

Vinícius Santana Silva

M HARMONY BR

**O uso de formalismos para a composição algorítmica baseada em
conhecimento**

Garanhuns

2018

Vinícius Santana Silva

M HARMONY BR

O uso de formalismos para a composição algorítmica baseada em conhecimento

Trabalho de conclusão de curso apresentado à coordenação do curso de Ciência da Computação da Universidade Federal Rural de Pernambuco – UFRPE, como requisito para obtenção do título de Bacharel em Ciências da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Unidade Acadêmica de Garanhuns

Curso de Bacharelado em Ciência da Computação

Orientador: Tiago Buarque Assunção de Carvalho

Coorientador: Jerônimo Barbosa da Costa Junior

Garanhuns

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Ariano Suassuna, Garanhuns-PE, Brasil

S586m Silva, Vinícius Santana
 M HARMONY BR O uso de formalismos para a composição
 algorítmica baseada em conhecimento / Vinícius Santana Silva.
 - 2018.
 66 f. : il.

 Orientador: Tiago Buarque Assunção de Carvalho.
 Coorientador: Jerônimo Barbosa da Costa Júnior
 Trabalho de Conclusão de Curso (Graduação em Ciência
da Computação) - Universidade Federal Rural de Pernambuco,
Departamento de Ciência da Computação, Garanhuns, BR-PE,
2018.

 Inclui referências e apêndice.

 1. Computação 2. Algoritmos 3. Música 4. Computação
aplicada I. Carvalho, Tiago Buarque Assunção de, orient.
II. Costa Júnior, Jerônimo Barbosa da, coorient. III. Título

CDD 004

Vinícius Santana Silva

M HARMONY BR
O uso de formalismos para a composição algorítmica baseada em conhecimento

Trabalho de conclusão de curso apresentado à coordenação do curso de Ciência da Computação da Universidade Federal Rural de Pernambuco – UFRPE, como requisito para obtenção do título de Bacharel em Ciências da Computação.

Trabalho aprovado. Garanhuns, 23 de fevereiro de 2018:

Tiago Buarque Assunção de Carvalho
Orientador

Maria Aparecida Amorim Sibaldo
Convidado 1

Igor Medeiros Vanderlei
Convidado 2

Garanhuns
2018

Resumo

Ao longo da história, diversos estudos buscaram introduzir o uso dos computadores em outras áreas do conhecimento, inclusive na música, várias pesquisas foram realizadas com a finalidade de mostrar a capacidade dos computadores em compor músicas. O presente trabalho teve como objetivo verificar a possibilidade da criação de um *software* simples capaz de gerar harmonias de determinado estilo musical, por meio de gramáticas livres de contexto, baseadas no conhecimento obtido de composições musicais já existentes. O desenvolvimento da ferramenta teve como base a análise das características de dois estilos musicais distintos: Punk Rock e Sertanejo Universitário, por meio da coleta e análise de dados de 40 músicas. Posteriormente, a base de dados foi ampliada com a inclusão de 50 músicas do estilo Reggae e de 30 músicas do estilo Sertanejo Universitário, totalizando 120 músicas. Utilizando os dados extraídos da base de músicas foram geradas 3 gramáticas específicas para produção de harmonias de cada estilo analisado por meio de um algoritmo de composição. O estudo teve como resultado a produção de um aplicativo que compõe harmonias baseadas em conhecimento, que se mostrou útil no auxílio ao processo de composição musical.

Palavras-chave: Composição Algorítmica. Computação Musical. Computação Aplicada.

Abstract

Throughout history, several studies have tried to introduce the use of computers in other areas of knowledge, including music, and several researches have been done to show the ability of computers to compose music. The present work had the objective to verify the possibility of the creation of a simple *software* capable of generating harmonies of a certain musical style, using context-free grammars, based on knowledge. The development of the tool was based on the analysis of the characteristics of two different musical styles: Punk Rock and Sertanejo Universitário, through the collection and analysis of data of 40 songs. Subsequently the database was expanded with the inclusion of 50 Reggae songs and 30 songs to the Sertanejo Universitário style, totaling 120 songs. Based on the information extracted from the music dataset, 3 specific grammars were generated to produce harmonies of each style analyzed by means of a composition algorithm. The study resulted in the production of an application that composes knowledge-based harmonies, which proved useful in aiding musical composition process.

Keywords: Algorithmic composition. Sound and Music Computing. Applied Computing.

Lista de ilustrações

Figura 1 – Gramática livre de contexto do campo harmônico	15
Figura 2 – Gramática livre de contexto do campo harmônico de Dó maior	16
Figura 3 – Tela inicial do MHarmonyBR	23
Figura 4 – Tela da função Compositor, do MharmonyBR	24
Figura 5 – Tela da função Avaliar Harmonia, do MharmonyBR	25
Figura 6 – Tela da função Alterar Tom, do MharmonyBR	25
Figura 7 – Trecho de código contendo acordes colocados em um HashMap	29
Figura 8 – Trecho de código contendo o <i>array</i> correspondente ao campo harmônico	29
Figura 9 – Trecho de código contendo o método Java responsável por analisar o tom da harmonia	30
Figura 10 – Trecho de código contendo o método Java responsável por analisar a progressão harmônica	31
Figura 11 – Porcentagem de progressões do estilo Punk Rock que iniciam com os graus 1 e 6 do campo harmônico - primeira análise	34
Figura 12 – Quantidade de acordes contidos nas progressões do estilo Punk Rock - primeira análise	34
Figura 13 – Quantidade de elementos contidos nas progressões do estilo Punk Rock - primeira análise	35
Figura 14 – Porcentagem de progressões do estilo Sertanejo universitário que iniciam com os graus 1 e 6 do campo harmônico - primeira análise	36
Figura 15 – Quantidade de acordes contidos nas progressões do estilo Sertanejo Universitário - primeira análise	36
Figura 16 – Quantidade de elementos contidos nas progressões do estilo Sertanejo Universitário - primeira análise	37
Figura 17 – Porcentagem de progressões do estilo Sertanejo Universitário que iniciam com os graus 1 e 6 do campo harmônico, após ampliação da amostra	38
Figura 18 – Quantidade de acordes contidos nas progressões do estilo Sertanejo Universitário, após ampliação da amostra	39
Figura 19 – Quantidade de elementos contidos nas progressões do estilo Sertanejo Universitário, após ampliação da amostra	39
Figura 20 – Porcentagem de progressões do estilo Reggae que iniciam com os graus 1 e 6 do campo harmônico	40
Figura 21 – Quantidade de acordes contidos nas progressões do estilo Reggae	41
Figura 22 – Quantidade de elementos contidos nas progressões do estilo Reggae	41
Figura 23 – Gramática livre de contexto para o estilo Punk Rock	43
Figura 24 – Gramática livre de contexto resumida para o estilo Sertanejo Universitário	44

Figura 25 – Gramática livre de contexto resumida para o estilo Reggae	45
Figura 26 – Trecho de código contendo o método Java responsável por gerar harmo- nias musicais	47
Figura 27 – Quantidade de vezes que cada produção foi escolhida – Primeiro teste .	48
Figura 28 – Quantidade de vezes que cada produção foi escolhida – Segundo teste	49
Figura 29 – Quantidade de vezes que cada produção foi escolhida – Terceiro teste .	49
Figura 30 – Quantidade de vezes que cada produção foi escolhida – Quarto teste . .	50
Figura 31 – Porcentagem de usuários que aprovaram a aparência do aplicativo . . .	52
Figura 32 – Porcentagem de usuários que tiveram problemas técnicos no uso do aplicativo	52
Figura 33 – Porcentagem de usuários que tiveram dificuldade no uso do aplicativo .	53
Figura 34 – Comentários dos usuários a respeito das dificuldades na utilização do aplicativo	53
Figura 35 – Porcentagem de usuários que julgaram o aplicativo útil para o cotidiano e para o aprendizado	54
Figura 36 – Qualidade de experiência de uso, segundo usuários do aplicativo	54
Figura 37 – Comentários dos usuários a respeito das dificuldades na utilização do aplicativo	55

Lista de tabelas

Tabela 1 – Nomenclatura dos acordes em cifras	14
Tabela 2 – Campo harmônico maior	15
Tabela 3 – Campo Harmônico de “Lá Menor”	15
Tabela 4 – Exemplo de tabela de dados harmônicos de músicas	33
Tabela 5 – Número de aparições que se repetiram dentro da amostra - Punk Rock	33
Tabela 6 – Número de aparições de progressões dentro da amostra - Sertanejo Universitário	35
Tabela 7 – Número de aparições de progressões que se repetiram dentro da amostra após ampliação- Sertanejo Universitário	38
Tabela 8 – Número de aparições de progressões que se repetiram dentro da amostra- Reggae	40

Lista de símbolos

b	Bemol
#	Sustenido

Sumário

1	INTRODUÇÃO	12
1.1	Motivação	12
1.2	Problemática	13
1.3	Objetivos	13
1.3.1	Objetivo geral	13
1.3.2	Objetivos específicos	13
2	REFERENCIAL TEÓRICO	14
2.1	Gramática livre de contexto	14
2.2	Gramáticas aplicadas ao campo harmônico	14
3	TRABALHOS RELACIONADOS	17
3.1	Primeiros estudos com composição algoritmica	17
3.2	Inteligência artificial na composição por computadores	17
3.3	Softwares comerciais	21
3.4	Considerações finais	21
4	MHARMONYBR	23
4.1	Roteiro de pesquisa	26
4.1.1	Seleção das músicas	26
4.1.2	Análise dos dados	26
4.1.3	Desenvolvimento e testes	27
5	ALGORITMOS BÁSICOS	28
6	BASE DE MÚSICAS	32
6.1	Primeira análise de músicas da base	33
6.2	Segunda análise de músicas da base	37
7	DESENVOLVIMENTO	42
7.1	Restrição de gramática	42
7.2	Algoritmo de composição	45
8	TESTES	48
8.1	Testes do algoritmo de composição	48
8.2	Testes de classificação	50
8.2.1	Primeiro teste	50

8.2.2	Segundo teste	50
8.3	Teste de usabilidade	51
9	RESULTADOS E CONCLUSÕES	56
	REFERÊNCIAS	58
	ANEXO A – MÚSICAS ANALISADAS, RESPECTIVOS ESTILOS MU- SICAIS E PROGRESSÕES HARMÔNICAS DETECTADAS MANUALMENTE	60
	ANEXO B – CLASSE QUE CONTÉM OS MÉTODOS RESPONSÁVEIS PELA COMPOSIÇÃO AUTOMÁTICA	63

1 Introdução

Presenciou-se nas últimas décadas um grande avanço do uso da tecnologia em diversas áreas do conhecimento humano, com o intuito de facilitar, dentre outras coisas, atividades cotidianas, e uma dessas, trata-se do processo criativo.

“A criatividade é essencial para a criação de arte. Dentre as diversas formas de arte, a música destaca-se, sendo a mais difundida. E, por tocar no intelecto e na emoção humana, a composição de música por computador é um desafio às áreas de pesquisa em inteligência artificial e computação musical. Esse desafio é tão grande que, ainda hoje, poucos foram os avanços na área de simulação convincente da criatividade.” (CAMARATTA; FISCHER, 2014, p. 1).

No atual cenário mundial, a utilização de ferramentas computacionais na produção musical mostra-se indispensável. Músicos, bandas e grandes estúdios utilizam desde simples *softwares* de gravação, aos mais complexos simuladores de instrumentos e efeitos como auxílio para o processo criativo, como pode ser observado após análise dos estudos sobre a indústria fonográfica, descritos em (ABREU, 2009).

A composição musical, por meio de algoritmos, não é um tema recente. Na verdade, os próprios músicos fazem uso de padrões para construir suas músicas, e isso fez com que diversos estilos musicais criassem uma identidade rítmica, harmônica, melódica e poética.

Neste trabalho, será descrito todo o processo de produção da harmonia musical, através de alguns formalismos utilizados na computação, explorando a possibilidade de gerar música de determinados estilos por meio da combinação dos elementos e regras do campo harmônico com esses formalismos.

1.1 Motivação

Mesmo para músicos experientes, o processo de composição pode ser um desafio. Tendo em vista o crescimento da utilização de ferramentas computacionais que auxiliem no processo criativo e no ensino, visualiza-se a oportunidade do desenvolvimento de um aplicativo que, além de auxiliar na criação musical, também facilite o acesso ao conhecimento sobre teoria musical e composição, principalmente para músicos iniciantes.

“A música é um produto social e simbólico de grande importância nas diferentes formações culturais, principalmente se considerarmos a sua capacidade de criar vínculos afetivos entre as pessoas. A música pode usar diferentes formas de linguagem e expressão, sendo produto cultural de características muito especiais: nenhum produto cultural tem mostrado tamanha capacidade de adaptação aos diferentes meios de comunicação.” (SANTINI; LIMA, 2001, p. 2).

Sendo a computação, assim como a música, uma área de conhecimento de tamanha relevância para a sociedade, que se propõe e consegue integrar-se com todas as outras áreas do conhecimento, essa união com a música se torna bastante interessante e motivadora.

1.2 Problemática

Apesar de existirem alguns estudos na área de computação musical, mais especificamente na área de composição algorítmica, as ferramentas geradas por esses trabalhos não são facilmente encontradas pelo público não-acadêmico interessado em ferramentas que auxiliem no processo de composição, pois tais pesquisas não têm como foco esse público.

Do crescente interesse pelo aprendizado da música, surge a necessidade de novos dispositivos capazes de facilitar o caminho para auxiliar na prática da composição musical. Atualmente, a maior parte dos programas voltados à produção musical tem como público-alvo músicos profissionais que procuram ampliar o conhecimento já adquirido. É possível para uma ferramenta computacional auxiliar e facilitar o processo criativo de músicos iniciantes?

1.3 Objetivos

1.3.1 Objetivo geral

Criar um *software* que seja capaz de gerar progressões harmônicas por meio de gramáticas baseadas no campo harmônico, e que sirva como ferramenta capaz de auxiliar o processo de composição musical.

1.3.2 Objetivos específicos

- Verificar a capacidade da gramática adotada de gerar combinações de acordes que se encaixem nas regras do campo harmônico;
- Conseguir por meio do *software* gerar harmonias que possuam a identidade sonora de estilos musicais conhecidos;
- Criar uma ferramenta eficiente;
- Fazer com que o *software* tenha uma usabilidade intuitiva e acessível para o público em geral.

2 Referencial Teórico

2.1 Gramática livre de contexto

Segundo MENEZES (1998), podemos definir uma gramática livre de contexto da seguinte maneira:

Uma gramática $G = (V, T, S, P)$, onde V é o conjunto das variáveis da gramática, T o conjunto dos símbolos da linguagem, S o símbolo inicial e P as produções, é dita ser uma Gramática Livre de Contexto (GLC) se todas as suas produções, em P , são da forma $A \rightarrow a$, onde A é uma variável de V e a é uma palavra de $(VUT)^*$. Sendo assim, entende-se que uma GLC é uma gramática cujo lado esquerdo de suas produções contém exatamente uma variável.

2.2 Gramáticas aplicadas ao campo harmônico

Para iniciar a confecção da gramática tomou-se como referência o campo harmônico maior, conforme apresentado na Tabela 1, na qual são apresentados os graus relativos à cada acorde do campo harmônico em números romanos. O grau I representa o acorde tônico, que é aquele que dá nome ao seu respectivo campo harmônico, aqui representado por meio das linhas. Os graus seguintes são, respectivamente, II (Menor), III (Menor), IV (Maior), V (Maior), VI (Relativo Menor) e VII (Diminuto). Nota-se, então, que esse padrão é replicado para todos os campos harmônicos e só é alterado para os campos harmônicos menores, consistindo em uma transposição do campo harmônico maior, colocando o grau que anteriormente era o relativo menor, como tônica, como mostra a Tabela 2, apresentando o campo harmônico de Lá Menor, que é a relativa menor de Dó Maior..

Tabela 1 – Nomenclatura dos acordes em cifras

Acorde	Dó	Ré	Mi	Fá	Sol	Lá	Si
Acorde maior - Cifra	C	D	E	F	G	A	B
Acorde menor - Cifra	Cm	Dm	Em	Fm	Gm	Am	Bm
Acorde sustenido - Cifra	C#	D#	-	F#	G#	A#	B#
Acorde bemol - Cifra	-	Db	Eb	-	Gb	Ab	Bb

Fonte: Próprio autor

Baseado na análise anteriormente descrita, foi possível a elaboração de gramáticas do tipo $G = (V, T, S, P)$, seguindo o modelo da Figura 1.

Tabela 2 – Campo harmônico maior

I Tônica	II Menor	III Menor	IV Maior	V Maior	VI Relativo Menor	VII Diminuto
C	Dm	Em	F	G	Am	B _o
C#	D#m	Fm	F#	G#	A#m	C _o
D	Em	F#m	G	A	Bm	C# _o
D#	Fm	Gm	G#	A#	Cm	D _o
E	F#m	G#m	A	B	C#m	D# _o
F	Gm	Am	A#	C	Dm	E _o
F#	G#m	A#m	B	C#	D#m	F _o
G	Am	Bm	C	D	Em	F# _o
G#	A#m	Cm	C#	D#	Fm	G _o
A	Bm	C#m	D	E	F#m	G# _o
A#	Cm	Dm	D#	F	Gm	A _o
B	C#m	D#m	E	F#	G#m	A# _o

Fonte: Próprio autor

Tabela 3 – Campo Harmônico de “Lá Menor”

I Tônica	II Diminuto	III Maior	IV Menor	V Menor	VI Maior	VII Maior
Am	B _o	C	Dm	Em	F	G

Fonte: Próprio autor

Figura 1 – Gramática livre de contexto do campo harmônico

$$G = (\{S, T\}, \{i, ii, iii, iv, v, vi, vii\}, S, P) \text{ tal que,}$$

$$P = \{S \rightarrow TS \mid i \mid ii \mid iii \mid iv \mid v \mid vi \mid vii\}$$

$$T \rightarrow i \mid ii \mid iii \mid iv \mid v \mid vi \mid vii\}$$

Fonte: Próprio autor

A exemplo, temos uma gramática utilizando os acordes do campo harmônico de Dó Maior como terminais (Figura 2).

Essa construção deve-se ao fato de que, dentro de um mesmo campo harmônico, podemos utilizar qualquer produção de progressão harmônica e todas elas resultaram em harmonias válidas, como é descrito em CHEDIAK (1986). Vale ressaltar que não tratamos neste trabalho da modulação tonal, que acontece quando dentro de uma mesma música ocorre uma mudança de tom da harmonia, dentro de uma mesma música. O objetivo é

Figura 2 – Gramática livre de contexto do campo harmônico de Dó maior

$$G = (\{S, T\}, \{c, dm, em, f, g, am, bdim\}, S, P) \text{ tal que,}$$
$$P = \{S \rightarrow TS \mid c \mid dm \mid em \mid f \mid g \mid am \mid bdim$$
$$T \rightarrow \mid c \mid dm \mid em \mid f \mid g \mid am \mid bdim\}$$

Fonte: Próprio autor

apresentar como sequências de acordes devem ser organizadas para que as regras de construção da harmonia sejam respeitadas.

3 Trabalhos Relacionados

Nesse capítulo, apresentar-se-á uma análise de alguns dos trabalhos realizados na área de composição algorítmica, com a finalidade de refletir acerca dos métodos utilizados e os resultados alcançados em pesquisas anteriores.

3.1 Primeiros estudos com composição algorítmica

Após leitura de diversos trabalhos como, AMES (1987), CAMARATTA e FISCHER (2014), BASSETO (2000), é possível observar que o uso de algoritmos voltados à composição musical data da década de 50, mais precisamente do ano de 1957, o qual tem-se os primeiros registros de trabalhos nesta área, em estudos realizados por Lejaren Hiller e Leonard Isaacson, JR; ISAACSON (1957), na Universidade de Illinois.

Como resultado da pesquisa, foi produzida uma música, batizada de Illiac Suite. Para fazer a seleção, de quais notas fariam parte da música, foram utilizadas heurísticas. Havia casos nos quais as notas não eram geradas, nesses casos o sistema voltava ao estado inicial e, a partir daí o programa executava todas as etapas novamente. “Eles justificam esse método dizendo que antes de resolver o problema de simular a expressividade musical, era necessário resolver problemas mais simples, como a própria composição da música, como sugeriu CAMARATTA e FISCHER ” (2014). Posteriormente, criou-se o MUSICOMP, que é um trabalho de colaboração entre Hiller e Robert Baker, utilizando as técnicas descritas anteriormente por Hiller e Isaacson, foram feitos testes da ferramenta em HILLER e BAKER (HILLER; BAKER, 1964).

Apesar de ser pioneiro, tanto o trabalho de Lejaren Hiller e Leonard Isaacson, quanto outros trabalhos posteriores não conseguiram criar músicas com riqueza artística, que pode ser traduzida como a simulação sentimentos humanos através da composição, utilizando apenas técnicas probabilísticas, segundo CAMARATTA e FISCHER (2014).

3.2 Inteligência artificial na composição por computadores

A partir dos anos 70, houve diversos trabalhos publicados na área de composição algorítmica que se utilizaram de inteligência artificial em seus experimentos, dentre os quais está RADER (1974) 1974), onde foram realizados experimentos utilizando um sistema de regras e pesos, os quais serviam para determinar a maneira que a melodia da música era feita, segundo os relatos, essa foi a maneira encontrada para tentar transmitir sentimentos por meio das composições. Curtis Roads (1979) apresentou um estudo, que teve como

objetivo fazer uma revisão da literatura e aplicar conceitos linguísticos aos estudos musicais. Quatro questões foram levantadas, são elas:

- “Can techniques from linguistics be usefully applied to the study of music?”
- “Can the use of grammar representations be effective in describing and specifying music structures?”
- “Are grammars intrinsically normative, i.e., are they fit only to describe traditional music?”
- “What are the implications of the representation of music as a grammar?”

O autor propôs algumas regras de produção baseadas em restrição e fez uma revisão de (RUWET, 1972 apud ROADS; WIENEKE, 1979), (RUWET, 1975 apud ROADS; WIENEKE, 1979), (NATTIEZ; DUNSBY, 1977 apud ROADS; WIENEKE, 1979), (LASKE, 1973 apud ROADS; WIENEKE, 1979), (LASKE, 1977 apud ROADS; WIENEKE, 1979), (LASKE, 1978 apud ROADS; WIENEKE, 1979), (SMOLIAR, 1976 apud ROADS; WIENEKE, 1979), (MOORER, 1972 apud ROADS; WIENEKE, 1979), (WINOGRAD, 1968 apud ROADS; WIENEKE, 1979), (WINOGRAD, 1973 apud ROADS; WIENEKE, 1979), (LERDAHL; JACKENDOFF, 1977 apud ROADS; WIENEKE, 1979), (ROADS, 1978 apud ROADS; WIENEKE, 1979), (LINDBLOM; SUNDBERG, 1970 apud ROADS; WIENEKE, 1979), (ULRICH, 1977 apud ROADS; WIENEKE, 1979), (SMITH, 1972 apud ROADS; WIENEKE, 1979) e (BUXTON, 1977 apud ROADS; WIENEKE, 1979) para concluir que gramáticas podem oferecer clareza na modelagem da sintaxe superficial e da macroestrutura das composições. E ainda, que a gama de gramáticas pode ser ampla, com bastante alcance e que não são úteis apenas para representação de música tradicional. O autor argumenta que a estrutura das expressões musicais e da gramática relacionadas podem ter uma modelagem concisa, pois as regras de produção de uma gramática são abstrações poderosas. Também concluiu-se que somente o uso de gramáticas e regras de produção não era suficiente para representação musical, quando a gramática usada for livre de contexto, e gramáticas sensíveis ao contexto não podem simplesmente ser usadas devido aos problemas para especificar regras sensíveis ao contexto e problemas técnicos associados à análise, tornam essa ferramenta pouco atrativa, além do que, quando a noção de sensibilidade ao contexto é colocada em conjunto com a forma da gramática, o seu uso fica inadequado para a música.

“Ultimately, the question is not whether music conforms to the structure of formal grammars, but rather whether particular formal grammars can be designed which are useful representations of certain compositions. Grammars with embedded procedures can be powerful descriptive and expressive tools, but certainly formal languages will evolve, and in general, knowledge representations will grow more elegant.”(ROADS; WIENEKE, 1979, p. 54)

Em MINSKY (1982) fez-se um estudo sobre inteligência artificial, o qual o autor levanta diversas questões populares a respeito da capacidade dos computadores de pensarem igual a um ser humano, inclusive de ser criativo. Esse trabalho, posteriormente, resultou no desenvolvimento de um sistema, em 1983 que, segundo CAMARATTA e FISCHER (2014), supõe-se possuir certa habilidade musical.

Em MCCORMACK ((MCCORMACK, 1996)) foi descrito o desenvolvimento de um sistema de composição algorítmica baseada em L-Systems. Alguns comentários relevantes a respeito da construção e percepção musical são expostos, como:

“It has been observed that almost all forms of music involve repetition [13], either of individual sequences of notes or at some higher levels of structural grouping. Often these repetitions appear at a number of different levels simultaneously. Some compositions repeat patterns that are slightly changed at each repetition, or revolve around some musical ‘theme’, whereby complex harmonic, timing or key shifts in a basic theme provide a pleasing musical diversity.” (MCCORMACK, 1996, p. 2)

Assim como “It is also commonly known that what allows us to identify an individual piece of music is the change in pitch between notes, not the pitch of the notes themselves.”, a abordagem adotada pelo autor para o sistema desenvolvido, foi aplicar na computação musical, gramáticas desenvolvidas para modelagem gráfica de morfogênese¹ e herbáceas². Após implementado, o sistema foi conectado a um sintetizador digital MIDI³. Segundo o autor, após a análise sintática da gramática, ela é capaz de gerar sequências de notas em tempo real.

“The system builds lists of strings. Productions are applied to these strings and the results interpreted as commands that are output as MIDI data to various synthesizers connected to the system. MIDI data can be converted to standard notation, suitable for interpretation by real musicians (as opposed to machines) if required.” (MCCORMACK, 1996, p. 11)

Na época em que foi publicado o sistema ainda estava em fase inicial de desenvolvimento, mas foi importante para a análise do poder das gramáticas em composições musicais.

PAPADOPOULOS e WIGGINS (1999) buscou avaliar o uso de diferentes métodos de inteligência artificial na composição algorítmica, apresentando vantagens e desvantagens de cada método avaliado. Os métodos foram divididos em: modelos matemáticos, sistemas baseados em conhecimento, gramáticas, métodos evolucionários, sistemas que aprendem e sistemas híbridos. Após a análise dos métodos, chegou-se à conclusão de que sistemas baseados em apenas um método não são muito efetivos, devido ao fato de que a maior parte

¹ bio desenvolvimento das formas e estruturas características de uma espécie a partir do embrião; morfogenia.

² 1. Relativo a ervas 2. Planta que não tem parte lenhosa

³ *inf mús* interface (‘elemento’) que permite a conexão de sintetizadores, instrumentos musicais e computadores.

dos métodos gera músicas inexpressivas, pois computadores não possuem sentimentos ou intenções. Ainda segundo os autores, a tendência é que os métodos de composição algorítmica passem a ser mesclados com o objetivo de tirar vantagem das forças de cada um.

Como exemplo recente de pesquisas realizadas em território brasileiro, tem-se os trabalhos de Bruno Basseto (2000), em que foi proposto o uso de gramáticas sensíveis ao contexto, empregadas como modelo, no qual desenvolve-se um formalismo adaptativo, mediante uma variação dos modelos não-determinísticos em rede de Markov, além de ter sido feito um estudo acerca dos sistemas de composição algorítmica e formalização teórica.

Após os experimentos realizados, destacaram-se algumas vantagens no uso das técnicas propostas, são elas:

- Descrição formal de conhecimento;
- Relativa simplicidade da implementação final;
- Tolerância a grandes volumes de informação;
- Aplicação na solução de problemas complexos.

Ainda apresentadas por Basseto (2000), algumas propostas de pesquisa para trabalhos futuros, destacam-se:

- Estabelecimento de paralelos com outros formalismos: em que o autor propõe o estudo da possibilidade de utilizar outros formalismos para solucionar o mesmo problema;
- Integração com outros formalismos: em que o autor propõe utilização de outros formalismos em conjunto com o utilizado na pesquisa;
- Aplicação em outros domínios linguísticos: em que o autor propõe o uso das técnicas estudadas em outras linguagens, como a linguagem natural;
- Modelamento de interfaces homem-máquina mais adequadas: em que o autor propõe a criação de ferramentas que possam ser utilizadas da melhor maneira por outros usuários;
- Aplicação em educação musical: em que o autor propõe o estudo da aplicação da composição algorítmica nas para auxiliar no ensino da música;
- Aplicação em sistemas multimídia: em que o autor propõe estudar o resultado da aplicação de ferramentas do tipo em sistemas multimídia;

- Criação e utilização de meta-instrumentos musicais: que são instrumentos puramente digitais.

3.3 Softwares comerciais

Durante a pesquisa, foi encontrado o *software* Harmonia-Brasil. O *software* é pago e possui uma versão gratuita para testes em *Flash*. Essa análise foi feita com base na versão de testes.

O Harmonia-Brasil tem como principal objetivo guiar o usuário na composição de harmonias. Na tela principal, são apresentados diversos acordes para que o usuário escolha um como inicial. A partir do primeiro acorde, o programa recomenda vários outros possíveis acordes, e assim por diante, até que o usuário não queira mais adicionar acordes à sua harmonia.

É permitido ao usuário clicar em cima do acorde desejado antes de incluí-lo, ao fazer isso, o programa apresenta o som do acorde por meio de um piano sintetizado.

Na funcionalidade "GRAVAR ENCADEAMENTO", o usuário possui uma biblioteca de ritmos à sua disposição, quando este é selecionado, ele passa a ser reproduzido por meio de uma bateria sintetizada, porém não é possível determinar o andamento do ritmo. O usuário, então, pode clicar nos acordes livremente para criar uma harmonia em conjunto com o ritmo escolhido.

O programa tem uma proposta interessante, porém após o uso da ferramenta a interface pareceu confusa ao usuário ao primeiro contato e pouco intuitiva, e embora o programa faça sugestões ao usuário na hora de compor a música, ele não é capaz de gerar harmonias automaticamente e algumas sugestões de acordes se mostraram fora do campo harmônico durante o uso.

Os pontos fortes do *software* são:

- As sugestões de acordes fazem com que o usuário aprenda as possibilidades de uso de acordes numa progressão harmônica;
- É possível ouvir o som de todos os acordes.

3.4 Considerações finais

A análise histórica da composição algorítmica se mostrou essencial para guiar o desenvolvimento desse estudo.

Após análise de trabalhos anteriores, comprova-se que a utilização de computadores para a composição musical não é assunto recente, porém com pouco avanço. Pode-se levar

em consideração diversos aspectos das pesquisas apresentadas, para combinar aspectos dos seus desenvolvimentos e buscar a apresentação de novas ferramentas, que possam ser desenvolvidas em prol da comunidade, estimulando o aprendizado da música, auxiliando na composição e improvisação e se integrando com outras ferramentas, para gerar novos conteúdos na área da música.

A análise dos trabalhos relacionados mostrou que as técnicas utilizadas anteriormente não foram suficientes para solucionar completamente o problema, fato que indica que novas abordagens precisam ser exploradas com a finalidade de construir ferramentas que consigam sumilar com maior fidelidade a capacidade humana de compor, e por esse motivo para o desenvolvimento da ferramenta proposta nesse trabalho a combinação de mais de um método de composição algorítmica mostrou-se a melhor maneira de encarar o problema, logo foram utilizados dois modelos de composição algorítmica são eles:

- Sistemas baseados em conhecimento;
- Gramáticas.

Dessa maneira, é possível fazer a análise de composições musicais conhecidas de modo a identificar características harmônicas que podem ser utilizadas para gerar novas composições do mesmo estilo, por meio da criação de gramáticas específicas para cada estilo musical.

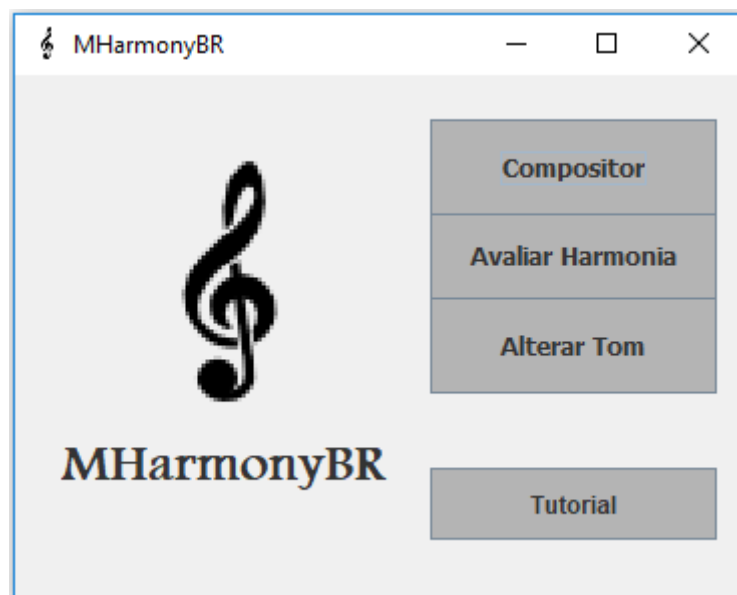
4 MHarmonyBR

O presente trabalho buscou estudar a construção harmônica de músicas de estilos musicais distintos, e com base nas suas características, desenvolver um sistema capaz de reproduzir essas características harmônicas em novas composições. O aplicativo desenvolvido neste estudo tem como função principal receber do usuário um estilo musical, um tom e a quantidade de sequências de acordes que ele deseja e recebe como resposta uma harmonia em forma de cifras. O usuário deve selecionar os atributos em uma lista pré-definida.

Na tela inicial (Figura 3), o destaque é dado para as funcionalidades da ferramenta, contidas no menu composto por botões no canto direito, em especial para a principal funcionalidade, que é o "Compositor"(Figura 4).

O botão mais afastado (Tutorial) deverá apresentar uma tela contendo um manual de uso, quando pressionada. Até o término do estudo, não foi possível a confecção do manual.

Figura 3 – Tela inicial do MHarmonyBR



Fonte: Próprio autor

No "Compositor"(Figura 4), o usuário tem a possibilidade de escolher o estilo, o tom e a quantidade de sequências da harmonia que será criada, no qual a sequência é limitada a no máximo 6, para não haver a necessidade de uma barra de rolagem na área de texto que apresenta o resultado, evitando que o usuário tenha que parar de tocar a harmonia para usar a rolagem do programa. Nos botões "Gerar Harmonia" e "Voltar", foram aplicadas técnicas de IHC (Interação Homem-Computador), apresentando o botão "Gerar Harmonia",

com destaque, para que o usuário seja incentivado a continuar o processo e reduzindo a possibilidade de haver cancelamento indesejado do processo ao reduzir o tamanho do botão “Voltar”. Ao clicar em “Gerar em Harmonia”, o resultado é apresentado na mesma tela em uma área de texto, e caso o usuário queira voltar à tela inicial do aplicativo, basta clicar em “Voltar”.

Figura 4 – Tela da função Compositor, do MharmonyBR

Fonte: Próprio autor

Após clicar no botão "Avaliar Harmonia" o programa exibirá a tela da figura 5, o usuário deve digitar uma sequência de acordes com no mínimo 2 acordes separados por espaço em formato de cifras, caso algum dos requisitos não seja atendido, uma mensagem de advertência contendo a descrição do erro é apresentada na tela em forma de Pop-up. Após clicar em “Analisar”, o usuário obtém as informações “tom da música” e “progressão harmônica” referentes à harmonia passada. Nos botões “Analisar” e “Voltar”, foram aplicadas técnicas de IHC (Interação Humano-Computador), apresentando o botão “Analisar” com destaque, para que o usuário seja incentivado a continuar o processo e reduzindo a possibilidade de haver cancelamento indesejado do processo ao reduzir o tamanho do botão “Voltar”.

Na funcionalidade "Avaliar Harmonia"(Figura 5), foram utilizados os algoritmos de análise de tom e análise de progressão harmônica que serão apresentados no Capítulo 5.

Por fim, após clicar no botão Alterar Tom presente na Figura 3, é apresentada a tela da (Figura 6), que do mesmo modo que o "Avaliar Harmonia", utiliza um campo de texto para que o usuário digite a harmonia, que deve ser uma harmonia válida que cumpra os mesmos requisitos do "Avaliar Harmonia". Após digitar a harmonia, o usuário deverá indicar o tom para o qual a harmonia será transformada. O resultado deverá ser mostrado

Figura 5 – Tela da função Avaliar Harmonia, do MharmonyBR

Fonte: Próprio autor

na área de texto rotulada “Harmonia”. Nos botões “Mudar Tom” e “Voltar”, foram aplicadas técnicas de IHC, apresentando o botão “Mudar Tom” com destaque, para que o usuário seja incentivado a continuar o processo e reduzindo a possibilidade de haver cancelamento indesejado do processo ao reduzir o tamanho do botão “Voltar”.

Figura 6 – Tela da função Alterar Tom, do MharmonyBR

Fonte: Próprio autor

A seguir, é apresentado um caso de uso do "Compositor"(Figura 4):

- O usuário seleciona o estilo Sertanejo Universitário;

- O usuário seleciona o tom G (Sol Maior);
- O usuário escolhe a opção de 3 sequências de acordes;
- O usuário clica num botão para iniciar a composição;
- O aplicativo apresenta a seguinte resposta:
 - Sequência 1: Em C G D, sequência 2: Am D Em C, sequência 3: Am G Am.

4.1 Roteiro de pesquisa

4.1.1 Seleção das músicas

As músicas que serviram como base para a obtenção do conhecimento necessário para o desenvolvimento do trabalho foram selecionadas com base em pesquisas em listas de hits do estilo Sertanejo Universitário e do estilo Punk Rock. Dentro dessas listas 20 músicas de cada estilo foram escolhidas de maneira aleatória. Após análise das primeiras amostras, foi verificada a necessidade da inclusão de um novo estilo e da adição de mais músicas com o objetivo de verificar o impacto na distinção de composição harmônica entre os estilos analisados, então foram incluídas no estudo 50 músicas do estilo Reggae e adicionadas 30 músicas ao estilo Sertanejo Universitário, os estilos foram escolhidos buscando a maior distinção possível em relação à sua origem cultural, não havendo relação entre eles.

As musicas foram selecionadas usando o seguinte algoritmo:

- Foi feita uma busca no Google utilizando os termos "Musicas Punk Rock", "Musicas Sertanejo Universitário" e "Músicas Reggae";
- As musicas apresentadas pelo Google e as musicas 3 primeiras listas apresentadas como resultado da busca foram colocadas em três planilhas (uma para cada estilo);
- Para definir quais seriam as musicas utilizadas, foi utilizado o método *Math.random()* da linguagem Java. Para cada número obtido pelo método, era selecionada uma música na planilha de acordo com o número da linha correspondente ao numero retornado pelo método Java;

4.1.2 Análise dos dados

A coleta dos dados harmônicos foi feita de forma manual, pois todas as harmonias foram encontradas em buscas em sites de cifras e validadas pelo autor executando as harmonias com o violão. Nos casos em que a harmonia encontrada nas buscas não estava

correta (tomando como base o campo harmônico e comparação auditiva em relação ao áudio original), fez-se necessária a transcrição, realizada por parte do autor, da música com base no áudio original. A segunda parte da obtenção dos dados foi auxiliada por algoritmos desenvolvidos para esse propósito específico. Com esses algoritmos, descritos no Capítulo 5, foi possível determinar o tom e as progressões harmônicas das músicas de maneira rápida.

4.1.3 Desenvolvimento e testes

As informações extraídas dos dados coletados foram utilizadas como base para o desenvolvimento de um *software* de composição automática, que foi desenvolvido em linguagem de programação Java ¹.

Os resultados da pesquisa foram validados por meio de testes, os quais buscaram verificar se as harmonias geradas pelo *software* eram criadas de acordo com as regras que foram estabelecidas na criação das gramáticas e do algoritmo de composição. Devido a um dos focos da pesquisa ter sido o de desenvolver um *software* para o usuário comum e músico iniciante se fez necessário a aplicação de um teste de usabilidade, que foi realizado com auxílio de um questionário de experiência de uso, enviado para um público selecionado em grupos do Facebook, após entrevista detalhada na Seção 8.3.

Nos próximos capítulos serão apresentadas todas as etapas do estudo e do desenvolvimento do aplicativo MHarmonyBR, assim como a organização da estrutura do código, a implementação e comentários a respeito dos algoritmos, análise das amostras dos estilos musicais, interface gráfica do aplicativo e testes de validação.

¹ Java é uma linguagem de programação interpretada orientada a objetos

5 Algoritmos Básicos

Nesse capítulo são apresentados os códigos Java dos algoritmos básicos. Mais a frente será abordado o método para criar gramáticas de acordo com o estilo musical escolhido.

Os algoritmos descritos nesse capítulo foram utilizados como base para toda a análise e desenvolvimento da ferramenta MHarmonyBR. Inicialmente foram desenvolvidos dois algoritmos:

- Algoritmo de detecção de tom;
- Algoritmo análise de progressão harmônica;

Não houve pesquisa para determinar se esses algoritmos já foram implementados em algum outro trabalho, pois os problemas os quais são tratados por esses algoritmos já foram resolvidos por conceitos básicos da teoria musical, nesse estudo ambos foram implementados baseados nesses conceitos (definição de tom e progressão harmônica).

Antes de discorrer a respeito dos algoritmos citados acima, deve ser apresentado como foi feita a organização no código, dos elementos da gramática utilizada, para que os algoritmos possam fazer as análises de maneira correta.

Foram colocados os acordes em um HashMap como apresentado no código da (Figura 7). Essa escolha de projeto foi feita para facilitar a identificação dos acordes passados pelo usuário em forma de notação de cifras, por meio de uma conversão para inteiro, onde os algoritmos tratam cada acorde do campo harmônico de acordo com o seu valor. Também foi construído um mapa invertido para facilitar a conversão de maneira inversa, quando o *software* precisar apresentar uma resposta em forma de notação musical para o usuário.

Os algoritmos podem verificar a existência do acorde em cada campo harmônico, os quais foram organizados em um *array* bidimensional de inteiros da maneira utilizada na Figura 8 (já utilizando os valores correspondentes para cada acorde).

Apresentados os trechos de código na Figura 7 e Figura 8, é possível retomar o estudo dos algoritmos básicos.

O algoritmo de análise de tom (Figura 9) foi desenvolvido inicialmente para detectar em qual campo harmônico está situada uma determinada sequência de acordes, tal sequência deve ser passada para o programa de forma que não haja modulações tonais (automaticamente classificando-a em mais de um tom), isso facilitou a análise das músicas escolhidas para gerar uma base de conhecimento, pois o algoritmo agilizou o processo de

Figura 7 – Trecho de código contendo acordes colocados em um HashMap

```
public class MapaAcordes {

    public static Map<String, Integer> map;

    static{

        map = new HashMap<String, Integer>();

        map.put("C", 1);
        map.put("Cm", 2);
        map.put("C#", 3);
        map.put("C#m", 4);
        map.put("C#dim", 26);
    }
}
```

Fonte: Próprio autor

Figura 8 – Trecho de código contendo o *array* correspondente ao campo harmônico

```
int[][] campoMaior = {{1 , 6 , 10, 11, 15, 20, 36}, //C
    {3 , 8 , 12, 13, 17, 22, 25}, //C#
    {5 , 10, 14, 15, 19, 24, 26}, //D
    {7 , 12, 16, 17, 21, 2 , 27}, //D#
    {9 , 14, 18, 19, 23, 4 , 28}, //E
    {11, 16, 20, 21, 1 , 6 , 29}, //F
    {13, 18, 22, 23, 3 , 8 , 30}, //F#
    {15, 20, 24, 1 , 5 , 10, 31}, //G
    {17, 22, 2 , 3 , 7 , 12, 32}, //G#
    {19, 24, 4 , 5 , 9 , 14, 33}, //A
    {21, 2 , 6 , 7 , 11, 16, 34}, //A#
    {23, 4 , 8 , 9 , 13, 18, 35}}; //B
```

Fonte: Próprio autor

detecção de mudanças de tom dentro de uma mesma música, essa informação será usada mais à frente para extrair informações dos dados obtidos.

O algoritmo, após conversão de uma string contendo uma harmonia, recebe um *ArrayList* contendo uma sequência de acordes, então faz buscas no *array* array que contém o campo harmônico, a busca termina quando o algoritmo encontra um campo harmônico que contenha todos os acordes da progressão passada pelo usuário, retornando o acorde correspondente ao grau tônico (tom da sequência de acordes) do campo, ou retornando 0, quando a sequência de acordes não se encaixa em um campo harmônico. O algoritmo de análise de progressão harmônica (Figura 10) foi desenvolvido para analisar (após o algoritmo de análise de tom identificar em qual campo harmônico está a sequência de acordes) qual é a progressão harmônica correspondente à harmonia passada pelo usuário, de acordo com o campo harmônico ao qual ela pertence.

Assim como o algoritmo de análise de tom, o uso desse algoritmo facilitou a análise das músicas escolhidas para gerar uma base de conhecimento

Figura 9 – Trecho de código contendo o método Java responsável por analisar o tom da harmonia

```
public static int defineTom(ArrayList<Integer> sequencia){

    int resultado = 0;

    int controle = 1;

    while(controle != 0 && resultado==0) {
        for (int j = 0; j < CampoHarmonico.campoHarmonico().length && resultado ==0; j++){
            int match = 0;
            int i = 0;
            int l = 0;

            while( l < 7 && resultado ==0){
                if(i < sequencia.size()){
                    if(sequencia.get(i)==CampoHarmonico.campoHarmonico()[j][l]) {
                        i++;
                        l=-1;
                        match++;
                    }
                }
                if (match == sequencia.size()){
                    resultado = CampoHarmonico.campoHarmonico()[j][0];
                }
                l++;
            }

            controle = 0;
        }

        return resultado;
    }
}
```

Fonte: Próprio autor

O algoritmo recebe, após conversão de uma string contendo uma harmonia, um *ArrayList* contendo uma sequência de acordes, faz a chamada do método que contém o algoritmo de análise de progressão.

Todos os algoritmos apresentados nessa seção foram reaproveitados posteriormente para auxiliar na composição musical ou como funcionalidade da ferramenta. Como já foi citado, os algoritmos apresentados nessa seção facilitaram a coleta dos dados das músicas contidas na amostra dos estilos musicais. No Capítulo 4 foi detalhada a utilidade desses algoritmos básicos na ferramenta produzida por esse estudo.

Figura 10 – Trecho de código contendo o método Java responsável por analisar a progressão harmônica

```
private static ArrayList<Integer> defineProgressao(ArrayList<Integer> sequencia){  
    ArrayList<Integer> progressao = new ArrayList<Integer>();  
  
    int tom = defineTom(sequencia);  
    int i = 0;  
    int match = -1;  
  
    if(tom != 0 ){  
  
        while(match == -1 && i < CampoHarmonico.campoHarmonico().length){  
            if(CampoHarmonico.campoHarmonico()[i][0] == tom){  
                match = i;  
            }  
            i++;  
        }  
  
        for(int j = 0; j < sequencia.size(); j++){  
            for(int l = 0; l < 7; l++){  
                if(sequencia.get(j)==CampoHarmonico.campoHarmonico()[match][l]){  
                    progressao.add(j, l+1);  
                }  
            }  
        }  
  
        return progressao;  
    }else{  
        return null;  
    }  
}
```

Fonte: Próprio autor

6 Base de Músicas

Para prosseguir com o desenvolvimento da ferramenta, é preciso definir como será feita a construção das harmonias geradas pelo programa. Para isso, foram analisadas um total de 40 composições já existentes e popularmente conhecidas de dois estilos musicais (20 músicas de cada estilo), são eles:

- Sertanejo Universitário – Estilo popularmente conhecido no Brasil;
- Punk Rock – Estilo conhecido internacionalmente.

Dessas análises, foram retirados dados harmônicos de cada estilo musical para posterior identificação de características harmônicas de cada estilo. A lista completa das músicas analisadas pode ser vista no Anexo A. Utilizando os algoritmos básicos apresentados no Capítulo 5 foi possível a obtenção de dados no formato exemplificado na Tabela 4. Todas as músicas tiveram sua harmonia manualmente particionada, seguindo o critério da repetição das progressões harmônicas, com o objetivo de facilitar a análise dos padrões das progressões. Observando a Tabela 4 é possível ver diferentes possibilidades de construção em uma música:

- Música 1: possui duas progressões harmônicas, das quais a primeira se encontra no campo harmônico de D (Ré maior) e a segunda em C (Dó maior), que significa que houve modulação na harmonia da música.
- Música 2: apresenta duas progressões harmônicas, porém não há modulação na harmonia, que significa que o tom se mantém o mesmo durante toda a música (Dó Maior).
- Música 3: possui apenas uma progressão harmônica, que se repete durante todo o decorrer da música.

Existem casos em que a música possui apenas uma progressão harmônica, mas sofre uma mudança de tom no meio da harmonia (nesses casos, a progressão foi contabilizada apenas uma vez). Uma música pode conter quantas progressões harmônicas seu compositor desejar.

As informações extraídas da base de dados foram obtidas por observação de forma manual. Esses dados são apresentados nas próximas seções.

Tabela 4 – Exemplo de tabela de dados harmônicos de músicas

MÚSICA	ESTILO	TOM	PROGRESSÃO 1	TOM	PROGRESSÃO 2
MÚSICA 1	PUNK	D	[1, 4, 1]	C	[5, 4, 5]
MÚSICA 2	PUNK	C	[1, 4, 3]	C	[5, 4, 3, 2, 1]
MÚSICA 3	PUNK	F	[6, 1, 4, 5]	-	-

Fonte: Próprio autor

6.1 Primeira análise de músicas da base

A primeira análise das músicas contidas na base foi feita utilizando todas as músicas do estilo Punk Rock e as primeiras 20 músicas do estilo Sertanejo Universitário descritas no Anexo A.

- Das músicas de ambos os estilos, foi obtida uma soma bruta de 79 progressões harmônicas, das quais 59 delas eram distintas e 49 delas tiveram apenas uma aparição na amostra, em músicas distintas.
- Apenas uma das progressões harmônicas contidas em um estilo está contida no outro.
- Das músicas do estilo Punk Rock foi obtida uma soma bruta de 41 progressões harmônicas, das quais 34 delas eram distintas e 29 delas tiveram apenas uma aparição na amostra.
- As progressões do tipo [6, 4, 5], [6, 1, 4, 5], [1, 5, 4, 5], [1, 2, 4, 5, 1] e [3, 1, 5, 6], apareceram mais de uma vez na amostra, em músicas distintas do estilo Punk Rock, das quais a progressão [6, 4, 5] teve 4 aparições e as demais tiveram 2 aparições (Tabela 5).

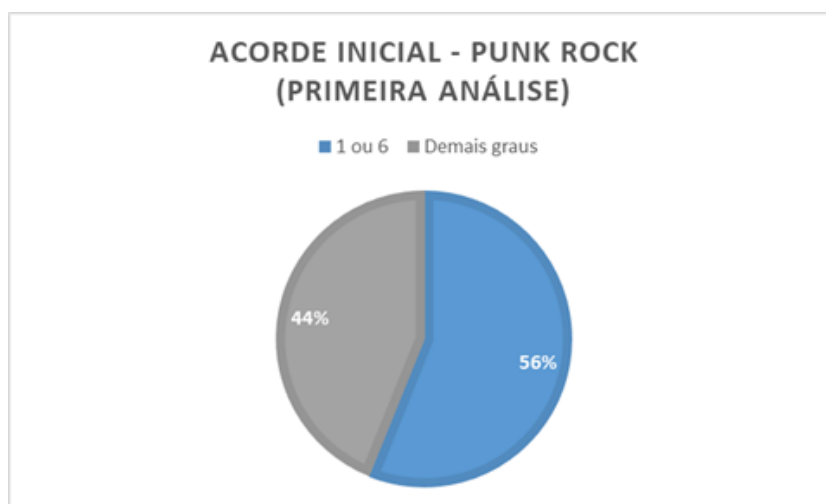
Tabela 5 – Número de aparições que se repetiram dentro da amostra - Punk Rock

PROGRESSÃO	NUMERO DE APARIÇÕES NA AMOSTRA
[6, 4, 5]	4
[6, 1, 4, 5]	2
[1, 5, 4, 5]	2
[1, 2, 4, 5, 1]	2
[3, 1, 5, 6]	2
Demais progressões	1

Fonte: Próprio autor

- Do total de 41 progressões do estilo Punk Rock, 23 (56%) começam pelo grau tônico ou relativo menor (Figura 11).

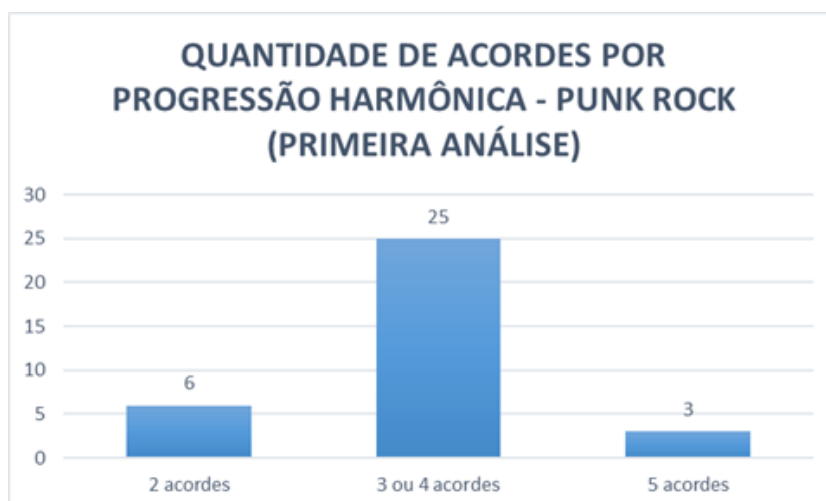
Figura 11 – Porcentagem de progressões do estilo Punk Rock que iniciam com os graus 1 e 6 do campo harmônico - primeira análise



Fonte: Próprio autor

- Das 34 diferentes sequências de acordes da amostra do estilo Punk Rock, apenas 3 (8,8%) contém 5 ou mais acordes distintos e 6 (17,6%) contém 2 acordes distintos. As demais sequências possuem 3 ou 4 acordes (Figura 12).

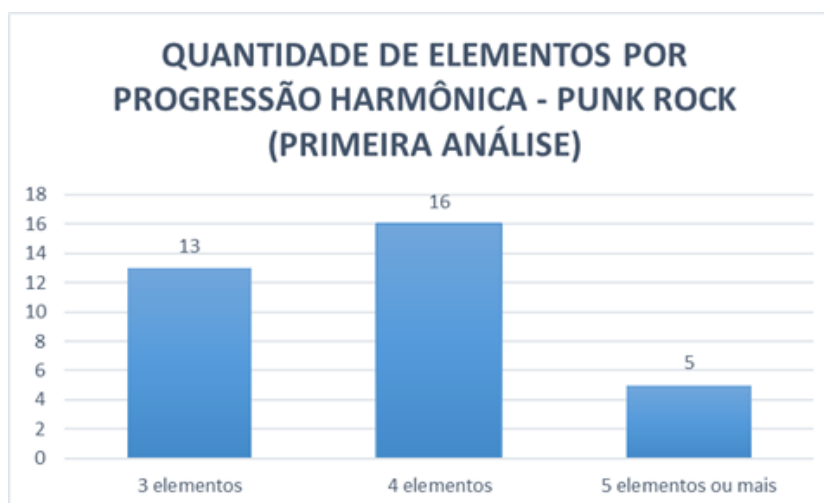
Figura 12 – Quantidade de acordes contidos nas progressões do estilo Punk Rock - primeira análise



Fonte: Próprio autor

- Apenas 5 (14,7%) das 34 diferentes progressões harmônicas possui 5 ou mais elementos na sua estrutura (Figura 13).

Figura 13 – Quantidade de elementos contidos nas progressões do estilo Punk Rock - primeira análise



Fonte: Próprio autor

- Apenas uma música do estilo Punk Rock possui modulação tonal.
- Das músicas do estilo Sertanejo Universitário foi obtida uma soma bruta de 38 progressões harmônicas, das quais 25 delas eram distintas e 19 delas tiveram apenas uma aparição na amostra.
- As progressões do tipo [1, 5, 6, 4], [6, 4, 1, 5], [1, 5, 4, 1], [2, 4, 1, 5] e [1, 5, 2, 4], apareceram mais de uma vez na amostra, em músicas distintas do estilo Sertanejo Universitário, das quais a progressão [1, 5, 6, 4] teve 6 aparições, a progressão [6, 4, 1, 5] teve 5 aparições, a progressão [2, 4, 1, 5] teve 3 aparições e as demais tiveram 2 aparições (Tabela 6).

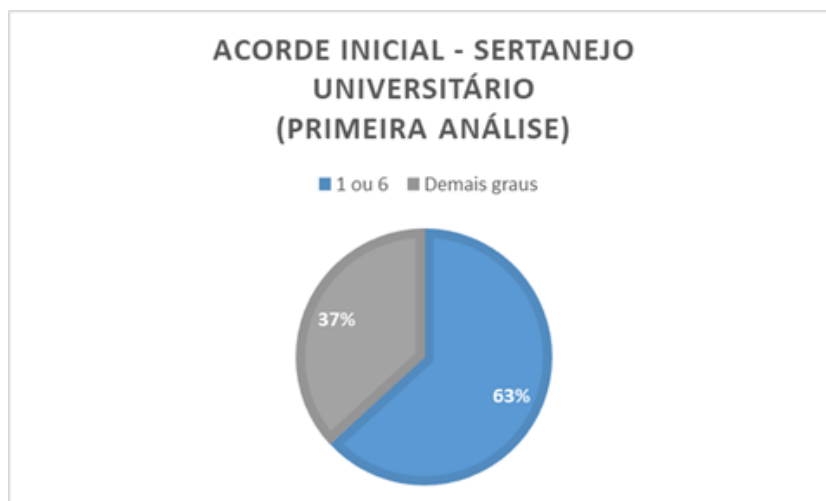
Tabela 6 – Número de aparições de progressões dentro da amostra - Sertanejo Universitário

PROGRESSÃO	NUMERO DE APARIÇÕES NA AMOSTRA
[1, 5, 6, 4]	6
[6, 4, 1, 5]	5
[2, 4, 1, 5]	3
[1, 5, 2, 4]	2
[1, 5, 4, 1]	2
Demais progressões	1

Fonte: Próprio autor

- Do total de 38 progressões do estilo Sertanejo Universitário, 24 (63%) começam pelo grauônico ou relativo menor (Figura 14).

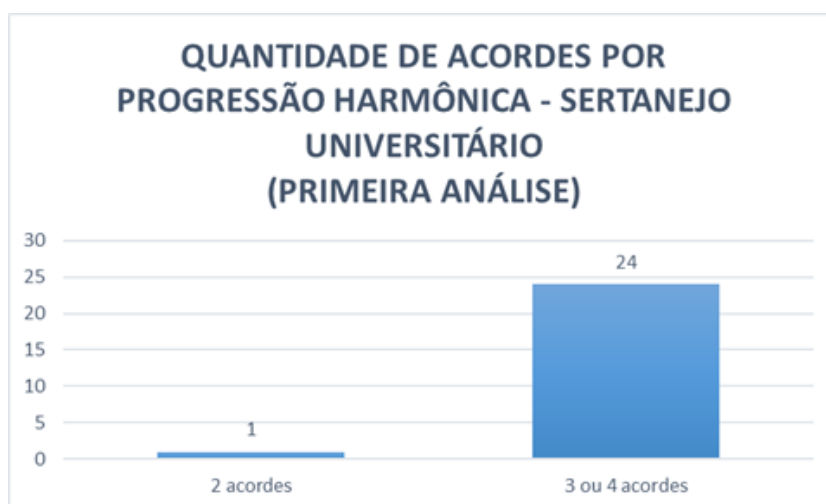
Figura 14 – Porcentagem de progressões do estilo Sertanejo universitário que iniciam com os graus 1 e 6 do campo harmônico - primeira análise



Fonte: Próprio autor

- Das 25 diferentes sequências de acordes da amostra do estilo Sertanejo Universitário, nenhuma contém 5 ou mais acordes e apenas 1 (4,1%) contém 2 acordes (Figura 15).

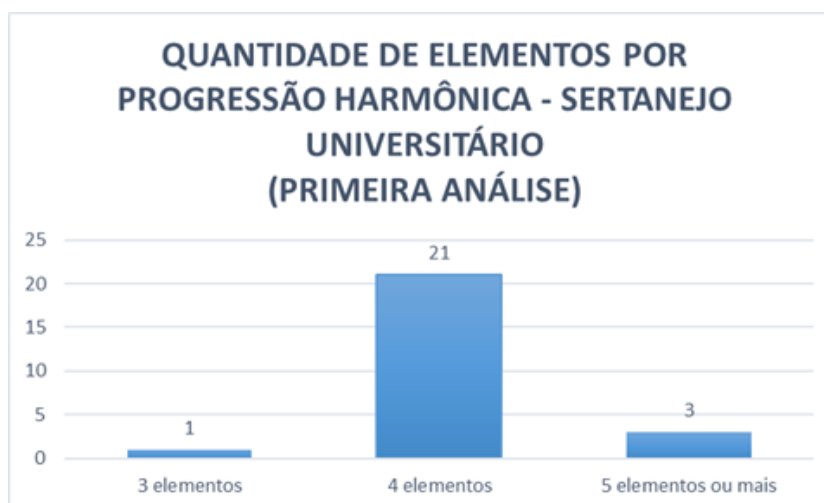
Figura 15 – Quantidade de acordes contidos nas progressões do estilo Sertanejo Universitário - primeira análise



Fonte: Próprio autor

- Apenas 1 (4,1%) das 25 diferentes progressões harmônicas possui 3 elementos na sua estrutura e apenas 3 (12,5%) progressões harmônicas possuem 5 ou mais elementos (Figura 16).
- Nenhuma música do estilo Sertanejo Universitário possui modulação tonal.

Figura 16 – Quantidade de elementos contidos nas progressões do estilo Sertanejo Universitário - primeira análise



Fonte: Próprio autor

6.2 Segunda análise de músicas da base

Após a realização dos testes descritos na Seção 8.2.1 e devido às composições de Punk Rock serem popularmente conhecidas por não seguirem padrões bem definidos próprios, foi verificada a necessidade da adição de mais músicas e da adição de um novo estilo musical, com o objetivo de obter conclusões mais claras a respeito da capacidade do algoritmo de classificação distinguir os estilos musicais. Foram acrescentadas 30 músicas ao estilo Sertanejo Universitário, resultando num total de 50 músicas, e também foram adicionadas 50 músicas do estilo Reggae. Os dados da nova análise são apresentados a seguir.

- Após adição de novas músicas, das 136 progressões harmônicas identificadas entre os estilos Punk Rock, Sertanejo Universitário e Reggae, 3 foram comuns a todos os estilos musicais do estudo, 9 progressões contidas no estilo Reggae também estão contidas no estilo Punk Rock e 4, no estilo Sertanejo Universitário, e 5 das progressões contidas no estilo Sertanejo Universitário também estão no estilo Punk Rock. Isso mostra que com a adição de mais amostras, existe a possibilidade de que mais características harmônicas sejam comuns aos estilos.
- Das 50 músicas do estilo Sertanejo Universitário, foi obtida uma soma bruta de 106 progressões harmônicas, das quais 58 são distintas e 43 delas tiveram apenas uma aparição na amostra.
- As progressões do tipo [1, 5, 6, 4], [2, 4, 1, 5], [1, 5, 2, 4], [6, 4, 1, 5], [4, 5, 6], [6, 4, 5, 6], [6, 4, 5, 3], [1, 5, 4, 1], [1, 6, 4, 5], [6, 1, 5, 2], [4, 1, 5, 6], [4, 5, 6, 1], [6, 5, 2, 4], [4, 5, 4, 5] e [2, 4, 1], apareceram mais de uma vez na amostra, em músicas distintas do estilo

Sertanejo Universitário, das quais a progressão [1, 5, 6, 4] teve 12 aparições, [2, 4, 1, 5] e [1, 5, 2, 4] tiveram 9 aparições, [6, 4, 1, 5] teve 8 aparições, [4, 5, 6] teve 4 aparições, [6, 4, 5, 6] teve 3 aparições e os demais tiveram 2 aparições cada (Tabela 7).

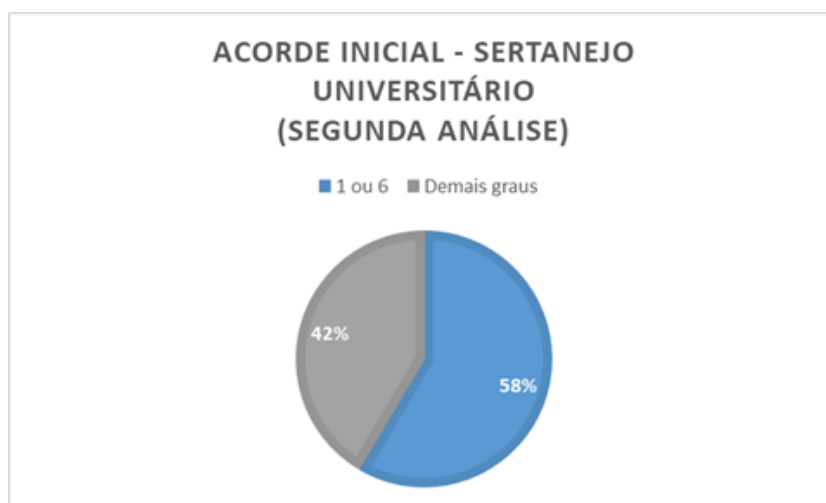
Tabela 7 – Número de aparições de progressões que se repetiram dentro da amostra após ampliação-Sertanejo Universitário

PROGRESSÃO	NUMERO DE APARIÇÕES NA AMOSTRA
[1, 5, 6, 4]	12
[2, 4, 1, 5]	9
[1, 5, 2, 4]	9
[6, 4, 1, 5]	8
[4, 5, 6]	4
[6, 4, 5, 6]	3
Demais progressões	2

Fonte: Próprio autor

- Do total de 106 progressões do estilo Sertanejo Universitário, 62 (58%) começam pelo grau tônico ou relativo menor (Figura 17).

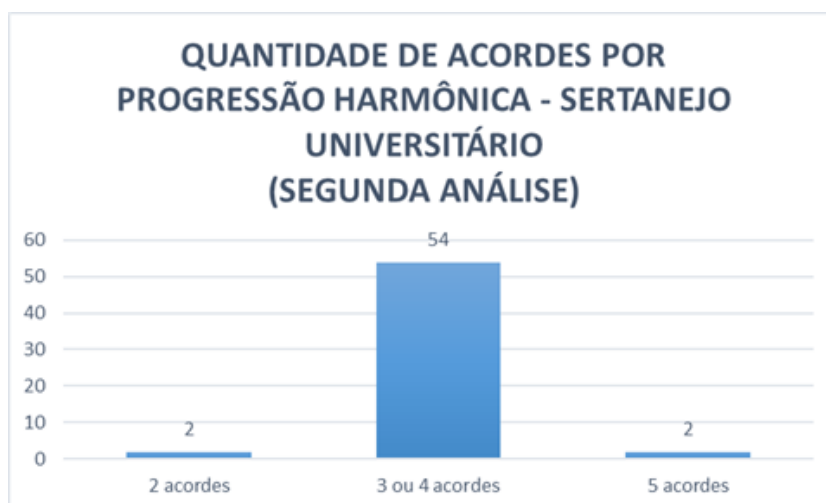
Figura 17 – Porcentagem de progressões do estilo Sertanejo Universitário que iniciam com os graus 1 e 6 do campo harmônico, após ampliação da amostra



Fonte: Próprio autor

- Das 58 diferentes sequências de acordes da amostra do estilo Sertanejo Universitário, apenas 2 (3,4%) contém 5 ou mais acordes distintos, e apenas 2 (3,4%) contém somente 2 acordes distintos. As demais sequências possuem 3 ou 4 acordes (Figura 18).

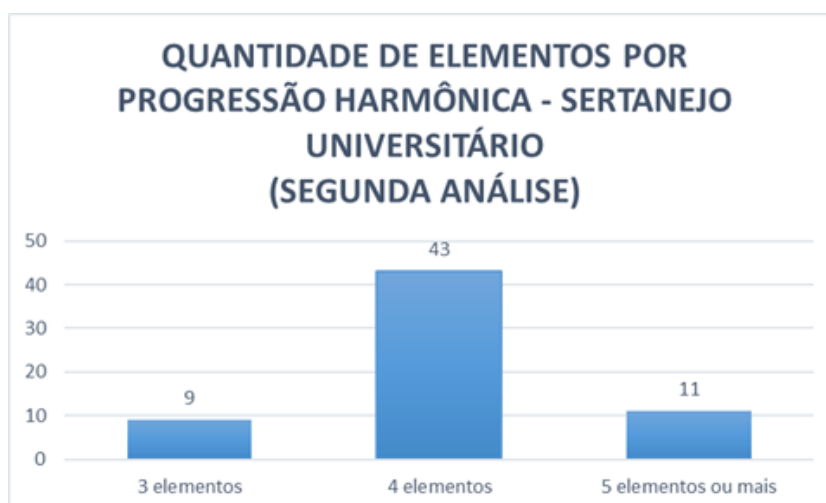
Figura 18 – Quantidade de acordes contidos nas progressões do estilo Sertanejo Universitário, após ampliação da amostra



Fonte: Próprio autor

- Apenas 11 (18,9%) das 58 diferentes progressões harmônicas possuem 5 ou mais elementos na sua estrutura e 9 (15,5%) progressões possuem 3 elementos (Figura 19).

Figura 19 – Quantidade de elementos contidos nas progressões do estilo Sertanejo Universitário, após ampliação da amostra



Fonte: Próprio autor

- Apenas uma música do estilo Sertanejo Universitário possui modulação tonal.
- Das 50 músicas do estilo Reggae foi obtida uma soma bruta de 84 progressões harmônicas, das quais 63 delas são distintas e 52 tiveram apenas 1 aparição na amostra.
- As progressões do tipo [3, 2, 3, 2], [6, 3, 6, 3], [5, 1, 5, 1], [1, 5, 6, 4], [6, 4, 5], [1, 4, 1], [2, 3, 6], [3, 6, 3, 6], [1, 4, 1, 4], [5, 4, 5, 4] e [6, 2, 6, 2] apareceram mais de uma vez

na amostra, em músicas distintas do estilo Reggae, das quais as progressões [3, 2, 3, 2], [6, 3, 6, 3] tiveram 5 aparições, [5, 1, 5, 1] teve 4 aparições, [1, 5, 6, 4], [6, 4, 5] tiveram 3 aparições e as demais tiveram 2 aparições (Tabela 8).

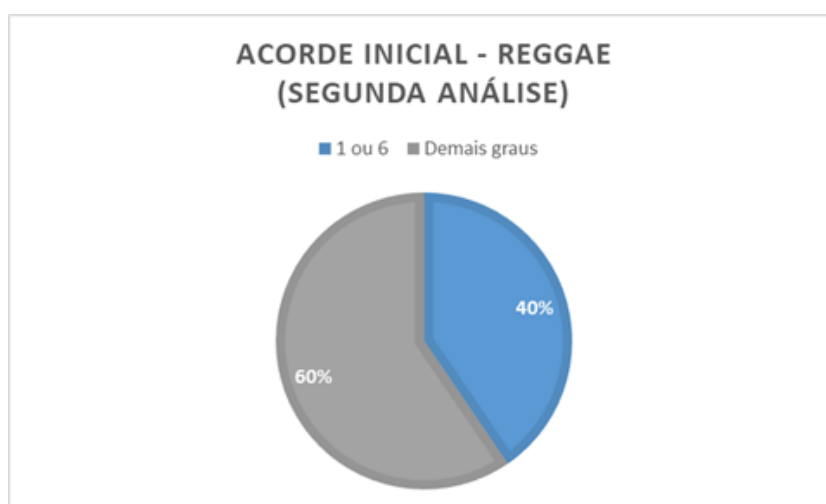
Tabela 8 – Número de aparições de progressões que se repetiram dentro da amostra- Reggae

PROGRESSÃO	NUMERO DE APARIÇÕES NA AMOSTRA
[3, 2, 3, 2]	5
[6, 3, 6, 3]	5
[5, 1, 5, 1]	4
[1, 5, 6, 4]	3
[6, 4, 5]	3
Demais progressões	2

Fonte: Próprio autor

- Do total de 84 progressões harmônicas do estilo Reggae, 34 (40%) começam pelo grau tônico ou relativo menor (Figura 20).

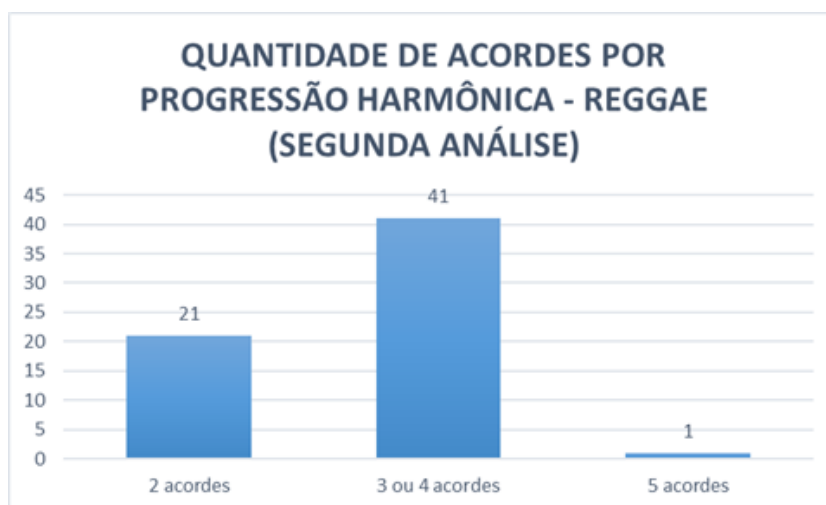
Figura 20 – Porcentagem de progressões do estilo Reggae que iniciam com os graus 1 e 6 do campo harmônico



Fonte: Próprio autor

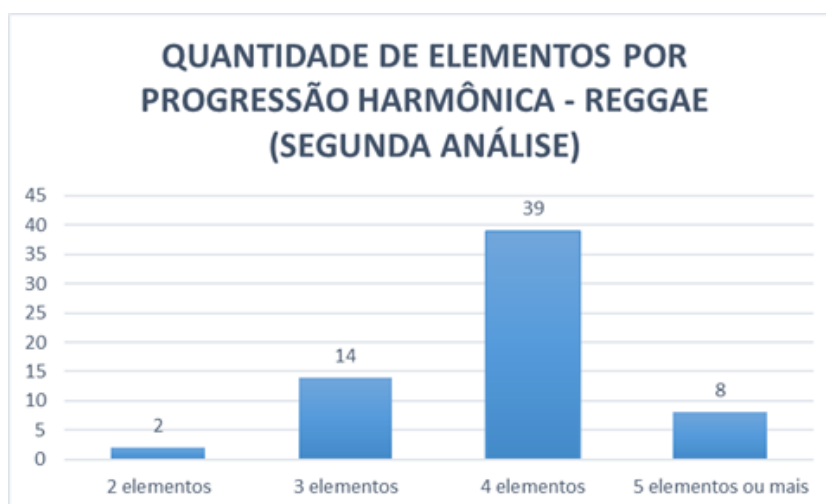
- Das 63 diferentes seqüências de acordes da amostra do estilo Reggae, apenas 1 (1,58%) contém 5 ou mais acordes distintos, 21 (33,3%) seqüências possuem apenas 2 acordes distintos e as demais possuem 3 ou 4 acordes (Figura 21).
- Apenas 8 (12,7%) das 63 progressões harmônicas, possuem 5 ou mais elementos na sua estrutura, 14 (22,2%) progressões possuem 3 elementos e apenas 2 progressões possuem 2 (3,1%) elementos na sua estrutura (Figura 22).

Figura 21 – Quantidade de acordes contidos nas progressões do estilo Reggae



Fonte: Próprio autor

Figura 22 – Quantidade de elementos contidos nas progressões do estilo Reggae



Fonte: Próprio autor

- Apenas 3 músicas do estilo Reggae possuem modulação tonal.

Diante das informações extraídas, tem-se a possibilidade de avançar com o trabalho e o estudo passa para a fase à qual foram feitas decisões no desenvolvimento, a fim de permitir que o programa faça a composição das harmonias.

7 Desenvolvimento

Nesse capítulo é feita uma descrição das etapas de desenvolvimento referentes ao algoritmo de composição do MHarmonyBR. Foram descritas a elaboração das gramáticas utilizadas e a implementação dos algoritmos responsáveis por gerar as harmonias automaticamente.

7.1 Restrição de gramática

Como foi visto na Seção 2.2, houve a confecção de uma gramática que permite a construção de harmonias em qualquer campo harmônico, embora essa gramática já possa ser utilizada para compor qualquer harmonia, esse não é o foco do estudo, o qual busca gerar harmonias que tenham identificação com um determinado estilo musical. Tendo essa ideia em mente, existem diversas possibilidades de gramáticas a serem criadas para cada estilo, porém após a análise das informações apresentadas na Seção 6.1 três gramáticas, que terão suas escolhas justificadas mais a frente, foram criadas.

- Gramática para Punk Rock (Figura 23).
- gramática para Sertanejo Universitário (Figura 24)
- gramática para Reggae (Figura 25)

De acordo com as informações obtidas das amostras, as gramáticas apresentadas nessa seção nas figuras 23, 24 e 25 foram feitas de maneira a separar terminais em cada variável de acordo com características de cada um de modo a dar um peso a cada uma das variáveis para determinar qual será escolhida pelo algoritmo de composição.

A variável A de todas as gramáticas contém as progressões harmônicas que comecem com o grau tônico (grau 1) ou grau relativo menor (grau 6), pois foi observado uma dominância dessa característica nos três estilos musicais, em contraponto, devido a incidência baixa de progressões com 5 ou mais acordes distintos, com 5 ou mais elementos e com apenas 2 acordes distintos (apenas para Punk Rock e Sertanejo Universitário), todas as progressões com essas características fora da curva foram agrupadas como terminais das variáveis C e D das gramáticas, não necessariamente nessa ordem.

Na variável B das gramáticas para Punk Rock e Sertanejo Universitário se encontram as progressões que não iniciam com o grau tônico ou relativo menor, mas que possuem 3 ou 4 acordes distintos na sua estrutura e que são compostas por 3 ou 4 elementos.

Figura 23 – Gramática livre de contexto para o estilo Punk Rock

$G_{PUNK} = (\{S, A, B, C, D\}, T, S, P)$ tal que,
 $T = \{[6\ 4\ 5\ 1], [6\ 1\ 4\ 5], [1\ 2\ 4\ 5], [1\ 4\ 5], [1\ 4\ 6], [6\ 1\ 5], [1\ 4\ 3], [6\ 5\ 4],$
 $[6\ 3\ 5], [1\ 5\ 4], [1\ 4\ 5\ 4], [6\ 4\ 1\ 5], [4\ 3\ 2\ 1], [4\ 1\ 5\ 1], [1\ 5\ 4\ 5],$
 $[3\ 1\ 5\ 6], [3\ 4\ 2\ 3], [4\ 5\ 1], [4\ 2\ 4\ 5], [4\ 5\ 6\ 5], [4\ 5\ 1\ 6], [4\ 1\ 5],$
 $[5\ 1\ 6], [1\ 5\ 1\ 5], [1\ 2\ 1\ 2], [1\ 4\ 1], [5\ 6\ 5\ 6], [5\ 4\ 5], [2\ 5\ 2\ 5],$
 $[5\ 1\ 4\ 1\ 5\ 4], [1\ 5\ 6\ 3\ 4\ 1\ 5], [1\ 5\ 4\ 6\ 2\ 4\ 1], [5\ 4\ 3\ 2\ 1], [1\ 5\ 1\ 4\ 1\ 5]\}$
 $P = \{S \rightarrow AS \mid BS \mid CS \mid DS \mid A \mid B \mid C \mid D$
 $A \rightarrow [6\ 4\ 5\ 1] \mid [6\ 1\ 4\ 5] \mid [1\ 2\ 4\ 5] \mid [1\ 4\ 5] \mid [1\ 4\ 6] \mid [6\ 1\ 5] \mid$
 $[1\ 4\ 3] \mid [6\ 5\ 4] \mid [6\ 3\ 5] \mid [1\ 5\ 4] \mid [1\ 4\ 5\ 4] \mid [6\ 4\ 1\ 5] \mid [1\ 5\ 4\ 5]$
 $B \rightarrow [4\ 3\ 2\ 1] \mid [4\ 1\ 5\ 1] \mid [3\ 1\ 5\ 6] \mid [3\ 4\ 2\ 3] \mid [4\ 5\ 1] \mid$
 $[4\ 2\ 4\ 5] \mid [4\ 5\ 6\ 5] \mid [4\ 5\ 1\ 6] \mid [4\ 1\ 5] \mid [5\ 1\ 6]$
 $C \rightarrow [1\ 5\ 1\ 5] \mid [1\ 2\ 1\ 2] \mid [1\ 4\ 1] \mid [5\ 6\ 5\ 6] \mid [5\ 4\ 5] \mid [2\ 5\ 2\ 5]$
 $D \rightarrow [5\ 1\ 4\ 1\ 5\ 4] \mid [1\ 5\ 6\ 3\ 4\ 1\ 5] \mid [1\ 5\ 4\ 6\ 2\ 4\ 1] \mid [5\ 4\ 3\ 2\ 1] \mid$
 $[1\ 5\ 1\ 4\ 1\ 5]\}$

Fonte: Próprio autor

Na variável B da gramáticas para Reggae se encontram as progressões que não iniciam com o grau tônico ou relativo menor, mas que possuem de 2 a 4 acordes distintos na sua estrutura.

Na variável C da gramática para Punk Rock se encontram as progressões com 2 acordes distintos na composição.

Na variável C da gramática para Sertanejo Universitário se encontram as progressões com 5 ou mais acordes distintos, ou quem possuem 5 ou mais elementos na sua construção.

Na variável C da gramática para Reggae se encontram as progressões com 5 ou mais elementos em sua composição.

Na variável D da gramática para Punk Rock se encontram as progressões com 5 ou mais acordes distintos, ou quem possuem 5 ou mais elementos na sua construção.

Na variável D da gramática para Sertanejo Universitário se encontram as progressões com 2 acordes distintos na composição.

Na variável D da gramática para Reggae se encontram as progressões com 5 ou mais acordes distintos.

Figura 24 – Gramática livre de contexto resumida para o estilo Sertanejo Universitário

$G_{\text{SERTANEJO}} = (\{S, A, B, C, D\}, T, S, P)$ tal que,
 $T = \{ [6\ 1\ 5\ 6], [1\ 5\ 6\ 2\ 5], [1\ 5\ 6\ 4], [1\ 4\ 1\ 6\ 5], [1\ 4\ 1\ 5], [6\ 4\ 1\ 5], [4\ 5\ 3\ 6],$
 $[6\ 4\ 5\ 3], [6\ 4\ 5\ 6], [4\ 5\ 1\ 4], [1\ 5\ 4\ 1], [1\ 2\ 4\ 1], [6\ 5\ 4], [2\ 4\ 6\ 5],$
 $[4\ 5\ 6\ 1], [4\ 1\ 5], [6\ 2\ 5\ 1], [1\ 4\ 2\ 5], [1\ 5\ 2\ 4], [6\ 5\ 2\ 4], [4\ 1\ 2\ 1],$
 $[1\ 2\ 4], [1\ 5\ 4\ 2], [2\ 4\ 1\ 5], [1\ 5\ 2\ 4\ 5], [1\ 6\ 4\ 5], [1\ 5\ 4], [4\ 5\ 1\ 6],$
 $[4\ 5\ 6], [4\ 1\ 4\ 1\ 5], [1\ 4\ 6\ 1], [2\ 1\ 2], [6\ 1\ 5\ 2], [2\ 5\ 1\ 2], [2\ 5\ 2\ 4],$
 $[5\ 4\ 1\ 5], [4\ 5\ 4\ 5], [4\ 1\ 4\ 1], [5\ 6\ 3\ 4], [6\ 5\ 1\ 2\ 1\ 4], [4\ 1\ 6\ 5], [1\ 5\ 4\ 5],$
 $[4\ 5\ 6\ 4], [4\ 5\ 1], [1\ 6\ 4\ 2\ 5], [1\ 5\ 6\ 5], [2\ 4\ 5\ 6], [4\ 1\ 5\ 6], [6\ 4\ 2\ 3], [2\ 4\ 1],$
 $[1\ 5\ 6\ 4\ 5], [1\ 5\ 2\ 4\ 1], [4\ 5\ 1\ 5\ 6], [5\ 2\ 1], [5\ 4\ 5\ 1], [2\ 6\ 4\ 5], [4\ 5\ 6\ 1\ 4],$
 $[1\ 5\ 2\ 1\ 4] \}$
 $P = \{ S \rightarrow AS \mid BS \mid CS \mid DS \mid A \mid B \mid C \mid D$
 $A \rightarrow [1\ 5\ 6\ 4] \mid [6\ 4\ 1\ 5] \mid [1\ 5\ 4\ 1] \mid [1\ 5\ 2\ 4] \mid [6\ 1\ 5\ 6] \mid [1\ 6\ 4\ 5] \mid$
 $[1\ 4\ 6\ 1] \mid [1\ 4\ 1\ 5] \mid [6\ 4\ 5\ 3] \mid [6\ 4\ 5\ 6] \mid [6\ 1\ 5\ 2] \mid [1\ 2\ 4\ 1] \mid$
 $[6\ 2\ 5\ 1] \mid [1\ 4\ 2\ 5] \mid [6\ 5\ 2\ 4] \mid [6\ 5\ 4] \mid [1\ 5\ 4\ 2] \mid [1\ 5\ 4] \mid$
 $[1\ 5\ 4\ 5] \mid [1\ 5\ 6\ 5] \mid [6\ 4\ 2\ 3] \mid$
 $B \rightarrow [2\ 4\ 1\ 5] \mid [2\ 4\ 5\ 6] \mid [4\ 1\ 5\ 6] \mid [5\ 4\ 5\ 1] \mid [4\ 5\ 3\ 6] \mid [4\ 5\ 1\ 4] \mid$
 $[4\ 1\ 2\ 1] \mid [2\ 5\ 1\ 2] \mid [2\ 5\ 2\ 4] \mid [5\ 4\ 1\ 5] \mid [4\ 5\ 1\ 6] \mid [4\ 5\ 6\ 1] \mid$
 $[4\ 1\ 5] \mid [4\ 5\ 1\ 6] \mid [4\ 5\ 6] \mid [5\ 6\ 3\ 4] \mid [4\ 1\ 6\ 5] \mid [4\ 5\ 6\ 4] \mid$
 $[4\ 5\ 1] \mid [2\ 4\ 1] \mid [5\ 2\ 1] \mid [5\ 4\ 5\ 1]$
 $C \rightarrow [1\ 5\ 6\ 2\ 5] \mid [4\ 1\ 4\ 1\ 5] \mid [1\ 4\ 1\ 6\ 5] \mid [4\ 5\ 6\ 1\ 4] \mid$
 $[1\ 5\ 2\ 1\ 4] \mid [1\ 5\ 6\ 4\ 5] \mid [1\ 5\ 2\ 4\ 1] \mid [4\ 5\ 1\ 5\ 6] \mid [1\ 6\ 4\ 2\ 5] \mid$
 $[6\ 5\ 1\ 2\ 1\ 4] \mid [1\ 5\ 2\ 4\ 5]$
 $D \rightarrow [2\ 1\ 2] \mid [4\ 5\ 4\ 5] \mid [4\ 1\ 4\ 1] \}$

Fonte: Próprio autor

Em todas as gramáticas as progressões contidas na variável “C”, tiveram um índice de incidência mais alto do que as da variável “D”, apesar dessas variáveis possuírem características distintas, se houver a comparação entre as variáveis homônimas das três gramáticas.

S contém as produções AS , BS , CS , DS , A , B , C e D . Essa construção foi escolhida para possibilitar a criação de músicas com uma ou mais sequências de acordes. Essa é a variável responsável por guiar a semântica das composições.

No algoritmo de composição, o peso atribuído à variável A , deverá ser maior do que o peso atribuído à B , que deverá ser maior do que o peso de C , que deverá ser maior do

que o peso atribuído a D . Essa regra vale para todas as gramáticas descritas nessa seção.

Figura 25 – Gramática livre de contexto resumida para o estilo Reggae

GREGGAE = ({S, A, B, C, D}, T, S, P) tal que,

$T = \{ [3\ 1\ 5], [1\ 5\$ 6\ 4], [1\ 6\ 4\ \$1\ 2], [1\ 4\$ 1], [1\ 5\ \$4\ 1\ 5\ 1], [5\ 3\ 5], [3\ \$2\ 3\ 2],$
 $[5\ 1\ 5], [1\ 2\ 3\$ 2], [6\ 5\ 6\$ 5], [6\ 3\ 2\ 5], [2\$ 3\ 6], [6\ 3\ \$6\ 3], [5\ \$1\ 5\ 1],$
 $[1\ 5\ 4\ 2], [2\ \$1\ 2\ 1], [3\ 6\$ 3\ 6], [1\ 5\ 2\ 4], [6\ 1\ 4\ 5\ \$6\ 5\ 6], [1\ 4\$ 1\ 4],$
 $[4\ 1\ 4\ 5], [2\ 5\ 6], [5\ 6\ 5\ 6], [5\ 3\ 5\ 3], [1\ 3\ 2\ 4], [4\ 6\ 5\ 6], [6\ 5\ 4],$
 $[4\ 6\ 5], [6\ 4\ 1\ 5], [6\ 2\ 3], [6\ 5\ 2\ 3], [1\ 4\ 5], [6\ \$3\ 4\ 3], [5\ 4\$ 5\ 4],$
 $[1\ 2], [6\ 3\ 2], [6\ 3], [1\ 2\ 4\ 5\ 1], [6\ 4\ 5], [4\ 5\ 1\ 6], [1\ 4\ 5\ 1], [1\ 5\ 1\ 4],$
 $[1\ 6\ 4\ 1], [4\ 3\ 4\ 3], [1\ 2\ 3\ 5], [1\ 5\ 6\ 4\ 5], [6\ 3\ 1\ 5], [6\ 2\ \$6\ 2], [5\ 4\ 3\ 2\ 1],$
 $[2\ 6\ 2\ 6], [2\ 5\ 2\ 5], [4\ 5\ 4\ 6], [1\ 6\ 1\ 6], [4\ 1\ 4\ 3], [2\ 3\ 2\ 3], [3\ 2\ 5\ 3], [4\ 3\ 2\ 5],$
 $[3\ 6\ \$3], [4\ 5\ 6\ 2], [5\ 6\ 2\ 3\ 2\ 6], [1\ 2\ 4\ 5], [6\ 4\ 3], [2\ 5\ 1\ 6\ 4\ 5\ 1\ 6] \}$

$P = \{ S \rightarrow AS \mid BS \mid CS \mid DS \mid A \mid B \mid C \mid D$
 $A \rightarrow [6\ 3\ 6\ 3] \mid [1\ 5\ 6\ 4] \mid [1\ 4\ 1\ 4] \mid [1\ 4\ 1] \mid [6\ 5\ 6\ 5] \mid [6\ 3\ 4\ 3] \mid$
 $[6\ 2\ 6\ 2] \mid [6\ 4\ 5] \mid [1\ 2\ 3\ 2] \mid [6\ 3\ 2\ 5] \mid [1\ 5\ 4\ 2] \mid [1\ 5\ 2\ 4] \mid$
 $[1\ 3\ 2\ 4] \mid [6\ 5\ 4] \mid [6\ 4\ 1\ 5] \mid [6\ 2\ 3] \mid [6\ 5\ 2\ 3] \mid [1\ 4\ 5] \mid [1\ 2] \mid$
 $[6\ 3\ 2] \mid [6\ 3] \mid [6\ 4\ 5] \mid [1\ 4\ 5\ 1] \mid [1\ 5\ 1\ 4] \mid [1\ 6\ 4\ 1] \mid [1\ 2\ 3\ 5] \mid$
 $[6\ 3\ 1\ 5] \mid [1\ 6\ 1\ 6]$
 $B \rightarrow [3\ 2\ 3\ 2] \mid [5\ 1\ 5\ 1] \mid [3\ 6\ 3\ 6] \mid [2\ 3\ 6] \mid [3\ 6\ 3] \mid [5\ 4\ 5\ 4] \mid$
 $[2\ 1\ 2\ 1] \mid [3\ 1\ 5] \mid [5\ 3\ 5] \mid [5\ 1\ 5] \mid [4\ 6\ 5\ 6] \mid [4\ 1\ 4\ 5] \mid$
 $[2\ 5\ 6] \mid [5\ 6\ 5\ 6] \mid [5\ 3\ 5\ 3] \mid [4\ 6\ 5] \mid [4\ 5\ 1\ 6] \mid [4\ 3\ 4\ 3] \mid$
 $[2\ 6\ 2\ 6] \mid [2\ 5\ 2\ 5] \mid [4\ 5\ 4\ 6] \mid [4\ 1\ 4\ 3] \mid [2\ 3\ 2\ 3] \mid [3\ 2\ 5\ 3] \mid$
 $[4\ 3\ 2\ 5] \mid [4\ 5\ 6\ 2]$
 $C \rightarrow [1\ 5\ 4\ 1\ 5\ 1] \mid [1\ 6\ 4\ 1\ 2] \mid [6\ 1\ 4\ 5\ 6\ 5\ 6] \mid [1\ 5\ 6\ 4\ 5] \mid [1\ 2\ 4\ 5\ 1] \mid$
 $[5\ 6\ 2\ 3\ 2\ 6] \mid [2\ 5\ 1\ 6\ 4\ 5\ 1\ 6]$
 $D \rightarrow [5\ 4\ 3\ 2\ 1] \}$

Fonte: Próprio autor

7.2 Algoritmo de composição

Para poder iniciar os comentários a respeito do algoritmo de composição, faz-se necessária a apresentação de como as gramáticas estão organizadas no código.

As variáveis e terminais das gramáticas foram organizadas em *arrays* *arrays* tridimensionais, de modo a auxiliar o algoritmo de composição (Anexo B). Na Figura 26 é

apresentado um trecho do método responsável pela construção da harmonia. O método recebe do usuário a informação do tom, o qual deve pertencer a harmonia e o estilo de música selecionado, ambos apresentados para números inteiros, com o objetivo de facilitar a programação.

Inicialmente, é instanciado um *ArrayList* que será responsável por armazenar a progressão gerada pelo algoritmo e que posteriormente será convertida em acordes. Para determinar qual será a construção da harmonia, foi utilizado o método *random*, da classe *Math* de Java, para escolher um número de maneira pseudoaleatória (escolha baseada em JR e ISAACSON (1957), cujas notas musicais eram obtidas dessa mesma maneira). Assim, foi usado um *switch* para que a gramática utilizada esteja de acordo com a escolha do usuário, na imagem é mostrado o trecho do código em que o *switch* entra no caso da escolha do estilo Punk Rock.

Na parte interna do *case 1* (linha 16), é mostrado de que maneira os números gerados pseudoaleatoriamente são utilizados para determinar qual produção será escolhida. Nos *if* (linhas 18, 27, 36 e 45) fica clara a utilização do peso para cada produção da gramática, nos quais é mostrado que *A* será escolhido quando o número gerado estiver no intervalo fechado de 50 a 100 (50%), *B* é escolhido nos casos em que o número estiver entre 20 e 49 (30%), *C* será escolhido quando o número estiver entre 7 e 19 (13%) e para os números menos que 7 (7%) a escolha será a *D*. Os pesos foram atribuídos de acordo com quantidade de aparições em cada amostras de componentes com características semelhantes.

Quando a progressão harmônica é escolhida, ela é armazenada no *ArrayList*, para que possa ser transformada em uma *string* contendo os acordes que serão exibidos para o usuário.

Na próxima seção serão apresentados os resultados dos testes realizados para verificar o funcionamento do algoritmo de composição e da ferramenta.

Figura 26 – Trecho de código contendo o método Java responsável por gerar harmonias musicais

```
7 public static String compositor(int tom, int estilo){
8
9     ArrayList<Integer> progressao = new ArrayList<Integer>();
10
11     int valor = (int) (Math.random() * 101);
12     int i = 0;
13
14     switch(estilo){
15
16     case 1:
17
18         if (valor >= 50){
19
20             int posicao = (int) (Math.random() * Gramaticas.punk()[0].length);
21
22             while(i<Gramaticas.punk()[0][posicao].length){
23                 progressao.add(i, Gramaticas.punk()[0][posicao][i]);
24                 i++;
25             }
26
27         }else if(valor >= 20 ){
28
29             int posicao = (int) (Math.random() * Gramaticas.punk()[1].length);
30
31             while(i<Gramaticas.punk()[1][posicao].length){
32                 progressao.add(i, Gramaticas.punk()[1][posicao][i]);
33                 i++;
34             }
35
36         }else if(valor >= 7 ){
37
38             int posicao = (int) (Math.random() * Gramaticas.punk()[2].length);
39
40             while(i<Gramaticas.punk()[2][posicao].length){
41                 progressao.add(i, Gramaticas.punk()[2][posicao][i]);
42                 i++;
43             }
44
45         }else{
46
47             int posicao = (int) (Math.random() * Gramaticas.punk()[3].length);
48
49             while(i<Gramaticas.punk()[3][posicao].length){
50                 progressao.add(i, Gramaticas.punk()[3][posicao][i]);
51                 i++;
52             }
53
54         }
55     }
56     break;
```

Fonte: Próprio autor

8 Testes

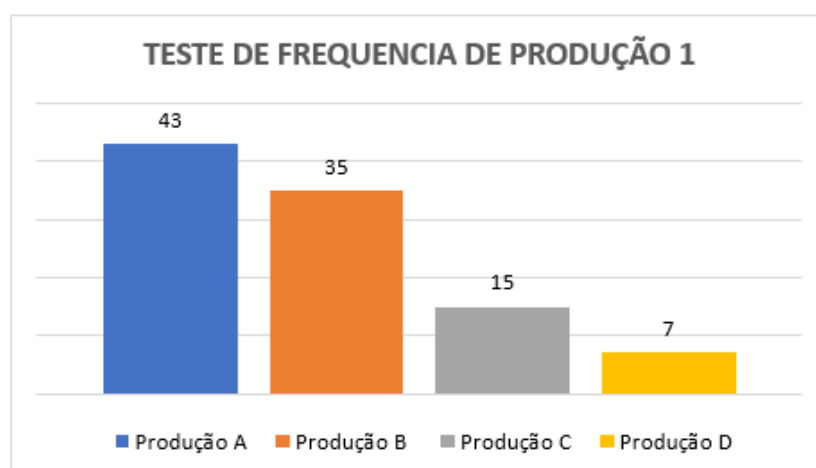
Foram realizados dois tipos de testes que serão detalhados nesta seção. Primeiramente, foi testado o algoritmo de composição para determinar se a geração de harmonias está funcionando de acordo com o peso determinado para cada tipo de produção, e um teste de usabilidade, para determinar se a experiência de uso da ferramenta é satisfatória.

8.1 Testes do algoritmo de composição

Para verificar a frequência com que cada tipo de harmonia está sendo gerada, foram utilizados como base, os dados da Seção 7.2. e gerou-se 100 harmonias distintas geradas sem interrupção do algoritmo. O programa foi reiniciado após o fim da execução para refazer o teste. Nesta seção, as produções serão nomeadas de acordo com as gramáticas apresentadas na Seção 7.1, em que *A*, *B*, *C* e *D*, são variáveis de cada gramática que contém as progressões harmônicas. O teste foi refeito 4 vezes e obteve-se os resultados apresentados a seguir.

- No primeiro teste (Figura 27), após 100 execuções, as produções do tipo A foram escolhidas pelo algoritmo 43 vezes, as do tipo B foram escolhidas 35 vezes, as do tipo C foram escolhidas 15 vezes e as do tipo D foram escolhidas 7 vezes.

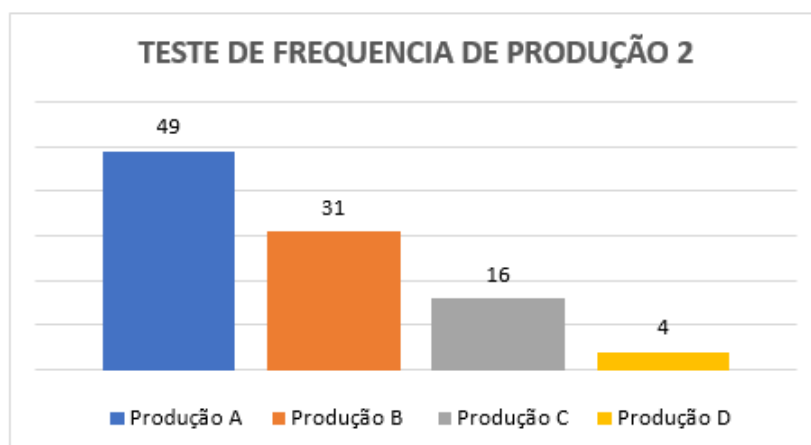
Figura 27 – Quantidade de vezes que cada produção foi escolhida – Primeiro teste



Fonte: Próprio autor

- No segundo teste (Figura 28), após 100 execuções, as produções do tipo A foram escolhidas pelo algoritmo 49 vezes, as do tipo B foram escolhidas 31 vezes, as do tipo C foram escolhidas 16 vezes e as do tipo D foram escolhidas 4 vezes.

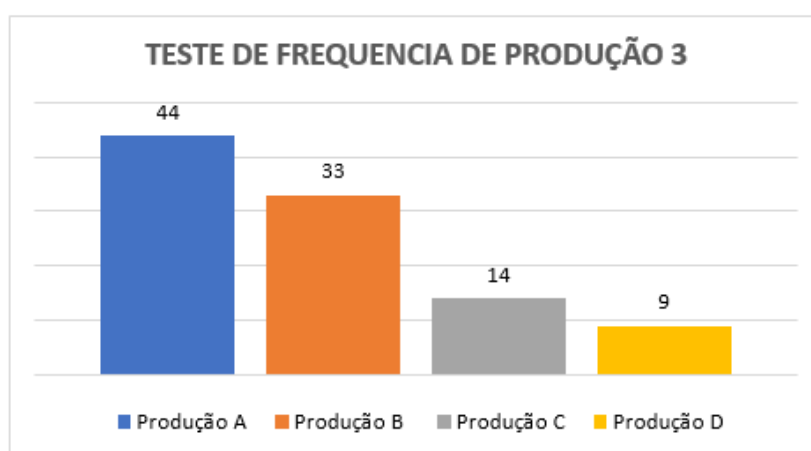
Figura 28 – Quantidade de vezes que cada produção foi escolhida – Segundo teste



Fonte: Próprio autor

- No terceiro teste (Figura 29), após 100 execuções, as produções do tipo A foram escolhidas pelo algoritmo 44 vezes, as do tipo B foram escolhidas 33 vezes, as do tipo C foram escolhidas 14 vezes e as do tipo D foram escolhidas 9 vezes.

Figura 29 – Quantidade de vezes que cada produção foi escolhida – Terceiro teste

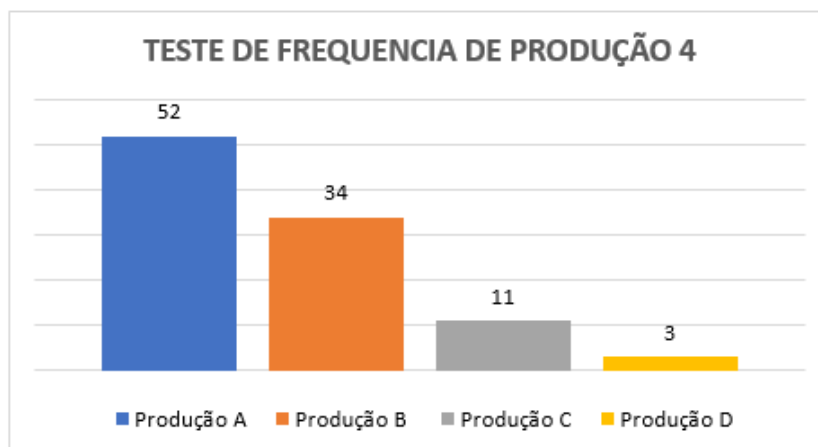


Fonte: Próprio autor

- No quarto teste (Figura 30), após 100 execuções, as produções do tipo A foram escolhidas pelo algoritmo 52 vezes, as do tipo B foram escolhidas 34 vezes, as do tipo C foram escolhidas 11 vezes e as do tipo D foram escolhidas 3 vezes.

Os testes mostraram que a frequência com que cada produção é gerada está de acordo com a ordem de prioridade que foi proposta quando a gramática geradora foi criada, e estão coerentes com as análises feitas na amostra de músicas de cada estilo na Seção 6.1.

Figura 30 – Quantidade de vezes que cada produção foi escolhida – Quarto teste



Fonte: Próprio autor

8.2 Testes de classificação

8.2.1 Primeiro teste

Neste teste, foi utilizado um classificador Naive Bayes, que é um classificador probabilístico baseado no "Teorema de Bayes", e como base para treinamento utilizaram-se as progressões harmônicas extraídas das 40 músicas da primeira amostra na seção 6.1. Após a implementação do algoritmo de composição, foram geradas 20 harmonias, cada uma contendo 3 progressões harmônicas, para os estilos Punk Rock e Sertanejo Universitário, sendo 10 harmonias para cada estilo.

O teste foi executado 10 vezes, e em cada bateria de testes foram geradas 20 novas harmonias. A taxa de acerto média dos 10 testes foi de 92,5% com desvio padrão de $\approx 0,0425$.

Embora a taxa de acerto esteja apropriada, é preciso testar o impacto que a adição de novas amostras terá na capacidade do algoritmo de classificação distinguir os estilos musicais.

8.2.2 Segundo teste

Após a ampliação da amostra e inclusão de um novo estilo musical, descrita na Seção 6.2, os testes de classificação foram refeitos.

Foi utilizado um classificador Naive Bayes e, como base para treinamento, escolheu-se as progressões harmônicas extraídas de todas as músicas da amostra (Anexo A). Após a implementação do algoritmo de composição, foram geradas 30 harmonias, cada uma contendo 3 progressões harmônicas, para os estilos Punk Rock, Sertanejo Universitário e Reggae, sendo 10 harmonias para cada estilo.

O teste foi executado 10 vezes, e em cada bateria de testes foram geradas 30 novas harmonias. A taxa de acerto média dos 10 testes foi de 76,66% com desvio padrão de $\approx 0,0141$.

A redução drástica da taxa de acerto do primeiro teste (Seção 8.2.1) para o teste atual, pode ser explicada pelo fato do aumento das progressões harmônicas que são comuns entre os estilos (aumento de 1 para 18).

8.3 Teste de usabilidade

Para avaliar a experiência de usuário, foram selecionadas 30 pessoas em grupos do Facebook relacionados à música que preencheram os seguintes requisitos:

- Possuir idade igual ou maior do que 18 anos;
- Ser usuário de computador;
- Tocar algum instrumento musical, com exceção de instrumentos de percussão;
- Ter estudado música por no máximo 1 ano;
- Possuir conhecimento de notação de cifras;
- Possuir conhecimento mínimo sobre o campo harmônico;

Após a seleção, a ferramenta (com o botão “Tutorial” desabilitado) foi disponibilizada junto com um questionário, para que o usuário respondesse a algumas questões acerca da sua experiência durante o uso da ferramenta. Os resultados obtidos são apresentados a seguir:

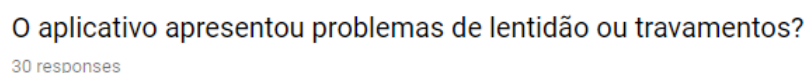
- Dos 30 usuários, 2 não aprovaram a aparência do aplicativo (Figura 31).
- Dos 30 usuários, 3 tiveram problemas de travamento ou lentidão (Figura 32).
- 5 usuários tiveram alguma dificuldade para utilizar o aplicativo (Figura 33).
- Das 5 pessoas que tiveram alguma dificuldade, 2 tiveram dificuldades de aprendizado, 1 comentou o fato de não haver possibilidade de mudar de funcionalidade sem precisar voltar para a tela inicial, 1 teve dificuldade pois não tinha conhecimento do que é progressão harmônica e 1 teve dificuldades pois achou que as letras eram muito pequenas (Figura 34).
- Dos 30 usuários, 9 não consideram o aplicativo útil para a vida cotidiana e para o aprendizado musical (Figura 35).

Figura 31 – Porcentagem de usuários que aprovaram a aparência do aplicativo



Fonte: Próprio autor

Figura 32 – Porcentagem de usuários que tiveram problemas técnicos no uso do aplicativo



Fonte: Próprio autor

- Dos 30 usuários, 2 (6,7%) tiveram uma experiência ruim com o aplicativo, 6 (20%) tiveram uma experiência neutra, 19 (63,7%) tiveram uma experiência boa e 3 (10%) tiveram uma experiência ótima (Figura 36).
- 12 usuários opinaram para ajudar na melhoria do aplicativo (Figura 37). 6 usuários sugeriram a inclusão de mais estilos musicais, 2 sugeriram modificações nos botões, 1 sugeriu a inclusão de um manual no aplicativo, 1 sugeriu o aumento no tamanho das fontes contidas no aplicativo, 1 sugeriu que as harmonias geradas fossem mais complexas e 1 sugeriu a criação de um aplicativo para Android.

Após um cruzamento de informações, constatou-se que os 2 usuários que não gostaram da aparência do aplicativo reclamaram dos botões. Não foi possível identificar o motivo dos problemas de travamento e lentidão dos 3 usuários que relataram esse tipo de

Figura 33 – Porcentagem de usuários que tiveram dificuldade no uso do aplicativo



Fonte: Próprio autor

Figura 34 – Comentários dos usuários a respeito das dificuldades na utilização do aplicativo

Se a resposta acima foi "sim", quais foram as dificuldades?

5 responses

O aplicativo não permitiu que eu de uma função para outra sem voltar para a primeira tela
Demorei pra conseguir aprender a usar
Não sei o que é preogressão harmonica
letras muito pequenas
algumas funções são difíceis de usar

Fonte: Próprio autor

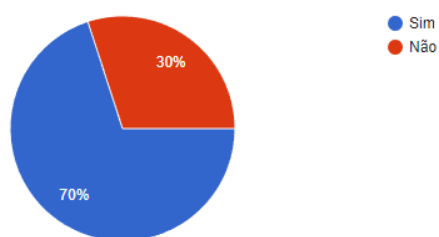
problema. O usuário que sugeriu a criação de um aplicativo para Android foi um dos que teve uma ótima experiência com o aplicativo.

Após análise, é possível concluir que o aplicativo se mostrou útil para a maioria dos usuários e que as técnicas utilizadas para gerar harmonias funcionaram para a maioria dos usuários. A aparência e experiência de uso do aplicativo mostraram-se agradável para a maioria das pessoas, porém existem pontos a considerar-se para a melhoria desses fatores.

Figura 35 – Porcentagem de usuários que julgaram o aplicativo útil para o cotidiano e para o aprendizado

O aplicativo seria útil na sua vida cotidiana e no seu aprendizado como músico?

30 responses

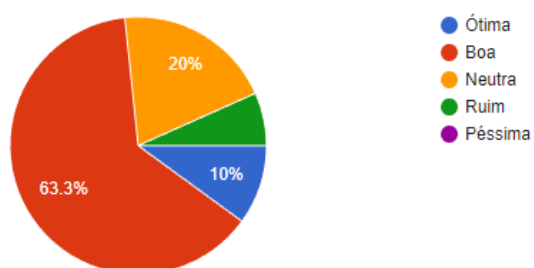


Fonte: Próprio autor

Figura 36 – Qualidade de experiência de uso, segundo usuários do aplicativo

Como você classifica sua experiência com o uso do aplicativo?

30 responses



Fonte: Próprio autor

Figura 37 – Comentários dos usuários a respeito das dificuldades na utilização do aplicativo

O que você sugere para que haja uma melhoria do aplicativo?

12 responses

Adicionar mais estilos
Adicionar o estilo MPB
Colocar mais estilos de música
Colocar Forró
colocar manual pra facilitar
seria bom acrescentar o mais estilos brasileiros
aumentar o tamanho das letras
aumentar a complexidade das harmonias
Melhorar aparencia dos botões
os botões não parecem botões
colocar mais estilos de músicas no programa
fazer aplicativo pra Android

Fonte: Próprio autor

9 Resultados e Conclusões

Concluídos os testes e de posse dos resultados das pesquisas de usabilidade, os resultados obtidos no estudo foram diversos.

As gramáticas foram capazes de gerar combinações de acordes de acordo com as regras do campo harmônico, o que comprova que é possível utilizar gramáticas livres de contexto para composição algorítmica - como outros estudos anteriores também já haviam comprovado. O destaque desse estudo vem pela adoção da combinação das duas técnicas utilizadas e é diferenciado pela maneira como a análise das músicas foi feita e o foco na usabilidade, buscando o feedback de usuários para validar a aceitação da ferramenta.

Os estilos adotados para o desenvolvimento da pesquisa apresentaram distinção entre suas construções harmônicas apenas na primeira amostra. Na segunda, apresentaram-se convergências entre os estilos, que fez com que os testes de classificação tivessem uma piora nos resultados, fato que mostra que com o crescimento da amostra, pode ficar mais difícil distinguir estilos musicais de maneira objetiva apenas pela harmonia, sendo necessário a inclusão de novos elementos como melodia ou ritmo.

De acordo com a pesquisa de usabilidade, a maioria dos usuários considerou a ferramenta útil. Pode-se concluir que o aplicativo se mostrou didático e capaz de auxiliar no processo de composição. Em relação à sua usabilidade, pode se dizer intuitiva, porém com diversos pontos a serem melhorados.

Diante da finalização do estudo, os objetivos traçados foram parcialmente alcançados e o produto final se mostrou satisfatório para o público a que ele se destina, porém diversas melhorias e evoluções podem ser feitas para obtenção de melhores resultados.

Apesar de algumas limitações do uso de uma gramática livre de contexto (apresentadas na Seção 3.2) terem sido amenizadas por meio do uso do conhecimento extraído de composições conhecidas como base para a sua construção, a implementação atual ainda apresenta algumas limitações:

- A implementação de métodos para gerar acordes tétrades, que consiste em acordes que possuem o sétimo grau em sua formação, ainda precisa ser pensada.
- A implementação atual não apresenta um método para definir compasso e tempo, que faz com que a adição de um sintetizador MIDI (Interface Digital de Instrumentos Musicais) se torne inviável, pois o usuário teria que selecionar os parâmetros rítmicos e de divisão musical, ou adotar uma política como a descrita em (MCCORMACK, 1996), em que os acordes são colocados cada um em $1/4$ de um compasso $4/4$.

- Para gerar composições complexas, mais tecnologias precisam ser empregadas.
- Os resultados dos testes de classificação, mostram que é possível que a adição de mais amostras faça com que a distinção entre estilos fique cada vez mais difícil de ser feita apenas de acordo com os padrões harmônicos, se fazendo necessária a implementação de outros elementos para exaltar a identidade de cada estilo, como ritmo e melodia.

A composição algorítmica ainda é um campo que pode ser bastante explorado, e com a evolução da inteligência artificial, é provável que surjam novas técnicas para gerar música por meio de um computador, como vem ocorrendo desde 1956, quando ocorreu o marco inicial desses estudos.

O campo harmônico se mostrou adequado e simples de ser reproduzido por meio de outros formalismos e diversas possibilidades podem ser exploradas a fim de gerar novas pesquisas em outras áreas.

Além das conclusões obtidas por meio desse trabalho, também podem ser levantadas diversas hipóteses, que podem ser úteis para trabalhos futuros, como as seguintes:

- Verificar a possibilidade da utilização das mesmas técnicas utilizadas nesse trabalho, para estilos musicais com harmonias mais complexas;
- Verificar se a inclusão de novos estilos musicais resultaria na necessidade da criação de novas gramáticas com estruturas e regras de produção diferentes;
- Verificar a possibilidade de gerar melodias e ritmos utilizando as mesmas técnicas apresentadas nesse estudo;
- Testar o impacto que uma versão para dispositivos móveis dessa ferramenta teria na experiência do usuário;
- Testar se a inclusão de plug-ins para reprodução de áudio, em conjunto com a inclusão de ritmos e elementos de partitura, resultaria numa melhor experiência no que diz respeito a percepção do usuário a respeito da identidade musical das composições.
- Testar a eficiência da ferramenta na educação musical.

Referências

- ABREU, P. A indústria fonográfica e o mercado da música gravada—histórias de um longo desentendimento. *Revista Crítica de Ciências Sociais*, Centro de Estudos Sociais da Universidade de Coimbra, n. 85, p. 105–129, 2009. Citado na página 12.
- AMES, C. Automated composition in retrospect: 1956-1986. *Leonardo*, JSTOR, p. 169–185, 1987. Citado na página 17.
- BASSETO, B. A. Um sistema de composição musical automatizada, baseado em gramáticas sensíveis ao contexto, implementado com formalismos adaptativos. *USP, São Paulo, Brasil*, 2000. Citado 2 vezes nas páginas 17 e 20.
- BUXTON, W. A. A composer's introduction to computer music. *Journal of New Music Research*, Taylor & Francis, v. 6, n. 2, p. 57–71, 1977. Citado na página 18.
- CAMARATTA, E. D.; FISCHER, L. Composição de música com o uso de algoritmos. 08 2014. Disponível em: <https://www.researchgate.net/publication/228544963_Composicao_de_Musica_com_o_Uso_de_Algoritmos>. Acesso em: 14 jun. 2017. Citado 3 vezes nas páginas 12, 17 e 19.
- CHEDIAK, A. *Harmonia & Improvisação-Vol. 1*. [S.l.]: Irmãos Vitale, 1986. Citado na página 15.
- HILLER, L. A.; BAKER, R. A. Computer cantata: A study in compositional method. *Perspectives of New Music*, JSTOR, p. 62–90, 1964. Citado na página 17.
- JR, L. A. H.; ISAACSON, L. M. Musical composition with a high speed digital computer. In: AUDIO ENGINEERING SOCIETY. *Audio Engineering Society Convention 9*. [S.l.], 1957. Citado 2 vezes nas páginas 17 e 46.
- LASKE, O. E. *Introduction to a generative theory of music*. [S.l.]: Institute of Sonology, Utrecht State University, 1973. Citado na página 18.
- LASKE, O. E. *Music, memory, and thought: Explorations in cognitive musicology*. [S.l.]: University Microfilms, 1977. Citado na página 18.
- LASKE, O. E. Considering human memory in designing user interfaces for computer music. *Computer Music Journal*, JSTOR, p. 39–45, 1978. Citado na página 18.
- LERDAHL, F.; JACKENDOFF, R. Toward a formal theory of tonal music. *Journal of music theory*, JSTOR, v. 21, n. 1, p. 111–171, 1977. Citado na página 18.
- LINDBLOM, B.; SUNDBERG, J. *Towards a generative theory of melody*. [S.l.]: Department of Phonetics, Institute of Linguistics, University of Stockholm, 1970. Citado na página 18.
- MCCORMACK, J. Grammar based music composition. *Complex systems*, v. 96, p. 321–336, 1996. Citado 2 vezes nas páginas 19 e 56.
- MENEZES, P. B. *Linguagens formais e autômatos*. [S.l.]: Sagra-Dcluzzato, 1998. Citado na página 14.

- MINSKY, M. L. Why people think computers can't. *AI Magazine*, v. 3, n. 4, p. 3, 1982. Citado na página 19.
- MOORER, J. A. Music and computer composition. *Communications of the ACM*, ACM, v. 15, n. 2, p. 104–113, 1972. Citado na página 18.
- NATTIEZ, J.-J.; DUNSBY, J. M. Fondements d'une sémiologie de la musique. *Perspectives of New Music*, JSTOR, p. 226–233, 1977. Citado na página 18.
- PAPADOPOULOS, G.; WIGGINS, G. Ai methods for algorithmic composition: A survey, a critical view and future prospects. In: EDINBURGH, UK. *AISB Symposium on Musical Creativity*. [S.l.], 1999. p. 110–117. Citado na página 19.
- RADER, G. M. A method for composing simple traditional music by computer. *Communications of the ACM*, ACM, v. 17, n. 11, p. 631–638, 1974. Citado na página 17.
- ROADS, C. Composing grammars, revised. *Presented to the 1977 International Computer Music Conference, UCSD, La Jolla*, 1978. Citado na página 18.
- ROADS, C.; WIENEKE, P. Grammars as representations for music. *Computer Music Journal*, JSTOR, p. 48–55, 1979. Citado 2 vezes nas páginas 17 e 18.
- RUWET, N. Musique, langage, poésie. *Le Seuil, Paris*, 1972. Citado na página 18.
- RUWET, N. Théorie et méthodes dans les études musicales, «musique en jeu», n. 17. *Seuil, Paris*, p. 11–36, 1975. Citado na página 18.
- SANTINI, R. M.; LIMA, C. Difusão de música na era da internet. *V Encontro Latino de Economia Política da Informação, Comunicação e Cultura*, 2001. Citado na página 12.
- SMITH, L. Score-a musician's approach to computer music. *Journal of the Audio Engineering Society*, Audio Engineering Society, v. 20, n. 1, p. 7–14, 1972. Citado na página 18.
- SMOLIAR, S. W. Music programs: An approach to music theory through computational linguistics. *Journal of Music Theory*, JSTOR, v. 20, n. 1, p. 105–131, 1976. Citado na página 18.
- ULRICH, J. W. The analysis and synthesis of jazz by computer. In: *Proceedings of the 5th International Joint Conference on Artificial Intelligence*,. [S.l.]: Department of Computer Science, Carnegie-Mellon University, Pittsburgh, 1977. Vol. 2, p. 865–872. Citado na página 18.
- WINOGRAD, T. Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, JSTOR, v. 12, n. 1, p. 2–49, 1968. Citado na página 18.
- WINOGRAD, T. A procedural model of language understanding, in computer models of thought and language. *Shank, Colby (Eds.) W.H.Freeman, San Francisco*, 1973. Citado na página 18.

ANEXO A – Músicas analisadas, respectivos estilos musicais e progressões harmônicas detectadas manualmente

MÚSICA	ESTILO	PROGRESSÃO 1	PROGRESSÃO 2	PROGRESSÃO 3
BAD RELIGION - AMERICAN JESUS	PUNK	[6 1 4 5]	—	—
RAMONES - BLITZKRIEG BOP	PUNK	[1 4 5]	[1 5 1 5]	[4 2 4 5]
RAMONES - COMMANDO	PUNK	[6 1 4 5]	[5 6 5 6]	[1 2 1 2]
THE OFFSPRING - THE KIDS ARE'T ALRIGHT	PUNK	[6 4 5]	[6 4 1 5]	[4 5 6 5]
THE SEX PISTOLS - HOLIDAYS IN SUN	PUNK	[4 3 2 1]	[1 4 5 4]	[2 5 2 5]
CAPITAL INICIAL - VERANEIO VASCAINA	PUNK	[3 4 2 3]	[3 4 2 3]	[1 2 4 5]
CAPITAL INICIAL - FÁTIMA	PUNK	[6 4 5]	—	—
GREEN DAY - AMERICAN IDIOT	PUNK	[4 1 5 1]	[5 1 4 1 5 4]	[5 1 4 1 5]
GREEN DAY - BASKET CASE	PUNK	[1 5 6 3 4 1 5]	[4 5 1]	[4 5 1 6]
BLINK 182 - ALL THE SMALL THINGS	PUNK	[1 5 4 5 1]	—	—
BLINK 182 - FIRST DATE	PUNK	[1 2 4 5]	[1 5 4]	[1 5 4 5 1]
RATOS DE PORAO - AIDS POP REPRESSAO	PUNK	[1 4 6]	—	—
LEGIÃO URBANA - QUE PAIS É ESSE	PUNK	[6 4 5]	—	—
RAIMUNDOS - MULHER DE FASES	PUNK	[1 5 4 6 2 4 1]	[6 3 5]	[4 1 5]
ELES - VÊI FULERO	PUNK	[3 1 5 6]	—	—
LEGIÃO URBANA - GERAÇÃO COCA COLA	PUNK	[6 1 5]	[6 5 4]	[5 1 6]
THE CLASH - SHOUL I STAY OR SHOULD I GO	PUNK	[1 4 1]	[5 4 5]	—
SEX PISTOLS - ANARCHY IN THE U.K.	PUNK	[1 4 3]	[5 4 3 2 1]	—
BAD RELIGION - INFECTED	PUNK	[3 1 5 6]	—	—
GAROTOS PODRES - PAPAI NOEL, VELHO BATUTA	PUNK	[6 4 5]	—	—

MÚSICA	ESTILO	PROGRESSÃO 1	PROGRESSÃO 2	PROGRESSÃO 3	PROGRESSÃO 4	PROGRESSÃO 5
VICTOR E LEO - BORBOLETAS	SERTANEJO	[6 1 5 6]	[2 4 1 5]	[2 4 5 6]	[4 1 5 6]	—
VICTOR E LEO - AMIGO APAIXONADO	SERTANEJO	[1 5 6 2 5]	[4 1 4 1 5]	[1 5 6 4]	[5 4 5 1]	—
LUAN SANTANA - METEORO	SERTANEJO	[1 5 6 4]	[1 6 4 5]	—	—	—
LUAN SANTANA - TUDO QUE VOCE QUISER	SERTANEJO	[1 4 1 6 5]	[1 4 6 1]	—	—	—
MICHEL TELO - HUMILDE RESIDENCIA	SERTANEJO	[1 4 1 5]	—	—	—	—
MICHEL TELO - AI SE EU TE PEGO	SERTANEJO	[1 5 6 4]	—	—	—	—
JORGE E MATEUS - EU SOSSEGUEI	SERTANEJO	[1 5 6 4]	—	—	—	—
MARCOS E BELUTTI - AQUELE 1%	SERTANEJO	[6 4 1 5]	—	—	—	—
MAIARA E MARAISA - 10%	SERTANEJO	[4 5 3 6]	—	—	—	—
LUCAS LUCCO - VAI VENDO	SERTANEJO	[6 4 5 3]	[2 1 2]	—	—	—
MARILIA MENDONÇA - INFIEL	SERTANEJO	[6 4 5 6]	[6 1 5 2]	—	—	—
MAIARA E MARAISA - SORTE QUE BEIJA BEM	SERTANEJO	[4 5 1 4]	[2 5 1 2]	[1 5 2 4]	—	—
FERNANDO E SOROCABA - BALA DE PRATA	SERTANEJO	[1 5 6 4]	[1 5 4 1]	—	—	—
FERNANDO E SOROCABA - AS MINA PIRA	SERTANEJO	[1 5 6 4]	—	—	—	—
GUSTTAVO LIMA - BALADA BOA	SERTANEJO	[6 4 1 5]	—	—	—	—
SIMONE E SIMARIA - LOKA	SERTANEJO	[6 4 1 5]	[2 5 2 4]	—	—	—
SIMONE E SIMARIA - QUANDO O MEL É BOM	SERTANEJO	[6 4 1 5]	[2 4 1 5]	—	—	—
CESAR MENOTTI E FABIANO - LEILÃO	SERTANEJO	[1 5 4 1]	[5 4 1 5]	—	—	—
MARILIA MENDONÇA - AMANTE NÃO TEM LAR	SERTANEJO	[6 4 1 5]	[2 4 1 5]	—	—	—
MARCOS E BELUTTI - DOMINGO DE MANHA	SERTANEJO	[1 2 4 1]	[1 5 2 4]	—	—	—
BRUNO E BARRETO - FARRA, PING E FOGUETE	SERTANEJO	[6 4 1 5]	[6 4 5 6]	[4 1 5 6]	—	—
LUAN SANTANA - ACORDANDO O PREDIO	SERTANEJO	[6 5 4]	[6 1 5 2]	[6 4 5 6]	—	—
LUAN SANTANA - CHUVA DE ARROZ	SERTANEJO	[2 4 6 5]	[4 5 4 5]	—	—	—
MAIARA E MARAISA - MEDO BOBO	SERTANEJO	[4 5 6 1]	—	—	—	—
LUAN SANTANA - ADRENALINA	SERTANEJO	[1 5 6 4]	[2 4 1 5]	—	—	—
SIMONE E SIMARIA - REGIME FECHADO	SERTANEJO	[4 5 6 1]	[6 4 5 3]	[6 4 2 3]	[4 5 1 5 6]	—
MICHEL TELO - COISA DE DEUS	SERTANEJO	[4 1 5]	[4 1 4 1]	[2 4 1]	[5 2 1]	—
WESLEY SAFADAO - NINGUEM E DE FERRO	SERTANEJO	[1 5 6 4]	[4 5 6]	—	—	—
GABRIEL DINIZ - PARAQUEDAS	SERTANEJO	[1 5 6 4]	[5 6 3 4]	—	—	—
LUCAS LUCCO - PERMANECER	SERTANEJO	[6 2 5 1]	—	—	—	—
CRISTIANO ARAUJO - MAUS BOCADOS	SERTANEJO	[1 4 2 5]	—	—	—	—
GUSTAVO LIMA - GATINHA ASSANHADA	SERTANEJO	[6 4 1 5]	—	—	—	—
MARILIA MENDONÇA - DE QUEM É A CULPA	SERTANEJO	[1 5 2 4]	[6 5 1 2 1 4]	[6 5 2 4]	—	—
JOÃO BOSCO E VINÍCIUS - NAO ERA VOCE	SERTANEJO	[6 5 2 4]	[4 1 6 5]	—	—	—
JORGE E MATEUS - MEDIDA CERTA	SERTANEJO	[4 1 2 1]	[1 5 4 5]	[1 5 6 4 5]	[2 4 1]	[1 5 2 1 4]
HENRIQUE E JULIANO - VIDINHA DE BALADA	SERTANEJO	[1 2 4]	[2 4 1 5]	[1 5 2 4]	—	—
BRUNO E BARRETO - COPIA MAL FEITA	SERTANEJO	[1 5 4 2]	[1 5 2 4]	—	—	—
MANO WALTER - NAO DEIXO NAO	SERTANEJO	[2 4 1 5]	[4 5 6]	[1 5 2 4]	—	—
FERNANDO E SOROCABA - BOM RAPAZ	SERTANEJO	[2 4 1 5]	[1 5 6 4]	—	—	—
NAIARA AZEVEDO - 50 REAIS	SERTANEJO	[6 4 1 5]	[4 5 6 4]	[1 5 2 4 1]	—	—
JORGE E MATEUS - CONTRATO	SERTANEJO	[1 5 2 4 5]	[4 5 1]	[4 5 6]	—	—
MARCOS E BELUTTI - MAIS UM ANO JUNTOS	SERTANEJO	[1 6 4 5]	[1 6 4 2 5]	—	—	—
JEFFERSSON MORAES - BEBER COM EMERGENCIA	SERTANEJO	[1 5 6 4]	—	—	—	—
MUNOZ E MARIANO - TOMADOR DE WHISKY	SERTANEJO	[1 5 2 4]	—	—	—	—
JONAS ESTICADO - MOCA DO ESPELHO	SERTANEJO	[2 4 1 5]	[1 5 2 4]	—	—	—
HUGO E THIAGO - DECADENCIA	SERTANEJO	[1 5 4]	[2 4 1 5]	—	—	—
JADS E JADSON - NA RIQUEZA E NA POBREZA	SERTANEJO	[4 5 1 6]	—	—	—	—
HENRIQUE E JULIANO - FLOR E O BEIJA-FLOR	SERTANEJO	[1 5 2 4]	[1 5 6 5]	[2 6 4 5]	—	—
HENRIQUE E DIEGO - RASPAO	SERTANEJO	[4 5 6]	[4 5 4 5]	[4 5 6 1 4]	—	—
FRED E GUSTAVO - ACODE AI	SERTANEJO	[1 5 6 4]	—	—	—	—

MÚSICA	ESTILO	PROGRESSÃO 1	PROGRESSÃO 2	PROGRESSÃO 3	PROGRESSÃO 4
IS THIS LOVE - BOB MARLEY	REGGAE	[3 1 5]	[3 2 3 2]	[2 3 4 5]	—
NO WOMAN NO CRY - BOB MARLEY	REGGAE	[1 5 6 4]	[1 4 1]	[1 5 6 4]	—
REDEMPTION SONG - BOB MARLEY	REGGAE	[1 6 4 1 2]	[1 4 5 1]	[4 5 6 4 5 1]	—
THREE LITTLE BIRDS - BOB MARLEY	REGGAE	[1 4 1]	[1 5 1 4]	—	—
ONE LOVE - BOB MARLEY	REGGAE	[1 5 4 1 5 1]	[1 6 4 1]	[1 6 4 5 1]	—
BUFFALO SOLDIER - BOB MARLEY	REGGAE	[5 3 5]	[4 3 4 3]	[3 2 6]	—
QUERO SER FELIZ TAMBÉM - NATIRUTS	REGGAE	[3 2 3 2]	[6 3 6 3]	—	—
ME NAMORA - EDU RIBEIRO	REGGAE	[5 1 5]	—	—	—
GIRASSOL - CIDADE NEGRA	REGGAE	[1 2 3 2]	[1 2 3 5]	—	—
JESUS, O SOL E O REGGAE - SALOMÃO DO REGGAE	REGGAE	[6 5 6 5]	[1 5 6 4 5]	[1 5 4 5 1]	—
DO LADO DE CÃ - CHIMARRUTS	REGGAE	[1 5 6 4]	[6 3 1 5]	[6 4 1 5]	[4 1 5]
NATIRUTS REGGAE POWER - NATIRUTS	REGGAE	[6 3 2 5]	—	—	—
DANCE O REGGAE - SALOMÃO DO REGGAE	REGGAE	[2 3 6]	—	—	—
REGUEIROS GUERREIROS - TRIBO DE JAH	REGGAE	[6 3 6 3]	[6 2 6 2]	[6 5 6 2]	—
DESENHO DE DEUS - ARMANDINHO	REGGAE	[5 1 5 1]	[3 2 3 2]	—	—
ONDE VOCE MORA - CIDADE NEGRA	REGGAE	[1 5 4 2]	[5 4 3 2 1]	—	—
JOHNHY B. GOODE - CIDADE NEGRA	REGGAE	[2 1 2 1]	—	—	—
FOGO NA BABILONIA - EDSON GOMES	REGGAE	[3 6 3 6]	[2 6 2 6]	[5 6 5 6]	[4 1 4 5]
FALA SÓ DE AMOR - EDSON GOMES	REGGAE	[1 5 2 4]	[6 2 6 2]	—	—
INQUILINO DAS PRISÕES - EDSON GOMES	REGGAE	[6 1 4 5 6 5 6]	[2 5 2 5]	[6 2 4 5 6]	—
PERDIDO DE AMOR - EDSON GOMES	REGGAE	[1 4 1 4]	[6 4 5]	—	—
ACORDE, LEVANTE, LUTE - EDSON GOMES	REGGAE	[4 1 4 5]	[4 5 4 6]	[5 6 5 6]	—
SORRI, SOU REI - NATIRUTS	REGGAE	[2 5 6]	—	—	—
ARVORE - EDSON GOMES	REGGAE	[5 6 5 6]	[1 6 1 6]	—	—
CAMPO DE BATALHA - EDSON GOMES	REGGAE	[5 3 5 3]	[6 3 6 3]	[2 3 2 3]	—
LIBERDADE PRA DENTRO DA CABEÇA - NATIRUTS	REGGAE	[1 3 2 4]	[3 2 3 2]	—	—
RASTA COURAGE - S.O.J.A	REGGAE	[4 6 5 6]	—	—	—
REST OF MY LIFE - S.O.J.A	REGGAE	[6 5 4]	—	—	—
YOU AND ME - S.O.J.A	REGGAE	[4 6 5]	—	—	—
OPEN MY EYES - S.O.J.A	REGGAE	[6 4 1 5]	—	—	—
OH CHUVA - PLANTA E RAIZ	REGGAE	[6 3 6 3]	[4 1 4 3]	[6 4 3 6]	—
COM CERTEZA - PLANTA E RAIZ	REGGAE	[6 2 3]	[6 4 5]	—	—
AQUELE LUGAR - PLANTA E RAIZ	REGGAE	[5 1 5 1]	[3 6 3 6]	—	—
BASEADO EM QUE - SALOMÃO DO REGGAE	REGGAE	[1 5 6 4]	—	—	—
IGUAL A VOCÊ - SALOMÃO DO REGGAE	REGGAE	[6 5 2 3]	—	—	—
ANDEI SÓ - NATIRUTS	REGGAE	[6 3 6 3]	[2 3 2 3]	[6 3 2 5]	[6 4 2 5]
VERSOS SIMPLES - CHIMARRUTS	REGGAE	[1 4 5]	[3 2 5 3]	[3 4 5]	—
OUTRA VIDA - ARMANDINHO	REGGAE	[6 3 4 3]	[4 3 2 5]	[3 4 3 4]	—
MORENA RAIZ - TRIBO DE JAH	REGGAE	[1 4 1 4]	[3 6 3]	—	—
BABILONIA EM CHAMAS - TRIBO DE JAH	REGGAE	[5 4 5 4]	—	—	—
NEGUINHA - TRIBO DE JAH	REGGAE	[2 3 6]	[4 5 6 2]	—	—
AONDE VAI CHEGAR - PONTO DE EQUILIBRIO	REGGAE	[1 2]	—	—	—
NOVO DIA - PONTO DE EQUILIBRIO	REGGAE	[3 2 3 2]	—	—	—
SÓ QUERO O QUE É MEU - PONTO DE EQUILIBRIO	REGGAE	[6 3 2]	[5 6 2 3 2 6]	—	—
LEGALIZE IT - PETER TOSH	REGGAE	[5 4 5 4]	—	—	—
GET UP STAND UP - PETER TOSH	REGGAE	[6 3]	—	—	—
DONT LOOK BACK - PETER TOSH	REGGAE	[5 1 5 1]	[1 2 4 5]	—	—
EQUAL RIGHTS - PETER TOSH	REGGAE	[1 2 4 5 1]	[5 1 5 1]	—	—
SOL OU CHUVA - FORFUN	REGGAE	[6 4 5]	[6 4 3]	[6 5 6]	—
MORENA - SCRACHO	REGGAE	[4 5 1 6]	[2 5 1 6 4 5 1 6]	—	—

ANEXO B – Classe que contém os métodos responsáveis pela composição automática

```
public class Compositor {

    public static String compositor(int tom, int estilo){

        ArrayList<Integer> progressao = new ArrayList<Integer>();

        int valor = (int) (Math.random() * 101);
        int i = 0;

        switch(estilo){

            case 1:

                if (valor >= 50){

                    int posicao = (int) (Math.random() * Gramaticas.punk()[0].length);

                    while(i<Gramaticas.punk()[0][posicao].length){
                        progressao.add(i, Gramaticas.punk()[0][posicao][i]);
                        i++;
                    }

                }else if(valor >= 20 ){

                    int posicao = (int) (Math.random() * Gramaticas.punk()[1].length);

                    while(i<Gramaticas.punk()[1][posicao].length){
                        progressao.add(i, Gramaticas.punk()[1][posicao][i]);
                        i++;
                    }

                }else if(valor >= 7 ){

                    int posicao = (int) (Math.random() * Gramaticas.punk()[2].length);

                    while(i<Gramaticas.punk()[2][posicao].length){
                        progressao.add(i, Gramaticas.punk()[2][posicao][i]);
                        i++;
                    }

                }else{

                    int posicao = (int) (Math.random() * Gramaticas.punk()[3].length);

                    while(i<Gramaticas.punk()[3][posicao].length){
                        progressao.add(i, Gramaticas.punk()[3][posicao][i]);
                        i++;
                    }

                }
                break;
            }
        }
    }
}
```

case 2:

```
    if (valor >= 50){  
        int posicao = (int) (Math.random() * Gramaticas.sertanejo()[0].length);  
        while(i<Gramaticas.sertanejo()[0][posicao].length){  
            progressao.add(i, Gramaticas.sertanejo()[0][posicao][i]);  
            i++;  
        }  
    }else if(valor >= 20 ){  
        int posicao = (int) (Math.random() * Gramaticas.sertanejo()[1].length);  
        while(i<Gramaticas.sertanejo()[1][posicao].length){  
            progressao.add(i, Gramaticas.sertanejo()[1][posicao][i]);  
            i++;  
        }  
    }else if(valor >= 7 ){  
        int posicao = (int) (Math.random() * Gramaticas.sertanejo()[2].length);  
        while(i<Gramaticas.sertanejo()[2][posicao].length){  
            progressao.add(i, Gramaticas.sertanejo()[2][posicao][i]);  
            i++;  
        }  
    }else{  
        int posicao = (int) (Math.random() * Gramaticas.sertanejo()[3].length);  
        while(i<Gramaticas.sertanejo()[3][posicao].length){  
            progressao.add(i, Gramaticas.sertanejo()[3][posicao][i]);  
            i++;  
        }  
    }  
    break;
```

case 3:

```
    if (valor >= 50){

        int posicao = (int) (Math.random() * Gramaticas.reggae()[0].length);

        while(i<Gramaticas.reggae()[0][posicao].length){
            progressao.add(i, Gramaticas.reggae()[0][posicao][i]);
            i++;
        }

    }else if(valor >= 20 ){

        int posicao = (int) (Math.random() * Gramaticas.reggae()[1].length);

        while(i<Gramaticas.reggae()[1][posicao].length){
            progressao.add(i, Gramaticas.reggae()[1][posicao][i]);
            i++;
        }

    }else if(valor >= 7 ){

        int posicao = (int) (Math.random() * Gramaticas.reggae()[2].length);

        while(i<Gramaticas.reggae()[2][posicao].length){
            progressao.add(i, Gramaticas.reggae()[2][posicao][i]);
            i++;
        }

    }else{

        int posicao = (int) (Math.random() * Gramaticas.reggae()[3].length);

        while(i<Gramaticas.reggae()[3][posicao].length){
            progressao.add(i, Gramaticas.reggae()[3][posicao][i]);
            i++;
        }

    }

    break;

}
```

```
        return AlgoritmosBasicos.conversorSequenciaToString(getAcordes(tom,progressao));
    }

    public static ArrayList<Integer> getAcordes(int tom, ArrayList<Integer>progressao){
        ArrayList<Integer> acordes = new ArrayList<Integer>();
        for(int l = 0; l < CampoHarmonico.campoHarmonico().length; l++){
            if( CampoHarmonico.campoHarmonico()[l][0]== tom){
                for(int j = 0; j < progressao.size(); j++){
                    acordes.add(j,CampoHarmonico.campoHarmonico()[l][progressao.get(j)-1]);
                }
            }
        }
        return acordes;
    }
}
```