

# Framework de Composição Musical Algorítmica

Fernando Augusto Paz, Benjamin Grando Moreira

Centro de Ciências Tecnológicas da Terra e do Mar – Universidade do Vale do Itajaí  
(UNIVALI)  
88.302-202 – Itajaí – SC – Brazil  
{fernandopaz,benjamin}@univali.br

***Abstract.** This paper describes the project of a framework to algorithmic music composition, a tool that aims to reproduce the music composition process using computer resources. The system offers the inclusion of composition algorithms responsible for making musical phrases according to a set of parameters specified in advance. Thus, the feature may be coupled to external agents, such as applications and electronic games, so its results can be modified to meet their specific demands.*

***Resumo.** Este artigo descreve o projeto de um framework para composição musical algorítmica, uma ferramenta que se propõe a reproduzir o processo de composição musical através de recursos computacionais. O sistema permite a inclusão de algoritmos de composição responsáveis por criar trechos musicais de acordo com um conjunto de parâmetros previamente especificados. Dessa forma, o recurso pode ser acoplado a agentes externos, como aplicações e jogos eletrônicos, podendo seus resultados ser manipulados e assim atender sua demanda específica.*

## 1. Problematização

A intensa presença da música na indústria mundial de mídia e entretenimento desperta interesse no mercado a ponto de existir estudos para automatizá-la [Kirke e Miranda 2009], experiências que se tem registro desde a década de 60 [Kurzweil 2002].

O processo de composição, apesar de ser altamente subjetivo, pode ter a complexidade reduzida para que se permita sua reprodução de forma computacional [Delgado, Fajardo e Molina-Solana 2008]. Algoritmos de computadores podem aproveitar conceitos e padrões existentes na teoria musical básica para desenvolverem trechos musicais coerentes e satisfatórios de forma automatizada.

O mercado de jogos, devido à sua abordagem direcionada a músicas temáticas e de ambiente [Belinkie 1999], é um potencial consumidor desse tipo de algoritmo, principalmente considerando a composição em tempo de execução, que proporcionaria experiências mais profundas para os jogadores [Molina-Solana 2009].

## 2. Solução Proposta

A proposta apresentada neste artigo abrange um framework para auxiliar na composição algorítmica, ou composição musical através de computação, permitindo o

desenvolvimento de bibliotecas de programação capazes de criar músicas de acordo com determinados parâmetros fornecidos por agentes externos.

Através do algoritmo de composição o framework apresenta o resultando em dados no formato MIDI (*Musical Instrument Digital Interface* - Interface Digital para Instrumentos Musicais), podendo estes ser aproveitados e adaptados às devidas necessidades.

O framework foi projetado para ser fracamente acoplado, ou seja, permitir a inclusão de outros algoritmos para desenvolver os trechos musicais de forma automática. Cada algoritmo pode ser configurado com o recebimento de parâmetros que são definidos pelos algoritmos que podem ser agregados ao sistema, contribuindo para a independência dos métodos de composição.

### **3. Fundamentação teórica**

Como esse artigo aborda apenas o desenvolvimento do framework e não algoritmos de composição a fundamentação teórica abordará apenas aspectos tecnológicos do desenvolvimento, sendo esses o formato MIDI e o aplicativo JMusic.

#### **3.1. MIDI**

O resultado da composição musical através da biblioteca será um conjunto de dados no formato MIDI, podendo este ser salvo em arquivo ou simplesmente reproduzido diretamente pelo computador em tempo real. Dados no formato MIDI armazenam instruções para a execução de uma música. Pode-se compará-lo a uma partitura digital, pois contém os parâmetros de cada evento (frequência, duração e altura da nota, entre outros), possibilitando assim a reprodução musical de maneira digital [Huber 2007].

Além de ser um formato difundido computacionalmente, o armazenamento em MIDI facilita uma posterior edição da música, bem como a rápida sua geração e execução em tempo real por parte de programas externos.

#### **3.2. JMusic**

Para auxiliar no módulo de composição e reprodução musical, o sistema será integrado com a jMusic, uma biblioteca de programação Java que simplifica a manipulação da estrutura de dados no formato MIDI. Os métodos de acesso às instruções musicais são bem robustos e intuitivos. É possível ainda reproduzir dados MIDI em tempo de execução ou armazená-los em formato de arquivo.

Optou-se pela utilização da jMusic pois foi o maior destaque em um estudo que envolveu 14 bibliotecas musicais para Java. A biblioteca apresentou o melhor desempenho nas categorias de análise estipuladas pelos autores, superando até mesmo o Java Sound, recurso nativo da plataforma Java [Costalonga, et al. 2005].

### **4. Aplicações Similares**

A análise de aplicações similares ajuda a compreender a complexidade da proposta, bem como reconhecer as possibilidades e capacidades de um sistema compositor. Para a comparação (Tabela 1), foram considerados fatores técnicos e musicais, que de certa forma influenciaram no desenvolvimento desse projeto.

[Molina-solana 2009] é um projeto que se destaca por utilizar sistemas multiagentes para compor músicas que tentam corresponder a sentimentos humanos. consiste em utilizar um agente compositor que interage com agentes geradores (ou instrumentistas) para criar trechos musicais. Devido à possibilidade de manter a música coerente somente aplicando regras e padrões matemáticos, a aplicação também aproveita o conceito de sistema especialista na definição de regras para gerar conjuntos de melodia, harmonia e ritmo. Entretanto, embora haja várias possibilidades combinatórias, basear-se em regras pré-determinadas traz como consequência a limitação da “criatividade musical” do sistema.

O [Impro-Visor 2008] é um sistema especialista que cria solos dada uma certa progressão de acordes (Figura 7). Para obter resultados satisfatórios, foram definidas regras envolvendo teoria musical de nível intermediário e avançado (modos gregorianos, campo harmônico, entre outros) pertinentes a diversos estilos musicais, principalmente o jazz. A presença de um sistema baseado em regras pré-determinadas tem como principal desvantagem a suposta limitação da composição, porém ainda permite produzir diversos resultados interessantes.

[Menezes 2008], apresentou em sua monografia o FugueMaker, software gerador de fugas no estilo do consagrado compositor J. S. Bach. Por ser um estilo musical extremamente matemático, é possível representá-lo através de algoritmos que, dado um determinado sujeito (frase musical que será a base da fuga), produzirá uma sequência de notas para cada instrumento, resultando em uma legítima fuga bachiana.

**Tabela 1. Análise comparativa das propostas semelhantes**

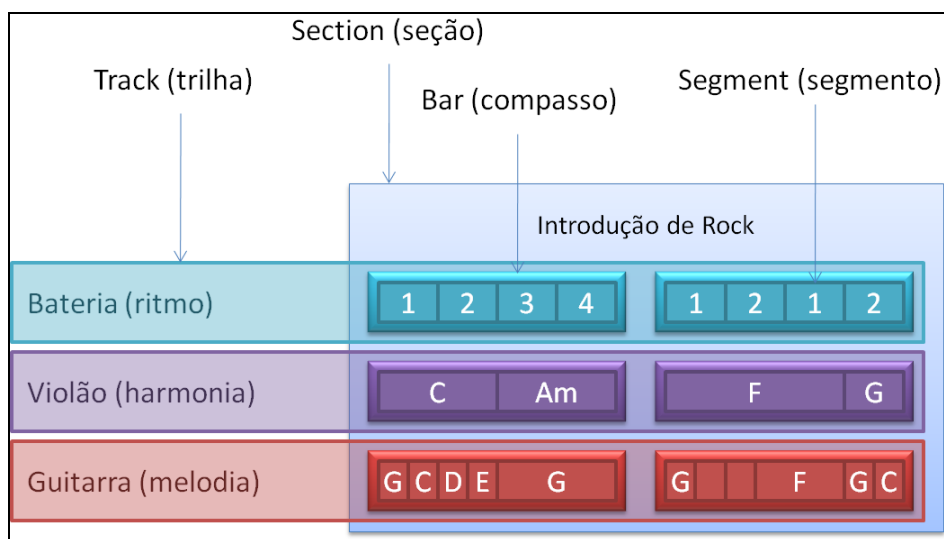
| <b>Fatores</b>           | <b>INMAMUSYS</b>                           | <b>Impro-Visor</b>          | <b>FugueMaker</b>                                 | <b>Framework proposto</b> |
|--------------------------|--|-----------------------------|---|---------------------------|
| Linguagem de Programação | Java                                       | Java                        | Java  | Java                      |
| Abordagem computacional  | Sistema Multiagentes, Sistema especialista | Gramática livre de contexto | Regras, funções matemáticas                       | Independente              |
| Estilo musical           | Vários                                     | Jazz (nativo)               | Fuga  | Independente              |
| Formato da música gerada | MIDI                                       | MIDI, MusicXML              | MIDI, PDF (partitura), LilyPond (notação musical) | MIDI, MusicXML            |
| Instrumentos             | 3  | Ilimitado                   | 3   | Ilimitado                 |

Com relação ao framework proposto, sobre a abordagem computacional, esse se apresenta independente pois depende do algoritmo desenvolvido. Em arquivos de exemplo incluídos no framework os algoritmos foram desenvolvidos com o método de Monte Carlo. O mesmo acontece com o estilo musical, onde este depende de como foi trabalhado o algoritmo. Nos algoritmos incluídos foi trabalhada a composição de músicas para jogos RPG ambientadas em floresta e cidades (vilas).

## 5. Estrutura do Projeto

Para o framework proposto, apenas os elementos musicais precisam ser informados e a execução desses elementos é feita pelo framework. Em adição a simples geração das músicas, o framework suporta a inclusão de telas de configuração e leitura de parâmetros por XML para gerar uma biblioteca musical, facilitando o uso dos algoritmos por pessoas sem grandes conhecimentos de programação e/ou música. Sendo assim, o framework também é utilizado para gerar uma biblioteca musical com acesso e configuração via interface gráfica.

A Figura 1 ilustra a estrutura que se implementou no projeto. Há uma hierarquia de elementos que se inicia com um segmento. Dependendo do elemento no qual está inserido, o segmento pode ser formado por uma simples nota, um conjunto de notas ou uma batida (no caso do ritmo).



**Figura 1. Estrutura de composição musical sugerida.**

Para fins de organização, o compasso contempla uma quantidade variável de segmentos. Dessa forma, é possível aplicar modificações em todos os segmentos de um compasso de uma só vez, por exemplo.

Os compassos ficam armazenados em trilhas, que são faixas destinadas a um determinado instrumento, podendo estas serem atribuídas aos mais diversos elementos musicais (melodia, acompanhamento, etc). Em um cenário normal, como o proposto na Figura 1, estão sendo apresentados uma melodia, uma harmonia e um ritmo, mas o sistema é capaz também de permitir a inserção de mais de um elemento do mesmo tipo, bem como a omissão de um deles (duas melodias e duas harmonias, por exemplo).

Por fim, o conjunto de trilhas está contido em um trecho, que representa uma parte bem definida da música. A decisão de manter esta hierarquia possibilita o algoritmo compositor trabalhar com mais de um compasso por vez, o que implica diretamente em criar frases musicais mais extensas.

As trilhas contêm conjuntos de compassos, e não ao contrário, pois dessa forma a música é criada uma trilha por vez, já com todos os compassos preenchidos. Além de respeitar a mesma estrutura do formato MIDI, organizar os elementos dessa maneira otimizou a integração entre os algoritmos de composição, uma vez que tornou-se mais fácil encaixar trechos de músicas diferentes entre si.

## 6. Interface Gráfica

Para facilitar a interação com usuários leigos, o sistema desenvolvido acompanha uma interface gráfica simples que permite manipular os algoritmos de composição de maneira mais prática e rápida.

Através da tela principal (Figura 2), é possível adicionar diferentes algoritmos de composição de forma sequencial.

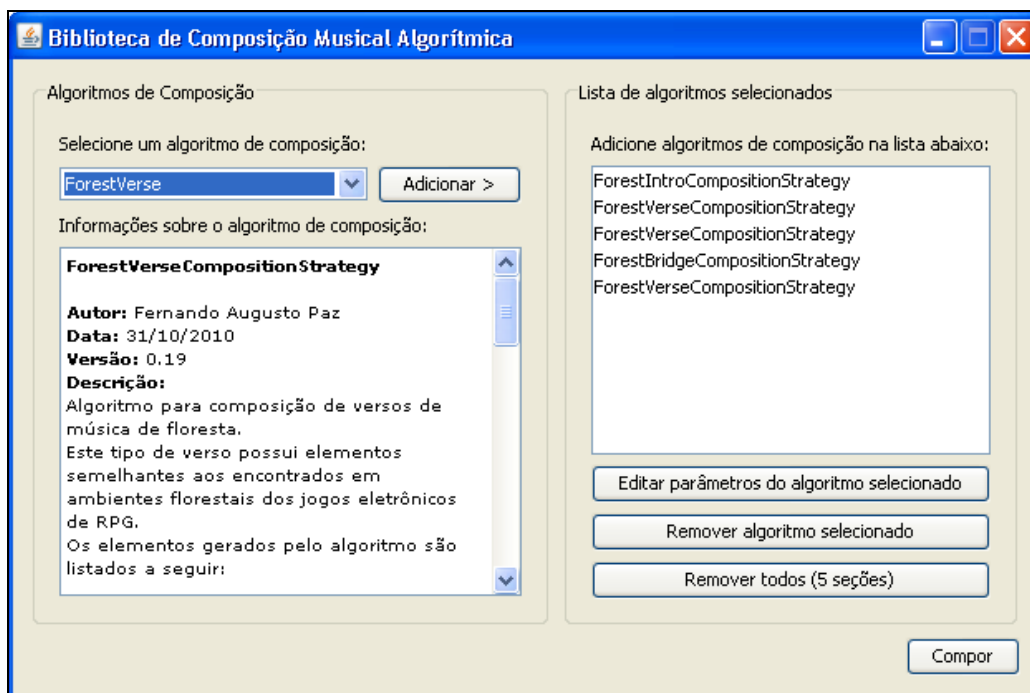
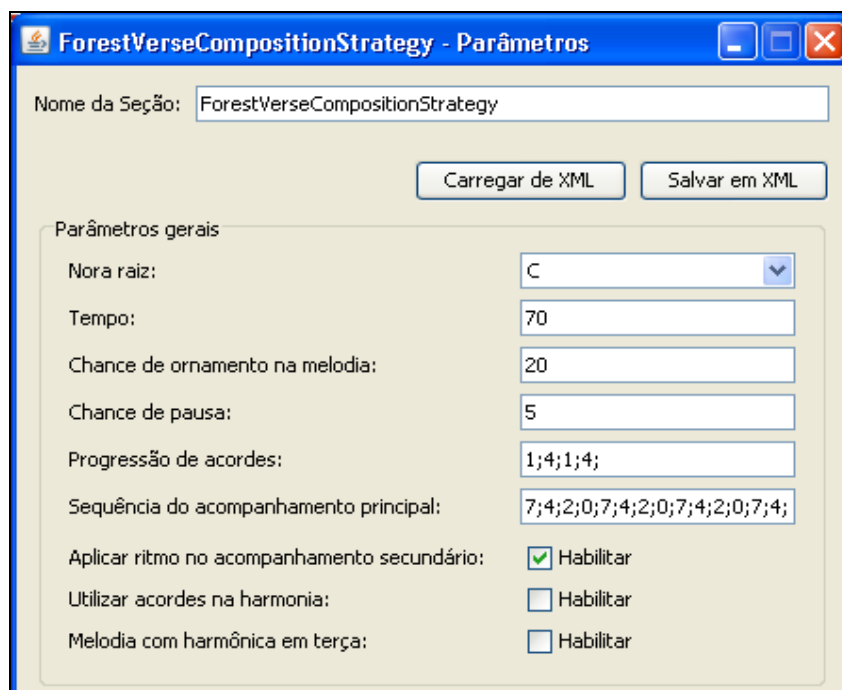


Figura 2. Tela principal da interface gráfica

Nesta tela também consta a função de editar os parâmetros de composição do algoritmo, caso este tenha o recurso disponível. Essa tela de configuração é criada pelo próprio algoritmo de composição e por isso pode ser especificada com os parâmetros necessários para o algoritmo. A Figura 3 apresenta parte de uma tela de configuração de parâmetros de um dos algoritmos incluídos no framework.



**Figura 3. Parte de uma tela de parâmetros de composição**

Alguns dos algoritmos que acompanham o sistema disponibilizam a tela com a edição dos parâmetros, permitindo gerar dados aleatórios, salvar e carregar XML (*eXtended Markup Language* – Linguagem de Marcação Extendida). Essa tela é de responsabilidade do desenvolvedor do algoritmo, ou seja, nem todos terão a obrigatoriedade de disponibilizá-la.

Solicitando a composição das seções, o sistema acionará o método de composição de cada algoritmo selecionado. Por fim, todos os dados serão reunidos em uma única variável contendo as informações em MIDI.

## 7. Aplicação de Demonstração

Apesar da interface gráfica já ser uma aplicação que demonstra as capacidades do sistema desenvolvido, considerou-se importante a possível integração com outros programas, preferencialmente aqueles que compartilham o mesmo contexto. Nesse caso, optou-se pela interação com um jogo eletrônico, que poderia usufruir das músicas geradas automaticamente pelo sistema compositor.

Por questões de praticidade, o jogo eletrônico foi desenvolvido através da *game engine* (motor de jogo) RPG Maker XP. Além de oferecer mais facilidade na criação, a ferramenta possui suporte a RGSS (*Ruby Game Scripting System* – Sistema de Script de Jogos em Ruby), uma variação da linguagem de programação Ruby, permitindo a chamada externa ao sistema compositor.

O jogo desenvolvido apresenta dois cenários principais: uma vila e uma floresta. Para cada ambiente, foi programado para que seja gerada uma música específica através do sistema compositor. Dessa forma, no momento em que o personagem entra no ambiente florestal, é executada uma chamada ao sistema compositor, solicitando uma música utilizando os algoritmos de composição de temas de floresta. O mesmo acontece

ao entrar na vila. Uma demonstração desse teste está disponível no Youtube em: <http://www.youtube.com/watch?v=ToFz1wuKPoY>.

Os algoritmos de composição que acompanham o sistema foram desenvolvidos especificamente para oferecerem a música adequada para o ambiente ao qual foram propostos. No caso da vila, foram utilizados elementos que remetessem a esse ambiente, assim como na floresta.

Para a escolha dos elementos mais propícios, analisou-se também a estrutura de músicas ambientes de outros jogos eletrônicos, como os da série Final Fantasy, Chrono Trigger, RPG Maker, entre outros. As semelhanças identificadas influenciaram significativamente nos métodos de composição dos algoritmos.

## 8. Conclusões

A criação de um algoritmo de composição musical ainda depende do conhecimento dos elementos necessários para a composição de uma música, incluindo recomendações teóricas para facilitar a criação sem perder a coerência. É preciso saber a estrutura básica de uma música e como padrões matemáticos podem auxiliar em sua composição, fazendo com que este conceito possa ser transportado para o meio computacional.

Mesmo sem conhecimento da teoria musical, pessoas podem gerar músicas com base em algoritmos prontos, apenas modificando seus parâmetros, o que gerará uma nova música, podendo, por exemplo, deixá-la mais rápida. Essa alteração nos parâmetros não requer conhecimento algum sobre a teoria musical e pode ser feito por tentativa e erro, verificando se a música gerada foi satisfatória.

A análise de ferramentas semelhantes permitiu identificar os fatores comuns que sistemas da área de computação musical compartilham. Há várias propostas aplicadas e questões a serem exploradas no ramo. Nesse estudo, ficou evidente a possibilidade de utilizar o poder da computação para reproduzir o processo de composição musical. Embora ainda esteja em fase inicial de desenvolvimento, a área de composição algorítmica possui um grande potencial que pode ser aproveitado na indústria de mídia e entretenimento conforme o tema for evoluindo.

Após compreender as abordagens necessárias e analisar propostas semelhantes, definir as tecnologias necessárias tornou-se uma tarefa mais fácil. Optou-se por ferramentas que auxiliassem na manipulação de recursos musicais de maneira eficiente. Entre elas foi encontrado a jMusic, uma biblioteca que contribuiu significativamente para o tratamento dos dados musicais mais brutos.

Para justificar sua aplicabilidade, o software foi submetido a uma interação com um jogo eletrônico de demonstração – este solicitando a composição de um tema específico em tempo de execução. O sistema se comportou conforme esperado, recebendo a requisição do jogo e logo em seguida oferecendo o resultado da composição no formato MIDI.

Uma inclusão interessante para o framework seria acoplar um software sintetizador para gerar uma saída em um formato como o MP3. Um arquivo MIDI é sintetizado na placa de som e sua qualidade é dependente da qualidade da placa de som de onde o arquivo está sendo executado. Com uma saída em MP3, essa dependência na qualidade não mais existiria.

Há inúmeros meios de se reproduzir o processo de composição musical, sendo este interpretado de maneiras diferentes pelas distintas culturas presentes ao redor do mundo. Por mais variados que sejam os métodos criativos, caso estes possam ser simulados computacionalmente, o framework de composição musical algorítmica estará disposto a reuni-los e torná-los uma só representação artística.

### Referências

- Belinkie, M. (1999) “Video game music: not just kid stuff”, <http://www.vgmusic.com/vgpaper.shtml>, Janeiro.
- Costalonga, L. et al. (2005) “Bibliotecas Java Aplicadas a Computação Musical”, <http://gsd.ime.usp.br/sbcm/2005/papers/tech-12452.pdf>, Janeiro.
- Delgado, M., Fajardo, W. e Molina-solana, M. (2008) “Inmamusys: Intelligent multiagent music system”, In: Expert Systems with Applications, v. 36, p. 4574-4580.
- Huber, D. (2007) “The MIDI manual: a practical guide to MIDI in the Project Studio”, In: Burlington, Inglaterra, Elsevier, 2007. 362 p.
- Impro-Visor. (2010) “Impro-Visor”, <http://www.cs.hmc.edu/~keller/jazz/improvisor/>, Março.
- Kirke, A. e Miranda, E. (2009) “A Survey of Computer Systems for Expressive Music Performance”, In: ACM Computing Surveys, v. 42, n.1.
- Kurzweil, R. (2002) “Kurzweil inducted into National Inventors Hall of Fame”, <http://www.kurzweilai.net/meme/frame.html?main=/articles/art0467.html>, Janeiro.
- Mello, B. A. (2001) “Modelagem e simulação de sistemas”, In: Graduação (Bacharelado em Ciência da Computação) URI – Unidade Regional Integrada , Rio Grande do Sul.
- MENEZES, J. M. A. (2008) “Composição algorítmica de fugas ao estilo de J. S. Bach”, In: Graduação (Bacharelado em Ciência da Computação) UnB - Universidade de Brasília, Brasília.
- Molina-solana, M. (2009) “INMAMUSYS”, <http://www.ugr.es/~miguelmolina/inmamusys.php>, Março.