

ARTIGO ORIGINAL

Planejamento composicional de *Trivium* para flauta doce

Helder alto

Liduíno Pitombeira 

Universidade Federal do Rio de Janeiro | Brasil

Resumo: Este artigo descreve o planejamento composicional de *Trivium*, para flauta doce Helder alto, articulando três referenciais associados ao *Trivium*: Gramática, Retórica e Lógica. No primeiro movimento, dois programas em Python implementam uma gramática chomskyana do Tipo 3 para gerar *strings* de classes de alturas submetidas à seleção, aplicação, complementação e polimento composicional. No segundo, um sistema composicional transforma um motivo inicial por meio de figuras retóricas, com retroalimentação heurística. No terceiro, operações booleanas derivadas do planejamento de *Herma*, de Xenakis, são reutilizadas para produzir conjuntos de classes de alturas organizados em seções formais contrastantes. O estudo propõe uma articulação aplicada desses referenciais em um processo composicional documentado. A contribuição principal é metodológica: demonstrar como modelos formais distintos podem ser convertidos em procedimentos operacionais, submetidos à curadoria composicional e filtrados pelas condições concretas da escrita instrumental. Além disso, o artigo oferece subsídios para o ensino e a prática do planejamento composicional.

Palavras-chave: Planejamento composicional, Sistemas composicionais, Flauta doce Helder alto, *Trivium*, Composição.

Abstract: This article describes the compositional planning of *Trivium*, for Helder alto recorder, by articulating three referential frameworks associated with *Trivium*: Grammar, Rhetoric, and Logic. In the first movement, two Python programs implement a Type-3 Chomskyan grammar to generate strings of pitch classes, which are subjected to selection, application, complementation, and compositional polishing. In the second movement, a compositional system transforms an initial motive through rhetorical figures, using heuristic feedback. In the third movement, Boolean operations derived from the planning of Xenakis's *Herma* are recontextualized to produce pitch-class sets organized into contrasting formal sections. The study proposes an applied articulation of these frameworks within a documented compositional process. Its main contribution is methodological: it demonstrates how distinct formal models can be converted into operational procedures, subjected to compositional curation, and filtered through concrete conditions of instrumental writing. In addition, the article provides resources for the teaching and practice of compositional planning.

Keywords: Compositional Planning, Compositional Systems, Helder Alto Recorder, *Trivium*, Composition.

Este trabalho descreve o planejamento composicional¹ da obra *Trivium*, para flauta doce Helder alto, partindo de três referenciais teóricos intimamente associados ao conceito de Trivium. O Trivium (Joseph, 2002), assim como o Quadrivium (Martineau, 2010), é um conceito que remonta à tradição educacional da Idade Média, especialmente no contexto da educação clássica nas sete artes liberais, que formavam a base do currículo educacional naquele período. O Trivium consiste em três disciplinas fundamentais relacionadas ao desenvolvimento das habilidades de linguagem e comunicação: 1) Gramática, que se relaciona à compreensão da estrutura e função da linguagem; 2) Retórica, ou seja, a arte da expressão persuasiva e eficaz, tanto verbalmente quanto por escrito; e 3) Lógica, associada ao desenvolvimento do pensamento lógico e do raciocínio. O Quadrivium, por sua vez, é composto por quatro disciplinas voltadas para o desenvolvimento de habilidades matemáticas: 1) Aritmética: estudo dos números e das operações fundamentais; 2) Geometria: exploração das propriedades do espaço e das figuras; 3) Astronomia: estudo dos corpos celestes e dos padrões no céu; e 4) Música: compreensão das relações matemáticas inerentes à música. A música é a disciplina que trata da manifestação do número no tempo. Essas sete disciplinas eram consideradas fundamentais para a educação de um estudante na tradição clássica e medieval.

Esses referenciais serão abordados aqui em uma perspectiva contemporânea. O uso da Gramática no planejamento composicional se inspira essencialmente no texto fundamental de Roads e Wieneke (1979), que trata da Gramática sob uma perspectiva musical, bem como nos textos de Menezes (2000), McCormack (1996) e Holtzman (1980). Com relação à Retórica, embora o texto referencial aqui utilizado (Bartel, 1997) explore fundamentalmente a música do barroco alemão, o uso das figuras retóricas será implementado como uma rede transformacional que parte de um gesto inicial, com o intuito de gerar um repositório composicional. Já a Lógica será empregada nos moldes propostos por Xenakis (1991) para a composição de sua obra *Herma*, ou seja, utilizando Álgebra Booleana.

¹ O estudo do planejamento composicional integra o projeto de pesquisa no Programa de Pós-Graduação em Música da Universidade Federal do Rio de Janeiro (UFRJ), a partir do qual se ramificam os diversos projetos de mestrado, doutorado, pós-doutorado, bem como os planos de trabalho de iniciação científica. O projeto pode ser acessado no link <https://ppgm.musica.ufrj.br/planejamento-composicional-reflexoes-teorico-esteticas-no-ambito-de-uma-producao-artistica-formalizada-2023/>.

As informações relativas à flauta doce Helder alto provêm tanto do trabalho de Fröhlich (2020) como dos diálogos com a flautista Francielle Paixão, que encomendou *Trivium*, como parte de seu mestrado em performance musical na Escola Superior de Artes Aplicadas, em Castelo Branco, Portugal. A instrumentista preparou uma série de vídeos demonstrando tanto o uso tradicional do instrumento como o uso de técnicas estendidas.

Neste artigo, a flauta doce Helder alto é tratada como condição instrumental do planejamento composicional: extensão, registro, articulação, respiração, frulato, voz e multifônicos atuam como filtros de aplicação, complementação e polimento dos materiais gerados. O *Trivium* é abordado como modelo de articulação entre três operações composicionais: Gramática, como geração de cadeias; Retórica, como transformação e reinterpretação gestual; e Lógica, como seleção e organização de conjuntos. No campo das poéticas da criação musical, o estudo enfatiza que a complexidade do referencial empregado não determina a complexidade perceptível do texto musical: referenciais sofisticados podem gerar superfícies simples, rarefeitas ou econômicas, enquanto procedimentos simples podem produzir resultados densos. A contribuição metodológica consiste em integrar geração algorítmica, transformação retórica, seleção lógica, curadoria composicional e mediação performática em um planejamento documentado e verificável.

1. Referenciais teóricos

Nesta seção serão examinados os conceitos de Gramática (1.1), Retórica (1.2) e Lógica (1.3), sob uma perspectiva musical. O objetivo é fornecer uma base teórica mínima consistente para que o(a) leitor(a) possa navegar com tranquilidade na seção seguinte (2), na qual será desenvolvido o planejamento composicional da obra *Trivium*.

1.1 Gramática

A relação entre gramática e música é uma área rica e interdisciplinar que tem atraído a atenção de musicólogos, linguistas e teóricos da música. Enquanto a gramática está intrinsecamente ligada à estrutura e à organização da linguagem, a música possui suas próprias estruturas complexas que, em

alguns casos, mostram paralelos com as estruturas linguísticas. São pontos de interseção importantes: 1) a sintaxe musical (aplicada em nível melódico, harmônico, textural etc.), que, quando empregada analiticamente, tenta identificar padrões recorrentes e estruturas formais na busca da determinação analítica de um modelo organizado e coeso;² e 2) a análise de estruturas usando conceitos gramaticais, como frase, sentença, período etc., para compreender a organização hierárquica na música.³ Estudos recentes, considerando os últimos cinquenta anos, nas áreas de computação e inteligência artificial (Briot, 2020), Cognição musical (Zbikowski, 2017) e Música Gerativa (Lerdahl; Jackendoff, 1983), de certa forma aproximam música e gramática.

Como afirma McCormack (1996, p.326), o uso de gramáticas na composição musical não é algo novo. Diversos autores do final da década de 1970 contribuíram de alguma maneira para essa abordagem, entre eles: Buxton *et al* (1978), que exploram o uso de gramática para a produção de música eletroacústica, propondo diversos algoritmos na linguagem de programação C; Roads e Wieneke (1979), que descrevem a tipologia gramatical de Chomsky aplicada à música e apresentam as contribuições de diversos autores para esse campo;⁴ Holtzman (1980), que criou um compilador específico para a aplicação da tipologia gramatical de Chomsky em composição musical; e Jones (1981), que aborda os processos estocásticos e também realiza uma minuciosa descrição de gramáticas formais. O assunto continua sendo considerado como um importante método para a geração de materiais composicionais, como se verifica pela inclusão de uma seção sobre gramáticas formais no livro de Miranda (2001) e pelo denso quarto capítulo sobre gramática gerativa no livro de Nierhaus (2009). Mais recentemente, um curso sobre música gerativa no *Youtube*, em que o nono episódio aborda especificamente gramáticas gerativas aplicadas à composição musical, foi disponibilizado por Velardo (2023).

² A sintaxe musical também pode ser utilizada de maneira prescritiva, como será visto na seção 2 deste trabalho, que descreve detalhadamente uma sintaxe de conexão entre classes de alturas.

³ Da mesma forma que a sintaxe, a estruturação musical também pode ser empregada de maneira prescritiva. Isso é especialmente importante na abordagem *top-down* de planejamento (Miranda, 2001), na qual a estrutura é arquitetonicamente projetada em diversos níveis, desde os movimentos, as seções, os agrupamentos de frases (períodos, sentenças, cadeias, grupos etc.), as frases, semifrases (quando for o caso), as figuras (motívicas e auxiliares), até o nível das notas.

⁴ O texto de Roads e Wieneke traça breves reflexões sobre os trabalhos de Ruwet, Nattiez, Laske, Smoliar, Winograd, Moorer, Lerdahl e Jackendoff, entre outros.

Roads e Wieneke (1979) definem gramática musical como uma representação simbólica, diferenciada, portanto, de uma representação icônica.⁵ Esses autores consideram (1979, p.48) que as partituras são representações intermediárias, que se situam entre os ícones e os símbolos. Assim, as partituras gráficas e as tablaturas seriam mais próximas dos ícones enquanto as que usam notação tradicional seriam mais próximas dos símbolos.⁶ McCormack (1996, p. 322) acrescenta que “fórmula de compasso, claves e barras de compasso são simbólicas; codificação de altura, crescendo, diminuendo são icônicos”.⁷

Uma gramática gerativa (G) pode ser definida formalmente pela quádrupla $G = (N, T, P, S)$, na qual N é um alfabeto de símbolos não-terminais, ou seja, símbolos que representam as estruturas mais profundas (musicalmente: frases, sentenças, motivos etc.); T representa os símbolos terminais ou de superfície, tais como, em termos musicais, notas, durações, dinâmicas etc.; P são as regras de produção que funcionam substituindo termos localizados à esquerda (α) de uma expressão por termos localizados à direita (β), ou seja, $\alpha \rightarrow \beta$, como por exemplo $n \rightarrow \text{Dó}$ (aqui o símbolo n é substituído pelo Dó, que é um símbolo terminal, e portanto, não será substituído por mais nada, encerrando o ciclo operacional); S é o símbolo inicial, através do qual o processo de geração se inicia.

A especificação da gramática G é mostrada no QUADRO 1. Os símbolos não terminais (N) são S e D . Os símbolos terminais (T) são as notas de uma escala pentatônica. Nas regras de produção, S pode ser substituído por D ou DS e D é substituído, com igual probabilidade (distribuição homogênea), por uma das classes de alturas da escala pentatônica. A FIGURA 1 mostra o fluxo representativo de operacionalidade da gramática G . O ciclo só chega ao final quando atinge um estado em que só existem símbolos terminais, os quais não podem ser substituídos. O QUADRO 2

⁵ Sobre os conceitos de ícone, índice e símbolo na semiótica de Peirce ver Chandler (2007). Sobre semiótica musical ver Nattiez (1990).

⁶ Roads e Wieneke afirmam que “entre o icônico e o simbólico estão a maioria das partituras, que são representações intermediárias. Partituras gráficas e tablaturas tendem a ser mais icônicas, enquanto a notação de pauta tradicional contém elementos mais formais, como cabeças e hastes de notas, termos dinâmicos, acidentes e assim por diante. Em ambos os casos, normalmente há espaço para performance e interpretação analítica, em que o contexto ou macroestrutura, apenas inferido pela partitura, é articulado ou explicado. No original: “*standing between the iconic and the symbolic are most scores, which are intermediate representations. Graphic scores and tablature tend to be more iconic, while traditional stave notation contains more formal elements, such as note heads and stems, dynamic terms, accidentals, and the like. In both cases, there is typically room for performance and analysis interpretation, wherein the background or macrostructure, only inferred by the score, is articulated, or explicated*”.

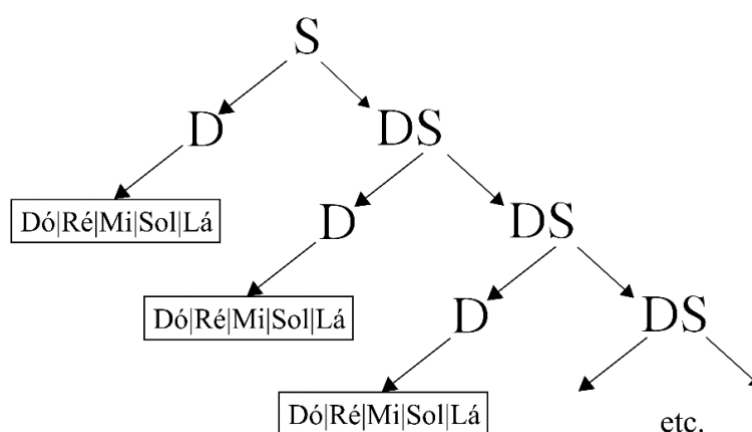
⁷ No original: “*time signature, clefs, bar lines are symbolic; pitch coding, crescendo, diminuendo are iconic*”.

apresenta um exemplo de ciclo operacional (que se distingue do fluxo mostrado na FIGURA 1) da gramática G gerando um motivo musical de quatro classes de alturas: Dó, Mi, Lá, Mi. O Apêndice 1 (p. 54) apresenta um programa em *Python* que produz motivos musicais de acordo com a gramática G .

QUADRO 1 – Especificação da gramática G

$G = (N, T, P, S)$
$N = \{S, D\}$
$T = \{Dó, Ré, Mi, Sol, Lá\}$
$P = \{S \rightarrow D, S \rightarrow DS, D \rightarrow Dó \mid Ré \mid Mi \mid Sol \mid Lá\}$

Fonte: Autor

FIGURA 1 – Fluxo representativo de operacionalidade da gramática G .

Fonte: Autor

QUADRO 2 – Exemplo de ciclo operacional da gramática G gerando um motivo musical de quatro classes de alturas.

S	Início
DS	S é substituído por DS
Dó S	D é substituído por Dó e o S se mantém
Dó DS	S é substituído por DS
Dó Mi S	D é substituído por Mi e o S se mantém
Dó Mi DS	S é substituído por DS
Dó Mi Lá S	D é substituído por Lá e o S se mantém
Dó Mi Lá D	S é substituído por D
Dó Mi Lá Mi	D é substituído por Mi – fim do processo

Fonte: Autor

Chomsky (1959) propôs uma tipologia de gramáticas gerativas com quatro tipos básicos. O **Tipo 3**, conhecido como gramática de estado finito (ou regular), é o mais restritivo dos quatro. Nesse tipo de gramática somente um símbolo não terminal pode aparecer em ambos os lados de uma regra de produção. As regras de produção de uma gramática regular são da forma: $A \rightarrow aB$, $A \rightarrow a$, $A \rightarrow \varepsilon$, onde A e B são símbolos não-terminais, a é um símbolo terminal (um símbolo da linguagem que não pode ser expandido) e ε representa a cadeia vazia (nada).

Cada uma das regras é explicada da seguinte maneira:

1. $A \rightarrow aB$: Isso significa que quando se encontra o símbolo não-terminal A , ele pode ser substituído por um terminal a seguido pelo não-terminal B .
2. $A \rightarrow a$: Indica que o símbolo não-terminal A pode ser substituído diretamente pelo terminal a .
3. $A \rightarrow \varepsilon$: Isso significa que o não-terminal A pode ser eliminado, gerando uma cadeia vazia ε .

Em termos intuitivos, uma Gramática do Tipo 3 pode ser compreendida como um mecanismo de encadeamento linear: a cada passo, o sistema produz um terminal e decide se continua ou encerra a cadeia. Em um exemplo linguístico elementar, seria como uma regra que permite formar sequências simples do tipo “artigo + substantivo + verbo”, sem exigir dependências hierárquicas profundas entre partes distantes da frase. Em música, esse tipo de gramática é particularmente útil para gerar linhas ou segmentos sucessivos: uma nota, um intervalo, uma célula ou uma pequena unidade motívica conduz à próxima unidade segundo regras locais. Um exemplo próximo é uma cadeia markoviana de acordes em que cada acorde “decide” probabilisticamente o acorde seguinte, sem conservar necessariamente a memória de acontecimentos mais distantes. Nesse sentido, a Gramática Tipo 3 pode ser pensada, em termos didáticos, como uma gramática de memória curta: ela é eficiente para produzir continuidade linear, mas limitada para representar estruturas frasais aninhadas, dependências de longa distância ou macroformas de múltiplos níveis.⁸

⁸ Conforme Roads e Wieneke (1979, p. 50), ao comentarem a tipologia de Chomsky, os processos de Markov podem ser caracterizados como gramáticas do Tipo 3, cuja limitação principal, no contexto da representação musical, está na dificuldade de lidar com estruturas hierárquicas ou frasais. Um exemplo de gramática do Tipo 3 será detalhadamente mostrado na Seção 2.1, durante o planejamento composicional do primeiro movimento de *Trivium*, intitulado *Gramática*.

A gramática do **Tipo 2**, também chamada Gramática Livre de Contexto (GLC), tem regras de produção mais flexíveis do que as gramáticas do **Tipo 3** (regulares) e é mais poderosa em termos de expressividade: consegue lidar com estrutura de frase e operacionalizar estruturas hierárquicas. As regras de produção do **Tipo 2** têm a forma $A \rightarrow \gamma$, onde A é um símbolo não-terminal e γ é uma sequência de símbolos (terminais ou não-terminais).

A gramática apresentada no QUADRO 1 pode ser escrita no formato de uma gramática livre de contexto (**Tipo 2**), pois suas produções têm um único símbolo não terminal no lado esquerdo. No entanto, a linguagem efetivamente gerada por esse exemplo é simples: consiste em cadeias finitas de notas tomadas do alfabeto {Dó, Ré, Mi, Sol, Lá}, sem dependências globais de contagem, espelhamento ou aninhamento. Por isso, embora a formulação adotada seja compatível com o formato do Tipo 2, a linguagem resultante pertence ao subconjunto regular e poderia também ser descrita por uma gramática do Tipo 3 equivalente. Um exemplo mais característico de gramática do Tipo 2, baseado em espelhamento palindrômico, é mostrado no Apêndice 2, juntamente com um programa em Python.⁹

Intuitivamente, uma Gramática do Tipo 2 já permite pensar em estruturas compostas por níveis internos. Em vez de apenas acrescentar um elemento depois do outro, ela pode fazer com que uma unidade maior gere unidades menores: uma frase pode gerar semifrases, uma semifrase pode gerar motivos, e um motivo pode gerar notas. Em um exemplo linguístico simples, uma oração pode ser descrita como a combinação de um sintagma nominal e um sintagma verbal, cada um deles podendo se desdobrar em outras unidades. Em música, isso corresponde à possibilidade de modelar estruturas hierárquicas, como frases constituídas por motivos, períodos formados por antecedente e consequente, ou formações palindrômicas e aninhadas. Esse desdobramento em árvore é precisamente o tipo de organização que a Gramática Livre de Contexto modela com mais clareza do que uma gramática regular. O exemplo do Apêndice 2, baseado em espelhamento, mostra essa propriedade em escala reduzida.

⁹ Em termos formais, há uma distinção entre o formato das regras e a classe da linguagem gerada. A gramática do QUADRO 1 é escrita com produções compatíveis com uma gramática livre de contexto, mas a linguagem que ela gera é regular, pois corresponde a sequências finitas não vazias de notas sem dependências hierárquicas. Assim, existe uma gramática Tipo 3 equivalente capaz de gerar o mesmo conjunto de cadeias.

Na gramática do **Tipo 1** (sensível ao contexto), “em ambos os lados das regras de produção, um número arbitrário de sequências de símbolos terminais ou não terminais é possível, mas o número de símbolos no lado direito não deve ser menor que o número do lado esquerdo” (Nierhaus, 2009, p. 88). O QUADRO 3 apresenta exemplos de regras de produção desse tipo de gramática. Observa-se que, em todas as regras listadas, a condição de não contração é satisfeita, isto é, para cada produção $\alpha \rightarrow \beta$ vale $|\alpha| \leq |\beta|$. Em consequência, o lado direito de cada regra mantém ou aumenta o comprimento da forma sentencial, de modo que o número de símbolos produzidos nunca é inferior ao número de símbolos substituídos. As produções listadas nesse quadro ilustram formatos típicos permitidos em gramáticas do **Tipo 1** (sensíveis ao contexto), respeitando a condição $|\alpha| \leq |\beta|$. Cada linha é um exemplo independente de produção e não se assume que o conjunto forme um sistema de regras conectado em uma única derivação.

QUADRO 3 – Exemplo independente de regras de produção em gramáticas do **Tipo 1**.

Regra de produção	Interpretação (leitura informal)	$ \alpha $	$ \beta $	Condição $ \alpha \leq \beta $?	Observação
$S \rightarrow a A B c$	o símbolo inicial S pode ser expandido para a A B c.	1	4	Sim	Expansão (a,b,c são terminais; A, B são não-terminais).
$A B \rightarrow X Y$	o bloco A B pode ser reescrito como X Y.	2	2	Sim	Reescrita sem contração.
$X \rightarrow p q$	o não-terminal X pode ser expandido para p q.	1	2	Sim	Expansão (p, q terminais).
$a X c \rightarrow a d c$	em contexto a _ c, o símbolo X pode ser reescrito como d.	3	3	Sim	Regra sensível ao contexto: X pode ser reescrito como d quando ocorre entre a e c, sem redução de comprimento.

Fonte: Autor

Para demonstrar operacionalmente esse princípio em um exemplo com interpretação musical, o programa mostrado no Apêndice 3 (p.57) gera uma cadeia terminal pertencente à linguagem $\{a^n b^n c^n \mid n \geq 1\}$, que é um exemplo clássico de linguagem sensível ao contexto. A execução apresentada produz $n = 3$, resultando na sequência de motivos “a a a b b b c c c”. Do ponto de vista formal, o valor de n controla uma dependência global: a cadeia final deve conter o mesmo número de ocorrências de a , b e c . Esse acoplamento triplo de contagens não é capturável por gramáticas

regulares (Tipo 3) e tampouco, em geral, por gramáticas livres de contexto (**Tipo 2**), motivando seu enquadramento no **Tipo 1**.

O traço de derivação listado pelo programa (Apêndice 3) e mostrado no QUADRO 4 evidencia as etapas fundamentais do mecanismo: parte-se de S (linha 00) e aplica-se repetidamente uma expansão que introduz um símbolo terminal a e dois marcadores auxiliares B e C (linhas 01–02), até que a derivação seja “fechada” pela regra base que introduz abc (linha 03). Nesse ponto, a forma sentencial contém blocos mistos com símbolos auxiliares e terminais, e o processo precisa reorganizar esses marcadores para obter a distribuição desejada. A etapa mostrada em “ $aaabcBBCC$ ” (linha 04) indica o reordenamento que posiciona todos os B antes dos C , permitindo que cada B seja convertido em b e cada C em c por regras dependentes do contexto (linhas 05–09). Ao final, chega-se à forma puramente terminal “ $aaabbbccc$ ” (linha 10), isto é, $a^3b^3c^3$. Assim, a derivação não apenas produz uma cadeia terminal, mas garante estruturalmente a igualdade das três contagens por meio de símbolos auxiliares que só podem ser eliminados/convertidos em contextos apropriados — característica típica de sistemas sensíveis ao contexto.

Por fim, para produzir uma saída musical concreta (mostrada na última linha do QUADRO 4), o programa interpreta cada símbolo terminal a , b e c como um motivo definido pelo(a) compositor(a) em termos de classes de alturas (por exemplo, $a \mapsto [0,2,4]$, $b \mapsto [7,5,4]$, $c \mapsto [9,11,0,2]$). A cadeia “ $aaabbbccc$ ” passa então a significar: repetir o motivo a três vezes, em seguida repetir o motivo b três vezes e, por fim, repetir o motivo c três vezes. A lista final de classes de alturas apresentada na saída do programa é precisamente a concatenação desses motivos, oferecendo um exemplo em que a gramática controla uma propriedade formal global (igualdade de contagens) e, ao mesmo tempo, produz um objeto musical observável (uma sequência de classes de alturas organizada em três blocos motivicamente caracterizados).

QUADRO 4 – Saída do programa que opera com a gramática do **Tipo 1**

n	3
Sequência de motivos	a a a b b b c c c
Derivação (formas sentenciais)	00: S 01: aSBC 02: aaSBCBC 03: aaabcBCBC 04: aaabcBBCC 05: aaabBcBCC 06: aaabBBcCC 07: aaabbBcCC 08: aaabbbccC 09: aaabbbccC 10: aaabbbccc
Classes de alturas	[0, 2, 4, 0, 2, 4, 0, 2, 4, 7, 5, 4, 7, 5, 4, 7, 5, 4, 9, 11, 0, 2, 9, 11, 0, 2, 9, 11, 0, 2]

Fonte: Autor

A Gramática do Tipo 1 pode ser entendida como uma gramática em que certas transformações só são permitidas em contextos específicos. Em termos linguísticos, isso se aproxima de situações em que a forma de uma palavra depende de seu ambiente, como ocorre na concordância entre sujeito e verbo ou entre substantivo e adjetivo. A regra não atua isoladamente sobre um símbolo, mas depende do que está ao seu redor. Em música, essa ideia pode ser intuída por comportamentos nos quais um material só se transforma quando ocupa determinada posição formal, aparece cercado por certos eventos ou satisfaz uma condição previamente estabelecida. Um exemplo tonal simples, entendido apenas como analogia, é o comportamento da sensível: sua tendência de resolução depende do contexto harmônico e contrapontístico em que ocorre, e não apenas de sua identidade como altura isolada. O exemplo apresentado no Apêndice 3, com a cadeia $a^n b^n c^n$, mostra essa propriedade de modo mais formal.

Por último, a Gramática do **Tipo 0**, ou livre,

é uma construção teórica que não impõe nenhuma restrição na forma de regras de produção. Isto permite que as *strings* intermediárias se expandam ou contraíam em comprimento, o que poderia produzir o curioso caso de uma "macroestrutura fantasma", isto é, uma macroestrutura à qual não corresponde nenhuma microestrutura! Além disso, uma gramática do Tipo 0, por definição, permite a produção de *strings* verdadeiramente

infinitas (enumeráveis recursivamente), bem como produções nulas, ambas as quais não fazem sentido musical (Roads; Wieneke, 1979, p.49-50).¹⁰

Para exemplificar didaticamente uma Gramática do **Tipo 0**, parte-se da mesma quádrupla $G = (N, T, P, S)$ e acrescenta-se um não-terminal funcional que permita contração e apagamento. Na gramática G_0 , $N = \{S, D, X\}$, $T = \{\text{Dó}, \text{Ré}, \text{Mi}, \text{Sol}, \text{Lá}\}$ e S é o símbolo inicial. Intuitivamente, S controla o crescimento da cadeia, D é um slot abstrato que ainda se transformará em nota de superfície, e X é um marcador que introduz o comportamento livre do **Tipo 0**. As regras são: $S \rightarrow DS \mid XS \mid D \mid X$, $D \rightarrow \text{Dó} \mid \text{Ré} \mid \text{Mi} \mid \text{Sol} \mid \text{Lá}$, $XD \rightarrow X$ e $X \rightarrow \varepsilon$. O programa em *Python* apresentado no Apêndice 4 tem função demonstrativa no âmbito da discussão teórica sobre a hierarquia de Chomsky. O *script* inicia a derivação a partir do símbolo S , aplica regras de expansão, terminação, contração e apagamento, e registra um *trace* passo a passo da forma sentencial. Para tornar visível a propriedade distintiva da Gramática do Tipo 0, o programa inclui a regra contrativa $XD \rightarrow X$, pela qual a forma sentencial tem seu comprimento reduzido, e a regra de apagamento $X \rightarrow \varepsilon$. A execução publicada utiliza *seed*=11, *max_passos*=60 e *minimo_notas*=7, produzindo a saída mostrada no Apêndice 5. Nessa saída, observa-se uma derivação terminal formada por nove notas e, no passo 09, uma ocorrência explícita da contração $XD \rightarrow X$.

Intuitivamente, a Gramática do Tipo 0 é a mais livre das quatro: suas regras podem expandir, contrair, substituir ou apagar partes da forma sentencial. Em um exemplo não musical, ela se aproximaria de um processo editorial radical, no qual um rascunho pode ganhar palavras, perder palavras, deslocar unidades ou eliminar trechos inteiros antes de chegar à versão final. Em música, essa liberdade permite imaginar procedimentos nos quais uma macroestrutura planejada não necessariamente se realiza integralmente na superfície, pois certos marcadores podem ser apagados ou contraídos durante a derivação. No presente artigo, o exemplo do Apêndice 4 é utilizado apenas de modo demonstrativo: sua função é tornar visível, em pequena escala, a possibilidade de contração e apagamento, propriedades distintivas desse tipo gramatical. O QUADRO 5 resume as características

¹⁰ No original: “A Type 0 or free grammar is a theoretical construct which imposes no restriction on the form of the production rules. This allows intermediate strings to either expand or contract in length, which could produce the curious case of a “phantom macrostructure”, i.e., a macrostructure to which there corresponds no microstructure! Further, a Type 0 grammar by definition allows the production of truly infinite (recursively-enumerable) strings as well as null productions both of which make no musical sense”.

das gramáticas chomskianas com as intuições operacionais, exemplos da linguística e da música bem como potenciais composicionais.

QUADRO 5 – Resumo dos tipos gramaticais chomskianos

Tipo gramatical	Intuição operacional	Exemplo linguístico	Exemplo musical	Potencial composicional
Tipo 3 — regular (estado finito)	Encadeamento linear; memória curta. Controla a ordem dos elementos, não a compatibilidade entre eles.	Sequência de fórmulas fixas (“bom dia”, “boa tarde”); o próximo item depende só do anterior imediato, sem verificar concordância.	Cadeia markoviana de acordes: Dó maior conduz a Fá maior ou Lá menor por regras locais, sem memória de eventos distantes.	Linhas melódicas, cadeias harmônicas, continuidade linear.
Tipo 2 — livre de contexto	Uma unidade maior gera unidades menores; organização hierárquica em árvore.	Uma oração gera sintagma nominal e sintagma verbal; cada sintagma desdobra-se em outras unidades.	Frase → semifrase → motivo → notas. Em escala maior: exposição de sonata → tema A + transição + tema B → frases → motivos.	Frases, períodos, palíndromos, formas em árvore. Limite: dependências simultâneas entre regiões distintas.
Tipo 1 — sensível ao contexto	Uma regra só atua no ambiente correto; verifica compatibilidade, não apenas ordem.	“O menino” é válido; “o menina” é bloqueado. Mesma sequência artigo + substantivo, mas agora com exigência de concordância.	Resolução da sensível depende do contexto harmônico. Blocos motivicamente distintos controlados por igualdade de contagem ($a^n b^n c^n$). Seção climática planejada pode ser comprimida a um acorde ou eliminada — gerando uma “macroestrutura fantasma” (Roads; Wieneke).	Dependências globais, simetria, concordâncias estruturais, transformações condicionadas.
Tipo 0 — irrestrita	Regras podem expandir, contrair, substituir ou apagar símbolos em qualquer ponto.	Processo editorial radical: acréscimos, cortes, substituições e apagamentos antes da versão final.		Modelo teórico de máxima expressividade. Cautela: pode gerar estruturas não realizáveis na superfície.

Fonte: Autor, a partir de Roads e Wieneke (1979)

1.2 Retórica

Aristóteles (1857, p.11) define Retórica como “uma faculdade de considerar todos os meios possíveis de persuasão sobre cada assunto”.¹¹ A revolucionária contribuição de Aristóteles para a Retórica está na maneira meticulosa como incorpora os três elementos essenciais do discurso: *Ethos*, *Pathos* e *Logos* (Meyer, 1999, p.47). O *Ethos* refere-se à credibilidade, autoridade e ética do orador. É a persuasão baseada na confiança que o público deposita no caráter, no conhecimento, na experiência e na integridade do falante. O *Pathos* apela às emoções do público. É a persuasão através da criação de

¹¹ No original: “a faculty of considering all the possible means of persuasion on every subject”.

uma conexão emocional, buscando provocar sentimentos que podem influenciar as atitudes e decisões. O *Logos* representa a persuasão baseada na lógica, na razão e nos argumentos racionais. A interação eficaz entre *ethos*, *pathos* e *logos* é essencial para criar discursos persuasivos. Um discurso equilibrado pode envolver a construção de credibilidade (*ethos*) para ganhar a confiança do público, apelar às emoções (*pathos*) para criar empatia e conexão, e fornecer argumentos lógicos e racionais (*logos*) para fundamentar as reivindicações.

As conexões entre música e retórica podem ser identificadas desde a Antiguidade.¹² Como afirma McCreless (2007, p. 847),

Dado que a retórica compartilha com a música o desdobramento estruturado do som no tempo, aspectos de performance e apresentação, e até mesmo uma noção rudimentar de “obra” (a oração na retórica, a composição na música), era natural e até mesmo inevitável que analogias fossem estabelecidas entre ambas.¹³

A maior aproximação entre retórica e música, chegando mesmo a um estágio de verdadeira simbiose, ocorreu provavelmente entre a segunda metade do século XVI e o final do século XVIII,¹⁴ sendo a produção musical do barroco alemão especialmente influenciada pela retórica. Dois textos fundamentais nesse campo são a *Musica Poetica* de Burmeister, originalmente publicado em 1606, e *Der vollkommene Capellmeister (O Mestre-de-Capela Perfeito)*, de Johann Mattheson, publicado em 1739. O primeiro, examinado por Bent e Pople (2011), traz o que se considera hoje ser a primeira análise musical: Burmeister analisa o moteto *In me transierunt*, de Orlando de Lassus. Palisca (1994) discorre de maneira aprofundada sobre essa análise, inteiramente desenvolvida sob uma perspectiva retórica. O segundo “contém a exposição mais detalhada disponível acerca da *musica poetica*” (Lucas,

¹² Quintiliano (35-95 d.C), em seu tratado *Institutio Oratoria*, redigido em 12 livros, afirma que “a música é uma necessidade até para um orador”. No original: “*music will be a necessity even for an orator*” (p. 165). Palisca (1994, p.287) comenta que o tratado de Quintiliano “difundiu a ideia de que a música está intimamente ligada à oratória e que, tal como a oratória, tem a função de levar os ouvintes a diversas paixões”. No original: “*It spread the idea that music is closely allied to oratory and that, like oratory, it has the function of moving listeners to various passions*”.

¹³ No original: “*Given that rhetoric shares with music the structured unfolding of sound in time, aspects of performance and delivery, and even a rudimentary notion of the “work” (the oration in rhetoric, the composition in music), it was natural and even inevitable that analogies would be drawn between the two.*”

¹⁴ Segundo Versolato e Kerr (2008, p.65), “a redescoberta, em 1416, da *Institutio Oratoria*, de Quintiliano, constituiu uma das principais fontes para o desenvolvimento do processo de simbiose entre a retórica e a música ocorrida no século dezesseis”.

2016, p. 76), termo que designava anteriormente a arte da composição musical. O primeiro texto, portanto, é de natureza descritiva (analítica) e o segundo de natureza prescritiva (composicional).

Mattheson apresenta um plano racional de composição musical com base na estruturação retórica (Persone, 1991, p.31-37). Essa estruturação, de uma maneira geral, como explica Bartel (1997, p.68), inclui cinco partes: 1) *Inventio* (determinação do tema); 2) *Dispositio* (organização do material); 3) *Elocutio* (conversão de ideias em palavras e sentenças); 4) *Memoria* (retenção de material); e 5) *Pronunciatio* (polimento e apresentação do texto).

A segunda parte, *dispositio*, por sua vez, se divide em seis subpartes: 1) *Exordium* (introdução); 2) *Narratio* (apresentação do fato); 3) *Propositio* (apresentação do argumento principal); 4) *Confirmatio* (argumentos de apoio); 5) *Confutatio* (argumentos contrários); e 6) *Peroratio* (conclusão).

A terceira parte, *elocutio*, trata de quatro aspectos ou virtudes de elocução: 1) *puritas*, que trata da sintaxe correta; 2) *perspicuitas*, que trata da clareza; 3) *ornatus*, no qual se aplicam as figuras retóricas, que é uma importante área da retórica musical; e 4) *aptum*, que trata da adequação da forma ao conteúdo.

Superficialmente, pode-se conjecturar que a estruturação do discurso musical partindo de preceitos retóricos se aproxima fortemente de uma estruturação formal em dois níveis: um nível macroestrutural,¹⁵ no qual atua o *dispositio* com suas seis subdivisões; e um nível microestrutural, no qual se implementa o *elocutio*, particularmente no que diz às figuras retóricas (Bartel, 1997), um campo teórico que apresenta uma taxonomia já bastante desenvolvida.

No entanto, sobre esse uso estrutural da retórica, Bonds (1991, p. 5) comenta:

O termo “retórica” tem sido utilizado tão amplamente nos últimos anos, muitas vezes indiscriminadamente, que é melhor afirmar desde o início o que um conceito retórico de forma musical não implica. Não deve ser diretamente equiparado à conhecida, mas amplamente mal compreendida, tentativa de Mattheson de estabelecer paralelos entre a forma de um movimento musical e a estrutura de uma oração (exórdio, narração, proposição, e assim por diante). (...) Basta dizer, por enquanto, que o esboço de Mattheson representa apenas uma manifestação de uma concepção retórica de forma. O conceito retórico de forma também não se refere ao uso de “figuras” que, quando dispostas em uma

¹⁵ O texto de Persone (1991) traduz para a língua portuguesa um exemplo de aplicação macroestrutural de Mattheson.

ordem sequencial, poderiam criar um todo em grande escala.¹⁶

Na mesma linha, Zampronha (2006, p. 49), para quem “a retórica é um tipo especial de construção em que os ouvintes são levados a interpretar de forma diferente o que foi ouvido anteriormente”,¹⁷ afirma que retórica e construção formal são duas categorias totalmente diferentes, embora seja possível uma intervenção intensa da retórica no sentido de subverter paradigmas formais gerando, como consequência, outras estruturas.¹⁸ Para Zampronha, a retórica musical retoma sua atividade no século XX como uma alternativa humanística aos paradigmas composicionais associados ao estruturalismo, aleatoriedade e música algorítmica.

Vê-se, portanto, que tanto Zampronha como Bonds desencorajam um uso estruturalista dos dispositivos retóricos na composição musical, sobretudo quando a retórica é tomada como molde formal totalizante ou como simples justaposição sequencial de figuras, seja no plano macroestrutural da *dispositio*, seja no plano microestrutural das figuras retóricas. Essa ressalva é metodologicamente importante para o presente trabalho. A retórica não será aqui empregada como reconstrução histórica da *dispositio* barroca, nem como modelo completo de forma musical. Seu uso será mais restrito e operacional: as figuras retóricas serão tratadas como operadores transformacionais aplicados a um motivo inicial, gerando variantes que passam posteriormente por avaliação, seleção, complementação e polimento composicional. Assim, a perspectiva estrutural adotada neste artigo não identifica retórica e forma em larga escala; ela utiliza determinadas figuras como procedimentos locais de geração e reinterpretação gestual. Essa apropriação operacional encontra paralelo em práticas composicionais contemporâneas, como a de Antunes (2014), que emprega a anadiplose como princípio estrutural em uma obra eletroacústica, demonstrando que uma figura retórica pode ser mobilizada formalmente no âmbito da composição contemporânea.

¹⁶ No original: “The term ‘rhetoric’ has been used so widely in recent years—often indiscriminately—that it is best to state at the outset what a rhetorical concept of musical form does not entail. It is not to be equated directly to Mattheson’s well-known but widely misunderstood attempt to draw parallels between the form of a musical movement and the structure of an oration (exordium, narratio, propositio, and so on). (...) Suffice it to say for the moment that Mattheson’s outline represents only one manifestation of a rhetorical conception of form. Nor does the rhetorical concept of form refer to the use of ‘figures’ which, when arrayed in a sequential order, could create a large-scale whole”.

¹⁷ No original: “Rhetoric is a special kind of construction in which listeners are led to interpret differently what had been previously listened to”.

¹⁸ “Rhetoric can be so predominant in a piece that it can subvert some formal paradigms and may generate other formal designs” (Zampronha, 2006, p.54).

A discussão de Bonds sobre retórica acontece particularmente no âmbito do estudo da forma. Ele descreve duas atitudes, ambas legítimas, com relação à forma. A primeira se associa ao que se denomina forma exterior (*outer form*) e diz respeito a características estruturais comuns entre diversas obras. Um grupo de obras classificadas como forma sonata, por exemplo, se situa nesse campo da forma exterior. Nessa primeira atitude, forma e conteúdo se diferenciam. A segunda atitude trata da forma interior (*inner form*) e se concentra na estrutura única de uma obra específica. Nessa atitude não há distinção entre forma e conteúdo.¹⁹

A retórica musical desapareceu depois de 1800 e o romantismo passou a adotar uma postura organicista, na qual a obra musical é vista como “um organismo e sua forma como uma relação orgânica de partes individuais com o todo” (Bonds, 1991, p. 4).²⁰ A retórica retomou gradualmente sua atividade nas artes em torno de meados do século XX por uma via semiótica, afastando-se de preceitos antigos. Como afirma Piedade (2012, p. 2-3): “menos voltada à produção de discurso e mais dedicada à sua interpretação, se afasta da oratória dos antigos e se aproxima da leitura semiótica”.²¹

Um importante movimento em prol da renovação do interesse pela retórica no século XX²² foi a Nova Retórica, que se refere a uma abordagem renovada e contemporânea ao estudo da retórica, desenvolvida principalmente por Chaïm Perelman e Lucie Olbrechts-Tyteca (2008) em sua obra seminal *Tratado da Argumentação: A Nova Retórica*. Esse trabalho foi publicado pela primeira vez em 1958. A Nova Retórica difere das abordagens tradicionais à retórica, que muitas vezes se concentram na busca de regras rígidas para a produção de discursos persuasivos. Em vez disso, Perelman e Tyteca propuseram uma abordagem mais descritiva, observando como os argumentos são construídos na prática e como eles podem ser eficazes em situações específicas. Diversos outros autores, no entanto, contribuíram para o estudo da retórica e da argumentação no século XX. Meyer (1999) apresenta um quadro desses principais autores (QUADRO 6) separando-os em duas

¹⁹ Esses conceitos se assemelham às abordagens composicionais conhecidas como top-down e bottom-up (Miranda, 2001).

²⁰ No original: “(...) *the musical work as an organism and its form as an organic relationship of individual partes to the whole*”.

²¹ A aproximação da retórica com a semiótica ocorre no âmbito do Grupo μ , fundado por semioticistas belgas em 1967.

²² A retórica geral, enquanto disciplina conectada principalmente ao Direito, nunca saiu de atividade como área de conhecimento. Meyer (1999) traça uma cronologia da retórica que vai interruptamente do séc. V a.C até o final do século XX.

dimensões.²³ A primeira se concentra na ênfase em um dos três aspectos fundamentais da retórica: autor (*ethos*), regras (*logos*) e recepção (*pathos*); a segunda diz respeito a um enfoque retórico ou argumentativo.

QUADRO 6 – As principais correntes da retórica e da argumentação no século XX.

	Ethos	Logos	Pathos
Retórica	Burke (1950)	Ducrot (1972) Grupo μ (1970)	Richards (1936) A retórica americana (Weaver, 1950)
Argumentação	Habermas (1968) Teoria da Comunicação e dos Atos de Linguagem (Searle, 1969)	Perelman (1958) Toulmin (1958)	Gadamer (1960) e a Teoria Hermenêutica da Recepção (Iser, 1976; Jauss, 1977)

Fonte: (Meyer, 1999, p. 255)

1.3 Lógica

Segundo Mortari (2001, p. 2), a Lógica “é a ciência que estuda princípios e métodos de inferência, tendo o objetivo principal de determinar em que condições certas coisas se seguem (são consequências), ou não, de outras”. A Lógica pode ser dividida em diversas categorias, dentre elas: Proposicional, LPO (Lógica de Primeira Ordem), Temporal, Modal Alética, Multivalorada, Fuzzy, Intuicionista, Relevante, Linear, Categórica e Matemática. A Lógica Matemática, por sua vez, pode ser dividida em Teoria dos Modelos, Teoria da Demonstração (*Proof Theory*), Teoria dos Conjuntos, Teoria da Recursão, e Álgebra (ou Lógica) de Boole. Essa última será o foco de trabalho desta seção. Desta forma, neste artigo, não tratarei das demais categorias.

A álgebra de Boole é uma subdivisão da Lógica Matemática que fornece uma estrutura para formalizar o comportamento lógico de proposições, utilizando operações como conjunção (E), disjunção (OU) e negação (NÃO), além de duas constantes, 0 (falso) e 1 (verdadeiro). A álgebra booleana é vista, portanto, como uma estrutura algébrica que generaliza o comportamento lógico. Há uma forte conexão entre a álgebra booleana, lógica proposicional, Teoria dos Conjuntos e

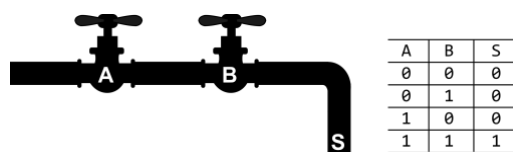
²³ Como o QUADRO 6 é reproduzido de Meyer (1999, p.255) as referências bibliográficas completas dos autores nele listados encontram-se na obra de Meyer.

circuitos lógicos computacionais. As operações básicas da álgebra booleana são: negação, conjunção, disjunção, disjunção exclusiva, condicional e bicondicional. Falarei em maior profundidade somente das três primeiras, uma vez que somente elas serão utilizadas no planejamento composicional do terceiro movimento de *Trivium*.

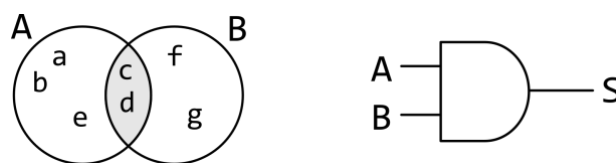
A negação ($\neg$) é uma operação lógica que inverte o valor de verdade de uma proposição. Se uma proposição é verdadeira (V ou 1), a negação será falsa (F ou 0), e vice-versa. A negação de A é representada por $\neg A$ ou $\bar{A}$. A conjunção ($\wedge$) é uma proposição composta que parte de duas proposições (A e B) e resulta verdadeira se, e somente se, A e B forem verdadeiras. Essa operação é denotada por $A \wedge B$ ou $A.B$. Uma representação prática dessa operação é uma estrutura hidráulica, composta por duas torneiras, A e B , em série, que podem estar abertas (1) ou fechadas (0), e cuja saída (S) é 1 quando a água flui e 0 quando a água é represada (FIGURA 2). Nesse modelo hidráulico a água só flui para a saída (S) quando as duas torneiras estiverem abertas. Em termos de lógica, A e B só resulta em 1, quando ambas as proposições, A e B , forem 1.

A lógica desse processo pode ser representada por uma estrutura denominada tabela de verdade, mostrada ao lado das torneiras, que indica todas as possibilidades de valores para cada uma das proposições. Observe que $S = 1$ somente quando $A = 1$ e $B = 1$. Essa operação equivale, na teoria dos conjuntos, à operação de Interseção ($\cap$) e na eletrônica computacional à porta lógica E . Essas duas representações são mostradas na FIGURA 3. Observe que, na interseção é testada uma condição semelhante ao E para cada elemento do universo de discurso: o elemento pertence a ambos os conjuntos? Se sim, ele faz parte da interseção (região cinza da figura), assim como uma proposição composta por um E só é verdadeira se ambas as proposições envolvidas forem verdadeiras. Formalmente, $A \cap B = \{x \mid x \in A \text{ e } x \in B\}$. No caso do exemplo mostrado na FIGURA 3, $A \cap B = \{c, d\}$.

FIGURA 2 – Operação de conjunção (E) representada por um sistema hidráulico



Fonte: Autor

FIGURA 3 – Operação de interseção ($\cap$) e porta lógica E .

Fonte: Autor

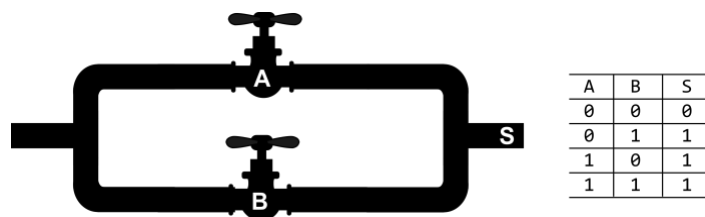
A disjunção (OU) é uma operação que resulta em verdadeiro se pelo menos uma das proposições for verdadeira. Apenas será falsa se ambas as proposições forem falsas. No modelo hidráulico, as torneiras estão em paralelo, de tal forma que basta que uma delas esteja aberta para que a água flua e S seja 1. A FIGURA 4 mostra o modelo hidráulico e a tabela de verdade. Essa operação equivale à operação de União (U) e na eletrônica computacional à porta lógica OU (FIGURA 5). Ela é formalmente representada por $A \vee B$ ou $A + B$.

A operação de disjunção exclusiva ($\oplus$) resulta em verdadeiro apenas se uma das proposições (A e B) for verdadeira, mas não ambas. Se ambas forem verdadeiras ou ambas forem falsas, o resultado é falso. Por sua vez, a operação condicional ($\rightarrow$) é falsa somente se A for verdadeira e B for falsa. Já a bicondicional ($\leftrightarrow$) é verdadeira apenas quando as duas proposições têm o mesmo valor lógico. As tabelas de verdade para essas três operações são mostradas na FIGURA 6.

As operações básicas podem se combinar para formar sentenças lógicas complexas. Exemplifico essa possibilidade definindo inicialmente três conjuntos de classes de alturas, $X = \{11, 0, 2, 4, 5, 7, 8, 9\}$, $Y = \{4, 5, 8, 1, 10\}$ e $Z = \{9, 5, 7, 8, 1, 3, 6\}$. A operação $\overline{X + Y} + X.Y$, por exemplo, produz o conjunto de classes de alturas $M_1 = \{4, 5, 8, 3, 6\}$. Para entender a resolução dessa expressão, passo a passo, deve-se compreender o resultado parcial de cada um de seus membros.

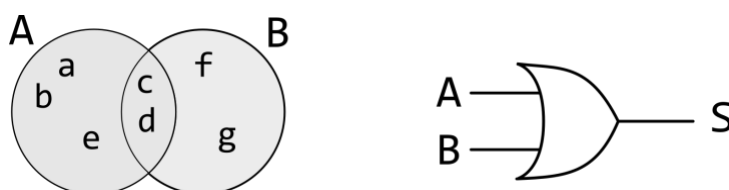
A operação $X + Y$ refere-se à união (U) dos conjuntos X e Y. A união contém todos os elementos que estão em X e Y, ou seja, $X + Y = \{0, 1, 2, 4, 5, 7, 8, 9, 10, 11\}$. Por outro lado, $\overline{X + Y}$, ou seja, a negação de $X + Y$, se refere a tudo que não está em $X + Y$, com relação a um universo, ou conjunto de referência. Essa operação produz, portanto, o complemento de um conjunto tomando um universo como referência. Para o propósito desta solução, assumi que o conjunto universo consiste em todos os elementos disponíveis nos três conjuntos. Portanto, nosso universo é $U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$, que contém todos os elementos possíveis em X, Y, e Z. O complemento de $X + Y$, ou seja, $\overline{X + Y}$, produz o conjunto $\{3, 6\}$.

FIGURA 4 – Operação de disjunção (OU) representada por um sistema hidráulico



Fonte: Autor

FIGURA 5 – Operação de união (U) e porta lógica OU.



Fonte: Autor

Por sua vez, a operação $X.Y$ refere-se à interseção dos conjuntos X e Y , ou seja, os elementos que pertencem simultaneamente a ambos os conjuntos. Esses elementos, $\{4, 5, 8\}$, podem ser facilmente visualizados em um diagrama de Venn (FIGURA 7).²⁴

FIGURA 6 – Disjunção exclusiva, condicional e bicondicional

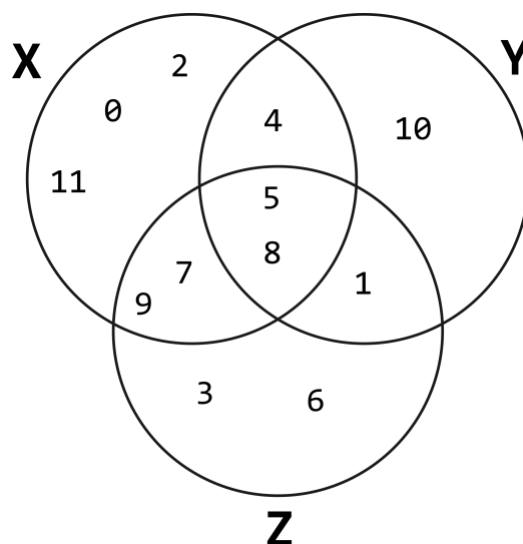
Disjunção Exclusiva			Condicional			Bicondicional		
A	B	S	A	B	S	A	B	S
0	0	0	0	0	1	0	0	1
0	1	1	0	1	1	0	1	0
1	0	1	1	0	0	1	0	0
1	1	0	1	1	1	1	1	1

Fonte: Feitosa e Paulovich (2005)

²⁴ Um diagrama de Venn é uma representação gráfica que ilustra a relação entre diferentes conjuntos de dados ou categorias. Ele foi criado pelo matemático britânico John Venn no século XIX e é amplamente utilizado em diversas áreas como matemática, lógica, estatística, ciências e filosofia para mostrar como os conjuntos se intersectam, se unem ou se diferenciam. Para um maior detalhamento sobre teoria dos conjuntos, incluindo a utilização de diagramas de Venn, ver o capítulo 3 de Feitosa e Paulovich (2005).

Finalmente, o resultado da expressão inicial, $\overline{X + Y} + X.Y$, consiste na soma de cada um dos resultados parciais: {3,6,4,5,8}. A proposição de diversas expressões permite a seleção de classes de alturas, as quais podem ser aplicadas em determinado trecho de um fragmento musical. Uma estratégia similar de planejamento composicional foi utilizada por Iannis Xenakis em sua obra *Herma*, cujo planejamento em larga escala foi baseado em uma série de operações booleanas (FIGURA 8) acopladas a um diagrama temporal (FIGURA 9), enquanto o planejamento local se estruturou a partir de processos estocásticos. No caso dessa obra de Xenakis, o conjunto universo consistiu nas 88 teclas do piano. Esse conjunto (R) foi subdividido em três conjuntos A, B e C, com interseções não vazias (ou seja, contendo elementos comuns entre si).²⁵

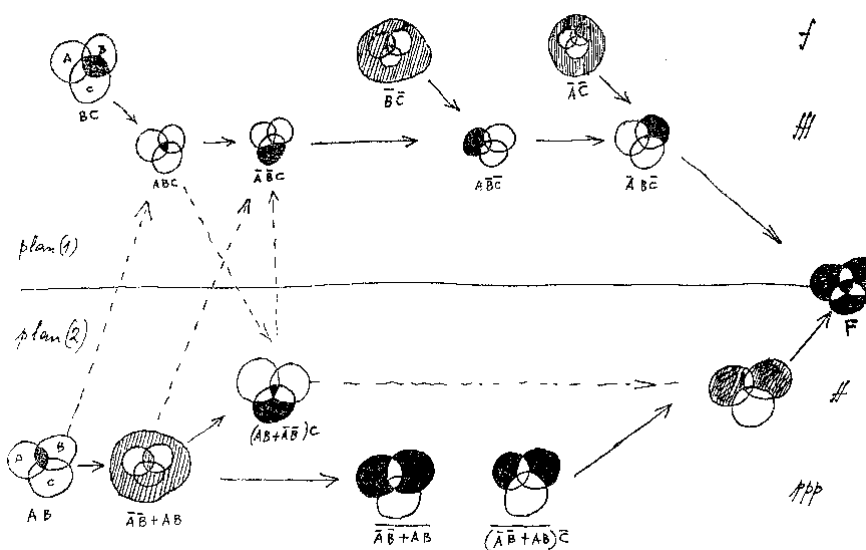
FIGURA 7 – Diagrama de Venn dos conjuntos X, Y e Z



Fonte: Autor

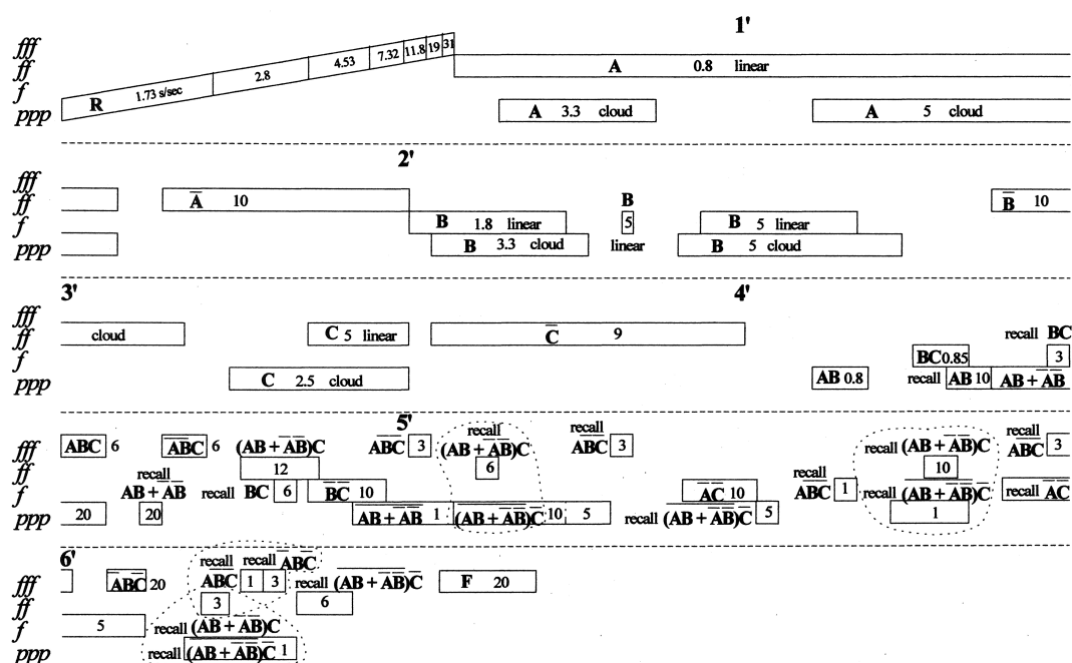
²⁵ Explicações detalhadas sobre o processo composicional de *Herma*, incluindo o conteúdo dos conjuntos A, B e C, podem ser encontradas nos textos de Wannamaker (2001), Squibbs (2000), Agon *et al* (2004) e Montague (2024).

FIGURA 8 – Diagramas de Venn e expressões booleanas utilizadas por Xenakis no planejamento composicional de *Herma*.



Fonte: Xenakis (1981, p. 206)

FIGURA 9 – Diagrama de planejamento temporal de *Herma*.



Fonte: Squibbs (2000, p. 144)

2. Planejamento composicional de *Trivium*

Nesta seção, será detalhado o planejamento composicional de *Trivium*, a partir dos três referenciais teóricos supramencionados. O primeiro movimento (seção 2.1), intitulado *Gramática*, foi construído a partir de alturas produzidas por dois aplicativos escritos em *Python* que manipularam a gramática do Tipo 3. O segundo movimento (seção 2.2), *Retórica*, utilizou a estruturação e as figuras retóricas para produzir variações de um fragmento original. Finalmente, o terceiro movimento (seção 2.3), *Lógica*, se baseou nas operações booleanas propostas por Xenakis para a composição de *Herma*.

Antes de detalhar os três movimentos, convém esclarecer que os procedimentos descritos não visam aumentar a complexidade superficial da partitura, mas organizar geração, seleção, aplicação, complementação e polimento de materiais. O planejamento composicional articular objetos abstratos e superfície instrumental por meio de curadoria composicional.

2.1 Primeiro movimento: Gramática

A fase pré-composicional de *Gramática* consistiu inicialmente na elaboração de dois programas em *Python*, baseados na gramática chomskyana do Tipo 3. Esses programas são representações algorítmicas de sistemas composicionais (Pitombeira, 2020) que lidam com objetos genéricos e cujas relações são definidas internamente pelas próprias regras gramaticais. O programa 1 (Apêndice 6) pode gerar uma palavra de pequena dimensão, na maioria das vezes um único símbolo (uma classe de alturas), e por isso precisa ter iterações definidas pelo compositor. Durante o planejamento composicional (Pitombeira, 2024), foram escolhidas dez iterações, a partir de um procedimento de tentativa e erro. Além disso, o programa foi rodado dez vezes, produzindo dez segmentos. As classes de alturas resultantes (*strings*) em cada rodada são separadas por um ponto (QUADRO 7).

QUADRO 7 – *Strings* resultantes da execução do programa 1 (Gramática Tipo 3).

01	7.4.9.21.88684.4.2.6.8.3.
02	5.7.3.4.9.1.633B0A.A.5.B.
03	8.6.9.4.086.A.6.0.B.A.
04	B.6.28431.2.34.4.4.5.4.3.
05	2.3.A.B.0.5.B.8.80.1.
06	8B0.5.B.8.1.B1.0.A.6.798.
07	A.3.2B.37.575.9.3.8.B.7.
08	1.6.0.5.0.210.76.0.A.6.
09	292.9.A.6.6197.3.7.7.7.3.
10	0.8.7.82.9.6.51.1.9.B.

Fonte: Autor

O programa 2 (Apêndice 7), por sua vez, tem a capacidade de gerar palavras de maior porte em um único ciclo operacional, com base na seguinte gramática, expressa no formato de um dicionário: `music_grammar = {'S': ['08B', 'C'], 'B': ['12C', '895'], 'C': ['359D', '14B'], 'D': ['46E'], 'E': ['7abS']}`. Os programas geraram os segmentos (*strings*) mostrados no QUADRO 7 (programa 1) e QUADRO 8 (programa 2). A produção desses segmentos se constitui na fase de planejamento composicional denominada *particularização*, na qual regras produzem valores específicos a partir de valores ou situações genéricas. Os resultados estão no formato de classes de alturas.

As saídas do QUADRO 7 e do QUADRO 8 correspondem às execuções preservadas pelo compositor; os *scripts*, sem *seed* fixa, destinam-se à geração de variantes e não à reprodução literal desses quadros.

QUADRO 8 – *Strings* resultantes da execução do programa 2 (Gramática Tipo 3).

01	1412359467ab081214895
02	0812359467ab359467ab08895
03	359467ab359467ab359467ab08895
04	359467ab1412359467ab359467ab1412359467ab1412359467ab14121412359467ab14121412359467ab08895
05	359467ab0812359467ab1412359467ab081214895
06	359467ab08121412141214895
07	0812359467ab359467ab08895
08	08121412359467ab08895
09	081214121412359467ab14895
10	359467ab081214895

Fonte: Autor

É importante mencionar que o segundo programa escolheu dois parâmetros de maneira randômica: os registros e as durações. A escolha dos registros corresponde à fase de planejamento composicional denominada *aplicação*, que consiste em definir um cenário de superfície para objetos abstratos. Vale mencionar que essa fase só é necessária quando os objetos são abstratos, ou seja, não são observáveis na superfície.²⁶ Nesse caso, como as classes de alturas são objetos abstratos, o planejamento do registro se constituiu na aplicação propriamente. Esse planejamento foi realizado em nível algorítmico interno durante a execução do programa e previu a aleatoriedade como fator de determinação dos registros. A escolha das durações foi realizada a partir de procedimentos randômicos definidos no programa 2. Essa escolha corresponde à fase de *complementação*. Nessa fase do planejamento, os parâmetros não declarados no sistema são adicionados. Como o sistema lida prioritariamente com o parâmetro altura, qualquer inclusão paramétrica adicional está no campo da complementação. O programa 2 forneceu o programa 2 forneceu uma sequência em *music21*, posteriormente exportada/convertida para notação musical. A FIGURA 10 mostra o quinto segmento produzido pelo programa 2 (rotulado como 5/2), já com registros e durações, mas ainda sem articulações e dinâmicas.

FIGURA 10 – Fragmento 5/2 (quinto fragmento gerado pelo Programa 2).

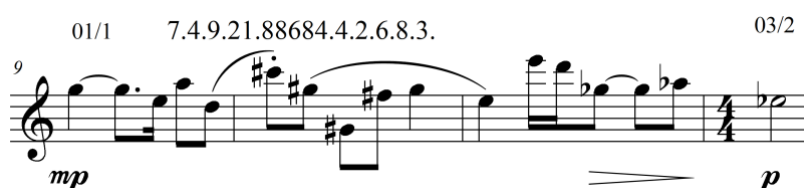


Fonte: Autor.

²⁶ É importante ter em mente, para implementações futuras, o caso dos contornos de alturas. Quando se observa uma sequência de alturas na superfície musical, pode-se inferir daí facilmente seu contorno: nota mais grave, nota mais aguda, nota intermediária, por exemplo. Essa inferência trata-se precisamente de uma abstração. No sentido etimológico, “abstração” refere-se ao ato de retirar ou separar algo de seu contexto original. Essa ideia está relacionada à capacidade de isolar mentalmente um elemento ou conceito de seu ambiente concreto ou imediato, transformando-o em algo mais geral ou conceitual. Assim, em relação às alturas, que são elementos genuinamente superficiais, os contornos são abstrações, mesmo que produzidas sem muito esforço analítico. Um contorno $\langle 0 \ 2 \ 1 \rangle$, por exemplo, só se manifesta na superfície quando a ele são aplicadas alturas correspondentes.

Já o programa 1, forneceu apenas classes de alturas. A aplicação (escolha dos registros) e a complementação (durações, articulações e dinâmicas) foram transparentes, de tal maneira que esses parâmetros foram livremente introduzidos durante o ato composicional. A FIGURA 11 mostra o primeiro segmento gerado pelo programa 1, já com durações, dinâmicas e articulações aplicadas. As ligaduras tiveram um papel importante de agrupar as *strings* formadas por mais de um elemento (como o “21” e o “88684”).

FIGURA 11 – Fragmento 1/1 (primeiro fragmento gerado pelo Programa 1) já com durações, dinâmicas e articulações aplicadas.



Fonte: Autor.

Uma etapa de experimentação, com o auxílio de um instrumento musical, me ajudou a selecionar nove dos vinte segmentos gerados pelos dois programas: 05/2, 01/1, 03/2, 02/1, 02/2, 03/1, 10/2, 04/1 e 09/2. Nessa lista, o segmento 05/2 é o quinto segmento produzido pelo programa 2; o 01/1 é o primeiro segmento produzido pelo programa 1, e assim por diante.

A seleção desses nove segmentos resultou de uma curadoria composicional orientada por critérios musicais, instrumentais e formais, sem aceitação automática dos resultados algorítmicos. Avaliaram-se a clareza gestual de cada segmento (contorno, direção melódica, perfil, densidade intervalar, potencial expressivo e rítmico) e sua viabilidade instrumental na flauta doce Helder alto, considerando extensão, registro, respiração, articulação e idiomatismo. Também se observou a função de cada material no fluxo musical, distinguindo segmentos aptos a iniciar ideias, prolongá-las, criar contraste, intensificar o movimento ou produzir fechamento. Por fim, examinou-se seu potencial de complementação por durações, dinâmicas, articulações, ligaduras, técnicas estendidas e inflexões agógicas. Assim, a curadoria operou como etapa intermediária entre geração algorítmica e escrita instrumental, preservando a autonomia autoral do compositor.

Uma fase final de polimento consistiu no ajuste transposicional dos segmentos selecionados que foram gerados pelo programa 2, uma vez que esses segmentos já foram fornecidos com registros (e durações). Esses ajustes foram realizados para adequar o segmento à extensão possível do instrumento (flauta doce Helder alto) ou por livres preferências composicionais. As transposições foram cromáticas integrais (todas as alturas) ou mutacionais (algumas alturas). O QUADRO 9 mostra a lista dos segmentos e descreve os ajustes transpositivos. As figuras do Apêndice 8 apresentam, na linha inferior, os segmentos originais gerados pelo programa 2 para o primeiro movimento, enquanto na linha superior são exibidos os ajustes desses segmentos e as escolhas composicionais aplicadas aos dados gerados pelo programa 1, já incorporando dinâmicas, articulações, técnicas estendidas e variações agógicas.

QUADRO 9 – Seleção e ajuste dos segmentos gerados pelos dois programas (os segmentos gerados pelo programa 2 estão em negrito)

-
- **05/2** – transposto uma terça maior acima para que a nota mais grave fosse o Mi4
 - 01/1 – ligaduras indicadas pelas unidades separadas na geração – Elisão da última nota dessa *string* com a primeira nota da *string* seguinte.
 - **03/2** – a maioria das notas foi transposta uma oitava acima
 - 02/1 – sem ajuste
 - **02/2** – ajuste de oitava
 - 03/1 – sem ajuste
 - **10/2** – livres ajustes de oitava
 - 04/1 – sem ajuste
 - **09/2** – ajustes livres de oitava
-

Fonte: Autor

Do ponto de vista musical, o primeiro movimento resulta de uma tensão entre fragmentação e continuidade. Os materiais gerados pelo Programa 1 tendem a produzir unidades mais curtas, frequentemente percebidas como células ou impulsos gestuais, enquanto os materiais do Programa 2, por serem mais extensos, favorecem linhas mais contínuas e direcionalmente reconhecíveis. A alternância entre esses dois tipos de segmento contribui para uma superfície móvel, na qual a recorrência não se dá pela repetição literal de temas, mas pelo retorno de comportamentos: agrupamentos curtos, mudanças de direção, perfis intervalares contrastantes, elisões, variações de

energia²⁷ e articulações que aproximam ou separam as *strings*. A escrita final preserva a identidade das *strings* como matriz de alturas e de segmentação, mas as transforma por meio de registros, durações, ligaduras, dinâmicas, articulações e técnicas estendidas.

2.2 Segundo movimento: *Retórica*

O passo inicial para a criação de *Retórica*, segundo movimento de *Trivium*, consistiu na elaboração de um sistema composicional original baseado em figuras retóricas. Sua característica principal é a retroalimentação interna: os resultados do planejamento são avaliados pelo triângulo *Ethos-Pathos-Logos*, não como estratégia persuasiva discursiva, mas como critério heurístico de curadoria. O *Ethos* verifica a confiabilidade do procedimento, isto é, se a figura foi aplicada de modo reconhecível e coerente. O *Pathos* avalia a eficácia sensível do resultado, considerando seu efeito sonoro, expressivo ou gestual. O *Logos* examina a coerência interna do material em relação ao motivo inicial, ao repositório produzido e ao fluxo do movimento. Essa avaliação depende da escuta, experiência e escolhas técnico-estéticas do compositor, preservando a dimensão volitiva e a abertura heurística do sistema, sintetizado no QUADRO 10 e na FIGURA 12.

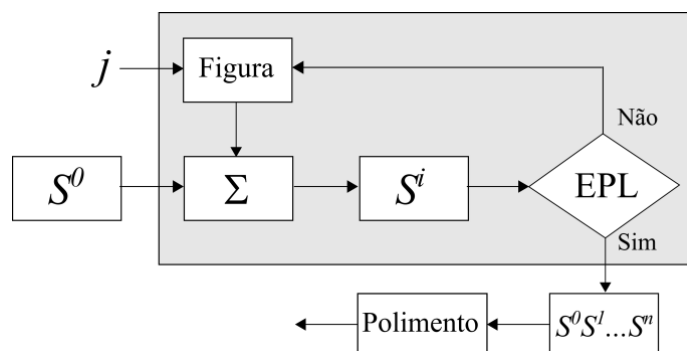
QUADRO 10 – Definições do sistema composicional retórico

Definição	Descrição
1.	O sistema recebe um motivo inicial (original ou intertextual), o qual é particionado em <i>clangs</i> .
2.	O(a) compositor(a) escolhe uma figura retórica (<i>j</i>) e aplica a um motivo (<i>Sⁱ</i>) formando o motivo <i>Sⁱ⁺¹</i> .
3.	O resultado é verificado do ponto de vista do <i>Ethos</i> (credibilidade), <i>Pathos</i> (emoção) e <i>Logos</i> (lógica). Se for aprovado o motivo é enviado para um repositório sequencial. Se for reprovado, nova figura é aplicada ao motivo <i>Sⁱ</i> . O ciclo de verificação se repete até que o resultado seja aprovado e siga para o repositório.
4.	Concluído o ciclo da definição 3, nova figura (<i>j</i>) é aplicada a um dos motivos do repositório, livremente selecionado pelo(a) compositor(a).
5.	O tamanho do ciclo operacional é definido durante o planejamento
6.	O repositório passa por complementação (acréscimo de dinâmicas, articulações, variações agógicas, técnicas estendidas etc.) e polimento, no qual pequenos ajustes são introduzidos

Fonte: Autor

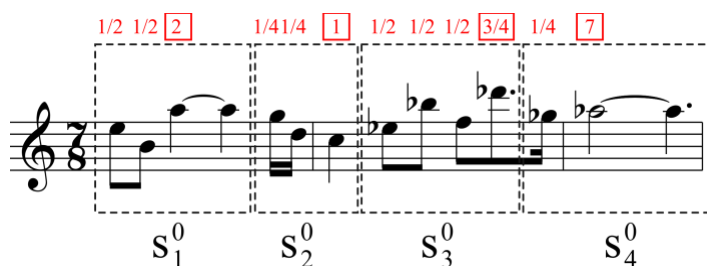
²⁷ Sobre a metáfora energética aplicada à melodia, ver o capítulo “*La mélodie comme flux énergétique*”, em Kaltenecker (2024), no qual o autor discute a experiência melódica no século XX, incluindo a abordagem dinâmica de Ernst Kurth.

FIGURA 12 – Sistema composicional retórico.



Fonte: Autor

A primeira fase do planejamento composicional de Retórica, particularização, consistiu inicialmente na elaboração de um segmento, mostrado na FIGURA 13, com seus quatro *clangs*.²⁸ Nessa figura, as durações das notas são especificadas na parte superior, em cor vermelha. A divisão de um *clang* ocorre quando a duração de uma nota é maior que a duração de uma nota anterior e posterior. Por exemplo, a terceira nota do primeiro compasso (FIGURA 13) tem uma duração de duas semínimas, que é maior que a duração da segunda nota (1/2 semínima) e da quarta (1/4 de semínima).

FIGURA 13 – Segmento inicial (S^0) de *Retórica*

Fonte: Autor

Em termos de classes de alturas, esse segmento teve sua origem em uma série dodecafônica derivada tricordal (Straus, 2013), isto é, uma série em que os tricordes são todos pertencentes à mesma

²⁸ *Clang* é um som ou configuração sonora percebida como unidade musical primária ou gestalt auditiva. No original: “A sound or sound-configuration which is perceived as a primary musical unit or aural gestalt” (Tenney, 1988, p.87).

classe de conjuntos, no caso, a 027. A matriz dodecafônica referente a essa série é mostrada no QUADRO 11. Em seguida, iniciei o processo de aplicação das figuras, seguindo as definições de Bartel (1997). As figuras utilizadas são mostradas no QUADRO 12.²⁹ Um grafo explicando o uso das figuras no contexto do movimento é mostrado na FIGURA 14. Alguns *clangs* foram levemente expandidos ou contraídos. Essa alteração é indicada por um sinal de mais (+) ou de menos (-). A linha pontilhada conectando os segmentos 5 e 6 não indica uma transformação, mas uma conexão, uma vez que o sexto segmento (S^6) se origina a partir do primeiro segmento (S^0). O Apêndice 9 contém o segundo movimento com as indicações dos segmentos, mas ainda sem dinâmicas, articulações ou andamento.

QUADRO 11 – Matriz dodecafônica da série de *Retórica*

	I ₄	I ₁₁	I ₉	I ₇	I ₂	I ₀	I ₃	I ₁₀	I ₅	I ₁	I ₆	I ₈	
P ₄	E	B	A	G	D	C	E $\flat$	B $\flat$	F	D $\flat$	G $\flat$	A $\flat$	R ₄
P ₉	A	E	D	C	G	F	A $\flat$	E $\flat$	B $\flat$	G $\flat$	B	D $\flat$	R ₉
P ₁₁	B	G $\flat$	E	D	A	G	B $\flat$	F	C	A $\flat$	D $\flat$	E $\flat$	R ₁₁
P ₁	D $\flat$	A $\flat$	G $\flat$	E	B	A	C	G	D	B $\flat$	E $\flat$	F	R ₁
P ₆	G $\flat$	D $\flat$	B	A	E	D	F	C	G	E $\flat$	A $\flat$	B $\flat$	R ₆
P ₈	A $\flat$	E $\flat$	D $\flat$	B	G $\flat$	E	G	D	A	F	B $\flat$	C	R ₈
P ₅	F	C	B $\flat$	A $\flat$	E $\flat$	D $\flat$	E	B	G $\flat$	D	G	A	R ₅
P ₁₀	B $\flat$	F	E $\flat$	D $\flat$	A $\flat$	G $\flat$	A	E	B	G	C	D	R ₁₀
P ₃	E $\flat$	B $\flat$	A $\flat$	G $\flat$	D $\flat$	B	D	A	E	C	F	G	R ₃
P ₇	G	D	C	B $\flat$	F	E $\flat$	G $\flat$	D $\flat$	A $\flat$	E	A	B	R ₇
P ₂	D	A	G	F	C	B $\flat$	D $\flat$	A $\flat$	E $\flat$	B	E	G $\flat$	R ₂
P ₀	C	G	F	E $\flat$	B $\flat$	A $\flat$	B	G $\flat$	D $\flat$	A	D	E	R ₀
	RI ₄	RI ₁₁	RI ₉	RI ₇	RI ₂	RI ₀	RI ₃	RI ₁₀	RI ₅	RI ₁	RI ₆	RI ₈	

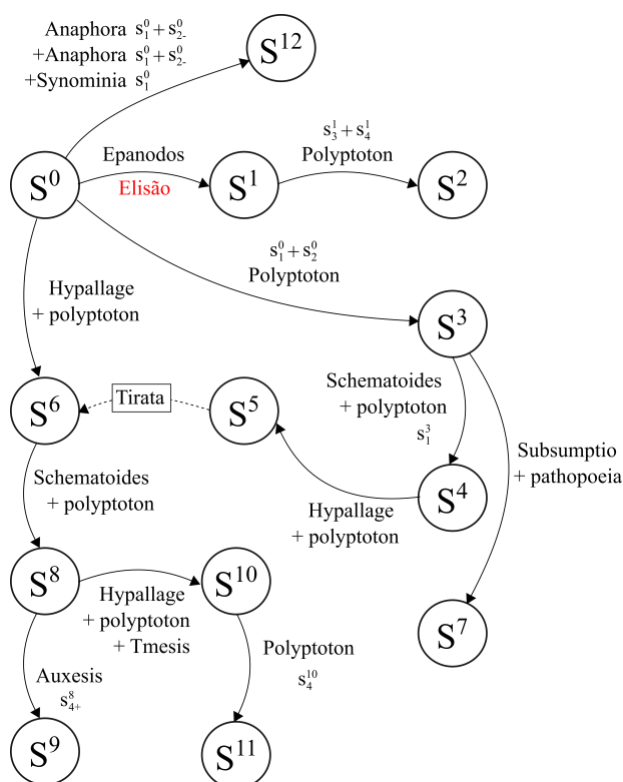
Fonte: Autor

²⁹ No original: *Anaphora* – a repetition of the opening phrase or motive in a number of successive passages; *Auxesis* – successive repetitions of a musical passage which rise by step; *Epanodos* – the retrograde repetition of a phrase; *Hypallage* – an inversion of the fugal theme; *Pathopoeia* – a musical passage which seeks to arouse a passionate affection through chromaticism or some other means; *Polyptoton* – a repetition of a melodic passage at different pitches; *Schematoides* – a figure which restructures a previous passage or configuration, either through changing text underlay or through durational augmentation or diminution; *Subsumptio* – various additions of lower neighboring notes; *Synonymia* – an altered or modified repetition of a musical idea; *Tirata* – a rapid scalar passage, spanning a fourth to an octave or more; *Tmesis* – a sudden interruption or fragmentation of the melody through rests.

QUADRO 12 – Figuras retóricas utilizadas no segundo movimento de *Trivium*, *Retórica*

Figura	Definição
Anaphora	repetição de uma frase inicial ou motivo em um número sucessivo de passagens (Bartel, 1997, p.184).
Auxesis	repetições sucessivas de uma passagem musical que sobem por graus conjuntos (Bartel, 1997, p. 209).
Epanodos	a repetição retrógrada de uma frase (Bartel, 1997, p. 259).
Hypallage	uma inversão do tema [da fuga] (Bartel, 1997, p. 298).
Pathopoeia	uma passagem musical que busca despertar um afeto passional por meio do cromatismo ou de algum outro recurso (Bartel, 1997, p. 359).
Polyptoton	uma repetição de uma passagem melódica em diferentes alturas (Bartel, 1997, p. 367).
Schematoides	uma figura que reestrutura uma passagem ou configuração anterior, seja por mudança na aplicação do texto, seja por aumento ou diminuição de duração (Bartel, 1997, p. 382).
Subsumptio	várias adições de bordaduras inferiores (Bartel, 1997, p. 385).
Synonymia	a repetição alterada ou modificada de uma ideia musical (Bartel, 1997, p. 405).
Tirata	uma passagem rápida de escala, abrangendo uma quarta, uma oitava ou mais (Bartel, 1997, p. 409).
Tmesis	uma interrupção súbita ou fragmentação da melodia por meio de pausas (Bartel, 1997, p. 412).

Fonte: Bartel (1997)

FIGURA 14 – Grafo transformacional dos motivos em *Retórica*

Fonte: Autor

Musicalmente, o sistema retórico produz uma continuidade baseada na memória transformada. O motivo inicial reaparece sob diferentes condições de identidade: reiterado, invertido, retrogradado, fragmentado, expandido, contraído, cromatizado, interrompido ou deslocado. Figuras como *anaphora*, *epanodos*, *hypallage*, *pathopoeia* e *tmesis* operam como procedimentos de reinterpretação gestual: reforçam recorrências, alteram a orientação temporal, modificam a direção intervalar, intensificam a superfície cromática e tornam audíveis interrupções. O movimento organiza-se, assim, como sequência de segmentos vinculados ao material de origem e expressivamente reconfigurados.

2.3 Terceiro movimento: *Lógica*

A composição do terceiro movimento de *Trivium*, intitulado *Lógica*, iniciou-se com a elaboração de um programa em *Python* (Apêndice 10) para gerar 21 segmentos de classes de alturas a partir das mesmas operações booleanas da obra *Herma*, de Xenakis (veja seção 1.3 deste artigo). Esses segmentos (QUADRO 13) foram agrupados em três seções (A, B, A'), cada uma com sete segmentos. O contraste da seção central (B) se dá pelo uso de trinados, *frulatos* e uso da voz. As classes de alturas dos segmentos foram dispostas livremente em uma faixa de um a cinco compassos durante o ato composicional. Da mesma forma, ritmos (durações e pontos de ataque), dinâmicas, articulações, variações agógicas e técnicas estendidas foram acrescentados durante a composição do trecho. O Apêndice 11 contém o terceiro movimento, com as indicações dos segmentos e das seções, já mostrando articulações, dinâmicas, variações agógicas e técnicas estendidas. A partitura completa da obra é mostrada no Apêndice 12.

QUADRO 13 – Operações lógicas de *Herma*, utilizadas no terceiro movimento de *Trivium*, *Lógica*

Seção	Seg	Comp.	Operações → Segmentos
A	1.	1-3	$R = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$
	2.	4-5	$A = \{0, 2, 4, 7, 9\}$
	3.	6-8	$\sim A = \{1, 3, 5, 6, 8, 10, 11\}$
	4.	9-10	$B = \{1, 3, 5, 7, 9, 11\}$
	5.	11-12	$\sim B = \{0, 2, 4, 6, 8, 10\}$
	6.	13-15	$C = \{4, 6, 7, 8, 10, 11\}$
	7.	16-17	$\sim C = \{0, 1, 2, 3, 5, 9\}$
B	8.	18-19	$A.B = \{9, 7\}$
	9.	20	$B.C = \{11, 7\}$
	10.	21-25	$(\sim A.\sim B) + A.B = \{6, 7, 8, 9, 10\}$
	11.	26	$A.B.C = \{7\}$
	12.	27-30	$\sim A.\sim B.C = \{8, 10, 6\}$
	13.	31-32	$(\sim A.\sim B + A.B).C = \{8, 10, 6, 7\}$
	14.	33-34	$\sim B.\sim C = \{0, 2\}$
A'	15.	35-38	$\sim(\sim A.\sim B + A.B) = \{0, 1, 2, 3, 4, 5, 11\}$
	16.	39-41	$(\sim(\sim A.\sim B + A.B)).\sim C = \{0, 1, 2, 3, 5\}$
	17.	42	$(\sim(\sim A.\sim B + A.B)).C = \{11, 4\}$
	18.	43	$A.\sim B.\sim C = \{0, 2\}$
	19.	44-45	$\sim A.\sim C = \{1, 3, 5\}$
	20.	46-47	$\sim A.B.\sim C = \{1, 3, 5\}$
	21.	48-52	$F = (A.B + \sim A.\sim B).C + \sim(A.B + \sim A.\sim B).\sim C = \{0, 1, 2, 3, 5, 6, 7, 8, 10\}$

Fonte: Adaptado de Xenakis (1991)

No terceiro movimento, as operações booleanas convertem relações abstratas de inclusão, exclusão, complemento e interseção em campos de classes de alturas com diferentes densidades. As seções A e A' seguem mais diretamente a sucessão dessas operações, enquanto B intensifica a percepção por meio de trinados, *frulatos* e voz. A forma A–B–A' articula cálculo lógico, derivado do modelo de *Herma*, e distribuição formal baseada na diferenciação instrumental e tímbrica.

3. Considerações finais

Este trabalho descreve o planejamento composicional de *Trivium*, obra em três movimentos para flauta doce Helder alto, estruturada a partir de Gramática, Retórica e Lógica. Cada referencial constitui uma base teórica para explicitar o encadeamento entre níveis da criação musical: o conceito delimita o objeto musical, o procedimento formal ou algorítmico o particulariza em materiais concretos, e a escrita instrumental o converte em representação musical de superfície, dotada de

potencial sonoro, por meio de escolhas de registro, duração, articulação, dinâmica, técnicas estendidas e ajustes finos. Assim, *Trivium* apresenta um percurso documentável do abstrato ao concreto, apoiado em resultados computacionais, quadros de seleção e procedimentos de revisão.

Em *Gramática*, dois programas desenvolvidos para o estudo geraram segmentos derivados de uma gramática do Tipo 3. Em *Retórica*, um sistema composicional produziu novos segmentos a partir de um motivo inicial transformado por figuras retóricas e *feedback* composicional. Em *Lógica*, operações booleanas, em diálogo com procedimentos de Xenakis em *Herma*, organizaram relações estruturais em seções formais contrastantes.

A relevância do trabalho é pedagógica e profissional. No ensino, oferece uma via para tratar a composição como prática reflexiva, distinguindo estruturas geradoras, critérios de seleção e realização performática. Na prática profissional, articula macroforma, coerência local, geração, curadoria e escrita instrumental. O diálogo com a flautista Francielle Paixão integra esse processo, envolvendo encomenda, testes, registro, técnicas estendidas e agógica como etapas de negociação performática e iteração do material. O artigo integra o projeto “Planejamento Composicional: reflexões teórico-estéticas no âmbito de uma produção artística formalizada”, desenvolvido no PPGM-UFRJ.³⁰

REFERÊNCIAS

AGON, Carlos; ANDREATTA, Moreno; ASSAYAG, Gérard; SCHAUB, Stéphan. Formal Aspects of Iannis Xenakis' "Symbolic Music": A Computer-Aided Exploration of Compositional Processes. **Journal of New Music Research**, v. 33, n. 2, p. 145-149, 2004.

ANTUNES, Jorge. A anadiplose aplicada à estrutura musical. **Música Hódie**, Goiânia, v. 14, n. 1, 2014. Disponível em: <https://revistas.ufg.br/musica/article/view/32865/17433>. Acesso em: 9 maio 2026.

ARISTÓTELES. **Treatise on Rhetoric**. Londres: Henry G. Bohn, 1857.

BARTEL, Dietrich. **Musica Poetica: Musical Rhetorical Figures in German Baroque Music**. Lincoln: University of Nebraska Press, 1997.

³⁰ Disponível em <https://ppgm.musica.ufrj.br/planejamento-composicional-reflexoes-teorico-esteticas-no-ambito-de-uma-producao-artistica-formalizada-2023/>

BENT, Ian; POPLÉ, Anthony. Analysis. In: **Grove Music Online**. Oxford: Oxford Music Online, [s. d.]. Disponível em: <http://www.oxfordmusiconline.com/subscriber/article/grove/music/41862pg2>. Acesso em: 28 mar. 2011.

BONDS, M. E. **Wordless Rhetoric: Musical Form and the Metaphor of Oration**. Cambridge: Harvard University Press, 1991.

BRIOT, Jean-Pierre; HADJERES, Gaëtan; PACHET, François-David. **Deep learning techniques for music generation**. Cham: Springer, 2020.

BURMEISTER, Joachim. **Musical Poetics**. New Haven: Yale University Press, 1993.

BUXTON, William; REEVES, William; BAECKER, Ronald; MEZEL, Leslie. The Use of Hierarchy and Instance in a Data Structure for Computer Music. **Computer Music Journal**, v. 2, n. 4, p. 10-20, 1978.

CHANDLER, Daniel. **Semiotics: the basics**. 2. ed. London; New York: Routledge, 2007.

CHOMSKY, Noam. On certain formal properties of grammars. **Information and Control**, v. 2, n. 2, p. 137-167, 1959.

FEITOSA, H. de A.; PAULOVICH, L. **Um prelúdio à lógica**. São Paulo: UNESP, 2005.

FRÖHLICH, Susanne. **The New Potential of a 21st Century Recorder**. 2020. Tese (Doutorado em Música). University of Music and Performing Arts Graz, Graz, 2020.

HOLTZMAN, Steven R. **Generative Grammars and the Computer-Aided Composition of Music**. 1980. Tese (Doutorado em Ciências da Computação). University of Edinburgh, Edinburgh, 1980.

JONES, Kevin. Compositional Applications of Stochastic Processes. **Computer Music Journal**, v. 5, n. 2, p. 45-61, 1981.

JOSEPH, Sister Miriam. **The Trivium: The Liberal Arts of Logic, Grammar, and Rhetoric**. Philadelphia: Paul Dry Books, 2002.

KALTENECKER, Martin. **L'expérience mélodique au XXe siècle**. Genève: Contrechamps, 2024.

LERDAHL, Fred; JACKENDOFF, Ray. **A Generative Theory of Tonal Music**. Cambridge, MA: MIT Press, 1983.

LUCAS, Mônica. A invenção musical em Der vollkommene Capellmeister [“o mestre-de-capela

- perfeito”, 1739], de Johann Mattheson. **ArteFilosofia**, v. 11, n. 21, p. 75-92, 2016.
- MARTINEAU, John (ed.). **The Quadrivium: The Four Classical Liberal Arts of Number, Geometry, Music, & Cosmology**. New York: Walker & Company, 2010.
- MCCORMACK, J. Grammar Based Music Composition. In: STOCKER, R. et al. (ed.). **Complex Systems 96: from Local Interactions to Global Phenomena**. Amsterdam: IOS Press, 1996. p. 320-336.
- MCCRELESS, Patrick. Music and Rhetoric. In: CHRISTENSEN, Thomas (ed.). **The Cambridge History of Western Music Theory**. Cambridge: Cambridge University Press, 2007. p. 847-879.
- MENEZES, Paulo Blauth. **Linguagens formais e autômatos**. Porto Alegre: Sagra Luzzatto, 2000.
- MEYER, Michel. **Histoire de la Rhétorique des Grecs à nos jours**. Paris: Librairie Générale Française, 1999.
- MIRANDA, Eduardo Reck. **Composing Music with Computers**. Oxford: Focal Press, 2001.
- MONTAGUE, Eugene. **The Limits of Logic: Structure and Aesthetics in Xenakis’s Herma**. [S. l.]: [s. n.], [s. d.]. Disponível em: <http://www.ex-tempore.org/montague/>. Acesso em: 30 set. 2024.
- MORTARI, C. A. **Introdução à lógica**. São Paulo: UNESP, 2001.
- NATTIEZ, Jean-Jacques. **Music and discourse: toward a semiology of music**. Princeton, NJ: Princeton University Press, 1990.
- NIERHAUS, Gerhard. **Algorithmic composition: Paradigms of automated music generation**. New York: Springer, 2009.
- OLIVEIRA, Helder A. **Modelagem Gestáltica: modelos sistêmicos a partir de princípios da Teoria da Gestalt**. 2020. 300 f. Tese (Doutorado em Música) – Escola de Música, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2020.
- PALISCA, Claude V. **Studies in the History of Italian Music and Music Theory**. Oxford: Clarendon Press, 1994.
- PERELMAN, Chaïm; OLBRECHTS-TYTECA, L. **The New Rhetoric: A Treatise on Argumentation**. Notre Dame: University of Notre Dame Press, 2008.
- PERSONE, Pedro. Sobre ‘O Tratado das Paixões da Alma’ e o discurso musical nos séculos XVII e XVIII. **Cadernos de Estudos: Análise Musical**, n. 4, p. 23-45, 1991.

PIEDADE, Acácio. Música e retoricidade. In: ENCONTRO DE MUSICOLOGIA DE RIBEIRÃO PRETO, 4., 2012, Ribeirão Preto. **Anais [...]**. Ribeirão Preto: [s. n.], 2012.

PITOMBEIRA, L. Compositional Systems: Overview and Applications. **MusMat – Brazilian Journal of Music and Mathematics**, v. 4, n. 1, p. 39-62, jun. 2020.

PITOMBEIRA, Liduino. Planejamento Composicional a partir de paradigmas arquetípicos. **Musica Theorica**, v. 9, n. 2, p. 1-56, 2024.

RIBEIRO, Roberto Macedo. **A Análise Estilística como ferramenta para o desenvolvimento de sistemas composicionais dinâmicos**. 2022. Tese (Doutorado em Música). Programa de Pós-Graduação em Música, Centro de Letras e Artes, Escola de Música, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2022.

ROADS, Curtis; WIENEKE, Paul. Grammar as Representations for Music. **Computer Music Journal**, v. 3, n. 1, p. 48-55, 1979.

SQUIBBS, Ronald. Musical Composition as Applied Mathematics: Set Theory and Probability in Iannis Xenakis's Herma. In: BRIDGES: MATHEMATICAL CONNECTIONS IN ART, MUSIC, AND SCIENCE, 2000, Winfield, Kansas. **Conference proceedings 2000**. Winfield, Kansas: [s. n.], 2000. p. 141-152.

STRAUS, Joseph Nathan. **Introdução à Teoria Pós-Tonal**. Tradução de Ricardo Bordini. São Paulo: Editora UNESP, 2013.

TENNEY, James; POLANSKY, Larry. Temporal Gestalt Perception in Music. **Journal of Music Theory**, v. 24, n. 2, p. 205-242, 1980.

TENNEY, James. **Meta+Hodos: A Phenomenology of 20th-Century Musical Materials and an Approach to the Study of Form and META Meta+Hodos**. Editado por Larry Polansky. 2. ed. Oakland: Frog Peak Music, 1988.

VELARDO, Valerio. **Generative Grammars for Music Generation - Generative Music AI Course**. YouTube, 14 nov. 2023. Disponível em: <https://www.youtube.com/watch?v=h0uuVelImro>. Acesso em: 17 jan. 2024.

VERSOLATO, J.; KERR, D. M. A teoria e a análise musical sob o influxo da retórica no período Barroco. **Per Musi**, Belo Horizonte, n. 17, p. 64-68, 2008.

WANNAMAKER, Robert. Structure and Perception in Herma by Iannis Xenakis. **Music Theory Online**, v. 7, n. 3, 2001. Disponível em: <https://www.mtosmt.org/issues/mto.01.7.3/mto.01.7.3.wannamaker.html>. Acesso em: 30 set. 2024.

XENAKIS, Iannis. **Formalized Music: Thought and Mathematics in Composition**. Stuyvesant, NY: Pendragon Press, 1991.

XENAKIS, Iannis. **Musiques formelles: Nouveaux principes formels de composition musicale**. Paris: Stock, 1981.

ZAMPRONHA, Edson. A Rebirth of Rhetoric in Recent Contemporary Music. **KunstMusik**, Köln, v. 7, p. 49-57, 2006.

ZBIKOWSKI, Lawrence M. **Foundations of Musical Grammar**. Oxford: Oxford University Press, 2017.

SOBRE O AUTOR

Liduino Pitombeira é professor de composição na Escola de Música da Universidade Federal do Rio de Janeiro e integra o grupo de pesquisa MusMat (PPGM-UFRJ). Suas obras têm sido executadas por diversos grupos no Brasil e no exterior, entre eles o Quinteto de Sopros da Filarmônica de Berlim e a Orquestra Sinfônica do Estado de São Paulo. Pitombeira tem Doutorado em Composição e Teoria pela Louisiana State University (EUA). Tem diversos artigos publicados em revistas e anais de congressos no Brasil. Email: pitombeira@musica.ufrj.br. ORCID: <https://orcid.org/0000-0002-1141-655X>

DISPONIBILIDADE DE DADOS DE PESQUISA

- Uso de dados não informado; nenhum dado de pesquisa gerado ou utilizado.

APÊNDICE

Condições de reprodutibilidade dos programas — Os programas foram executados em *Python* 3.12.12, no *Windows* 11, em ambiente *Anaconda*, com a biblioteca *music21* versão 9.9.1. Quando a reprodução literal de uma saída é necessária, a *seed* utilizada é indicada no próprio *script*. Quando *seed=None* ou quando não há fixação explícita de *seed*, a aleatoriedade integra o procedimento gerativo, e a saída não se destina à reprodução literal, mas à produção de variantes dentro das regras estabelecidas. O Apêndice 4, dedicado à demonstração de uma Gramática do Tipo 0, utiliza *seed=11*, *max_passos=60* e *minimo_notas=7*, produzindo a saída apresentada no Apêndice

5. Os programas efetivamente empregados na geração dos materiais do primeiro movimento de *Trivium* são apresentados nos Apêndices 6 e 7. Os scripts foram elaborados, revisados e testados com auxílio do ChatGPT, utilizado como ferramenta de apoio à programação, depuração e documentação do código; a concepção metodológica, a seleção dos procedimentos e a validação musical dos resultados permanecem sob responsabilidade do autor.

Apêndice 1: Programa 1 – Gramática G

```
import random

# Conjunto de terminais (notas)
notas = ['Dó', 'Ré', 'Mi', 'Sol', 'Lá']

# Regras de produção:
# S → D | D S
# D → Dó | Ré | Mi | Sol | Lá

def gerar_D():
    """Produção: D → Dó | Ré | Mi | Sol | Lá"""
    return random.choice(notas)

def gerar_S():
    """Produção: S → D | D S """
    if random.choice([True, False]): # Decide se continua a derivação
        return [gerar_D()] + gerar_S()
    else:
        return [gerar_D()]

def gerar_composicao():
    """Símbolo inicial: S"""
    tokens = gerar_S()
    return " ".join(tokens)

if __name__ == "__main__":
    composicao = gerar_composicao()
    print("Composição gerada:", composicao)
```

Apêndice 2: Outro exemplo de Gramática do Tipo 2

Neste outro exemplo de gramática do Tipo 2 (livre de contexto), considerem-se as seguintes regras de produção: $S \rightarrow F$; $F \rightarrow \text{Dó F Dó} \mid \text{Ré F Ré} \mid \text{Mi F Mi} \mid \text{Sol F Sol} \mid \text{Lá F Lá} \mid M$; $M \rightarrow N \mid \text{Dó Dó} \mid \text{Ré Ré} \mid \text{Mi Mi} \mid \text{Sol Sol} \mid \text{Lá Lá}$; $N \rightarrow \text{Dó} \mid \text{Ré} \mid \text{Mi} \mid \text{Sol} \mid \text{Lá}$. Aqui, S representa uma composição musical completa; F representa uma frase musical; M representa um motivo central (com uma nota, ou com duas notas idênticas); e N representa uma nota musical. Diferentemente de um gerador apenas concatenativo, a recursão em F impõe uma dependência global de espelhamento (a mesma nota nas extremidades), fazendo com que a cadeia terminal resultante seja palindrômica. Por essa razão, a linguagem gerada não é regular (isto é, não pode ser gerada por uma gramática do Tipo 3). Um exemplo de operação dessa gramática é mostrado detalhadamente no QUADRO 14. Um programa em *Python* que operacionaliza essa gramática está disponível abaixo, neste mesmo apêndice.

QUADRO 14 – Exemplo de ciclo operacional de uma gramática do Tipo 2.

Forma sentencial (derivação)	Regra aplicada
S	Símbolo inicial.
$S \Rightarrow F$	$S \rightarrow F$
Ré F Ré	$F \rightarrow \text{Ré F Ré}$
Ré Lá F Lá Ré	$F \rightarrow \text{Lá F Lá}$
Ré Lá Lá F Lá Lá Ré	$F \rightarrow \text{Lá F Lá}$
Ré Lá Lá Lá F Lá Lá Lá Ré	$F \rightarrow \text{Lá F Lá}$
Ré Lá Lá Lá Dó F Dó Lá Lá Lá Ré	$F \rightarrow \text{Dó F Dó}$
Ré Lá Lá Lá Dó Ré F Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Ré F Ré}$
Ré Lá Lá Lá Dó Ré Dó F Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Dó F Dó}$
Ré Lá Lá Lá Dó Ré Dó Dó F Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Dó F Dó}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré F Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Ré F Ré}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi F Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Mi F Mi}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré F Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Ré F Ré}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré Lá F Lá Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Lá F Lá}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré Lá Ré F Ré Lá Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow \text{Ré F Ré}$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré Lá Ré M Ré Lá Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$F \rightarrow M$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré Lá Ré N Ré Lá Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$M \rightarrow N$
Ré Lá Lá Lá Dó Ré Dó Dó Ré Mi Ré Lá Ré Lá Ré Lá Ré Mi Ré Dó Dó Ré Dó Lá Lá Lá Ré	$N \rightarrow \text{Lá}$

Fonte: Autor

```

import random

NOTAS = ['Dó', 'Ré', 'Mi', 'Sol', 'Lá']
NAO_TERMINAIS = {'S', 'F', 'M', 'N'}

def gerar_composicao(seed=None, p_recurcao=0.6, max_passos=50):
    """
    Gera uma composição a partir do símbolo inicial S e devolve:
    - a lista de tokens terminais (notas)
    - o rastro de derivação (para montar o QUADRO 3)
    """
    rng = random.Random(seed)

    forma = ['S'] # forma sentencial inicial
    trace = []
    trace.append(("S", "Símbolo inicial (composição)."))

    passos = 0
    while any(tok in NAO_TERMINAIS for tok in forma):
        if passos >= max_passos:
            raise RuntimeError("Número máximo de passos atingido. Ajuste
p_recurcao/max_passos.")
            passos += 1

        # Expansão mais simples: sempre expande o primeiro não-terminal
        (leftmost)
        i = next(idx for idx, tok in enumerate(forma) if tok in
NAO_TERMINAIS)
        nt = forma[i]

        if nt == 'S':
            # S → F
            forma[i:i+1] = ['F']
            trace.append(("S ⇒ F", "Substitui-se o símbolo inicial S pela
frase F (S → F)."))

        elif nt == 'F':
            # F → a F a | M
            if rng.random() < p_recurcao:
                a = rng.choice(NOTAS)
                forma[i:i+1] = [a, 'F', a]
                trace.append((f"F ⇒ {a} F {a}", f"Seleciona-se {a} como
nota de espelhamento (F → {a} F {a})."))
            else:
                forma[i:i+1] = ['M']
                trace.append(("F ⇒ M", "Encerra-se a recursão e escolhe-se
o motivo central (F → M)."))

        elif nt == 'M':
            # M → N | aa (duas notas idênticas) – para preservar
            palíndromo também em comprimento par
            if rng.choice([True, False]):
                forma[i:i+1] = ['N']
                trace.append(("M ⇒ N", "Motivo central com uma nota (M →
N)."))
            else:
                a = rng.choice(NOTAS)
                forma[i:i+1] = [a, a]
                trace.append((f"M ⇒ {a} {a}", f"Motivo central com duas

```

```

notas idênticas (M → {a} {a})."))

    elif nt == 'N':
        # N → a
        a = rng.choice(NOTAS)
        forma[i:i+1] = [a]
        trace.append((f"N ⇒ {a}", f"Seleciona-se a nota {a} (N →
{a})."))

    return forma, trace

if __name__ == "__main__":
    tokens, trace = gerar_composicao(seed=None, p_recurcao=0.6,
max_passos=50)
    print("Composição gerada:", " ".join(tokens))
    print("\nRastro de derivação (2 colunas, separadas por TAB):\n")
    for lhs, rhs in trace:
        print(f"{lhs}\t{rhs}")

```

Apêndice 3: Programa 3 – Gramática do Tipo 1

A gramática aqui desenvolvida garante que três blocos motivicamente distintos apareçam com o mesmo número de ocorrências, produzindo uma dependência global de contagem que não poderia ser assegurada por uma gramática meramente regular. Em termos composicionais, esse tipo de gramática é útil quando se deseja controlar simetrias, correspondências, recorrências proporcionais ou transformações condicionadas pelo contexto.

```
import random
from typing import Dict, List, Tuple

def apply_once(s: str, lhs: str, rhs: str) -> Tuple[bool, str]:
    """Aplica a primeira ocorrência de lhs->rhs, se existir."""
    idx = s.find(lhs)
    if idx == -1:
        return False, s
    return True, s[:idx] + rhs + s[idx + len(lhs):]

def derive_an_bn_cn(n: int, record: bool = True) -> Tuple[str, List[str]]:
    """
    Derivação controlada (Tipo 1) para gerar  $L = \{ a^n b^n c^n \mid n \geq 1 \}$ .
    Regras usadas (todas não-contrativas):
    S -> aSBC | abc
    CB -> BC
    cB -> Bc          (permuta adicional necessária)
    aB -> ab
    bB -> bb
    bC -> bc
    cC -> cc
    """
    if n < 1:
        raise ValueError("n deve ser >= 1.")

    trace: List[str] = []
    s = "S"
    if record:
        trace.append(s)

    # 1) Expandir S -> aSBC (n-1) vezes
    for _ in range(n - 1):
        ok, s = apply_once(s, "S", "aSBC")
        if not ok:
            raise RuntimeError("Falha ao aplicar S -> aSBC.")
        if record:
            trace.append(s)

    # 2) Fechar com S -> abc
    ok, s = apply_once(s, "S", "abc")
    if not ok:
        raise RuntimeError("Falha ao aplicar S -> abc.")
    if record:
        trace.append(s)
```

```

        trace.append(s)

# 3) Reordenar B e C: mover C para a direita de B (CB -> BC)
while "CB" in s:
    ok, s2 = apply_once(s, "CB", "BC")
    if not ok:
        break
    s = s2
    if record:
        trace.append(s)

# 4) Regra adicional: mover B para a esquerda quando estiver após c (cB
-> Bc)
# Isso garante que B fique em contexto onde possa virar b (aB ou bB).
while "cB" in s:
    ok, s2 = apply_once(s, "cB", "Bc")
    if not ok:
        break
    s = s2
    if record:
        trace.append(s)

# 5) Converter B -> b com contexto (aB->ab e bB->bb)
while "B" in s:
    if "aB" in s:
        ok, s = apply_once(s, "aB", "ab")
    elif "bB" in s:
        ok, s = apply_once(s, "bB", "bb")
    else:
        # Se ainda sobrar B, tentamos trazê-lo para o contexto correto
        via cB->Bc
        if "cB" in s:
            ok, s = apply_once(s, "cB", "Bc")
        else:
            raise RuntimeError(f"Falha ao converter B. Forma atual:
{s}")
    if record:
        trace.append(s)

# 6) Converter C -> c com contexto (bC->bc e cC->cc)
while "C" in s:
    if "bC" in s:
        ok, s = apply_once(s, "bC", "bc")
    elif "cC" in s:
        ok, s = apply_once(s, "cC", "cc")
    else:
        raise RuntimeError(f"Falha ao converter C. Forma atual: {s}")
    if record:
        trace.append(s)

# Agora s é terminal: a^n b^n c^n
return s, trace

def motivos_to_pitch_classes(symbols: str, motivos: Dict[str, List[int]]) ->
List[int]:
    """Converte 'aaabbbcccc' em uma sequência de PCs concatenando
    motivos."""
    pcs: List[int] = []
    for ch in symbols:
        if ch not in motivos:

```

```

        raise ValueError(f"Símbolo terminal inesperado: {ch}")
    pcs.extend([p % 12 for p in motivos[ch]])
    return pcs

if __name__ == "__main__":
    # n controla quantas vezes cada motivo aparece: a^n b^n c^n
    n = random.randint(2, 5)

    # Motivos como classes de alturas (edite à vontade)
    motivos = {
        "a": [0, 2, 4],      # motivo a
        "b": [7, 5, 4],      # motivo b
        "c": [9, 11, 0, 2],  # motivo c
    }

    cadeia, derivacao = derive_an_bn_cn(n=n, record=True)
    pcs = motivos_to_pitch_classes(cadeia, motivos)

    print(f"n = {n}")
    print("Sequência de motivos:", " ".join(list(cadeia))) # ex.: a a a b
b b c c c
    print("Classes de alturas:", pcs)

    # Mostra derivação (útil para quadro)
    print("\nDerivação (formas sentenciais):")
    for i, s in enumerate(derivacao):
        print(f"{i:02d}: {s}")

```

Apêndice 4: Programa 4 – Gramática do Tipo 0

O programa apresentado neste Apêndice 4 tem função demonstrativa no âmbito da discussão teórica sobre a hierarquia de Chomsky; os programas efetivamente empregados na geração dos materiais do primeiro movimento são apresentados nos Apêndices 6 e 7.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Programa 4 - Demonstração didática de uma Gramática do Tipo 0.

Gramática G0 = (N, T, P, S)

N = {S, D, X}
T = {Dó, Ré, Mi, Sol, Lá}
S = símbolo inicial

P:
    S → D S | X S | D | X
    D → Dó | Ré | Mi | Sol | Lá
    X D → X
    X → ε

A regra X D → X é contrativa: ela reduz o comprimento da forma sentencial.
A regra X → ε realiza apagamento. Ambas são admitidas em uma Gramática
do Tipo 0.

Parâmetros da execução publicada:
    Python: 3.12.12
    seed: 11
    max_passos: 60
    minimo_notas: 7
    p_contracao: 1.0

Com esses parâmetros, a saída é reprodutível, contém uma ocorrência real
da regra contrativa X D → X e produz uma sequência terminal musicalmente
mais informativa.
"""

from __future__ import annotations

import random
from dataclasses import dataclass
from typing import Dict, List, Optional, Tuple

TERMINAIS = ["Dó", "Ré", "Mi", "Sol", "Lá"]
NAO_TERMINAIS = {"S", "D", "X"}

@dataclass
class Config:
    seed: Optional[int] = 11
    max_passos: int = 60
```

```

minimo_notas: int = 7

# Probabilidades usadas depois que a contração já ocorreu
# e a quantidade mínima de notas já foi atingida.
p_S_to_DS: float = 0.50
p_S_to_XS: float = 0.20
p_S_to_D: float = 0.25
p_S_to_X: float = 0.05

# Probabilidade de aplicar  $X D \rightarrow X$  quando esse padrão aparece.
p_contracao: float = 1.0

def tokens_to_str(tokens: List[str]) -> str:
    """Converte a lista de símbolos em string. Usa ε para a cadeia vazia."""
    return " ".join(tokens) if tokens else "ε"

def contar_terminais(tokens: List[str]) -> int:
    """Conta quantas notas terminais já existem na forma sentencial."""
    return sum(1 for tok in tokens if tok in TERMINAIS)

def escolher_nota(rng: random.Random) -> str:
    """Escolhe uma nota terminal."""
    return rng.choice(TERMINAIS)

def escolher_regra_S(
    rng: random.Random,
    cfg: Config,
    houve_contracao: bool,
    n_terminais: int,
) -> List[str]:
    """
    Escolhe uma produção para S.

    Enquanto a quantidade mínima de notas ainda não foi atingida,
    favorece  $S \rightarrow D S$ , para produzir uma saída musicalmente mais clara.

    Antes que uma contração tenha ocorrido, força ocasionalmente  $S \rightarrow X S$ ,
    para criar a possibilidade de formação do padrão  $X D$ .
    """

    # Antes da contração, introduz X para permitir a formação posterior de
    #  $X D$ .
    if not houve_contracao and n_terminais >= 3:
        return ["X", "S"]

    # Até atingir o número mínimo de notas, continua expandindo.
    if n_terminais < cfg.minimo_notas:
        return ["D", "S"]

    # Depois disso, usa as probabilidades normais.
    r = rng.random()

    limite_1 = cfg.p_S_to_DS
    limite_2 = limite_1 + cfg.p_S_to_XS
    limite_3 = limite_2 + cfg.p_S_to_D

```

```

if r < limite_1:
    return ["D", "S"]      # S → D S
if r < limite_2:
    return ["X", "S"]      # S → X S
if r < limite_3:
    return ["D"]           # S → D
return ["X"]              # S → X

def derivar(cfg: Config) -> Tuple[List[str], List[Dict[str, str]]]:
    """
    Executa uma derivação da gramática G0.

    Retorna:
        finais: lista final de terminais;
        trace: lista de dicionários com passo, forma, regra e comentário.
    """
    rng = random.Random(cfg.seed)

    tokens: List[str] = ["S"]
    houve_contracao = False

    trace: List[Dict[str, str]] = [
        {
            "passo": "00",
            "forma": tokens_to_str(tokens),
            "regra": "(início)",
            "comentario": "Expressão inicial: S.",
        }
    ]

    for passo in range(1, cfg.max_passos + 1):
        aplicado = False

        # 1. Regra contrativa Tipo 0: X D → X.
        # Tem prioridade sempre que o padrão X D aparece.
        for i in range(len(tokens) - 1):
            if tokens[i] == "X" and tokens[i + 1] == "D":
                if rng.random() < cfg.p_contracao:
                    tokens = tokens[:i] + ["X"] + tokens[i + 2:]
                    houve_contracao = True
                    trace.append(
                        {
                            "passo": f"{passo:02d}",
                            "forma": tokens_to_str(tokens),
                            "regra": "X D → X",
                            "comentario": "Contração: X absorve D e reduz o
comprimento da forma sentencial.",
                        }
                    )
                aplicado = True
                break

        if aplicado:
            continue

        # 2. Caso exista X S antes da primeira contração, expande-se o S à
        direita de X.
        # Isso permite formar X D e demonstrar a regra contrativa.

```

```

idx_xs = None
if not houve_contracao:
    for i in range(len(tokens) - 1):
        if tokens[i] == "X" and tokens[i + 1] == "S":
            idx_xs = i + 1
            break

if idx_xs is not None:
    n_terminais = contar_terminais(tokens)
    rhs = ["D", "S"]
    tokens = tokens[:idx_xs] + rhs + tokens[idx_xs + 1:]
    trace.append(
        {
            "passo": f"{passo:02d}",
            "forma": tokens_to_str(tokens),
            "regra": "S → D S",
            "comentario": "Expansão de S à direita de X, formando o
contexto para X D → X.",
        }
    )
    continue

# 3. Caso geral: reescreve-se o primeiro não-terminal à esquerda.
idx = next((i for i, tok in enumerate(tokens) if tok in
NAO_TERMINAIS), None)

if idx is None:
    break

simbolo = tokens[idx]

if simbolo == "S":
    n_terminais = contar_terminais(tokens)
    rhs = escolher_regra_S(rng, cfg, houve_contracao, n_terminais)
    tokens = tokens[:idx] + rhs + tokens[idx + 1:]

    if rhs == ["D", "S"]:
        comentario = "Expansão: acrescenta um slot de nota e mantém
o processo ativo."
    elif rhs == ["X", "S"]:
        comentario = "Expansão: introduz o marcador X,
característico da demonstração Tipo 0."
    elif rhs == ["D"]:
        comentario = "Encerramento: acrescenta o último slot de
nota."
    else:
        comentario = "Encerramento: introduz X, que poderá ser
apagado."

    trace.append(
        {
            "passo": f"{passo:02d}",
            "forma": tokens_to_str(tokens),
            "regra": f"S → {tokens_to_str(rhs)}",
            "comentario": comentario,
        }
    )

elif simbolo == "D":
    nota = escolher_nota(rng)

```

```

tokens = tokens[:idx] + [nota] + tokens[idx + 1:]
trace.append(
    {
        "passo": f"{passo:02d}",
        "forma": tokens_to_str(tokens),
        "regra": f"D → {nota}",
        "comentario": "Terminação: D é substituído por uma
nota.",
    }
)

elif simbolo == "X":
    tokens = tokens[:idx] + tokens[idx + 1:]
    trace.append(
        {
            "passo": f"{passo:02d}",
            "forma": tokens_to_str(tokens),
            "regra": "X → ε",
            "comentario": "Apagamento: o marcador X é eliminado.",
        }
    )

return tokens, trace

def main() -> None:
    cfg = Config(seed=11, max_passos=60, minimo_notas=7)
    finais, trace = derivar(cfg)

    print("Gramática Tipo 0 - demonstração")
    print("Terminais finais:", tokens_to_str(finais))
    print()
    print("Passo\tForma sentencial\tRegra aplicada\tComentário")

    for row in trace:
        print(
            f"{row['passo']}\t"
            f"{row['forma']}\t"
            f"{row['regra']}\t"
            f"{row['comentario']}"
        )

if __name__ == "__main__":
    main()

```

Apêndice 5: Passo a passo do programa que realiza uma Gramática do Tipo 0

<i>Terminais finais: Sol Lá Sol Lá Ré Ré Lá Lá Dó</i>			
<i>Parâmetros: seed = 11; max_passos = 60; mínimo_notas = 7; p_contração = 1.0</i>			
Passo	Forma sentencial	Regra aplicada	Comentário
00	S	(início)	Expressão inicial: S.
01	D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
02	Sol S	$D \rightarrow \text{Sol}$	Terminação: D é substituído por uma nota.
03	Sol D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
04	Sol Lá S	$D \rightarrow \text{Lá}$	Terminação: D é substituído por uma nota.
05	Sol Lá D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
06	Sol Lá Sol S	$D \rightarrow \text{Sol}$	Terminação: D é substituído por uma nota.
07	Sol Lá Sol X S	$S \rightarrow X S$	Expansão: introduz o marcador X, característico da demonstração Tipo 0.
08	Sol Lá Sol X D S	$S \rightarrow D S$	Expansão de S à direita de X, formando o contexto para $X D \rightarrow X$.
09	Sol Lá Sol X S	$X D \rightarrow X$	Contração: X absorve D e reduz o comprimento da forma sentencial.
10	Sol Lá Sol S	$X \rightarrow \epsilon$	Apagamento: o marcador X é eliminado.
11	Sol Lá Sol D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
12	Sol Lá Sol Lá S	$D \rightarrow \text{Lá}$	Terminação: D é substituído por uma nota.
13	Sol Lá Sol Lá D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
14	Sol Lá Sol Lá Ré S	$D \rightarrow \text{Ré}$	Terminação: D é substituído por uma nota.
15	Sol Lá Sol Lá Ré D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
16	Sol Lá Sol Lá Ré Ré S	$D \rightarrow \text{Ré}$	Terminação: D é substituído por uma nota.
17	Sol Lá Sol Lá Ré Ré D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
18	Sol Lá Sol Lá Ré Ré Lá S	$D \rightarrow \text{Lá}$	Terminação: D é substituído por uma nota.
19	Sol Lá Sol Lá Ré Ré Lá D S	$S \rightarrow D S$	Expansão: acrescenta um slot de nota e mantém o processo ativo.
20	Sol Lá Sol Lá Ré Ré Lá Lá S	$D \rightarrow \text{Lá}$	Terminação: D é substituído por uma nota.
21	Sol Lá Sol Lá Ré Ré Lá Lá D	$S \rightarrow D$	Encerramento: acrescenta o último slot de nota.
22	Sol Lá Sol Lá Ré Ré Lá Lá Dó	$D \rightarrow \text{Dó}$	Terminação: D é substituído por uma nota.

Apêndice 6: Programa 1 da Gramática Tipo 3

```

from random import *
from copy import *

def T():

    cromas = ['0','1','2','3','4','5','6','7','8','9','A','B']
    T = choice(cromas)

    return T

def S():

    option = choice(['T', 'TS'])

    return option

iterations = 10
palavra = ''

for i in range(iterations):

    tree = S()

    if tree=='T':

        palavra = palavra + T() + '.'

    else:

        subtree = deepcopy (tree)

        while subtree!='T':

            palavra = palavra + T()
            subtree = S()

        palavra = palavra + '.'

print(palavra)

```

Apêndice 7: Programa 2 da Gramática Tipo 3

```

import random
from music21 import *

def generate_music(grammar, current_symbol):
    if current_symbol in grammar:
        production = random.choice(grammar[current_symbol])
        result = ''
        for symbol in production:
            result = result + generate_music(grammar, symbol)
        return result
    else:
        return current_symbol

# Definindo a gramática regular
music_grammar = {
    'S': ['08B', 'C'],
    'B': ['12C', '895'],
    'C': ['359D', '14B'],
    'D': ['46E', ],
    'E': ['7abS'] }

# Traduzindo as classes de alturas para inteiros
def pc (x):

    if x=='a': nota = 10
    elif x == 'b': nota = 11
    else:
        nota = int(x)
    return nota

# Gerando uma sequência musical
generated_music = generate_music(music_grammar, 'S')
print("Sequência Musical Gerada:", generated_music)

# Convertendo para notação musical
melodia = stream.Stream()

for n in range(len(generated_music)):
    nota = note.Note(pc(generated_music[n])+60 + random.choices([0,12],
weights=[0.8,0.2], k=1) [0])
    if n != len(generated_music)-1:
        nota.quarterLength = random.choices([0.5,1,2],
weights=[0.7,0.2,0.1], k=1) [0]
    else:
        nota.quarterLength = 2
    melodia.append(nota)

```

Apêndice 8: Primeiro movimento de *Trivium*

Trivium 1. Gramática

$\text{♩} = 112$

Flauta doce
Contralto Helder

mf

05/2 359467ab0812359467ab1412359467ab081214895

Segmentos
(Programa 2)

4

4

01/1 7.4.9.21.88684.4.2.6.8.3.

8

p *mp*

8

12

sputato ord.

p *cresc.* *f* *dim.*

03/2 359467ab359467ab359467ab08895

12

02/1 5.7.3.4.9.1.633B0A.A.5.B.

sputato ord. rit. a tempo

17 *mp* *p* *cresc.* *mf*

02/2 0812359467ab359467ab08895

flz. ord. flz.

21 *p* *cresc.* *f* *p*

03/1 8.6.9.4.086.A.6.0.B.A.

ord.

26 *mp* *mf* *p* *cresc.*

10/2 359467ab081214895

04/1 B.6.28431.2.34.4.4.5.4.3.

rit.

31 *f* *mp* *p* *cresc.* *f*

The image displays a musical score for the piece "Trivium" by Liduino Pitombeira, specifically for the alto flute part. The score is written in 4/4 time and consists of two systems of staves. The first system begins at measure 37 and includes a dynamic marking of *p* (piano) and a crescendo instruction (*cresc.*). The second system begins at measure 40 and includes a dynamic marking of *p* and a tempo instruction of *molto rall.* (molto rallentando). The score features various musical notations, including notes, rests, and slurs. A sequence of numbers, 09/2 081214121412359467ab14895, is written above the second staff. The score concludes with a double bar line.

37 *p* *cresc.* *f*

37 09/2 081214121412359467ab14895

40 *p* *molto rall.*

40

Apêndice 9: Segundo movimento de *Trivium*

Trivium 2. Retórica

Flauta doce
Contralto Helder

S^0 S^1

S^2 S^3 S^4 S^5 Tirata

S^6

S^7 S^8

S^9 S^{10}

S^{11} S^{12}

Apêndice 10: Programa de operações booleanas

```
# -*- coding: utf-8 -*-
"""
Created on Sat May 21 19:07:03 2022

"""

A = {0,2,4,7,9}
B = {1,3,5,7,9,11}
C = {4,6,7,8,10,11}

R = A|B|C

def n(x):

    return R - x

ordem = [R, A, n(A), B, n(B), C, n(C), A&B, B&C, (n(A)&n(B))|A&B, A&B&C,
        n(A)&n(B)&C, ((n(A)&n(B))|A&B)&C, n(B)&n(C), n(n(A)&n(B)|A&B),
        (n(n(A)&n(B)|A&B))&n(C), (n(n(A)&n(B)|A&B))&C, A&n(B)&n(C),
n(A)&n(C),
        n(A)&B&n(C), (((n(A)&n(B))|A&B)&C)|((n(n(A)&n(B)|A&B))&n(C)))]

nomes = ['R', 'A', '~A', 'B', '~B', 'C', '~C', 'A.B', 'B.C', '~(A~B) +
A.B', 'A.B.C',
        '~A~B.C', '~(A~B + A.B).C', '~B~C', '~(A~B + A.B)',
        '~((A~B + A.B))~C', '~((A~B + A.B)).C', 'A~B~C', '~A~C',
        '~A.B~C', 'F = (A.B + ~A~B).C + ~(A.B + ~A~B).~C']

for i in range(len(ordem)):
    print(nomes[i], '=', ordem[i])
```

Apêndice 11: Terceiro movimento de *Trivium*

Trivium

3. Lógica

Flauta doce
Contralto Helder

$\text{♩} = 117$

A

1 2 3

mf *dim.*

4 5

p *cresc.*

B

6 7 8 9

f *p* *+ voz*

10 11 12

mf *p* *mf* *p* *flz. 3:2* *ord.* *mp*

13 14

dim. *p* *cresc.*

A'

15 16

mf

17 18 19 20

mf *p*

21

rall. *dim.* *p*

Apêndice 12: Partitura de *Trivium*

Trivium

1. Gramática

$\text{♩} = 112$

Flauta doce
Contralto Helder

mf

4

p

9

mp

p *cresc.*

sputato ord.

14

f *dim.* *mp* *rit.* *a tempo* *p* *cresc.*

sputato ord.

20

mf *p* *cresc.* *f*

flz. ord.

25

p *mp* *mf* *p* *cresc.*

30

f *mp* *rit.* *p* *cresc.*

8

2

35

Trivium

f *p* *cresc.* *f*

40

p *molto rall.*

8

2. Retórica

Flauta doce
Contralto Helder

$\text{♩} = 48$

mp *cresc.* *f* *p* *mp*

5

cresc. *f* *dim.* *p*

9

mf *p*

12

mp *f* *mp*

14 (♩ = ♩)

mp

19

3. Lógica

$\text{♩} = 117$

Flauta doce
Contralto Helder

mf

6 *dim.* *p* *cresc.*

13 *f*

18 *p* *mf* *p* *mf* *p* *flz. 3:2*

26 *ord.* *mp* *dim.* *p*

33 *cresc.* *mf*

40

45

51 *rall.* *dim.* *p*