Clenio B. Gonçalves Junior

# Representação Multiparadigma de Conhecimento Musical Utilizando Programação Lógica Indutiva

**Sorocaba, SP**

**06 de Fevereiro de 2017**

Clenio B. Gonçalves Junior

# Representação Multiparadigma de Conhecimento Musical Utilizando Programação Lógica Indutiva

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Teoria Aplicada à Computação.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Orientador: Prof. Dr. Murillo Rodrigo Petrucelli Homem

Sorocaba, SP

06 de Fevereiro de 2017

Gonçalves Jr., Clenio B.

Representação Multiparadigma de Conhecimento Musical Utilizando Programação Lógica Indutiva / Clenio B. Gonçalves Jr.. -- 2017.
113 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador: Murillo Rodrigo Petrucelli Homem
Banca examinadora: José Eduardo Fornari Novo Junior, Flávio Luiz Schiavoni
Bibliografia

1. Computação musical. 2. Representação de conhecimento. 3. Programação lógica indutiva. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).
DADOS FORNECIDOS PELO(A) AUTOR(A)

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

## Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Clenio Batista Gonçalves Jr, realizada em 06/02/2017:

Prof. Dr. Murillo Rodrigo Petrucelli Homem
UFSCar

Prof. Dr. José Eduardo Fornari Novo Junior
UNICAMP

Prof. Dr. Flávio Luiz Schiavoni
UFSJ

*Dedico esse trabalho aos meus pais*
*Clenião e Lóidinha*
*que sempre me deram mais do que tinham*
*muito além das suas possibilidades*
*E ainda hoje sou agraciado*
*pela fonte inesgotável de seu amor*

# Agradecimentos

Esse trabalho é a consolidação de esforços profissionais e pessoais. Algumas pessoas muito importantes estiveram ao meu lado, às quais sou imensamente grato:

Ao professor Murillo, meu orientador, por todas as formas de apoio, obrigado por ter acreditado no trabalho e viabilizado essa oportunidade que me proporcionou um grande crescimento.

À minha família pelo apoio incondicional, atemporal, assíncrono, incessante ...

À menina Maria Júlia, pelas informações fornecidas sobre a obra de seu pai, Paulinho Nogueira, pelo acervo disponibilizado, pela amizade.

À Ana Paula, minha princesa_amada_muitoMaisQueEsposa, você esteve comigo durante todo esse trabalho, viagens, noites em claro, alegrias, dificuldades. Nossas conversas foram fonte de inspiração para muitas coisas que estão aqui.

À minha sogra, dona Zê, por todo apoio, sempre me lembrarei dos cafezinhos e doces pra me manter acordado na hora de "fazer a lição" no computador.

Por fim, agradeço a quem é o princípio de todas as coisas, fonte do conhecimento e da vida. E ao mesmo tempo, o destino de tudo.

*"Existe uma paixão pelo entendimento,*
*tal como existe uma paixão pela música.*
*Essa paixão é comum nas crianças,*
*mas a maioria das pessoas*
*perde-a posteriormente.*
*Sem essa paixão, não teria havido*
*matemática, nem ciências naturais."*

*Albert Einstein*

# Resumo

O processo de representação de conhecimento em Computação Musical constitui um elemento essencial para o desenvolvimento de sistemas. Métodos têm sido aplicados visando fornecer ao computador a capacidade de inferir informações a partir da experiência e definições previamente estabelecidas. Neste sentido, a Programação Lógica Indutiva apresenta-se como um crescente campo de pesquisa que incorpora conceitos de Programação em Lógica e Aprendizado de Máquina.

O presente trabalho aborda a Representação de Conhecimento Musical sob a ótica da programação multiparadigma, com uso da técnica de Programação Lógica Indutiva. Inclui o desenvolvimento do sistema musical baseado em conhecimento Fraseado. Por fim é apresentado um método para avaliação de sistemas de composição algorítmica - o Teste de Turing Expandido.

**Palavras-chave**: Computação musical. Paradigmas de programação. Representação de conhecimento. Programação lógica indutiva. Programação multiparadigma.

# Abstract

Knowledge representation process is an essential matter regarding Computer Music systems. Methods have been applied in order to provide computers with the capability to generate conclusions based on experience in specialized domains. Inductive Logic Programming is a research field which combines concepts of Logic Programming and Machine Learning. Due to its declarative feature, both acquired and produced knowledge can be presented to not-expert users in a naturally understandable way.

This work deals with Music Knowledge Representation from the perspective of multi-paradigm programming, using Inductive Logic Programming technique and including the development of the knowledge-based music system Fraseado. Finally, a method for the evaluation of algorithmic composition systems - the Expanded Turing Test - is presented.

**Keywords**: Computer music. Programming paradigms. Knowledge representation. Inductive logic programming. Multi-paradigm programming.

# Lista de ilustrações

# Lista de códigos

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| MIDI | Musical Instrument Digital Interface |
| PLI | Programação Lógica Indutiva |
| RC | Representação de Conhecimento |
| RCM | Representação de Conhecimento Musical |
| SBC | Sociedade Brasileira de Computação |
| API | Application Programming Interface |
| FBF | Fórmula Bem Formada |
| CA | Composição Algorítmica |
| TT | Teste de Turing |
| IDE | Integrated Development Environment |

# Prefácio

*A vida é como uma sala de espetáculos; entra-se, vê-se e sai-se.*

---

Pitágoras de Samos (c. 570 - c. 495 a.C.)

Esse trabalho foi desenvolvido com o intuito de contribuir com a pesquisa em Ciência da Computação.

O principal foco concentra-se no campo da Computação Musical, cujo caráter transdisciplinar conduziu à realização de estudos envolvendo temas como Paradigmas de Programação, Engenharia de Software e Programação Lógica Indutiva.

Nesse sentido, ao serem abordados, adicionalmente, assuntos ligados à Inteligência Artificial, como Representação de Conhecimento, Aprendizado de Máquina e Composição Algorítmica, a noção literal da "arte da programação de computadores" encontra um vasto ambiente para desenvolvimento.

A união de tais elementos com o trabalho de investigação científica gerou a principal motivação para a realização da presente obra.

# Sumário

# PARTE I
# FUNDAMENTAÇÃO

# Prelúdio I

*Se a informação musical fosse bem compreendida e fixa,*
*sua representação seria um problema muito mais simples.*

---

Roger B. Dannenberg (1955 - )

A primeira parte deste trabalho, consistindo de 3 capítulos, é resultante das pesquisas realizadas no sentido de identificar a atuação da Computação Musical e suas relações com áreas como Engenharia de Software, Inteligência Artificial e Paradigmas de Programação.

Algumas questões nesse sentido são apresentadas na introdução do trabalho (Capítulo 1) e em sua contextualização (Capítulo 2). Durante esse processo, percebeu-se a problemática no que diz respeito à Representação de Conhecimento Musical.

Visando identificar contribuições nesse escopo, detectou-se a programação multiparadigma aliada à Programação Lógica Indutiva como possibilidades relevantes.

No sentido de avaliar trabalhos em Computação Musical que utilizassem a Programação Lógica Indutiva, foi realizada uma Revisão Sistemática da Literatura, a qual é apresentada no Capítulo 3.

# 1 Introdução

*O que poderíamos descobrir se*
*estudássemos o pensamento musical?*

MARVIN L. MINSKY (1927 - 2016)

Computação Musical (CM) é um campo de pesquisa em Ciência da Computação com características fortemente interdisciplinares (MILETTO et al., 2004). Estudos são direcionados a aspectos que envolvem conceitos como interação humano-computador, sistemas de recomendação, inteligência computacional, projeto de hardware, educação auxiliada por computador, sistemas interativos de tempo real, entre outros. Em seu processo de desenvolvimento, diversificadas áreas de conhecimento - como Pedagogia, Saúde, Engenharias e Psicologia - têm sido relacionadas em variados tipos de aplicações (GIMENES; SANTOS; MANZOLLI, 2003).

De acordo com Sociedade Brasileira de Computação (SBC), Computação Musical "compreende pesquisa científica, tecnológica e artística nas áreas de composição algorítmica, análise/síntese de som, acústica musical, análise musical assistida por computador, composição musical assistida por computador, processamento digital de áudio, multimídia e qualidade de serviço" (SBC, 2015). Atualmente, CM se enquadra em dois grupos temáticos da SBC: GA3 - Técnicas e Tecnologias de Computação e GA4 - Aplicações da Computação.

Uma decorrência de tal vastidão de conhecimento é que os desenvolvimentos realizados nesta área possibilitam ampliar a compreensão de conceitos relacionados a outros campos de pesquisa. Pode-se tomar como exemplo o estudo de processos cognitivos associados à atividade de um compositor musical (LIMA, 1998; OLIVEIRA, 2007). Além destes, a investigação de elementos ligados aos ramos da musicoterapia, comunicação, aspectos legais, sociais, éticos e de negócios, possibilita realizar a análise, processamento e síntese de dados acústicos e simbólicos (FORNARI, 2015), trazendo importantes contribuições em cada *métier*. Em Ciência da Computação, estudos em CM têm fornecido auxílio no trabalho envolvendo interação humano-computador, paradigmas de programação, processos cognitivos etc (MINSKY, 1981).

Dentre os aspectos abordados, um tópico de interesse é o que lida com a questão da música gerada por computador (WHALLEY, 2005). Historicamente, elementos ligados à composição algorítmica e criação automática de trechos musicais têm obtido especial destaque (FRANCIS, 2015). Métodos têm sido desenvolvidos no sentido de que o computador seja capaz de obter conhecimento e adquirir experiência em domínios especializados[1]. Tais métodos possibilitam que a máquina seja capaz de exibir comportamento equivalente

[1] Os termos *conhecimento* e *experiência* são jargões em Inteligência Artificial, considerados no contexto de métodos avançados para manipulação da informação.

a um ser humano no que tange a características como manifestações culturais, expressões performáticas e assimilação de estilos específicos a um determinado compositor ou período histórico (J MCKAY C, 2015; MAXIMILIANO et al., 2015).

Em Inteligência Artificial particularmente, ocorre uma variedade de trabalhos ligados à CM, mantendo estreita relação e cuja afinidade remonta várias décadas atrás (FERNÁNDEZ; VICO, 2013; WIGGINS; SMAILL, 2000; DELGADO; FAJARDO; MOLINA-SOLANA, 2011; ROADS, 1985). Nessa área, intimamente ligada à abordagem simbólica, existe a questão de como representar computacionalmente aquilo que diga respeito a um campo de conhecimento específico - Representação de Conhecimento (RC). Esse tema é citado na literatura como sendo de fundamental importância no transcorrer dos processos envolvidos em computação musical (RAMIREZ; HAZAN, 2005). RC aborda questões de modelagem da informação, de modo que o computador tenha a capacidade de extrair conclusões lógicas de maneira eficiente, possibilitando a execução de atividades complexas que envolvam componentes como raciocínio e criatividade. De acordo com Miranda (MIRANDA; ALVARO; BARROS, 2005), a definição de uma ferramenta adequada para representação de conhecimento é fundamental para que o desenvolvimento de um sistema de inteligência artificial seja bem sucedido.

Adicionalmente, um importante aspecto relacionado ao tema trata de questões ligadas à representação musical pelo computador. Elementos como notação simbólica, formatos de arquivo, recursos para codificação, técnicas de programação e abstração de dados têm sido estudados no nível do processamento da informação (SMAIL; WIGGINS; MIRANDA, 1994). Deste modo, o trabalho envolvendo conhecimento musical aborda os aspectos que dizem respeito a paradigmas de programação, características relativas a lógicas formais, métodos de busca, tomada de decisão eficientes e aprendizado - esse processo é designado Representação de Conhecimento Musical (RCM) (MANTARAS, 2006).

Sendo assim, a modelagem da estrutura de RC é de suma importância em atividades de computação musical, sendo que os processos envolvendo tal atividade devem compor a primeira fase no desenvolvimento de um sistema (BALABAN, 1996). De forma mais específica, a modelagem declarativa, por meio da Lógica de Primeira Ordem, constitui um importante método. Tanto sua estrutura descritiva como os mecanismos de aplicação levantam questões particularmente interessantes (WHALLEY, 2005). Tal modelo apresenta um formalismo natural para a definição de componentes musicais, possibilitando que inferências sejam realizadas sobre uma base musical previamente definida, além de propiciar a descoberta de novos padrões estruturais.

Nesse contexto, a Programação Lógica Indutiva (PLI) apresenta-se como um campo de pesquisa crescente que combina os conceitos de Programação em Lógica e Aprendizado de Máquina (MORALES; MORALES, 1995). PLI baseia-se na lógica de primeira ordem,

o que lhe confere um caráter declarativo, isso permite que o resultado do processamento gerado por um sistema possa ser apresentado a usuários não especialistas de uma forma simples e intuitiva (DIXON; MAUCH; ANGLADE, 2011). Por meio da programação lógica indutiva, novo conhecimento musical pode ser gerado automaticamente a partir da derivação de estruturas e regras expressas na forma de cláusulas de Horn (MORALES, 1994). Uma série de modelos têm sido propostos em trabalhos envolvendo conceitos como contraponto, expressividade performática, representação harmônica, inferência modal, entre outros. Adicionalmente, a PLI tem sido aplicada como um eficiente recurso para representação de conhecimento no que tange a caracterização de gêneros e estilos musicais (POMPE; KONONENKO; MAKSE, 1996; NUMAO; TAKAGI; NAKAMURA, 2002; ANGLADE et al., 2010; DIXON; MAUCH; ANGLADE, 2011; PEREZ; RAMIREZ; KERSTEN, 2008). Variados sistemas que implementam indução lógica têm sido utilizados em aplicações musicais específicas, tais como Aleph (SRINIVASAN, 2007), TILDE (BLOCKEEL, 1999) e PAL (MORALES, 1994). Tais sistemas realizam o processo indutivo a partir de linguagens como Prolog. Além disso, teorias cognitivas têm sido aplicadas na análise musical, como Narmour (SAKHARE; HANCHATE, 2014), visando a compreensão de estruturas melódicas.

## 1.1 Motivação e justificativa

O aspecto transdisciplinar aqui tratado implica em uma consequência adicional relativa à grande abrangência de ramos de conhecimento envolvidos. Isso impacta o fato de que determinados componentes ainda demandam estudos mais aprofundados. Pode-se observar a existência de lacunas no que tange à especificação de padrões para experimentos, testes, parâmetros e métricas para avaliação e comparação de métodos. Em muitos casos, a representação de conhecimento musical é tratada de maneira não tópica, citam-se determinadas opções escolhidas e comentam-se alguns motivos para a tomada de decisão, sem que seja realizado um estudo completo das características envolvendo essa questão.

Com o intuito de estabelecer critérios mais acurados para que os elementos em questão sejam devidamente fundamentados, é importante a realização de um estudo mais específico em representação de conhecimento musical. Tal levantamento deve investigar os métodos existentes, tanto do ponto de vista da evolução histórica, como também de sua aplicação. Isso possibilitará o delineamento de padrões no sentido de se obter uma avaliação mais precisa de métodos, com a possibilidade de indicação de caminhos a serem traçados de acordo com as especificidades inerentes às variadas aplicações.

Além disso, para que seja feita a aplicação do tema em questão é importante que o objeto de estudo seja utilizado em um problema típico. Nesse sentido, a integração de paradigmas de programação - incorporando conceitos da lógica de primeira ordem, mode-

lagem orientada a objetos, programação funcional e imperativa - bem como ferramentas para processamento digital de áudio e síntese sonora - torna-se uma importante forma de representação de conhecimento musical. Isso tanto do ponto de vista de recursos de programação e modelagem da informação, como também com relação à praticidade no processo de desenvolvimento e interação humano-computador. Essa integração aplicada a um sistema para composição automática de trechos musicais constitui uma investigação importante nesse campo de conhecimento.

Deste modo, destaca-se a importância de um trabalho focado na aplicação da PLI no processo de representação de conhecimento musical, incluindo uma investigação voltada ao levantamento de critérios e práticas adotados em trabalhos existentes. Esta investigação deve envolver tanto recursos computacionais utilizados, bem como conceitos no campo da Musicologia a serem modelados. Tal atividade deve fornecer subsídios baseados em evidências, que possibilitem o delineamento de diretrizes no desenvolvimento de trabalhos relacionados (KEELE, 2007).

## 1.2 Caracterização do Trabalho

O presente trabalho tem seu escopo definido no campo da Representação de Conhecimento Musical. Seu desenvolvimento foi norteado pelo propósito de se obter um sistema de programação multiparadigma, que possibilite a aplicação da técnica de PLI. Tal abordagem deverá propiciar atividades como:

- Síntese sonora;
- Manipulação de áudio;
- Armazenamento, tratamento e reprodução de composições musicais;
- Controle de hardware e notação;
- Representação de Conhecimento Musical;
- Aprendizado musical;
- Composição musical automática.

## 1.3 Objetivo e Metodologia

O objetivo principal do trabalho é desenvolver uma plataforma de programação com a capacidade de servir como infraestrutura na aplicação do processo de Representação de Conhecimento Musical. O desenvolvimento foi realizado por meio de uma abordagem

integrando conceitos de programação multiparadigma com a técnica de Programação Lógica Indutiva.

A seguinte metodologia foi utilizada:

- Realização de levantamento sobre paradigmas de programação utilizados na RCM;
- Análise de aplicações da PLI em Computação Musical;
- Elaboração de uma abordagem de programação atendendo aos requisitos da produção de software musical;
- Desenvolvimento de aplicações para RCM baseadas em programação multiparadigma;
- Desenvolvimento de aplicações para aquisição de conhecimento musical utilizando a técnica de PLI;
- Desenvolvimento de um sistema para aplicação dos conceitos tratados;
- Avaliação do sistema desenvolvido.

## 1.4 Organização do Trabalho

O restante deste trabalho está organizado de acordo com a seguinte estrutura:

- O Capítulo 2 apresenta uma contextualização bibliográfica, sob a ótica de paradigmas de programação e PLI aplicada à Computação Musical, incluindo trabalhos relacionados.
- No Capítulo 3 é realizada uma revisão sistemática com relação a trabalhos que aplicam a técnica de PLI à Computação Musical.
- O Capítulo 4 consiste na descrição dos componentes de programação voltados ao desenvolvimento de aplicações, incluindo linguagens, bibliotecas e APIs.
- A primeira contribuição do trabalho é apresentada no Capítulo 5. Ela inclui conceitos e aplicações da programação multiparadigma na Representação de Conhecimento Musical.
- O Capítulo 6 contém a segunda contribuição - utilização da Programação Lógica Indutiva na aquisição de conhecimento, atividade que compõe o processo de RCM.
- O sistema musical Fraseado é descrito no Capítulo 7 (terceira contribuição).

- O Capítulo 8 inicia com uma breve revisão sobre a problemática envolvendo a avaliação de sistemas de Composição Algorítmica. Em seguida é proposto um método para avaliação desse tipo de sistema, o *Teste de Turing Expandido*. Por fim, é apresentada uma aplicação desse teste.
- O Capítulo 9 apresenta os resultados da aplicação do teste, tecendo considerações finais, o que inclui conclusões e possíveis trabalhos futuros.
- Por fim, o Capítulo 10 apresenta publicações realizadas no âmbito da presente pesquisa.

# 2 Contextualização

*Assume-se geralmente que a atividade inteligente*
*seja mediada por representações internas.*

EDUARDO RECK MIRANDA (1963 - )

Ao se tratar os elementos relacionados à representação musical por meio do computador, diferenciadas arquiteturas têm sido propostas. Componentes como notação simbólica, formatos de arquivo, recursos para codificação, técnicas de programação e abstração de dados têm sido aplicados no nível do processamento da informação. Adicionalmente, o trabalho envolvendo representação de conhecimento musical aborda aspectos referentes a paradigmas de programação e componentes ligados a lógicas formais, além de métodos de busca e tomada de decisão eficientes (ALVARO; MIRANDA; BARROS, 2005).

## 2.1 Representação Multiparadigma

Existe uma variedade de paradigmas de programação aplicados em Computação Musical de modo geral, cuja seleção é feita no sentido de valer-se dos benefícios específicos decorrentes de sua utilização (ANDERS, 2007). O paradigma procedural representa explicitamente os dados manipulados por um programa, sendo que esses dados podem sofrer mudanças ao longo do tempo por meio da invocação de módulos de execução como procedimentos e funções. A programação orientada a objetos organiza atividades computacionais em entidades chamadas classes. Os objetos encapsulam o estado e os métodos que possibilitam sua manipulação. Essa abstração busca simular a maneira pela qual se percebe o mundo externo. A herança possibilita o desenvolvimento incremental e reutilização de código.

Na programação funcional é realizada a avaliação de expressões, sendo que uma função recebe argumentos e devolve um resultado. A programação concorrente fornece suporte ao processamento de diferentes tarefas, sendo que a ordem de execução de processos é indeterminada a priori. O processamento pode ocorrer de forma paralela, de acordo com as características dos recursos disponíveis. O paradigma da programação em lógica possibilita que sejam definidas relações simbólicas entre valores por meio da utilização de fatos e regras - inclusive valores desconhecidos. O usuário pode solicitar ao sistema que encontre uma ou mais soluções, as quais são apresentadas de acordo com as relações fornecidas ao programa (GUIMARAES, 2015).

Adicionalmente, utiliza-se com frequência na programação musical a combinação de paradigmas. Isso possibilita ao usuário maior flexibilidade ao lidar com a complexidade existente. Linguagens como Common Lisp e Oz fornecem a expressividade da programação

multiparadigma, além de ambientes de composição como OpenMusic e MOzart (ANDERS; ANAGNOSTOPOULOU; ALCORN, 2004).

A seguir, apresentam-se trabalhos relacionados envolvendo aplicações realizadas em CM. Tais trabalhos incorporam características tanto do ponto de vista dos paradigmas de programação, como da PLI, além de ser abordada a questão da representação de conhecimento no campo musical.

## 2.2 Aplicações em Computação Musical

**Strasheela** (ANDERS, 2007) é um sistema de composição multiparadigma baseado na definição de problemas de satisfação de restrições musicais. Enfatiza a programabilidade de três componentes fundamentais: representação musical, mecanismo de aplicação de regras e processo de busca. Qualquer informação pertencente à partitura é acessível por meio de objetos específicos e pode ser usada para se obter derivações. O usuário pode otimizar a busca pela satisfação de uma restrição particular através da programação de estratégias distribuídas, utilizando variáveis dinâmicas.

**EV Meta-Model** (ALVARO; MIRANDA; BARROS, 2005) é um sistema para representação de conhecimento musical voltado à composição assistida por computador. Foi proposto como uma ferramenta genérica para representação de diferentes níveis de abstração musical. É um sistema de representação dinâmico, transmitindo essa característica ao componente musical representado.

Pachet *et al.* (PACHET; RAMALHO; CARRIVE, 1996) basearam-se na Programação Orientada a Objetos, utilizando a linguagem Smalltalk, para criar um framework no contexto de música tonal. Desenvolveram o sistema **MusES**, que contém uma representação dos conceitos básicos de harmonia tonal, como notas, intervalos, acordes, escalas e melodias.

Serapião desenvolveu o sistema **Ritornello** (SERAPIAO, 2004), um Framework para Representação do Conhecimento Musical. Lima (LIMA, 1998) desenvolveu um sistema de composição musical baseado na modelagem de aprendizado por estilo musical. Este sistema é capaz de criar composições inteiramente inéditas e totalmente fiéis ao estilo musical de um determinado compositor. Utilizando a linguagem funcional Clean, elaborou um sistema que analisa uma partitura musical léxica e sintaticamente, extraindo o estilo musical implícito; um outro módulo realiza a composição, baseando-se nas informações extraídas. A partitura gerada é convertida em um arquivo MIDI.

**GeNotator** (THYWISSEN, 1999) é um ambiente que explora a aplicação de técnicas evolutivas na composição musical. O sistema concentra-se na definição de mecanismos inspirados em processos da teoria evolutiva como algoritmos genéticos utilizando a técnica

de busca heurística. **VexPat** (SANTANA et al., 2003) é um sistema de auxílio à tarefa de extração de padrões de seqüências musicais. **SOM-G** (SILVA, 2009) é uma linguagem de processamento sonoro para síntese granular que possui uma estrutura sintática concisa e eficiente. É voltado tanto à orquestração de instrumentos, com alto grau de controle sobre parâmetros granulares, bem como à interpretação/renderização de partituras polifônicas que utilizem esses instrumentos.

Alguns outros trabalhos são voltados ao estabelecimento de elementos conceituais. Wulfhorst (WULFHORST et al., 2003) propôs uma arquitetura aberta para um sistema multi-agente musical. Essa arquitetura possibilita a simulação de comportamento de um grupo vocal/instrumental por meio da interação de uma comunidade de agentes baseada em eventos musicas. Bittencourt (BITTENCOURT, 1998) relaciona diversas técnicas utilizadas com seus fundamentos filosóficos e matemáticos, partindo desde a descoberta por Pitágoras da relação de intervalos musicais associados a experiências subjetivas, passando pela lógica de primeira ordem e chegando a métodos atuais de representação de conhecimento. Teixeira (TEIXEIRA, 1997) utilizou a noção de ponto de vista em harmonia para representar conhecimento e eventos musicais. Mello, em sua dissertação de mestrado (MELLO, 2003), utilizou sistemas semióticos para propor uma investigação epistemológica envolvendo pesquisas atuais no campo da cognição musical. Miranda, em sua tese de doutorado (MIRANDA, 1995) utilizou gramáticas formais voltadas ao aprendizado computacional e síntese acústica.

Dentre os trabalhos apresentados, destacam-se características fundamentais com relação à representação de conhecimento musical, paradigmas de programação e composição algorítmica. Pode-se ainda observar que a utilização da técnica de PLI não ocorre em nenhum caso. Tal constatação corrobora com o levantamento feito na revisão da literatura, onde, a partir da aplicação dos mecanismos de busca sistemática em um total de 7075 trabalhos, apenas 3 utilizam a PLI em composição algorítmica, sendo que nenhum deles faz uso do recurso de programação multiparadigma (GONCALVES; HOMEM, 2015).

## 2.3 Programação Lógica Indutiva Aplicada à CM

A utilização da Programação Lógica Indutiva em sistemas musicais ocorre desde a especificação do formalismo da Lógica Indutiva de Primeira Ordem nos anos 90 - FOIL (QUINLAN, 1990). Chong (CHONG; DING, 2014) aplica um sistema baseado em regras para incorporar conhecimento no processo de interpretação da informação musical. As regras são formuladas a partir de considerações como deslocamento da fundamental, movimentação do baixo e fatores ligados à condução da voz principal. A abordagem de Anglade (ANGLADE et al., 2010) é voltada para a identificação de estilos e gêneros musicas. Progressões harmônicas são tratadas utilizando a descrição sequencial de acordes.

Realiza-se a combinação de descritores de áudio de baixo nível com uma ferramenta de aprendizado de máquina, o que possibilita a classificação baseada em cadências. Integrando-se ao mecanismo indutivo, elementos estruturais foram utilizados de modo a compor cada modelo de representação de conhecimento.

Dixon (DIXON; MAUCH; ANGLADE, 2011) apresenta duas abordagens voltadas à modelagem harmônica: probabilística e lógica. Com isso pode-se obter tanto a categoria de conhecimento musical, como também o raciocínio utilizado por um músico ao realizar tarefa análoga. Acordes são transcritos a partir de gravações de áudio. O sistema realiza a modelagem de contexto musical em alto nível. Pompe (POMPE; KONONENKO; MAKSE, 1996) desenvolveu uma ferramenta com a finalidade de realizar o processo de descoberta de conhecimento em uma base de dados com dezenas de milhares de instâncias. O sistema foi implementado de modo que a indução de uma hipótese é tratada como um problema de otimização.

Morales (MORALES; MORALES, 1995) utiliza a lógica de primeira ordem para expressar regras de contraponto, possibilitando que relações entre estados musicas sejam descritas de uma maneira compacta e compreensível. O modelo de Maestre (MAESTRE et al., 2009) realiza a síntese expressiva por meio da aquisição de conhecimento obtido em gravações de áudio. Numao (NUMAO; TAKAGI; NAKAMURA, 2002) lida com estruturas musicais capazes de causar sentimentos humanos característicos. Foi construído um sistema para arranjo e composição de modo automático, que gera peças musicais com a capacidade de produzir sentimentos específicos em alguns indivíduos.

A abordagem de Perez (PEREZ; RAMIREZ; KERSTEN, 2008) é voltada ao aspecto instrumental de obras executadas no violino. O sistema armazena padrões expressivos que são adquiridos de modo automático. Ramirez (RAMIREZ; HAZAN, 2005) realiza uma abordagem com o intuito de investigar a performance musical em melodias de *Jazz*. Utiliza técnicas de aprendizado de máquina para extrair movimentos regulares e padrões de desempenho.

Os recursos computacionais utilizados envolvem mecanismos para indução lógica, ferramentas de pré-processamento, linguagens de programação, conceituação teórica para percepção e cognição melódica. O mecanismo de inferência lógica TILDE (*Top-down Induction of Logical Decision Trees*) fundamenta-se na lógica de primeira ordem e aplica a indução por meio de árvores de decisão. Sua funcionalidade é considerada uma extensão ao algoritmo C4.5 (SALZBERG, 1994) que ao invés de testar valores de atributos em nós de uma árvore, testa predicados lógicos (MAESTRE et al., 2009).

No trabalho de Anglade (ANGLADE et al., 2010) regras descrevendo padrões harmônicos de um dado gênero podem coexistir com regras de outros gêneros em uma mesma árvore. Dixon (DIXON; MAUCH; ANGLADE, 2011) realiza a descrição de acordes em termos de sua nota fundamental, grau na escala e categorias de intervalos. Maes-

tre (MAESTRE et al., 2009) utiliza árvores proposicionais de decisão com técnicas de poda. Perez (PEREZ; RAMIREZ; KERSTEN, 2008) realiza a mineração de dados estruturados aplicando o algoritmo *top-down* em árvores de decisão.

O sistema de programação lógica indutiva Aleph (*A Learning Engine for Proposing Hypotheses*) foi desenvolvido com o propósito de explorar ideias expressas por cláusulas de Horn com alta capacidade representativa. Escrito em Prolog, possibilita a descrição de expressões complexas, incorporando simultaneamente o conhecimento adquirido, com a capacidade de escolha da ordem de geração das regras, alteração nas funções de avaliação e busca. No trabalho de Dixon (DIXON; MAUCH; ANGLADE, 2011), esses conceitos são aplicados para se encontrar um conjunto mínimo de regras que seja capaz de descrever a totalidade das amostras positivas e um número mínimo de amostras negativas. Ramirez (RAMIREZ; HAZAN, 2005) utiliza um algoritmo padrão para manipulação em conjuntos com a finalidade de construir hipóteses individuais.

A teoria Narmour busca a identificação de padrões para percepção e cognição de melodias, tendo o campo da análise musical como principal foco (SAKHARE; HANCHATE, 2014). Essa teoria contribui tanto na compreensão do significado melódico como do conhecimento envolvido em sua criação. Maestre (MAESTRE et al., 2009) utiliza o modelo de implicação/realização dessa teoria, onde cada nota pertence a uma estrutura Narmour. Perez (PEREZ; RAMIREZ; KERSTEN, 2008) utiliza o contexto Narmour, onde são definidos grupos específicos com os quais uma nota mantenha uma relação de pertinência. Ramirez (RAMIREZ; HAZAN, 2005) faz uso extensivo deste conceito, incorporando informações sobre notas prévias e sucessoras, além de propriedades intrínsecas a intervalos.

Considerando os elementos apresentados, no que tange à utilização da programação multiparadigma como forma de representação de conhecimento - incluindo a técnica de Programação Lógica Indutiva - pode-se constatar a existência de variados trabalhos relacionados ao tema em questão, sendo o desenvolvimento desses trabalhos aplicado diretamente ao campo da Computação Musical.

# 3 Revisão Sistemática da Literatura

*Dados inteiros positivos $\alpha$ e $\lambda$, existe uma música cuja complexidade seja $(20 + \lambda + \alpha)\sqrt{n/(30 + 2\lambda)} + O(1)$.*

DONALD E. KNUTH (1938 - )

Com o intuito de realizar um levantamento sobre trabalhos que abordem a representação de conhecimento musical por meio da Programação Lógica Indutiva, foi elaborada uma revisão sistemática de literatura. A metodologia de pesquisa foi aplicada de acordo com critérios e recomendações apresentados em (KITCHENHAM et al., 2009). Foram levantadas duas questões para servir de diretriz ao trabalho:

**Questão 1:** Qual arquitetura é proposta para representação do conhecimento musical?

**Questão 2:** Como a abordagem aplica os recursos computacionais ligados à PLI?

Deste modo, buscou-se encontrar abordagens que possibilitem o estudo da técnica em questão. A seguir, são descritas as etapas realizadas para a seleção dos trabalhos.

## 3.1 Estratégia de Pesquisa

A pesquisa foi realizada a partir de publicações de trabalhos acadêmicos, artigos em periódicos científicos e participações em conferências. As seguintes bases de consulta foram utilizadas:

- Academic Search Premier
- ACM Digital Library
- Cambridge Journals Online
- Computer and Information Systems
- Oxford Journals
- ScienceDirect
- SpringerLink
- Web of Science
- Computers and Applied Sciences
- SCOPUS
- IEEE Xplore
- Google Scholar

Com relação à estratégia de pesquisa, o trabalho foi baseada nas línguas Inglesa e Portuguesa. Foram elencados os principais termos de acordo com as questões levantadas. As atividades foram realizadas nos mesmos moldes das abordagens apresentadas em (TOMAS et al., 2013), (KHURUM; GORSCHEK, 2009), (ABDALLA; DAMASCENO; NAKAGAWA, 2015) e (CHEN; BABAR; ALI, 2009). Apenas trabalhos realizados a partir de 1990 foram considerados, devido ao fato de que nesse ano foi introduzido o formalismo da Lógica Indutiva de Primeira Ordem (FOIL) (QUINLAN, 1990).

## 3.2 *Strings* de busca

Duas expressões iniciais foram tomadas como base para a busca, enfatizando o elemento musical e a programação lógica indutiva. Ambas foram aplicadas em conjunção a componentes voltados à representação de conhecimento. Em seguida foram incluídas expressões literais equivalentes e adicionadas derivações de pronúncia. Os elementos fundamentais para a seleção foram unidos pelo operador "AND"e as variações alternativas foram conectadas pelo operador "OR". Deste modo, considerando as duas línguas em questão, as seguintes *strings* de busca foram elaboradas:

**1:** `("knowledge representation" AND musi*) OR ("musi* representation" AND knowledge) OR ("musi* knowledge" AND representation) OR (knowledge AND representation AND "computer musi*")`

**2:** `("representacao de conhecimento" AND musi*) OR ("representacao musi*" AND conhecimento) OR ("conhecimento musi*" AND representacao) OR (conhecimento AND representacao AND "computacao musical")`

**3:** `"inductive logic programming" AND knowledge AND representation`

**4:** `"programação lógica indutiva" AND conhecimento AND representação`

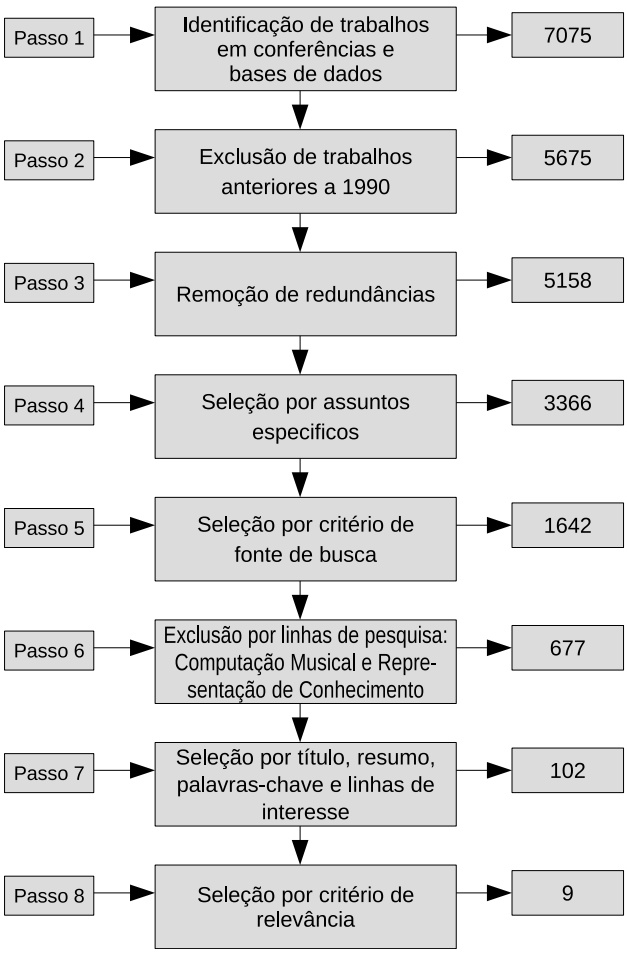
## 3.3 Seleção de Trabalhos

Para a seleção dos trabalhos, foram aplicados filtros com o intuito de destacar os estudos com maior importância de acordo com a finalidade proposta. Para um adequado refinamento dos tópicos de relevância, a busca foi direcionada de modo mais específico aos seguintes assuntos:

> Artificial intelligence, Expert systems, Automatic composer, Algorithmic composition, Music representation, Logic programming, Answer set programming, Prolog, Logical language, Automatic composition, Constraint programming, Declarative languages, Declarative system, Knowledge-based system.

A Figura 1 apresenta os critérios utilizados na elaboração dos filtros, bem como a quantidade de trabalhos selecionados em cada etapa.

Figura 1 – Etapas na estratégia de busca




A partir de uma quantidade inicial de 7075 trabalhos, chegou-se ao número de 9 trabalhos a serem estudados. A desigualdade entre esses valores é explicada em (KITCHENHAM et al., 2009). A Tabela 1 apresenta as abordagens, cada uma recebeu uma identificação numérica e um nome formado pelas duas primeiras palavras do título.

## 3.4 Análise dos Dados Obtidos

Dando continuidade aos processos de pesquisa e filtragem de trabalhos, as 9 abordagens selecionadas foram analisadas do ponto de vista dos componentes relevantes à revisão. Foram identificados elementos fundamentais bem como aspectos característicos de acordo com as questões básicas. A seguir apresenta-se uma discussão sobre as abordagens feitas com relação a tais questões. Considerando o inter-relacionamento entre os componentes levantados, alguns aspectos serão analisado juntamente com sua descrição.

Tabela 1 – Relação de trabalhos revisados

| ID | ABORDAGEM | TÍTULO | REFERÊNCIA |
|---|---|---|---|
| **1** | SymbolicRepresentation | Symbolic Representation of Chords for Rule-Based Evaluation of Tonal Progressions | Chong and Ding, 2014 |
| **2** | ImprovingMusic | Improving Music Genre Classification Using Automatically Induced Harmony Rules | Anglade et al., 2010 |
| **3** | ProbabilisticLogic | Probabilistic and Logic-Based Modelling of Harmony | Dixon et al., 2011 |
| **4** | ApplicationILP | Application of ILP in a musical database: learning to compose the two-voice counterpoint | Pompe et al., 1996 |
| **5** | LearningMusical | Learning Musical Rules | Morales and Morales, 1995 |
| **6** | ExpressiveConcatenative | Expressive Concatenative Synthesis by Reusing Samples from Real Performances Recordings | Maestre et al., 2009 |
| **7** | ConstructiveAdaptive | Constructive Adaptive User Interfaces - Composing Music Based on Human Feelings | Numao et al., 2002 |
| **8** | ModelingMoods | Modeling Moods in Violin Performances | Perez et al., 2008 |
| **9** | ModelingExpressive | Modeling Expressive Music Performance in Jazz | Ramirez and Hazan, 2005 |

### 3.4.1 Questão 1 - Qual arquitetura é proposta para representação do conhecimento musical?

Ao se tratar os elementos relacionados à representação musical por meio do computador, componentes como notação simbólica, formatos de arquivo, recursos para codificação, técnicas de programação e abstração de dados têm sido aplicados no nível do processamento da informação. Adicionalmente, o trabalho envolvendo representação de conhecimento musical aborda aspectos referentes a paradigmas de programação e componentes ligados a lógicas formais, além de métodos de busca e tomada de decisão eficientes.

SymbolicRepresentation (CHONG; DING, 2014) utiliza um sistema baseado em regras para incorporar o conhecimento aplicado no processo de interpretação da informação musical. As regras são formuladas a partir de considerações como deslocamento da fundamental, movimentação do baixo e fatores ligados à condução da voz principal. Visando modelar o movimento do campo harmônico, a estrutura gramatical é representada por regras da teoria musical ocidental vigente. Em contraste aos algoritmos tradicionais do tipo “se-então-senão” é adotada uma estrutura de árvore de decisão, a qual possibilita

resultados mais eficientes. É feita a implementação de duas categorias principais de regras.

A primeira define restrições com o objetivo de direcionar o foco do processo de inferência para conjuntos específicos de progressões harmônicas. Assim, se houver a necessidade de fazer distinção entre acordes cromáticos e diatônicos, pode-se criar uma regra em que todos os acordes de uma progressão devam pertencer à mesma clave. A segunda categoria realiza considerações harmônicas sobre a condução da voz principal. Neste caso, um conjunto de regras pode ser implementado para garantir o tratamento apropriado para resoluções de acordes cadenciais.

A abordagem de ImprovingMusic (ANGLADE et al., 2010) é voltada para a identificação de estilos e gêneros musicas. Progressões harmônicas são tratadas utilizando a descrição sequencial de acordes. Realiza-se a combinação de descritores de áudio de baixo nível com uma ferramenta de aprendizado de máquina, o que possibilita a classificação baseada em cadências. As sequências servem como base para o treinamento e são obtidas de modo indutivo a partir de transcrições automáticas. Desse modo, pode-se obter um desempenho satisfatório na classificação de gêneros musicais. Um conjunto de atributos é utilizado na definição dos parâmetros representativos. Esse conjunto envolve elementos como tempo e dinâmica, além do espectro sonoro e descritores de altura. Trechos musicais são representados por listas que incorporam os acordes neles contidos. Gêneros são representados por conjuntos de trechos.

Para cada gênero, busca-se encontrar regras harmônicas que descrevam sequências características em acordes presentes nas músicas armazenadas. Estas regras harmônicas constituem uma Gramática Livre de Contexto. Apenas as características harmônicas pertencentes aos gêneros conhecidos são consideradas neste modelo. Dessa forma, acordes com finalidades puramente ornamentais e sequências não características aos estilos mantidos são desconsiderados. Nesse formalismo, os acordes são representados por letras do alfabeto e rotulados usando a notação de cifras. As propriedades dos acordes são descritas por meio de predicados lógicos que retornam valores verdadeiros ou falsos.

Integrando-se ao mecanismo indutivo, alguns elementos estruturais foram utilizados nas abordagens. Tais componentes foram aplicados de modo a compor o modelo de representação de conhecimento específico. A Tabela 2 apresenta esses elementos.

ProbabilisticLogic (DIXON; MAUCH; ANGLADE, 2011) apresenta duas abordagens voltadas à modelagem harmônica: probabilística e lógica. Com isso pode-se obter tanto a categoria de conhecimento musical, como também o raciocínio utilizado por um músico ao realizar tarefa análoga. Acordes são transcritos a partir de gravações de áudio. O sistema realiza a modelagem de contexto musical em alto nível. Nesse modelo, elementos como acordes, claves, posicionamento métrico, nota de baixo, características cromáticas e estruturas de repetição são integrados em um sistema Bayesiano.

Tabela 2 – Elementos estruturais para representação de conhecimento

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Utiliza o padrão MIDI | | | ■ | | | ■ | ■ | | ■ |
| Realiza coleta analógica de dados | | | | | | | | ■ | ■ |
| Realiza pré-processamento | | ■ | | | | | | | ■ |
| Realiza síntese aditiva | | | | | | ■ | | | |
| Estrutura básica: melodia | | | | | | ■ | | | ■ |
| Estrutura básica: contraponto | | | | ■ | ■ | | | | |
| Estrutura básica: harmonia | ■ | ■ | ■ | | | | | | |
| Técnica integrada: Algoritmos genéticos | | | | | | | ■ | | |
| Técnica integrada: Mineração de dados | | | | | | | | | ■ |
| Técnica integrada: Processos estocásticos | | | ■ | | | | | | |
| Técnica integrada: Restrições lógicas | ■ | | | | ■ | | | | |
| Representação de textura polifônica | | | | | | ■ | | | |

Esse sistema produz o conteúdo de uma pauta-mestre que contém uma sequência simbólica representando acordes. Cada símbolo inclui variações de clave e modulações no transcorrer do tempo. Além disso, o sistema concentra-se na descrição lógica de sequências harmônicas com o intuito de caracterizar estilos musicais e gêneros particulares. As relações harmônicas são representadas por meio do formalismo da lógica de primeira ordem. A abordagem baseia-se em representações por árvores de decisão, utilizadas para classificar amostras não visitadas ou fornecer sugestões com respeito a características variadas da base de dados.

ApplicationILP (POMPE; KONONENKO; MAKSE, 1996) desenvolveu uma ferramenta com a finalidade de realizar o processo de descoberta de conhecimento em uma base de dados com dezenas de milhares de instâncias. O sistema foi implementado de modo que a indução de uma hipótese é tratada como um problema de otimização. Tal abordagem é direcionada ao tratamento da problemática envolvendo a composição do contraponto a duas vozes. A aplicação de técnicas de aprendizado de máquina mostrou-se adequada a esse subdomínio da música tonal que possui características imprecisas, envolvendo aspectos não convencionais e exigentes do ponto de vista da acurácia. Dois predicados alvo foram utilizados para representar o conhecimento prévio a partir do universo de constantes musicais. Adicionalmente, foi realizado o treinamento tanto das instâncias positivas como das negativas, com o propósito de aplicar a indução por meio de regras.

LearningMusical (MORALES; MORALES, 1995) utiliza a lógica de primeira ordem para expressar regras de contraponto, possibilitando que relações entre estados musicas sejam descritas de uma maneira compacta e compreensível. O sistema executa a análise a partir de um conjunto finito de regras bem estabelecidas. O aprendizado de regras é implementado por meio do sistema PAL (MORALES, 1994). Este ambiente é

alimentado por um subconjunto de cláusulas de Horn. Essas cláusulas contém conhecimento armazenado a partir de duas origens: amostras e informações prévias, sendo em ambos os casos expressas de modo declarativo. PAL incorpora regras de contraponto e as utiliza na análise musical e geração de notas de acordo com as convenções do *cantus firmus*.

Desse modo, uma sequência de notas individuais é utilizada como base para a aplicação das regras de contraponto. Com isso, novas notas são geradas de acordo com as restrições harmônicas estabelecidas. Por meio do sistema PAL, as restrições podem ser utilizadas como diretrizes na busca por hipóteses, tanto no contexto imediato, como também de modo mais extensivo.

O modelo de ExpressiveConcatenative (MAESTRE et al., 2009) realiza a síntese expressiva por meio da aquisição de conhecimento obtido em gravações de áudio. A partir destas gravações, amostras de notas pontuais são criteriosamente concatenadas, formando melodias com um corpo direcionado à sua execução. O sistema busca produzir uma sequência de áudio a partir da análise de partituras. Para isso, utiliza um modelo performático indutivo, o qual é integrado a uma base de dados gerada a partir de execuções reais. A abordagem aplica a ideia de utilizar o mesmo banco de dados nas principais etapas do processo. Desse modo, tanto na introdução do modelo de performance como na concatenação sonora das amostras, a mesma base de conhecimento é manipulada. Assim, fica estabelecida uma conexão durante o processo de síntese entre o som do instrumento em execução e as características modeladas a partir desta execução.

ConstructiveAdaptive (NUMAO; TAKAGI; NAKAMURA, 2002) lida com estruturas musicais capazes de causar sentimentos humanos característicos. Foi construído um sistema para arranjo e composição de modo automático, que gera peças musicais com a capacidade de produzir sentimentos específicos em uma pessoa. Inicialmente, o sistema realiza a coleta de sentimentos a partir de obras musicais. Baseando-se em tal informação, são extraídas estruturas musicais que serão utilizadas na produção dos sentimentos. Com relação à geração musical, o sistema é direcionado para duas possibilidades: elaborar arranjos em músicas pré-existentes e criar novas composições de maneira automática. Em ambos os casos, o resultado produzido tem a capacidade de produzir em um indivíduo sentimentos previamente determinados.

A abordagem de ModelingMoods (PEREZ; RAMIREZ; KERSTEN, 2008) é voltada ao aspecto instrumental de obras executadas no violino. O sistema armazena padrões expressivos que são adquiridos de modo automático. Realiza a modelagem de aspectos interpretativos como variações respectivas ao tempo, dinâmica e altura. Além disso, o modelo recebe informações relativas ao controle gestual, com ênfase no que diz respeito à direção do arco e posicionamento dos dedos. O sistema baseia-se em quatro tipos de humores performáticos: Tristeza, Felicidade, Medo e Raiva. As características expressivas analisadas são duas: o tempo e um conjunto de descritores em nível de nota. Estes

descritores compreendem qualidades ligadas ao ataque, duração, energia, direção do arco e corda sendo tangida. Tanto as sequências de notas como as variações de expressividade na execução são definidas de um modo estruturado por meio de predicados da lógica de primeira ordem.

Há dois grupos de predicados para descrever o contexto musical de cada nota e as variações expressivas. Os predicados para contexto musical definem informações sobre propriedades intrínsecas das notas, incluindo duração e posição métrica. Também armazenam suas notas predecessora e subsequente, além da extensão e direção dos intervalos. Os predicados de variações expressivas definem o fator de extensão de uma nota no que diz respeito à sua duração na pauta. Adicionalmente, possuem a capacidade de incorporar o ponto de mudança na direção do arco, a corda em que a nota é tocada e o nível de energia na execução.

Em ModelingExpressive (RAMIREZ; HAZAN, 2005) a abordagem tem o intuito de investigar a performance musical em melodias de *Jazz*. Utiliza técnicas de aprendizado de máquina para extrair movimentos regulares e padrões de desempenho. Para tanto, são aplicados métodos de classificação e regressão em bases de dados obtidas a partir de execuções *jazzísticas* reais. Considerando os métodos de regressão, busca-se a precisão preditiva para gerar boas soluções do ponto de vista da acurácia. No caso da classificação, sua utilização tem o objetivo de possibilitar uma descrição compreensível com relação às predições do sistema. Os modelos baseados em regressão são introduzidos com o objetivo de implementar transformações que possibilitem performances características. Os modelos de classificação são utilizados para possibilitar a compreensão dos princípios e critérios no que diz respeito à execução dos trechos musicais.

### 3.4.2 Questão 2 - Como a abordagem aplica os recursos computacionais ligados à PLI?

Os recursos computacionais utilizados envolvem mecanismos para indução lógica, ferramentas de pré-processamento, linguagens de programação, conceituação teórica para percepção e cognição melódica. Inicialmente, o mecanismo de inferência lógica TILDE (*Top-down Induction of Logical Decision Trees*) (BLOCKEEL, 1999) fundamenta-se na lógica de primeira ordem e aplica a indução por meio de árvores de decisão. Sua funcionalidade é considerada uma extensão ao algoritmo C4.5 (SALZBERG, 1994), que em vez de testar valores de atributos em nós de uma árvore, testa predicados lógicos (MAESTRE et al., 2009). Nesse modelo, cada árvore trata um problema de classificação específico. Em ImprovingMusic, regras descrevendo padrões harmônicos de um dado gênero podem coexistir com regras de outros gêneros em uma mesma árvore. ProbabilisticLogic realiza a descrição de acordes em termos de sua nota fundamental, grau na escala e categorias de intervalos entre notas sucessivas.

A utilização de TILDE em ExpressiveConcatenative possibilita vantagens no sentido de se obter árvores proposicionais de decisão com eficiência e técnicas de poda, que consistem na eliminação de ramos insignificantes da árvore, o que leva a algoritmos mais eficientes. Além disso, é aplicada a lógica de primeira ordem com a capacidade de incluir conhecimento prévio no processo de aprendizagem. ModelingMoods realiza a mineração de dados estruturados por meio da abordagem indutiva, aplicando o algoritmo *top-down* em árvores de decisão. Os recursos e ferramentas computacionais utilizados em cada abordagem são apresentados na Tabela 3.

Tabela 3 – Recursos computacionais ligados à PLI

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Aleph | | ■ | ■ | | | | | | ■ |
| C4.5 | | | | | | | | | ■ |
| SFOIL | | | | ■ | | | | | |
| Narmour | | | | | | ■ | | ■ | ■ |
| SMSTools | | | | | | | | | ■ |
| TILDE | | ■ | ■ | | | ■ | | ■ | |
| Pal | | | | | ■ | | | | |
| JBoss Drools | ■ | | | | | | | | |
| Indução a partir de Prolog | | ■ | ■ | | ■ | | ■ | ■ | ■ |
| Utiliza WEKA | | | | | | | | | ■ |
| Utiliza base de treinamento adicional | | ■ | ■ | | | ■ | ■ | ■ | ■ |

O sistema de programação lógica indutiva Aleph (*A Learning Engine for Proposing Hypotheses*) (SRINIVASAN, 2007) foi desenvolvido com o propósito de explorar ideias expressas por cláusulas de Horn com alta capacidade representativa. Escrito em Prolog, possibilita a descrição de expressões complexas, incorporando simultaneamente o conhecimento adquirido, com a capacidade de escolha da ordem de geração das regras, alteração nas funções de avaliação e busca (CONCEIÇÃO, 2008). Em ProbabilisticLogic, esses conceitos são aplicados para se encontrar um conjunto mínimo de regras que seja capaz de descrever a totalidade das amostras positivas e um número mínimo de amostras negativas. Essas regras cobrem todas as sequências de 4 acordes, sendo que cada sequência é tratada uma única vez. ModelingExpressive utiliza seu algoritmo padrão para manipulação em conjuntos com a finalidade de construir hipóteses individuais. Deste modo, Aleph utiliza a primeira amostra positiva não visitada como semente para a busca e cobre o espaço de cláusulas de acordo com uma restrição de comprimento previamente definida.

Narmour é uma teoria para percepção e cognição de melodias aplicada à análise musical. Auxilia na compreensão tanto do significado melódico como do conhecimento envolvido em sua criação (SAKHARE; HANCHATE, 2014). ExpressiveConcatenative utiliza o modelo de implicação/realização dessa teoria, onde cada nota pertence a uma

estrutura Narmour. ModelingMoods utiliza o contexto Narmour, onde são definidos grupos específicos com os quais uma nota mantenha uma relação de pertinência. ModelingExpressive faz uso extensivo deste conceito, incorporando informações sobre notas prévias e sucessoras, além de propriedades intrínsecas a intervalos.

ModelingExpressive utiliza SMSTools (SALAMON, 2013) para realizar o preprocessamento, criando uma descrição em alto nível de gravações em áudio. Também gera uma sonorização expressiva de acordo com transformações obtidas por aprendizado de máquina. Essa abordagem utiliza o algoritmo C4.5 (SALZBERG, 1994) para obter um conjunto de regras de classificação diretamente da árvore de decisão gerada. O mecanismo de regras aplicado em SymbolicRepresentation baseia-se em JBoss Drools (LEY; JACOBS, 2011) que possibilita a operação de acordo com linhas de desenvolvimento pré-definidas. Esse mecanismo utiliza o algoritmo Rete (LIU; GU; XUE, 2010) para realizar o emparelhamento de padrões. LearningMusical fundamenta-se no sistema PAL (MORALES, 1994) para o aprendizado lógico. Esse sistema é utilizado no processo de análise de contraponto para executar o aprendizado de regras de transição. Tais regras são expressas na forma de cláusulas de Horn a partir de pares de estados musicais (conjuntos de notas), adicionalmente representando o conhecimento musical de propósito geral.

O sistema FOIL (QUINLAN, 1990) é um passo inicial no sentido de se obter ferramentas eficientes para lidar com problemas do mundo real. É considerado um marco no campo da PLI (POMPE; KONONENKO; MAKSE, 1996; PATEL; OZA, 2014; KOSTER; SABATER-MIR; SCHORLEMMER, 2011). ApplicationILP utiliza o algoritmo SFOIL (*Stochastic* FOIL), esse algoritmo combina a eficiência de FOIL com a implementação de uma estratégia de busca estocástica. Essa abordagem utiliza uma representação extensional de experiências, baseando-se em predicados alvo. Tal representação possibilitou o processo de descoberta de conhecimento em uma base de dados superior a 10.000 instâncias.

### 3.4.3 Análise

Fatores adicionais foram levantados a partir dos direcionamentos aplicados pelas abordagens. Com relação aos componentes de Musicologia tratados, destacam-se a caracterização de gênero e estilo, aplicados à performance expressiva, composição automática e instrumentação. No que diz respeito aos experimentos e testes, realizaram-se considerações sobre complexidade computacional, *benchmarks* e avaliação de acurácia, entre outros. A Tabela 4 apresenta os elementos abordados com relação à Musicologia.

Com relação ao sistema de representação de regras lógicas, considera-se a utilização da linguagem Prolog, cuja importância decorre de seu aspecto fortemente declarativo. O mecanismo de unificação de Prolog - cujo processo de inferência possui características imperativas (GUIMARAES, 2015) - é acionado em segundo plano, devido aos algoritmos de indução lógica aplicados sobre ele. Nesse sentido, destacam-se as ferramentas Aleph e

Tabela 4 – Elementos da Musicologia

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Caracterização de estilo | | | ■ | | | | | | |
| Caracterização de gênero | | ■ | ■ | | | | | | |
| Análise em nível analógico | | | | | | ■ | | | |
| Tratamento melódico/harmônico | | ■ | ■ | | | | | | ■ |
| Funções tonais | ■ | | | | | ■ | ■ | | |
| Graus diatônicos | ■ | ■ | | | | | | | |
| Representação de partitura | | | | | | | ■ | ■ | |
| Sistema Modal | | | | | ■ | | | | |
| Processamento interativo | ■ | | ■ | | | | | | |
| Voltado à performance expressiva | | | | | | ■ | ■ | ■ | ■ |
| Voltado à composição automática | | | | ■ | ■ | | ■ | | |
| Voltado à instrumentação | | | | | | ■ | | ■ | |
| Propósito educacional | ■ | | | | | | | | |

TILDE. Ambas realizam a indução lógica sobre os predicados de Prolog. A teoria Narmour também destaca-se como um importante recurso descritivo. Por meio dessa modelagem, melodias podem ser expressas por listas de estruturas sobrepostas (sequências de Narmour). O conceito de contexto Narmour possibilita a representação conveniente de agrupamentos, onde pode-se expressar tanto a noção de notas sucessoras como a de pertinência em grupos específicos.

No que concerne à realização de experimentos e testes, a Tabela 5 caracteriza as atividades realizadas e apresenta a utilização das mesmas.

Tabela 5 – Experimentos e Testes

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Considerações sobre complexidade computacional | | ■ | | | | ■ | | | |
| Estudo por processos empíricos | | ■ | | | | | | | |
| Avaliação por músicos e não músicos | | | | ■ | | | | | |
| Realiza benchmark | | | ■ | ■ | | ■ | ■ | | ■ |
| Realiza beta-teste | ■ | ■ | | | | | | ■ | |
| Realiza testes completos | | ■ | | ■ | | | | ■ | |
| Avaliação de acurácia | | ■ | ■ | | | ■ | | ■ | ■ |

A presente revisão sistemática possibilitou a avaliação de trabalhos que realizam a representação de conhecimento musical por meio da Programação Lógica Indutiva. A metodologia de pesquisa foi descrita na segunda seção e os resultados foram analisados na terceira seção de acordo com as questões de pesquisa. Uma série de características abordadas pelos trabalhos foi identificada, de modo que a revisão fornece uma visão

abrangendo componentes fundamentais na representação de conhecimento musical. Com relação à arquitetura para representação de conhecimento, observou-se a importância fundamental de 3 elementos: linguagem de programação declarativa, mecanismo para implementação da indução lógica e técnica integrada para aquisição de conhecimento.

Nesse sentido, a linguagem Prolog destaca-se pela característica expressiva, a qual, aliada a um mecanismo de indução como Aleph ou TILDE, possibilita elevada capacidade representativa, além de simplicidade ao se apresentar tal conhecimento. Os métodos envolvendo algoritmos genéticos, restrições lógicas e mineração de dados tiveram aplicação análoga pelas abordagens estudadas com relação à aquisição de conhecimento.

Pode-se observar que os trabalhos analisados encontram-se em diferentes estágios de validação. Como é característico nesse campo, poucas consideração foram desenvolvidas com relação à complexidade computacional. Testes de *benchmark* e avaliações de acurácia foram realizados, entretanto não se observou estudo com relação ao direcionamento a públicos específicos, tanto leigos como especializados. Semelhantemente, não foram realizadas avaliações do ponto de vista da interação humano-computador. Em estudos futuros torna-se importante o levantamento de critérios voltados a uma melhor compreensão de tais aspectos.

# PARTE II
# CONTRIBUIÇÕES

# Prelúdio II

*A capacidade de inferir conceitos a partir de descrições parciais é uma demanda nos modelos atuais de estrutura musical.*

---

Curtis Roads (1951 - )

A segunda parte do presente trabalho envolve a aplicação da programação multiparadigma, em união à técnica de PLI, como um método para realizar o processo de Representação de Conhecimento Musical.

Para tanto, foi elaborado um arcabouço de programação visando servir como infraestrutura ao desenvolvimento de aplicações. Desse modo, foram integrados os 4 paradigmas fundamentais de programação: Imperativo, Orientado a Objetos (ambos pela linguagem Java), Funcional (utilizando Scala) e Lógico (por meio de Prolog). 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, tendo a linguagem Java como base. A integração de Java com Prolog foi feita pelo módulo JPL. Prolog faz o interfaceamento direto com Aleph (que foi desenvolvido em Prolog.)

As bibliotecas jMusic e JavaSound fazem a integração com Java, assim como a linguagem Scala, sendo esta última realizada por meio do módulo Maven.

Os Capítulos 4, 5, 6 e 7 fazem uso deste arcabouço, sendo que no último deles, a integração descrita é representada pela Figura 8.

O Capítulo 8 apresenta uma proposta para avaliação de sistemas de Composição Algorítmica, enquanto o Capítulo 9 apresenta os resultados e considerações finais.

O sistema Fraseado, desenvolvido nesse trabalho, será tratado Capítulo 7, entretanto os capítulos anteriores apresentarão demonstrações e exemplos de código utilizando este sistema.

A versão eletrônica desse trabalho é acompanhada de arquivos de áudio gerados no formato mp3, contendo exemplos produzidos pelos códigos apresentados. Todos esses trechos musicais foram produzidos pelo sistema Fraseado. Esses arquivos poderão ser ouvidos por meio do ícone: .

# 4 Arquitetura de Software

*A Música é um exercício de aritmética secreto e aquele que a ela se entrega, às vezes ignora que maneja números.*

Gottfried Wilhelm Leibniz (1646 - 1716)

Na produção 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 utilizadas (NIERHAUS, 2009). A capacidade de representação do conhecimento envolvido possibilita que características intrínsecas à atividade musical sejam mantidas por um modelo que permita, da forma mais natural possível, expressar o raciocínio lógico e a estrutura da informação manipulada.

Neste sentido, adotou-se no presente trabalho uma abordagem multiparadigma para a configuração de um ambiente de desenvolvimento musical. Tal ambiente possibilita a criação de software direcionado à produção sonora de modo abrangente, além de atender requisitos específicos da atividade de composição musical automática. Tal ambiente resultou em uma plataforma de software livre, por meio da utilização de sistemas de código aberto, os quais são apresentados na próxima sessão.

## 4.1 Infraestrutura para o Desenvolvimento de Software

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 (HOLDEN, 2005). Além destes, modelos atuais fundamentam-se em sistemas matemáticos formais, como Programação Lógica Indutiva, ASP (Answer Set Programming) e Programação por Restrições (ANDERS, 2007). 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. 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. Atendendo às necessidades declarativas, a linguagem Prolog implementa diretamente a lógica de primeira ordem por meio de cláusulas de Horn.

Com relação ao tratamento das estruturas musicais, foi realizado um levantamento das plataformas de programação que possibilitam a manipulação do arcabouço musical. Algumas bibliotecas e frameworks foram testados destacando-se o ambiente jMusic, desenvolvido na Universidade de Tecnologia de Queensland (BROWN, 2005). Esta biblioteca implementa elementos sonoros básicos, como frequência tonal, timbre e rítmica, além de componentes musicais complexos como sequenciamento melódico, estruturas harmônicas, encadeamentos e cadências.

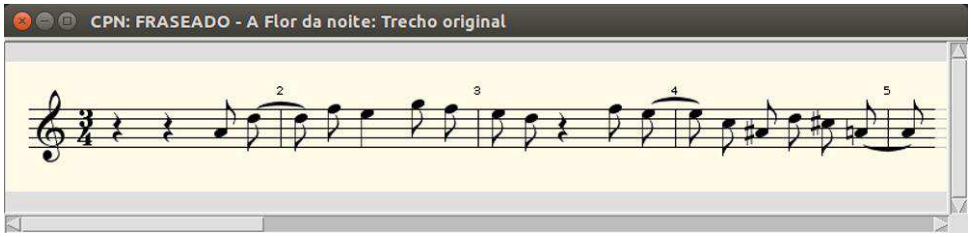
Por meio de sua API, jMusic possibilita a utilização de formatos para dados musicais variados, incluindo a integração com ferramentas para manipulação de arquivos MIDI (*Musical Instrument Digital Interface*), LilyPond (sistema para transcrição musical baseado em LaTeX) e notação de partitura convencional. Além disso, sua implementação inclui o controle do hardware específico possibilitando o trabalho de síntese sonora, fazendo uso de bancos de áudio disponíveis nas placas controladoras.

Considerando as especificações acima, bem como o modelo de software de código aberto, realizou-se a instalação e configuração dos seguintes componentes: sistema operacional, utilitários e interface gráfica Debian GNU/Linux, ambiente integrado de desenvolvimento Eclipse, implementação SWI-Prolog, plataforma Java Enterprise Edition, bibliotecas Java Sound e jMusic.

A implantação realizada levou ao estabelecimento de um ambiente híbrido atendendo requisitos fundamentais para a Computação Musical. Possibilitou-se a programação multiparadigma, manipulação e síntese sonora, bem como a representação do conhecimento musical por meio de clausulas de Horn.

Como exemplo de processamento utilizando este sistema, a Figura 2 apresenta um trecho da canção "*A flor da noite*", de autoria de Vinícius de Moraes e Toquinho (MORAES; TOQUINHO, 1971). O trecho se encontra em sua tonalidade original em *Ré menor* (🔊).

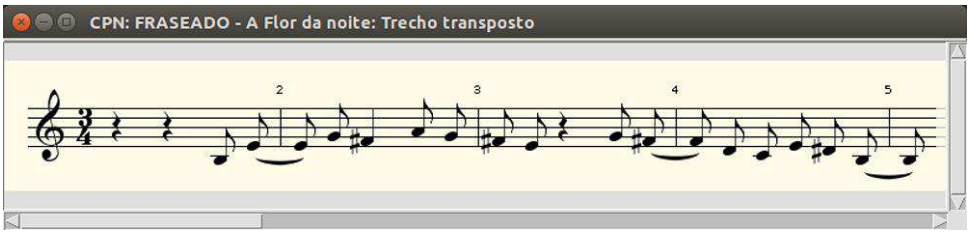
Figura 2 – Trecho melódico original.




Ao realizar uma operação de transposição tonal descendente de 6 graus, o trecho passa a ter a tonalidade de *Mi menor*, pertencente à oitava inferior, conforme apresentado na Figura 3 (🔊).

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

Figura 3 – Trecho melódico transposto.




de Prolog pode-se valer da expressividade declarativa, sendo o engenho de inferência para lógica de primeira ordem acionado diretamente a partir do código Java.

## 4.2 Considerações sobre Linguagens de Programação

A presente abordagem baseia-se nos 4 paradigmas fundamentais de programação: Imperativo, Orientado a Objeto, Funcional e Lógico, conforme apresentados na seção 2.1. Desse modo, o conhecimento musical será representado a partir de dois aspecto 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 (SEBESTA, 2012).

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 respectivamente adotados e foram escolhidas por serem referências em suas linhas de programação, além terem seu uso estabelecido no contexto das comunidades de desenvolvedores específicas (GIL, 2015).

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 (NAIM et al., 2010). A integração dessas linguagens no presente trabalho é apresentada na Figura 8.

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 (GUIMARAES, 2015).

Tabela 6 – Classificação TIOBE das linguagens Java, Scala e Prolog.

| Linguagem | Classificação | Disponibilização | Paradigma explorado |
|---|---|---|---|
| Java | 1 | 1995 | Imperativo, Orientado a Objetos |
| Scala | 32 | 2004 | Funcional |
| Prolog | 38 | 1972 | Lógico |

Fonte: Índice TIOBE 2016 (TIOBE, 2016), Programming Language Explorations (TOAL et al., 2016).

A Tabela 6 apresenta informações sobre as linguagens utilizadas, incluindo suas classificações no índice TIOBE em dezembro de 2016. O índice TIOBE [1] 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 (RABAI; COHEN; MILI, 2015).

Conforme indicado, 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. A Tabela 7 baseia-se índice Stack Overflow [2], para apresentar os aspectos relativos à remuneração referente ao uso das linguagens nos Estados Unidos, bem como ao grau de afeição associado à elas, no ano de 2016. 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. A Tabela 8 apresenta a classificação das principais linguagens utilizadas no âmbito acadêmico nos Estados Unidos, nos anos de 2010 e 2013. 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 (mensagens, partituras, áudios etc) será feita pela linguagem Java, devido à capacidade interativa que ela possui; entretanto, para fins de demonstrações específicas, poderão ser considerados

[1] www.tiobe.com/tiobe-index/
[2] http://stackoverflow.com/research/developer-survey-2016

Tabela 7 – Índice Stack Overflow - 2016.

(a) Remuneração nos EUA

| Posição | Linguagem | Valor em Dólar |
|---|---|---|
| 1 | Spark | 125.000 |
| 2 | Scala | 125.000 |
| 3 | Cassandra | 115.000 |
| 4 | F# | 115.000 |
| 5 | Hadoop | 115.000 |
| 6 | Cloud | 105.000 |
| 7 | Redis | 105.000 |
| 8 | Go | 105.000 |
| 9 | Clojure | 105.000 |
| 10 | React | 105.000 |
| 11 | Perl | 105.000 |

(b) Grau de afeição à linguagem

| Posição | Linguagem | Porcentagem |
|---|---|---|
| 1 | Rust | 79,1 |
| 2 | Swift | 72,1 |
| 3 | F# | 70,7 |
| 4 | Scala | 69,4 |
| 5 | Go | 68,7 |
| 6 | Clojure | 66,7 |
| 7 | React | 66,0 |
| 8 | Haskell | 64,7 |
| 9 | Python | 62,5 |
| 10 | C# | 62,0 |
| 11 | Node.js | 59,6 |

Fonte: Stack Overflow Developer Survey 2016 (STACKOVERFLOW, 2016).

Tabela 8 – Linguagens adotadas academicamente nos EUA em 2010 e 2013.

| Linguagem | Class. 2010 | Porc. 2010 | Class. 2013 | Porc. 2013 |
|---|---|---|---|---|
| Prolog | 1 | 12,50 | 2 | 11,76 |
| Java | 2 | 11,76 | 1 | 13,02 |
| C++ | 3 | 10,29 | 3 | 10,92 |
| Scheme | 3 | 10,29 | 4 | 9,66 |
| Python | 5 | 8,82 | 5 | 7,56 |
| Haskell | 6 | 6,62 | 6 | 7,14 |
| C | 7 | 5,88 | 10 | 3,36 |
| ML | 8 | 5,15 | 7 | 5,46 |
| Lisp | 9 | 3,68 | 8 | 4,20 |
| OCAML | 10 | 2,94 | 10 | 3,36 |
| Perl | 10 | 2,94 | 13 | 2,52 |
| SML | 10 | 2,94 | 13 | 2,52 |
| Ada | 13 | 2,20 | 10 | 3,36 |
| Smalltalk | 13 | 2,20 | 15 | 2,10 |
| Pascal | 13 | 2,20 | 18 | 0,84 |
| Ruby | 13 | 2,20 | 31 | 0,0 |
| Racket | 17 | 0,73 | 9 | 3,78 |
| Algol | 17 | 0,73 | 16 | 1,26 |
| Scala | 17 | 0,73 | 16 | 1,26 |
| Lua | 17 | 0,73 | 18 | 0,84 |

Fonte: (RABAI; COHEN; MILI, 2015).

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 (TOAL et al., 2016).

## 4.3 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 (QUEIROZ, 2010). 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) (MALHEIRO; CAVACO, 2011). Esse padrão foi definido pela ISO 16:1975 - *Standard tuning frequency (Standard musical pitch)*, cuja especificação foi revisada e confirmada em 2011 (ISO, 1975).

Para a variação de oitavas, será utilizada como referência a nota Lá, pertencente à quarta oitava do piano moderno 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 (WEINSTEIN, 1952). 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 (SCHOENBERG, 1990). Essa mesma notação será utilizada as tratar de características harmônicas, acordes e funções será utilizada a notação (ADOLFO, 1989).

Tabela 9 – Representação das 7 notas musicais.

| Nota | Dó | Ré | Mi | Fá | Sol | Lá | Si |
|---|---|---|---|---|---|---|---|
| Representação | C | D | E | F | G | A | B |

A Tabela 9 apresenta o padrão adotado para representação das 7 notas musicais, enquanto a Tabela 10 contém a representação para referências utilizadas na cifragem de acordes. 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, indicando (como *sexta* ou *grau13*).

Tabela 10 – Referências para representação de acordes.

| Referência | Maior | Menor | Diminuto | Aumentado | Sustenido | Bemol |
|---|---|---|---|---|---|---|
| Representação | M | m | Dim | Aum | s | b |

Seguindo essas convenções, a corda mais aguda do violino é representada por E5, os acordes de Sol Maior e Fá sustenido diminuto, será representado por GM e FsDim, enquanto um acorde com função mediante (terceiro grau menor) poderá ser representado por III ou grau3.

# 5 Primeira Contribuição

*O que é música, senão sons organizados?*

Edgar Varèse (1883 - 1965)

## 5.1 Representação Multiparadigma e Musicologia Sistemática

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 (CABRAL, 2014).

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 (PARNCUTT, 2012).

## 5.2 Representando Componentes Musicais

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 (NUSSBAUM, 2007).

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 (DANNENBERG, 1993). Tendo posse de tal capacidade, pode-se aplicar componentes psicoacústicos na automação de tarefas como análise musical, composição, recomendação musical e categorização de estilos (ROADS, 1996; GEBHARDT; DAVIES; SEEBER, 2016).

Na próximas seções, 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" (LACERDA, 2008), "The physics of music and musical instruments" (LAPP, 2003) e "Music: a mathematical offering" (BENSON, 2008).

## 5.3 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* (ROBERTS, 2009).

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 e eixo das ordenadas expressa a amplitude, com frequência $f$, amplitude máxima $A$ e fase inicial $\theta$ pode ser representado por uma senoide (Figura 4a). A função senoidal que representa uma onda é dada por:

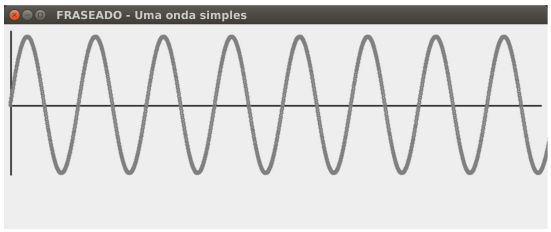
$$y = A\,\mathrm{sen}(2\pi f x + \theta) \tag{5.1}$$

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*.

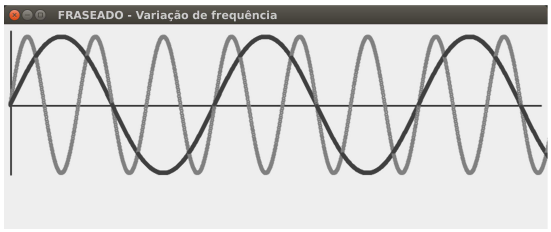
Na programação de um computador, o **Paradigma Imperativo** apresenta recursos de programação fundamentais para a manipulação das grandezas citadas. Tais recursos constituem um modo natural de representação por meio de atribuições, sequências, repetições de comandos e sub-rotinas. O algoritmo apresentado na Listagem 5.1, exemplifica a programação imperativa aplicada a esse conceito.

Figura 4 – Modelagem de ondas pelo sistema Fraseado.

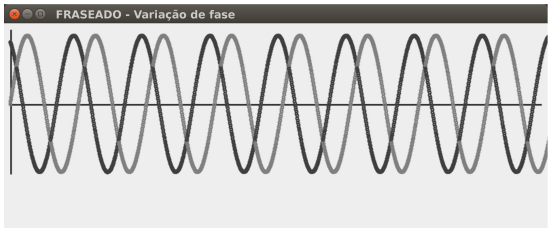
(a) A3: $f$=220 Hz



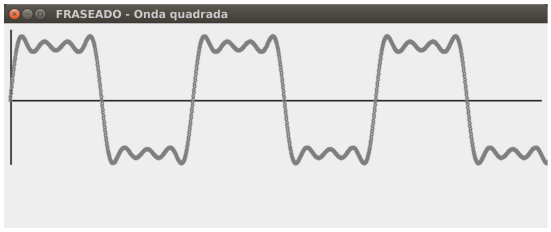

(b) A3: $f$=220 Hz (cinza), D2: $f$=73,416 Hz




(c) A3: $\theta$=0 (cinza), A3: $\theta$=2




(d) Onda quadrada para a nota E2




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.

Listagem 5.1 – Algoritmo para uma onda senoidal.

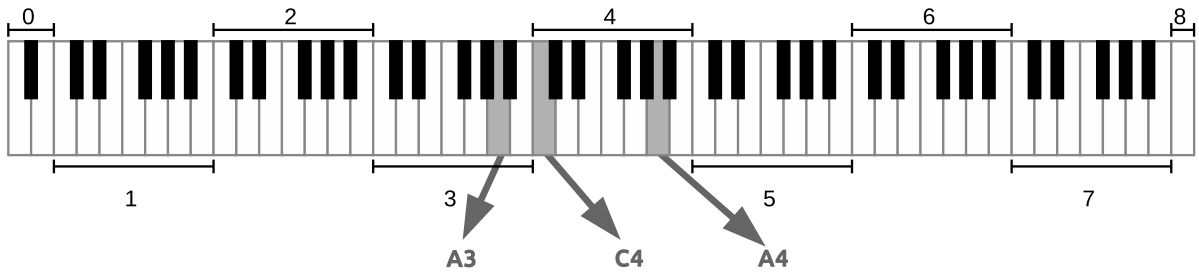
```
1  funcao incrementaAudio (x)
2      abreCanalAudio
3      gravaAudio(x)
4      fechaCanalAudio
5
6  frequencia = 220
7  amplitudeMax = 100
8  fase = 0
9  taxaAmostragem = 44100
10 para i variando de 1 ate taxaAmostragem faca
11      s = seno(2 * 3,14 * frequencia * i/taxAmostragem + fase)
12      incrementaAudio(amplitudeMax * s)
```

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 (ROBERTS, 2009).

Tomando-se o piano moderno 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 Figura 5 indica a disposição dessas notas.

Figura 5 – As notas A3, C4 e A4 no piano moderno.




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 do algoritmo da Listagem 5.1 possibilita que o gráfico da Figura 4a seja produzido. A nota A3, modelada por esse algoritmo, pode ser sintetizada, armazenada e reproduzida por meio da API JavaSound (🔊).

A Figura 4b apresenta, juntamente com A3, o gráfico da nota D2 (73,416 Hz), cujo valor é aproximadamente 1/3 da nota A3. Conforme essa relação, o gráfico indica que um período de D2 corresponde a 3 períodos de A3. Na Figura 4c, a mesma nota A3 possui fase $\theta$=0 no primeiro sinal e $\theta$=2 no segundo. A Figura 4d apresenta a onda quadrada, que pode ser produzida pela alteração do laço de repetição no código original. Com isso, uma nova forma de onda pode ser representada - esse processo é denominado Síntese Aditiva. A Listagem 5.2 apresenta o código em Java para essa onda (🔊).

Listagem 5.2 – Uma onda quadrada em Java.

```
1 for(int i=0; i<taxaAmostragem; i++) {
2   double s1 = Math.sin(2.0*Math.PI*freq*i/txAmostra*1.0+fase);
3   double s2 = Math.sin(2.0*Math.PI*freq*i/txAmostra*3.0+fase);
4   double s3 = Math.sin(2.0*Math.PI*freq*i/txAmostra*5.0+fase);
5   double s4 = Math.sin(2.0*Math.PI*freq*i/txAmostra*7.0+fase);
6
7   buf[0] = (byte) (s1*vol + s2*vol/3 + s3*vol/5 + s4*vol/7);
8   sourceDL.write(buf,0,1);
9 }
```

## 5.4 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:

## 5.5 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 5.3 apresenta a estrutura base para representação de um trecho da obra "*Judas Maccabaeus*", de George Frideric Handel (HANDEL; MORELL, 1747) (🔊).

Listagem 5.3 – Representação musical em Java/jMusic.

```
1  public final class maccabaeus implements JMC {
2    Score partitura;
3    Part homens, mulheres, cordas;
4    Phrase fraseH, fraseM, fraseC;
5
6    public static void main(String[] args) {
7      homens   = new Part("Homens", OOH, 0);
8      mulheres = new Part("Mulheres", OOH, 1);
9      cordas = new Part("Cordas", SLOW_STRINGS , 2);
10     partitura = new Score("Maccabaeus", 106.0);
11
12     criaMusica();
13
14     partitura.addPart(homens);
15     partitura.addPart(mulheres);
16     partitura.addPart(cordas);
17
18     Play.midi(partitura);
19   }
20 }
```

## 5.6 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[1]. 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, uma vasta 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*. A Tabela 11 apresenta as diferentes razões para esses temperamentos, definindo as afinações para os graus de uma oitava.

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 (SEBESTA, 2012). O método `map`, da linguagem Scala, retorna um contêiner de pares atributo/valor. A utilização desse

[1] Apesar do intervalo de oitava ser composto por 12 notas, ele recebe essa designação por envolver 8 graus da escala diatônica.

Tabela 11 – Razões utilizadas nos principais temperamentos.

| | 1º | 2º | 3º | 4º | 5º | 6º | 7º | 8º |
|---|---|---|---|---|---|---|---|---|
| Pitagórico | 1:1 | 9:8 | 81:64 | 4:3 | 3:2 | 27:16 | 243:128 | 2:1 |
| Justo | 1:1 | 9:8 | 5:4 | 4:3 | 3:2 | 5:3 | 15:8 | 2:1 |
| Mesotônico | 1:1 | $\sqrt{5}$:2 | 5:4 | $2{:}5^{1/4}$ | $5^{1/4}$:1 | $5^{3/4}$:2 | $5^{5/4}$:4 | 2:1 |
| Igual | 1:1 | 1,122:1 | 1,260:1 | 1,335:1 | 1,498:1 | 1,682:1 | 1,888:1 | 2:1 |

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.

A Listagem 5.4 apresenta dois níveis de mapeamento para implementar os sistemas de temperamento Pitagórico, Justo, Mesotônico e igual. O valor de uma altura específica pode ser obtido pela associação `temp(<sistema>)(<nº nota>)`.

Listagem 5.4 – Temperamentos em Scala.

```
var temp = Map(
 "pit" -> Map(1->1.0,   2->1.053, 3->1.125,  4->1.185,  5->1.265,  6->1.333,
              7->1.404, 8->1.5,   9->1.58,  10->1.687, 11->1.778, 12->1.898),
 "jus" -> Map(1->1.0,   2->1.067, 3->1.125,  4->1.2,    5->1.25,   6->1.333,
              7->1.406, 8->1.5,   9->1.6,   10->1.667, 11->1.8,   12->1.875),
 "mes" -> Map(1->1.0,   2->1.07,  3->1.118,  4->1.196,  5->1.25,   6->1.337,
              7->1.398, 8->1.496, 9->1.6,   10->1.672, 11->1.789, 12->1.869),
 "igu" -> Map(1->1.0,   2->1.059, 3->1.122,  4->1.189,  5->1.26,   6->1.335,
              7->1.414, 8->1.498, 9->1.587, 10->1.682, 11->1.782, 12->1.888)
)
```

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)
```

## 5.7 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 serão tratados com maior detalhe no Capítulo 6 e 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 5.5.

Uma chamada à meta `triadeMaior('D',T)` será satisfeita com a instanciação `T = ['D', 'Fs', 'A']`.

Listagem 5.5 – Tríade Maior em Prolog.

```
1  triadeMaior(Fund, Triade) :-
2    insereFund(Fund, L1),
3    insereTerca(L1, L2),
4    insereQuinta(L2, Triade).
5
6  insereFund(Fund, [L]) :-
7    append([], Fund, L).
8
9  insereTerca([C|R], L) :-
10   tercaMaior(C, Ter),
11   append([C|R], [Ter], L).
12
13 insereQuinta([C|R], L) :-
14   quintaMaior(C, Qui),
15   append([C|R], [Qui], L).
```

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 5.7 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:

Listagem 5.6 – Conversão de átomos em Prolog para o valor MIDI.

```
1  midi('C',60).  midi('Cs',61). midi('D',62).  midi('Ds',63).
2  midi('E',64).  midi('F',65).  midi('Fs',66). midi('G',67).
3  midi('Gs',68). midi('A',69).  midi('As',70). midi('B',71).
4
5  converteMidi([], []).
6  converteMidi([C|R], [C1|R1]) :-
7    midi(C, C1),
8    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 Listagem 5.7 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 (NASCIMENTO; BRANT, 1983) (🔊).

Listagem 5.7 – Especificação de uma frase harmônica em Java.

```
1 int[] miB  = {AS3,DS4,G4};
2 int[] siM7 = {A3,D4,FS4};
3 fHarm = new CPhrase(0);
4 fHarm.addChord(miB, HN);
5 fHarm.addChord(siM7, HN);
```

# 6 Segunda Contribuição

*A arte da Programação Lógica está em construir programas concisos e elegantes, que apresentem o significado desejado.*

Leon S. Sterling, Ehud Y. Shapiro (1955 - , 1955 - )

## 6.1 Programação Lógica, Indução e Representação de Conhecimento Musical

Sistemas baseados em conhecimento utilizam técnicas de Inteligência Artificial aplicadas à resolução de problemas, com o objetivo de dar suporte à tomada de decisão, incluindo a capacidade de aprendizado. O processo de modelagem computacional do conhecimento inclui a definição de um ou mais mecanismos de aprendizado, visando dotar o sistema com a capacidade de gerar informações estruturadas a partir de bases previamente estabelecidas (MIRA et al., 2004).

De acordo com o processo de modelagem de conhecimento, a Programação Lógica Indutiva constitui um formalismo para representação de conhecimento fundamentado em dois aspectos: Programação Lógica e Aprendizado de Máquina. A PLI foi utilizada originalmente como uma linguagem representacional uniforme para manutenção de amostras, conhecimento preliminar e hipóteses, sendo aplicada posteriormente na modelagem multi-relacional e mineração de dados (INOUE; OHWADA; YAMAMOTO, 2016).

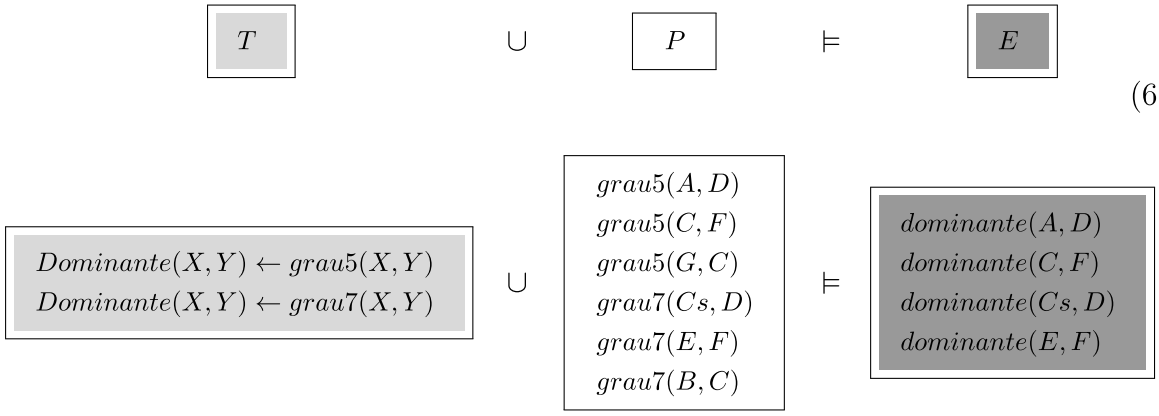
A seguir, apresenta-se a abordagem utilizando PLI aplicada à Representação de Conhecimento Musical de acordo com seus dois aspectos característicos. Serão utilizadas definições extraídas dos livros "Knowledge-based systems" (AKERKAR; SAJJA, 2010), "Foundations of logic programming" (LLOYD, 2012) e "An introduction to inductive logic programming" (DŽEROSKI; LAVRAČ, 2001).

## 6.2 Raciocínio Musical Dedutivo

*Lógica* consiste em um sistema algébrico que se baseia no estudo de métodos e princípios utilizados com o propósito de se distinguir entre o raciocínio correto e o incorreto expressos por meio de declarações. Uma *proposição* é uma declaração que pode assumir o valor verdadeiro ou falso, mas não ambos ao mesmo tempo. *Inferência* é o processo pelo qual uma proposição é confirmada, tendo como base uma ou mais proposições anteriores. Um *predicado* consiste em uma relação que utiliza valores constantes ou variáveis como seus argumentos.

O raciocínio dedutivo é estabelecido a partir de princípios gerais ou regras lógicas. Esses princípios são combinados com uma base de conhecimento, produzindo conclusões lógicas. Essas regras são comumente chamadas de *Teoria* (T), a base de conhecimento é chamada de *Conhecimento Prévio* (P), enquanto as conclusões são chamadas de *Exemplos* ou amostras (E).

Considerando, como ilustração, o campo harmônico de um acorde, a função Dominante pode ser exercida pelos graus V e VII. A especificação dessa regra em união com o conhecimento prévio definindo amostras para os graus envolvidos possibilita que sejam inferidas conclusões com relação aos acordes que exercem a função Dominante:




(6.1)

A formalização do processo dedutivo possibilita que regras de inferência possam ser aplicadas produzindo resoluções lógicas. Nesse sentido, a *Lógica de Predicados de Primeira Ordem* constitui uma formalização para esse processo. Essa lógica é constituída a partir do Cálculo Proposicional e do Cálculo de Predicados. A aplicação musical destes formalismos será apresentada a seguir.

### 6.2.1 O Cálculo proposicional

O Cálculo Proposicional caracteriza-se pelo uso de dois elementos básicos: átomos (ou termos) e conectivos lógicos. Sobre esses elementos são aplicadas regras de derivação que possibilitam o estabelecimento de teoremas formais. Átomos são objetos como `nota`, `clave` e `compasso`. Proposições são sentenças do tipo:

***Fá é uma nota musical*** (Verdadeiro)
***Pausa é um período de som*** (Falso)

Os conectivos constituem as relações lógicas de conjunção, disjunção, negação, implicação e equivalência. Considerando que uma proposição assuma o valor verdadeiro em um determinado contexto, o sistema dedutivo (conjunto de regras de derivação) possibilita a inferência de uma ou mais novas proposições, a partir da original, que serão verdadeiras

nesse mesmo contexto. Por exemplo, a regra de inferência por *modus ponens* é dada por:

$$\frac{P \rightarrow Q, P}{\therefore Q}$$

Esta regra possibilita que a partir de afirmações como:

***Se o trecho está na escala de Mi então a dominante é Si***

e

***O trecho está na escala de mi***,

possa ser inferido:

***A dominante do trecho é si***.

Desse modo, a Lógica Proposicional constitui uma linguagem formal para expressão do conhecimento referente ao mundo sendo modelado. Entretanto, ela não possui a capacidade de quantificação entre as proposições. Considerando, por exemplo, as declarações:

**P: Fá é uma nota musical**
**Q: Sol é uma nota musical**
**R: Si é uma nota musical**

Não se pode extrair qualquer conclusão sobre similaridades entre P, Q e R. Além disso, não podem ser representadas sentenças como "**Todas as frequências entre as faixas F1 e F2 são audíveis**", pois para expressar tais afirmações, deveria ser especificada uma proposição para cada frequência possível.

O Cálculo de Predicados possui a capacidade de superar essas limitações, o que é realizado por meio de quantificadores que podem ser aplicados a variáveis específicas.

### 6.2.2 O Cálculo de Predicados

O Cálculo de Predicados, ou Cálculo Relacional, é uma ampliação do Cálculo Proposicional, incluindo maior capacidade para modelar o conhecimento do mundo. Para isso, 3 componentes são tomados como base: Termos, Predicados e Quantificadores.

Termos são objetos pertencentes a um domínio específico e se dividem em:

- Constante - um átomo individual ou um conceito fixo (ex: andante, 4, oitava).
- Variável - um componente que pode assumir múltiplos valores.
- Função - um mapeamento que associa n termos a um termo. A expressão $f(t_1 \ldots t_n)$ é um termo se cada argumento $t_i$ for um termo.

Predicados, são relacionamentos mapeando termos para um valor `Verdadeiro` ou `Falso`. Assumindo-se, como exemplo, que os termos mais básicos (átomos) sejam graus de uma escala, podemos ter predicados do tipo:

**harmoniza**(fundamental, quinta)
**harmoniza**(**oitava**(fundamental), fundamental)

Nesse caso, **harmoniza** é um predicado e **oitava** é uma função.

Quantificadores são componentes que têm a função de mensurar cardinalmente uma variável. Há dois tipos de quantificadores no Cálculo de Predicados: "Existencial" ($\exists$) e "Universal" ($\forall$). Por exemplo, a sentença "**Todo acorde maior possui um relativo**" pode ser representada por: $(\forall x)(acordeMaior(x) \rightarrow possuiRelativo(x))$.

De acordo com essas definições, a descrição de notas musicas pode ser feita por meio de fatos como `nota(fá)`, `nota(sol)`, `nota(si)`. A sentença "**Todas as frequências entre as faixas F1 e F2 são audíveis**" pode ser representada por: $(\forall x)(freq(x, f1, f1) \rightarrow audivel(x))$.

No cálculo de predicados, uma sentença como "**A é maior que B**" pode ser representada por *maior(A,B)*. Essa relação é definida por:

```
maior(A,B) = Verdadeiro, se A > B
           = Falso, caso contrario
```

O predicado identificado por `maior` recebe dois termos e mapeia para `Verdadeiro` ou `Falso`, dependendo do valor desses termos. A sentença "**O piano interpreta qualquer música**" é representada por: $(\forall x)$ `interpreta(piano, x)`, o que é mapeado para `Verdadeiro` quando `x` for instanciado com qualquer valor.

Uma declaração do tipo "**Toda oitava harmoniza com sua fundamental**" é representada por: $(\forall x)$ `harmoniza(oitava(x), x)`. Nesse caso, `oitava` é uma função que mapeia x para sua oitava. O predicado `harmoniza` recebe dois argumentos e mapeia para `Verdadeiro` ou `Falso` dependendo do valor de seus termos.

### 6.2.3 Cálculo de Predicados de Primeira Ordem

Como continuidade dos conceitos abordados, o Cálculo de Predicados de Primeira Ordem constitui um sistema formal baseado em uma linguagem representacional, livre de contexto e declarativa. Por meio dele, representações de raciocínio e conhecimento podem ser feitas de modo conciso e sem ambiguidades. As regras básicas para formulação de sentenças são as mesmas do Cálculo Proposicional e Cálculo de Predicados. Adicionalmente, utilizam-se as *fórmulas bem formadas* (FBF) para representar elementos do mundo real.

A constituição de uma fórmula bem formada é estabelecida como se segue:

- Uma fórmula atômica $p(t_1, \ldots, t_n)$ é uma FBF, onde p é um símbolo predicado e $t_1, \ldots, t_n$ são os termos. Ela também é chamada de átomo.
- Se $\alpha$ e $\beta$ são FBFs, então $(\neg\alpha)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$, $(\alpha \leftrightarrow \beta)$ são fórmulas bem formadas.
- Se $\alpha$ é uma FBF e $x$ é uma variável livre em $\alpha$, então $(\forall x)\alpha$ e $(\exists x)\alpha$ são FBFs.
- FBFs são geradas por um número finito de aplicações sobre as regras acima.

Dada a descrição: "**Todo acorde maior possui um relativo**. **RéM é um acorde maior**. **Portanto o acorde de RéM possui um relativo**". Ao ser representada por uma fórmula do Cálculo de Predicados de Primeira Ordem, serão realizados os seguintes passos:

**Todo acorde maior possui um relativo** :

$$(\forall x)(acordeMaior(x) \rightarrow possuiRelativo(x))$$

**RéM é um acorde maior** :

$$acordeMaior(RéM)$$

**O acorde de RéM possui um relativo** :

$$possuiRelativo(RéM)$$

Desse modo, a descrição completa pode ser representada pela fórmula:

$$(\forall x) \ ( \ (acordeMaior(x) \ \rightarrow \ possuiRelativo(x)) \ \wedge$$
$$acordeMaior(RéM) \ ) \ \rightarrow \ possuiRelativo(RéM)$$

### 6.2.4 Cláusulas de Horn

Cláusulas de Horn são formulações lógicas que seguem um estilo particular na formalização de regras. Tal estilo é definido pela disjunção de literais, com a característica de no máximo um deles ser positivo. Um *literal* é uma fórmula atômica não estruturada (sem conectivos lógicos), que pode ser dividida em dois tipos:

- Um **literal positivo**, equivalente a um átomo.
- Um **literal negativo**, equivalente à negação de um átomo.

Uma *cláusula* é uma expressão que utiliza uma coleção finita de literais, tornando-se verdadeira sempre que pelo menos um de seus literais seja verdadeiro (disjunção) ou quando todos os literais o sejam (conjunção).

A especificação de uma cláusula de Horn segue a forma:

$$L_1, \ldots, L_n \ \Rightarrow \ L(\equiv \neg L_1 \vee \ldots \vee \neg L_n \vee L)$$

A elaboração dessa expressão parte de uma relação de equivalência aplicada sobre a operação de *condicional lógica*.

$$A \longrightarrow B$$

Sobre essa condicional, também denominada *implicação material*, incide a relação de equivalência:

$$A \longrightarrow B \quad \Longrightarrow \quad \neg A \quad \vee \quad B \tag{6.2}$$

Esse relação pode ser verificada pela comparação das *tabelas-verdade* de cada fórmula. Sendo $A$ e $B$ literais, em ambos os casos os valores-verdade são idênticos:

| $A$ | $B$ | $A \rightarrow B$ |
|---|---|---|
| V | V | V |
| V | F | F |
| F | V | V |
| F | F | V |

| $A$ | $B$ | $\neg A \vee B$ |
|---|---|---|
| V | V | V |
| V | F | F |
| F | V | V |
| F | F | V |

Generalizando a expressão 6.2 obtém-se:

$$(A_1 \wedge \cdots \wedge A_N) \longrightarrow B \quad \Longrightarrow \quad \neg(A_1 \wedge \cdots \wedge A_N) \quad \vee \quad B$$

Aplicando o Teorema de *De Morgan* atinge-se a estrutura básica de uma cláusula de Horn:

$$(A_1 \wedge \cdots \wedge A_N) \longrightarrow B \quad \Longrightarrow \quad \neg A_1 \vee \cdots \vee \neg A_N \quad \vee \quad B$$

Escrevendo-se a primeira parte dessa equivalência em notação clausal, a expressão assume a forma:

$$\alpha \longleftarrow \beta_1, \ldots, \beta_N \tag{6.3}$$

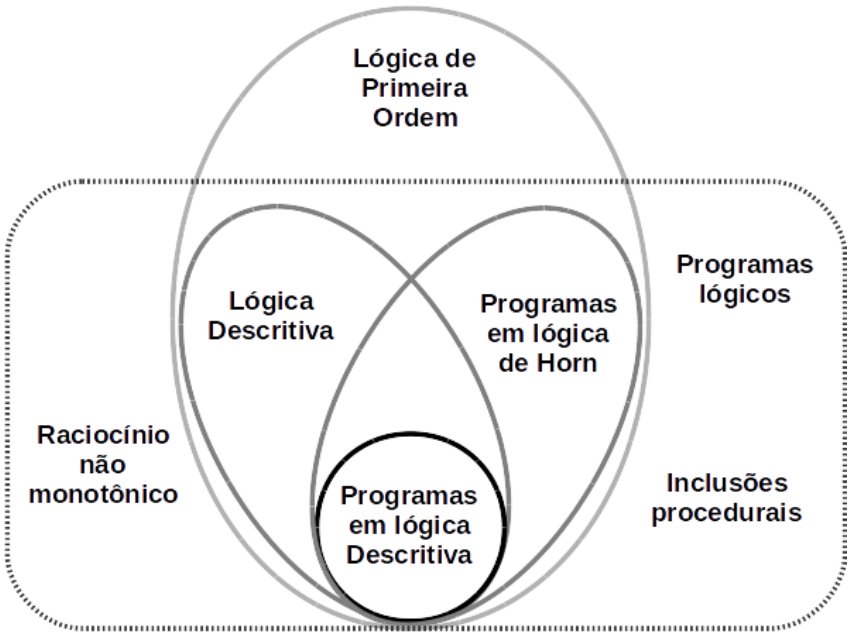
Nessa forma, $\alpha$ é dito ser a *cabeça* da cláusula, enquanto $\beta_1, \ldots, \beta_N$ é chamado de *corpo*. Cláusulas de Horn expressam um subconjunto de declarações do Cálculo de Predicados de Primeira Ordem. A Figura 6 apresenta relações entre a Lógica de Primeira Ordem e os conjuntos lógicos expressivos associados à Representação de Conhecimento.

A aplicação musical da Lógica de Primeira Ordem pode ser feita diretamente sobre elementos básicos como nota, intervalo, além de conceitos mais complexos ligados à harmonia funcional. Em ambos os casos, cláusulas de Horn podem ser utilizadas para tal representação. Uma cláusula de Horn em que o corpo seja vazio, é chamada de unitária, ou *fato*, como em:

$$semibreve$$

$$nota(mi)$$

Figura 6 – Sobreposições expressivas na Representação de Conhecimento.




Fonte: Vulcan Inc., Benjamin Grosof, Mike Dean, and Michael Kifer (GROSOF; DEAN; KIFER, 2009)

Uma cláusula de Horn contendo exatamente um literal positivo é chamada de *cláusula definida*.

$$terca(la) \leftarrow fundamental(fá)$$

$$tetradeMaior(fá) \leftarrow nota(fá), nota(lá), nota(dó), nota(mi)$$

Uma cláusula de Horn que não possua literal positivo é chamada de *meta* ou *goal*.

$$\leftarrow resoluçãoFinal(sol)$$

$$\leftarrow cadência1, cadência2, cadência3$$

A linguagem Prolog consiste em uma implementação lógica baseada diretamente nas cláusulas de Horn. A expressão 6.3, que constitui a formulação básica de uma cláusula de Horn, é representada em Prolog por declarações do tipo:

```
a :- b1,...,bn.
```

Desse modo, o conhecimento musical pode ser expresso por meio de predicados em Prolog como:

```
modo(dorico).
acordeMaior('D').
possuiRelativo('B') :- acordeMaior('B').
```

A Listagem 6.1 apresenta a codificação em Prolog referente ao conjunto de regras 6.1.

Listagem 6.1 – Representação da função Dominante em Prolog.

```
1  grau5('A','D').
2  grau5('C','F').
3  grau5('G','C').
4  grau7('Cs','D').
5  grau7('E','F').
6  grau7('B','C').
7
8  dominante(X,Y) :- grau5(X,Y).
9
10 dominante(X,Y) :- grau7(X,Y).
```

Para essa base de conhecimento, a execução sucessiva da meta `dominante(X,Y)` até a totalidade de instanciações, produzirá a seguinte sequência de execução:

```
?- dominante(X,Y).
X = 'A',  Y = 'D';
X = 'C',  Y = 'F';
X = 'G',  Y = 'C';
X = 'Cs', Y = 'D';
X = 'E',  Y = 'F';
X = 'B',  Y = 'C'.
?-
```

## 6.3 Raciocínio Musical Indutivo

Enquanto no processo dedutivo parte-se de princípios gerais para a produção de conclusões, o raciocínio indutivo parte de observações a partir de exemplos, levando ao estabelecimento de princípios gerais que tenham a capacidade de sustentar logicamente as observações feitas. O formulação básica do processo indutivo consiste na especificação do seguinte problema:

***Dados***

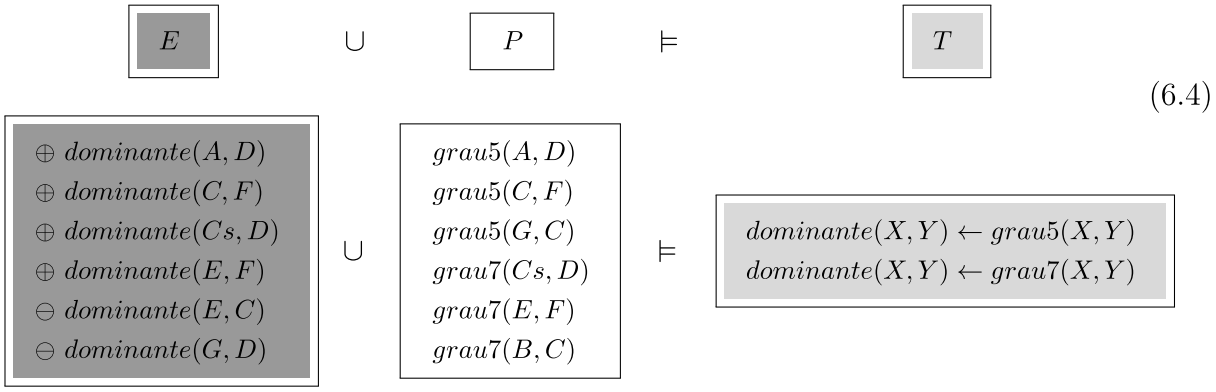
- *Um conjunto de exemplos observados (E)*
- *Uma base consistente de conhecimento prévio (P)*

***Encontre***

- *Um conjunto de cláusulas (hipótese), formando uma teoria (T), que acarrete logicamente todos os exemplos conhecidos.*

Considerando a determinação da função Dominante, a partir de exemplos obtidos como entrada, juntamente com o conhecimento prévio sobre as relações dos graus V e VII,

o processo indutivo produz o conjunto de regras que define essa função:




(6.4)

Desse modo, o processo de aprendizado indutivo consiste em encontrar uma relação desconhecida, e elaborar sua especificação por meio de um *predicado alvo*. Isso é obtido a partir dos exemplos observados, sendo dadas relações conhecidas como conhecimento prévio. Os exemplos de treinamento podem ser tanto positivos (valores verdadeiros) como negativos (valores falsos). Esses exemplos são representados como fatos básicos referentes ao predicado alvo, sendo que, de modo geral, a informação contida no conhecimento prévio possui a mesma forma de representação (DŽEROSKI; LAVRAČ, 2001).

### 6.3.1 Aplicação da Programação Lógica Indutiva

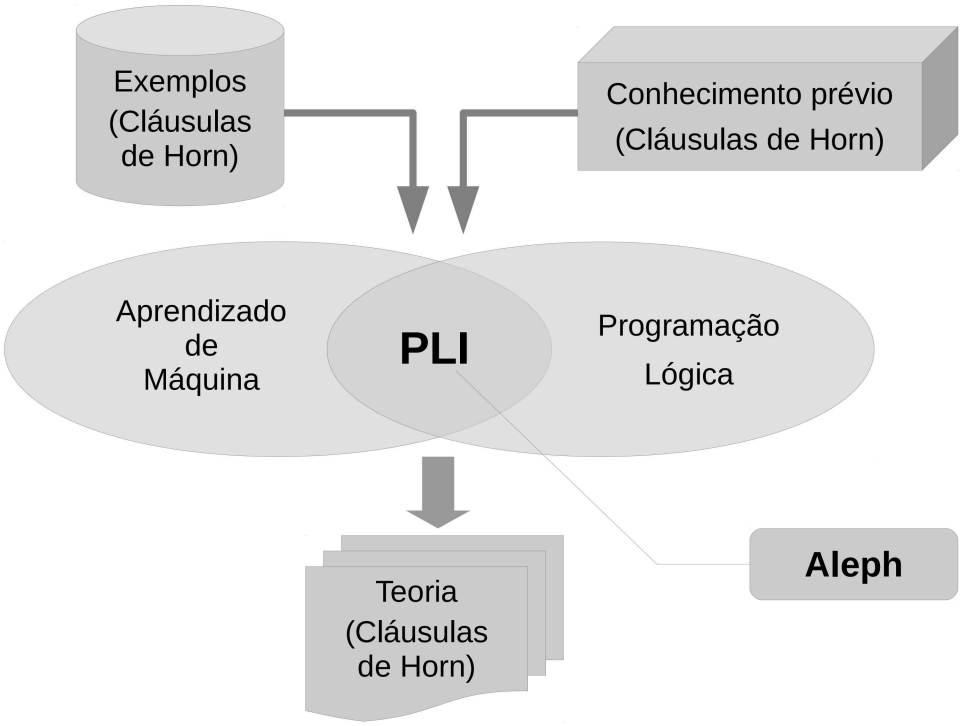
Em termos práticos, a aplicação do processo indutivo pela PLI define em uma série de recursos (ferramentas e técnicas) que possibilitam a indução de hipóteses (teorias) a partir de observações (exemplos), produzindo novo conhecimento a partir da experiência.

Os sistemas de PLI utilizam o formalismo lógico como linguagem representativa, mais especificamente a Lógica de Primeira Ordem. Isso possibilita a declaração de conceitos que não poderiam ser expressos pelas técnicas tradicionais, baseadas em atributo-valor do cálculo proposicional. Como decorrência, tais sistemas são frequentemente especificados utilizando cláusulas de Horn como base, por meio da linguagem de programação Prolog.

O sistema indutivo Aleph utiliza esse modelo, possibilitando a descoberta de conhecimento por meio de módulos de programação que são aplicados à exploração de conceitos do ponto de vista lógico (SRINIVASAN, 2007). A Figura 7 apresenta o esquema geral envolvendo a PLI e o sistema Aleph.

Na aplicação desse processo ao conceito de função Dominante, exemplificado pelo conjunto de regras 6.4, busca-se o aprendizado da relação *dominante(X, Y)*. Tal relação é determinada pelo conceito da função harmônica exercida por um grau específico $X$ de uma escala sobre outro grau $Y$ da mesma escala. Isso é feito por meio das relações $P$ especificadas na base de conhecimento prévio. A essas relações são incorporados os

Figura 7 – Programação Lógica Indutiva.




exemplos de treinamento $E$, constando de fatos positivos e negativos com relação ao predicado alvo, representados pelos símbolos $\oplus$ e $\ominus$.

A realização do processo indutivo possibilita que, dadas as relações *grau5(X, Y)* e *grau7(X, Y)*, seja derivada a relação *dominante(X, Y)*, o que produzirá o predicado *dominante*, formado por 2 cláusulas:

```
dominante(X,Y) :-
            grau5(X,Y).
dominante(X,Y) :-
            grau7(X,Y).
```

Nesse caso, o predicado alvo *dominante* que foi produzido indutivamente, constitui o resultado da teoria $T$ formulada. Na realização do presente trabalho, o processo indutivo, aplicado por meio do sistema Aleph para formulação da teoria em questão, foi desenvolvido com o uso de quatro tipos fundamentais de arquivos:

- **conceito_dominante.pl** - É o arquivo inicial, sendo executado diretamente pelo interpretador Prolog. Realiza as operações básicas de carregamento da biblioteca Aleph, leitura das bases de conhecimento prévio, exemplos positivos e negativos, além de ser o ponto de partida para o processo de indução. Uma vez finalizado o processo, esse arquivo coleta as possibilidades geradas a partir da teoria construída e as retorna no formato apropriado.

- **conceito_dominante.b** - Contém o conhecimento prévio, também chamado de conhecimento de fundo, na forma de cláusulas Prolog que especificam informações

relevantes para o domínio. Também pode conter diretivas entendidas pelo compilador sendo utilizado.

- **conceito_dominante.f** - Exemplos positivos do conceito a ser aprendido pelo sistema. Esses exemplos são constituídos de predicados básicos e representam os fatos observados a partir dos quais será buscada a derivação do predicado alvo.
- **conceito_dominante.n** - Exemplos negativos do conceito a ser aprendido. Assim como os exemplos positivos, a representação ocorre por meio de predicados básicos. Apesar do uso desse arquivo não ser obrigatório para Aleph, ele pode definir em alguns casos a obtenção satisfatória do predicado alvo.

### 6.3.2 Especificação do Sistema Indutivo

O processamento lógico indutivo realizado por Aleph, ocorre por meio de um algoritmo fundamentado na seguinte estrutura:

1. **Seleção de exemplo** - Selecione um exemplo a ser generalizado. Se não houver, pare; caso contrário, vá para a próxima etapa.
2. **Saturação** - Construa a cláusula mais específica que envolva o exemplo selecionado e atenda as restrições fornecidas pela linguagem. Geralmente esse é uma cláusula definida com muitos literais, identificada como *cláusula base*.
3. **Busca** - Busque uma cláusula mais geral do que a cláusula base. Para isso, procure pelo subconjunto de literais na cláusula base que possua a maior pontuação. A pontuação dos literais é definida pelos exemplos positivos e negativos que eles possam envolver.
4. **Remoção de redundâncias** - A cláusula com maior pontuação é adicionada à teoria atual, e todos os exemplos redundantes são removidos. Deve-se considerar que a melhor cláusula pode tornar redundantes as cláusulas que sejam diferentes dos exemplos. Volte ao Passo 1.

Além do conteúdo apresentado, as informações envolvendo conhecimento prévio abrangem elementos como restrições específicas de linguagem e diretivas de pesquisa para Aleph. Dentre estas, o presente trabalho faz uso das especificações fundamentais, as quais referem-se a *modos*, *tipos* e *determinações*, que serão apresentados nas próximas seções.

### 6.3.3 Especificação de Modos

O arquivo `conceito_dominante.b` contém descrições referentes aos predicados que declaram objetos de dados e os tipos desses dados. É por meio dessas descrições que o

sistema indutivo identifica se um predicado pode ser usado na cabeça de uma regra, por meio da declaração `modeh`, ou em seu corpo, utilizando `modeb`. Ainda nesta seção, são especificados os tipos de argumento que cada predicado pode assumir, incluindo as formas de instanciações possíveis de serem realizadas. Nestas declarações de modo, são definidos 2 elementos: o valor do *número de chamadas* e o *modo do predicado*.

```
mode<h|b>(<número de chamadas>, <modo>)
```

**Número de chamadas (recall)** - Esse valor determina a quantidade máxima de chamadas que podem ser realizadas na execução da busca por um predicado alvo. Com isso pode-se limitar o número de instanciações alternativas que um predicado pode assumir. Essas instanciações consistem em substituições de tipo que serão aplicadas a variáveis ou constantes e influenciam diretamente no tempo de aplicação do processo indutivo.

No caso da relação Dominante, considerando os graus V e VII, que possuem as qualidades funcionais *Forte* e *Meio-forte*, o valor do número de chamadas está configurado para 2, pois em qualquer tonalidade, o grau referente à Tônica terá dois graus dominantes. Já na especificação dos respectivos graus, o número de chamadas está configurado para 1, pois cada Tônica possui apenas um grau de posição V na escala e apenas um de posição VII.

**Modo do predicado** - O segundo valor definido na declaração de modo determina a tipagem das variáveis esperadas nos argumentos de entrada do predicado a ser induzido.

Dessa maneira, para o aprendizado da relação `dominante(X,Y)`, as declarações de modo assumem o formato apresentado na Listagem 6.2.

Listagem 6.2 – Especificação de modos para o predicado Dominante em Aleph.

```
1  :- modeh(2,dominante(+grau,-grau)).
2  :- modeb(1,grau5(+grau,-grau)).
3  :- modeb(1,grau7(+grau,-grau)).
```

Nesse código, a primeira linha define a relação em que se busca o aprendizado. Ela apresenta a declaração `modeh`, indicando que a relação `dominante(X,Y)` será utilizada na composição da cabeça das regras. Nesse caso, os parâmetros `X` e `Y` do tipo `grau`, representam os valores de entrada e saída. O valor de entrada, indicado pelo símbolo `'+'`, representa a Tônica (Fundamental) do modo em questão e deverá estar instanciado. O valor de saída, indicado pelo símbolo `'-'`, representa a dominante a ser encontrada, por isso deverá ser um parâmetro não instanciado.

As duas linhas seguintes no código apresentam a declaração `modeb`, indicando o corpo do predicado. Dessa forma, as regras produzidas poderão ter em seu corpo tanto a relação `grau5`, como `grau7`. A primeira delas poderá ser estabelecida no processo indutivo para adicionar o quinto grau ao corpo das regras, definindo uma nova cláusula ao predicado `dominante`. Nesse caso, o valor do número de chamadas definido na primeira

linha determina que a quantidade de regras adicionadas seja limitada ao número de 2. Analogamente, o sétimo grau poderá ser incorporado no predicado `dominante` seguindo o mesmo princípio para sua elaboração.

### 6.3.4 Especificação de Tipos

Nas especificações `modeh` ou `modeb`, o argumento referente ao *modo do predicado* determina o tipo dos elementos utilizados na sua formação. Esses tipos são declarados por meio de fatos que indicam os componentes simbólicos apropriados para a composição da hipótese. Esses fatos são adicionados à base de conhecimento prévio e no caso atual, constituem a especificação dos graus da escala:

```
grau('C').  grau('D').  grau('E').  grau('F').
grau('G').  grau('A').  grau('B').
grau('Cs'). grau('Ds'). grau('Fs'). grau('Gs').
grau('As'). grau('Db'). grau('Eb'). grau('Gb').
grau('Ab'). grau('Bb').
```

### 6.3.5 Especificação de Determinações

No processo de construção da hipótese, alguns predicados serão utilizados, os quais devem ser indicados pela declaração *determination*. Esses predicados são analisados na busca por generalizações, a partir da base de exemplos observados nos arquivos `conceito_dominante.f` e `conceito_dominante.n`. Na indução do conceito de grau dominante, essas declarações assumem a seguinte forma:

```
:- determination(dominante/2,grau5/2).
:- determination(dominante/2,grau7/2).
```

Nessas declarações são especificados dois argumentos, o primeiro indica o predicado alvo e sua aridade (número de argumentos), esse predicado será utilizado na cabeça da regra induzida. O segundo argumento consiste do nome e aridade dos predicados que poderão ser utilizados para compor o corpo do predicado alvo.

### 6.3.6 Exemplos Positivos

O arquivo `conceito_dominante.f` contém exemplos (amostras) que assumem o valor verdadeiro para o conceito a ser aprendido. Esses exemplos são descritos na forma de fatos em Prolog, formados por relações do tipo `dominante(X,Y)`. Desse modo, constituem o ponto de partida para a indução das regras. No caso em questão, as seguintes cláusulas são suficientes para o aprendizado:

```
dominante('G','C').
dominante('B','C').
dominante('A','D').
dominante('Cs','D').
```

### 6.3.7 Exemplos Negativos

As cláusulas pertencentes ao arquivo `conceito_dominante.n` representam negações do valor verdadeiro para o predicado `dominante(X,Y)`, ou seja `X` não é um grau dominante de `Y`. Para isso, as seguintes cláusula estão sendo utilizadas:

```
dominante('E','C').
dominante('G','D').
```

# 7 Terceira Contribuição

*Um programador que subconscientemente se vê como um artista,*
*vai desfrutar daquilo que faz e vai fazê-lo melhor.*

---

Donald E. Knuth (1938 - )

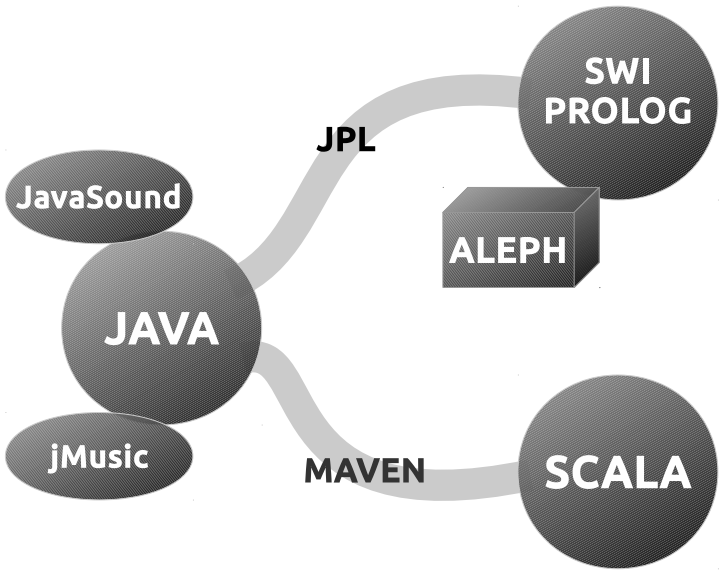
## 7.1 Fraseado - Um Sistema Musical Baseado em Conhecimento

Como forma de aplicação dos conceitos apresentados, foi desenvolvido um sistema voltado à manipulação de estruturas musicais - o sistema Fraseado.

Fraseado é um software musical baseado em conhecimento que utiliza uma abordagem multiparadigma na representação musical. O processo de representação de conhecimento é aplicado por meio de métodos declarativos (funcionais e lógicos). Além disso, o sistema realiza a aquisição de conhecimento fazendo uso da técnica de Programação Lógica Indutiva.

A Figura 8 apresenta a conexão dos módulos a partir do qual o sistema foi desenvolvido. Esses módulos constituem um arcabouço de programação que possibilita a aplicação das técnicas computacionais tratadas no presente trabalho.

Figura 8 – Módulos de programação no sistema Fraseado.




Desse modo, Fraseado constitui uma integração de variados conceitos computacionais, os quais têm sido consolidados no transcorrer das últimas décadas. Durante o processo de software envolvendo seu desenvolvimento, foram investigados elementos característicos à abordagem adotada:

- Simplicidade de programação e tempo de familiarização reduzido, por ser baseado em conceitos tradicionais de programação;
- Utilização de ferramentas de software livre consolidadas e facilmente acessíveis;
- Disponibilização da totalidade dos conceitos e componentes inerentes aos paradigmas de programação adotados;
- Representação de Conhecimento Musical por meio da Lógica de Primeira Ordem;
- Aquisição de Conhecimento por meio da Programação Lógica Indutiva;
- Interação sensorial com o usuário por meio de artefatos visuais e sonoros.

Na utilização da abordagem adotada, verificou-se que todas as vantagens decorrentes de cada técnica específica tornam-se disponíveis por meio da integração das ferramentas de programação. A seguir, serão apresentados aspectos relativos ao desenvolvimento do sistema, bem como sua aplicação.

## 7.2 Representação Musical

Como forma de representação musical, utilizou-se a canção Eleanor Rigby, uma composição creditada a Paul McCartney e John Lennon e lançada pelos Beatles em 1966 (LENNON; MCCARTNEY; WHITCOMB, 1966).

Esta canção se caracteriza por realizar uma mistura modal, onde os modos Eólio e Dórico se alternam. Tendo como base a progressão de acordes Em-C, a melodia passa pelos graus 3, 6 e 7 da escala, resolvendo na tônica. Com isso, a composição apresenta de forma simples e sofisticada, uma interessante sensação harmônica, fornecendo um nível de urgência ao humor melódico e finalizando o fluxo musical com um ar de inevitabilidade (PEDLER, 2010).

A implementação pelo sistema Fraseado foi baseada na criação de 5 classes representando, além do módulo principal, os objetos para Baixo, Melodia, Cordas (harmonia ) e Percussão. A Listagem 7.1 apresenta a Classe principal.

Listagem 7.1 – Classe principal para Eleanor Rigby.

```
1  package fraseado10143dsv;
2
3  import jm.music.data.Score;
4  import jm.util.Play;
5  import jm.music.data.*;
6  import jm.util.*;
7
8  public class eleanorPrinc {
9    static Score partitura;
10
11   public static void main(String[] args) {
```

```
12      eleanorBaixo eleanorBaixo = new eleanorBaixo();
13      eleanorMelod eleanorMelod = new eleanorMelod();
14      eleanorPerc eleanorPerc = new eleanorPerc();
15      eleanorHarm eleanorHarm = new eleanorHarm();
16      eleanorCordas eleanorCordas = new eleanorCordas();
17
18      partitura = new Score(
19        "FRASEADO - Eleanor Rigby (Paul McCartney & John Lennon)", 124.0);
20
21      eleanorBaixo.produz(); eleanorMelod.produz();
22      eleanorPerc.produz();  eleanorHarm.produz();
23      eleanorCordas.produz();
24
25      partitura.addPart(eleanorBaixo.pegaBaixo());
26      partitura.addPart(eleanorMelod.pegaMelod());
27      partitura.addPart(eleanorPerc.pegaPerc());
28      partitura.addPart(eleanorHarm.pegaHarm());
29      partitura.addPart(eleanorCordas.pegaCordas());
30
31      View.notate(partitura);
32    }
33  }
```

Com relação à harmonia, foram criados vetores contendo notas que formam variações dos acordes básicos. Cada variação representa uma forma de inversão ou nota de reforço adicional. A Listagem 7.2 apresenta a classe referente à harmonia.

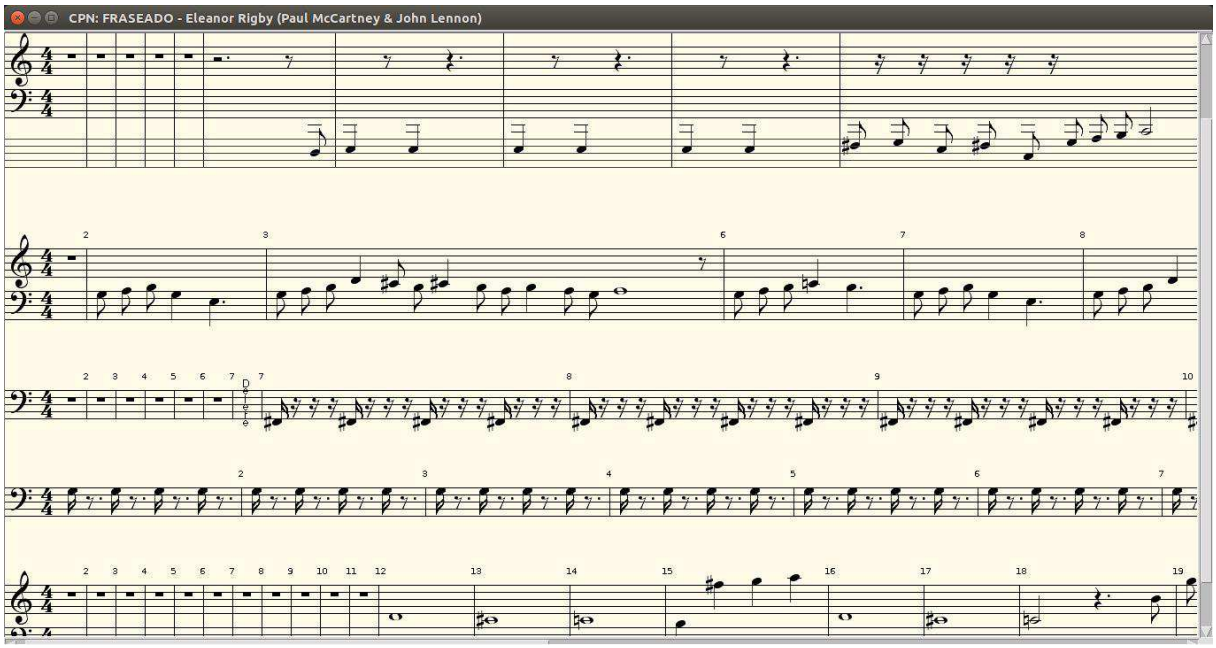
Listagem 7.2 – Classe com harmonia para Eleanor Rigby.

```
1  package fraseado10143dsv;
2
3  import jm.music.data.*;
4  import jm.JMC;
5  import jm.music.tools.Mod;
6
7  public class D10202eleanorHarm implements JMC{
8    private Part harm;
9    private CPhrase fHarm;
10
11   public D10202eleanorHarm() {
12         harm = new Part("Harmonia", SLOW_STRINGS, 2);
13         fHarm = new CPhrase(0.0);
14   }
15
16   public void produz() {
17     int[] pausa = {REST};              int[] acEm1 = {G3,B3,E4};
18     int[] acEm2 = {E2,G3,E4,B4,E5};    int[] acEm3 = {G3,E4,G4};
19     int[] acEm4 = {E2,G3,E4,B4,G4,E5}; int[] acC1  = {G3,C4,E4};
20     int[] acC2  = {E2,G3,E4,C4,E5};    int[] acC3  = {E2,G3,E4,C4,G4,E5};
21
22 // 1 Intro
23     fHarm.addChord(acEm1,SN); fHarm.addChord(pausa,DEN);
24
25 // ...
26
27 // 7 Father McKenzie
28     fHarm.addChord(acEm1,SN); fHarm.addChord(pausa,DEN);
29
30 // ...
31
32 // Parte 2 ------------------------------------------------
33
34 // 20 Ah, look at all... (1)
35         fHarm.addChord(acC3,SN); fHarm.addChord(pausa,DEN);
36
37 // ...
```

```
38
39          harm.addCPhrase(fHarm);
40          harm.setDynamic(50);
41    }
42
43    public Part pegaHarm() {
44          return harm;
45    }
46 }
```

A Figura 9 apresenta a partitura gerada pelo sistema (🔊).

Figura 9 – Partitura de Eleanor Rigby no sistema Fraseado.




## 7.3 Composição Algorítmica

Com relação à atividade de composição musical automática (Composição Algorítmica), o caráter lógico do sistema Fraseado possibilita a declaração de regras que definem progressões harmônicas. A aplicação de tais regras conduz ao estabelecimento de estruturas musicais que irão reger o desenvolvimento melódico.

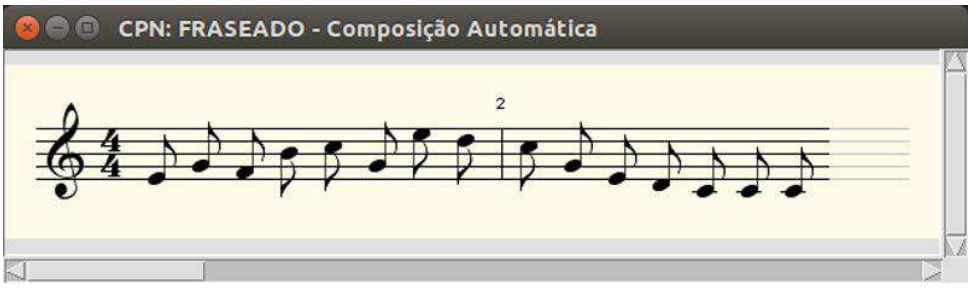
Considerando elementos básicos de harmonia funcional, a seguinte metodologia para composição possibilita a criação de estruturas harmônicas e a geração de melodias:

1. Representação de estruturas funcionais;
2. Aquisição de conhecimento;
3. Criação de grafo para progressões de acordes;
4. Seleção de um ou mais trechos harmônicos;

5. Aplicação de regras melódicas (intervalos, modos, saltos etc);
6. Produção da melodia;
7. Armazenamento, reprodução e/ou visualização do trecho composto.

A Figura 10 apresenta uma melodia criada de acordo com essa metodologia (🔊).

Figura 10 – Composição gerada pelo sistema Fraseado .




A Listagem 7.3 apresenta trecho do código para criação do grafo de progressões.

Listagem 7.3 – Progressões harmônicas para composição musical.

```
1  aresta(a,b).
2  aresta(a,c).
3  aresta(c,d).
4  aresta(c,e).
5  aresta(f,e).
6
7  inicial(a).
8  inicial(f).
9
10 terminal(b).
11 terminal(d).
12 terminal(e).
13
14 caminho(A,B,Caminho) :-
15   inicial(A),
16   terminal(B),
17   trajeto(A,B,[A],Q),
18   reverse(Q,Caminho).
19
20 trajeto(A,B,C,[B|C]) :-
21   aresta(A,B).
22
23 trajeto(A,B,Visitado,Caminho) :-
24         aresta(A,C),
25         C \== B,
26         \+member(C,Visitado),
27         trajeto(C,B,[C|Visitado],Caminho).
28
29 gera(Resp) :-
30   findall( C, caminho(_,_,C), L),
31   random_select( Selec, L, _ ),
32   write('Seq: '), writeln(Selec),
33   maplist(call, Selec, R1),
34   append(R1,Resp).
35
36 a([1,3,2,4]).
37 b([1,1,1]).
38 c([3,5,4,6]).
39 d([1,1,1]).
40 e([3,2,1]).
```

## 7.4 Gráficos

Além da visão de partitura, a visão *piano roll* e o gráfico de histograma de notas podem ser apresentados. As Figuras 11 e 12 apresentam esses componentes para a composição *Judas Maccabaeus*, apresentada na seção 5.5.

Figura 11 – Judas Maccabaeus - Visão Piano Roll.

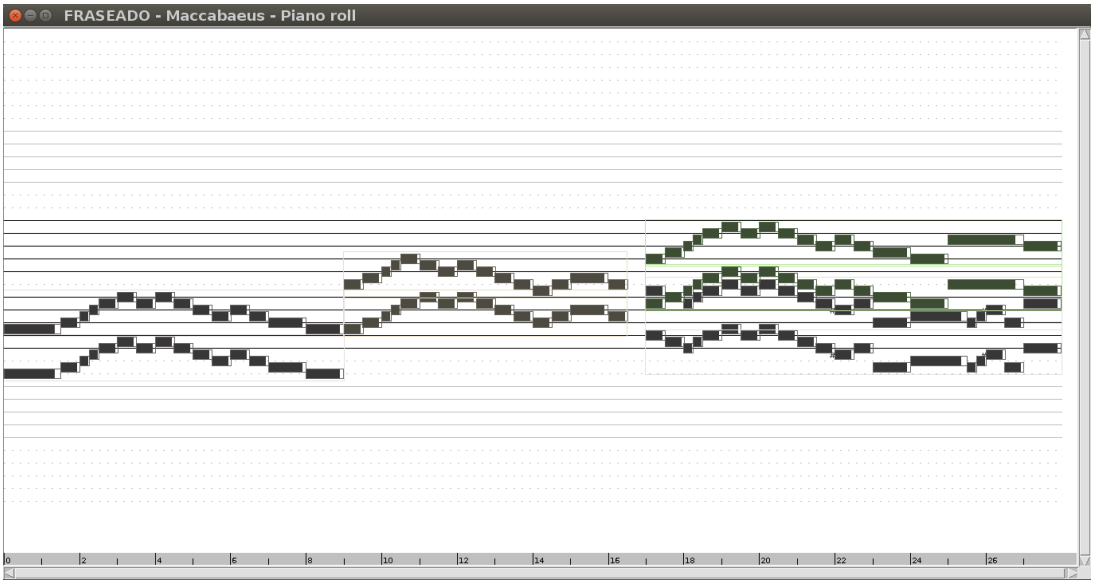
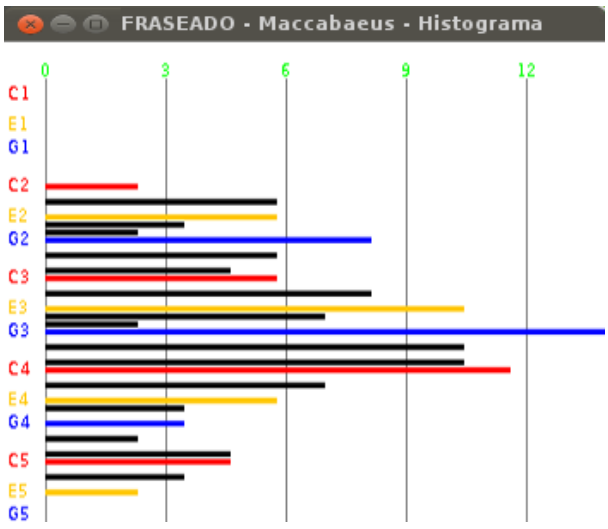



Figura 12 – Judas Maccabaeus - Histograma.



# 8 Quarta Contribuição

*A questão não é se os computadores possuem alma,*
*mas se nós possuímos uma.*

DAVID COPE: (1941 - )

Dentre as aplicações decorrentes do processo de Representação de Conhecimento Musical, a Composição Algorítmica (CA) caracteriza-se por ser uma atividade transdisciplinar, envolvendo elementos tanto cognitivos, como relacionados a técnicas de programação. Historicamente, seu principal campo de pesquisa concentra-se no domínio da Inteligência Artificial, no que concerne a aspectos como aprendizado de máquina e raciocínio automático.

As motivações que têm conduzido ao desenvolvimento de sistemas de composição algorítmica podem ser elencadas de acordo com alguns aspectos. Dentre eles, destacam-se o interesse de um compositor em criar programas como um modo de expressão de seu processo particular de criação. Um outro fator refere-se ao desenvolvimento de programas que sirvam como ferramentas gerais auxiliando outros compositores durante esse processo. Há ainda o interesse em implementar teorias referentes a estilos musicais específicos. Pode-se ainda elencar a motivação no sentido de se codificar teorias referentes aos processos cognitivos associados à composição (PEARCE; MEREDITH; WIGGINS, 2002). A Tabela 12 caracteriza essas motivações, indicando seus domínios e as atividades relacionadas.

Tabela 12 – Motivações para o desenvolvimento de programas de Composição Algorítmica.

| Domínio | Atividade | Classificação |
|---|---|---|
| Composição | Composição algorítmica | Expansão do repertório composicional |
| Engenharia de software | Projeto de ferramentas para composição | Desenvolvimento de ferramentas para compositores |
| Musicologia | Modelagem computacional de estilos musicais | Proposta e avaliação de teorias para estilos musicais |
| Ciências cognitivas | Modelagem computacional de cognição musical | Proposta e avaliação de teorias cognitivas para composição musical |

Fonte: (PEARCE; MEREDITH; WIGGINS, 2002).

O sistema Fraseado realiza a composição algorítmica tendo o campo da programação (Engenharia de Software) como principal motivação. Nesse sentido, a abordagem apresentada - realizando a Representação de Conhecimento Musical - fornece subsídios

para o desenvolvimento de sistemas que, entre outras atividades, tenham sua aplicação voltada à composição musical automática.

Como forma de avaliação dessa abordagem, será proposto o *Teste de Turing Expandido*, aplicado à Computação Musical. Essa proposta parte do princípio fundamental estabelecido pelo Teste de Turing clássico (TT) (TURING, 1950) e realiza um levantamento com o intuito de obter informações adicionais, estudando características na percepção de composições geradas pelo computador. Tais informações serão analisadas no sentido de investigar aspectos produzidos pela composição no ouvinte humano.

## 8.1 Como Avaliar um Sistema de CA?

Desde as origens da computação moderna, algumas questões têm sido levantadas no sentido de tecer avaliações sobre um sistema de software:

- O sistema se comporta como um ser humano?
- É possível modelar a inteligência por meio do computador?
- Como avaliar conceitos como criatividade e arte?
- O computador apresenta comportamento inteligente?

A resposta a essas questões é dependente da definição utilizada para expressões como *inteligência*, *pensamento* e *criatividade*. Com o decorrer do desenvolvimento da área computacional, questões mais objetivas têm sido abordadas. Tais questões lidam com aspectos como eficiência, segurança, portabilidade e otimização de recursos.

Com relação aos paradigmas de programação, questões como clareza, abstração de dados, reusabilidade, polimorfismo e expressividade têm sido tratadas. Em Engenharia de Software, o conceito de qualidade relaciona-se a um conjunto de características intrínsecas a um produto, processo ou sistema, sendo avaliado o grau em que esses elementos atendem aos objetivos inicialmente estipulados. Nesse caso, são abordadas questões como satisfação do usuário e conformidade quanto a requisitos (SANTOS et al., 2008).

### 8.1.1 Problemática Envolvida

Collins destaca a existência de uma série de críticas com relação à abrangência da Inteligência Artificial, a maioria deles antecipada por Turing em seu artigo original. Esses críticas referem-se à impossibilidade de um computador comportar-se como um ser humano em aspectos sensoriais, emocionais, intuitivos etc (COLLINS, 2006).

Conforme Souza e Faria apontam, uma interpretação incorreta com relação aos componentes de computabilidade e validação de um sistema pode conduzir a argumentos falaciosos quanto ao potencial de um sistema (SOUZA; FARIA, 2011). Para tanto, seria necessário o estabelecimento do conceito de criatividade a partir de uma definição formal, efetiva e sem ambiguidades. Com isso, torna-se impossível realizar uma avaliação de softwares que pertençam a essa categoria, que venha a ser completamente rigorosa e livre de controvérsias (ARIZA, 2013).

Adicionalmente, Pearce et al., apontam uma lacuna existente em grande parte dos trabalhos publicados envolvendo composição algorítmica. Tal lacuna refere-se a três aspectos (PEARCE; MEREDITH; WIGGINS, 2002):

1. Especificação imprecisa dos objetivos práticos ou teóricos da investigação;
2. Uso de uma metodologia inadequada para atingir esses objetivos;
3. Falta de um modelo para avaliação de resultados que seja controlado, mensurável e repetível.

Ariza afirma que a utilização do chamado "Teste de Turing Musical" como forma de validação de um sistema de CA, somente faz sentido se as aspirações do sistema são voltadas à imitação do processo criativo, sem o objetivo de que ele seja realmente criativo, ou possa criar uma obra de arte completamente inovadora (ARIZA, 2009).

### 8.1.2 Definindo o Objetivo da Investigação

Berrar e Konagaya apresentam as limitações quanto à avaliação de um sistema no que se refere a conceitos como "inteligência", "pensamento" ou "criatividade" por sua dificuldade de compreensão e consenso. Com isso, indicam a utilização de testes no estilo de Turing para o campo da música criada artificialmente. Nesse sentido, destacam a importância do TT com o intuito de motivar o desenvolvimento de formas alternativas de inteligência, diferentes do modo humano de pensar (BERRAR; KONAGAYA; SCHUSTER, 2013).

Conforme Bishop et al. afirmam, o TT está "comportamentalmente" consolidado por meio de exemplos não-interativos como *AARON* e *Emmy*. Isso não implica obrigatoriamente na capacidade artística genuína ou criatividade pelos computadores - pois tal discussão depende de argumentos filosóficos altamente discutíveis. Entretanto, este fato constata características da performance do sistema computadorizado, que foi o foco principal do trabalho de Turing (BISHOP; BODEN, 2010).

Kosteletos e Georgaki propõem um novo escopo para o uso do TT, não como um critério de avaliação de inteligência, mas como um "instrumento" para traçar certas

características sobre o julgamento humano em vários campos. Desse modo, testes no estilo de Turing estabelecem um procedimento no qual "o que é julgado passa a ser o próprio julgamento" (KOSTELETOS; GEORGAKI, ).

Com relação à importância da coleta de dados adicionais, Bown mostra que resultados positivos podem ser alcançados por meio do TT, ou seja, um sistema eficiente pode passar por humano. Entretanto a maior riqueza de elementos consiste na avaliação dos artefatos cognitivos e sensoriais produzidos pelo teste. Com esse intuito, este pesquisador aplicou um teste em que o aspecto computacional do sistema não foi ocultado. Em vez disso, o estudo analisou questões de engajamento, experiência e percepção em uma interação improvisada, analisando as reações produzidas nos participantes (BOWN, 2015).

### 8.1.3 A Imitação da Imitação

Como uma analogia com relação ao processo criativo, podemos observar no campo musical, as composições "Bachianas Brasileiras" de Heitor Villa-Lobos e "Bachianinha nº 1" de Paulinho Nogueira. As primeiras são um conjunto de nove composições, escritas entre 1930 e 1945, nas quais utiliza-se o estilo barroco, tomando-se composições de Johann Sebastian Bach como referência (LATHAM, 2004).

Conforme Felice apresenta, Villa-Lobos utilizou diversas influências externas em sua obra, valendo-se de composições modernistas da Europa, sendo que este intercâmbio técnico e estético não demonstra falta de originalidade (FELICE, 2016). Felice afirma ainda que as Bachianas Brasileiras constituem uma obra que se utiliza de "citações barrocas com uma apropriação neoclássica e, portanto, inclui conteúdos tradicionais europeus e ferramentas composicionais de vanguarda, ao mesmo tempo em que busca uma sonoridade nacional", sendo isso realizado sem detrimento ao espírito engenhoso e grande criatividade do autor brasileiro.

Apesar de contemporâneo de Villa-Lobos, Paulinho Nogueira não o conheceu pessoalmente, entretanto a composição desse grande músico teve em Villa-Lobos sua grande inspiração, maior até do que no próprio Bach. Nogueira era grande apreciador da "releitura" feita por Villa-Lobos da obra de Bach, principalmente por se tratar de um compositor comprometido com as terras brasileiras. Foi, inclusive, por meio de Villa-Lobos que Paulinho Nogueira entrou em maior contato com a obra de Bach (NOGUEIRA, 2017).

Diante disso, torna-se evidente o aspecto fundamentalmente subjetivo do que possa ser entendido por "criatividade", no que tange aos seus níveis de manifestação e formas de apreciação ou julgamento. Nesse ponto, Berrar e Schuster levantam a questão: "É realmente tão importante que um programa seja genuinamente criativo?". Pois enquanto nos atemos a determinados tipos de discussão, estão sendo produzidas belas pinturas, histórias comoventes são elaboradas, anedotas são criadas e movimentos brilhantes de

xadrez têm sido desenvolvidos (BERRAR; SCHUSTER, 2014).

## 8.2 Proposta para o Teste de Turing Expandido

O teste de Turing aplicado a composições musicais busca determinar se uma pessoa consegue distinguir se um trecho musical foi composto por um computador ou por um ser humano (HIRAGA et al., 2004). Diante das questões abrangidas na avaliação de um sistema de CA, o "Teste de Turig Expandido" (TTE) concentra-se em identificar aspectos de contextualização de uma composição gerada automaticamente, de acordo com um ou mais públicos-alvo específicos.

Com esse intuito, apresenta-se uma proposta de avaliação que, a partir do teste clássico, busca a realização de uma coleta de informações, no sentido de identificar aspectos relacionados aos efeitos que a composição possa produzir no ouvinte humano.

Enquanto a seção anterior apresentou um breve panorama da problemática envolvida, a presente proposta não tem o objetivo de aprofundar-se nas questões filosóficas levantadas. Além disso, ela não define um trabalho conclusivo, mas consiste em uma concepção que visa fornecer elementos de apoio na avaliação desse tipo de sistema.

### 8.2.1 Caracterização do Teste

O teste realiza a coleta de dados a partir de parâmetros qualitativos, buscando identificar elementos despertados nos ouvintes com relação a:

- Memórias;
- Emoções;
- Associações culturais;
- Associações históricas;
- Preferências estéticas;
- Utilização em contextos específicos.

Nesse sentido, o foco da avaliação consiste em identificar se a composição produzida adéqua-se a determinados sistemas musicais (por exemplo, música ocidental tradicional). O modo de apresentação musical deve ser definido, seja a composição propriamente dita, ou componentes como performance ou interação. De acordo com o TT clássico, é importante que juntamente com a composição automática, seja apresentada uma ou mais composições humanas. Além disso, o público-alvo deve ser previamente estabelecido, podendo ser formado por participantes que atuem profissionalmente na área, ou que possuam conhecimento musical especializado, público geral, ou ainda outros tipos de categorização.

A seguir, apresentam-se aspectos a serem levantados na pesquisa. Alguns deles se repetem por caracterizarem pontos de vista alternativos sobre os elementos analisados.

### 8.2.2 Abordagem

Quanto à abordagem dois tipos podem ser utilizados: Direcionada e Espontânea. Na abordagem Direcionada, busca-se despertar elementos pré-definidos no participante, o que não ocorre na abordagem Espontânea.

**Abordagem Direcionada:**

- Composições buscando transmitir sensações específicas, como tranquilidade ou estranheza; ou que sejam propositalmente indiferentes;
- Identificação de sentimentos despertados;
- Utilização de estilos pré-definidos.

**Abordagem Espontânea:**

- Memórias despertadas;
- Identificação de contextos em que a composição se aplique;
- Componentes históricos associados.

### 8.2.3 Tipos de Dados

Quanto aos dados, estes podem ser tanto Objetivos como Subjetivos. Nos dados Objetivos, as respostas são fornecidas aos participantes, que de acordo com a situação, pode escolher por uma, várias ou nenhuma resposta. Nos dados Subjetivos, não são fornecidas alternativas de respostas, as quais ficam integralmente a critério do participante.

**Dados Objetivos:**

- Identificação com composições já ouvidas;
- Origem da composição Humano X Computador (TT clássico);
- Identificação de preferência quanto à composição.

**Dados Subjetivos:**

- Identificação cultural;
- Identificação cronológica.

### 8.2.4 Aspectos Associativos, Conceituais e Experienciais

Com relação aos aspectos Associativos, Conceituais e Experienciais, busca-se identificar elementos pré-estabelecidos nos participantes. Adicionalmente realiza-se a observação de sua experiência ao ouvir os trechos, além de buscar associações feitas.

**Aspectos Associativos:**

- Identificação com composições já ouvidas;
- Identificação cultural;
- Identificação cronológica;
- Origem da composição Humano X Computador (TT clássico).

**Aspectos Conceituais**

- Identificação de contextos em que a composição se aplique;
- Componentes históricos associados;
- Identificação cronológica;
- Associação de estilo.

**Aspectos Experienciais**

- Memórias despertadas;
- Sentimentos despertados;
- Preferência quanto à composição.

### 8.2.5 Questões a serem Aplicadas

Considerando os aspectos apresentados nas seções anteriores, propõe-se a utilização das seguintes questões:

1. Você já ouviu alguma música com estilo semelhante?
2. Você associa a composição à cultura de algum país? Cite em caso afirmativo.
3. Há alguma situação em que músicas nesse estilo poderiam ser tocadas? Cite em caso afirmativo.
4. O trecho lhe traz algum tipo de memória? Cite em caso afirmativo.
5. Você pode associar o trecho a algum evento histórico? Cite em caso afirmativo.
6. Quais sentimentos lhe despertaram ao ouvir o trecho?
7. Em um período de quanto tempo atrás o trecho se enquadra melhor?
8. Marque um ou mais estilos musicais que você associa ao trecho.
9. Você gostaria de ouvir músicas no estilo do trecho apresentado?
10. O trecho foi composto por um ser humano ou um computador?

### 8.2.6 Utilização de Dados Exploratórios

A partir dos dados coletados no teste, a exploração realizada contribui na identificação de aspectos concernentes às avaliações dos participantes. Além dos elementos apresentados nas seções anteriores, medidas estatísticas podem ser utilizadas nessa exploração:

- **Medidas de Dispersão:** Média, Variância, Desvio Padrão e Curtose;
- **Medidas de Inter-relação:** Correlação, Covariância e Teste-F.

Essa exploração busca identificar, a partir dos valores qualitativos, se as reações apresentadas pelas pessoas ao ouvir o trecho do computador possuem características que se enquadram no padrão dos trechos compostos por autores humanos. Busca-se com isso verificar o grau de uniformidade na percepção do trecho automático com relação aos outros.

## 8.3 Aplicação do Teste

O Teste de Turing Expandido foi aplicado entre os dias 21 a 29 de dezembro de 2016 nas dependências do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus São Roque. A aplicação sob o acompanhamento da Coordenadoria de Apoio ao Ensino e em parceria com o Núcleo Interdisciplinar de Produção e Pesquisa Audiovisuais do campus.

Houve a participação anônima de 237 voluntários pertencentes às comunidades interna e externa ao campus, categorizados em 3 tipos, de acordo com o grau de envolvimento com a área musical, com um número proporcional entre cada categoria de aproximadamente 4 para 1.

- **PROF** - Atuam na área musical (10 participantes);
- **ESP** - Possuem conhecimento musical especializado (39 participantes);
- **GER** - Não possuem conhecimento especializado em música (188 participantes).

Foram utilizados trechos de 3 composições, sendo uma delas gerada pelo sistema Fraseado, e duas composições clássicas. Os trechos são os seguintes:

- **Trecho 1 (T1) :** Uirapuru - Heitor Villa-Lobos (🔊)
- **Trecho 2 (T2) :** Cálculo Ametista - Composição automática (🔊)
- **Trecho 3 (T3) :** Overture - Piotr Ilitch Tchaikovsky (🔊)

Os trechos musicais foram previamente gravados utilizando o seguinte equipamento:

- **Sistema de Captação:** Hexafônico
- **Controlador MIDI:** Roland GK-3
- **Sistema de sintetização:** Sintetizador Roland GR-55

O Apêndice A apresenta uma descrição desse equipamento.

Com relação às composições humanas, o primeiro trecho (T1) foi selecionado sem a intenção de um motivo melódico específico, enquanto que no segundo (T3), buscou-se causar a sensação de "estranheza".

Para o trecho produzido pelo sistema Fraseado (T2), realizou-se a seleção de 3 frases para as partes referentes ao início, meio e fim da composição, buscando atender a 2 critérios: a) adequação ao padrão clássico tradicional; b) motivo melódico associado ao sentimento de "tranquilidade".

Devido à grande quantidade de informações coletadas, os dados abaixo não serão apresentados nesse trabalho:

- Associação de aspectos culturais;
- Contextos de aplicação da composição;
- Associação histórica;
- Associação por período de tempo;
- Associação por estilo musical.

Os resultados obtidos com a aplicação do teste serão apresentados no próximo capítulo.

# 9 Resultados e Considerações Finais

*Finalmente, eu olho para o futuro, para o momento em que o computador será o próprio compositor.*

MAX MATHEWS: (1926 - 2011)

Neste capítulo apresentam-se elementos coletados pela aplicação do Teste de Turing Expandido. Inicialmente, será realizada uma exploração sobre os dados e em seguida serão feitas considerações baseadas nos resultados.

## 9.1 Apresentação Exploratória de Dados

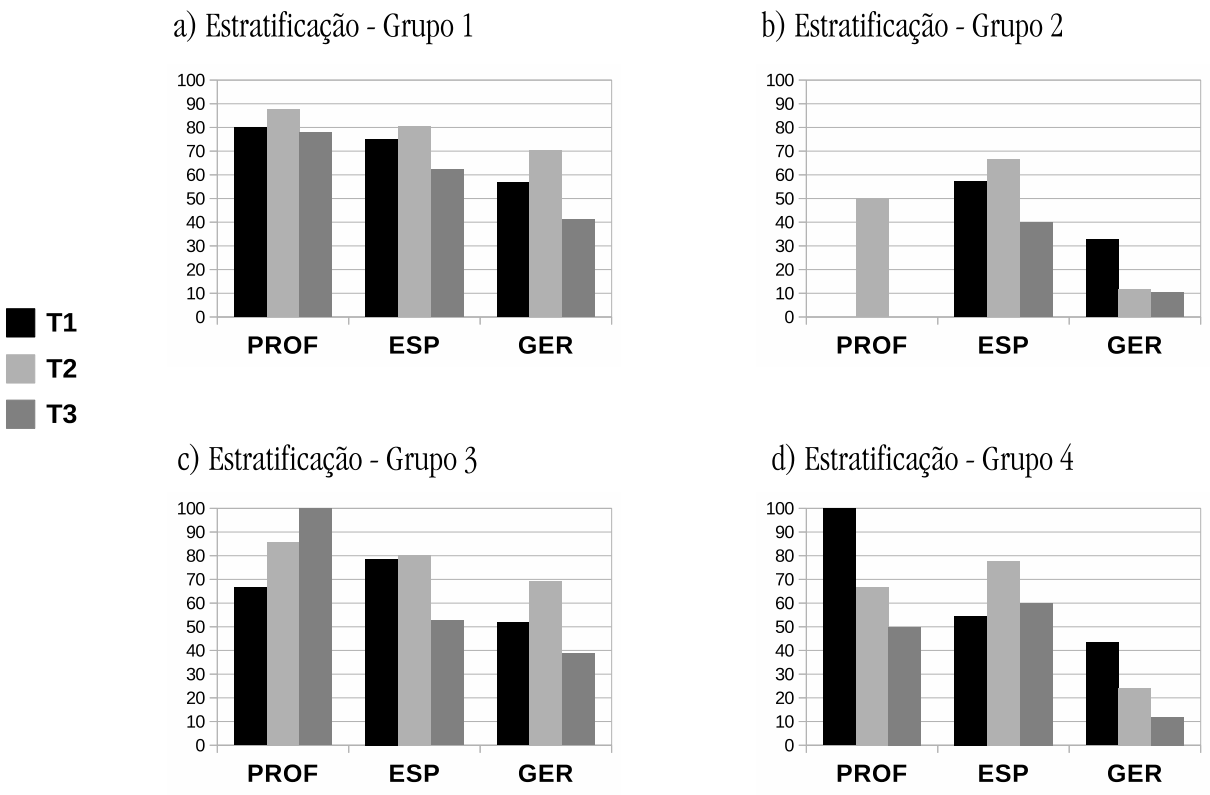
Para a realização desta análise, os conjuntos de dados foram estratificados a partir dos valores objetivos coletados. Buscou-se compreender aspectos relativos aos participantes que **gostariam de ouvir** trechos semelhantes aos apresentados. Desse modo, o conjunto-alvo consiste nos grupos de participantes respondendo afirmativamente à questão 9 "*Você ouviria músicas no estilo do trecho apresentado?*".

A estratificação foi realizada partindo-se dos conjuntos gerados pelas questões 1 "*Você já ouviu alguma música com estilo semelhante?*" e 10 "*O trecho foi composto por um ser humano ou um computador?*". Com isso foram gerados 4 conjuntos de origem, tomados como base para os seguintes grupos utilizados nesta análise:

- **G1:** Já ouviram e gostariam de ouvir trechos semelhantes.
- **G2:** Não ouviram e gostariam de ouvir trechos semelhantes.
- **G3:** Trecho composto por humano e gostariam de ouvir trechos semelhantes.
- **G4:** Trecho composto por computador e gostariam de ouvir trechos semelhantes.

A Figura 13 apresenta os grupos que gostariam de ouvir os trechos compostos, categorizados por participantes de acordo com os grupos apresentados na seção 8.3.

Figura 13 – Participantes que gostariam de ouvir trechos semelhantes aos apresentados.




### 9.1.1 Exploração de Dispersão

A exploração de dispersão foi realizada a partir das seguintes medidas: Média, Variância, Desvio padrão e Curtose. A Tabela 13 indica esses valores absolutos, enquanto a Figura 14 apresenta os respectivos gráficos.

Tabela 13 – Medidas de dispersão.

(a) Média

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF | 82,22 | 72,47 | 56,94 |
| ESP | 66,31 | 76,25 | 53,67 |
| GER | 46,23 | 43,80 | 25,59 |

(b) Variância

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF | 187,65 | 234,97 | 1394,67 |
| ESP | 112,07 | 31,69 | 74,63 |
| GER | 83,86 | 694,41 | 209,65 |

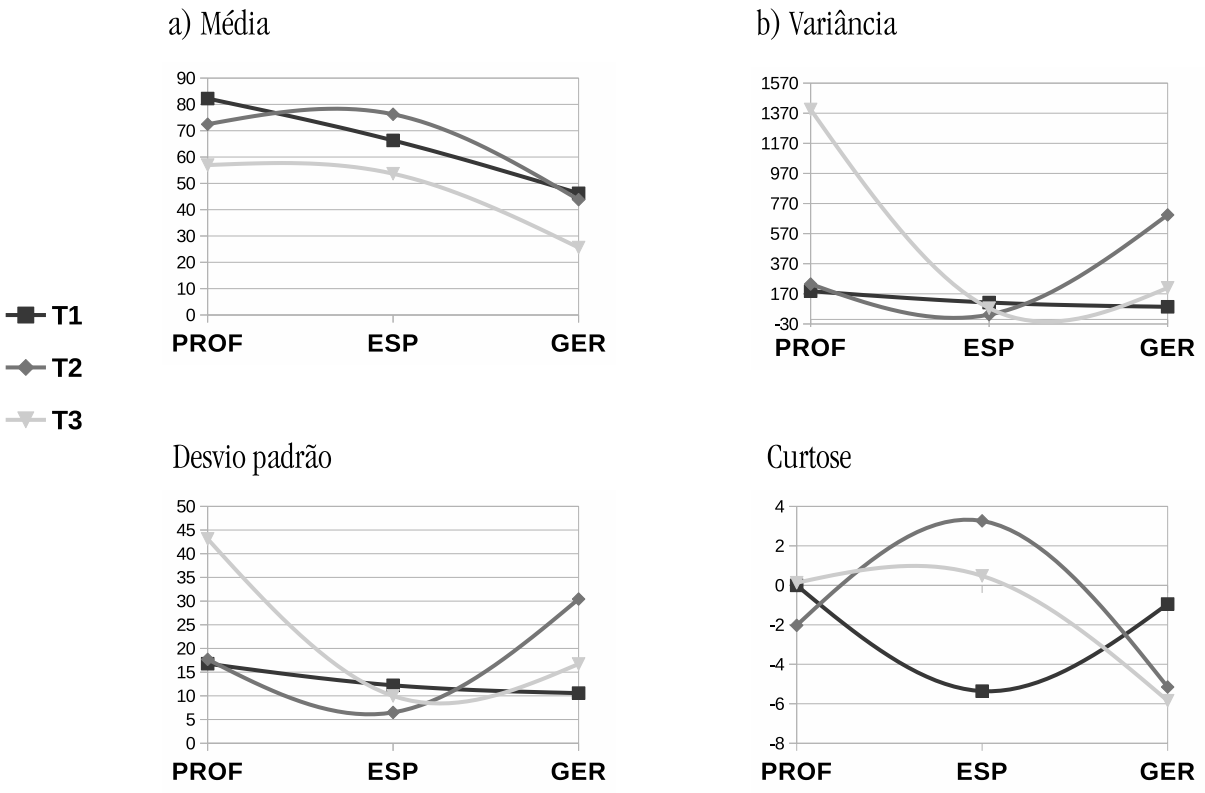
(c) Desvio Padrão

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF | 16,77 | 17,70 | 43,12 |
| ESP | 12,22 | 6,50 | 9,97 |
| GER | 10,57 | 30,42 | 16,71 |

(d) Curtose

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF | 0,00 | -2,02 | 0,13 |
| ESP | -5,36 | 3,26 | 0,48 |
| GER | -0,94 | -5,15 | -5,84 |

Figura 14 – Gráficos de dispersão.




### 9.1.2 Exploração de Inter-relação

A exploração de inter-relação foi realizada considerando 2 aspectos:

- **Aspecto 1:** Comparando as categorias de participantes (PROF, ESP, GER).
- **Aspecto 2:** Comparando os trechos musicais avaliados (T1, T2, T3)

Sobre cada um desses conjuntos foram aplicadas as medidas de Correlação de Pearson, Covariância e o Teste-F.

As Tabelas 14 e 15 apresentam esses valores.

Tabela 14 – Medidas de inter-relação entre categorias de participantes (Aspecto 1).

(a) Correlação de Pearson

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF x ESP | -0,96 | 0,92 | 0,67 |
| PROF x GER | -0,70 | 0,97 | 0,85 |
| ESP x GER | 0,82 | 0,82 | 0,47 |

(b) Covariância

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF x ESP | -139,73 | 80,21 | 216,18 |
| PROF x GER | -53,86 | 393,63 | 463,09 |
| ESP x GER | 79,67 | 122,17 | 59,64 |

(c) Teste-F

| | T1 | T2 | T3 |
|---|---|---|---|
| PROF x ESP | 0,59 | 0,13 | 0,03 |
| PROF x GER | 0,45 | 0,39 | 0,39 |
| ESP x GER | 0,81 | 0,03 | 0,03 |

Tabela 15 – Medidas de inter-relação entre os trechos musicais (Aspecto 2).

(a) Correlação de Pearson

| | PROF | ESP | GER |
|---|---|---|---|
| T1 x T2 | -0,88 | 0,64 | 0,95 |
| T1 x T3 | -0,99 | 0,30 | 0,91 |
| T2 x T3 | 0,96 | 0,88 | 0,98 |

(b) Covariância

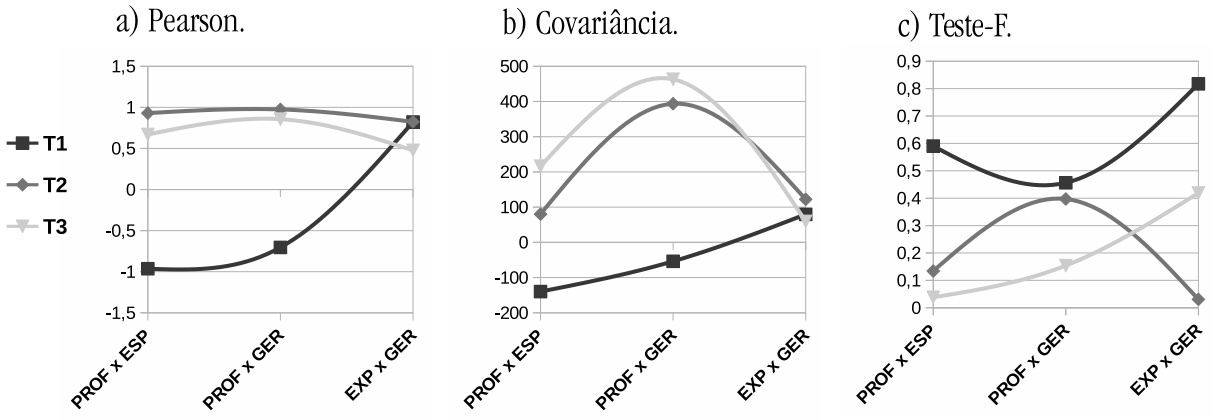
| | PROF | ESP | GER |
|---|---|---|---|
| T1 x T2 | -114,19 | 38,31 | 229,28 |
| T1 x T3 | -279,83 | 27,77 | 121,28 |
| T2 x T3 | 550,80 | 43,23 | 377,66 |

(c) Teste-F

| | PROF | ESP | GER |
|---|---|---|---|
| T1 x T2 | 1,01 | 0,32 | 0,11 |
| T1 x T3 | 0,26 | 0,74 | 0,47 |
| T2 x T3 | 0,17 | 0,50 | 0,35 |

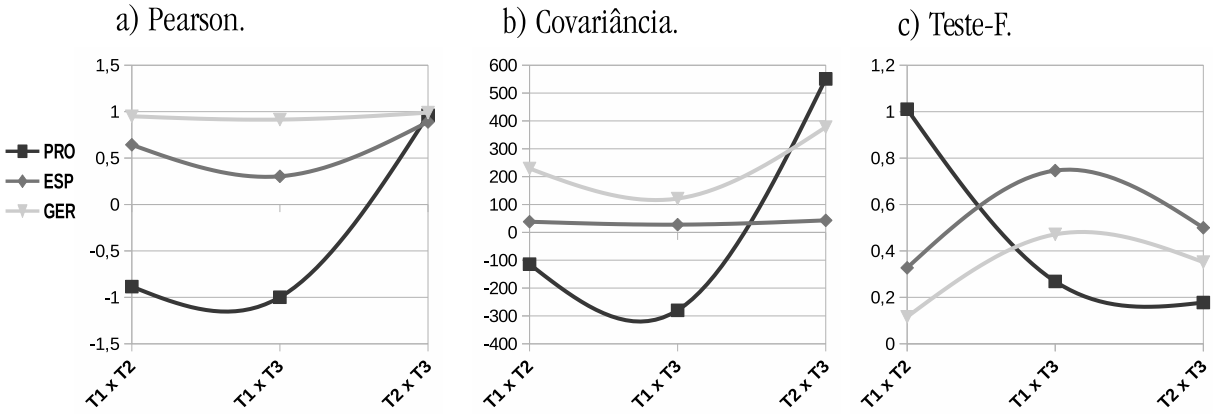
Com relação ao Aspecto 1, a Figura 15 apresenta os gráficos referentes aos valores absolutos.

Figura 15 – Gráficos de inter-relação entre categorias de participantes (Aspecto 1).




Com relação ao Aspecto 2, a Figura 16 apresenta os gráficos referentes aos valores absolutos.

Figura 16 – Gráficos de inter-relação entre os trechos musicais (Aspecto 2).

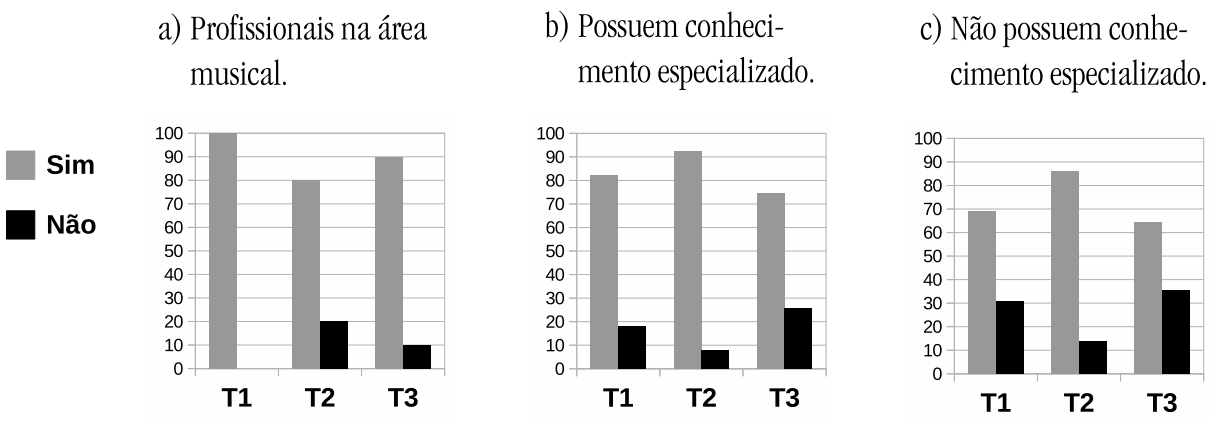



## 9.2 Associações, Sentimentos e Memórias Despertados

### 9.2.1 Associação por estilo musical

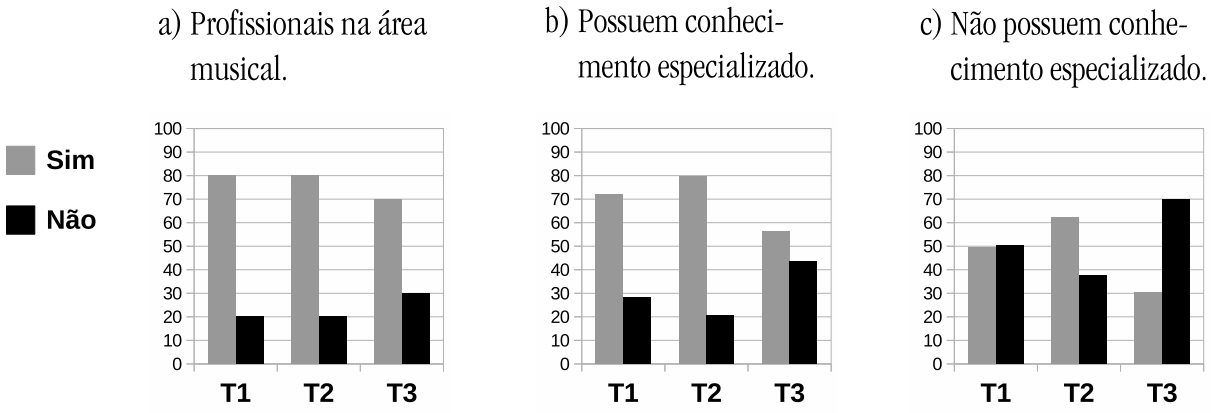
A Figura 17 apresenta participantes que já ouviram trechos semelhantes aos apresentados.

Figura 17 – Já ouviu alguma música com estilo semelhante.




A Figura 18 apresenta os participantes que informaram que ouviriam trechos semelhantes aos apresentados.

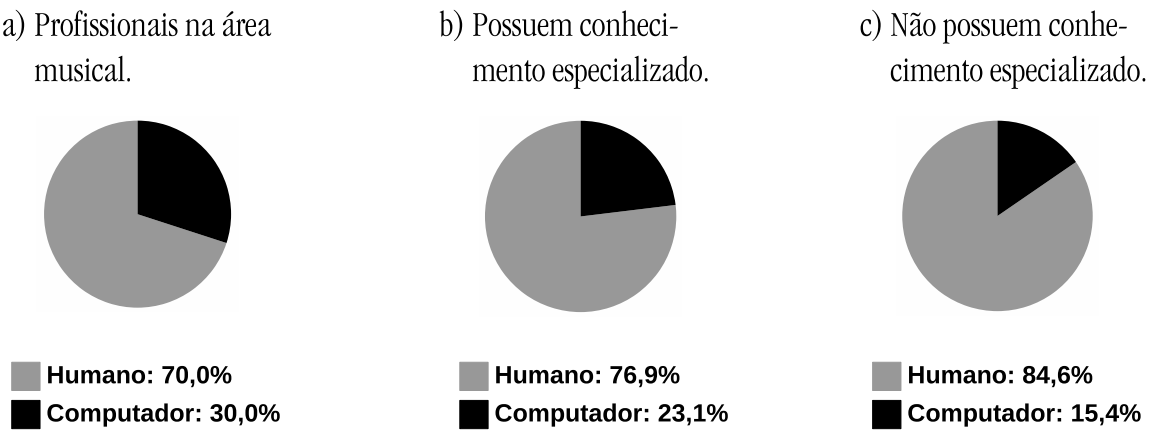
Figura 18 – Ouviria composições com estilo semelhante.




### 9.2.2 Composição Humana X Computador

A Figura 19 apresenta a pesquisa clássica do teste de Turing, aplicada ao trecho composto pelo computador.

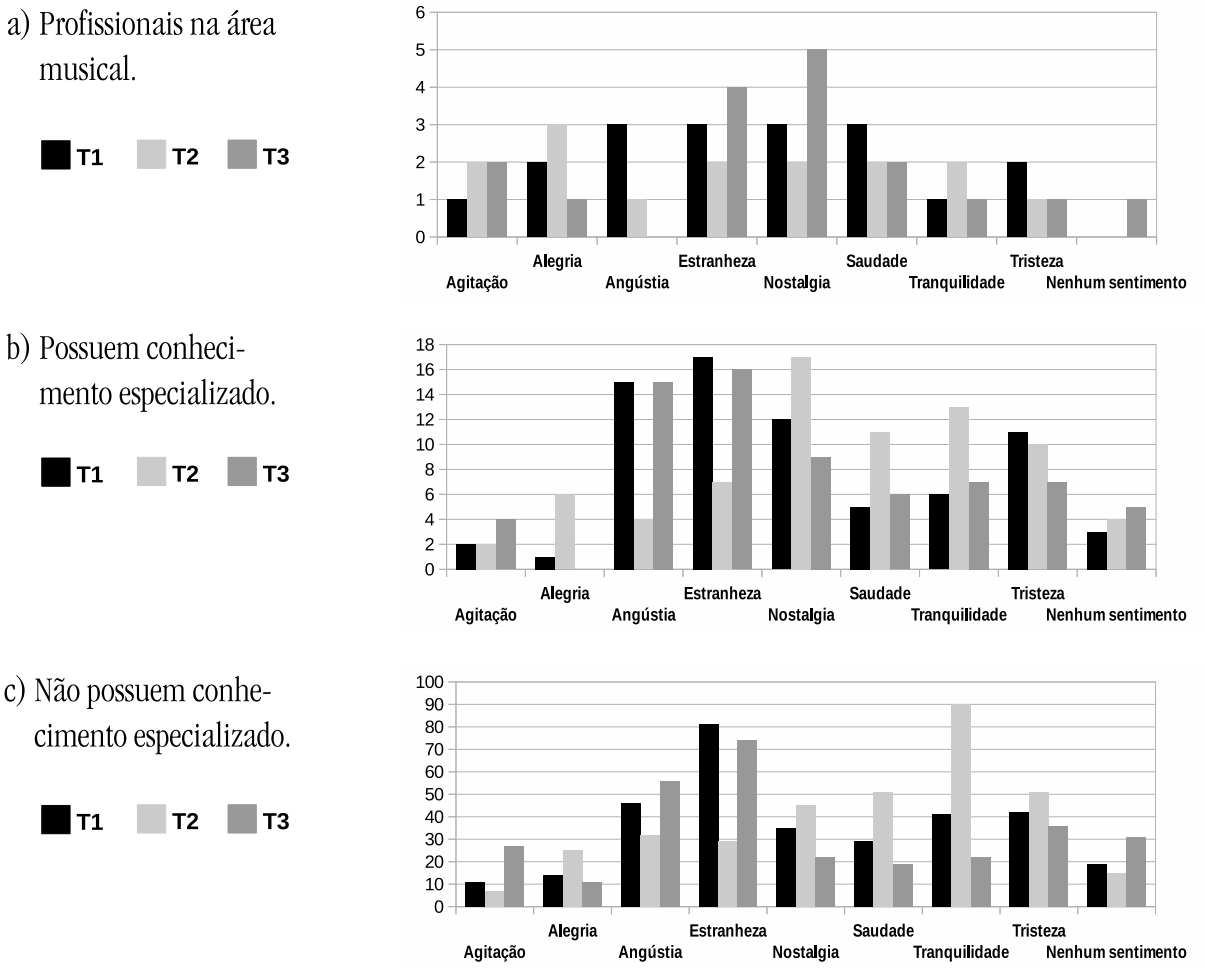
Figura 19 – O trecho foi composto por um ser humano ou por um computador.




### 9.2.3 Sentimentos despertados

A Figura 20 apresenta os sentimentos despertados ao ouvir cada trecho musical.

Figura 20 – Sentimentos despertados ao ouvir o trecho.



### 9.2.4 Associação por Memória

A Figura 16 apresenta as memórias advindas ao se ouvir os trechos musicais

Tabela 16 – Associação por memória.

| | |
|---|---|
| Uma vez fui para uma ópera e tinha musicas desse estilo | Minhas apresentações |
| Filmes do cinema mudo | Um passeio de bicicleta |
| Minha família | Filmes que tem como tema principal a época medieval |
| A morte de um ente querido | Exercício com instrumento musical |
| Algo em torno de musicas classicas como as Frédéric Chopin | Do filme "Meu Primeiro Amor" |
| Quando fiz parte de um grupo de ballet | Europa do séc. XVII |
| Esse tipo de música me traz a mente a música sacra | Velório do meu avô |
| Lembro de um filme"O som som coração" | O desfile de sete de setembro |
| Um pouco semelhante as músicas do renomado Beethoven | Me faz lembrar de momentos tristes da minha vida |
| O dia do meu casemento | Momentos de tranquilidade |
| Me lembra o filme Amadeus | Clipes da lindsey stirling |
| Apresentações do ballet bolchoi | A musica "Suite No. 1 In G Major for Solo Cello" |
| Trilha sonora chocolate com pimenta | A época em que eu fazia Ballet |
| Me lembra de um casamento ou uma festa de debutante | Quando comecei a ver Lord of Rings |
| Minha infância, talvez uma nostalgia | A morte da minha cadela |
| Como se uma pessoa que eu amei um dia tivesse soltado minha mão | Filmes antigos da Disney |
| Quando era mais novo e meu pai me mostrava musicas como esta | TV Cultura |
| Minha primeira apresentação, quando tinha uns 10-11 anos | Me lembra a música "Somewhere Over The Rainbow" |
| O nascimento de meus 7 filhos | Quando ganhei meu primeiro brinquedo |
| Me lembra o período em que fazia aulas de violino | Como era nossas vidas |
| Meus pais | Tempos difíceis que passei |
| O filme "a lista de shindler" | Pessoas dançando |
| Uma apresentação de ballet que minha fila apresentou | Memórias da minha infância |

## 9.3 Resultados, Conclusões e Trabalhos Futuros

Conforme os 4 grupos de participantes que ouviriam trechos semelhantes aos apresentados (Figura 13), observa-se que os menores valores encontram-se nos grupos: G2 - onde não houve associações de semelhança; e G4 - em que os trechos são identificados como compostos por um computador.

Desse modo, com relação ao trecho automático, os elevados valores apresentados no teste clássico, apontando a composição como sendo feita por um ser humano, corroboram com a identificação de preferência feita pelos participantes. Isso demonstra uma associação particular para com o trecho automático.

Conforme indicado na Figura 14 os três trechos apresentam um comportamento uniforme com relação aos 4 grupos estratificados, o que também é observado pelos gráficos das Figuras 15 e 16.

Pode-se observar que à medida em que o grau de especialidade dos participantes diminui, aumenta a sensação de que o trecho foi composto por um ser humano (Figura 19). Nesse caso, os maiores graus de familiaridade musical indicam uma maior percepção quanto à origem da composição, o que não influencia nos demais aspectos relacionados à contextualização ou “envolvimento” com o trecho automático.

Destaca-se adicionalmente que os valores mais baixos para o teste de Turing clássico, entre as 3 categorias de participantes ficaram na faixa dos 70%. Isso representa uma elevada indicação de que foi produzida uma composição contextualizado pelo sistema computadorizado.

Conforme o componente direcionado abrangido pelo teste, os gráficos da Figura 20 mostram que os sentimentos que se buscaram despertar com os trechos 2 e 3 foram atingidos, considerando os aspectos de "tranquilidade" e "estranheza".

Com relação aos aspectos envolvendo memórias, associações de preferência e sentimentos, os elementos coletados destacam a identificação dos participantes com a composição automática. Isso aponta na direção de que o propósito fundamental da composição gerada pelo computador foi atingido.

Tomando-se como base os resultados apresentados, pode-se concluir que a abordagem utilizada no presente trabalho atingiu satisfatoriamente seu objetivo. 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 dessa abordagem, o sistema Fraseado possibilita a composição musical algorítmica, com a capacidade de produção de trechos que podem ser associados a determinados contextos humanos. Essa capacidade pôde ser avaliada pelo método proposto - o Teste de Turing Expandido.

### 9.3.1 Trabalhos Futuros

Com relação a trabalhos futuros, podem ser identificados desdobramentos do presente trabalho com relação a 2 aspectos:

**Representação de Conhecimento Musical**

- Incorporação de acordes em composições;
- Representação de estruturas rítmicas;
- Apresentação de progressões harmônicas (caminhos em grafos) utilizando a Lógica de Primeira Ordem;
- Incorporação de novos componentes para aquisição de conhecimento.

**Interação Humano-Computador**

- Edição visual de dados com uso de partitura;
- Inserção de regras musicais pelo usuário;

- Criação de um ambiente colaborativo, sobre a plataforma web, para incorporação, descoberta, compartilhamento e análise de conhecimento musical.

# 10 Publicações

*Com a multiplicação contínua das máquinas, um dia seremos capazes de distinguir entre dez, vinte ou trinta mil sons diferentes.*

Luigi Russolo (1885 - 1947)

No transcorrer da presente pesquisa, os seguintes trabalhos foram publicados:

- Um ambiente de programação multi-paradigma voltado à composição musical automática - 67ª Reunião da Sociedade Brasileira Para o Progresso da Ciência.
- Representação de Conhecimento Musical e Programação Lógica Indutiva - Uma Revisão Sistemática - 15º Simpósio Brasileiro de Computação Musical.
- Programação Lógica Indutiva aplicada à Computação Musical - 14º Congresso de Engenharia de Áudio AES Brasil.

# Referências

ABDALLA, G.; DAMASCENO, C. D. N.; NAKAGAWA, E. Y. A systematic literature review on systems-of-systems knowledge representation. 2015. Citado na página 30.

ADOLFO, A. *O livro do músico*. [S.l.]: Irmãos Vitale, 1989. Citado na página 49.

AKERKAR, R.; SAJJA, P. *Knowledge-based systems*. [S.l.]: Jones & Bartlett Publishers, 2010. Citado na página 62.

ALVARO, J. L.; MIRANDA, E. R.; BARROS, B. Ev: Multilevel music knowlegde representation and programming. *Proceedings of SBCM, Belo Horizonte, Brazil*, 2005. Citado 2 vezes nas páginas 25 e 26.

ANDERS, T. *Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System*. Tese (Doutorado) — School of Music & Sonic Arts, Queen's University, Belfast., 2007. Citado 3 vezes nas páginas 25, 26 e 44.

ANDERS, T.; ANAGNOSTOPOULOU, C.; ALCORN, M. Strasheela: Design and usage of a music composition environment based on the oz programming model. In: *Multiparadigm Programming in Mozart/OZ*. [S.l.]: Springer Berlin Heidelberg, 2004. p. 277–291. Citado na página 26.

ANGLADE, A. et al. Improving music genre classification using automatically induced harmony rules. *Journal of New Music Research*, Taylor & Francis, v. 39, n. 4, p. 349–361, 2010. Citado 4 vezes nas páginas 21, 27, 28 e 34.

ARIZA, C. The interrogator as critic: The turing test and the evaluation of generative music systems. *Computer Music Journal*, MIT Press, v. 33, n. 2, p. 48–70, 2009. Citado na página 84.

ARIZA, C. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, v. 48, p. 513–582, 2013. Citado na página 84.

BALABAN, M. The music structures approach in knowledge representation for. *Computer Music Journal*, Citeseer, v. 20, n. 2, p. 96–111, 1996. Citado na página 20.

BENSON, D. J. Music: a mathematical offering. *The Mathematical Intelligencer*, Springer, v. 30, n. 1, p. 76–77, 2008. Citado na página 52.

BERRAR, D.; KONAGAYA, A.; SCHUSTER, A. Turing test considered mostly harmless. *New Generation Computing*, Springer, v. 31, n. 4, p. 241–263, 2013. Citado na página 84.

BERRAR, D. P.; SCHUSTER, A. Computing machinery and creativity: lessons learned from the turing test. *Kybernetes*, Emerald Group Publishing Limited, v. 43, n. 1, p. 82–91, 2014. Citado na página 86.

BISHOP, M.; BODEN, M. A. The turing test and artistic creativity. *Kybernetes*, Emerald Group Publishing Limited, v. 39, n. 3, p. 409–413, 2010. Citado na página 84.

BITTENCOURT, G. Representação de conhecimento: da metafísica aos programas. *Cursos JAI*, v. 98, p. 355, 1998. Citado na página 27.

BLOCKEEL, H. Top-down induction of first order logical decision trees. *Aicommunications*, Ios press, v. 12, n. 1-2, p. 119–120, 1999. Citado 2 vezes nas páginas 21 e 37.

BOWN, O. Player responses to a live algorithm: Conceptualising computational creativity without recourse to human comparisons. In: *Proceedings of the Sixth International Conference on Computational Creativity*. [S.l.: s.n.], 2015. p. 126–133. Citado na página 85.

BROWN, A. *Making Music with Java: An Introduction to Computer Music, Java Programming and the JMusic Library*. [S.l.]: Lulu. com, 2005. Citado na página 45.

CABRAL, T. Musicologia sistemática, humanismo e contemporaneidade. *OPUS-Revista Eletrônica da ANPPOM*, v. 20, n. 2, p. 125–150, 2014. Citado na página 51.

CHEN, L.; BABAR, M. A.; ALI, N. Variability management in software product lines: a systematic review. In: CARNEGIE MELLON UNIVERSITY. *Proceedings of the 13th International Software Product Line Conference*. [S.l.], 2009. p. 81–90. Citado na página 30.

CHONG, E. K. M.; DING, Q. Symbolic representation of chords for rule-based evaluation of tonal progressions. 2014. Citado 2 vezes nas páginas 27 e 33.

COLLINS, N. M. *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems*. Tese (Doutorado) — University of Cambridge, 2006. Citado na página 83.

CONCEIÇÃO, J. P. D. The aleph system made easy. *Integrated Master in Electrical and Computers Engineering, Faculdade de Engenharia da Universidade do Porto*, 2008. Citado na página 38.

DANNENBERG, R. B. A brief survey of music representation issues, techniques, and systems. MIT Press, 1993. Citado na página 51.

DELGADO, M.; FAJARDO, W.; MOLINA-SOLANA, M. A state of the art on computational music performance. *Expert Systems with Applications*, Elsevier, v. 38, n. 1, p. 155–160, 2011. Citado na página 20.

DIXON, S.; MAUCH, M.; ANGLADE, A. Probabilistic and logic-based modelling of harmony. In: *Exploring Music Contents*. [S.l.]: Springer, 2011. p. 1–19. Citado 4 vezes nas páginas 21, 28, 29 e 34.

DŽEROSKI, S.; LAVRAČ, N. An introduction to inductive logic programming. In: *Relational data mining*. [S.l.]: Springer, 2001. p. 48–73. Citado 2 vezes nas páginas 62 e 70.

FELICE, R. C. R. Referenciais neoclássicos e originalidade nas bachianas brasileiras n. 4 de heitor villa-lobos. Universidade Estadual Paulista (UNESP), 2016. Citado na página 85.

FERNÁNDEZ, J. D.; VICO, F. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, p. 513–582, 2013. Citado na página 20.

FORNARI, J. E. N. J. Introduction to the 15º brazilian symposium on computer music. In: *SBCM 2015, Campinas, SP, ISSN 2175-6759*. [S.l.: s.n.], 2015. p. 5. ISSN 2175-6759. Citado na página 19.

FRANCIS, J. R. *Algorithmic Computer Music*. [S.l.]: Creative Commons Attribution - John R. Francis, 2015. Citado na página 19.

GEBHARDT, R. B.; DAVIES, M. E.; SEEBER, B. U. Psychoacoustic approaches for harmonic music mixing. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 6, n. 5, p. 123, 2016. Citado na página 51.

GIL, J. Y. Cs 234319: Programming languages. 2015. Citado na página 46.

GIMENES, M.; SANTOS, A. R. C.; MANZOLLI, J. As estruturas verticais na improvisação de bill evans. In: *Anais do XIV Congresso da ANPPOM, Porto Alegre, RS*. [S.l.: s.n.], 2003. Citado na página 19.

GONCALVES, C. B. J.; HOMEM, M. R. P. Representação de conhecimento musical e programação lógica indutiva - uma revisção sistemática. In: *SBCM 2015, Campinas, SP, ISSN 2175-6759*. [S.l.: s.n.], 2015. p. 138–141. ISSN 2175-6759. Citado na página 27.

GROSOF, B.; DEAN, M.; KIFER, M. Semantic rules on the web. In: *International Semantic Web Conference (ISWC)*. [S.l.: s.n.], 2009. Citado na página 68.

GUIMARAES, J. O. Programming languages paradigms. In: . [S.l.]: UFSCar Sorocaba, 2015. p. 96–114. Citado 3 vezes nas páginas 25, 39 e 47.

HANDEL, G. F.; MORELL, T. *Judas Maccabaeus*. [S.l.: s.n.], 1747. Citado na página 56.

HIRAGA, R. et al. Rencon 2004: Turing test for musical expression. In: NATIONAL UNIVERSITY OF SINGAPORE. *Proceedings of the 2004 conference on New interfaces for musical expression*. [S.l.], 2004. p. 120–123. Citado na página 86.

HOLDEN, S. *Introduction to knowledge representation and reasoning*. [S.l.]: Sean Holden, 2005. Citado na página 44.

INOUE, K.; OHWADA, H.; YAMAMOTO, A. Inductive logic programming: Challenges. In: *Thirtieth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2016. Citado na página 62.

ISO. *Standard tuning frequency (Standard musical pitch)*. Geneva, Switzerland, 1975. Citado na página 49.

J MCKAY C, F. I. B. Evaluating automated classification techniques for folk music genres from the brazilian northeast. In: *SBCM 2015, Campinas, SP, ISSN 2175-6759*. [S.l.: s.n.], 2015. p. 3–12. ISSN 2175-6759. Citado na página 20.

KEELE, S. Guidelines for performing systematic literature reviews in software engineering. In: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. [S.l.: s.n.], 2007. Citado na página 22.

KHURUM, M.; GORSCHEK, T. A systematic review of domain analysis solutions for product lines. *Journal of Systems and Software*, Elsevier, v. 82, n. 12, p. 1982–2003, 2009. Citado na página 30.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, Elsevier, v. 51, n. 1, p. 7–15, 2009. Citado 2 vezes nas páginas 30 e 32.

KOSTELETOS, G.; GEORGAKI, A. A turing test for the singing voice as an anthropological tool: epistemological and technical issues. Citado na página 85.

KOSTER, A.; SABATER-MIR, J.; SCHORLEMMER, M. Trust alignment: a sine qua non of open multi-agent systems. In: *On the Move to Meaningful Internet Systems: OTM 2011*. [S.l.]: Springer, 2011. p. 182–199. Citado na página 39.

LACERDA, O. *COMPENDIO DE TEORIA ELEMENTAR DA MUSICA*. [S.l.]: RICORDI DO BRASIL, 2008. Citado na página 52.

LAPP, D. R. *The physics of music and musical instruments*. [S.l.]: Wright Center for Innovative Science Education, Tufts University, 2003. Citado na página 52.

LATHAM, A. *The Oxford dictionary of musical works*. [S.l.]: Oxford University Press, USA, 2004. Citado na página 85.

LENNON, J.; MCCARTNEY, P.; WHITCOMB, K. *Eleanor Rigby*. [S.l.]: Maclen Music, 1966. Citado na página 77.

LEY, E. D.; JACOBS, D. Rules-based analysis with jboss drools: adding intelligence to automation. *Proceedings of ICALEPCS 2011*, p. 790–793, 2011. Citado na página 39.

LIMA, L. V. *Um Sistema de Composição Musical Dirigido por Estilo*. Tese (Doutorado) — Departamento de Engenharia Elétrica, Escola Politécnica, Universidade de São Paulo, SP., 1998. Citado 2 vezes nas páginas 19 e 26.

LIU, D.; GU, T.; XUE, J.-P. Rule engine based on improvement rete algorithm. In: IEEE. *Apperceiving Computing and Intelligence Analysis (ICACIA), 2010 International Conference on*. [S.l.], 2010. p. 346–349. Citado na página 39.

LLOYD, J. W. *Foundations of logic programming*. [S.l.]: Springer Science & Business Media, 2012. Citado na página 62.

MAESTRE, E. et al. Expressive concatenative synthesis by reusing samples from real performance recordings. *Computer Music Journal*, MIT Press, v. 33, n. 4, p. 23–42, 2009. Citado 4 vezes nas páginas 28, 29, 36 e 37.

MALHEIRO, F.; CAVACO, S. Automatic musical instrument and note recognition. In: ISMIR. [S.l.], 2011. Citado na página 49.

MANTARAS, R. L. de. Making music with ai: Some examples. In: ENTREPRENEUR AND MOUNTAINEER. *Proceedings of the 2006 Conference on Rob Milne: A tribute to a Pioneering AI Scientist*. [S.l.], 2006. p. 90–100. Citado na página 20.

MAXIMILIANO, A. P. et al. Copista - sistema de omr para a recuperação de acervo histórico musical. In: *SBCM 2015, Campinas, SP, ISSN 2175-6759*. [S.l.: s.n.], 2015. p. 48–59. ISSN 2175-6759. Citado na página 20.

MELLO, M. d. S. F. Reflexões sobre linguistica e cognição musical. Campinas, SP, 2003. Citado na página 27.

MILETTO, E. M. et al. Introdução a computação musical. In: *CBComp - X Congresso Brasileiro de Computação, 4., Itajaí, 2004. "Anais...". Itajaí, SC - Brasil, ISSN 1677-2822.* [S.l.: s.n.], 2004. p. 883–902. ISSN 1677-2822. Citado na página 19.

MINSKY, M. Music, mind, and meaning. *Computer Music Journal*, JSTOR, p. 28–44, 1981. Citado na página 19.

MIRA, J. et al. Knowledge modelling for the motion detection task: The algorithmic lateral inhibition method. *Expert Systems with Applications*, Elsevier, v. 27, n. 2, p. 169–185, 2004. Citado na página 62.

MIRANDA, E. R. *Sound design: an artificial intelligence approach.* Tese (Doutorado) — University of Edinburgh, 1995. Citado na página 27.

MIRANDA, E. R.; ALVARO, J. L.; BARROS, B. Music knowledge analysis: Towards an efficient representation for composition. *Current Topics in Artificial Intelligence. Springer Berlin Heidelberg*, p. 331–341, 2005. Citado na página 20.

MORAES, V.; TOQUINHO. *A flor da noite.* 1971. Citado na página 45.

MORALES, E. Learning patterns for playing strategies. *International Computer Chess Association Journal*, Citeseer, v. 17, n. 1, p. 15–26, 1994. Citado 3 vezes nas páginas 21, 35 e 39.

MORALES, E.; MORALES, R. Learning musical rules. In: CITESEER. *Proceedings of the International Joint Conference on Artificial Inteligence.* [S.l.], 1995. Citado 3 vezes nas páginas 20, 28 e 35.

NAIM, R. et al. Comparative studies of 10 programming languages within 10 diverse criteria-a team 10 comp6411-s10 term report. *arXiv preprint arXiv:1008.3561*, 2010. Citado na página 46.

NASCIMENTO, M.; BRANT, F. *Maria Maria.* [S.l.]: editado y distribuido por Fonogram, 1983. Citado na página 61.

NIERHAUS, G. *Algorithmic composition: paradigms of automated music generation.* [S.l.]: Springer Science & Business Media, 2009. Citado na página 44.

NOGUEIRA, M. J. M. *Paulinho Nogueira - Influências na obra Bachianinha n. 1.* [S.l.]: Entrevista - IFSP - Campus São Roque, 2017. Citado na página 85.

NUMAO, M.; TAKAGI, S.; NAKAMURA, K. Constructive adaptive user interfaces-composing music based on human feelings. In: MENLO PARK, CA; CAMBRIDGE, MA; LONDON; AAAI PRESS; MIT PRESS; 1999. *Proceedings of the National Conference on Artificial Intelligence.* [S.l.], 2002. p. 193–198. Citado 3 vezes nas páginas 21, 28 e 36.

NUSSBAUM, C. O. *The musical representation: Meaning, ontology, and emotion.* [S.l.]: Mit Press, 2007. Citado na página 51.

OLIVEIRA, C. E. S. Utilizando mapas conceituais no ensino de ciências da 8ª série. *Universidade Federal de Alagoas, AL*, 2007. Citado na página 19.

PACHET, F.; RAMALHO, G.; CARRIVE, J. Representing temporal musical objects and reasoning in the muses system. *Journal of New Music Research*, v. 25, n. 3, p. 253–73, 1996. Citado na página 26.

PARNCUTT, R. Musicologia sistemática: a história e o futuro do ensino acadêmico musical no ocidente. *Em Pauta*, v. 20, n. 34/35, p. 145–185, 2012. Citado na página 51.

PATEL, J.; OZA, B. A. Survey on graph pattern mining approach. In: IJEDR. *International Journal of Engineering Development and Research*. [S.l.], 2014. v. 2, n. 1 (March 2014). Citado na página 39.

PEARCE, M.; MEREDITH, D.; WIGGINS, G. Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, SAGE Publications Sage UK: London, England, v. 6, n. 2, p. 119–147, 2002. Citado 2 vezes nas páginas 82 e 84.

PEDLER, D. *The songwriting secrets of the Beatles*. [S.l.]: Omnibus Press, 2010. Citado na página 77.

PEREZ, A.; RAMIREZ, R.; KERSTEN, S. Modeling moods in violin performances. In: UNIVERSITÄTSVERLAG DER TU, UNIVERSITÄTSBIBLIOTHEK. *SMC 08: 5th Sound and Music Computing Conference: Sound in Space-Space in Sound, July 31st-August 3rd, 2008, Berlin, Germany: Proceedings*. [S.l.], 2008. p. 30. Citado 4 vezes nas páginas 21, 28, 29 e 36.

POMPE, U.; KONONENKO, I.; MAKSE, T. An application of ilp in a musical database: Learning to compose the two-voice counterpoint. Citeseer, 1996. Citado 4 vezes nas páginas 21, 28, 35 e 39.

QUEIROZ, A. A. de. Uma notação musical para representação de progressões harmônicas utilizando grafos. *Revista Música Hodie*, v. 9, n. 1, 2010. Citado na página 49.

QUINLAN, J. R. Learning logical definitions from relations. *Machine learning*, Springer, v. 5, n. 3, p. 239–266, 1990. Citado 3 vezes nas páginas 27, 30 e 39.

RABAI, L. B. A.; COHEN, B.; MILI, A. Programming language use in us academia and industry. *Informatics in Education*, Institute of Mathematics and Informatics, v. 14, n. 2, p. 143, 2015. Citado 2 vezes nas páginas 47 e 48.

RAMIREZ, R.; HAZAN, A. Modeling expressive music performance in jazz. In: *FLAIRS Conference*. [S.l.: s.n.], 2005. p. 86–91. Citado 4 vezes nas páginas 20, 28, 29 e 37.

ROADS, C. Research in music and artificial intelligence. *ACM Computing Surveys (CSUR)*, ACM, v. 17, n. 2, p. 163–190, 1985. Citado na página 20.

ROADS, C. *The computer music tutorial*. [S.l.]: MIT press, 1996. Citado na página 51.

ROBERTS, M. J. *Fundamentos de sinais e sistemas*. [S.l.]: AMGH Editora, 2009. Citado 2 vezes nas páginas 52 e 53.

SAKHARE, Y. N.; HANCHATE, M. D. Comparative study of musical performance by machine learning. *International Journal on Recent and Innovation Trends in Computing and Communication, ISSN 2321-8169*, v. 2, n. 2, p. 347–352, 2014. ISSN 2321-8169. Citado 3 vezes nas páginas 21, 29 e 38.

SALAMON, J. *Melody extraction from polyphonic music signals*. Tese (Doutorado) — Ph. D. thesis, Department of Information and Communication Technologies Universitat Pompeu Fabra, Barcelona, Spain, 2013. Citado na página 39.

SALZBERG, S. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning, ISSN 0885-6125*, Kluwer Academic Publishers, v. 16, n. 3, p. 235–240, 1994. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1007/BF00993309>. Citado 3 vezes nas páginas 28, 37 e 39.

SANTANA, H. et al. Vexpat: An analysis tool for the discovery of musical patterns. In: *Proceedings of IX Brazilian Symposium on Computer Music. Campinas, SP*. [S.l.: s.n.], 2003. Citado na página 27.

SANTOS, J. M. dos et al. Avaliação de desempenho do impacto de metodologias ágeis no desenvolvimento de software. In: *Workshop de Avaliação de desempenho da Fasete*. [S.l.: s.n.], 2008. Citado na página 83.

SBC. *Sociedade Brasileira de Computação, disponivel em http://www.sbc.org.br, acessado em 8/09/2015*. 2015. Disponível em: <http://www.sbc.org.br>. Citado na página 19.

SCHOENBERG, A. *Fundamentos da composição musical*. [S.l.]: Edusp, 1990. Citado na página 49.

SEBESTA, R. W. *Concepts of programming languages*. [S.l.]: Pearson Educatiion, 2012. Citado 2 vezes nas páginas 46 e 57.

SERAPIAO, S. P. *Ritornello: um Framework para Representação de Conhecimento Musical*. [S.l.]: Dissertação de Mestrado em Ciência da Computação. Universidade Federal de Pernambuco, Recife, PE, 2004. Citado na página 26.

SILVA, P. R. G. Derivation of som-g granular synthesis instruments from audio signals by atomic decomposition. In: *Proceedings of XII Brazilian Symposium on Computer Music. Recife, PE*. [S.l.: s.n.], 2009. Citado na página 27.

SMAIL, A.; WIGGINS, G. A.; MIRANDA, E. R. Music representation - between the musician and the computer. *Springer London*, p. 108–119, 1994. Citado na página 20.

SOUZA, R. C. de; FARIA, R. R. A. Oito reflexões sobre a criatividade na composição auxiliada por computadores. 2011. Citado na página 84.

SRINIVASAN, A. *The Aleph manual: version 4 and above*. 2007. Citado 3 vezes nas páginas 21, 38 e 70.

STACKOVERFLOW. *Stack Overflow Developer Survey Results 2016*. 2016. Disponível em: <http://stackoverflow.com/research/developer-survey-2016>. Citado na página 48.

TEIXEIRA, L. d. M. *Da representação do conhecimento musical ao esboço conceitual de uma sociedade de agentes em Harmonia*. Tese (Doutorado) — Dissertação de Mestrado, COPIN-UFPB, 1997. Citado na página 27.

THYWISSEN, K. *GeNotator: an environment for exploring the application of evolutionary techniques in computer assisted composition*. [S.l.]: Cambridge University Press. New York, NY, USA, 1999. Citado na página 26.

TIOBE. *Tiobe Software Index 2016, Tiobe Programming Community*. 2016. Disponível em: <http://www.tiobe.com/tiobe-index/>. Citado na página 47.

TOAL, R. et al. *Programming Language Explorations*. [S.l.]: CRC Press, 2016. Citado 2 vezes nas páginas 47 e 49.

TOMAS, G. H. et al. Smart cities architectures-a systematic review. In: *ICEIS (2)*. [S.l.: s.n.], 2013. p. 410–417. Citado na página 30.

TURING, A. M. Computing machinery and intelligence. *Mind*, JSTOR, v. 59, n. 236, p. 433–460, 1950. Citado na página 83.

WEINSTEIN, J. L. *Musical pitch and international agreement*. [S.l.]: JSTOR, 1952. Citado na página 49.

WHALLEY, I. International computer music conference 2004: Papers. In: *Proceedings International Computer Music Conference, Miami, ICMA Press*. [S.l.]: JSTOR, 2005. Citado 2 vezes nas páginas 19 e 20.

WIGGINS, G. A.; SMAILL, A. *Musical Knowledge: what can Artificial Intelligence bring to the musician?* 2000. Citado na página 20.

WULFHORST, R. D. et al. A multiagent approach for musical interactive systems. In: ACM. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. [S.l.], 2003. p. 584–591. Citado na página 27.

# APÊNDICE A - Sistema de sintetização

**Esquema geral** (Figura 21):

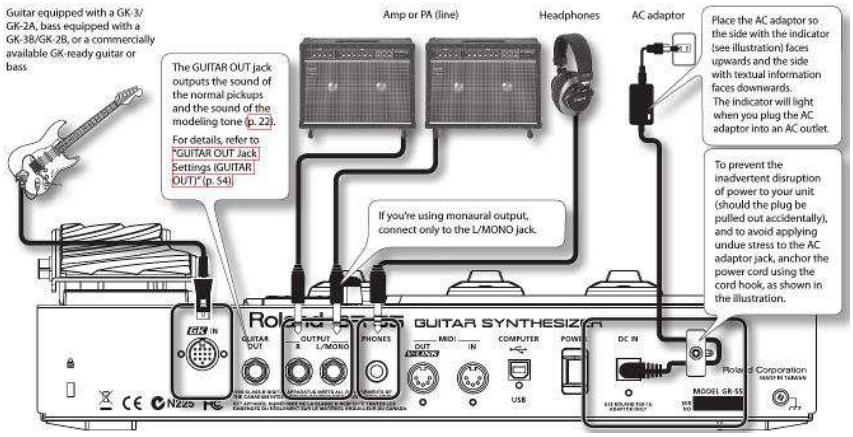



Figura 21 – Sintetização - Conexões GR-55 (Fonte: Manual Roland).

**Acoplamento à guitarra** (Figura 22):

Figura 22 – Sintetização - Acoplamento à guitarra.

**Captação hexafônica** (Figura 23):

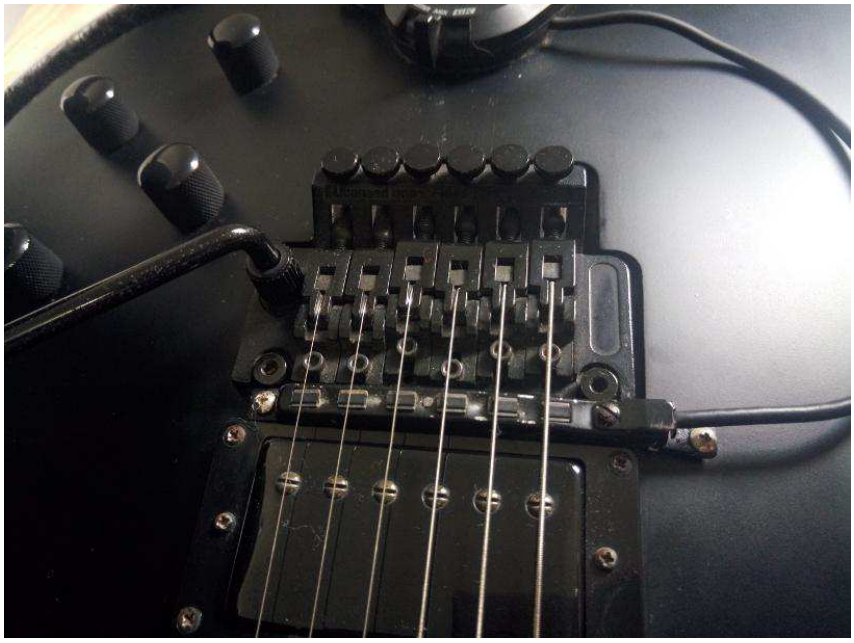


Figura 23 – captador hexafônico.

**O controlador Roland GK-3** (Figura 24):


Figura 24 – controlador Roland GK-3.

**O sintetizador Roland GR-55** (Figura 25):


Figura 25 – Sintetizador Roland GR-55.

**Armazenamento de áudio** (Figura 26):


Figura 26 – GR-55 - Armazenamento.