

Um ambiente de programação multiparadigma voltado à composição musical automática

A multi-paradigm programming environment for automatic music composition

DOI: 10.46814/lajdv3n1-039

Recebimento dos originais: 30/10/2020

Aceitação para publicação: 23/12/2020

Clenio B. Gonçalves Junior

Mestre e Doutorando em Ciência da Computação
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)
Universidade Federal de São Carlos (UFSCar)
Rod. Prefeito Quintino de Lima, 2100 - São Roque/SP
E-mail: clenio@ifsp.edu.br

Murillo Rodrigo Petrucelli Homem

Doutor em Ciência da Computação
Universidade Federal de São Carlos (UFSCar)
Rod. Washington Luís, km 235 - São Carlos/SP
E-mail: murillo@dc.ufscar.br

RESUMO

Na produção de software musical, existe uma forte dependência de fatores como linguagens de programação, paradigmas de raciocínio, bibliotecas, APIs e frameworks. Neste trabalho é apresentada a configuração de um ambiente de desenvolvimento que possibilita a criação de software direcionado à produção sonora de modo abrangente, além de atender requisitos específicos da atividade de composição musical automática. Esse ambiente constitui uma plataforma baseada em software livre por meio da utilização de sistemas de código aberto.

Palavras-chave: Computação musical, representação de conhecimento, programação multiparadigma, composição algorítmica.

ABSTRACT

In the production of music software, there is a strong dependence on factors such as programming languages, reasoning paradigms, libraries, APIs and frameworks. This work presents the configuration of a development environment that enables the creation of software directed to sound production in a comprehensive way, in addition to meeting specific requirements of the automatic musical composition activity. This environment constitutes a platform based on free software through the use of open source systems.

Keywords: Music computing, knowledge representation, multi-paradigm programming, algorithmic composition.

1 INTRODUÇÃO

A utilização da música como forma de expressão humana apresenta diferentes formas de manifestação desde seus primórdios - o que inclui, entre outros aspectos, atividades sociais, educativas e terapêuticas [1]. A aplicação musical com propósitos educacionais tem possibilitado o desenvolvimento cultural do indivíduo, potencializando sua capacidade de percepção, integração com o ambiente e consciência [2]. Na área comportamental, desenvolvimentos têm conduzido à criação de oportunidades sociais, além de novas possibilidades a partir de recursos tecnológicos que propiciam formas promissoras de consciência, aumento da autoestima e desenvolvimento coletivo [3]. Na área terapêutica, sensações particulares são produzidas no paciente a partir de sua interação com o componente sonoro. Isso se dá a partir da experiência conduzida por expectativas e predições associados ao bem estar, o que inclui a ampliação de capacidades expressivas e redução de experiências de percepção de dor [4][5].

A pesquisa em Computação Musical (CM) caracteriza-se por combinar aspectos relacionados a diversificadas áreas de conhecimento. Do ponto de vista da composição automática, questões ligadas à síntese sonora, controle de hardware, aprendizado de máquina, musicologia sistemática e processos cognitivos, são comumente abordadas. No que tange à criação de software musical, existe uma forte dependência de fatores como paradigmas de raciocínio, linguagens de programação, bibliotecas, APIs e frameworks, além da escolha de um ambiente de desenvolvimento que possibilite a integração eficiente das tecnologias em questão. Adicionalmente, um fator de grande importância é a capacidade de representação computacional do conhecimento envolvido. Características intrínsecas à atividade musical devem ser mantidas por um modelo que permita, da forma mais natural possível, expressar o raciocínio lógico e a estrutura da informação manipulada [6].

O presente trabalho descreve parte da pesquisa de mestrado voltada à representação computacional de conhecimento musical, essa pesquisa foi constituída a partir de uma abordagem multiparadigma para a configuração de um ambiente de desenvolvimento [7]. Tal ambiente possibilita a realização de síntese sonora, tratamento psicoacústico, aprendizado lógico-indutivo e composição algorítmica. Além disso, a representação do conhecimento envolvido possibilita que características intrínsecas à atividade musical sejam mantidas pelo modelo. Para tanto, foi elaborado um arcabouço de programação visando servir como infraestrutura ao desenvolvimento de aplicações. Desse modo, foram integrados os paradigmas fundamentais de programação. O tratamento sonoro básico foi feito pelas bibliotecas jMusic e JavaSound. A aplicação da programação lógica indutiva foi feita por meio do sistema Aleph. A integração desses componentes foi feita pelo IDE Eclipse, sendo fundamentado na linguagem Java. A integração de Java com Prolog foi feita pelo módulo JPL. Prolog faz o interfaceamento com o sistema indutivo Aleph, também desenvolvido em Prolog. As bibliotecas

jMusic e JavaSound são integradas por meio de Java, assim como a linguagem Scala, sendo esta última realizada por meio do módulo Maven.

2 CONFIGURAÇÃO BÁSICA

Inicialmente, foi abordada a questão dos paradigmas de programação, que constituem o ferramental básico para a representação do conhecimento. Métodos como Sistemas de Produção, Redes Semânticas e Frames são utilizados para essa finalidade. Tais métodos se fundamentam em sistemas matemáticos formais, como ASP (Answer Set Programming) e Programação por Restrições. Tais requisitos são característicos do paradigma declarativo e baseiam-se na Lógica de Primeira Ordem.

Do ponto de vista da abstração de dados, o paradigma orientado a objetos possibilita uma descrição natural dos componentes musicais com a expressividade que as respectivas linguagens fornecem. Nelas, os componentes lógicos e procedurais, necessários à programação imperativa, também são disponibilizados. Atendendo às necessidades declarativas, a linguagem Prolog implementa diretamente a lógica de primeira ordem por meio de cláusulas de Horn. A linguagem Java, além de incorporar as principais características de orientação aos objetos, possui uma biblioteca nativa específica para o processamento de áudio: Java Sound.

Com relação ao tratamento das estruturas musicais, foi realizado um levantamento das plataformas de programação que possibilitassem a manipulação musical. Algumas bibliotecas e frameworks foram testados destacando-se o ambiente jMusic, desenvolvido na Universidade de Tecnologia de Queensland (Brisbane – Austrália). Essa biblioteca implementa elementos sonoros básicos, como frequência tonal, timbre e rítmica, além de componentes musicais complexos como sequenciamento melódico, estruturas harmônicas, encadeamentos e cadências. Por meio de sua API (Application Programming Interface) jMusic possibilita a utilização de formatos para dados musicais variados, incluindo a integração com ferramentas para manipulação de arquivos MIDI, LilyPond e notação de partitura convencional. Além disso, sua implementação inclui o controle do hardware específico possibilitando o trabalho de síntese sonora e fazendo uso de bancos de áudio disponíveis em placas controladoras.

3 LINGUAGENS DE PROGRAMAÇÃO

O modelo foi desenvolvido levando-se em conta os 4 paradigmas fundamentais de programação: Imperativo, Orientado a Objetos, Funcional e Lógico. O conhecimento musical é representado a partir de dois aspectos centrais: a modelagem procedural e a declarativa. A primeira baseia-se em especificações detalhadas de características dos dados, bem como os processos aplicados a eles. A segunda forma de modelagem baseia-se em descrições em nível elevado, sem a dependência

do detalhamento algorítmico. Enquanto na programação procedural as linguagens possuem significado dependente da dinâmica do programa, na programação declarativa as linguagens possuem significado estático [8]. Considerando o aspecto procedimental, os paradigmas Imperativo e Orientado a Objetos serão utilizados por meio da linguagem Java. O aspecto declarativo será aplicado seguindo os paradigmas funcional e lógico, com o uso das linguagens Scala e Prolog. Tais linguagens possuem a capacidade de implementar os principais componentes dos paradigmas adotados e foram escolhidas por serem referências em suas linhas de programação, além terem seu uso estabelecido no contexto das comunidades de desenvolvedores específicas [9].

Destaca-se que as 3 linguagens utilizadas possuem características multiparadigmas. Em Prolog, o aspecto procedimental pode ser definido pela ordem de disposição das cláusulas, além de possibilitar a instanciação de variáveis por meio do predicado `is`. Scala possibilita a utilização de classes, herança por meio de mixins, encapsulamento, além de prover recursos de programação lógica. Java possui os elementos da programação estruturada e inclui também os componentes fundamentais da programação imperativa, como atribuição e sequenciamento de comandos [10]. Considerando o caráter prático deste trabalho, os aspectos multiparadigmas de cada linguagem não serão exploradas, com exceção de Java. Isso deve-se ao fato de que o mesmo se concentra nos paradigmas fundamentais associados a tais linguagens, sendo estas um meio de aplicação dos conceitos essenciais de cada paradigma. Nesse caso, Java é uma exceção pois o paradigma imperativo constitui um elemento básico nessa linguagem, por meio da programação estruturada, métodos, sequenciamento de comandos, programação centrada em atribuições e variáveis com estados múltiplos [11].

O índice TIOBE (<http://www.tiobe.com/tiobe-index>) tem se estabelecido como uma referência no que diz respeito à avaliação de popularidade de linguagens de programação. Ele oferece uma classificação abrangente e atual, utilizando recursos on-line para avaliar o uso de linguagens em todo o mundo. Esse índice é voltado ao ambiente industrial e fornece o ranqueamento das 50 linguagens mais populares nesse meio, contanto com informações atualizadas mensalmente [12]. Conforme classificações TIOBE em dezembro de 2016, a linguagem Java possui a maior classificação entre as linguagens ranqueadas. A lista completa apresenta Prolog com a maior classificação entre as linguagens com o paradigma lógico. Nessa classificação, Scala ocupa posições inferiores com relação ao paradigma funcional.

O índice Stack Overflow de 2016, apresenta aspectos relativos à remuneração referente ao uso das linguagens nos Estados Unidos (<http://stackoverflow.com/research/developer-survey-2016>), bem como ao grau de afeição associado a elas nesse ano. Com relação a esses aspectos, destaca-se o ranqueamento elevado da linguagem Scala. Tanto os índices TIOBE como Stack Overflow, são gerados fundamentalmente a partir de dados envolvendo ambientes de produção associados ao

desenvolvimento de software no âmbito na indústria. Rabai (2015) apresenta a classificação das principais linguagens utilizadas no âmbito acadêmico nos Estados Unidos, nos anos de 2010 e 2013 [12]. Neste aspecto, destacam-se as linguagens Prolog, na primeira posição em 2010 e Java na segunda, sendo essa classificação invertida em 2013.

Do ponto de vista da interface com o usuário, toda forma de comunicação neste trabalho (mensagens, partituras, áudios etc) é realizada pela linguagem Java, devido à capacidade interativa que ela possui; entretanto, para fins de demonstrações específicas, poderão ser considerados os sistemas de entrada e saída de outras linguagens. Além disso, Java será responsável pela mediação entre os módulos de programação (linguagens, APIs e bibliotecas), devido ao seu elevado grau de interoperabilidade e gerenciamento de dependência de módulos [13].

4 CONVENÇÕES SOBRE NOTAÇÃO MUSICAL

Considerando o aspecto da interoperabilidade, a representação simbólica deve atender às características fundamentais das linguagens utilizadas. Tal representação deve levar em consideração a compatibilidade sintática e também possibilitar a especificação de código de modo claro e objetivo.

Dentre as possibilidades relativas à representação dos componentes musicais, alguns padrões tem sido definidos. Esses padrões envolvem a definição de frequências de tonalidades, escrita de notas, acordes, além de outras convenções [14]. Esses componentes serão tratados no decorrer do trabalho, as convenções adotadas são apresentadas a seguir. Com relação à representação de notas musicais, será utilizada a Notação Científica de Alturas - Scientific Pitch Notation (SPN) [15]. Esse padrão foi definido pela ISO 16:1975 - Standard tuning frequency (Standard musical pitch), cuja especificação foi revisada e confirmada em 2011 [16].

Para a variação de oitavas, será utilizada como referência a nota Lá, pertencente à quarta oitava do piano de 88 teclas. Essa nota, referenciada por A4, tem sua frequência definida em 440 Hertz. Essa convenção foi estabelecida em 1925, pela Câmara de Comércio de Indústrias Musicais dos Estados Unidos e adotada internacionalmente em 1939 em acordo firmado em conjunto com a Associação Internacional de Normalização [17]. Tal notação, além de referenciar a oitava em que se encontra uma nota, é utilizada como base para a afinação de instrumentos e categorização de tessituras vocais.

Para tonalidades e graus de escalas, a notação contemporânea será tomada como base [18]. Essa mesma notação será utilizada as tratar de características harmônicas, acordes e funções será utilizada a notação [19]. Os graus indicando a posição dos acordes no campo harmônico ou suas extensões serão referenciados por algarismos romanos (como II ou XI) ou pelo seu próprio nome (como sexta ou grau13). Seguindo essas convenções, a corda mais aguda do violino é representada por E5, os acordes

de Sol Maior e Fá suspenso diminuto, serão representado por GM e FsDim, enquanto um acorde com função mediante (terceiro grau menor) poderá ser representado por III ou grau3.

5 REPRESENTAÇÃO MUSICAL MULTIPARADIGMA

O estudo dos elementos da música ocidental, de modo estruturado, recebe a designação de musicologia. Sua aplicação de modo sistêmico, empírico e baseado em dados é tratada como um processo científico, envolvendo atividades cuja afinidade se apresenta em campos como lógica, gramática, fisiologia, estética e mídia entre outros [20]. Tal abordagem é tratada por musicologia sistemática, onde estes campos são unificados por epistemologias e métodos característicos das ciências duras. Nesse sentido, o desenvolvimento da tecnologia computacional tem estimulado o crescimento em áreas científicas da musicologia, como acústica, matemática, psicologia, pedagogia e neurociências [21].

Ao se tratar o universo musical, a questão da representação da informação envolve aspectos subjetivos e não-conceituais, isso se deve ao fato de que a experiência musical tende a enfraquecer a distinção entre sujeito e objeto. Estruturas linguísticas, variadas formas de expressão musical e regras gramaticais têm sido utilizadas nesse sentido. A música tonal ocidental pode ser tomada como exemplo de tal prática, pois define uma estrutura representacional regida por normas bem definidas. Isso possibilita que as composições musicais deste gênero e suas performances estejam sujeitas a uma avaliação crítica, tanto do ponto de vista técnico como estético [22].

Em termos computacionais, o processo de modelagem da informação musical lida com abordagens formais. Essas abordagens visam representar tanto os elementos fundamentais da música, como harmonia, ritmo, altura e articulações, como também interpretar o modo como esses elementos são aplicados, isto é, o conhecimento [23]. Tendo posse de tal capacidade, pode-se aplicar componentes psicoacústicos na automação de tarefas como análise musical, composição, recomendação musical e categorização de estilos [24][25].

A seguir, serão tratados inicialmente componentes fundamentais da música de acordo com a abordagem proposta e em seguida serão apresentados elementos relativos ao estudo musicológico. As definições apresentadas foram extraídas dos livros “Compêndio de Teoria Elementar da Música” [26], “The physics of music and musical instruments” [27] e “Music: a mathematical offering” [28].

6 FUNDAMENTOS DA REPRESENTAÇÃO SONORA

O som consiste em um fenômeno físico produzido pela vibração de materiais como uma corda, metal ou madeira. Essa vibração desencadeia flutuações de ondas de pressão no ar ao redor do material. A representação gráfica do modo como uma onda oscila ao longo do tempo é chamada de forma de

onda. As funções trigonométricas seno ou cosseno possibilitam que formas de onda simples sejam expressas pela chamada onda senoidal. As ondas senoidais possuem a característica de que, qualquer forma de onda periódica pode ser decomposta em senos ou cossenos de diferentes frequências e amplitudes, conforme estabelecido na Teoria de Fourier [29].

Desse modo, áudio é a combinação de ondas senoidais, cada uma representando um tom ou nota. Uma nota pura consiste em uma única onda senoidal com amplitude e frequência fixas. Frequência corresponde ao número de ondas senoidais que passam por um determinado ponto em um segundo. A frequência é medida em Hertz (Hz), ou ciclos por segundo. Amplitude é a magnitude da oscilação dessas ondas, medida em decibéis (dB). Fase é o deslocamento da onda considerando alguma posição de partida específica. A fase é medida em graus e representa essencialmente a relação de duas ondas com a mesma frequência. O Período de uma onda indica o tempo gasto para que ela complete um ciclo.

O gráfico de uma nota pura, onde o eixo das abscissas expressa o tempo e o eixo das ordenadas expressa a amplitude, com frequência f , amplitude máxima A e fase inicial θ pode ser representado por uma onda senoidal. A função que representa essa onda é dada por:

$$y = A \text{ sen}(2\pi f x + \theta)$$

Esta equação expressa um fenômeno contínuo, entretanto, os computadores impõem a necessidade da modelagem discreta, onde o som é representado por uma lista de valores que indicam sua amplitude em intervalos uniformemente espaçados. A quantidade desses valores, dispostos no transcorrer do tempo, é chamada de taxa de amostragem.

Na programação de um computador, o Paradigma Imperativo apresenta recursos de programação fundamentais para a manipulação das grandezas citadas. Tais recursos constituem um modo natural de representação por meio de atribuições, sequências, repetições de comandos e sub-rotinas.

Com relação ao ser humano, o aspecto sensorial ocorre como resultado da propagação das ondas sonoras no ar ou em outro meio. Desse modo, o som poderá ser percebido após as ondas terem atingirem o ouvido. A frequência da onda relaciona-se com a altura de um som e definem seu caráter mais grave ou agudo, a amplitude refere-se à intensidade ou volume, enquanto que formas de onda diferentes influenciam no timbre do som produzido. O intervalo de frequências que o ouvido humano pode perceber varia aproximadamente de 20 Hz a 20.000 Hz. A taxa de amostragem de 44.100 Hz, comumente usada em dispositivos eletrônicos, decorre do teorema da amostragem de Nyquist–

Shannon, que estabelece que a frequência amostrada deve ser igual ou superior ao dobro da frequência máxima do sinal digital [29].

Tomando-se o piano como base, a nota Dó central, ou nota fundamental, vibra a uma frequência de 261,626 Hz. Essa nota é referenciada por C4 por encontrar-se no 4º grupo completo de notas do piano. Nesse mesmo grupo, a nota Lá (A4) possui a frequência de 440 Hz (frequência fundamental) e no grupo de notas anterior, a nota A3 possui uma frequência de 220 Hz. A modelagem digital possibilita que o fenômeno sonoro seja representado por uma sequência de inteiros, isso simplifica o seu armazenamento, reprodução e transformação. Desse modo, atividades envolvendo análise, representação gráfica, cifragem e outros meios de notação podem ser realizadas. Considerando o aspecto imperativo, a implementação na linguagem Java possibilita que a respectiva onda sonora seja produzida. A nota A3, modelada por esse algoritmo, pode ser sintetizada, armazenada e reproduzida por meio da API JavaSound.

7 MONOFONIA

Uma nota é um dos componentes mais elementares em um som musical. O som, por sua vez, possui propriedades básicas que são utilizadas pelo computador como atributos para representação digital nota:

- Altura - frequência de uma nota, a característica que define se um som é mais grave ou agudo.
- Duração - tempo de produção do som, define o valor rítmico da nota.
- Intensidade - a propriedade do som ser mais fraco ou mais forte, define seu valor dinâmico.
- Timbre - a qualidade do som, característica associada à fonte produtora do som.

De acordo com a especificação do **Paradigma Orientado a Objetos**, a criação de uma instância `minhaNota`, pertencente a uma classe `Nota`, pode ser feita utilizando-se quatro parâmetros. Esses parâmetros referem-se aos atributos que caracterizam respectivamente as propriedades citadas:

```
minhaNota = new Nota(E4, COLCHEIA, FORTE, FAGOTE)
```

Pausa é um intervalo de silêncio na música, cuja duração pode assumir os mesmos valores referentes aos de uma nota. Linguagens de programação que implementam o mecanismo, em orientação a objetos, de sobrecarga do método construtor, possibilitam representações do tipo:

```
pausa = new Nota(PAUSA, COLCHEIA)
```

Desse modo, uma pausa pode ser tratada como uma nota cujo atributo de altura indique o silêncio e excluindo-se os atributos de intensidade e duração.

Melodia é uma sucessão de notas que podem assumir alturas e valores diferentes, bem como serem repetidos, obedecendo a um sentido lógico musical. Uma frase melódica constitui uma sequência coerente de notas, expressando, de modo geral, um pensamento articulado no transcorrer da música. Estruturas de dados do tipo coleções, como conjuntos, listas, mapas ou vetores, constituem um recurso para armazenamento, manipulação e recuperação de dados melódicos. Operações como repetição, transposição, modulação etc podem ser aplicadas tanto a melodias básicas como a frases melódicas interconectadas, valendo-se da expressividade e eficiência dessas estruturas. Desse modo, uma classe Frase pode ser definida como um contêiner para notas:

```
frase = new Frase(5) frase.insereNota(minhaNota)
```

Como as frases podem estar dispostas em variados pontos de uma composição, é conveniente que, ao serem criadas, o seu tempo inicial seja especificado. Posteriormente, o objeto frase poderá ser reutilizado, assumindo um novo valor para o tempo inicial. O código anterior, apresenta a criação de uma frase iniciando no segundo compasso de uma música quaternária, incluindo a adição de uma nota à ela:

8 POLIFONIA

Enquanto na monofonia os elementos musicais são concebidos a partir de uma única melodia, a polifonia trata a música do ponto de vista de variadas linhas melódicas sobrepostas e independentes entre si. Existe a possibilidade de integração de múltiplos instrumentos ou vozes com a característica de se contrapor ao longo de uma obra.

Nesse sentido, a união de frases musicais constitui a forma de organização básica para uma voz ou instrumento. De acordo com contextos específicos, essa estrutura possui variadas denominações: voz, instrumento, sistema, pauta. Por questão de generalidade, neste trabalho será utilizada a última. Ao se estabelecer uma pauta, é conveniente especificar o elemento sonoro ao qual ela se refere (um piano, por exemplo). Esse elemento será utilizado como base para a determinação do timbre de todas as notas pertencentes a essa pauta. Considerando o padrão MIDI, a especificação de tal elemento deve estabelecer tanto o instrumento como o canal respectivos. Desse modo, o atributo do timbre pode ser

suprimido na criação de uma nota, indicando que será utilizado o padrão definido pela pauta. Pode-se também incluir uma descrição do instrumento, a ser apresentada na transcrição inicial de cada pauta:

```
pauta = new Pauta(Violino_principal, VIOLINO, 2)
pauta.insereFrase(frase)
```

Uma partitura consiste na integração dos elementos anteriores, formando o último nível para a estruturação básica de uma música. Sua definição possibilita a junção de pautas dispostas em paralelo. Além disso, a partitura contém o andamento da composição, especificado em batidas por minuto (bpm). Adicionalmente, pode-se incluir o título da composição, associado à partitura:

```
musica = new Partitura(Concerto_para_cordas, 120)
musica.inserePauta(pauta)
```

A aplicação de tais conceitos à linguagem Java, por meio da biblioteca jMusic possibilita a utilização de classes, operadores e constantes específicos para o estabelecimento dos componentes fundamentais de uma obra musical. A listagem abaixo apresenta a estrutura base para representação de um trecho da obra “Judas Maccabaeus”, de George Frideric Handel [30].

```
public final class maccabaeus implements JMC {
    Score partitura;
    Part homens, mulheres, cordas;
    Phrase fraseH, fraseM, fraseC;

    public static void main(String[] args) {
        homens = new Part("Homens", OOH, 0);
        mulheres = new Part("Mulheres", OOH, 1);
        cordas = new Part("Cordas", SLOW_STRINGS, 2);
        partitura = new Score("Maccabaeus", 106.0);

        criaMusica();
        partitura.addPart(homens); partitura.addPart(mulheres);
        partitura.addPart(cordas);
        Play.midi(partitura);
    }
}
```

9 TEMPERAMENTO

A distância existente entre duas notas diferentes é denominada intervalo. Quando uma nota possui o dobro da frequência de outra, esse intervalo é chamado de oitava. Dividindo-se a oitava em

12 partes, pode-se obter o conjunto de notas abrangidas na música ocidental. O método utilizado para essa divisão define o sistema de afinação aplicado a um instrumento musical. Esse sistema é chamado de temperamento. No decorrer da história, gama de sistemas de afinação foi proposta. Dentre eles, o mais básico consiste na simples divisão da oitava em 12 partes iguais, sendo denominado de temperamento Igual. Além dele, alguns outros têm sido objeto de estudos, dentre os quais pode-se destacar os temperamentos Pitagórico, Justo e Mesotônico.

Ao se afinar um instrumento, devem ser aplicadas as razões específicas às 12 notas da oitava para que o temperamento desejado seja obtido. A expressividade representativa, característica da Programação Funcional possibilita que os cálculos das razões de temperamento sejam realizados com eficiência e clareza [8]. O método `map`, da linguagem Scala, retorna um contêiner de pares atributo/valor. A utilização desse método na representação de temperamentos possibilita a aplicação das razões para cada uma das 12 notas da oitava, atendendo aos quatro sistemas de temperamento citados.

Como exemplo, a razão referente a uma terça menor no sistema Pitagórico é obtida pela chamada:

```
temp("pitag")(3)
```

O conceito de dado imutável, definido em programação funcional, refere-se a elementos que não podem ter seus valores alterados no transcorrer de um programa. Esse recurso possibilita que um temperamento específico seja atribuído à uma composição musical, de modo que as notas sejam produzidas diretamente de acordo com um determinado sistema. O uso da declaração `val`, de Scala, permite que um temperamento seja fixado, variando-se apenas as alturas das notas:

```
val tempBase = "mesotonico"
  temp(tempBase)(2)
  temp(tempBase)(11)
```

10 HOMOFONIA

Os componentes tratados nas seções anteriores constituem uma visão horizontal com relação à forma de pensamento musical. O caráter homofônico determina a dimensão vertical: aonde se tratava uma nota independente, passa-se a considerar o efeito de notas em conjunto. Enquanto na polifonia as vozes se articulam separadamente, a homofonia considera a concatenação de grupos de notas, chamados de acordes.

Um acorde é produzido pela combinação de sons musicais diferentes, sendo gerados de modo integrado. A ocorrência de um acorde é definida pela superposição de notas que irão soar simultaneamente. O estudo dos intervalos estabelece as relações básicas entre notas e define o modo como elas serão unidas.

O **Paradigma Lógico** de programação fornece elementos que possibilitam a descrição direta de notas, alturas e seus respectivos intervalos. Tais elementos consistem na utilização de fatos, regras, cláusulas, predicados lógicos, além do mecanismo de inferência dedutiva aplicado sobre eles. Esses elementos possibilitam tanto a modelagem, como a representação do raciocínio utilizado na seleção, integração e sequenciamento musical. Com isso, o conhecimento respectivo pode ser codificado declarativamente, possibilitando atividades como análise, descrição, e estruturação musical.

Considerando as relações existentes entre os intervalos de notas, as regras e os padrões aplicados na sua integração, a linguagem Prolog possibilita que essas relações sejam especificadas de modo que as 12 notas de uma oitava sejam representadas por fatos como:

```
escala ([ 'C', 'Cs', 'D', 'Ds', 'E', 'F', 'Fs', 'G', 'Gs', 'A', 'As', 'B' ]).
```

A estrutura básica utilizada na declaração escala é a lista. Neste caso, ela está sendo formada por átomos, que são tipos de dados fundamentais nas linguagens lógicas. Para compor a escala musical, estão sendo utilizados 12 átomos, representando as notas musicais.

Os sistemas Prolog utilizam uma série de predicados predefinidos por padrão, sendo disponibilizados nas implementações da linguagem, como nth0(P, L, E). Esse predicado pode ser utilizado para retornar a posição de um elemento em uma lista, sendo satisfeito quando o elemento E está na posição P da lista L. Desse modo, a definição de um intervalo de Terça Maior pode ser feita pela cláusula:

```
tercaMaior(Fundamental, Terca) :-
    escala(E),
    nth0(PosFund, E, Fundamental),
    PosTer is PosFund + 4,
    nth0(PosTer, E, Terca).
```

Nesse caso, a tentativa de satisfação da meta: tercaMaior(D, X), resultará na instanciação: X = Fs. Analogamente, um intervalo de quinta justa pode ser obtido pelo incremento de 7 posições a partir da fundamental.

O predicado predefinido `append(Lista1, Lista2, Lista3)` realiza a concatenação das 2 primeiras listas, produzindo a terceira. Seguindo a linha de definições apresentadas, uma tríade (acorde com 3 notas) Maior pode ser representada pela listagem a seguir.

```

triadeMaior(Fund, Triade) :-
    insereFund(Fund, L1),
    insereTerca(L1, L2),
    insereQuinta(L2, Triade).

insereFund(Fund, [L]) :-
    append([], Fund, L).

insereTerca([C|R], L) :-
    tercaMaior(C, Ter),
    append([C|R], [Ter], L).

insereQuinta([C|R], L) :-
    quintaMaior(C, Qui),
    append([C|R], [Qui], L).
  
```

Como exemplo, uma chamada à meta `triadeMaior(D,T)` será satisfeita com a instânciação $T = [D, Fs, A]$.

A representação simbólica das notas musicas como C, Cs, D, constitui uma definição básica do paradigma lógico, por meio de átomos. Entretanto, o padrão MIDI estabelece números inteiros para essa representação. A nota dó central (C4), por exemplo, é representado pelo número 60, enquanto que as notas ré, mi e fá subsequentes são representadas pelos números 62, 64 e 65. Um fato em Prolog é uma afirmação sempre verdadeira como `homem(joao)`, `pai(joao, maria)` ou `relativa(D, Bm)`. Utilizando-se desse recurso, podem ser estabelecidas relações lógicas que possibilitem associações como `midi(C,60)` ou `midi(Cs,61)`.

A listagem abaixo apresenta o predicado recursivo `converteMidi(L1, L2)`. Ele recebe uma lista L1 contendo a representação simbólica de notas e produz L2 com a representação numérica:

```

midi('C',60).   midi('Cs',61).   midi('D',62).   midi('Ds',63).
midi('E',64).   midi('F',65).   midi('Fs',66).   midi('G',67).
midi('Gs',68).  midi('A',69).   midi('As',70).   midi('B',71).
converteMidi([], []).
converteMidi([C|R], [C1|R1]) :-
    midi(C, C1),
    converteMidi(R, R1).
  
```

O modelo de representação apresentado garante a interoperabilidade entre as linguagens. Como jMusic utiliza o padrão MIDI, uma lista de notas em Prolog pode ser diretamente acessada por Java e armazenada em um vetor de inteiros. Desse modo, ambas estruturas terão o mesmo valor semântico quanto à representação do componente musical, armazenando as respectivas alturas das notas.

Em jMusic, a estrutura necessária para se manter um acorde é semelhante à de uma nota; entretanto, ao invés de frases melódicas, serão manipuladas frases harmônicas. A aistagem a seguir apresenta a especificação de uma frase harmônica com dois acordes utilizados na representação de trecho da canção “Maria, Maria”, de Milton Nascimento e Fernando Brant [31].

```
int[] miB = AS3,DS4,G4;  
int[] siM7 = A3,D4,FS4;  
fHarm = new Cphrase(0);  
fHarm.addChord(miB, HN);  
fHarm.addChord(siM7, HN);
```

11 CONCLUSÃO

A desenvolvimento realizada levou a um ambiente híbrido que atende aos requisitos fundamentais para CM. Possibilitou-se a programação multi-paradigma, manipulação e síntese sonora, bem como a representação do conhecimento musical. Obteve-se, deste modo, uma plataforma centralizada em Java, à qual foram incorporadas APIs para as bibliotecas Java Sound e jMusic. Com a integração do mecanismo de Prolog pode-se conseguir acesso às funcionalidades declarativas. Isso resultou em que o engenho de inferência para lógica de primeira ordem pudesse ser acionado diretamente a partir do código Java. Testes realizados indicam que a técnica de programação multiparadigma aliada à programação lógica indutiva possibilitou um método efetivo para desenvolvimento do processo de representação de conhecimento musical. Dentre as possibilidades de aplicação, destaca-se a composição musical algorítmica, com a capacidade de produção de trechos que podem ser associados a determinados contextos humanos.

AGRADECIMENTO

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

- Raglio, A., & Vico, F. (2017). Music and technology: The curative algorithm. *Frontiers in Psychology*, 8, 2055.
- Faria, de L. C. F., Franceschini, S. R., & Rocha, da K. R. (2020). O uso dos jogos eletrônicos na educação musical e sua possibilidade enquanto recurso didático. *Latin American Journal of Development*, 2(5), 304–310.
- Reis, A. C., & Chevitaese, M. J. (2020). Canto coral-espaco de aprendizagem e desenvolvimento musical. *Latin American Journal of Development*, 2(5), 243–250.
- Salimpoor, V. N., Zald, D. H., Zatorre, R. J., Dagher, A., & McIntosh, A. R. (2015). Predictions and the brain: How musical sounds become rewarding. *Trends in Cognitive Sciences*, 19(2), 86–91.
- Souza, de M. C. M. M., & others. (2020). Um novo amanhecer através da arteterapia... Resgatando relacionamentos, após a mastectomia. *Latin American Journal of Development*, 2(5), 186–189.
- Nierhaus, G. (2009). *Algorithmic composition: Paradigms of automated music generation*. Springer Science & Business Media.
- Gonçalves Junior, C. B. (2017). Representação multiparadigma de conhecimento musical utilizando programação lógica indutiva.
- Sebesta, R. W. (2012). *Concepts of programming languages (Vol. 281)*. Pearson Education.
- Gil, J. Y. (2015). CS 234319: Programming languages.
- Naim, R., Nizam, M. F., Hanamasagar, S., Nouredine, J., & Miladinova, M. (2010). Comparative studies of 10 programming languages within 10 diverse criteria-a team 10 comp6411-s10 term report. ArXiv Preprint ArXiv:1008.3561.
- Guimaraes, J. O. (2015). *Programming languages paradigms (pp. 96–114)*. UFSCar Sorocaba.
- Rabai, L. B. A., Cohen, B., & Mili, A. (2015). Programming language use in us academia and industry. *Informatics in Education*, 14(2), 143.
- Toal, R., Rivera, R., Schneider, A., & Choe, E. (2016). *Programming language explorations*. CRC Press.
- Queiroz, de A. A. (2010). Uma notação musical para representação de progressões harmônicas utilizando grafos. *Revista Música Hodie*, 9(1).
- Malheiro, F., & Cavaco, S. (2011). Automatic musical instrument and note recognition. ISMIR.
- ISO. (1975). *Standard tuning frequency (standard musical pitch)*. Geneva, Switzerland: International Organization for Standardization.
- Weinstein, J. L. (1952). Musical pitch and international agreement. JSTOR.

Schoenberg, A. (1990). Fundamentos da composição musical. Edusp.

Adolfo, A. (1989). O livro do músico. Irmãos Vitale.

Cabral, T. (2014). Musicologia sistemática, humanismo e contemporaneidade. OPUS-Revista Eletrônica Da ANPPOM, 20(2), 125–150.

Parncutt, R. (2012). Musicologia sistemática: A história e o futuro do ensino acadêmico musical no ocidente. Em Pauta, 20(34/35), 145–185.

Nussbaum, C. O. (2007). The musical representation: Meaning, ontology, and emotion. Mit Press.

Dannenberg, R. B. (1993). A brief survey of music representation issues, techniques, and systems.

Roads, C. (1996). The computer music tutorial. MIT press.

Gebhardt, R. B., Davies, M. E., & Seeber, B. U. (2016). Psychoacoustic approaches for harmonic music mixing. Applied Sciences, 6(5), 123.

Lacerda, O. (2008). Compendio de teoria elementar da musica. Ricordi do brasil.

Lapp, D. R. (2003). The physics of music and musical instruments. Wright Center for Innovative Science Education, Tufts University.

Benson, D. J. (2008). Music: A mathematical offering. The Mathematical Intelligencer, 30(1), 76–77.

Roberts, M. J. (2009). Fundamentos de sinais e sistemas. AMGH Editora.

Handel, G. F., & Morell, T. (1747). Judas maccabaeus. Courier Corporation.

Nascimento, M., & Brant, F. (1983). Maria maria. editado y distribuido por Fonogram.