Neste trabalho, o grau da variabilidade, chamado de temperatura, pode ser manualmente ajustado de acordo com a necessidade e julgamento do compositor, podendo gerar desde melodias e acordes similares ao material original até completamente diferentes, mas que continuam seguindo o estilo geral da música, adicionando um grau de liberdade e customização ao músico.

Outro problema observado nas soluções existentes foi a falta de adaptabilidade aos estilos musicais diversos. Neste trabalho, isso foi resolvido devido à implementação de uma função que automaticamente verifica o valor das batidas por minuto da música inserida, que quantitativamente determinam o quão rápida ela é. Dessa forma, a identificação da maneira de como as notas estão distribuídas para cada música adicionada na ferramenta possui confiabilidade e consistência muito maiores em relação às soluções anteriores.

3. DESCRIÇÃO DO PROJETO PROPOSTO

Desenvolvemos uma aplicação web que utiliza inteligência artificial para compor melodias, se baseando na entrada de dados de um arquivo .mid. A aplicação funciona através de um *Transformer*, com um modelo de aprendizado treinado, através de uma grande massa de dados de arquivos .mid já existentes, que são utilizados para agregar ao “conhecimento” do modelo, deixando-o apto para a tarefa de composição musical, utilizando o music21 e fast.ai.

Com o modelo, é utilizado o conhecimento obtido para a previsão de notas seguintes, a partir de uma música de arquivo .mid inserida previamente. A IA se baseia no arquivo inserido para ajustar-se ao estilo e composição originais, ao mesmo tempo que procura trazer variações e novas melodias e acordes, sendo possível controlar esse aspecto de variabilidade e repetição por meio de parâmetros, customizados pelo próprio usuário.

Além disso, também é permitida a especificação de algumas características da composição, como a duração do áudio a ser criado, o beat de corte para definir até onde a IA vai levar em consideração na música inserida, e a liberdade dada para a IA, como a variância de notas e de tempo.

A ferramenta possui uma interface projetada para permitir a entrada desses parâmetros que modificam o comportamento da IA, assim como a inserção de arquivos .mid base por meio de anexo de arquivos ou pelo *Drag and Drop*, arrastando o arquivo para a tela. Além disso, pode ser feita a reprodução e download individual dos áudios gerados, tanto para a composição completa (com o arquivo .mid inicial) quanto para a composição da IA (apenas o que a IA gerou baseada no arquivo .mid inicial).

4. TECNOLOGIAS PESQUISADAS E ESCOLHIDAS

Para o trabalho, algumas tecnologias foram pesquisadas e escolhidas para possibilitar o desenvolvimento da aplicação proposta.

4.1. Inteligência Artificial - Transformer

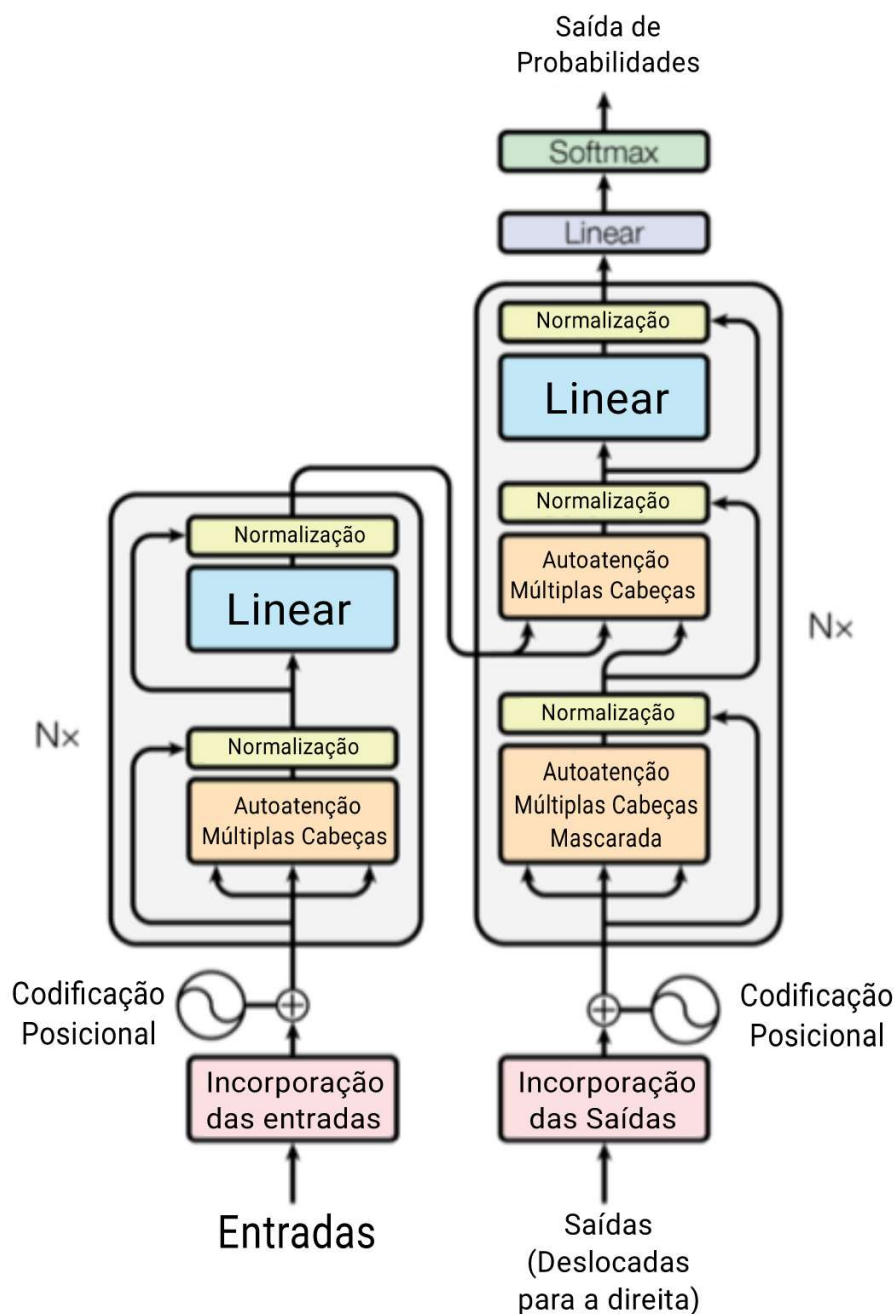
Foi utilizada uma inteligência artificial do tipo *Transformer* como o motor principal da ferramenta a ser desenvolvida. Além disso, foi usado um modelo treinado com uma massa de dados de diversos arquivos .mid para realizar as previsões de notas seguintes.

O *Transformer* é uma arquitetura de rede neural introduzida em 2017 no artigo "Attention is All You Need" (Vaswani, 2017), notável por usar mecanismos de atenção para processar sequências de dados de comprimento variável sem depender de estruturas recorrentes. Sua capacidade de atenção permite capturar relações de longo alcance, enquanto a eliminação da necessidade de recorrência facilita o treinamento e evita problemas associados ao gradiente. O alto paralelismo durante o treinamento, juntamente com camadas de autoatenção para ponderação de entrada, contribuem para a eficiência do modelo. O *Transformer* é amplamente utilizado em tarefas de processamento de linguagem natural e tradução automática, e se destaca por sua escalabilidade e capacidade de lidar com sequências de tamanhos diversos.

Sua estrutura é composta por camadas, estas contendo duas subcamadas: *Multi-Head Attention*, caracterizada pelo processamento das informações de entrada, a fim de selecionar e obter seus pesos de acordo com suas importâncias. A segunda subcamada é o Feedforward, responsável por aplicar operações com os dados de entrada, a fim de detectar padrões nos dados e permitir a capacidade de aprendizado do modelo. Há também o processo de *Add and Normalize*, feitos após a execução das subcamadas, com a finalidade de ter maior estabilidade no treinamento. A etapa de *Add* serve para somar o resultado da operação com a entrada original, a fim de permitir a retenção do conhecimento prévio, e a etapa de *Normalize* funciona para fazer uma normalização dos valores a fim de mantê-los em tamanhos adequados a fim de não prejudicar o funcionamento do modelo. Além disso, há também conexões residuais entre camadas, a fim de transmitir informações aprendidas anteriormente para garantir estabilidade no treinamento, preservando partes relevantes de informações obtidas previamente. Na saída, é aplicada a função softmax, em que o resultado é uma representação de probabilidades, a fim de poder interpretar e ajustar as informações obtidas de

maneira mais precisa conforme as iterações passam. É possível ver seu funcionamento na Figura 1.

Figura 1. Estrutura de um *Transformer*



Fonte: Vaswani, 2017. Traduzido.

O *Transformer* destaca-se em previsões musicais devido à sua capacidade de atenção, que permite considerar relações temporais complexas e dependências não lineares em sequências musicais. A eliminação da necessidade de recorrência facilita a modelagem de padrões de longo alcance, enquanto a escalabilidade para sequências de comprimento variável

é vantajosa para lidar com músicas de diferentes durações. Essas características tornam o *Transformer* eficaz em tarefas de geração automática de músicas.

O *Transformer* foi escolhido pelo fato de que é uma arquitetura de modelo de aprendizado de máquina projetada para processar sequências de dados, sendo notável por sua capacidade de capturar relações complexas em dados sequenciais.

A utilização desse modelo nesse caso é ideal para prever músicas porque pode aprender padrões temporais e dependências complexas em sequências musicais, tornando-se eficaz para tarefas de geração automática de composições devido à sua habilidade de lidar com estruturas musicais complexas.

4.2. Linguagem Python

Foi utilizada a linguagem de programação Python e diversas bibliotecas em Python para o desenvolvimento da ferramenta, como a transformação do arquivo .mid em um arquivo .wav para que possa ser reproduzido, entre outros aspectos relacionados à manipulação de dados.

Python foi a linguagem escolhida pois é a mais indicada para ciência de dados, especialmente por não ser fortemente tipada, ou seja, o tipo das variáveis são estabelecidos durante a execução e também podem ser modificados dinamicamente.

Para que o lado do cliente possa se comunicar com os métodos desenvolvidos em Python, um servidor Flask foi utilizado. O Flask é um microframework que possibilita o desenvolvimento de aplicações web utilizando um back-end em Python.

Além disso, foi utilizada a biblioteca music21, útil para trabalhos musicais em Python. Também foi usado o fast.ai como uma ferramenta que disponibiliza modelos para as predições, e o Fluidsynth, um software de código aberto que proporciona a conversão de notas MIDI em áudio por meio de soundfont, um formato de arquivo de áudio próprio para o MIDI.

4.3. Linguagem Javascript

Para desenvolvimentos relacionados ao lado do cliente, a linguagem JavaScript foi escolhida. Javascript é uma linguagem de programação de alto nível, interpretada e amplamente utilizada para o desenvolvimento de aplicações web.

JavaScript permite manipular elementos HTML, interagir com o usuário, realizar requisições assíncronas, e atualizar partes específicas de uma página sem a necessidade de recarregá-la completamente. Sua versatilidade e integração natural com tecnologias web

fazem do JavaScript uma escolha central para o desenvolvimento de aplicações web modernas e responsivas.

4.4. React

React foi escolhido pois é uma biblioteca JavaScript amplamente utilizada para o desenvolvimento de aplicações web. A estrutura de componentização facilita a criação de elementos reutilizáveis, promovendo uma organização e manutenção mais simples do código.

4.5. Git

O Git foi uma escolha ideal para o versionamento do projeto devido à sua eficiência, flexibilidade e recursos poderosos. Ele permite o controle preciso das versões do código-fonte, e oferece ramificações eficientes, facilitando o desenvolvimento paralelo de funcionalidades. A velocidade e eficiência do Git também contribuem para uma experiência de versionamento ágil, tornando-o uma escolha ideal para projetos de desenvolvimento.

5. ESPECIFICAÇÃO DA FERRAMENTA

Para melhor demonstrar como a ferramenta funciona, a especificação descreve as principais características e funções da ferramenta, bem como suas telas e fluxo de funcionamento.

5.1. Especificação Funcional

Para a descrição das principais funções do software, são apresentados modelos que descrevem, em alto nível, as telas, botões e funcionalidades, suas entradas e saídas e a maneira de interatividade entre o usuário e o programa.

5.1.1. Descritiva

A especificação funcional descritiva a seguir apresenta a maneira que as telas se comportam, e sua interatividade com o usuário. A tela inicial permite visualizar os arquivos .mid pré-carregados e adicionar novos conforme a instrução *Drag and Drop*, ou anexar diretamente pela pasta de arquivos, além de possibilitar a remoção de arquivos dessa lista ao apertar o botão de remover música, em que o áudio original no computador do usuário não é juntamente apagado, apenas será removido da lista. Há um botão de *Start* ao lado de cada arquivo musical que, ao ser clicado, abre uma nova janela.

A nova janela permite ao usuário inserir parâmetros para alterar o comportamento da música gerada pela IA, além de poder abrir as configurações avançadas, em que é possível visualizar e editar os parâmetros de punição por repetição, denotados pela penalidade de repetição e a janela de penalidade. Após preencher esses parâmetros, o botão de gerar composição pode ser apertado. Após a geração, o usuário pode escutar o áudio gerado com a música original ou apenas a criação da IA, com botões de *Play* e *Sliders* para controlar a posição em que a música sendo reproduzida se encontra. Além disso, é possível baixar ambos os áudios gerados com o botão de *Download*. Há também um botão para fechar esta janela e retornar para a lista de músicas.

5.1.2. Use Cases

O usuário deve ser capaz de inserir e remover arquivos de áudio próprios. O usuário deve ser capaz de alterar os dados dos parâmetros e gerar resultados diferentes, com a mesma música de entrada. Após o botão de gerar composição for pressionado, dois arquivos .wav

serão gerados a partir do arquivo .mid original, que serviu como entrada para a geração da composição musical. O usuário deve ser capaz de baixar os áudios do resultado da geração dos áudios da IA.

5.1.3. Interface

A interface consiste em duas telas da aplicação web, a primeira, apontada pela Figura 2, e a segunda, demonstrada nas Figuras 3 e 4.

Figura 2. Tela principal

The screenshot displays a web interface titled "Files". It contains a table with two columns: "Name" and "Actions". The table lists seven files, each with a "DELETE" button (represented by a trash icon) and a "START" button (represented by a play icon). Below the table, there is a section titled "Drag a File" which includes a dashed rectangular box for file upload. Inside this box, there is a "SELECT FILE" button with a folder icon, and the text "or Drag & Drop a file here" below it.

Name	Actions
Canon In D Major - Johann Pachelbel - Chorus.mid	DELETE START
Fuer Elise - Ludwig Van Beethoven - Verse.mid	DELETE START
In The Hall Of The Mountain King - Edvard Grieg - Intro.mid	DELETE START
Labrynth Crystal.mid	DELETE START
Levels - Avicii - Verse.mid	DELETE START
Locked Out Of Heaven - Bruno Mars - Chorus.mid	DELETE START
Scary Monsters And Nice Sprites - Skrillex - Pre-Chorus.mid	DELETE START

Drag a File

SELECT FILE

or Drag & Drop a file here

Fonte: De autoria própria.

Figura 3. Tela de Composição, inserção de parâmetros.

The screenshot shows a 'Song Generation' settings window with a close button (X) in the top right corner. The settings include:

- Cut Off Beat:** A text input field containing the value '10'.
- Temperature:** A slider control with a blue knob positioned at 1.2.
- Rhythmic Variation:** A slider control with a blue knob positioned at 1.2.
- Duration:** A text input field containing the value '200'.
- Repeat Penalty:** A slider control with a blue knob positioned at 0.
- Repeat Penalty Window:** A slider control with a blue knob positioned at 0.
- Original Selected Song:** A section header above a playback interface consisting of a progress bar, a play button, a volume icon, a volume slider, and a download icon.

Fonte: De autoria própria.

Figura 4. Tela de composição, resultado da geração

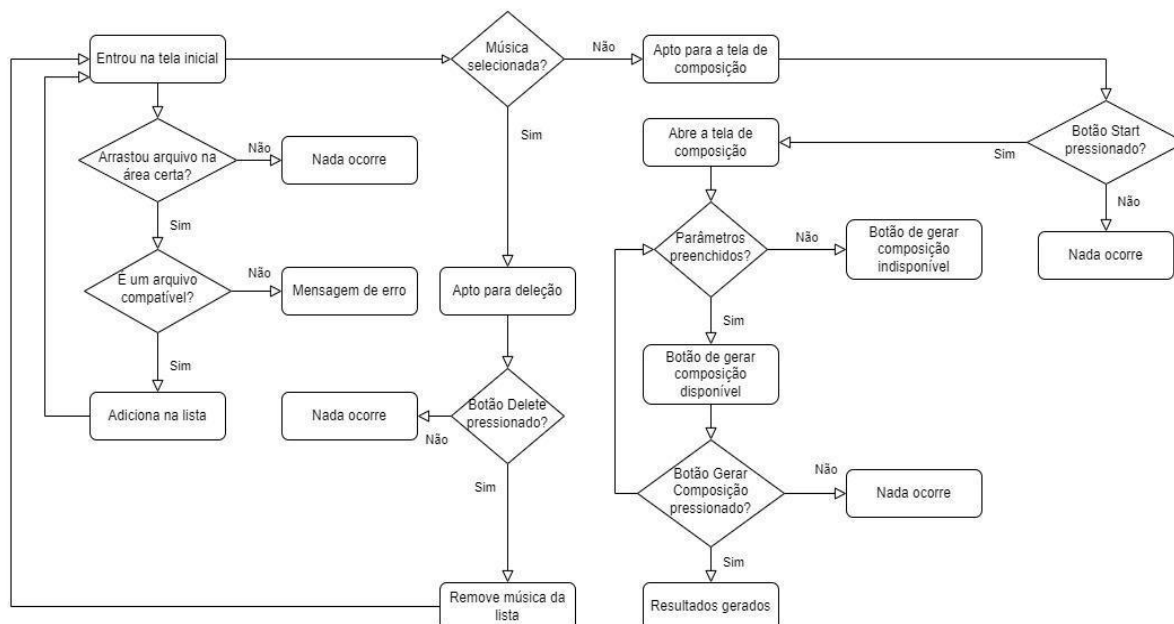
The screenshot shows the 'Full Generated Song' section of the application. It features a blue 'GENERATE' button at the top. Below it, the text 'Full Generated Song:' is displayed. Underneath, there are two identical playback interfaces, each consisting of a progress bar, a play button, a volume icon, a volume slider, and a download icon. The first interface is labeled 'Full Generated Song:' and the second is labeled 'AI Generated Song:'.

Fonte: De autoria própria.

5.1.4. Fluxo principal

O fluxo principal de funcionalidades que o programa possui está apresentado conforme a Figura 5, evidenciando o comportamento entre os botões e as telas.

Figura 5. Fluxograma das funções



Fonte: De autoria própria.

5.2. Especificação técnica

A especificação consiste na explicação de cada uma das telas e suas funcionalidades, detalhando como o programa irá se comportar.

5.2.1. Tela principal

A tela inicial consiste em uma lista de arquivos .mid, cada um acompanhado de um botão *Start* e uma checkbox. A tela instrui o usuário com o comando *Drag and Drop*, que permite arrastar e soltar um arquivo .mid de uma pasta para o programa, a fim de importar uma música de algum diretório do computador do usuário. O botão *Delete* remove os respectivos arquivos da interface do programa. O botão *Start* ao lado do arquivo musical permite utilizar a respectiva música na tela de composição, que será aberta após clicar no botão.

5.2.2. Tela de composição

A tela de composição consiste na inserção ou alteração de parâmetros, incluindo os avançados, juntamente com o botão de geração de composição, que, após gerada, disponibiliza os áudios para reprodução e *download*.

5.2.2.1. Parâmetros

O usuário pode inserir parâmetros para alterar como a música se comporta, sendo eles a Temperatura, que denota a liberdade melódica da IA, a Variação de Ritmo, denotando a liberdade rítmica entre as notas musicais e o Beat de Corte, cujo valor determina quando ocorre a transição entre a música original e a geração da IA. A inserção dos parâmetros é feita por meio de uma caixa de texto para o Beat de Corte e *Sliders* para inserir valores de temperatura e variação rítmica, explicados anteriormente. Além disso, há a opção de alterar parâmetros avançados, representados pela penalidade de repetição e a janela de penalidade.

O funcionamento dos parâmetros avançados de penalidade é caracterizado por um valor arbitrário de penalização (penalidade de repetição), e um valor arbitrário relacionado a janela de últimas notas tocadas a serem consideradas (janela de penalidade de repetição).

Dentro do método de geração, penalizamos todos os tokens de notas que já foram utilizados dentro da janela de penalidade, dessa forma, evitamos a repetição de notas e encorajamos uma maior criatividade no output. Para isso, foi desenvolvido um algoritmo específico. Esse algoritmo mantém armazenadas as últimas n notas, conforme definido pela janela de penalidade de repetição, e aplica uma penalidade arbitrária (definida pelo usuário como penalidade de repetição) ao token associado a essas notas. A penalização, cujo valor é especificado pelo usuário, é então aplicada à pontuação, ou logit, do token de nota relevante. Em seguida, a função softmax, caracterizada como uma função de ativação, é calculada utilizando esses logits modificados pela penalidade, e o token é selecionado levando em consideração essas penalidades durante o processo de escolha.

As notas são armazenadas em um array de capacidade n (correspondente à janela de penalidade de repetição), e em cada iteração do método de geração, essas penalidades são reintroduzidas. Cada vez que o array atinge o limite de n notas armazenadas (janela de penalidade de repetição), ocorre uma remoção da nota mais antiga adicionada, assegurando a conformidade com o limite de notas da janela de penalidade de repetição. Cada vez que uma nota é removida do array, sua pontuação recebe um aumento positivo equivalente ao valor arbitrário de penalidade.

5.2.2.2. Composição

Preenchidos e ajustados os três valores, o botão de Gerar Composição pode ser pressionado. O botão de gerar composição chama o método principal encontrado no lado do servidor Flask. Esse método processa os parâmetros enviados e utiliza o modelo pré-treinado para realizar uma predição (composição).

Como explicado no tópico 5.1, utilizamos um *Transformer* para realizar essas predições (composições). Para que um *Transformer* seja capaz de reconhecer inputs, esses inputs precisam ser tokenizados, o que é feito ativamente no servidor. O processo de geração de tokens é similar ao apresentado por Shaw (2019).

Depois do MIDI inicialmente inserido ser reconhecido e cortado no beat de corte informado, a IA então gera novos tokens. Um exemplo de um array contendo esses tokens gerados é o seguinte:

```
['xxsep', 'd2', 'n72', 'd4', 'xxsep', 'd4', 'n72', 'd4', 'xxsep', 'd4', 'n69', 'd4', 'n57', 'd8', 'n53', 'd8', 'n50', 'd8', 'xxsep', 'd4', 'n71', 'd2', 'xxsep', 'd2', 'n72', 'd2', 'xxsep', 'd2', 'n74', 'd2', 'n55', 'd8', 'n52', 'd8', 'n48', 'd8', 'xxsep', 'd2', 'n76', 'd2', 'xxsep', 'd2', 'n77', 'd2', 'xxsep', 'd2', 'n76', 'd2', 'xxsep', 'd2', 'n72', 'd4', 'n48', 'd16', 'n45', 'd16', 'n41', 'd16', 'xxsep', 'd4', 'n69', 'd4', 'xxsep', 'd4', 'n72', 'd4', 'xxsep', 'd4', 'n76', 'd4', 'xxsep', 'd4', 'n74', 'd4', 'n50', 'd16', 'n47', 'd16', 'n43', 'd16', 'xxsep', 'd4', 'n67', 'd4', 'xxsep', 'd4', 'n69', 'd2', 'xxsep', 'd2', 'n71', 'd2', 'xxsep', 'd2', 'n72', 'd2', 'xxsep', 'd2', 'n74', 'd2', 'xxsep', 'd2', 'n76', 'd4', 'n52', 'd16', 'n48', 'd16', 'n45', 'd16', 'xxsep', 'd4', 'n69', 'd4', 'xxsep', 'd4', 'n72', 'd4', 'xxsep', 'd4', 'n69', 'd4', 'xxsep', 'd4', 'n72', 'd4', 'n57', 'd8', 'n53', 'd8', 'n50', 'd8', 'xxsep', 'd4', 'n71', 'd2', 'xxsep', 'd2', 'n72', 'd2', 'xxsep', 'd2', 'n74', 'd2', 'n55', 'd8', 'n52', 'd8', 'n48', 'd8', 'xxsep', 'd2', 'n76', 'd2', 'xxsep', 'd2', 'n77', 'd2', 'xxsep', 'd2', 'n76', 'd2', 'xxsep']
```

Observando os tokens gerados podemos verificar três tipos de tokens distintos:

- **xxsep**: repetimos esse token toda vez que desejamos realizar uma separação, essa separação é caracterizada pela consolidação das notas e tempos anteriormente inseridos.
- **dx**: como podemos observar, diversos tokens são iniciados com a letra ‘d’, esses tokens representam uma duração, definida pelo valor numérico inserido logo depois da letra ‘d’.
- **nx**: verificamos também na lista que certos tokens são iniciados com a letra ‘n’, esses tokens representam uma nota. A nota a ser tocada é definida pelo valor numérico que vem logo em seguida.

5.2.2.3. Reprodução e Download

Após o botão de geração ser apertado e o carregamento ser concluído, aparecerão dois grupos de botões, um para a música completa, que inclui o áudio original e a criação advinda da inteligência artificial, e outro para a música composta apenas pela inteligência artificial, sem incluir o áudio original. Os grupos de botões são constituídos por *Play*, em que o áudio da música é reproduzido, um *Slider*, mostrando onde a reprodução da música está em relação ao áudio completo, sendo possível arrastar e selecionar o ponto onde o usuário deseja escutar especificamente, e por fim, o botão de *Download*, que permite baixar o conteúdo do áudio gerado para o computador do usuário, cujo nome é automaticamente gerado com o nome da música concatenado com a data e hora da geração, a fim de não obter arquivos com nomes repetidos. O usuário pode também inserir outros parâmetros e tentar gerar uma nova composição a partir do mesmo áudio, porém, a geração anterior será perdida se não for salva antes do botão *Generate* ser apertado novamente, visto que o novo resultado irá sobrescrever o anterior.

6. CASOS DE USO

A seguir, são apresentados dois casos de uso de criações musicais arranjadas utilizando como base composições geradas pela ferramenta deste trabalho. Para os arranjos, foi utilizado o teclado arranjador Yamaha PSR-E423, um instrumento musical similar a um piano, que permite gravar e reproduzir sons de diversos outros instrumentos. Os arranjos estão disponíveis para acesso no *Google Drive*.¹

6.1. Caso de uso 1 - Arranjo original

Para este arranjo, foi utilizado como base um áudio gerado a partir da música Canon in D do compositor Johann Pachelbel, escrita no início do século XVIII. Desse áudio gerado, apenas a primeira melodia foi extraída e foi feita uma transcrição integral, completa, exata, desta. Assim, o arranjo mostra-se como um caso de uso em que a criação da IA está totalmente inserida e sendo utilizada em uma música, afirmando a possibilidade de uso das criações de maneira íntegra, sem alterações. Ou seja, o propósito deste arranjo foi verificar se uma criação da IA pode ser transcrita de maneira idêntica em uma música e obter um resultado musical satisfatório, sendo evidentemente alcançado no resultado deste caso de uso. No arranjo, de autoria própria, primeiramente a melodia da IA é apresentada pelo piano eletrônico, seguida de linhas de bateria para o ritmo. Então é adicionada uma guitarra rítmica, um baixo de acompanhamento e, por fim, um saxofone. Além disso, foi aplicada uma técnica de síncope, em que a ênfase do ritmo é colocada em um tempo diferente do usual, evidenciando o aspecto futurístico da música, com arranjo inspirado altamente no gênero musical “*Prog Rock*”.

6.2. Caso de uso 2 - Composição original

Para este arranjo, dois áudios foram utilizados como base, gerados a partir da música Fur Elise, do compositor Ludwig van Beethoven, escrita no século XIX. Com os dois áudios gerados, foi utilizado um trecho da melodia do primeiro áudio como base para a composição original, em que a melodia foi levemente alterada e o espaçamento no tempo também foi alterado, adicionando mais pausas. O propósito deste arranjo foi verificar se o processo de composição realmente é acelerado ao poder utilizar um arquivo gerado pela IA apenas como

¹ Para escutar os arranjos e as composições utilizadas como base nos casos de uso, acesse: https://drive.google.com/drive/folders/1meBMkuC1y9vonXh_nPNOB7UhCc9YPfEk?usp=sharing.

inspiração, e o resultado foi satisfatório, com o arranjo, de autoria própria, sendo criado rapidamente durante sua criação, incitando a criatividade de maneira competente. Para o arranjo em si, foi escolhido o gênero musical de valsa, com elementos de música clássica e orquestral. Como Fur Elise é escrita no compasso de tempo de 3/8, a valsa combina mais no estilo da música original, ao ser escrita usualmente em 3/4, sendo compatível com as escolhas de arranjo escolhidas. Como apontado anteriormente, a melodia principal do piano no início foi altamente inspirada no primeiro áudio gerado, juntamente com um acordeão e uma seção de violinos para o acompanhamento e percussão com castanholas e tambores para o ritmo. Logo em seguida, a melodia principal novamente inspirada no primeiro áudio gerado pela IA torna-se do acordeão, e, após essa seção, há o desfecho da música com os violinos assumindo a melodia principal quase totalmente transcrita do final do segundo áudio gerado por IA, sendo apenas o final da melodia escrito manualmente, a fim de haver coesão para encerrar a música.

7. ANÁLISE DOS RESULTADOS

A princípio, os resultados obtidos com as composições geradas estavam ocorrendo com sucesso, entretanto, havia grande vício melódico e repetições de notas constantemente. Utilizar altas temperaturas não estava contribuindo de maneira significativa para a melhora das melodias, devido ao ritmo e melodia ficarem variados demais ao ponto de tornarem-se completamente desordenados, aleatórios. Além disso, as batidas por minuto específicas de cada música estavam promovendo alterações de ritmo acentuadas nas composições geradas. Este último problema foi resolvido com a criação de um algoritmo que automaticamente detecta o valor das batidas por minuto do arquivo .mid inserido, corrigindo todos os problemas rítmicos que estavam ocorrendo. Para o vício melódico e repetições, o problema foi consertado através do aperfeiçoamento de hiperparâmetros e reiterações no método de geração, bem como a inserção da penalidade de repetição possibilitada pela customização dos parâmetros avançados, reduzindo drasticamente o problema das repetições. Já o problema da aleatoriedade nas notas devido à temperatura foi resolvido por meio da substituição do preenchimento de temperatura por caixa de texto para um *slider*, que possui valor mínimo e máximo, e fica visualmente vermelho para o usuário compreender que está em valor alto.

Com isso, todos os problemas encontrados durante o trabalho foram resolvidos, com o resultado final das composições geradas sendo sempre consistente e criativo musicalmente, sendo perceptível analiticamente conforme os testes unitários feitos durante o desenvolvimento, que serviram para avaliar as músicas geradas qualitativamente. Como não há maneira objetiva de julgar as músicas, a metodologia utilizada foi apenas criar diversos áudios, escutar os resultados e comparar qual teve o melhor alinhamento e criatividade em relação ao áudio inserido. Assim, comparativamente e com testagem de diversas entradas de parâmetros, foi possível determinar que após todas as alterações concluídas, a ferramenta possui grande consistência e capacidade de gerar composições musicais de qualidade.

8. CONCLUSÃO

Este trabalho apresentou um uso de inteligência artificial para auxiliar a composição de músicas, buscando solucionar possíveis problemas de bloqueio artístico que compositores podem ter. Com o desenvolvimento da ferramenta concluído, os compositores podem gerar inspiração e conteúdo de forma rápida e eficiente, customizando os parâmetros como preferirem até atingirem um resultado que os satisfaçam.

Para trabalhos futuros, pensamos em inserir uma forma de permitir ao usuário gerar PDFs ou acessar com facilidade as notas geradas, isso seria feito para que o usuário possa reproduzir a composição com seu próprio instrumento, uma das formas pensadas é a disponibilização de uma partitura.

Outra possível adição é a reprodução com diferentes instrumentos além do piano, podendo ser implementada uma opção de customizar os sons dos instrumentos tocando no áudio gerado, tanto para os acordes quanto para a melodia principal. Por exemplo, escolher um instrumento virtual de violino para tocar a melodia principal e um outro instrumento virtual de violoncelo para tocar os acordes.

O treinamento de um modelo transformer mais robusto, com uma quantidade maior de arquivos MIDI, também é desejável. Nesse caso separaremos um dataset com uma quantidade favorável de arquivos. O treinamento seria feito com arquivos MIDI seletos de certos gêneros musicais ou com arquivos gerais sem nenhum gênero específico em mente.

Dessa forma, há um alinhamento com o rápido aumento da utilização de IA em diversas áreas e contribuindo, ainda mais, com a exploração dessa tecnologia, especialmente para o ramo de produção musical.

Referências

YANG, Li-Chia, CHOU, Szu-Yü, YANG, Yi-Hsuan. **MIDINET: A CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK FOR SYMBOLIC-DOMAIN MUSIC GENERATION**. 2017. 8 f. Research Center for IT innovation, Academia Sinica, Taipei, Taiwan. Disponível em: <<https://arxiv.org/abs/1703.10847>>. Acesso em: 10 abr. 2023.

MUSICAUTOBOT - The Pop Music Generator. **Musicautobot**, 2023. Disponível em: <<https://musicautobot.com/about>>. Acesso em: 12 abr. 2023.

SHAW, Andrew. **Creating a Pop Music Generator with the Transformer**. 2019. Disponível em: <<https://towardsdatascience.com/creating-a-pop-music-generator-with-the-transformer-5867511b382a>>. Acesso em 12 abr. 2023.

SHAW, Andrew. **Practical Tips for Training a Music Model**. 2019. Disponível em: <<https://towardsdatascience.com/practical-tips-for-training-a-music-model-755c62560ec2>>. Acesso em 12 abr. 2023.

Briot, Jean-Pierre, Gaëtan, Hadjeres. **Deep Learning Techniques for Music Generation – A Survey**. 2019. 209f. Sorbonne Université, Sony Computer Science Laboratories, Spotify Creator Technology Research Lab, Paris, França. Disponível em: <<https://arxiv.org/pdf/1709.01620.pdf>>. Acesso em: 10 abr. 2023.

VASWANI, Ashish et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.