

**Universidade Federal da Paraíba  
Centro de Informática  
Programa de Pós-Graduação em Computação, Comunicação e Artes**

**VICTOR AMARAL FREITAS**

**MusicMeaning: Ferramenta gamificada para introdução de conceitos de programação a partir de conhecimentos sobre música**

**João Pessoa**

**Maio / 2021**

**Victor Amaral Freitas**

MusicMeaning: Ferramenta gamificada para introdução de conceitos de programação a partir de conhecimentos sobre música

Dissertação apresentada ao Programa de Pós-Graduação em Computação, Comunicação e Artes (PPGCCA) da Universidade Federal da Paraíba, como requisito parcial para a obtenção do título de Mestre em Computação, Comunicação e Artes, na linha de pesquisa Arte Computacional.

Orientador: Prof. Dr. Alexandre Magno e Silva Ferreira

João Pessoa

Maio / 2021

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

F866m Freitas, Victor Amaral.

MusicMeaning : ferramenta gamificada para introdução de conceitos de programação a partir de conhecimentos sobre música / Victor Amaral Freitas. - João Pessoa, 2021.

79 f. : il.

Orientação: Alexandre Magno e Silva Ferreira.  
Dissertação (Mestrado) - UFPB/CI.

1. Linguagem de programação - Música. 2. Teoria aprendizagem significativa. 3. Ferramenta gamificada - MusicMeaning. I. Ferreira, Alexandre Magno e Silva. II. Título.

UFPB/BC

CDU 004.43:78(043)

**Victor Amaral Freitas**

**MusicMeaning: Ferramenta gamificada para introdução de conceitos de programação a partir de conhecimentos sobre música**

Dissertação apresentada ao Programa de Pós-Graduação em Computação, Comunicação e Artes da Universidade Federal da Paraíba, como requisito parcial para a obtenção do título de Mestre em Computação, Comunicação e Artes, na linha de pesquisa Arte Computacional.

A banca considera o presente Trabalho Final: **APROVADO**

Data: 31/ MAIO / 2021.



---

**Prof. Dr. Alexandre Magno e Silva Ferreira (Orientador – PPGCCA/UFPB)**



---

**Prof. Dr. Cleisson de Castro Melo (Membro externo - UFCG).**



---

**Prof. Dr Carlos Eduardo Coelho Freire Batista (PPGCCA/UFPB)**

*Dedico este trabalho a minha mãe,  
Marta Lúcia e a minha noiva  
Sandyerlly da Paz, pelo incentivo.*

## **AGRADECIMENTOS**

Este trabalho marca o fim de uma jornada de dois anos e uma das fases mais importantes da minha vida. Vou aproveitar este espaço para agradecer a todos que contribuíram nesta jornada, em especial a Deus, pela minha vida.

Agradeço ao meu orientador, Alexandre Magno, pela confiança, dedicação, paciência e por compartilhar suas experiências e ensinamentos que foram de grande importância para mim e para a realização deste trabalho.

Aos membros da banca, por suas contribuições, Carlos Eduardo e Cleisson de Castro.

A Professora Alana Marques que contribuiu com seu conhecimento e experiência acadêmica antes e durante a minha pós-graduação.

Minha cunhada Sanierlly da Paz pela motivação e apoio na etapa final deste estudo.

## RESUMO

O surgimento de novas tecnologias de automatização de ferramentas tem crescido em diversas áreas e, apesar de o processo de ensino-aprendizagem da linguagem de programação ter se tornando algo relativamente comum, existe a deficiência de formar profissionais de diferentes áreas habilitados nessa linguagem. As soluções aos entraves para a formação desses profissionais consistem em atrelar suas necessidades e mostrar a integração de diferentes disciplinas à programação de computadores, além da gamificação dos processos enquanto elemento motivador para o aprendizado de programação. Considerando a importância da programação nos tempos atuais, este trabalho objetiva planejar e construir uma ferramenta gamificada, denominada “*MusicMeaning*”, que se propõe a introduzir o ensino de linguagem de programação para músicos ou estudantes de música, relacionando as áreas de música e programação por meio da Teoria da Aprendizagem Significativa. O percurso metodológico adotado foi: (i) analisar os aplicativos selecionados no levantamento sistemático e avaliá-los em relação às seguintes características: público-alvo e conteúdo ensinado; (ii) realizar uma pesquisa exploratória para identificar aplicações computacionais desenvolvidas com o propósito de apoiar o ensino de programação; (iii) selecionar os conceitos de música e programação, contemplando a elaboração da proposta baseada na aprendizagem significativa; (iv) planejar e construir a ferramenta *MusicMeaning*, com base nos itens anteriores. Por fim, o desenvolvimento desta pesquisa colaborou com as áreas da música e programação durante o processo de implementação da ferramenta gamificada: *MusicMeaning*.

Palavras-chave: Linguagem de programação, Música, Aprendizagem Significativa, Gamificação

## **ABSTRACT**

The emerging new automation technologies have grown in several areas. Despite that, the process of teaching and learning programming languages has become something commonplace. But there is a deficiency in training qualified professionals in this field. The solutions to the obstacles on providing knowledge for these professionals consist of harnessing the professional needs of different areas and showing the integration of distinct disciplines with computer programming. In addition to this, the gamification of processes is a motivating element for learning programming. Based on the solutions proposed by the literature and the importance of programming nowadays, this research has created a gamified tool, entitled MusicMeaning. It aims to introduce the teaching of programming language to musicians or students of music. It seeks to relate the areas of music and programming through Ausubel's Subsumption Theory. The research included: (i) To analyze the applications selected in the bibliographic survey and evaluate them concerning both target audience and content taught. (ii) To conduct exploratory research to identify computational applications developed focusing on the teaching of programming. (iii) Selecting the concepts of music and programming, contemplating the elaboration of the proposal based on meaningful learning. In this way, a musician does not need to be familiar with the jargon of the programming field since the programming contents will be related to the music, facilitating the understanding. (iv) To plan and build the MusicMeaning tool. In this way, it will serve as an example for the potential learner (the musician). Lastly, the development of this research colluded with areas of music and programming during the process of implementing the gaming tool: MusicMeaning.

**Keywords:** Programming language, Music. Subsumption Theory

## LISTA DE ILUSTRAÇÕES

Figura 1- Exemplo de download no LilyBin .....	21
Figura 2 - Primeiro exemplo do <i>LilyPond</i> .....	23
Figura 3 - Erro de sintaxe no LilyBin.....	24
Figura 4 - Modelo da assimilação.....	35
Figura 5 - Organograma do menu principal da ferramenta .....	37
Figura 6 - Diagrama de caso de uso do <i>MusicMeaning</i> .....	38
Figura 7 – Fluxograma de navegação do <i>MusicMeaning</i> .....	39
Figura 8 - Dificuldade no entendimento de partituras .....	49
Figura 9 - Dificuldade no entendimento de programação .....	50
Figura 10 - Exemplo de melodia simples em compasso quaternário .....	50
Figura 11 - Exemplo de melodia maior em compasso ternário .....	51
Figura 12 - Exemplo de duas melodias simultâneas.....	52
Figura 13 - Tela Inicial (Versão 1) .....	51
Figura 14 - Tela Inicial (Versão 2) .....	54
Figura 15 - Exemplo de reprodução sonora (Versão 2).....	55
Figura 16 - MusicMeaning Quiz (Versão 2) .....	56
Figura 17 - Telas dos resultados das questões (Versão 2).....	56
Figura 18 - Tela de Resultados (Versão2).....	57

## LISTA DE TABELAS

Tabela 1 – <i>Hello World</i> em <i>LilyPond</i> .....	22
Tabela 2 - Notas no <i>LilyPond</i> .....	22
Tabela 3 - Características da ferramenta .....	27
Tabela 4 - Conceitos selecionados.....	46

## LISTA DE GRÁFICOS

Gráfico 1 - Principais temáticas.....	40
---------------------------------------	----

## LISTA DE ABREVIATURAS E SIGLAS

ACM	Association for Computing Machinery
CAFe	Comunidade Acadêmica Federada
CBIE	Congresso Brasileiro de Informática na Educação
CPU	Unidade Central de Processamento
GNU	General Public License
GTK	Kit de ferramentas do GIMP
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
IEEE	Institute of Electrical and Electronic Engineers
MIDI	Musical Instrument Digital Interface
PDF	Portable Document Format
PPGCCA	Programa de Pós-Graduação em Computação, Comunicação e Artes
RBIE	Revista Brasileira de Informática na Educação
RS	Revisão Sistemática
SBIE	Simpósio Brasileiro de Informática na Educação
TAS	Teoria da Aprendizagem Significativa
TIC	Tecnologias de Informação e Comunicação
UML	Unified Modeling Language

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>15</b>
<b>1.1 Problema de pesquisa .....</b>	<b>15</b>
<b>1.2 Objetivos.....</b>	<b>16</b>
<b>1.3 Justificativa .....</b>	<b>17</b>
<b>1.4 Estrutura do trabalho .....</b>	<b>17</b>
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>19</b>
<b>2.1 Linguagem Lilypond .....</b>	<b>20</b>
<b>2.2 Ensino de música e tecnologia .....</b>	<b>24</b>
2.2.1 Gamificação.....	25
2.2.2 Características da gamificação .....	26
<b>2.3 Teoria da Aprendizagem Significativa (TAS).....</b>	<b>28</b>
<b>3 PERCURSO METODOLÓGICO.....</b>	<b>31</b>
<b>3.1 Estado da arte .....</b>	<b>31</b>
3.1.1 Definição da busca.....	31
3.1.1.1 Termos e análise dos periódicos .....	32
3.1.1.2 Critério de inclusão e exclusão .....	32
3.1.1.3 Resultados.....	33
<b>3.2 Elaboração do glossário .....</b>	<b>33</b>
<b>3.3 Desenvolvimento e avaliação da ferramenta gamificada .....</b>	<b>34</b>
3.3.1 Aplicação da TAS no MusicMeaning .....	35
3.3.2 Exemplos da aplicação dos conceitos selecionados .....	36
3.3.3 Modelagem do MusicMeaning.....	37
<b>4 ESTADO DA ARTE .....</b>	<b>40</b>
<b>4.1 Jogos sérios na música.....</b>	<b>40</b>
<b>4.2 Ferramentas utilizadas no ensino e aprendizagem de programação.....</b>	<b>43</b>
<b>5 RESULTADOS E DISCUSSÕES.....</b>	<b>46</b>
<b>5.1 Elaboração do glossário .....</b>	<b>46</b>
<b>5.2 Exemplos da aplicação dos conceitos.....</b>	<b>50</b>
<b>5.3 Plataforma gamificada .....</b>	<b>53</b>
<b>6 CONSIDERAÇÕES FINAIS.....</b>	<b>58</b>

<b>6.1 Recomendações para estudos futuros.....</b>	<b>58</b>
<b>REFERÊNCIAS.....</b>	<b>60</b>
<b>APÊNDICES .....</b>	<b>63</b>
APÊNDICE A - LISTA DE CÓDIGOS BÁSICOS .....	63
APÊNDICE B - QUESTIONÁRIO.....	65
APÊNDICE C – PRIMEIRA VERSÃO.....	66
APÊNDICE D – SEGUNDA VERSÃO .....	72

## 1 INTRODUÇÃO

Os estudantes e profissionais, hoje, lançam mão da tecnologia para facilitar seu aprendizado ou tornar o seu serviço mais atrativo e dinâmico. É nesse contexto de uso contínuo e cada vez maior da internet, que se dão os avanços das Tecnologias de Informação e Comunicação (TICs). De acordo com Coelho (1986), essas tecnologias se referem a três domínios distintos e interligados, quais sejam: (i) o processamento, armazenamento e pesquisa de informação realizados pelo computador; (ii) o controle e automatização de máquinas, ferramentas e processos; e (iii) a comunicação, nomeadamente a transmissão e circulação da informação.

No que tange ao presente estudo, o interesse quanto à tecnologia da informação reside especificamente no item (ii). Isso porque é o surgimento de novas tecnologias de automatização de ferramentas e processos que propicia a utilização de tais ferramentas em âmbitos diversos.

Diante desse contexto, a demanda pela criação de novas ferramentas e tecnologias de automatização, tem sido cada vez mais incisiva em várias áreas do saber. Essa criação, no entanto, depende de sequências de instruções de programação, conhecimento este, que está relacionado às dificuldades encontradas no processo de ensino-aprendizagem de programação.

### 1.1 Problema de pesquisa

Apesar do processo de ensino-aprendizagem da linguagem de programação ter se tornando algo relativamente comum, devido à necessidade de diversas áreas de terem suas próprias ferramentas digitais; de modo simultâneo, existe a deficiência de formar profissionais habilitados nessa linguagem, tendo em vista que, a falta de motivação e a dificuldade de compreender e aplicar os conceitos aprendidos podem levar a desistência dos cursos (SOUZA, BATISTA; BARBOSA, 2016).

Uma das soluções para evitar essa desistência, segundo Zaina (2005) é atrelar as necessidades profissionais de diversas áreas e mostrar a integração de diferentes disciplinas à programação de computadores.

Outro entrave para a continuidade dos alunos no curso está no ritmo de aula imposto por professores, que desconsideram a existência da variação de estruturas cognitivas<sup>1</sup> existentes numa

---

<sup>1</sup> Estruturas cognitivas são elementos próprios de determinado domínio que fazem com que cada indivíduo absorva e processe informações de forma diferenciada em relação a indivíduos não especialistas no assunto.

mesma sala de aula. Em relação a essa dificuldade, a utilização de abordagens envolvendo jogos digitais e gamificação, enquanto elementos motivadores para o aprendizado de programação, vem sendo adotada com relativo sucesso (BLATT, BECKER; FERREIRA, 2017).

O processo de ensino-aprendizagem de programação por meio de ferramentas gamificadas é uma abordagem que soluciona a falta de interesse em programar, principalmente por ter como características momentos de descontração, atividades lúdicas/interativas com o intuito de estimular o aprendizado, melhorar o rendimento acerca dos assuntos abordados e conseqüentemente a absorção do conteúdo (MEDEIROS et al, 2013).

Segundo Silla (2018), a utilização do ensino de música relacionado ao de programação se mostrou um elemento crucial para o desenvolvimento de jogos para área da música. Nessa proposta a metodologia tinha como público-alvo meninas que aprenderam a tocar instrumentos e posteriormente a programar suas próprias soluções em formato de jogos como resultado final.

Embora o mapeamento sistemático de Blatt, Becker; Ferreira (2017) e a revisão sistemática de Medeiros (2013) tenham como um dos resultados o uso de ferramentas gamificadas ou jogos sérios como abordagem com relativo sucesso para o ensino de programação; e algumas pesquisas como a de Silla (2018) que atrela o ensino de programação a área da música, observamos que nenhuma das propostas apresentou uma ferramenta gamificada para auxiliar o ensino de programação por meio dos conhecimentos sobre música, principal propósito do presente estudo.

## 1.2 Objetivos

Este trabalho tem por objetivo geral planejar e construir uma ferramenta gamificada, denominada “*MusicMeaning*”, que se propõe a introduzir o ensino de linguagem de programação para alunos e professores de música.

Considerando que a inter-relação das áreas de música e programação por meio da Teoria da Aprendizagem Significativa (TAS) pode proporcionar um suporte para elaboração de soluções para problemas por partes de professores e alunos de música, além de promover formas de aprendizagem mais eficientes por meio da aplicação da tecnologia no ensino de música, pretende-se:

- Analisar os aplicativos selecionados no levantamento sistemático e avaliá-los em relação às seguintes características: público-alvo e conteúdo ensinado;

- Realizar uma pesquisa exploratória para identificar aplicações computacionais desenvolvidas com o propósito de apoiar o ensino de programação;
- Selecionar os conceitos de música e programação, contemplando a elaboração da proposta baseada na TAS;
- Planejar e construir a ferramenta gamificada: *MusicMeaning*;

### **1.3 Justificativa**

A inquietação em pesquisar sobre a temática surgiu no ano de 2019 ao associar a carga de conhecimento sobre programação e o interesse de conhecer a área da música. Seguida da participação no grupo de pesquisa de Computação Musical do Programa de Pós-Graduação em Computação, Comunicação e Artes (PPGCCA), sob a orientação do Dr. Alexandre Magno e Silva Ferreira.

Além das aulas de solfejo e programação musical com o Dr Alexandre Magno e Dr Carlos Eduardo, durante a participação no grupo de pesquisa e na disciplina, o trabalho do discente Lucas Blatt foi um dos fatores para escolha do tema de pesquisa, que relaciona as duas áreas: a programação de computadores e a música, tema do presente estudo, tendo como ponto de partida necessidade de o músico aprender a programar para desenvolver suas próprias ferramentas, tornando-se um profissional completo e autônomo associada; a escassez de ferramentas no mercado que atendam, em amplitude, a esse objetivo.

O músico com o intuito evoluir seus conhecimentos mediante ao avanço das tecnologias no mercado, necessita aprender a ler e escrever programas de computador tanto quanto aprender sobre notação musical, em especial é possível traçar uma semelhança entre a programação e notação musical, relacionados no presente trabalho (MOORE, 1995).

### **1.4 Estrutura do trabalho**

Os capítulos da presente dissertação foram divididos em conformidade com o andamento da pesquisa que buscou atender cada etapa do processo de construção de uma ferramenta gamificada estruturada para o objetivo: possibilitar que os músicos aprendam a programar e criem suas próprias ferramentas tecnológicas, associando a linguagem, conhecimentos e serviços relacionados à música na sociedade.

O segundo capítulo apresenta o referencial teórico com os principais fundamentos acerca da computação, música e gamificação, além das relações práticas por meio da linguagem *LilyPond*. Além disso, é nesta seção que será discutida a TAS, subsunçor<sup>2</sup> e o novo conhecimento. O terceiro capítulo trata do percurso metodológico utilizada no presente estudo, descrevendo o estado da arte, elaboração do glossário, criação dos exemplos práticos e a construção da ferramenta gamificada.

O quarto capítulo apresenta o estado da arte do ensino de música por meio de jogos sérios e as ferramentas utilizadas no ensino-aprendizagem de programação. Neste capítulo, analisa-se os trabalhos publicados entre 2005 e 2019, de acordo com as bases de dados e os termos de execução da Revisão Sistemática (RS) escolhidos.

O quinto capítulo destaca os resultados e discussões acerca da elaboração do glossário, exemplos da sua utilização na música e programação e a construção da ferramenta com base no percurso metodológico. O sexto capítulo, por fim, trata da importância de todo o processo deste estudo, trazendo as considerações finais sobre a ferramenta e suas principais contribuições.

---

<sup>2</sup> No método de Ausubel, como será explicitado no capítulo que aborda a Teoria da Aprendizagem Significativa, o conceito de subsunçor diz respeito ao conhecimento que o indivíduo já possui previamente e que agrega, portanto, proposições estáveis no indivíduo.

## 2 REFERENCIAL TEÓRICO

Historicamente, os computadores vieram facilitando algumas atividades que eram realizadas pelos humanos, sendo, atualmente, parte fundamental no processo de desenvolvimento humano, estabelecimentos comerciais, governamentais e de pesquisa. O termo computador começou a ganhar força em meados de 1920, sendo referido a qualquer máquina que calculasse de acordo com parâmetros mais complexos.

No período de 1946 a 1954, foram surgindo os primeiros computadores conhecidos como *Mainframe Computer*, uma máquina que usava a tecnologia de válvulas para processar um grande volume de dados. Partindo do primeiro computador lançado, o ENIAC, que executava sequências complexas de operações, incluindo loops, ramificações e sub-rotinas reprogramadas (FARIAS, 2013), os computadores foram se desenvolvendo, através de gerações, apresentando evoluções que demonstravam (ou marcaram) a transição entre os produtos. Cita-se a substituição do tubo de vácuo por transistores e o surgimento dos conceitos de unidade central de processamento (CPU), memória e instruções transmitidas por meio de linguagem de programação (BHATTACHARYA, 2010), na segunda geração; os microchips, na terceira geração e o surgimento dos mini computadores pessoais, nas quarta e quinta gerações. Além disso, houve o surgimento de sistemas operacionais e uma grande evolução nas linguagens de programação para linguagem de alto nível, chegando a ficar cada vez mais próxima da linguagem humana.

A linguagem de programação é formada por um conjunto de regras, sintaxe, instruções ou passos que formam um algoritmo. Por exemplo, um programa que execute a soma de dois números, a descrição de como será feito a soma é o que chamamos de algoritmo de programação. Além disso, ao final de cada instrução podemos produzir vários tipos de saídas.

Hoje em dia, computadores operam sob o controle dos sistemas operacionais e programas contendo instruções codificadas, chamadas código fonte. As instruções são responsáveis por executar uma série de alterações no computador. Entretanto, o código fonte não é facilmente compreendido pelos seres humanos. Os avanços permitiram criar códigos em uma linguagem de programação de alto nível, como *basic*, *java*, *python*, *perl*, *pascal*, *C*, *C++* ou similares, muito mais fáceis de entender.

As linguagens de alto nível geralmente contém uma sintaxe precisa, mais próxima do programador do que do computador e as suas estruturas permitem saber o significado das palavras que estão sendo declaradas pela linguagem. No processo de criação de um algoritmo, alguns

programas ajudam na hora de escrever as instruções com sugestões e possuem um tradutor ou compilador que traduzem as instruções escritas em linguagem de alto nível para um sistema.

O desenvolvimento de um sistema é trabalhoso, para muitos programadores, o tempo para estudar e escrever um programa são as considerações mais críticas, por mais que seja uma linguagem de alto nível, o programador necessita escrever códigos em um curto período e, ao mesmo tempo, entender como funcionam outras áreas, caso o sistema seja multidisciplinar.

O processo de criação de um sistema que crie, realize a leitura de partituras musicais e mostre sua representação sonora, é um exemplo que requer do programador um conhecimento de linguagens de programação e da área de música.

## 2.1 Linguagem Lilypond

Além da relação dos conceitos, podemos utilizar a linguagem *LilyPond* para ter uma prática de como devemos organizar uma estrutura de programação, com uma sintaxe, conforme demonstrado por Hugo Ribeiro<sup>3</sup>, os exemplos nessa linguagem possuem características das duas áreas.

O *LilyPond* é um sistema de código aberto com uma linguagem semelhante ao LaTeX, permite que programadores altere o código fonte, desenvolva melhorias dentro do próprio sistema ou crie softwares que traduzam a linguagem para partitura musicais<sup>4</sup>. Além disso, possui semelhança com linguagem de programação, diferente do *Encode*, *Muscore*, *Finale* ou *Sibelius* que criam partituras gráficas.

Os principais softwares que utilizam a linguagem *LilyPond* são *Frescobaldi*, um editor de partituras leve e fácil de usar além de ser um software livre, disponível sob a GNU *General Public License*, projetado para rodar em todos os principais sistemas operacionais (GNU/Linux, Mac OS X e Microsoft Windows) e seu nome é derivado de Girolamo Frescobaldi, um compositor italiano do final da Renascença e início do período Barroco. Outro software é o *Denemo*, com uma interface gráfica livre para notação musical, desenvolvido em 1999 usando Kit de ferramentas do GIMP (GTK)+ 2 ou 3 e funciona no Linux, no Microsoft Windows e no Mac OS X.

---

<sup>3</sup> Recurso online disponível em <http://docplayer.com.br/22546029-Introducao-ao-lilypond.html>

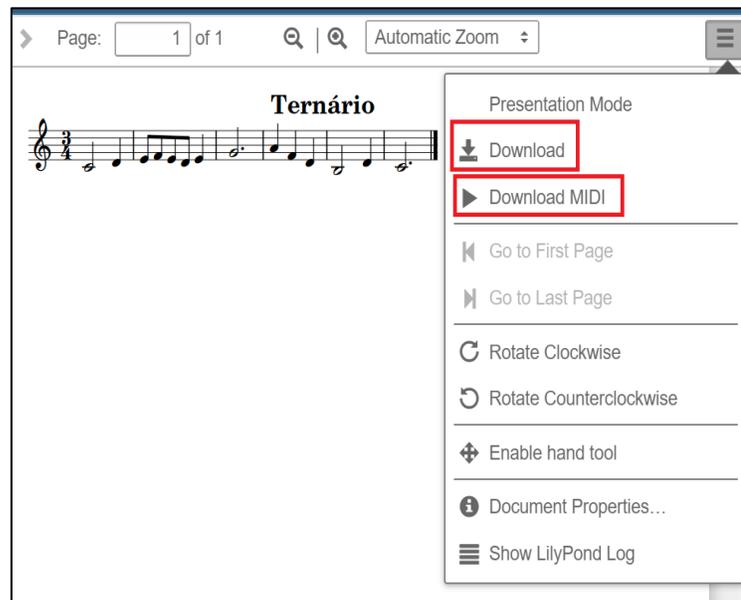
<sup>4</sup> Recurso online disponível em <http://lilypond.org/doc/v2.18/Documentation/notation/writing-pitches>

Algumas das possibilidades de utilizar o *LilyPond* online é através do *LilyBin* e *Hacklily*, editores de partituras que funcionam na internet. O princípio de funcionamento segue as regras dos outros softwares baseados em *LilyPond*, onde as regras de sintaxe precisam ser respeitadas. No *LilyBin* uma das desvantagens é que não temos o auto-complemento dos códigos.

O *Hacklily* é um projeto mais novo e possui recurso de *debug* automático com auto-complemento. Além disso, podemos acessar e baixar os arquivos *Musical Instrument Digital Interface* (MIDI) e *Portable Document Format* (PDF) de qualquer um dos softwares.

O *LilyBin* e o *Hacklily* permite realizar o download do MIDI se for inserido a função `\midi {}` dentro da função `\score {}`. Caso tenha digitado corretamente as funções `\score`, `\midi` e `\layout` para realizar o download, basta clicar nos três traços no canto direito da tela na parte superior, conforme a Figura 1.

Figura 1-Exemplo de download no *LilyBin*



Fonte: O autor, 2020.

Comparando a linguagem *LilyPond* a com linguagem de programação seria possível criar um *Hello World* a partir da função `\markup {Hello World}`. Essa é a primeira frase que todo programador experimenta como um símbolo de sucesso na criação do seu primeiro código, pelo fato de exibir como resultado a cópia do texto, conforme a tabela 1.

Tabela 1 – *Hello World* em *LilyPond*

Sintaxe	Resultado
<pre>\version "2.22.0"  \language "english"  \markup {Hello World!!}</pre>	<p>Hello World</p>

Fonte: O autor, 2020.

Além do mais, é uma linguagem que possui características parecidas com uma linguagem de programação, como exemplo, o *case-sensitive* e a utilização de sintaxe. Um exemplo de sintaxe são os códigos disponíveis no apêndice A e as notas musicais:

Tabela 2 - Notas no *LilyPond*

Nota	<i>LilyPond</i>
<i>do</i>	c
<i>re</i>	d
<i>mi</i>	e
<i>fa</i>	f
<i>sol</i>	g
<i>la</i>	a
<i>si</i>	b

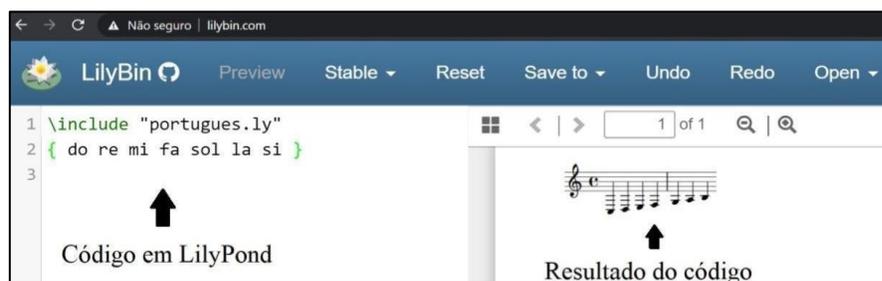
Fonte: O autor, 2020.

Para criar uma partitura escrita em *LilyPond* é necessário observar qual letra representa qual nota, conforme a tabela acima. No entanto, é possível definir as notas de acordo com seu idioma, para isso, é necessário utilizar o comando `\include` mais o nome da biblioteca “portugues.ly”, ou seja, antes de criar nosso primeiro código será necessário utilizar uma biblioteca para traduzir as notas musicais.

Essa tradução facilita o entendimento do código e ajuda na hora de relacionar os dois conteúdos, tendo em vista que à estrutura cognitiva dos músicos são compostas pelas notas musicais: dó, ré, mi, fá, sol, lá e si.

O comando chamando include serve para incluir bibliotecas, a inclusão deve ser realizada utilizando-se do comando `\include` mais o nome da biblioteca. Após a tradução das notas, linha deve conter os nomes das notas dentro de chaves `{}`. Exemplo: Primeira linha: `\include "portugues.ly"`, segunda linha: `{do re mi fa sol la si}`, conforme o *LilyBin*:

Figura 2 - Primeiro exemplo do *LilyPond*



Fonte: O autor, 2020.

A partitura acima escrita em *LilyPond*, utiliza uma biblioteca para traduzir, ou seja, houve uma mudança na sintaxe e as notas passaram a ser escritas em português. As ordens dos elementos `\include "português.ly"` e `{do re mi fa sol la si}` formam o passo a passo que podemos chamar de algoritmo (Figura 2).

Para visualizar uma partitura no *LilyBin* é preciso clicar no botão *PREVIEW* presente na parte superior da tela (Figura 2). Espera-se como resultado do código as linhas do pentagrama, clave de Sol, compasso quaternário, figuras rítmicas em semínima e o início da melodia no dó<sup>2</sup>.

Um detalhe importante da sintaxe ao usar dois pentagramas é lembrar de iniciar com `<<` depois do `\score` {e fechar com `>>` antes do `\midi {}`, para que não apresente erro na execução do código. Não escrever duas notas sem dar espaços, lembrar de utilizar o `\include "português.ly"` para traduzir as notas e sempre que abrir chaves `{}` lembrar de fechar. Nunca colocar as notas em maiúsculo pois o *LilyPond* é *case-sensitive* e usa a barra invertida `"\"` no início de cada comando.

Ordem por prioridade na escrita das notas: Nota + Alteração + Oitava + Figura Rítmica, exemplo na prática: `do + s [sustenido] + '[oitava acima] + 2` (número da figura: mínima), resultado do código: `(dos'2)`. Se houver mudança na ordem será apresentado uma mensagem de erro de sintaxe no *LilyBin* (Figura 3).

Figura 3 - Erro de sintaxe no *LilyBin*


```

Error
Command failed: /bin/sh -c lilypond --formats=pdf --include="/var/task/lib/fonts/font-stylesheet:
Processing `input.ly`
Parsing...
input.ly:4:6: error: syntax error, unexpected ''
{do2s
  '}
input.ly:4:1: error: errors found, ignoring music expression

{do2s'}
fatal error: failed files: "input.ly"

```

Fonte: O autor, 2020.

A Linguagem *LilyPond* possui em sua sintaxe elementos da música e características de programação que podem ser utilizados como exemplos práticos para ensinar programação para músicos. Com isso, há uma necessidade de utilizar TAS como recurso para realização de uma analogia entre linguagem de programação e a linguagem *LilyPond*, por intermédio da ferramenta gamificada, proposta do presente estudo. Portanto, para tornar o desenvolvimento mais fácil e conveniente, a próxima seção apresenta as relações do ensino de música e tecnologia presentes na literatura.

## 2.2 Ensino de música e tecnologia

A música e a tecnologia unidas são fundamentais para o ensino, principalmente na aplicação de ferramentas interdisciplinares. Isso porque

[...] a tecnologia exige novas abordagens, que resultam em novas linguagens artísticas. E o grande denominador tecnológico existente na atualidade são os computadores, que vêm provocando uma verdadeira revolução no campo das artes, alterando todo o modo tradicional de produção artística e provocando o surgimento de novas caminhos (PAIVA, 1992, p. 8).

A relação interdisciplinar entre as áreas se dá também pelo compartilhamento de recursos

[...] a tecnologia adquirida ao longo dos últimos séculos possibilitou inúmeros avanços e melhoramentos na produção musical, seja na construção de instrumentos, seja na arte da composição e até no desenvolvimento de suportes para a fixação e memorização da música (ZUBEN, 2004, p 7)

De acordo com Shea, Pickett e Pelz (2003), um ambiente de aprendizagem online eficaz estimula o contato entre alunos e membros do corpo docente, incentiva a correlação e cooperação entre os pares (*peer instruction* em inglês americano), proporciona um *feedback* imediato, estimula a comunicação e melhora o respeito à diversidade e formas de aprendizagem de cada aluno.

Portanto, as instituições de ensino buscam, cada vez mais, mecanismos para uma educação intuitiva e informatizada, como o ensino de música através de aplicativos e softwares educacionais.

Na Educação Musical, tem-se discutido sobre a utilização da tecnologia como método de ensino-aprendizagem. Segundo Vieira (2010, p.21), “em meio a sistemas computadorizados, o músico, a arte, a apreciação e o ensino de música ganham novas dimensões e novos mecanismos de apreensão, testemunhando quebra de paradigmas e rompendo fronteiras sejam territoriais, ideológicas ou culturais”.

Outro exemplo do uso de ferramentas tecnológicas é a possível interação dos alunos de percepção musical com softwares que reproduzem sons, essas ferramentas podem, inclusive, auxiliar docentes de outras áreas no entendimento de percepção musical. Neste meio, tem sido observada uma compreensão maior, tanto pelos discentes quanto pelos docentes, na interação com elementos musicais (CHAMORRO et al., 2017).

Existem ferramentas para a manipulação sonora e perceptiva, atividades composicionais e acústicas, dentre outras. Alguns editores de partituras aperfeiçoam o processo de composição e digitalização de partituras, como *Frescobaldi/Lilypond*, e possibilitam codificações gráficas através de símbolos específicos da notação musical, reproduzindo informações de mídias gráficas e sonoras. Além disso, esses processos podem ser relacionados com linguagem de programação, servindo de base para alunos e professores de outros cursos que queiram aprender sobre as áreas de música e programação.

### 2.2.1 Gamificação

Com a prática de inclusão digital em escolas e universidades, o ensino de computação por meio de jogos ou ferramentas gamificadas vem se tornando cada vez mais comum por possibilitar uma melhor qualidade na eficiência instrucional.

Além dos avanços tecnológicos que vêm permitindo uma rápida evolução na qualidade dos jogos sérios ou ferramentas gamificadas, observa-se também uma relação interdisciplinar quanto ao uso dessas aplicações para ensinar conteúdos de diversas áreas.

Segundo Nascimento, Silva e Lemke (2017), dentre os conteúdos abordados por meio dos jogos, estão sua aplicação multidisciplinar com 76%, em seguida ciências biológicas com 12% e linguísticas, letras e artes também com 12%. Outro levantamento importante apresentado pelos autores são os gêneros encontrados, por exemplo, lógica e raciocínio presentes em 82% dos jogos.

Aborda-se, neste trabalho, os jogos educacionais chamados de jogos sérios e foca nos aspectos de gamificação, fazendo relação interdisciplinar com a área, por meio da TAS, e, por consequência, unindo características de ensino em uma ferramenta gamificada digital.

O conceito de gamificação adotado, refere-se à técnica que permite inserir em diferentes áreas e campos de conhecimento partes de um jogo, sobretudo no que se refere à aprendizagem por meio da gamificação, de forma que o assunto se torna mais agradável do que abordado de maneira tradicional.

O ensino tradicional, mesmo que na sua fase inicial, tem se modificado a ponto de mudar sua abordagem verbal para uma maneira mais dinâmica, onde foram adicionados novos elementos tecnológicos proporcionando novas experiências pedagógicas. Um dos elementos é a gamificação, na qual adaptam diversas situações com objetivo de despertar sentimentos, descobrir aptidões ou até mesmo atribuir recompensas virtuais ou físicas diante da conclusão de uma atividade.

A gamificação necessariamente não é um jogo, mas sim um elemento que possui aspectos utilizados em jogos, pontuação, premiação, tempo entre outros. Os jogos têm regras e objetivos que não possuem relação com a vida humana, ao contrário a gamificação possui relação e permite resolução de problemas. Dessa forma, torna-se importante entender as características que podem contribuir para a construção da ferramenta gamificada, aqui proposta.

### 2.2.2 Características da gamificação

Embora seja uma área nova e relativamente incipiente, a aplicação da gamificação enquanto método para a transformação de processos de ensino e aprendizagem já conta com algumas indicações-base, construídas a partir de experiências de boas práticas em casos de excelência dessa aplicação, a partir de fundamentos teóricos. Destacam-se, assim, as seguintes potencialidades:

- Disponibilizar diferentes experimentações: assim como em grande parte dos jogos sérios, muitas vezes não há um único caminho que conduz ao sucesso. Propiciar diferentes caminhos para alcançar a solução de um problema incorpora diferentes características pessoais de aprendizagem no processo educativo, o que contribui para a experiência educativa de cada um;
- Incluir ciclos rápidos de *feedback*: nos games os jogadores são sempre capazes de visualizar o efeito de suas ações em tempo real. Nas escolas normalmente acontece o inverso e os alunos só conseguem visualizar seus resultados depois de certo tempo, muito maior do que aquele que estão acostumados nos games. Acelerar esse processo de *feedback* estimula a procura por novos caminhos para atingir os objetivos, bem como o redirecionamento de uma estratégia, caso ela não esteja apresentando os resultados esperados [...] (FARDO, 2013, p. 5-6).

As características escolhidas podem ser vistas na Tabela 3, tais como pontuação, regras, condições e entretenimento, se incorporadas ao processo educacional tornarão a forma de ensinar e aprender mais intuitiva e prazerosa.

Tabela 3 - Características da ferramenta

<b>Características da ferramenta</b>	
Pontuação	Características presente em muitos jogos, vale pontuar para ter um bom desempenho
Regras	Normas para serem seguidas. Caso tenha o descumprimento haverá punição.
Condições	Para ter uma boa avaliação é necessário atingir uma nota maior que 7 pontos
Entretenimento	Pode ser considerada uma forma de lazer e divertimento para pessoas que querem aprender a programar.

Fonte: O autor, 2020.

Por todo o exposto, o conceito de gamificação no presente estudo utilizado será aquele também utilizado por Vianna et al. (2013), quando abrange, na gamificação, além da utilização de mecanismos de jogos para a resolução de problemas, seu uso para a motivação e o engajamento de um determinado público. Dessa forma, a gamificação não seria apenas a utilização de elementos de jogos em outros ambientes, mas sim a utilização dos elementos mais eficientes a fim de alcançar os mesmos benefícios que se tem no ato de jogar.

Vianna et al. (2013), portanto, utilizam a gamificação como ferramenta potencializadora para despertar emoções positivas e explorar aptidões do sujeito, atreladas a recompensas dadas quando ele executa determinada tarefa. Essas recompensas compreendem a motivação enquanto articulação das experiências vividas pelos indivíduos com a proposição de novas perspectivas, a fim de gerar prazer e bem-estar.

O processo de gamificação per se e, por conseguinte, o processo de gamificação da aprendizagem só foram possíveis por conta das ferramentas cada vez mais tecnológicas e das linguagens de programação cada vez mais sofisticadas que foram criadas ao longo do tempo.

A criação de ferramentas e linguagens passou por diversas fases, até se tornar o que conhecemos hoje. Essa evolução deve estar sempre atrelada à evolução educacional da música, pois somente assim é possível traçar um território comum e interdisciplinar entre as áreas e é através do compartilhamento de recursos que há a possibilidade de uma evolução analógica.

Tais ferramentas abrangem cada vez mais esses horizontes linguísticos-interdisciplinares para uma construção analítica, que somente é possível através da TAS.

### **2.3 Teoria da Aprendizagem Significativa (TAS)**

A TAS é uma teoria de assimilação, desenvolvida por David Paul Ausubel na década de 80, que propõe um estudo aprofundado da aprendizagem, enquanto um processo que não se resume à simples execução de comandos.

A aprendizagem significativa permite que os alunos melhorem sua capacidade de aprender por meio de estratégias e metodologias que buscam evoluir os níveis de conhecimento, sendo mais efetiva a adaptações que minimizam as dificuldades cognitivas apresentada por parte dos alunos (AUSUBEL et al., 1980).

Esta abordagem considera o conhecimento que o aluno já possui sobre o assunto a ser estudado, tendo em vista que as analogias e relações trazem novas ideias e significados que surgem por meio da associação da sua estrutura cognitiva com um novo conteúdo (RIBEIRO; NUÑEZ, 2004).

Os aspectos importantes sobre as especificidades, demarcadas por Ausubel na TAS, dizem respeito ao fato de que, nos moldes “Ausubelianos,” este “é um processo por meio do qual uma nova informação se relaciona, de maneira substantiva (não-literal) e não-arbitrária, a um aspecto relevante da estrutura cognitiva do indivíduo” (MOREIRA, 1999).

Deste modo, o que se aprende é a relação daquilo que já foi compreendido em outro momento por parte dos alunos, sem que seja preciso absorver o novo assunto apresentado. Outro fato é que a relação de conteúdo não acontece com qualquer aspecto da estrutura cognitiva, mas sim com as informações prévias sobre os conhecimentos de cada aluno, ou seja, se os conceitos e ideias estiverem na estrutura cognitiva dos alunos, as novas ideias e conceitos serão aprendidos com significado (AUSUBEL et al., 1980).

No entanto, para que haja uma aprendizagem significativa, é necessário que algumas condições básicas sejam atendidas. De acordo com Ribeiro e Nuñez (2004), o novo assunto precisa ter um potencial significativo, ou seja, deve haver significado lógico coerente com o material e significado psicológico coerente com a estrutura cognitiva do aluno.

“A emergência do significado psicológico depende não somente da apresentação, ao aprendiz, de um material logicamente significativo, mas também da disponibilidade, por parte

desse aprendiz, do necessário conteúdo ideacional.” (MOREIRA, 1999, p. 22). Portanto, TAS não depende só do conteúdo, mas, principalmente, do interesse do aluno em absorver aquilo que será passado.

Objetivando alcançar um processo o mais eficiente possível, Ausubel et al (1980, p.144) sugerem o uso de organizadores prévios, que são estruturas de introdução ao que se pretende ensinar, interligando o conhecimento que já existe, com o que vai ser apresentado, criando um espaço delimitado e indicativo.

Sabe-se que o conhecimento humano é formado por conteúdos, ideias e assuntos que são adquiridos durante sua vida. As organizações das ideias muitas vezes são associadas e geram novas informações, potencialmente significativas a estrutura cognitiva humana, relacionadas por um conceito de subsunção, através do qual uma nova informação pode se integrar ao cérebro humano.

O conceito subsunção diz respeito ao conhecimento que o indivíduo já possui previamente e que agrega, portanto, proposições estáveis no indivíduo. Essa estabilidade garante ao aprendiz a possibilidade de conhecer ideias novas, adicionando aos seus conhecimentos prévios novas informações.

Portanto, o que se deseja na assimilação é um processo de racionalização não superficial. O aprendiz une o conhecimento prévio ao novo conteúdo, criando um novo subsunção. O conhecimento prévio nunca mais retornará a ser o que era antes – o que é chamado de assimilação obliteradora – e o produto interacional entre a nova ideia modificada e a ideia já existente, também modificada, apresenta um significado próprio que resulta em novos princípios (AUSUBEL et al., 1980).

Assim, o autor sugere a diferenciação progressiva, ou seja, a técnica de apresentar ideias mais gerais e inclusivas de uma matéria, *a priori*, e aprofundar-se nelas depois, o que levaria a um processo chamado reconciliação integrativa. É através dessa técnica que seriam selecionados os melhores materiais e instruções para o aprendiz, de modo a incentivá-lo a pensar, explorar significados e buscar diferenças e similaridades entre os conhecimentos.

A elaboração de novos materiais utilizando a TAS devem conter os conhecimentos existentes e os novos conteúdos. É nesse sentido que é apresentada a proposta de um mapa conceitual, que expõe os conceitos, partindo do mais abrangente, interligando os menos abrangentes, até alcançar os mais singulares.

A teoria dos Mapas Conceituais foi criada por Joseph Novak em 1972 e é uma ferramenta gráfica que serve para representar, organizar, construir e avaliar conhecimentos. A teoria dos mapas conceituais foi desenvolvida tomando por base a teoria da aprendizagem significativa de Ausubel, e possui diversas aplicações para a educação, tais como: a) apresentar um conteúdo; b) estudar um conteúdo; c) fazer síntese de texto; d) organizar o conteúdo programático de uma disciplina; e) avaliar a aprendizagem.

Embora haja contribuições relacionadas a TAS, Joseph (1972), Ausubel et al (1980), Moreira (1999), Ribeiro e Nuñez (2004), não apresentaram em seus estudos ferramentas ou metodologias para verificar a estrutura cognitiva dos alunos. Deste modo, o processo de verificação deve ser reformulado por cada pesquisador mediante a aplicação de suas metodologias.

No próximo capítulo, será apresentado as etapas realizadas no desenvolvimento da pesquisa, como os conceitos foram selecionados, bem como, a modelagem utilizada para criação da ferramenta e a descrição de como foi realizado o estado da arte.

### 3 PERCURSO METODOLÓGICO

A fim de esclarecer a estratégia interdisciplinar que envolve as áreas da música e programação é necessário traçar um percurso metodológico para levantar hipóteses, gerar discussões e formular conclusões, além de aportes conceituais destinados aos estudos das duas áreas, agindo em consonância com os objetivos traçados.

Em relação à natureza da pesquisa, classifica-se como aplicada, pois objetiva gerar novos conhecimentos relacionados ao ensino e aprendizagem de programação de computadores a partir da utilização da ferramenta gamificada.

Os novos conhecimentos de música e programação relacionados por meio da TAS, podem proporcionar um suporte para elaboração de soluções para problemas em sala de aula por partes de alunos e professores de música, além de possibilitar que o músico conheça uma nova área e aprofundar-se em linguagem de programação. Os métodos descritos a seguir estão divididos em três partes: estado da arte, elaboração do glossário e a modelagem da ferramenta gamificada.

#### 3.1 Estado da arte

A revisão sistemática (RS) da literatura é uma metodologia de busca científica que permite identificar, avaliar e interpretar pesquisas disponíveis e relevantes para uma determinada questão de pesquisa ou área temática (Kitchenham, 2004). Este método investigativo requer mais esforço do que investigações tradicionais (Biolchini et al., 2005). Sua principal vantagem é fornecer informações sobre os efeitos de algum fenômeno por meio de uma ampla gama de configurações e métodos empíricos.

Esta metodologia aumenta a probabilidade de detecção de efeitos reais que estudos individuais são incapazes de detectar. Portanto, o objetivo é identificar trabalhos acadêmicos que utilizam jogos sérios no ensino de Música, quais as áreas da Música são mais exploradas por estes softwares e como os jogos sérios são incorporados nas rotinas de ensino-aprendizagem da Música.

##### 3.1.1 Definição da busca

A etapa inicial da formulação da pergunta de pesquisa é fundamental para o planejamento e delineamento dos objetivos da RS. Neste sentido, algumas perguntas de pesquisa podem ser destacadas no presente trabalho, são elas:

- Quais áreas da Música são as mais exploradas nos jogos sérios?

- Como os jogos sérios estão sendo utilizados no ensino de Música?

Para realizar a investigação da produção científica e alcançar resultados significativos, foram escolhidas as seguintes bases de dados: ACM *Digital Library* (ACM), IEEE Explorer (IEEE), Simpósio Brasileiro de Informática na Educação (SBIE), onde estão publicados boa parte dos conteúdos relacionados à Informática na Educação sendo realizado em paralelo com Congresso Brasileiro de Informática na Educação (CBIE), Revista Brasileira de Informática na Educação (RBIE) e a *SCOPUS Elsevier* (SCOPUS) que contém diversas bases incorporadas. Nesse estudo foi definido que seriam analisados somente trabalhos publicados entre 2005 e 2019 nas bases anteriormente citadas. O período escolhido justifica-se pelo baixo número de pesquisas encontradas que tratam do tema.

#### *3.1.1.1 Termos e análise dos periódicos*

Os termos usados para execução da RS nas bases de dados ACM, IEEE, SBIE, RBIE e SCOPUS foram “(jogo musical) OR (jogos de música) OR (ensino de música) OR (music AND Game AND education OR Game AND music AND education)”. Na ACM, IEEE e Scopus apresentam em seus resultados publicações de outras bases não citadas, mas que foram consideradas dada a relevância das publicações.

A análise dos resultados, utilizando a chave de busca definida anteriormente, retornou um total de 353 artigos publicados na base SCOPUS, ACM retornou um total de 86 artigos, 60 artigos resultantes na base IEEE, na SBIE 8 artigos e na RBIE 5 artigos.

#### *3.1.1.2 Critério de inclusão e exclusão*

Os artigos obtidos passaram por um processo de filtragem. Neste processo, são lidos e analisados o resumo, o título e, se necessário, a introdução do trabalho. Caso um dos critérios seja atendido, automaticamente o trabalho é selecionado para a fase de leitura completa. Os critérios definidos para esta pesquisa foram:

Critério 1- Serão incluídos trabalhos que comparem ou avaliem instrumentos teóricos e/ou práticos de jogos sérios para o ensino-aprendizagem de assuntos voltados para áreas de Música.

Critério 2- Serão incluídos trabalhos que apresentem métodos de desenvolvimento de jogos sérios para o ensino-aprendizagem de música.

Critério 3- Serão incluídos trabalhos que proponham algum tipo de design de jogos sérios para ensino de Música.

Critério 4- Serão incluídos artigos que apresentem uma proposta interdisciplinar entre Música e Computação e que discutam sobre o ensinamento da Música.

Critério 5- Serão excluídos trabalhos repetidos que aparecerem em mais de uma base de busca, considerando apenas o resultado mais recente.

Critério 6- Serão excluídos trabalhos no formato de livro, resumos estendidos ou que não tiverem acesso por meio do portal de periódicos da Capes através da plataforma CAFe<sup>5</sup>.

### *3.1.1.3 Resultados*

Após a execução da busca nos periódicos definidos, todos os artigos resultantes passaram pelo processo de triagem e os critérios de inclusão e exclusão foram avaliados. O resultado desta compilação é apresentado no Capítulo 4, juntamente com as abordagens e ferramentas identificadas para o ensino de programação em diversas áreas.

## **3.2 Elaboração do glossário**

Para selecionar novos conceitos de música/programação e adicionar ao glossário em criação, contemplando a elaboração da proposta baseada na TAS, realizou-se uma pesquisa de cunho qualitativo, devido à necessidade do pesquisador se envolver com o assunto e ter a percepção da linguagem musical e de programação.

Tomou-se como base a utilização da apostila de Lucas Blatt (2019), cujo título é “conceitos básicos de programação de computador para músicos.” Na apostila contém a relação de linguagem de programação com a música que serviram como base para criação de um glossário com uma estrutura na qual os novos conceitos são apresentados com base nas aulas tradicionais de linguagem de programação.

Os conceitos selecionados foram utilizados em uma aula experimental, de 2 horas de duração, sobre música e programação, o público-alvo escolhido foram 16 alunos do PPGCCA. Por

---

<sup>5</sup>O acesso CAFe habilita o acesso gratuito a todos os Discentes, Docentes e Técnicos Administrativos ao Portal de Periódico da Capes por meio da Comunidade Acadêmica Federada – CAFe, que permite o acesso dentro ou fora da universidade.

meio do questionário aplicado aos alunos pretendeu-se avaliar o entendimento dos alunos sobre os conceitos apresentados e saber qual foi o entendimento acerca dos conteúdos vistos separadamente.

A aula experimental foi dividida em 3 partes, a primeira teve duração de 45 minutos e o assunto abordado foram os conceitos de teoria musical, na segunda parte foram explicados os conceitos de programação com duração de 45 minutos e por último os conceitos foram relacionados por meio da ferramenta *Frescobaldi/LilyPond* durante os 30 minutos finais.

Após a aula, foi disponibilizado um questionário online (criado e publicado via a ferramenta Typeform) para coletar as opiniões dos alunos utilizando perguntas (Apêndice B) e respostas em formato escalar. A escala de verificação escolhida foi de 1 a 5, com base na escala de Likert,

[...] onde o respondente, em cada questão, diz seu grau de concordância ou discordância sobre algo escolhendo um ponto numa escala com cinco gradações (sendo as mais comuns: concordo muito, concordo, neutro/indiferente, discordo, discordo muito). Em Game Design, tais escalas são uma ferramenta ágil na coleta de avaliações sobre o jogo, seja durante seu processo de produção, coletando avaliações da equipe como forma de identificar pontos a serem melhorados ou corrigidos; seja após o jogo ter sido lançado para o público, como forma de analisar a receptividade do mesmo frente aos jogadores (AGUIAR, Bernardo; CORREIA, Walter; CAMPOS, Fábio, 2011, p.1)

As escalas Likert são normalmente usadas para medir atitudes e opiniões com um maior grau de nuance e suas variações são: 1 - discordo totalmente, 2 - discordo, 3 - neutro, 4 - concordo, 5 - concordo totalmente. As variações indicam um nível maior de satisfação se comparado com perguntas que tenham tipos de respostas “sim ou não”. Os dados coletados servem como *feedback* para utilização dos conceitos na ferramenta gamificada, observados no Capítulo 5.2.

### **3.3 Desenvolvimento e avaliação da ferramenta gamificada**

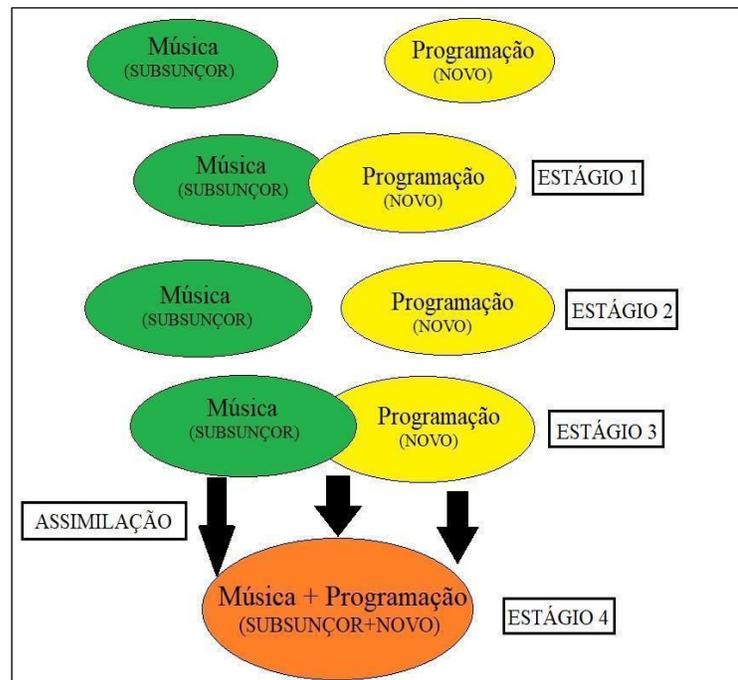
Como mencionado anteriormente, o presente estudo propôs o desenvolvimento de uma ferramenta gamificada interdisciplinar denominada *MusicMeaning*, cujo nome foi atribuído por apresentar significados de conceitos a partir do entendimento sobre a área da música.

Neste capítulo, apresentam-se os estudos que servirão de base para a construção da ferramenta, quais sejam, o modelo de assimilação de Ausubel e os exemplos da linguagem *LilyPond*. A próxima seção tem por fim demonstrar como é possível traçar um território comum e interdisciplinar entre Música e Programação, justificando ainda, de que forma é possível aos músicos aprender a linguagem de programação, tendo por base seu conhecimento musical prévio.

### 3.3.1 Aplicação da TAS no MusicMeaning

A aplicação da TAS se deu por meio do processo de assimilação, iniciado quando um novo conhecimento é apresentado ao aluno. Diante da interação com os conhecimentos prévios, tanto o novo conhecimento quanto os subsunçores sofrem transformações. Para explicar a assimilação da música com a programação foi criado um esquema baseado no modelo da assimilação da TAS (Figura 4).

Figura 4 - Modelo da assimilação



Fonte: O autor, 2020.

As interações com os conhecimentos prévios, tanto programação quanto a música sofrem transformações, gerando um produto interacional: programação modificada + música modificada. O produto assume uma nova organização na estrutura cognitiva, contribuindo com o desenvolvimento de um novo significado, que não se refere somente à música (subsunçor) ou a programação (novo conhecimento), mas à influência mútua entre os dois (Figura 4 – estágio 1).

O resultado do significado da programação modificada pode ser desagregado da sua relação estabelecida com o significado da música (subsunçor) também transformado. Os novos significados ainda guardam consigo algumas de suas características básicas, porém, em processo

de transformação (Figura 4 – estágio 2). Durante o processo de assimilação, os significados não conseguem mais ser dissociados. O significado das novas ideias (programação) é assimilado ou reduzido ao significado das ideias contidas nos subsunçores, no caso a música (Figura 4 – estágio 3).

Deste modo, o significado atribuído a programação torna-se progressivamente menos dissociável das ideias que compõem a música (subsunçor), ou seja, os conceitos não são mais vistos individualmente e não serão esquecidos, pois estão assimilados. O produto interacional composto pela música e a programação são alterados definitivamente e por fim incorporados aos novos significados (Figura 4 – estágio 4). Portanto, o esquecimento dos conhecimentos da programação e permanência de conhecimentos de programação mais música, atribui-se a assimilação obliteradora.

Os conceitos foram selecionados conforme relatado na seção 3.2 e adicionados ao *MusicMeaning*. Além disso, cada conceito terá um exemplo de relação com a música, ou seja, apresentará a assimilação do novo conhecimento (programação) com o conhecimento existente (música). Portanto, será apresentado na próxima seção como os exemplos serão aplicados ao *MusicMeaning*, colocando em prática a assimilação de Ausubel.

### 3.3.2 Exemplos da aplicação dos conceitos selecionados

O aprendizado e a experiência adquirida com a introdução a programação podem servir para aprofundar o conhecimento acerca da linguagem de programação ou até mesmo como uma metodologia educacional, podendo resultar na criação das suas próprias ferramentas para área. Partindo desse pensamento foram adicionados exemplos práticos ao *MusicMeaning* que utilizam a linguagem *LilyPond* para demonstrar a aplicação dos conceitos presentes no glossário.

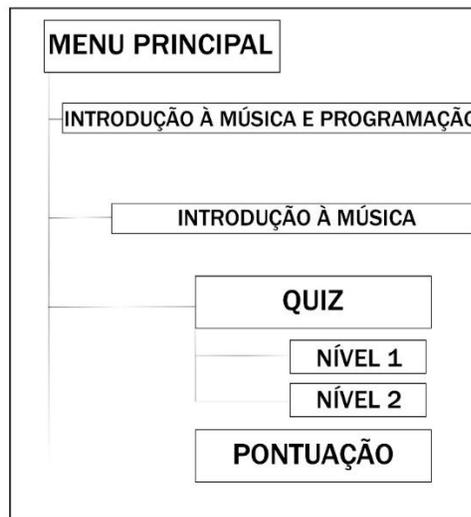
O uso da linguagem de programação *LilyPond* e as combinação de símbolos musicais é uma das possibilidades de aplicar o modelo de assimilação de Ausubel, situação está que será integrada ao *MusicMeaning*, permitindo desenvolvimento de partituras musicais em linha de comando, observado na seção 5.2.

Na próxima seção, serão apresentados os diagramas *Unified Modeling Language* (UML) com a finalidade de descrever o funcionamento da ferramenta e mostrar os requisitos funcionais e não funcionais.

### 3.3.3 Modelagem do MusicMeaning

A partir das informações apresentadas anteriormente, são levantados os requisitos do sistema para criação das modelagens, fornecendo uma projeção de como ficará a ferramenta *MusicMeaning* quando finalizada. Deste modo, pensando na estruturação da ferramenta foi criada a primeira modelagem com dois níveis e uma série de telas (Apêndice C) que auxiliam o usuário a navegar (Figura 5).

Figura 5 - Organograma do menu principal da ferramenta



Fonte: O autor, 2020.

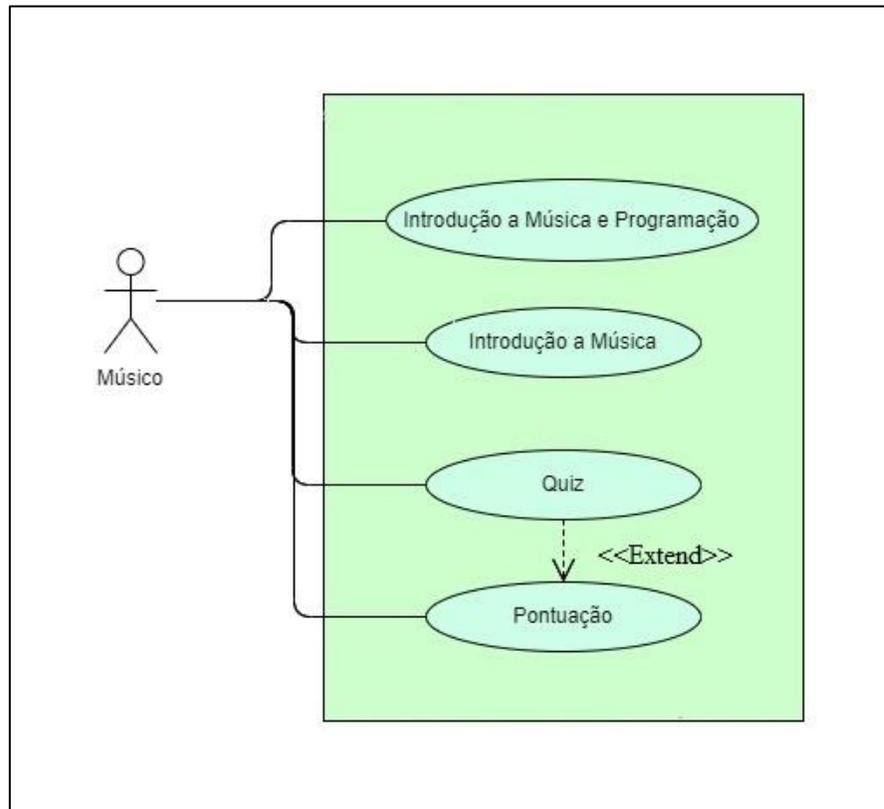
Por meio do menu principal, o usuário pode escolher uma das quatro opções:

- “Introdução à música e programação”, apresenta a relação da música (subsunçor) com a programação (novo conhecimento);
- “Introdução à Música” traz conteúdos sobre música com a intenção de relembrar conhecimentos existentes na estrutura cognitiva do aluno de música e serve como material de apoio para o ensino de música para pessoas que querem conhecer a área da música;
- “Quiz” contém dois níveis: o primeiro nível avalia o conhecimento existente sobre música e o segundo nível apresenta perguntas que relacionam música com programação;
- “Pontuação” possui uma nota, quantidade de questões acertadas e uma mensagem de texto que avalia o usuário de acordo com a nota.

Na Figura 6, é possível observar o diagrama de caso de uso contendo as funcionalidades da ferramenta, o ator e os nomes das interações. O caso de uso descreve o comportamento do

sistema visto pelo usuário apresentando narrativas do uso individual por meio do cenário da ferramenta.

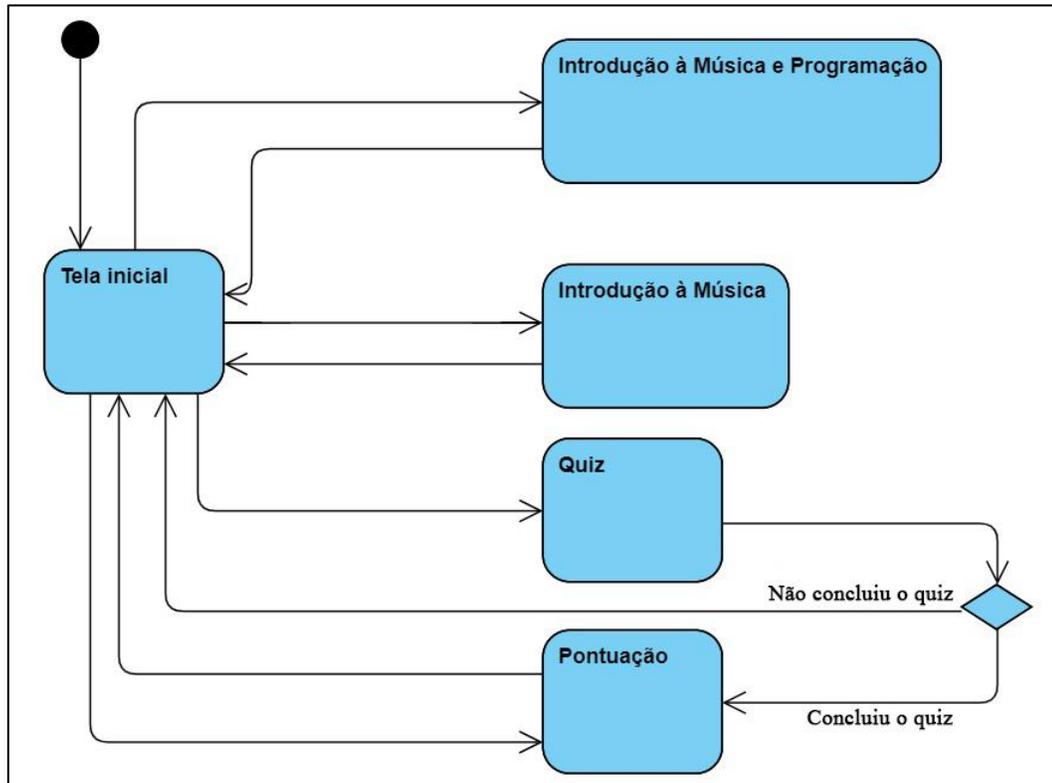
Figura 6 - Diagrama de caso de uso do *MusicMeaning*



Fonte: O autor, 2020.

O diagrama de caso de uso contém os atores envolvidos, interações e tipos de interações, a fim de descrever as interações do sistema (SOMMERVILLE, 2011). Na figura 6, o ator é o músico, seus acessos são a “introdução à música e programação”, “introdução à música”, “quiz” e “pontuação”. O acesso ao “quiz” faz um *extend* no caso de uso “pontuação”, ou seja, sempre que houver acesso ao “quiz” poderá ter uma pontuação ou não. No acesso a terceira opção, “quiz”, a “pontuação” só é visualizada se o músico concluir todo o questionário sem errar duas vezes, se pôr a caso o músico erre a resposta duas vezes o sistema volta pra tela inicial e apresenta uma mensagem para estudar novamente.

Para entender como funcionará o sistema, foi criado um fluxograma de navegação, com o objetivo de apresentar as janelas que compõem o sistema, seus possíveis desvios e visualizar se a sequência faz sentido para o usuário. Também serve como referência para requisitos de desenvolvimento, especificando que telas serão necessárias (Figura 7).

Figura 7 – Fluxograma de navegação do *MusicMeaning*

Fonte: O autor, 2020.

O fluxograma de navegação é composto por uma tela inicial, onde o usuário escolhe uma das opções: “introdução à música e programação”, “introdução à música”, “quiz” ou “pontuação”. A tela do “quiz” é a única que tem uma condição, por exemplo, se o usuário não concluir o “quiz” ele tem a opção de voltar para o menu, caso conclua o “quiz” exibirá a tela de pontuação.

Esse capítulo apresentou a descrição do método para criação do estado da arte que será aprofundado no Capítulo 4, quais materiais foram utilizados na elaboração do glossário e como foi sua validação, além de descrever como seria sua utilização por meio TAS na ferramenta gamificada. Além disso, foram criados exemplos utilizando diagramas para demonstrar o funcionamento da ferramenta.

## 4 ESTADO DA ARTE

A fim de projetar o melhor caminho para o desenvolvimento de uma ferramenta gamificada de ensino musical através da programação, será necessário entender, inicialmente, qual o atual estado da arte desse ensino. Para tanto, realizou-se uma revisão sistemática das técnicas utilizadas em jogos sérios aplicados à música para, ao fim, inferir a melhor forma de traçar um mapa conceitual interdisciplinar entre a música e a programação.

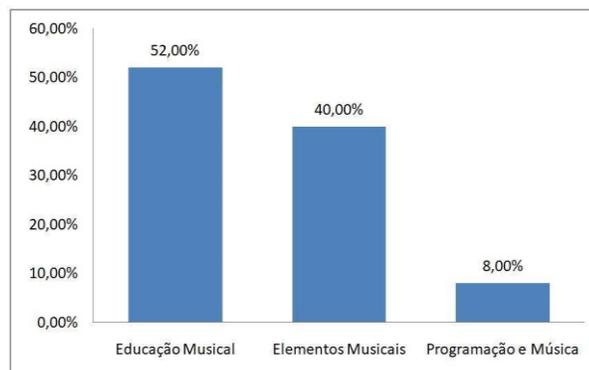
### 4.1 Jogos sérios na música

Na Computação, os jogos sérios podem ser definidos como sistemas computacionais que tenham regras bem definidas, interatividade, gráficos 3D ou 2D, objetivos claros, narrativas construídas em primeira ou terceira pessoa, componentes de Inteligência Artificial (IA) e elementos desafiantes ao jogador, como: pontuação, competição, tempo ou ranking (FREITAS; MORAIS, 2019).

As características, as variáveis de cada estudo e metodologias encontradas são parâmetros considerados relevantes para a comparação dos estudos selecionados (POCINHO, 2008). Neste sentido, destacam-se as principais temáticas abordadas no ensino de música: programação e música, elementos musicais e educação musical, ilustrados pelo Gráfico 1.

Desse modo, é possível perceber que 52% das publicações focam em abordar uma introdução a educação musical por meio de jogos sérios, em seguida, 40% focaram em um escopo voltado para elementos musicais como ritmo, melodia, harmonia, timbre, solfejo e, por fim, 8% dos artigos abordaram assuntos de programação e música (FREITAS; MORAIS, 2019).

Gráfico 1 - Principais temáticas



Fonte: FREITAS; MORAIS, 2019.

Dentre os jogos contemplados nesse estudo encontrou-se: “*Drums Listen And Execute Game*”, desenvolvido para melhorar a percepção musical; “*Keyboard Chord Trainer*”, com objetivo de melhorar a leitura da trilha sonora; “*Guitar Scale Mastery*”, que tem o objetivo de melhorar as habilidades ao tocar guitarra; “*Sight Reading Game*”, com intuito de ajudar pessoas a melhorar suas habilidades de “leitura à primeira vista”<sup>6</sup> com seus instrumentos (FREITAS; MORAIS, 2019).

Outros jogos encontrados foram “*Harmonic Walk*”, para ensinar harmonia tonal e harmonização de melodia; “*Jazz Improvisation*”, para ouvir partituras e recomposição musical; “*Following the Cuckoo Sound*”, para ajudar as crianças cegas a andarem em linha reta por meio do som; “*Game World*”, que ajuda as crianças a aprenderem os conhecimentos básicos sobre as teclas do piano e as notas musicais associadas e “*Flappy Crab*”, que tem como objetivo trabalhar áreas como a memória auditiva e a discriminação de alturas, numa abordagem simultânea à simbologia própria da ortografia musical (FREITAS; MORAIS, 2019).

Os demais jogos que estão relacionados ao estudo são os jogos “*Caklempong*”, que busca preservar a música tradicional *caklempong* na Malásia e ensina a tocar o instrumento canang. O “APP” é um jogo de aprendizagem digital voltado à musicalidade e ritmo; o jogo “*Touching Notes*” baseia-se em gestos que exigem uma movimentação do corpo para tocar as notas musicais; o jogo “*Note Code*” apresenta um quebra-cabeça de programação musical projetado como um dispositivo tangível acoplado a uma interface gráfica do usuário GUI (FREITAS; MORAIS, 2019).

O jogo “E-Cecilia” contém exercícios de canto gerados automaticamente, além de um sistema de avaliação do desempenho do usuário; “*Following Puccini*” acompanha o desempenho de uma peça musical, este processo é muito importante para os estudantes de música obterem conhecimentos e habilidades sobre solfejo, harmonia, processos de composição, formas de música e reconhecimento de timbre (FREITAS; MORAIS, 2019).

Destaca-se, também, o “*Tone Deafener*” jogo baseado em habilidade no qual o jogador controla o jogo ao tocar um instrumento musical acústico, cada nota tocada corretamente é atribuída uma pontuação ao usuário; “*E-Santur*”, desenvolvido para possibilitar o aprendizado de conceitos musicais de música clássica persa com o instrumento chamado *Santur*; “*Pitch Paradise*”,

---

<sup>6</sup> Dá-se o nome de leitura à primeira vista a prática de ler uma partitura nunca vista antes, ou seja, ver pela primeira vez a partitura e em tempo real executar a canção inteira ou um trecho sem nunca ter visto sua escrita musical.

um jogo para auxiliar o ensino de música às crianças, além de apresentar três mini-jogos sobre ritmo, altura de notas e melodia (FREITAS; MORAIS, 2019).

Cita-se, ainda, o jogo “*Piano Game*”, que utiliza o computador para a educação musical e estimula as atividades de audição, execução e melhora o reconhecimento das notas musicais; “*Chaluva Swing Festival*”, um jogo com o objetivo pedagógico geral de estimular a curiosidade e a criatividade dos jogadores, permitindo que eles facilmente reproduzam música usando *gamepads* como instrumentos musicais; “*TuneTrain*”, que foi desenvolvido para a capacitar crianças a compor música sem a necessidade de ensinar conceitos musicais explicitamente (FREITAS; MORAIS, 2019).

No jogo “*Chorlody*”, os alunos desenvolvem habilidades por meio da ação rítmica para reconhecer progressão de acordes e mostra os acordes para ajudar na cognição auditiva dos jogadores e apresenta diferentes combinações de notas; “*Guitar man*”, jogo com o objetivo de alcançar a cooperação musical e trabalhar performances com alguns instrumentos musicais por meio de computação gráfica e “*GenVirtual*”, jogo musical que possibilita seguir uma sequência de cores e sons emitidos a partir dos objetos virtuais projetados no mundo real (FREITAS; MORAIS, 2019).

Outra abordagem encontrada foi por meio da utilização do jogo “*Zorelha*”, no qual os sons dos instrumentos são aleatoriamente executados e a criança deve indicar o instrumento em execução clicando no personagem. O jogo “*Musikinésia*” auxilia alunos dos cursos de música na identificação das teclas de um teclado, além de ensinar outros conceitos musicais básicos, como andamentos, alterações cromáticas e leitura musical. O “*Descobrimo Sons*” é um jogo de escuta, em que o jogador/aluno deve prestar atenção ao som tocado e tentar adivinhar, pelo timbre, qual instrumento está emitindo o som (FREITAS; MORAIS, 2019).

Tais jogos concentram-se nos públicos-alvo infantil e adulto, sendo crianças com 56%, estudantes que queiram aprender música ou aprimorar seus conhecimentos 36%; e adultos 8%. Em relação à abordagem, o ensino para as crianças é focado na utilização das tecnologias com o intuito de introduzir música, já com os estudantes com objetivo de ingressar na área musical, a abordagem é centrada em despertar o interesse por música ou melhorar o entendimento e nos adultos melhorar o desempenho em uma peça musical principalmente o elemento ritmo (FREITAS; MORAIS, 2019).

O uso de jogos sérios para o ensino de música é considerado uma área relevante de pesquisa na Informática e Educação, seja para o público infantil ou adulto. Diversas pesquisas se propuseram a ensinar, mas, para o ensino-aprendizagem de programação e música, a quantidade de artigos obtidos nos resultados da RS foram dois. Com isso, os resultados obtidos serviram de análise para afirmar que há uma carência de jogos desenvolvidos para ensino de música, como o ponto de partida para o ensino de outras disciplinas.

Diante desse cenário é oportuno o desenvolvimento de uma ferramenta gamificada para suprir a necessidade de apoio ao ensino de música e programação, como objetiva esta pesquisa. Levando em conta a relação da programação com os conceitos de música, a TAS e o grande desafio de ingressar um músico na área da Computação para que ele tenha um diferencial no mercado de trabalho e consiga desenvolver novas tecnologias para a área de música (FREITAS; MORAIS, 2019), pretende-se, a seguir, apresentar ferramentas utilizadas no ensino e aprendizagem de programação, enquanto norteadores de uma técnica interdisciplinar.

#### **4.2 Ferramentas utilizadas no ensino e aprendizagem de programação**

Boa parte das ferramentas para o ensino de programação é disponibilizada por meio da internet e a velocidade e o impacto na disseminação desses conteúdos educacionais vêm beneficiando alunos e professores de várias áreas. Esta é uma nova forma de interagir que permite ultrapassar os limites do ensino presencial.

Segundo Blatt, Becker e Ferreira (2017), durante os anos de 2014 a 2016, o *Scratch* foi a principal ferramenta utilizada para o ensino de linguagem de programação. Os blocos lógicos que fazem parte dessa ferramenta permitem a criação de histórias, jogos e animações. O uso de programação em blocos tem um caráter lúdico e está sempre focando o visual, no primeiro momento, os códigos não são algo tão importante quanto os aspectos visuais.

Outras ferramentas que vêm ganhando destaque são os jogos sérios e as plataformas gamificadas. Em estudos como o de Silva e colaboradores (2015), as abordagens para o ensino de programação são compostas por 45% de pesquisas que propõem o uso de softwares, 20% utilização de jogos, 14% outras metodologias e 12% por meio de robótica.

As metodologias mais utilizadas para ensino de programação podem ser encontradas na revisão sistemática de Blatt, Becker e Ferreira (2017) destacando estudos de gamificação/lúdico

64,71%, em seguida interdisciplinar 20,59%, mediação online e robótica 5,88% e, por último, o modelo tradicional com 2,94%.

Os dados apresentados demonstram que as ferramentas lúdicas e interdisciplinares podem servir como material didático para transposição de conteúdos técnicos e de difícil compreensão, se passados por meio de aulas tradicionais e isso é o que tem acontecido.

O fenômeno da gamificação consiste na utilização de elementos dos jogos sérios (como narrativa, sistema de *feedback*, sistema de recompensas, conflito, cooperação, competição, objetivos e regras claras, níveis, tentativa e erro, diversão, interação, interatividade, entre outros) fora do contexto dos jogos sérios, com a intenção de resolver problemas e potencializar aprendizagens nas mais diversas áreas do conhecimento e da vida dos indivíduos (KAPP, 2012).

Originada em programas de marketing e aplicações para web, com a finalidade de motivar, engajar e fidelizar clientes e usuários (ZICHERMANN; CUNNINGHAM, 2012), a gamificação tem por finalidade tentar obter o mesmo grau de envolvimento e motivação que normalmente encontramos nos jogadores.

Fardo (2013) ressalta que a gamificação pode ainda deixar a interação das pessoas com o conhecimento de forma prazerosa. O que desperta o interesse pelo método:

- Levar em conta a diversão: os bons games são boas ferramentas para a aprendizagem. Pensar esse aspecto na educação pode melhorar a experiência que os indivíduos têm dentro dos ambientes de aprendizagem, o que acaba por potencializar a aprendizagem como um todo; (FARDO, 2013, p. 5-6).

A ferramenta desenvolvida na presente pesquisa, utilizará de base as referidas orientações, em suas diversas possibilidades de uso, e levando em conta, ainda, que elas não apontam para um único caminho de aplicação da gamificação em ambientes de aprendizagem, visando um ensino interdisciplinar.

A interdisciplinaridade rompe a divisão de conhecimentos através da união das ciências e aprimora problemas associados à divisão de categoria de saberes com um foco teórico metodológico na educação. O ensino interdisciplinar de programação é uma proposta apresentada por vários estudos. Por exemplo, o *Scratch* é usado para desenvolver atividades de cálculo do delta, ministrada na disciplina de matemática e na criação de algoritmos relacionados ao conteúdo da língua inglesa (BORBA, et al., 2015).

Outra pesquisa relata sobre o ensino de computação de forma interdisciplinar com português. Neste, os autores fizeram uma relação de programação com conteúdo de português, e, como resultados, tiveram *feedbacks* positivos e desenvolveram softwares para ensino da disciplina de português (SANTOS et al., 2017).

Além do ensino interdisciplinar de programação e português, outros estudos fazem a inclusão de programação em cursos de engenharias. Por exemplo, o estudo realizado em 2005 a 2007 compara a aplicação de uma metodologia para iniciação da aprendizagem de programação em cursos de engenharia, na qual algumas universidades ainda não tinham programação em seus cursos (GONZÁLEZ, 2008).

Além dos estudos anteriores, a relação da música com a programação em pesquisas como a de Silla et al. (2018), nas quais estudantes tiveram aulas em paralelo de programação e música. Ao final das duas disciplinas, os alunos desenvolveram quatro jogos, o primeiro jogo foi “*Drums Listen And Execute Game*”, desenvolvido para melhorar a percepção musical; já o segundo, “*Keyboard Chord Trainer*”, tinha o objetivo de melhorar a leitura de acordes e escalas ; o terceiro jogo “*Guitar Scale Mastery*”, tem o objetivo de melhorar as habilidades ao tocar guitarra através dos estudos de escalas e arpejos e o quarto jogo, “*Sight Reading Game*”, tem como intuito ajudar pessoas a melhorar suas habilidades de “leitura de partituras à primeira vista” (SILLA et al., 2018).

Diante dos estudos apresentados, infere-se que há uma carência de estudos que façam uma relação do ensino de programação envolvendo música, especialmente no que se refere à utilização da TAS, de modo que é possível traçar, a partir de algumas das técnicas já apresentadas, uma ferramenta gamificada interdisciplinar com conteúdo das duas áreas.

## 5 RESULTADOS E DISCUSSÕES

Unindo os dois métodos, quantitativo e qualitativo, buscou-se planejar e construir a ferramenta *MusicMeaning*. Com isso, tanto a revisão sistemática quanto os conceitos que relacionam a linguagem do músico com a programação, gerados com base na aula experimental, serviram de embasamento para a elaboração da ferramenta.

Portanto, nas próximas seções, serão apresentados os resultados da seleção dos conceitos, análise do *Frescobaldi/Lilypond* com conceitos selecionados, exemplos da aplicação dos conceitos utilizando a linguagem *Lilypond* e as telas da ferramenta gamificada: *MusicMeaning*.

### 5.1 Elaboração do glossário

Com o intuito de melhorar a compreensão com base na TAS, nesta seção serão apresentados os resultados da seleção dos principais conceitos de programação seguindo uma ordem lógica apresentada em disciplinas dos cursos de Computação. Ao lado de cada conceito terá um exemplo de relação com a música, ou seja, assimilação do novo conhecimento (programação) com o conhecimento existente (música). Os conceitos selecionados são:

Tabela 4 - Conceitos selecionados

Conceitos	Programação	Música
Algoritmo :	O algoritmo é conjunto de ações para a resolução de um problema em um número finito de passos, é também a parte mais complexa da programação, além de poder ser implementado em diferentes linguagens.	Na música o algoritmo se faz presente na organização dos sons e silêncios. Em uma orquestra para a obra musical ser executada é preciso que todos os músicos sigam os passos da partitura. A sequência de etapas de uma partitura pode ser comparada com um algoritmo.
Sintaxe:	A sintaxe de programação é um conjunto de regras que ditam como as sentenças podem ser escritas como sequências de componentes básicos. As regras determinam se a combinação ou “string” é válida ou não. São as regras que validam o formato das instruções que compõem um programa.	Para criamos uma partitura é necessário seguir uma série de regras que determinam como será a execução, por exemplo, a partitura precisa conter 5 linhas e 4 espaços, clave para determinar a altura das notas, fórmula de compasso e tonalidade. Tudo isso pode ser observado em uma partitura escrita à mão, ou seja, fazem parte da

Conceitos	Programação	Música
		<p>sintaxe musical, sendo partes necessárias para que o músico entenda o que deve ser tocado.</p>
Tipos de dados:	<p>Na programação os dados são divididos em três tipos: numéricos, literais e lógicos. Os numéricos são divididos em duas classes: inteiros e reais. Os literais são constituídos por uma sequência de caracteres contendo letras, dígitos e/ou símbolos especiais, também conhecidos no inglês como “String” ou alfanuméricos e por fim dados lógicos representados por dois únicos valores possíveis: verdadeiro e falso.</p>	<p>Os tipos de dados se comparado com uma partitura musical, seriam as linhas, notas, pausas, claves, acidentes, armaduras, compassos e oitavas que formam um conjunto de 32 símbolos conhecidos como notação musical, tornando possível sua interpretação por parte dos músicos.</p>
Variáveis:	<p>As variáveis são valores que variam de elemento para elemento. Na computação, durante a execução de qualquer tipo de programa os computadores estão manipulando informações representadas pelos diferentes tipos de dados e para que essas informações não sejam perdidas o computador precisa guardá-las em sua memória volátil.</p>	<p>Na música podemos encontrar estruturas semelhantes a variáveis. Ao lançarmos mão de uma grade de música vamos encontrar vários instrumentos dispostos nessa grade. Vamos considerar que a grade da música é o código principal de um programa e os instrumentos são as variáveis que compõem esse programa.</p>
Entrada e saída de dados:	<p>Entradas e saídas são a comunicação do programa com o mundo real, a forma com que o programa recebe os dados a serem processados do mundo real e devolve um resultado.</p>	<p>Na música ao assistir uma orquestra composta por maestro e músicos acompanhados de seus instrumentos, os músicos realizam a leitura dos dados que estão na partitura, faz o processamento dos mesmos e geram o som. Dessa forma, o músico executa os passos da partitura e tem como saída o som emitido pelo seu instrumento.</p>

Conceitos	Programação	Música
Funções:	<p>As funções também conhecidas como sub-rotinas, são utilizadas para resolver um problema muito específico. Um dos grandes benefícios é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva.</p>	<p>Na música existem estruturas que tem por objetivo realizar ações semelhantes às funções de programação. No Jazz, por exemplo, é normal que músicas contenham sessões dedicadas a improvisos. Nesse momento específico destinado ao improviso, todos os improvisadores devem respeitar as progressões contidas no chorus harmônicas ali dispostas, mesmo acontecendo improvisos, a sequência dos acordes tem que ser respeitada. Essa estrutura poderia ser chamada de função improviso, pois nela existem regras pré-definidas (imutáveis) e o resultado desta função depende especificamente de cada músico improvisador.</p>
Bibliotecas:	<p>Quando você usa uma linguagem de programação existe uma possibilidade de se usar funções pré-escritas por outro programador, essas funções resolvem certos problemas sem que você precise reinventar um código. E esse conjunto de funções se chama biblioteca.</p>	<p>Podemos fazer uma relação com a música e observar que esta função está presente nesse contexto também. Ao observarmos a grade (Score em Inglês), ou guia do regente, de uma música, vamos ali encontrar a partitura de cada instrumento. Assim, podemos considerar como uma biblioteca com características específicas inerentes a ele. Portanto, a grade pode ser considerada como um código que incluiu várias bibliotecas que colaboram com o código principal.</p>

Fonte: O autor, 2020.

A fim de aplicar os conceitos selecionados da apostila de Lucas Blatt (2019), foi feita uma investigação durante uma aula experimental com duas horas de duração para avaliar o entendimento dos alunos sobre os conceitos, bem como identificar se o conteúdo foi compreendido de forma separada.

Optou-se pela escolha dos alunos do PPGCCA pois o programa abrange discentes de diversas áreas, além de conter discentes da música, computação e de outras áreas que não conhecem

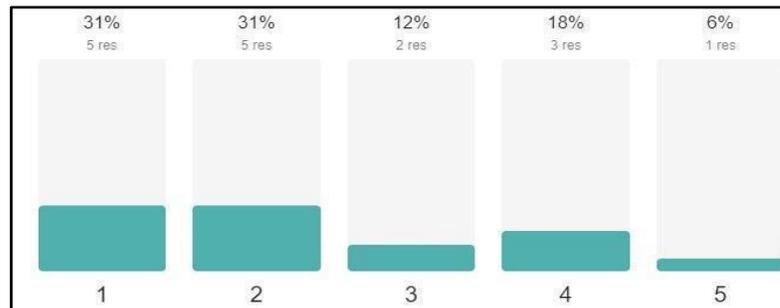
programação e a música. Desta forma, foi possível melhorar os conceitos e ter uma percepção de como organizar o conteúdo para aplicar a TAS na ferramenta.

O primeiro dado obtido foi sobre o conhecimento das áreas de música e programação. As perguntas iniciais foram a respeito do entendimento de partituras musicais e programação. Os dados coletados se caracterizam por: 38% dos discentes entrevistados compreendem partituras musicais, 62% não possuem conhecimento básico sobre música. Em relação à programação, 56% dos discentes sabiam programação e 44% não sabiam.

A partir da informação sobre a estrutura cognitiva de cada aluno, foi iniciada a aula e ao final foram perguntados se os conteúdos de programação relacionados à música serviram para entender partituras e se os conceitos de música relacionados à programação serviram para entender linguagem de programação, com base na escala de 1 a 5 (1- Sem dificuldade, 2- Pouca dificuldade, 3-Parcialmente difícil, 4- Dificuldade e 5-Muita dificuldade).

Os dados sobre a dificuldade de entender partituras após a aula utilizando a ferramenta *Frescobaldi*, 31% dos entrevistados não tiveram dificuldades ou com pouca dificuldade e apenas um discente apresentou muita dificuldade (Figura 8).

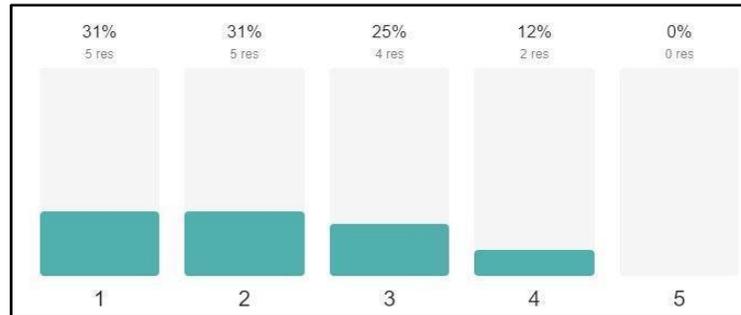
Figura 8 - Dificuldade no entendimento de partituras



Fonte: O autor, 2020.

A dificuldade de entender conceitos de programação foi medida utilizando a mesma escala da pergunta anterior. A figura 9 mostra os dados acerca da dificuldade, na qual 31% dos discentes assinalaram não ter dificuldades ou apresentaram poucas dificuldades e não houve caso de extrema dificuldade para o entendimento de programação.

Figura 9 - Dificuldade no entendimento de programação



Fonte: O autor, 2020.

A presente investigação serviu como base e parte fundamental para criação da ferramenta gamificada. A partir da percepção da estrutura cognitiva de estudantes de várias áreas, incluindo música e programação foi possível elaborar um novo conteúdo que engloba os conceitos da TAS, música e programação, sendo a música o subsunçor e a programação o novo conhecimento. Portanto, é por meio da estruturação do glossário e da aplicação da TAS que os músicos passarão a entender os conceitos da programação.

## 5.2 Exemplos da aplicação dos conceitos

A integração do glossário a exemplos práticos de programação irá facilitar ainda mais o entendimento, despertando o interesse do músico em aprofundar o conhecimento para criação de suas próprias ferramentas. Dessa forma, o primeiro exemplo apresenta a criação de uma partitura de melodia simples em compasso quaternário na linguagem *LilyPond* e sua representação visual. Nesse exemplo, os conceitos de biblioteca e função são aplicados juntamente com alguns códigos (Figura 10).

Figura 10 - Exemplo de melodia simples em compasso quaternário

```

1 \include "portugues.ly"
2 \header { title = "Título da música" }
3
4 \relative do' {
5 \clef treble
6 \key do \major
7 \time 4/4
8
9 do2 re mi fa
10
11 \bar "|"
12 }

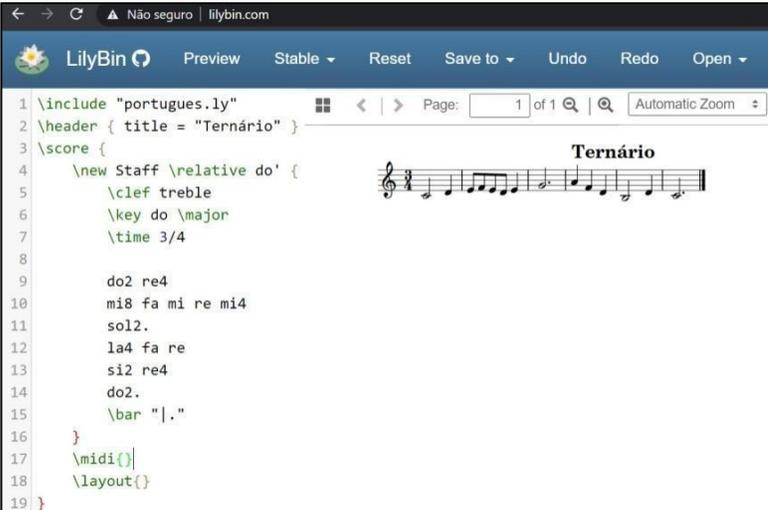
```

Fonte: O autor, 2020.

No exemplo da Figura 10, temos a inclusão da biblioteca “português.ly”, utilização da função `\header` para inserir o título da música, o `\relative do'` faz com que as notas sejam relacionadas de acordo com o dó e a apóstrofe (do'), serve para indicar que a nota será escrita uma oitava acima. O comando “`\clef`” permite a escolha das claves (e.g. treble, bass etc.) Já o “`\key`” determina tonalidade da música (e.g. fá maior, ré menor etc.). A fórmula de compasso é representada pelo código “`\time`”. Assim se alguém quiser determinar se uma música é em ritmo binário, ternário ou quaternário simples, inclui-se o comando “`\time`” seguido da fração correspondente à fórmula de compasso (e.g. 4/4 ;2/4 etc), o número 2 na frente do dó indica que a figura vai ser uma mínima e a linha `\bar “|.”` adicionará uma barra final.

O segundo exemplo apresenta a criação de uma partitura em dó maior em compasso ternário, criada com a linguagem *LilyPond*. Nesse exemplo, foram adicionados novos códigos que fazem parte da Sintaxe do programa *LilyPond*, disponíveis no apêndice A. Como apresentado na figura 11.

Figura 11 - Exemplo de melodia maior em compasso ternário



```

1 | \include "portugues.ly"
2 | \header { title = "Ternário" }
3 | \score {
4 |   \new Staff \relative do' {
5 |     \clef treble
6 |     \key do \major
7 |     \time 3/4
8 |
9 |     do2 re4
10 |    mi8 fa mi re mi4
11 |    sol2.
12 |    la4 fa re
13 |    si2 re4
14 |    do2.
15 |    \bar "|."
16 |  }
17 |  \midi{}
18 |  \layout{}
19 | }

```

The screenshot shows the LilyBin web editor interface. The code editor on the left contains the LilyPond code for a ternary melody in D major. The preview window on the right displays the resulting musical score, titled "Ternário", in treble clef with a 3/4 time signature. The score consists of a single staff with a melody starting on D4, followed by eighth notes and quarter notes, ending with a double bar line and repeat dots.

Fonte: O autor, 2020.

No código do compasso ternário foram adicionados novos elementos: `\score`, `\new Staff`, `\time 3/4`, números 4 e 8 e as linhas `\midi{}` e `\layout{}` contidas na função `\score` que geram os arquivos MIDI e o *template*. O código `\new Staff` indica que teremos um novo pentagrama, o `\time 3/4` informa o tipo do compasso e o número 4 representa a figura semínima e o 8 a figura colcheia (Figura 11).

O terceiro e último exemplo contém duas melodias simultâneas e a utilização de novos códigos que tornam a relação da música com a programação mais completa, possibilitando uma nova prática com novos recursos (Figura 12).

Para executar uma música simples contendo duas melodias simultâneas, é necessário utilizar << depois da linha `\score{}` e fechar com >> antes da linha `\midi{}`. Além disso, o código `\new Staff` e `\relative` será adicionado duas vezes, uma para cada pentagrama. Em algumas notas vamos utilizar o `s` de sustenido e o `\time 2/4` para indicar a quantidade de figuras por pentagrama (Figura 12).

Figura 12 - Exemplo de duas melodias simultâneas

The screenshot shows the LilyBin web editor. The left pane contains the following LilyPond code:

```

1 \include "portugues.ly"
2 \header { title = "MELODIAS SIMULTÂNEAS" }
3 \score {
4   <<
5     \new Staff \relative do' {
6       \clef treble
7       \key la \major
8       \time 2/4
9       la4 sols
10      la si
11      sols fas
12      la sols
13      la2
14    }
15    \new Staff \relative do' {
16      \clef treble
17      \key la \major
18      \time 2/4
19      la4 si
20      dos re
21      mi re
22      dos si
23      la2
24      \bar "|"
25    }
26  >>
27  \midi{}
28  \layout{}
29 }

```

The right pane displays the rendered musical score titled "MELODIAS SIMULTÂNEAS". It features two staves in 2/4 time, key of A major. The first staff contains the notes: A4, G4, F4, E4, D4, C4, B3, A3. The second staff contains the notes: A4, G4, F4, E4, D4, C4, B3, A3. The score is rendered in a standard musical notation style.

Fonte: O autor, 2020.

Os exemplos utilizando a linguagem *LilyPond* na qual podemos observar a relação da música com a programação e os novos conceitos presentes no glossários, permitem que músico transite entre a duas áreas e adicione ao seu currículo uma nova característica que permite que o

usuário crie suas próprias partituras digitais em linhas de código com conceitos de programação relacionados com a música por meio da TAS.

A seguir, será apresentada a modelagem da ferramenta gamificada, que apresenta enquanto proposta exatamente essa possibilidade de o profissional, aluno ou entusiasta de música possa transitar entre a Música e a Programação.

### 5.3 Plataforma gamificada

A ferramenta *MusicMeaning* possui duas versões uma para desktop disponível no endereço: (<https://bitly.com/XWIPJ>) e a outra para android disponibilizada na *Play Store*: (<https://play.google.com/store/apps/details?id=com.victoramaryl.musicmeaning>). A linguagem de programação utilizada na primeira versão (Apêndice C) é a Java no ambiente de desenvolvimento *Processing* muito usada dentro do contexto das artes visuais como software de aprendizagem e para criação de conteúdos visuais.

Na segunda versão (Apêndice D) optamos por utilizar os *frameworks* Ionic, Angular e Cordova. Tratando-se de ferramentas gratuitas de código aberto e com a possibilidade de criação de um sistema híbrido, acessado por diferentes plataformas (Android, IOS e Web). Um dos motivos para o desenvolvimento da segunda versão foi a facilidade de se ter um telefone móvel nos dias atuais. Segundo os dados da Pesquisa Nacional por Amostra de Domicílios (PNAD, 2019), divulgados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o percentual de 98,6% das pessoas utilizaram o telefone móvel para acessar à internet em 2019.

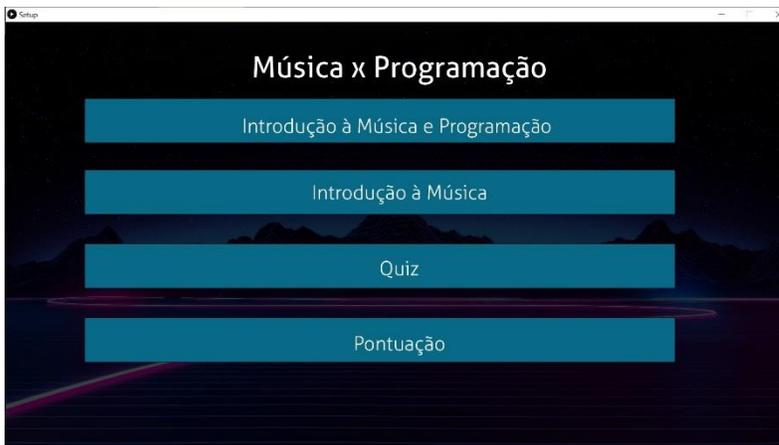
O *MusicMeaning* mistura elementos síncronos (interações simultâneas) e assíncronos (sem sincronismo). As interações simultâneas englobam resultados imediatos ao clicar em uma opção de resposta e os assíncronos resulta na comunicação que não ocorre ou não se efetiva ao mesmo tempo, por exemplo, a pontuação e o *feedback* só serão apresentados ao final de cada nível.

Em relação à exposição de conteúdo, a ferramenta se comporta de maneira assíncrona, contendo um total de 55 páginas de conteúdo na sua primeira versão e 70 páginas na versão mais recente, as quais tratam de uma introdução sobre música e programação e trazem exemplos em textos, áudios e imagens. Por fim, oferece uma apostila interativa composta de conteúdos de programação e música que possam ser relacionados, gamificação e questionários com exemplos sonoros.

Diante dos conceitos relacionados no modelo de assimilação, organograma, diagrama de caso de uso e fluxograma de navegação, temos que os resultados contemplaram a construção da ferramenta *MusicMeaning*. A ferramenta desenvolvida com base nas linguagens de programação *Java e Javascript*, tem como propósito auxiliar o entendimento de programação para músicos, baseado na TAS de Ausubel.

É importante destacar que esta ferramenta gamificada possui níveis para averiguar o conhecimento do aluno e ajudar na construção de um novo conhecimento. Deste modo, foram elaborados exemplos utilizando a linguagem *LilyPond* e criado um “quiz” gamificado que contém elementos de jogos. A ferramenta apresenta a página inicial com quatro opções tanto na versão 1 quanto na versão 2, “introdução à música e programação”, “introdução à música”, “quiz” e “pontuação”, conforme representado na Figura 13 e na Figura 14.

Figura 13 - Tela Inicial (Versão 1)



Fonte: O autor, 2021.

Figura 14 - Tela Inicial (Versão 2)



Fonte: O autor, 2021.

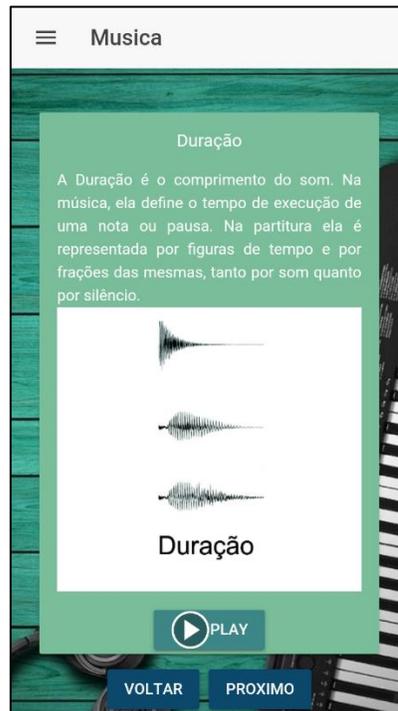
No menu principal temos a primeira opção “introdução à música e programação” com o objetivo de apresentar os conceitos contidos no glossário, relação da música com a programação e a utilização dos exemplos práticos da seção 5.2 como intermédio para absorção de um novo conhecimento.

Na segunda opção, introdução à música, temos conceitos de música com a intenção de averiguar o conhecimento existente e apresentar um pouco dos conceitos da área musical para aqueles que conhecem ou querem conhecer a área.

Além disso, algumas telas possuem exemplos sonoros que podem ser reproduzidos se clicados no botão “Play” no centro da página. Essa reprodução juntamente com as imagens ajuda

no entendimento dos termos técnicos. Outra informação presente nas páginas são os botões “voltar” para lembrar algum assunto passado e o botão “próximo” que adianta para a próxima página seguindo um roteiro lógico (Figura 15).

Figura 15 - Exemplo de reprodução sonora (Versão 2)



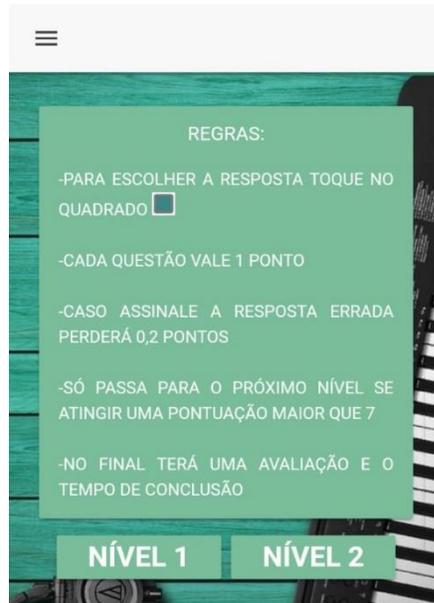
Fonte: O autor, 2021.

Após as opções dos conteúdos, é importante o usuário ter compreendido todo o assunto para que inicie a verificação do conhecimento, selecionando a terceira opção “quiz” para realizar a verificação do aprendizado acerca dos conteúdos apresentados.

Os elementos de gamificação podem ser observados a partir desta opção, por exemplo, o “quiz” possui dois níveis: o primeiro apresenta conteúdos relacionados à música com o intuito de avaliar a estrutura cognitiva do usuário. O segundo nível apresenta perguntas mais técnicas que envolvem assuntos de programação e música, o conhecimento prévio ancorado ao novo conhecimento (Figura 16).

Além do mais, Cada nível tem 10 questões objetivas que valem 1 ponto cada, com quatro alternativas de A à D, algumas alternativas possuem exemplos sonoros e os usuários têm uma tentativa para acertar a resposta correta, caso erre é subtraído 0,2 pontos, conforme as regras na Figura 16.

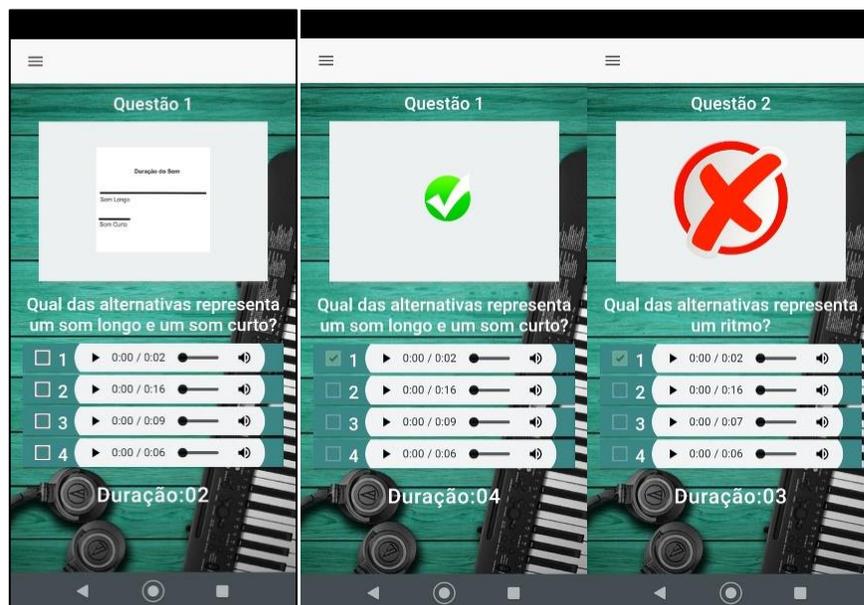
Figura 16 - MusicMeaning Quiz (Versão 2)



Fonte: O autor, 2021.

As respostas são automática, permitindo a visualização do resultado no mesmo instante. Deste modo, para que ocorra a exibição dos resultados foram criadas duas páginas, a primeira com símbolo na cor verde que representa: resposta correta, e a segunda página com um “X” na cor vermelha que corresponde a resposta errada, conforme a Figura 17.

Figura 17 - Telas dos resultados das questões (Versão 2)



Fonte: O autor, 2021.

Por fim, a última opção do menu, cujo nome atribuído foi pontuação. Esta opção contém uma página com os resultados do “quiz”, pontuação final, quantidade de acertos e uma avaliação que apresenta as seguintes mensagens: “Perfeito, Parabéns” quando o usuário responde corretamente todas as questões, “Muito bom” para pontuação acima de 8, “Na média” para pontuação 7 e “Estude mais um pouco” para notas abaixo de 7 (Figura 18).

Figura 18 - Tela de Resultados (Versão2)



Fonte: O autor, 2021.

O exemplo dos resultados na Figura 18, apresentou uma duração total de 44 segundos e uma pontuação final de 8.8, o fato ocorrido se deu por conta da seleção de uma resposta errada, subtraindo 0,2 da questão. O *feedback* “Muito Bom” é importante para uma autoavaliação por parte do usuário e dependo do resultado serve como meio motivador para continuar estudando o novo conhecimento adquirido.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho buscou estudar e analisar conceitos referentes ao processo de aplicação da TAS, entender quais conceitos de programação podem ser relacionados com a música, bem como, a criação de um glossário que serviu como material para construção do *MusicMeaning*, ferramenta voltado aos músicos que desejam aprender a programar e desenvolver suas próprias ferramentas.

Com o desenvolvimento deste projeto, o resultado pretendido de criar uma ferramenta gamificada como forma de interação lúdica que promovesse o conhecimento sobre programação relacionado ao conhecimento da música por intermédio da TAS, pode ser considerado um objetivo alcançado.

O levantamento realizado sobre os jogos sérios existentes na literatura, como também, as principais temáticas exploradas no ensino de música por meio dos jogos sérios foram relevantes e serviram para esclarecer que ainda são poucos estudos acerca da aplicação de jogos interdisciplinares na área da música. Portanto, é interessante que, para a construção de qualquer ferramenta em qualquer área, é necessário pesquisar e analisar as soluções existentes.

Para criação da ferramenta gamificada foi feita uma seleção dos conceitos que poderiam ter relação com a música e a programação. Além do mais, foi aplicado um questionário com perguntas objetivas que serviu como *feedback* em relação ao conceitos selecionados na apostila de Lucas Blatt (2019), tornando possível a elaboração de exemplos práticos utilizando conceitos e a linguagem *LilyPond*. Sem esta seleção, não seria possível utilizar a linguagem *LilyPond* e nem aplicar os conceitos da TAS.

Com isso, a arquitetura do “*MusicMeaning*” torna-se um recurso adicional a ser incorporado no ensino de programação para músicos, contribuindo com trabalhos interdisciplinares na área de música e programação.

O desenvolvimento deste processo deixou claro a importância do envolvimento de pessoas da área de computação e música na construção da ferramenta, pois sem o relacionamento das áreas não seria possível delinear o escopo da ferramenta, bem como as maneiras mais adequadas de abordar os conteúdos específicos.

### 6.1 Recomendações para estudos futuros

Como sugestão de trabalhos futuros no contexto desta ferramenta: a primeira sugestão surge no sentido de avaliar a ferramenta *MusicMeaning* realizando uma pesquisa com alunos e

professores de música sobre a usabilidade e os aspectos pedagógicos. Dessa forma, será possível obter resultados concretos acerca da eficácia da transmissão dos conhecimentos de programação que podem ser relacionados com a música. Tais resultados podem servir de base para melhorias na ferramenta e adequação em diferentes plataformas, como por meio de um endereço eletrônico.

Por fim, sugere-se também a adição de elementos gráficos, personagens, novos níveis para proporcionar uma aprendizagem mais lúdica e a inserção de novos exemplos que relacione a música com a programação.

## REFERÊNCIAS

AGUIAR, Bernardo; CORREIA, Walter; CAMPOS, Fábio. Uso da Escala Likert na Análise de Jogos. Anais do X Simpósio Brasileiro de Games e Entretenimento Digital, v. 7, n. 09, 2011.

AUSUBEL, D. P; et al. Psicologia Educacional. Rio de Janeiro: Interamericana, 1980.

ARAGÃO, Júlio. Introdução aos estudos quantitativos utilizados em pesquisas científicas. *Rev Práxis* 2011; 3(6): 59-62.

BHATTACHARYA, J. Rudiments of computer science. Academic Publishers, 2010.

BIOLCHINI, J.; Mian, P. G.; Natali, A. C. C.; Travassos, G. H. Systematic review in software engineering. Technical report, RT-ES 679/05 System Engineering and Computer Science Dept., COOPE/UFRJ. 2005.

BLATT, Lucas. "INTRODUÇÃO A PROGRAMAÇÃO A PARTIR DE CONHECIMENTOS SOBRE MÚSICA". Dissertação de Mestrado (curso de Computação da Universidade Federal da Paraíba (2019).

BLATT, Lucas; BECKER, Valdecir; FERREIRA, Alexandre. Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação. In: Anais do Workshop de Informática na Escola. 2017. p. 815.

BORBA, Clarissa et al. Utilizando o Scratch para Desenvolver a Interdisciplinaridade. Anais do Salão Internacional de Ensino, Pesquisa e Extensão, v. 7, n. 1, 2015.

CHAMORRO, Anelise et al. EDUCAÇÃO MUSICAL E AS TECNOLOGIAS DIGITAIS: O USO DE OBJETOS DE APRENDIZAGEM E A PERCEPÇÃO DOS DOCENTES. Revista Educação e Linguagens, v. 6, n. 11, 2017.

COELHO, H.: Tecnologias de informação. Lisboa: D. Quixote, 1986

FARDO, Marcelo Luis. A GAMIFICAÇÃO APLICADA EM AMBIENTES DE APRENDIZAGEM. CINTED-UFRGS Novas Tecnologias na Educação 6 V. 11 Nº 1, julho, 2013. Disponível em:<<https://www.seer.ufrgs.br/renote/article/view/41629/26409>> Acesso em 23 de out. de 2020

FARIAS, Gilberto; SANTANA, Eduardo. Introdução à computação. v1. 0, Universidade Aberta do Brasil, 2013.

FREITAS, Victor; MORAIS, Alana. Ensino de Música apoiado pelo uso de serious games: revisão sistemática sobre o panorama de publicações nacionais e internacionais. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2019. p. 833.

GADOTTI, M. Educação e globalização neoliberal: um olhar a partir da América Latina. Revista Educação & Linguagem, São Paulo, v. 1, p. 6278, 2009.

GONZÁLEZ, S. Montenegro; TAMARIZ, A. D. Real. Uma experiência no ensino de programação para cursos de engenharia, XXVIII Encontro nacional de engenharia de produção, 2008. Disponível em < [http://www.abepro.org.br/biblioteca/enegep2008\\_TN\\_STO\\_078\\_544\\_12163.pdf](http://www.abepro.org.br/biblioteca/enegep2008_TN_STO_078_544_12163.pdf)> Acessado em: Março de 2020

IBGE – Instituto Brasileiro de Geografia e Estatística. PNAD Contínua TIC 2019. Disponível em < <https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html> > Acessado em: Janeiro 2021.

KAPP, Karl. The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Pfeiffer, 2012.

KITCHENHAM, Barbara. Procedures for performing systematic reviews. Keele, UK, Keele University, v. 33, n. 2004, p. 1-26, 2004.

LEDDO, J.; ABELSON, R. P. (1986). The nature of explanations. In J. A. Galambos, R. P. Abelson & J. B. Black (Eds.), Knowledge Structures. New Jersey: Hillsdale, Lawrence Erlbaum and Associates.

LÜDKE, Menga e ANDRÉ, Marli E. D. A. **Pesquisa em educação: abordagens qualitativas**. São Paulo: EPU, 1986.

MOREIRA, M. A. Aprendizagem Significativa. Brasília: UNB, 1999.

MEDEIROS, Tainá Jesus; et al. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. **RENOTE-Revista Novas Tecnologias na Educação**, v. 11, n. 3, 2013.

MOORE, F. R. Uma abordagem tecnológica da música. *Música Hoje*, Belo Horizonte, v. 3, p. 135-155, 1995

NASCIMENTO, Karlise Soares; SILVA STAMBERG, Cristiane; LEMKE, Cláudia Eizandra. Jogos Educacionais: revisão bibliográfica com base em trabalhos publicados no CINTED. *Informática na educação: teoria & prática*, v. 20, n. 3 set/dez, 2017.

NOVAK, Joseph D. Instalações para Ensino de Ciências no Ensino Médio. Padrões em evolução em instalações e programas. 1972.

PAIVA, Jose Eduardo Ribeiro de et al. Uma análise crítica da relação música/tecnologia do pós guerra até a atualidade. 1992.

POCINHO, M. Lições de revisão sistemática e metanálise, 2008. 2014.

SANTOS, Higor et al. Ensino de Computação de forma Interdisciplinar com Português: Um Relato de Experiência do PIBID. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2017. p. 863.

SHEA, Peter J.; PICKETT, Alexandra M.; PELZ, William E. A follow-up investigation of “teaching presence” in the SUNY Learning Network. *Journal of Asynchronous Learning Networks*, v. 7, n. 2, p. 61-80, 2003.

SILLA, Carlos N. et al. Girls, music and computer science. In: 2018 IEEE Frontiers in Education Conference (FIE). IEEE, 2018. p. 1-6.

SILVA, Thiago Reis et al. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, v. 23, n. 01, p. 182, 2015.

SOMMERVILLE, I. Engenharia de software. 8ª ed., São Paulo: Pearson/Addison Wesley, 2011.

SOUZA, Draylson Micael; BATISTA, Marisa Helena da Silva; BARBOSA, Ellen Francine. Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. *Revista Brasileira de Informática na Educação*, 2016. 2

RIBEIRO, R.P.; NUÑEZ, I.B. Pensando a aprendizagem significativa: dos mapas conceituais às redes conceituais. In: NUÑEZ, I. B.; RAMALHO, B. L. Fundamentos do Ensinoaprendizagem das Ciências naturais e da Matemática: o novo ensino médio. Porto Alegre: Sulina, p. 201-225, 2004.

VIANNA, Ysmar; VIANNA, Maurício; MEDINA, Bruno; TANAKA, Samara. Gamification, Inc.: como reinventar empresas a partir de jogos. MJV Press: Rio de Janeiro, 2013.

VIEIRA, Gabriel da Silva. O home studio como ferramenta para o ensino da performance musical. Goiânia, p. 21, 2010.

ZAINA, L. A. M. Projeto Multidisciplinar: uma Experiência Prática no Ensino de Programação em um Curso de Engenharia da Computação. XXXIII Congresso Brasileiro de Ensino de Engenharia, 2005, Campina Grande. Disponível em: <<http://www.abenge.org.br/cobenge/arquivos/14/artigos/SP-15-28545321805-1118683317399.pdf>>. Acesso em: 11 mai. 2020.

ZICHERMANN, Gabe; CUNNINGHAM, Christopher. Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps. Canada: O’Reilly Media, 2011.

ZUBEN, Paulo. Música e tecnologia: o som e seus novos instrumentos. Irmãos Vitale, 2004.

## APÊNDICES

### APÊNDICE A- Lista de códigos básicos

Tabela 4 Lista de comandos

<code>\include</code>	Inclui bibliotecas com extensão .LY
<code>\language</code>	Traduz as notas para outro idioma
<code>\header{ }</code>	Função ou bloco onde podemos inserir títulos, compositores, subtítulos, arranjadores, autoria, entre outros.
'	(Apóstrofe) uma oitava acima
,	(Vírgula) uma oitava abaixo
%	Serve para adicionar um comentário.
<code>\clef treble</code>	Clave de sol, padrão
<code>\clef bass</code>	Clave de fá
<code>\clef alto</code>	Clave de dó na 3ª linha
<code>\clef tenor</code>	Clave de dó na 4ª linha
<code>\time 3/4</code>	Fórmula de compasso
<code>\key re \major</code>	Escolha de tonalidade ou Key Signature em Inglês
<code>\bar "     "</code>	Barra dupla
<code>\bar "  ."</code>	Barra final
<code>\relative{ }</code>	Faz com que o <i>Lilypond</i> intérprete uma nota em relação à anterior
<code>\score{ }</code>	No caso de grade, gerencia todos instrumentos criados e une em uma única partitura. Sempre que usar essa função podemos gerar arquivos midi.
<code>\layout{ }</code> e <code>\midi{ }</code>	São criadas dentro da função <code>\score{ }</code>
<code>\new Staff</code>	Utilizado para adicionar mais de um pentagrama.
s	Sustenido
b	Bemol
ss	Dobrado sustenido
bb	Dobrado bemol

!	Mostra o acidente
?	Acidente de precaução
1	Semibreve
2	Mínima
4	Semínima
8	Colcheia
16	Semicolcheia
32	Fusa
64	Semifusa
r1	Pausa de semibreve
r2	Pausa de mínima
r4	Pausa de semínima
r8	Pausa de colcheia
r16	Pausa de Semicolcheia
r32	Pausa de Fusa
r64	Pausa de Semifusa
(	Início da ligadura de expressão
)	Final da ligadura de expressão
~	Ligadura de valor (tempo)
<<	Abertura da simultaneidade (sempre que abrir precisa fechar)
>>	Fechamento da simultaneidade

**APÊNDICE B - Questionário**

Questão 1 - Olá, qual o curso de graduação você concluiu?

Questão 2 - Você entende partituras musicais?

Questão 3 - Sabe programar?

Questão 4 - Com base no que foi ensinado, marque qual foi sua dificuldade de entender partituras.

Questão 5 - Com base no que foi ensinado, marque qual foi sua dificuldade de entender conceitos de programação.

Questão 6 - Qual ferramenta foi a mais clara no ensino de música e programação?

Questão 7 - Diante do que foi ensinado as ferramentas foram úteis para o entendimento de música e programação?

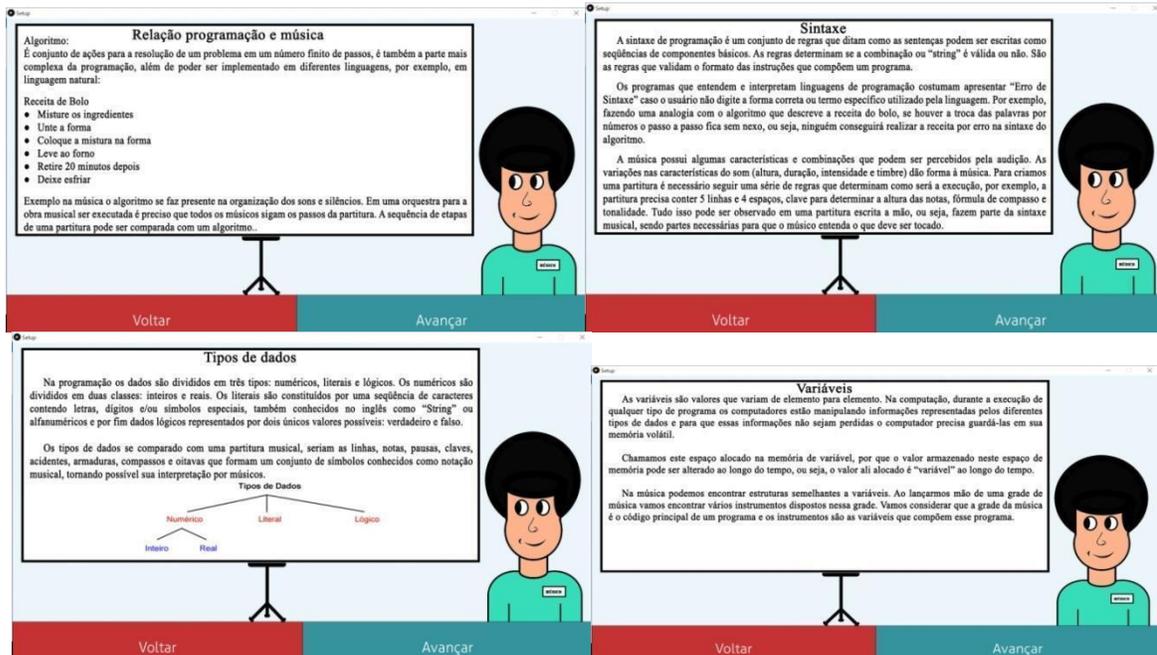
Questão 8 - Você tem alguma sugestão para o ensino de música e programação? (Caso não tenha de deixar em branco)

## APÊNDICE C –Primeira versão

### Menu



### Telas de introdução à Música e Programação:







### Frequências das Notas Musicais

D "064" por exemplo, tem 261,63 Hz. A sua oitava, que é um "100" mais agudo, tem o dobro de vibração que a primeira, ou seja, 523,25 Hz. Na partitura, a altura das notas são definidas pela Clave e suas respectivas linhas e espaços. Fisicamente, a quantidade de ciclos de vibração é o que define a altura de um som, ou seja, quanto mais ciclos por segundo, mais agudo é a nota.

Nota	Frequência
C	261,63
D	293,43
E	329,63
F	369,59
G	391,93
A	430,53
B	466,13
C	512,03

Exemplo de Frequências:

Stop-Exemplo

Voltar Avançar

### Melodia

A melodia é o conjunto de notas executadas de forma sucessiva. Na partitura elas são apresentadas horizontalmente por meio de intervalos ascendentes ou descendentes juntos e disjuntos.

Exemplo de Melodia:

Stop-Exemplo



Voltar Avançar

### Clave de Sol

As claves são responsáveis por normar as notas que estão nas linhas e espaço do pentagrama. Ela é escrita em uma das linhas do pentagrama, no caso da Clave de Sol, surge-se na segunda linha. As notas subsequentes (ascendentes e descendentes), dão nomes as outras notas do pentagrama.

Exemplo Clave de Sol:



Stop-Exemplo

Voltar Avançar

### Clave de Fá

A clave de Fá, assim como a clave de Sol, dá nome as notas no pentagrama. Ela é pode ser grafada, tanto na quarta linha quanto na terceira.

Exemplo Clave de Fá:



Stop-Exemplo

Voltar Avançar

### Clave de Dó

Assim como as claves de Fá e Sol, a clave de "Dó", determina o nome das notas a partir da sua localização na partitura. A clave de "Dó" na terceira linha é também conhecida como clave de viola, que é bastante usada para este instrumento, assim como para Trombone (contralto ou alto) e violas graves.

Exemplo Clave de Dó:



Stop-Exemplo

Voltar Avançar

## Telas do Quiz:

# Início do Quiz

Hora de testar seus conhecimentos sobre o que aprendeu aqui :)

Avançar

# Music Meaning

Quiz-Programação por meio de conhecimentos sobre música.

Número de chances: 2

NÍVEL 1

NÍVEL 2

Questão 1 - Qual das alternativas representa um som longo e um som curto?

A - Exemplo sonoro

B - Exemplo sonoro

C - Exemplo sonoro

D - Exemplo sonoro

Pontuação final: 0.0

Avançar

Questão 2 - O que é Timbre?

A - É o conjunto de notas executadas de forma sucessiva.

B - É definida pela quantidade que Hertz

C - É o volume de um som

D - É o formato de uma onda sonora.

Voltar Avançar

Questão 3 - O que é Intensidade?

A - É o conjunto de notas executadas de forma sucessiva.

B - É o volume de um som

C - É definida pela quantidade que Hertz

D - É o formato de uma onda sonora.

Voltar Avançar

Questão 4 - Qual das alternativas representa um ritmo?

A - Exemplo sonoro

B - Exemplo sonoro

C - Exemplo sonoro

D - Exemplo sonoro

Voltar Avançar

Questão 5 - O que é Melodia?

A - É o conjunto de notas executadas de forma sucessiva.

B - É o volume de um som

C - É definida pela quantidade que Hertz

D - É o formato de uma onda sonora.

Voltar Avançar

Questão 7 - Qual linha do pentagrama é escrita a Clave de Sol?

A - Primeira Linha

B - Segunda Linha

C - Terceira Linha

D - Quarta Linha

Voltar Avançar

Questão 8 - Quais linhas podem ser grafadas a Clave de Fá?

A - Primeira Linha e Terceira Linha

B - Segunda Linha e Quarta Linha

C - Terceira Linha e Quarta Linha

D - Primeira Linha e Quarta Linha

Voltar Avançar

The image displays three screenshots of a quiz application interface. Each screenshot shows a question, four multiple-choice options, a 'Voltar' (Back) button, and an 'Avançar' (Next) button. A 'Questões' (Questions) sidebar is visible in each, listing the question number and the selected answer.

**Questão 6 - O Dó4 tem quantos Hertz?**

- A - 293,63Hz
- B - 273,63Hz
- C - 393,63Hz
- D - 261,63Hz

**Questão 9 - Qual instrumento utiliza a Clave de Dó?**

- A - Trompete
- B - Flautas
- C - Violão
- D - Viola

**Questão 10 - O que define as alturas das notas?**

- A - Intensidade
- B - Timbre
- C - Claves e suas respectivas linhas e espaços
- D - Harmonia e Melodia

### Tela Final - Pontuação e Feedback:

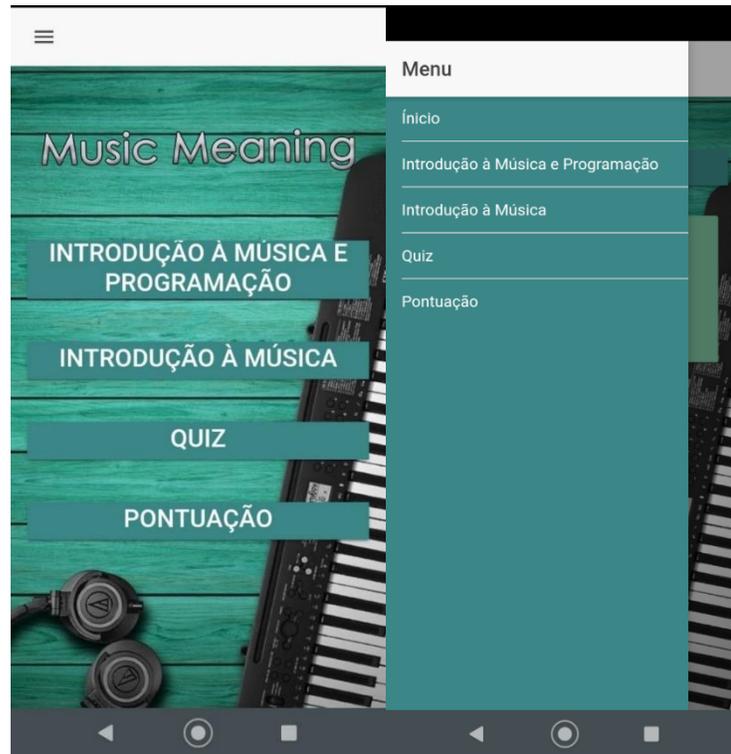
The final screen of the quiz is displayed on a black background with white text. It announces the end of the quiz and provides the final score and feedback.

**Fim do quiz!**  
Muito Bom

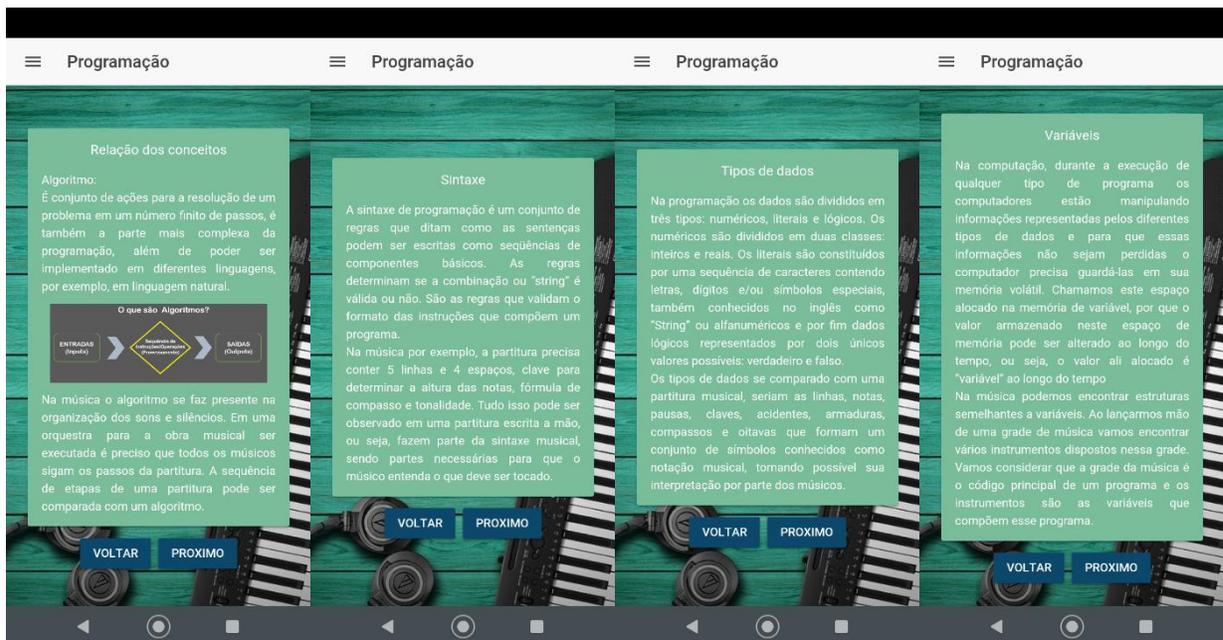
Pontuação final: 8.5  
Acertou: 10.0 de 10 questões.

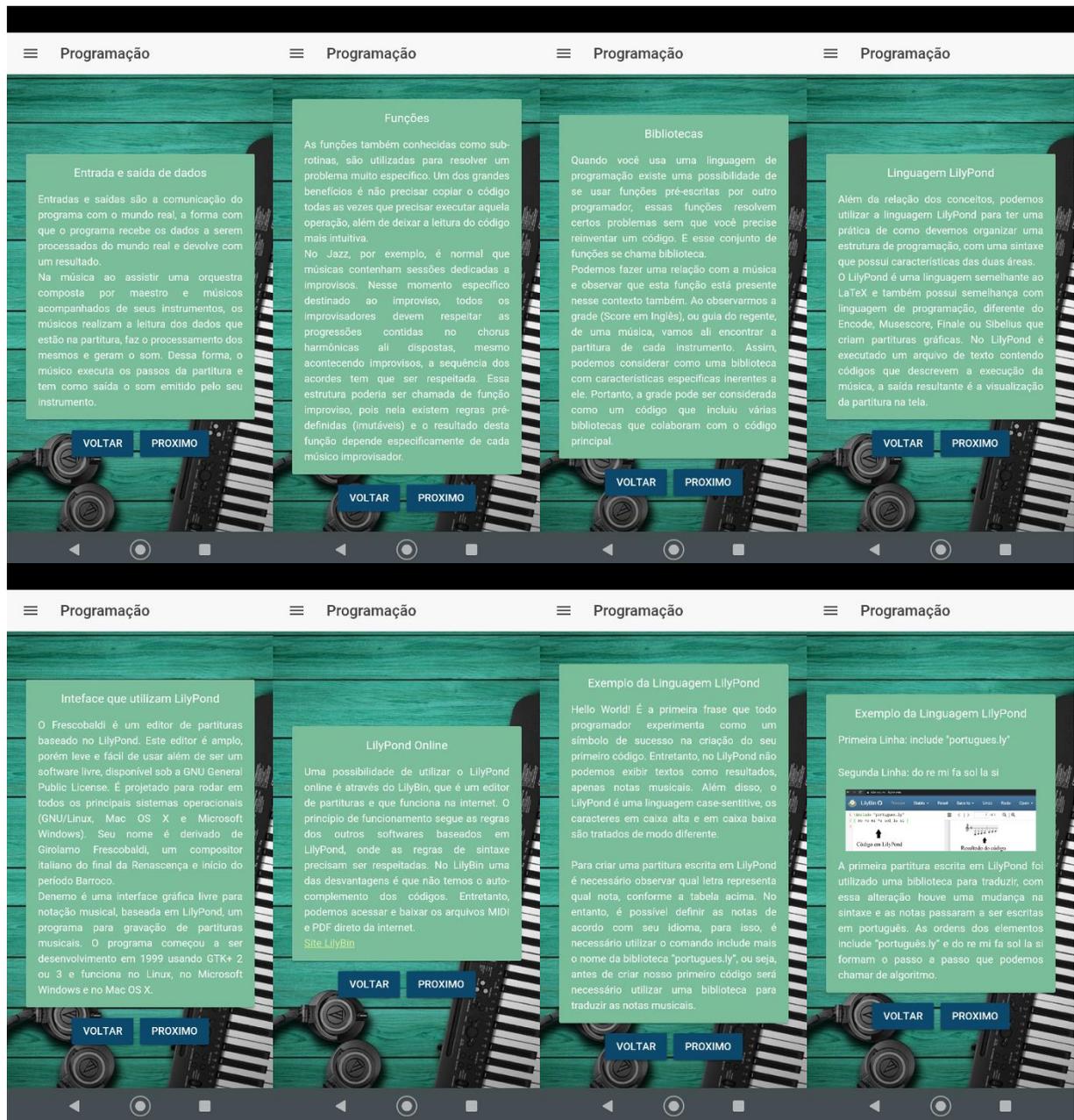
Retornar ao início

## APÊNDICE D – Segunda versão



### “Introdução à música e programação”:





Programação

Programação

Programação

Programação

Exemplo de melodia simples em compasso quaternário



No exemplo acima, temos a inclusão da biblioteca "portuguese.ly", utilização da função header para inserir o título da música, o relative do' faz com que as notas sejam relacionadas de acordo com o dó e o apóstrofe serve para indicar que a nota será escrita em uma oitava acima. O comando clef treble coloca a clave de sol padrão, já o key do major adiciona uma armadura de clave. O tempo é: ime 4/4, o número 2 na frente do dó indica que a figura vai ser uma mínima e a linha "r |" adicionar uma barra final.

VOLTAR PROXIMO

Exemplo de melodia maior em compasso ternário



No código do compasso ternário foram adicionados novos elementos: score, ew Staff, ime 3/4, números 4 e 8 e as linhas midi e layout contidas na função score que geram os arquivos MIDI e o template. O código ew Staff indica que teremos um novo pentagrama, o ime 3/4 informa o tipo do compasso e o número 4 representa a figura semínima e o 8 a figura colcheia.

VOLTAR PROXIMO

Download MIDI e partitura PDF



O LilyBin permite realizar o download do MIDI se for inserido a função midi dentro da função score. Caso tenha digitado corretamente as funções score, midi e layout para realizar o download, basta clicar nos três traços no canto direito da tela na parte superior.

VOLTAR PROXIMO

Exemplo de duas melodias simultâneas



Para executar uma música simples contendo duas melodias simultâneas, é necessário utilizar um símbolo depois da linha score e fechar com o mesmo símbolo antes da linha midi. Além disso, o código ew Staff e relative será adicionado duas vezes, uma para cada pentagrama. Em algumas notas vamos utilizar o s de sustenido e o ime 2/4 para indicar a quantidade de figuras por pentagrama.

VOLTAR PROXIMO

Programação

Erro de sintaxe no LilyBin



Um detalhe importante da sintaxe ao usar dois pentagramas é lembrar de iniciar com os símbolos presentes na imagem anterior e não esquecer de fechar com o mesmo símbolo para evitar os erros de sintaxe.

VOLTAR MENU

## “Introdução à música”:

**Duração**

A Duração é o comprimento do som. Na música, ela define o tempo de execução de uma nota ou pausa. Na partitura ela é representada por figuras de tempo e por frações das mesmas, tanto por som quanto por silêncio.

**Melodia**

A melodia é o conjunto de notas executadas de forma sucessiva. Na partitura elas são apresentadas horizontalmente por meio de intervalos ascendentes ou descendentes juntos e disjuntos.

**Altura**

A altura é definida pela quantidade que Hertz (vibração) de um som. Na música, cada nota tem uma quantidade específica de hertz. O "dó4" por exemplo, tem 261,63 Hz. A sua oitava, que é um "dó5" mais agudo, tem o dobro de vibração que a primeira, ou seja, 523, 25 Hz. Na partitura, a altura das notas são definidas pela Clave e suas respectivas linhas e espaços. Fisicamente, a quantidade de ciclos de vibração é o que define a altura de um som, ou seja, quanto mais ciclos por segundo, mais aguda é a nota.

**Intensidade**

Intensidade é o volume de um som. Fisicamente é a distância das extremidades de uma vibração. A figura abaixo mostra a altura de cada onda em relação ao seu ponto de partida. A verde tem menos intensidade, assim, tem um volume menor (fraco). A azul tem o volume médio e a vermelha tem um volume maior (forte).

**Timbre**

O Timbre é o formato de uma onda sonora. Na música, cada instrumento tem um desenho específico de som. Isso é causado pela interferência do corpo de vibração a uma caixa de ressonância.

**Ritmo**

Ritmo é a organização métrica do som e silêncio. Ele expressa o valor do pulso musical (lento ou rápido) através de um andamento pré-definido. O exemplo ao lado qualifica a pulsação de uma valsa.

**Andamento**

Na partitura o andamento é definido por BPM (batimento por minutos) exemplos ao lado: exemplo sonoro: Tomaso Albinoni - Adagio

Instrumento	BPM	Definição
Clarinete	120-140	Muito agilmente e estivo
Flauta	120-140	Muito agilmente
Violino	60-80	Lento
Alto	60-80	Organicamente, de suavidade e patético
Violoncelo	60-80	Velocidade de arco lento, melódico e orgânico
Contrabaixo	60-80	Muito lento, com a distância, agitado e compassivo
Mandolina	160-180	Melódica, viva, sem ritmo, sem batido
Maracas	110-120	Ritm. del. lento com Adagio
Alto	100-120	Lento e largo
Violão	100-120	Melódico e vivo
Violoncelo	80-100	Mel. agudo e rico que a Flauta
Flauta	100-120	Vivaz e melódico
Flautim	200-220	Muito rapidamente, com toda velocidade e precisão

**Harmonia**

Harmonia é o conjunto de notas simultâneas que "acomodam" as melodias, dando sentido ao que é executado por sons sucessivos.

Musica Musica Musica Musica

### Contraponto

Contraponto é um conjunto de melodias simultâneas e sucessivas. São frases executadas como um diálogo musical.



VOLTAR PROXIMO

### Pauta ou Pentagrama

Pauta ou pentagrama é um conjunto de 5 linhas e 4 espaços sobrepostos onde são grafadas as alturas das notas.



VOLTAR PROXIMO

### Clave de Sol

As claves são responsáveis por nomear as notas que estão nas linhas e espaço do pentagrama. Ela é escrita em uma das linhas do pentagrama, no caso da Clave de Sol, surge-se na segunda linha. As notas subsequentes (acidentes e descendentes), dão nomes as outras notas do pentagrama.

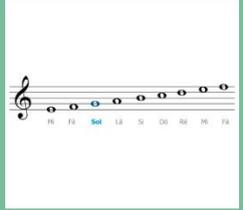


PLAY

VOLTAR PROXIMO

### Exemplo Clave de Sol – Notas no pe...

Notas da primeira linha até a última com Dó-Ré-Mi-Fá-Sol-Lá-Si. A clave de Sol já foi escrita na primeira linha, mas ela coincide com a clave de Fá na quarta linha.



VOLTAR PROXIMO

Musica Musica Musica Musica

### Clave de Fá

A clave de Fá, assim como a clave de Sol, dá nome as notas no pentagrama. Ela é pode ser grafada, tanto na quarta linha quanto na terceira.



PLAY

VOLTAR PROXIMO

### Exemplo Clave de Fá – Notas no pen...

A referência das notas nas linhas e espaço já divergem da clave de Sol



VOLTAR PROXIMO

### Clave de Dó

Assim como as claves de Fá e Sol, a clave de "Dó" determina o nome das notas a partir da sua localização na partitura. A clave de "Dó" na terceira linha é também conhecida como clave de viola, que é bastante usada para este instrumento, assim como para Trombone (contralto ou alto) e vozes graves.



PLAY

VOLTAR PROXIMO

### Exemplo Clave de Dó – Notas no pen...

Usando como referência a clave de Dó na terceira linha, temos o seguinte resultado:



VOLTAR PROXIMO

Musica Musica Musica Musica

### Figuras

As notas que comportam alturas definidas são representadas por figuras arredondadas (aberta ou fechadas). Nelas podem conter hastes ou não, Colchetes ou não. Isso porque cada combinação destes elementos altera a unidade de valor de cada nota.

Semibreve Mínima Semínima Colcheia Semicolcheia Fusa Semífusa  
1 2 4 8 16 32 64

### Figuras

Cada figura de valor positivo(som) tem sua respectiva negativa(pausa), ou seja, que representa silêncio.

Nome	Som	Silêncio	Duração
Semibreve			1
Mínima			1/2
Semínima			1/4
Colcheia			1/8
Semicolcheia			1/16
Fusa			1/32
Semífusa			1/64

### Semibreve

A semibreve, em relação as outras notas posteriores, é a nota de maior valor. A sua unidade equivale a um espaço "inteiro", ou seja, as outras notas são frações dela.

### Mínima

A Mínima é a fração  $\frac{1}{2}$  da Semibreve. Seu valor é metade da sua antecessora. A sua unidade equivale à metade de um espaço inteiro, ou seja, precisam duas mínimas para preencher uma semibreve.

VOLTAR PROXIMO VOLTAR PROXIMO VOLTAR PROXIMO VOLTAR PROXIMO

Musica Musica Musica Musica

### Semínima

A Semínima é a fração  $\frac{1}{4}$  da Semibreve. Seu valor é metade da Mínima. Precisam-se de quatro semínimas mínimas para preencher uma semibreve.

### Colcheia

A Colcheia é a fração  $\frac{1}{8}$  da Semibreve. Seu valor é metade da Semínima. Precisam-se de oito colcheias para preencher uma semibreve.

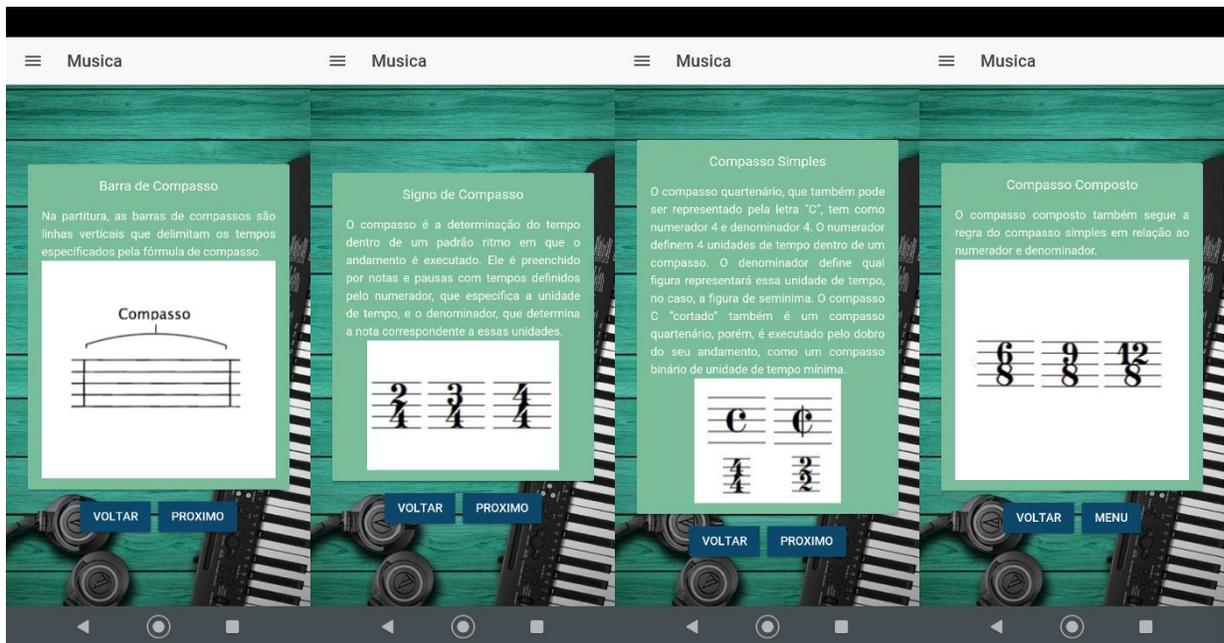
### Semicolcheia

A Semicolcheia é a fração  $\frac{1}{16}$  da Semibreve. Seu valor é metade da colcheia. Precisam-se de dezesseis colcheias para preencher uma semibreve.

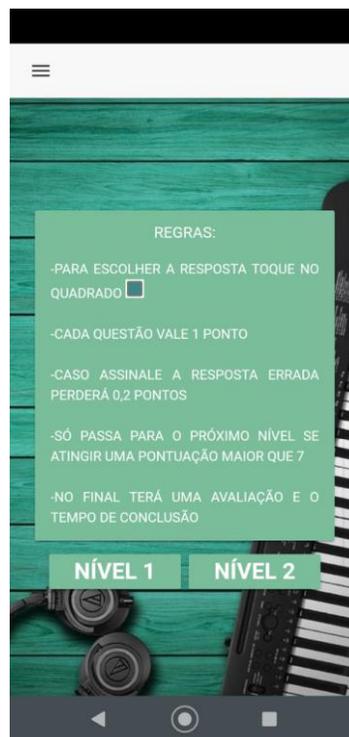
### Fusa

A Fusa é a fração  $\frac{1}{32}$  da Semibreve. Seu valor é metade da Semicolcheia. Ela é tem três colchetes em sua haste. Precisam-se de trinta e duas colcheias para preencher uma semibreve. Seguindo a sequência de metades, ainda tem a Semífusa que é  $\frac{1}{64}$  da Semibreve, e vale metade da Fusa. Esta última, até mesmo por questões de leitura, quase não é mais usada hoje em dia.

VOLTAR PROXIMO VOLTAR PROXIMO VOLTAR PROXIMO VOLTAR PROXIMO



## Regras e níveis:



## “Quiz”:

The image displays eight quiz questions arranged in a 2x4 grid. Each question is presented on a mobile app interface with a green wood-grain background and a keyboard. The questions are as follows:

- Questão 5:** O que é Melodia?
  - 1. É o conjunto de notas executadas de forma sucessiva.
  - 2. É definida pela quantidade que Hertz.
  - 3. É o volume de um som.
  - 4. É o formato de uma onda sonora.
- Questão 1:** Qual das alternativas representa um som longo e um som curto?
  - 1. 0:00 / 0:19
  - 2. 0:00 / 1:29
  - 3. 0:00 / 0:00
  - 4. 0:00 / 0:06
- Questão 4:** O que é intensidade?
  - 1. É o conjunto de notas executadas de forma sucessiva.
  - 2. É definida pela quantidade que Hertz
  - 3. É o volume de um som
  - 4. É o formato de uma onda sonora.
- Questão 3:** O que é Timbre?
  - 1. É o conjunto de notas executadas de forma sucessiva.
  - 2. É definida pela quantidade que Hertz
  - 3. É o volume de um som
  - 4. É o formato de uma onda sonora.
- Questão 2:** Qual das alternativas representa um ritmo?
  - 1. 0:00 / 0:19
  - 2. 0:00 / 1:29
  - 3. 0:00 / 0:07
  - 4. 0:00 / 0:06
- Questão 8:** Quais linhas podem ser grafadas a clave de Fá?
  - 1. Primeira linha e Terceira
  - 2. Segunda linha e Quarta
  - 3. Terceira linha e Quarta
  - 4. Primeira linha e Quarta
- Questão 7:** Qual linha é escrita a clave de Sol?
  - 1. Primeira linha
  - 2. Segunda linha
  - 3. Terceira linha
  - 4. Quarta linha
- Questão 6:** O Dó4 tem quantos Hertz?
  - 1. 293,63 Hz
  - 2. 273,63 Hz
  - 3. 393,63 Hz
  - 4. 261,63 Hz

Questão 9



Qual instrumento utiliza a Clave de Dó?

- 1. Trompete
- 2. Flautas
- 3. Violão
- 4. Viola

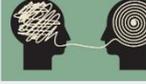
Questão 10

Nome	Som	Silêncio	Duração
Normal	o	-	1/2
Maior	o	-	1/2
Menor	o	-	1/2
Quarta	o	-	1/2
Quinta	o	-	1/2
Sexta	o	-	1/2
Sétima	o	-	1/2
Oitava	o	-	1/2

O que define as alturas das notas?

- 1. Intensidade
- 2. Timbre
- 3. Claves e suas respectivas linhas-espacos
- 4. Harmonia e Melodia

Questão 11



O que Sintaxe?

- 1. Conjunto de regras que mudam
- 2. Escrita em LilyPond
- 3. É um conjunto de regras que ditam como as sentenças serão escritas
- 4. Regras de negócio

Questão 12

```

int main() {
    int a = 10;
    int b = 20;
    int c = 30;
    int d = 40;
    int e = 50;
    int f = 60;
    int g = 70;
    int h = 80;
    int i = 90;
    int j = 100;
    return 0;
}

```

Fazem parte dos conceitos de programação:

- 1. Sintaxe, Algoritmo, Dados e Intensidade
- 2. Timbre, Claves, Funções e Melodia
- 3. Tipos de Dados, Variáveis e Sintaxe
- 4. Algoritmo, Harmonia e Melodia

Questão 13



O que é algoritmo?

- 1. Sequência de passos para solucionar um problema
- 2. Conjunto de regras que mudam
- 3. É um sistema de desenvolvimento
- 4. É um conjunto de notas

Questão 14



Qual alternativa é um tipo de dado em programação?

- 1. String
- 2. Clave de Sol
- 3. Do4
- 4. Função

Questão 15



Qual alternativa é um exemplo de sintaxe?

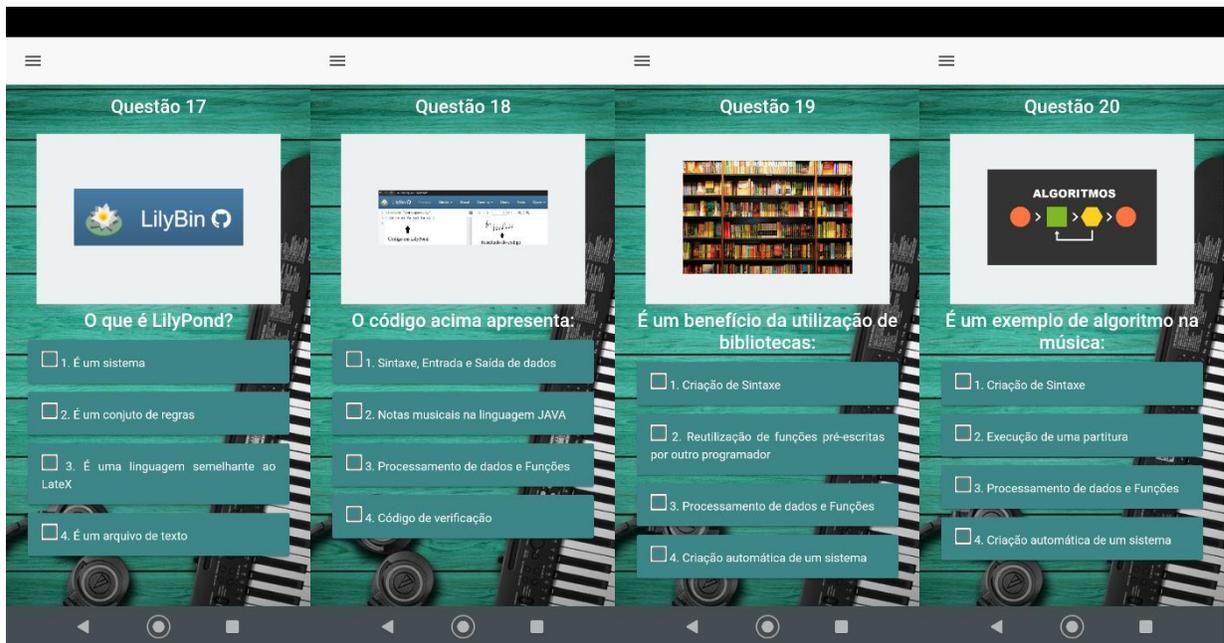
- 1. O sistema de Notação musical
- 2. Descrição dos processos
- 3. Criação de regras
- 4. Função

Questão 16



É um exemplo de entrada e saída de dados?

- 1. Mudança de um sistema
- 2. Leitura de uma partitura e a geração do som
- 3. Execução sonora
- 4. Leitura de uma partitura



### Duração, pontuação e feedback:

