



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ANTÔNIO JOABE ALVES MORAIS

**UMA IMPLEMENTAÇÃO COMPUTACIONAL PARA ANÁLISE DE TONALIDADE
EM PROGRESSÕES HARMÔNICAS VIA SEMÂNTICA DE MUNDOS POSSÍVEIS**

QUIXADÁ
2025

ANTÔNIO JOABE ALVES MORAIS

UMA IMPLEMENTAÇÃO COMPUTACIONAL PARA ANÁLISE DE TONALIDADE EM
PROGRESSÕES HARMÔNICAS VIA SEMÂNTICA DE MUNDOS POSSÍVEIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientador: Prof. Me. Francisco Erivel-
ton Fernandes de Aragão

QUIXADÁ

2025

ANTÔNIO JOABE ALVES MORAIS

UMA IMPLEMENTAÇÃO COMPUTACIONAL PARA ANÁLISE DE TONALIDADE EM
PROGRESSÕES HARMÔNICAS VIA SEMÂNTICA DE MUNDOS POSSÍVEIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Me. Francisco Erivelton Fernandes de
Aragão (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Dois (SIGLA)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Três (SIGLA)

Prof. Dr. XXXXXXX XXXXXX XXXXXXX
Universidade do Membro da Banca Quatro (SIGLA)

AGRADECIMENTOS

Sou grato ao meu Pai Eterno, que, por pura graça e não mérito meu, me escolheu;
Sou grato ao Verbo de Deus, Senhor da minha vida, Redentor da minha alma;
Sou grato ao Santo Espírito, meu Consolador, Guia e Iluminador.

Sou grato ao Deus Triúno, fonte de toda a verdade, de toda a vida, de toda a razão, de toda a beleza. Ele, o Criador da música; dos elementos que possibilitam a sua existência; e da razão humana, proporcionou todos os meios para que eu pudesse redigir este trabalho.

A Ti, Grande Artista, Grande Razão, dedico esta obra. Dedico a Ti a minha vida, obra de tuas mãos. Louvor e Honra, Glória e Poder pertencem a Ti.

Sim, somente a Ti.
Amor da minha vida,
Doçura da minha alma,
Motivo da minha existência.

Também sou grato a todos os meus Irmãos da Igreja Batista Regular em Quixadá; à minha amada mãe, Cecília Alves; ao meu amado pai, Antônio Moraes; à minha amada Kézia Victoria. Agradeço, de todo o meu coração, por serem instrumentos de Deus na formação do meu intelecto, do meu caráter, da minha saúde: física e espiritual.

“Ninguém que veja uma lira belamente elaborada, com sua disposição harmoniosa e ordenada, e ouça a música da lira, deixará de formar uma noção de seu artesão-músico, de recorrer a ele em pensamento, embora o ignore de vista.”

(Gregório Nazianzeno)

RESUMO

A análise computacional da harmonia tonal, um desafio na interseção entre música e computação, frequentemente esbarra na dificuldade de modelar ambiguidades e a dinâmica contextual da música. Diante da necessidade de um formalismo rigoroso e interpretável, o modelo teórico de Aragão (2021), que aplica a semântica de mundos possíveis de Kripke à análise de progressões harmônicas, surge como uma base promissora, porém ainda carente de uma implementação prática que valide sua aplicabilidade. Este trabalho tem como objetivo geral implementar computacionalmente o referido modelo, materializando-o em um sistema de software funcional denominado Tontology. A ferramenta visa não apenas classificar a tonalidade de uma sequência de acordes, mas também fornecer as justificativas lógicas para a análise, explorando seu potencial pedagógico. A metodologia consiste na tradução do formalismo lógico para estruturas de dados e algoritmos, utilizando a linguagem Python e o framework FastAPI para a construção de uma API de análise. O núcleo do sistema implementa um algoritmo de verificação de modelos que, ao receber uma progressão harmônica, avalia recursivamente sua satisfação em relação a uma estrutura de Kripke que modela as funções tonais. A análise permite a exploração de múltiplos contextos tonais (mundos possíveis) para lidar com modulações. O resultado central é o sistema Tontology, uma prova de conceito que valida a viabilidade da abordagem de Aragão e oferece uma ferramenta para análise harmônica transparente e passo a passo, contribuindo com um recurso computacional para o estudo e ensino da teoria musical.

Palavras-chave: Identificação de Tonalidade. Semântica Modal. Model Checking. Raciocínio Harmônico Automático. Educação Musical.

ABSTRACT

The computational analysis of tonal harmony, a challenge at the intersection of music and computing, often struggles with modeling ambiguities and context-dependent musical dynamics. Faced with the need for a rigorous and interpretable formalism, the theoretical model by Aragão (2021), which applies Kripke's possible worlds semantics to the analysis of harmonic progressions, emerges as a promising foundation, yet it lacks a practical implementation to validate its applicability. This work's general objective is to computationally implement the aforementioned model, materializing it into a functional software system named Tonality. The tool aims not only to classify the tonality of a chord sequence but also to provide the logical justifications for the analysis, exploring its pedagogical potential. The methodology consists of translating the logical formalism into data structures and algorithms, using the Python language and the FastAPI framework to build an analysis API. The system's core implements a model-checking algorithm that, upon receiving a harmonic progression, recursively evaluates its satisfaction against a Kripke structure that models tonal functions. The analysis allows for the exploration of multiple tonal contexts (possible worlds) to handle modulations. The main result is the Tonality system, a proof-of-concept that validates the feasibility of Aragão's approach and offers a tool for transparent, step-by-step harmonic analysis, contributing a computational resource for the study and teaching of music theory.

Keywords: Tonality Identification. Modal Semantics. Model Checking. Automatic Harmonic Reasoning. Musical Education.

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	11
1.1.1	<i>Objetivo Geral</i>	11
1.1.2	<i>Objetivos Específicos</i>	11
1.2	Estrutura do Trabalho	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Fundamentos Teóricos da Harmonia Tonal	12
2.2	Lógica Modal e Semântica de Kripke	15
2.3	Verificação de Modelos	17
2.4	Ferramentas e Tecnologias de Implementação	18
3	TRABALHOS RELACIONADOS	21
4	METODOLOGIA	25
4.1	Modelagem Formal	26
4.2	Implementação Computacional	27
4.3	Base de Dados e Testes	28
4.4	Fluxo Operacional do Sistema (Projetado)	29
4.5	Avaliação e Limitações da Abordagem	29
4.6	Cronograma	30
5	PROPOSTA PRELIMINAR DO SISTEMA	31
5.1	Modelagem Formal Detalhada	31
5.1.1	<i>A Estrutura de Kripke como Modelo Semântico</i>	31
5.1.2	<i>A Relação de Satisfação Tonal</i>	33
5.2	Arquitetura e Implementação Projetada	37
5.2.1	<i>Arquitetura Geral do Sistema</i>	37
5.2.2	<i>Componentes Centrais</i>	38
5.3	Visualização da Análise e Fluxo de Dados Projetado	39
5.3.1	<i>Módulo de Visualização da Análise</i>	39
5.3.2	<i>Fluxo Operacional do Sistema</i>	39
	REFERÊNCIAS	41

1 INTRODUÇÃO

A Computação Musical é um campo de interseção entre a Ciência da Computação, a Teoria Musical e a Matemática. Aqui, arte e algoritmo se encontram. Seu escopo é abrangente: aborda desde a geração algorítmica de música até a análise harmônica assistida por computador, passando por aplicações como sistemas de recomendação musical, extração de características de áudio, processamento digital de sinais musicais e modelagem simbólica de estruturas musicais (ROADS, 1996). A Sociedade Brasileira de Computação (2025) define o campo, clarificando que “tem como objetivo fomentar a pesquisa científica, tecnológica e artística nas áreas de composição algorítmica, análise/síntese de som, acústica musical, análise musical assistida por computador, composição musical assistida por computador, processamento digital de áudio, multimídia e qualidade de serviço”. Ao voltar-se para essa área, é possível reconhecer que ela lida, ao mesmo tempo, com arte e abstração, com intuição e lógica — e é nesse território duplo que emerge a presente pesquisa.

Entre os muitos desafios que a computação musical enfrenta, destaca-se a tarefa da análise harmônica. Ainda hoje, essa tarefa é, na maioria das vezes, realizada por especialistas humanos, exigindo sensibilidade estilística, escuta contextual e conhecimento teórico. A progressão de acordes em uma música tonal não é apenas sucessão mecânica: ela carrega tensão, ela constrói expectativa, ela espera repouso (RIEMANN, 1893). O caminhar tem sentido e a estrutura tem direção. Essa dinâmica tonal, por vezes clara, por vezes ambígua, exige modelos computacionais que vão além do aspecto sonoro externo — modelos que considerem as estruturas internas e os significados possíveis que os acordes carregam em seu contexto. Que considere as possibilidades interpretativas. Ao longo dos anos, diversos esforços surgiram: alguns baseados em heurísticas e regras gramaticais, outros em abordagens estatísticas e representações espaciais (CHEW, 2002). Todos, em alguma medida, contribuíram. No entanto, muitos esbarraram na dificuldade de lidar com a ambiguidade; com o modo menor emprestado; o acorde que modula; o contexto que reinterpreta. Com as escolhas interpretativas.

Diante desse cenário, surge a necessidade de um formalismo que seja ao mesmo tempo expressivo, rigoroso e interpretável. É nesse ponto que a lógica modal e, especificamente, a semântica de mundos possíveis proposta por Kripke (1963), se torna uma aliada. A lógica modal permite que proposições não sejam simplesmente verdadeiras ou falsas em termos absolutos, mas dependentes de contextos: os chamados “mundos possíveis”. Ao transpor essa estrutura para o domínio da música tonal, cada tonalidade pode ser representada como um mundo possível.

Os acordes, como fórmulas modais atômicas (estados) que recebem interpretação contextual. As resoluções e movimentos, como relações de acessibilidade entre estados. Essa proposta foi formalizada por Francisco Aragão (2021), que propôs um modelo formal de análise de progressões harmônicas utilizando as semânticas de Kripke. No entanto, essa proposta ainda espera uma implementação prática que permita, de fato, a inferência computacional de tonalidade ao longo de uma progressão harmônica. É nessa brecha que se insere este trabalho.

Para materializar a proposta de Aragão e explorar seu potencial, propõe-se a implementação computacional de seu modelo: um sistema de software. Este sistema, batizado de **Tonality**¹, constitui a contribuição central desta pesquisa. O objetivo do Tonality é desenvolver uma ferramenta que, ao receber uma progressão harmônica (em notação simbólica), seja capaz de inferir, passo a passo, se a progressão é tonal ou não; e, mais do que isso, que forneça todas as possibilidades de justificativas lógicas para cada resultado, capturando assim o aspecto dinâmico da música tonal com o auxílio da lógica.

¹ O neologismo Tonality combina “Tonal” e o sufixo “-logia” para criar uma ambiguidade intencional. Etimologicamente, o sufixo sugere “o estudo de” (a tonalidade), enquanto sua forma evoca a “Lógica” (o método). Essa dualidade reflete a abordagem do trabalho de usar um sistema lógico para analisar as múltiplas interpretações, ou “mundos possíveis”, da harmonia.

1.1 Objetivos

1.1.1 *Objetivo Geral*

Implementar e validar um sistema de software, fundamentado na semântica de mundos possíveis, para a análise e identificação de tonalidades em progressões harmônicas de cenários reais e sintéticos.

1.1.2 *Objetivos Específicos*

Para alcançar tal objetivo, elencam-se os seguintes objetivos específicos:

- **Especificar um modelo computacional** para a análise de tonalidade, formalizando a representação de progressões harmônicas, funções tonais e a semântica de mundos possíveis com base na teoria de Aragão (2021).
- **Implementar o modelo especificado** em um sistema de software funcional, denominado *Tonality*, que exponha suas funcionalidades através de uma API.
- **Desenvolver um mecanismo de visualização** que traduza a derivação lógica da análise harmônica em um diagrama de fluxo didático e interpretável, explorando o potencial pedagógico da ferramenta.
- **Validar as análises** do sistema, comparando os resultados gerados para um corpus de progressões reais e sintéticas com análises de referência da literatura musical.

1.2 Estrutura do Trabalho

Este trabalho, então, estrutura-se como segue: o Capítulo 2 apresenta os fundamentos teóricos necessários, incluindo noções de Teoria Harmônica Tonal, Lógica Modal e Verificação de Modelos. O Capítulo 3 discute trabalhos relacionados, com ênfase em abordagens computacionais para análise de tonalidade. O Capítulo 4 descreve a metodologia adotada, detalhando os princípios de implementação do *Tonality*. Por fim, o Capítulo 5 apresenta os resultados preliminares obtidos por meio da aplicação do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Aqui, são estabelecidos os pilares conceituais que suportam este trabalho. Para erigir a obra proposta — um sistema que dialoga com a música através da linguagem da lógica —, é necessário que os alicerces sejam assentados com clareza e profundidade.

Inicia-se com os **Fundamentos Teóricos da Harmonia Tonal**, buscando desvendar e explicar a gramática interna que guia a composição musical ocidental. Em seguida, apresenta-se o ferramental da **Lógica Modal e Semântica de Kripke**, que permite a modelagem da natureza contextual e diversa da harmonia. Também vai ser explorada a metodologia de **Verificação de Modelos**, demonstrando como este campo pode ser adaptado para resolver o problema apresentado. Por fim, as **Ferramentas e Tecnologias de Implementação** são apresentadas e justificadas.

2.1 Fundamentos Teóricos da Harmonia Tonal

A música, em sua essência, tem sido frequentemente compreendida como uma forma de comunicação que, semelhantemente à linguagem verbal, possui uma gramática e uma estrutura que organizam os sons de maneira a criar sentido, emoção e narrativa (LERDAHL; JACKENDOFF, 1983). Essa perspectiva permite a abordagem da análise musical usando ferramentas formais que buscam entender e expôr as regras de tal organização. Para o presente trabalho, que visa analisar essa estrutura computacionalmente, é fundamental que primeiro se compreenda essa gramática. Esta seção se dedica a apresentar, didática e progressivamente, os conceitos da harmonia tonal, o sistema que governou a música ocidental por séculos e que ainda hoje permeia a maior parte do que ouvimos (KOSTKA; PAYNE, 2008). O objetivo aqui é construir o alicerce teórico para a compreensão da lógica por trás da análise que o sistema se propõe a fazer.

Toda a estrutura da música tonal se organiza em torno de uma ideia central: a existência de um “lar” sonoro, um ponto de estabilidade de repouso. Este ponto é chamado de **tônica**. É uma nota específica que serve como centro de gravidade para todas as outras. A sensação de “jornada” e “chegada” sentida ao ouvir uma canção é, em grande parte, a arte empregada no afastar e retornar a essa tônica.

Este princípio de um centro de gravidade é a pedra angular de um sistema organizacional mais amplo, conhecido como Tonalidade. A Tonalidade pode ser definida como um sistema

hierárquico de relações musicais que organiza notas e acordes em torno de um centro focal de repouso — a tônica. Dentro deste sistema, cada elemento musical possui uma função e um grau de tensão definidos por sua relação com esse centro, criando uma sintaxe de expectativas que governa a dinâmica de tensão e resolução percebida pelo ouvinte (KOSTKA; PAYNE, 2008).

Para definir qual é a tônica e quais são as outras notas “válidas em uma peça”, utiliza-se um conjunto ordenado de notas chamado escala. A escala diatônica maior, por exemplo, é composta por sete notas, sendo a primeira delas a tônica. A partir das notas da escala, construímos estruturas mais complexas que são os verdadeiros pilares da harmonia: os acordes. Um acorde, em sua essência, é um agrupamento conceitual de três ou mais notas musicais. No contexto tonal, a constituição de um acorde fundamental se dá a partir de uma nota arbitrária (a fundamental), sobrepondo-se as notas de terça e quinta. Sobre cada grau da escala diatônica, é possível formar um acorde distinto, cada um com uma “personalidade” sonora e uma função potencial, como ilustrado na Tabela 1 para a tonalidade de Dó Maior.

Tabela 1 – Acordes Diatônicos e Suas Qualidades em Dó Maior

Grau	Tríade	Qualidade da Tríade
I	C	Maior
ii	Dm	menor
iii	Em	menor
IV	F	Maior
V	G	Maior
vi	Am	menor
vii°	Bdim	diminuta

Fonte: Adaptado de Kostka e Payne (2008)

Descrição: Uma tabela com colunas para "Grau", "Tríade", "Qualidade da Tríade". Cada linha corresponde a um grau da escala de Dó Maior (I a VII), mostrando o acorde correspondente e sua qualidade.

A sucessão desses acordes, a progressão harmônica, não é aleatória. Ela obedece a uma sintaxe baseada nas funções harmônicas. Esta teoria, sistematizada por teóricos como Riemann (1893), postula que os acordes desempenham papéis sintáticos dentro de uma frase musical, agrupando-se em três categorias primárias que governam a dinâmica de tensão e resolução: a Tônica (T), que representa a estabilidade (I, vi¹, iii); a Dominante (D), a tensão máxima (V, vii^{o2}, iii³); e a Subdominante (S), que atua como preparação para a dominante (IV, ii). A interação entre estas funções cria um modelo de frase canônico, $T \rightarrow S \rightarrow D \rightarrow T$, que descreve o fluxo de tensão e resolução mais comum e perceptualmente eficaz. A pontuação

¹ Letras minúsculas indicam que o acorde é *menor*: possui a estrutura "tônica - terça menor - quinta".

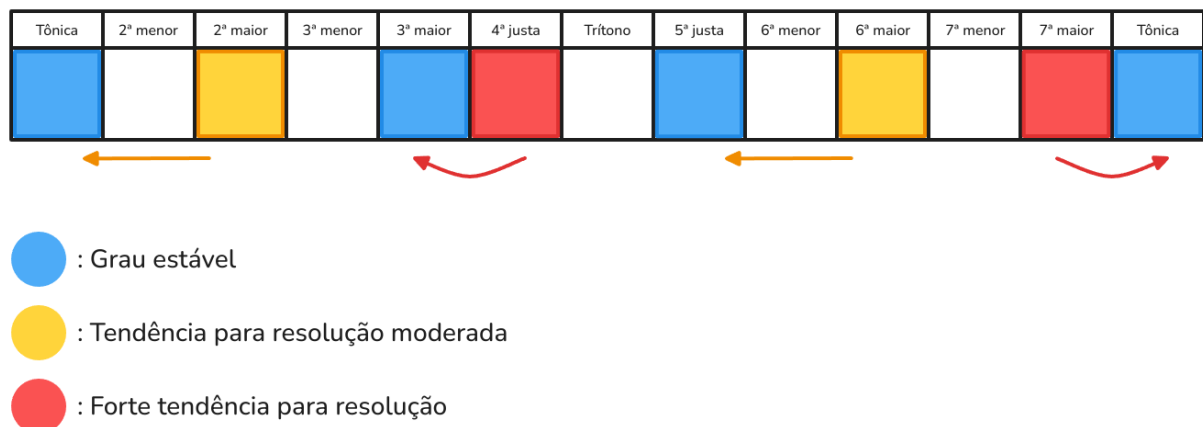
² Acorde *diminuto*: "tônica - terça menor - quinta diminuta".

³ O acorde *iii* pode desempenhar ambos os papéis de *tônica* e *dominante* (embora não o faça fortemente e dependa do contexto) porque compartilha duas notas com a tônica, mas contém o sétimo grau da escala, que demanda resolução. (ver Figura 1)

final desse discurso é realizada pelas cadências, fórmulas harmônicas que concluem as frases. A Cadência Autêntica (V-I), por exemplo, é o gesto mais conclusivo, confirmando o centro tonal. Para que nosso sistema possa identificar uma tonalidade, ele precisa reconhecer esses gestos sintáticos.

Mas por que, então, certos acordes geram tensão em relação ao próximo na sequência e outros, repouso em relação ao anterior? A resposta está na hierarquia das notas individuais que os compõem. Cada grau da escala possui uma “distância” e uma “tensão” específicas em relação à tônica, manifestando tendências de movimento. A prática e a teoria musical tonal, como sistematizadas por teóricos como Kostka e Payne (2008) e Laitz (2011), reconhecem que certas notas são percebidas como mais estáveis – notadamente a *quinta* e a *terça* em relação à fundamental de um acorde maior, e, claro, à própria *tônica*. Outras notas carregam consigo um impulso para a resolução: o *quarto* grau da escala frequentemente busca repouso no *terceiro*; o *segundo* grau tende a resolver na *tônica*; o *sexto* grau muitas vezes se move em direção ao *quinto*; e, de forma mais enfática, o *sétimo* grau possui uma forte atração ascendente pela *tônica*, conforme ilustrado na Figura 1. São essas tensões e resoluções melódicas individuais que, em conjunto, tecem a complexa rede da harmonia.

Figura 1 – Tendências de Resolução dos Graus na Escala Maior



Descrição: Um diagrama exibindo os graus de uma escala maior e suas resoluções. Setas indicam as tendências de movimento e resolução dos graus: O 4º grau resolvendo descendentemente para o 3º grau; o 2º grau resolvendo descendentemente para o 1º; o 6º grau resolvendo descendentemente para o 5º; e o 7º grau resolvendo ascendentemente com forte atração para o 1º grau. O 1º, 3º e 5º graus, componentes do acorde de tônica, são implicitamente apresentados como pontos de estabilidade.

Fonte: Adaptado de Dalessio (2020), conceitos de Laitz (2011)

Graças à organização hierárquica intrínseca desses graus da escala e às suas tendências de resolução em relação à tônica, a música tonal adquire sua profunda direcionalidade e sua capacidade expressiva. Não se trata de uma mera coleção de notas; cada grau possui uma

identidade e um potencial de movimento que é definido, em grande medida, por sua relação intervalar com o centro tonal e com os outros graus. Laitz (2011) enfatiza que essa hierarquia não é apenas uma construção teórica, mas um reflexo da percepção auditiva, onde certos graus soam mais estáveis e outros carregam uma tensão que busca resolução.

Compreendendo essa dinâmica, a relação entre o acorde Dominante e o acorde Tônica se revela como o pilar da tonalidade. O acorde construído sobre o quinto grau (por exemplo), o Dominante (V), que gera a tensão máxima, pois ele contém o instável sétimo grau da escala. Sua resolução no acorde Tônica (I), que é formado apenas por notas estáveis, torna-se uma consequência de força quase inevitável. Essa relação V-I é o gesto de tensão e resolução mais fundamental. Pensemos no acorde Tônica como um poderoso polo magnético: é o centro de repouso para o qual as outras notas convergem. O Dominante, ao se aproximar, gera a tensão máxima deste campo, tornando sua resolução de volta à tônica uma consequência de força quase inevitável. Schoenberg (1979) explora como a proeminência da tônica e da dominante pode ser rastreada até a própria série harmônica – os sons parciais que ressoam naturalmente acima de uma nota fundamental – onde a quinta justa (que define a dominante) é um dos primeiros e mais fortes harmônicos a surgir após a oitava. Isso sugere que a “lógica” da tonalidade tem raízes que vão além da convenção cultural, tocando em aspectos da física do som e da percepção humana. Outros graus também desempenham papéis cruciais nesta dinâmica. A função tonal subdominante, por exemplo, frequentemente cria um movimento de afastamento da tônica que prepara a chegada da dominante, funcionando como uma espécie de “antetensão”. É crucial entender que essas “tensões” e “distâncias” não são apenas conceitos abstratos; elas se traduzem diretamente na experiência auditiva. A habilidade de um ouvinte, mesmo sem treinamento formal, em antecipar um acorde de resolução ou sentir o “peso” de uma nota fora do esperado, atesta a força dessa gramática internalizada. Portanto, ao falarmos em analisar tonalidade, estamos, em essência, buscando desvendar como essas relações hierárquicas entre as notas são articuladas ao longo do tempo para criar um discurso musical coeso e significativo.

2.2 Lógica Modal e Semântica de Kripke

A natureza contextual da harmonia tonal — onde um mesmo acorde pode pertencer a múltiplos “universos” tonais — exige um ferramental lógico que transcenda a lógica clássica proposicional ou de predicados. A Lógica Modal emerge como uma extensão natural para lidar com tais cenários, introduzindo operadores que qualificam a natureza da verdade de uma

proposição. Os operadores modais canônicos são o de necessidade (\Box), usualmente interpretado como “é necessariamente verdadeiro que”, e o de possibilidade (\Diamond), interpretado como “é possivelmente verdadeiro que”. Estes operadores permitem raciocinar não apenas sobre o que é factualmente verdadeiro, mas sobre o que deve ser, o que poderia ser, ou o que será verdadeiro sob certas condições.

A interpretação formal e rigorosa desses operadores modais foi profundamente transformada pela introdução da semântica de mundos possíveis por Saul Kripke (1963). Antes da semântica de Kripke, as lógicas modais eram primariamente sistemas axiomáticos. Kripke forneceu um modelo semântico intuitivo e poderoso, no qual a verdade de uma fórmula modal é avaliada não de forma absoluta, mas em relação a um “mundo” dentro de uma coleção de mundos possíveis. Uma Estrutura de Kripke (ou modelo de Kripke) é formalmente um grafo onde os nós representam os mundos possíveis (ou estados) e as arestas representam uma relação de acessibilidade entre esses mundos. Esta relação define, para cada mundo, quais outros mundos são “considerados possíveis” ou “relevantes” a partir dele.

Assim, dentro da semântica de Kripke, uma proposição P é considerada necessariamente verdadeira ($\Box P$) em um mundo w se, e somente se, P é verdadeira em todos os mundos w' que são acessíveis a partir de w através da relação de acessibilidade. Analogamente, P é considerada possivelmente verdadeira ($\Diamond P$) em um mundo w se, e somente se, existe pelo menos um mundo w' acessível a partir de w onde P é verdadeira. Blackburn *et al.* (2001) detalham extensivamente a teoria e as aplicações dessas estruturas, demonstrando sua flexibilidade para modelar uma vasta gama de fenômenos em ciência da computação, inteligência artificial, filosofia e linguística.

A seguir, apresenta-se a definição formal da estrutura utilizada, conforme estabelecida por Aragão (2021) — que resume e adapta os conceitos seminais de Kripke (1963) — para o domínio formal que nos interessa.

Uma Estrutura de Kripke $K = \langle S, S_0, S_F, R, \mathbb{L} \rangle$, é definida como segue:

- S : Um conjunto finito e não vazio de estados abstratos. Estes estados representam os elementos discretos ou os “pontos” fundamentais do sistema que está sendo modelado. São os nós do grafo subjacente à estrutura.
- $S_0 \subseteq S$: Um subconjunto de S que define os estados iniciais do sistema. Estes são os pontos de partida designados para qualquer avaliação ou “caminho” dentro da estrutura.
- $S_F \subseteq S$: Um subconjunto de S que define os estados finais ou de aceitação. Estes estados

podem significar a conclusão bem-sucedida de um processo ou a satisfação de uma condição terminal.

- $R \subseteq S \times S$: Uma relação de acessibilidade binária sobre o conjunto de estados S . Esta relação define as transições permitidas entre os estados. Se $\langle s_i, s_j \rangle \in R$, diz-se que o estado s_j é acessível a partir do estado s_i . No modelo de Aragão (2021), uma propriedade importante de R é que ela é total: para cada estado $s \in S$, deve existir pelo menos um estado $s' \in S$ tal que $\langle s, s' \rangle \in R$. Esta propriedade garante que, de qualquer estado, sempre haja uma transição possível.
- \mathbb{L} : Uma coleção de funções de rótulo $L_i : S \rightarrow \mathcal{P}(\Pi)$, onde Π é um conjunto de proposições atômicas ou símbolos básicos, e $\mathcal{P}(\Pi)$ é o conjunto de partes de Π (o conjunto de todos os subconjuntos de Π). Cada função L_i na coleção \mathbb{L} representa um contexto de interpretação específico. Para um dado contexto L_i e um estado $s \in S$, $L_i(s)$ retorna o conjunto de proposições atômicas de Π que são consideradas verdadeiras ou válidas no estado s sob o contexto de interpretação L_i .

Esta estrutura formal, com sua capacidade de representar estados, transições e múltiplas interpretações contextuais através das funções de rótulo, fornece o alicerce matemático para a metodologia de Verificação de Modelos que será explorada na próxima seção.

2.3 Verificação de Modelos

A Verificação de Modelos (*Model Checking*), em seu uso clássico na ciência da computação, é uma técnica automática para determinar se um modelo de um sistema — como o projeto de um circuito de hardware ou um protocolo de comunicação — atende a uma especificação formal, que é uma propriedade desejada (CLARKE *et al.*, 2001). O objetivo primordial é encontrar falhas no modelo do sistema. A ferramenta de verificação responde à pergunta: “Este projeto (modelo) obedece a todas as regras (especificação)?”. Se a resposta for negativa, o verificador tipicamente fornece um contraexemplo: um caminho específico que demonstra exatamente como a falha ocorre, o que é crucial para a depuração.

Neste trabalho, contudo, essa ideia é deliberadamente subvertida, conforme proposto por Aragão (2021). Em vez de usar a especificação para encontrar falhas no modelo, nós usamos o modelo como uma “régua” para avaliar um objeto externo. O modelo, que representa o sistema tonal, é considerado o nosso padrão de correção, a nossa gramática estabelecida. A questão fundamental deixa de ser “O sistema está correto?” e passa a ser “Esta progressão de acordes é

reconhecida como válida por este sistema?”.

No Tonology, portanto, a estrutura de Kripke (K) atua como o modelo do sistema tonal, e a progressão harmônica (ϕ) a ser analisada funciona como a especificação. O processo de verificação busca responder à questão: 'A estrutura K , em um dado contexto tonal L , satisfaz a fórmula ϕ através de um caminho de estados π ?', representada formalmente como:

$$K, L \models_{\pi} \phi$$

Uma das vantagens mais significativas desta abordagem é sua capacidade de gerar explicações. Quando uma progressão é considerada tonalmente coesa, o sistema não apenas retorna “sim”, mas produz uma testemunha (o caminho π) que demonstra, passo a passo, a função de cada acorde e a lógica da análise. Por outro lado, se a progressão falha, um contraexemplo pode ilustrar o ponto exato onde a lógica tonal, segundo o modelo, foi quebrada. Essa capacidade de fornecer “provas” concretas e rastreáveis transforma o verificador em uma poderosa ferramenta de explanação e aprendizado, atingindo em cheio os objetivos didáticos desta pesquisa.

2.4 Ferramentas e Tecnologias de Implementação

Para a materialização do sistema Tonology, a linguagem de programação Python foi selecionada devido a um conjunto de características que se alinham intrinsecamente aos objetivos do projeto. Sua sintaxe clara e legível, um dos pilares de sua filosofia de design (Python Software Foundation, 2025b), facilita não apenas o desenvolvimento inicial, mas também a manutenção e a compreensão do código. Estes são aspectos cruciais em um projeto acadêmico que pode servir de base para trabalhos futuros, onde a clareza da implementação de uma lógica complexa, como a análise de tonalidade via semântica de mundos possíveis, é essencial. A filosofia de design do Python, encapsulada em princípios como “Simples é melhor que complexo” e “Legibilidade conta” (PETERS, 2004), é mais do que estética, impactando diretamente a produtividade do desenvolvedor e a manutenibilidade do código a longo prazo. Para um projeto como o Tonology, que envolve a tradução de um formalismo lógico sofisticado em código, esta filosofia minimiza a carga cognitiva associada ao desenvolvimento, facilita a depuração e simplifica a eventual evolução do sistema. Ademais, a vasta biblioteca padrão e o rico ecossistema de pacotes de terceiros, disponíveis através do Python Package Index (PyPI) (Python Software Foundation, 2025a), oferecem ferramentas prontas para diversas tarefas, desde a manipulação de dados até a construção de interfaces. Isso permite que o foco do desenvolvimento permaneça

na lógica central de análise harmônica, que constitui o núcleo da pesquisa. A linguagem é reconhecidamente versátil, adequando-se bem tanto para prototipagem rápida quanto para o desenvolvimento de sistemas complexos (PATIL; PATIL, 2024). Este vasto ecossistema, que inclui bibliotecas robustas para computação científica e manipulação de dados, permite que o pesquisador se concentre nos aspectos inovadores de sua pesquisa – neste caso, a aplicação da semântica de Kripke à análise tonal – em vez de gastar tempo implementando funcionalidades básicas já existentes e exaustivamente testadas pela comunidade.

Para expor as funcionalidades do Tontology e permitir a interação com o sistema – por exemplo, o envio de progressões harmônicas para análise e a recepção dos resultados dessa análise – optou-se pelo framework FastAPI. Esta escolha fundamenta-se na sua reconhecida alta performance, frequentemente comparada a frameworks em linguagens como Node.js e Go, e em sua arquitetura moderna baseada em ASGI (Asynchronous Server Gateway Interface), que habilita o processamento assíncrono de requisições, um fator determinante para a escalabilidade e responsividade de aplicações web (LATHKAR, 2023). O FastAPI se destaca por sua facilidade de uso e pela rapidez que proporciona ao desenvolvimento (“Fast to code”), permitindo um aumento significativo na velocidade de implementação de funcionalidades (VERBINA, 2025). Uma vantagem é a integração nativa e profunda com a biblioteca Pydantic para validação de dados, juntamente com a geração automática de documentação interativa da API (utilizando Swagger UI e ReDoc). Esta documentação é baseada nos padrões abertos OpenAPI e JSON Schema, o que facilita a compreensão e o teste da API (RAMÍREZ, 2025). A geração automática de documentação da API não é apenas uma conveniência para os eventuais consumidores dessa interface, mas configura-se como uma ferramenta de desenvolvimento e depuração extremamente poderosa para o próprio pesquisador. Ela permite visualizar e testar interativamente os endpoints e os modelos de dados (definidos com Pydantic) em tempo real, agilizando a verificação da correta “tradução da lógica em código” na interface do sistema, sem a necessidade de construir clientes de teste separados nas fases iniciais de desenvolvimento. A ênfase do FastAPI no uso de type hints padrão do Python para definir modelos de dados (através da sua integração com Pydantic) e parâmetros de rota impõe uma disciplina de “design por contrato” implícita. Isso significa que as expectativas de formato e tipo para os dados de cada endpoint são claramente definidas e, crucialmente, validadas automaticamente na entrada. Para um sistema como o Tontology, que manipula estruturas musicais simbólicas complexas (acordes, progressões), essa validação rigorosa na interface da API é vital para prevenir que dados malformados ou inesperados

comprometam a integridade da lógica de análise modal subsequente. Ao garantir que apenas dados válidos atinjam o núcleo de processamento, o sistema se torna inerentemente mais robusto e confiável.

Complementando o FastAPI, a biblioteca Pydantic desempenha um papel fundamental na arquitetura do Tontology. Ela é responsável pela validação de dados e pelo gerenciamento de configurações através do uso extensivo de type hints do Python (Pydantic Team, 2025; ALLEN, 2024). Todos os dados que fluem para dentro e para fora da API, bem como as estruturas de dados internas que representam elementos musicais cruciais para a análise (como acordes, tonalidades e as próprias progressões harmônicas), são modelados como classes Pydantic. Esta abordagem garante que os dados estejam sempre no formato esperado, prevenindo uma vasta gama de erros em tempo de execução e, conseqüentemente, aumentando a robustez geral do sistema Tontology. Uma característica particularmente valiosa do Pydantic é a geração de mensagens de erro claras e detalhadas quando uma validação falha, especificando exatamente qual dado é inválido e por quê, o que simplifica enormemente o processo de depuração (D., 2025). A sua integração com FastAPI é transparente e eficiente, permitindo que a validação dos dados de requisição e resposta ocorra automaticamente, sem a necessidade de código de validação explícito e repetitivo nos manipuladores de rota (RAMÍREZ, 2025). O uso de Pydantic para modelar as entidades musicais com type hints não apenas garante a validação dos dados, mas também serve como uma forma de “documentação viva” da estrutura de dados do domínio do problema.

Em suma, a pilha tecnológica do projeto foi concebida para ser um reflexo fiel e eficiente do modelo teórico. A união de Python, FastAPI e Pydantic proporciona uma plataforma que é, ao mesmo tempo, ágil para a pesquisa e robusta para a aplicação, permitindo o foco no que é mais importante: a correção e a profundidade da análise harmônica realizada pelo sistema.

3 TRABALHOS RELACIONADOS

A análise computacional da harmonia tonal constitui um campo de pesquisa com diversas soluções propostas ao longo do tempo. Esta seção dedica-se a contextualizar a presente proposta – uma implementação computacional para análise de tonalidade em progressões harmônicas via semântica de mundos possíveis – em relação ao panorama da literatura, organizando a discussão por afinidade metodológica e destacando, para cada contribuição relevante, seus objetivos, bem como suas semelhanças e diferenças em relação a este trabalho.

O alicerce teórico do Tonology é a proposta de Aragão (2021), intitulada “Identification of Tonal Progressions Through Kripke Semantics”. Este trabalho introduz um modelo formal para a análise de progressões harmônicas e para a identificação de tonalidades, utilizando a semântica de mundos possíveis de Kripke e a técnica de verificação de modelos. No formalismo de Aragão, cada tonalidade é concebida como um “mundo possível”, e as cadências são tratadas como relações de acessibilidade entre os acordes desses mundos, permitindo que o sistema explore diferentes interpretações contextuais de uma sequência de acordes de maneira sistemática e, crucialmente, explicável. O presente trabalho constitui uma implementação computacional direta e uma avaliação do modelo teórico proposto por Aragão. Ambos os trabalhos compartilham o mesmo fundamento (semântica de Kripke, lógica modal, verificação de modelos) e o objetivo de analisar a tonalidade de progressões harmônicas simbólicas de forma transparente. A principal diferença reside no foco: enquanto Aragão (2021) estabelece o arcabouço teórico, a pesquisa corrente concentra-se na sua tradução para um sistema de software funcional, na sua aplicação prática a exemplos musicais e na exploração do seu potencial didático através da geração de explicações passo a passo do processo analítico.

Paralelamente à abordagem de Aragão, outras iniciativas buscaram formalizar a análise harmônica por meios simbólicos e estruturais. Destaca-se, nesse contexto, o trabalho de Paulo Vilar (2005), que em sua dissertação explorou o “Reconhecimento Harmônico com Gramáticas Livre de Contexto” (sistema denominado “AHO”). Esta pesquisa investigou a aplicação de gramáticas formais para modelar e reconhecer padrões em progressões harmônicas. A semelhança com o presente trabalho reside no objetivo de aplicar um formalismo rigoroso à análise harmônica simbólica. Contudo, a metodologia difere substancialmente: enquanto Vilar emprega gramáticas livres de contexto, que são eficazes na descrição de estruturas hierárquicas e sequenciais bem definidas, a abordagem deste trabalho, baseada na semântica de mundos possíveis, oferece um mecanismo para lidar com a ambiguidade e a dependência de contexto de

forma mais flexível, modelando múltiplas interpretações tonais e suas inter-relações.

Outra linha de pesquisa significativa que emprega lógica modal na análise musical é a conduzida por Satoshi Tojo e colaboradores, incluindo trabalhos como “Modal Logic for Tonal Music” (TOJO, 2016) e investigações sobre o Cálculo de Lambek Rotulado (*Labelled Lambek Calculus* - LLC) para análise musical (BIZZARRI; TOJO, 2024). Estes estudos propõem a anotação de estruturas musicais hierárquicas com lógica modal para elucidar as referências de eventos musicais a regiões harmônicas. Existe uma convergência com o presente trabalho no uso da lógica modal e na conceituação de tonalidades como “mundos” ou “regiões” contextuais. Contudo, as metodologias e focos específicos divergem, pois a abordagem de Tojo emprega sistemas de dedução e prova para construir e validar as representações, enquanto este trabalho foca na verificação de modelos.

Para sintetizar visualmente as relações e, principalmente, as diferenças cruciais entre a abordagem adotada neste trabalho e as demais propostas formais, a Tabela 2 apresenta uma análise comparativa. Também, para aprofundar a distinção entre as metodologias, a Tabela 3 resume as características centrais de cada trabalho em um formato binário. Esta análise evidencia as escolhas de projeto e os focos que posicionam o Tonology de forma única no panorama da pesquisa.

Tabela 2 – Análise Comparativa das Abordagens de Análise Harmônica Formal

Critério	Aragão (2021)	Vilar (2005)	Tojo (2016); Bizzarri & Tojo (2024)	Este Trabalho (Tonality)
Objetivo Principal	Propor um modelo teórico para análise de tonalidade.	Reconhecer padrões harmônicos usando gramáticas.	Anotar estruturas musicais hierárquicas e elucidar suas referências harmônicas.	Implementar e validar um sistema de software funcional para análise harmônica.
Metodologia Central	Semântica de Kripke; Verificação de Modelos (teórica).	Gramáticas Livres de Contexto (GLC).	Lógica Modal; Cálculo de Lambek Rotulado (LLC); Sistemas de Dedução.	Semântica de Kripke; Verificação de Modelos (implementada).
Foco da Análise	Verificação de Satisfação (a progressão satisfaz o modelo?).	Análise Sintática / <i>Parsing</i> (a progressão é gerada pela gramática?).	Dedução / Prova (construção de uma derivação lógica para a estrutura).	Verificação de Satisfação com geração de “testemunha” (caminho da análise para fins didáticos).
Principal Relação ou Diferença	Fundamento Teórico.	Abordagem formal alternativa (sintática vs. semântica).	Uso da Lógica Modal para dedução, em contraste com o uso para verificação.	Materialização e prova de conceito do modelo de Aragón, com foco na explicabilidade.

Fonte: Elaborado pelo autor (2025).

A análise da Tabela 2 evidencia a posição singular do sistema Tonality. Ele parte diretamente do fundamento teórico de Aragón (2021), mas se distingue por ser sua materialização e validação prática. Em contraste com a abordagem sintática de Vilar (2005), que trata a análise como um problema de *parsing*, o Tonality adota uma perspectiva semântica, explorando múltiplos contextos tonais para encontrar a interpretação mais coerente. Por fim, enquanto os trabalhos de Tojo (2016) também empregam lógica modal, seu foco é na dedução e prova de estruturas. O Tonality, por outro lado, usa a verificação de modelos para um fim específico: não apenas classificar, mas gerar explicações passo a passo, focando no seu potencial didático.

Tabela 3 – Comparativo de Características Binárias entre Abordagens Formais

Característica	Aragão (2021)	Vilar (2005)	Tojo et al.	Este Trabalho
Foco em Implementação Prática	Não	Sim	Não	Sim
Usa Verificação de Modelos	Sim	Não	Não	Sim
Modela Ambiguidade (Múltiplos Mundos)	Sim	Não	Sim	Sim
Gera Explicação Visual/Didática	Não	Não	Não	Sim
Abordagem Semântica (vs. Sintática)	Sim	Não	Sim	Sim

Fonte: Elaborado pelo autor (2025).

A Tabela 3 destaca a combinação única de atributos do Tonality. Enquanto o trabalho de Vilar (2005) também possui um foco prático, ele adota um paradigma sintático distinto e não modela a ambiguidade harmônica. Por outro lado, os trabalhos de Aragão (2021) e Bizzarri e Tojo (2024) são semanticamente ricos e lidam com múltiplos contextos, mas permanecem no campo teórico, sem uma implementação funcional ou um foco na geração de explicações didáticas. Este trabalho, portanto, ocupa um nicho claro ao ser o único a sintetizar uma abordagem semântica baseada em verificação de modelos com uma implementação prática e um objetivo pedagógico explícito.

Em suma, ao implementar e avaliar o modelo de Aragão (2021) baseado em semântica de mundos possíveis e verificação de modelos, este trabalho situa-se em uma área específica da análise harmônica computacional que prioriza o rigor lógico, a interpretabilidade e o potencial didático. Distingue-se de abordagens majoritariamente focadas em áudio e/ou aprendizado de máquina pela sua natureza simbólica e explicabilidade formal. A principal contribuição é a materialização do sistema Tonality, que serve como uma prova de conceito e uma ferramenta para explorar a eficácia da abordagem de Aragão (2021). Adicionalmente, este trabalho contribui para um nicho da análise musical computacional – o da análise formalmente justificada e pedagogicamente orientada. Ao focar na explicabilidade, o Tonality visa ser mais do que um classificador, aspirando a ser um “companheiro de análise harmônica”.

4 METODOLOGIA

Este capítulo é dedicado à exposição dos procedimentos metodológicos que nortearão a concepção, o desenvolvimento e a avaliação do sistema Tontology. O objetivo é construir uma implementação computacional para a análise de tonalidade em progressões harmônicas, cuja arquitetura lógica e conceitual será integralmente fundamentada na semântica de mundos possíveis, uma abordagem teórica proposta por Aragão (2021). A construção de um sistema que se propõe a explicar e expor o raciocínio harmônico exige um rigor metodológico que se estende desde a formalização dos conceitos até a validação do sistema.

Figura 2 – Diagrama de passos metodológicos



Desta forma, a metodologia será segmentada em cinco pilares principais, conforme ilustrado na Figura 2. Primeiramente, a **Modelagem Formal** (Seção 4.1) estabelecerá a tradução dos fenômenos harmônicos para uma linguagem lógica. Em seguida, a **Implementação Computacional** (Seção 4.2) detalhará o plano para a construção do software, incluindo as tecnologias e a arquitetura planejada. A seção de **Base de Dados e Estratégia de Testes** (Seção 4.3) elucidará as fontes de dados musicais e as estratégias para verificação e validação do sistema. O **Fluxo Operacional do Sistema** (Seção 4.4) descreverá o funcionamento projetado da ferramenta. Por fim, a seção sobre **Avaliação e Limitações** (Seção 4.5) delineará os critérios para aferir a eficácia do sistema e reconhecer os contornos da pesquisa.

4.1 Modelagem Formal

A primeira etapa desta pesquisa consistirá na modelagem formal dos conceitos da harmonia tonal. Esta fase é crucial, pois traduzirá a complexidade e a ambiguidade da música para uma estrutura matemática rigorosa e computacionalmente tratável. A base para esta etapa será o trabalho de Aragão (2021), que adapta a semântica de mundos possíveis ao domínio musical.

O processo de modelagem formal seguirá os seguintes passos:

1. **Definição da Sintaxe e Semântica Preliminar:** Será definida uma linguagem formal onde os acordes (e.g., “C”, “Gm”) são os símbolos atômicos. As regras de formação de expressões mais complexas (as progressões) serão baseadas na concatenação e em operações de encadeamento (Justaposição, Crase e Elisão), conforme formalizado por Aragão. O foco aqui é estabelecer o vocabulário básico do sistema.
2. **Construção do Modelo Semântico via Estrutura de Kripke:** O núcleo da semântica será uma Estrutura de Kripke, definida pela tupla $K = \langle S, S_0, S_F, R, \mathbb{L} \rangle$. Neste passo, será definido como cada componente da tupla mapeará um conceito musical:
 - Estados (S): Representarão as funções tonais abstratas (Tônica, Dominante, Subdominante).
 - Relação de Acessibilidade (R): Definirá a “gramática” das transições funcionais permitidas (e.g., Dominante pode levar a Tônica).
 - Funções de Rótulo (L): Cada função de rótulo L_i representará um “mundo possível”, ou seja, uma tonalidade musical específica (e.g., Dó Maior). Ela associará os estados funcionais abstratos aos acordes concretos que podem exercer aquela função naquela tonalidade.
3. **Formalização da Relação de Satisfação Tonal:** Será estabelecida a relação de satisfação $(K, L \models_{\pi} \phi)$, que define as condições para uma progressão (ϕ) ser considerada tonalmente válida. Esta relação será implementada de forma recursiva, conforme a Definição 5 de Aragão (2021), permitindo que a análise decomponha a progressão, verifique a função de cada acorde e, crucialmente, explore mudanças de tonalidade (modulações) ao permitir a transição entre diferentes funções de rótulo ($L \rightarrow L'$). A análise será realizada sobre a progressão invertida para ancorar o processo no ponto de maior estabilidade tonal, a resolução final.

4.2 Implementação Computacional

A segunda fase da metodologia será a tradução do modelo formal, definido na etapa anterior, em um sistema de software funcional. Esta implementação visará criar uma ferramenta robusta, modular e fiel ao formalismo lógico.

O plano de implementação é o seguinte:

1. **Seleção de Ferramentas e Tecnologias:** Para a construção do sistema, serão utilizadas as seguintes tecnologias:
 - Linguagem *Python*: Escolhida por sua sintaxe clara, vasto ecossistema de bibliotecas e adequação para prototipagem e desenvolvimento de sistemas complexos.
 - Framework *FastAPI*: Será empregado para construir a *API* web do sistema, devido à sua alta performance, facilidade de uso e geração automática de documentação.
 - Biblioteca *Pydantic*: Será utilizada em conjunto com o *FastAPI* para validação de dados e modelagem das estruturas internas (acordes, tonalidades), garantindo a robustez e a integridade dos dados que fluem pelo sistema.
2. **Projeto da Arquitetura do Software:** A arquitetura do Tonology será projetada de forma modular, seguindo uma abordagem em camadas (Apresentação/*API*, Serviço e Domínio) para desacoplar as responsabilidades. O projeto resultará em um conjunto de classes principais, cada uma com um papel específico na tradução do formalismo para o código, como o *SatisfactionEvaluator* (que implementará a lógica de satisfação recursiva) e a *TonalKnowledgeBase* (que gerenciará as definições das tonalidades).
3. **Desenvolvimento dos Componentes Centrais:** Será desenvolvido o núcleo do sistema, com foco no algoritmo que implementa a relação de satisfação. Este motor de análise recursivo será projetado para explorar a árvore de possibilidades de análise, testando diferentes caminhos funcionais e diferentes tonalidades para justificar uma progressão.
4. **Implementação do Módulo de Visualização:** Para atender ao objetivo pedagógico, será desenvolvido um módulo no backend responsável por gerar uma representação visual da análise. Este módulo receberá o resultado da análise lógica e utilizará a ferramenta *Graphviz* para criar um diagrama de fluxo que ilustra, passo a passo, o raciocínio do sistema. A imagem do diagrama será parte da resposta da *API*.

4.3 Base de Dados e Testes

Para validar a implementação e avaliar a eficácia do Tontology, será adotada uma estratégia multifacetada de testes, utilizando um corpus diversificado de progressões harmônicas.

1. **Construção do Corpus de Análise:** O conjunto de dados para teste será composto por:
 - Obras da literatura musical: Progressões extraídas de peças canônicas (corais de Bach, standards de jazz) para testar a aplicabilidade em cenários reais.
 - Progressões sintéticas: Sequências criadas para testar casos específicos, como ambiguidades, modulações e violações de regras tonais.
 - Corpora digitais existentes: Será explorada a utilização de bases de dados públicas, como o *Chordonomicon* ou o *RockCorpus*, para testes em larga escala. Serão utilizadas ferramentas como a biblioteca *music21* para o processamento e extração programática dos acordes.
2. **Verificação e Validação do Sistema:** O sistema será submetido aos seguintes tipos de testes:
 - Testes Unitários: O núcleo lógico, especialmente a classe *SatisfactionEvaluator*, será testado rigorosamente com casos de teste baseados nos exemplos formais do artigo de Aragão, para garantir a correção da implementação em relação à teoria.
 - Testes de Integração: Serão simuladas requisições *HTTP* à *API* para validar o fluxo completo do sistema, desde a recepção da requisição até o retorno da resposta formatada.
 - Testes de Aceitação: Os resultados para um corpus extenso serão comparados com análises de referência para avaliar a acurácia e a validade musical do Tontology em cenários práticos.

4.4 Fluxo Operacional do Sistema (Projetado)

O funcionamento do sistema, desde a entrada do usuário até a saída da análise, seguirá um fluxo operacional bem definido. O processo se iniciará com a submissão de uma progressão harmônica pelo cliente via uma requisição *HTTP*. O backend, construído com *FastAPI*, receberá e validará essa requisição. A lógica de negócio, orquestrada por uma camada de serviço, invocará o *ProgressionAnalyzer*, que, por sua vez, utilizará o *SatisfactionEvaluator* para realizar a análise recursiva. A base de conhecimento, gerenciada pela *TonalKnowledgeBase*, fornecerá as definições de tonalidades e da estrutura de Kripke necessárias para a análise. Por fim, o resultado, contendo a indicação de tonalidade e a explicação detalhada, será formatado em *JSON* e retornado ao cliente.

4.5 Avaliação e Limitações da Abordagem

A etapa final da metodologia consistirá na avaliação da ferramenta e no reconhecimento de suas limitações.

1. **Estratégia de Avaliação:** A avaliação terá um foco qualitativo. Os resultados gerados pelo *Tonology* serão comparados com análises de especialistas para verificar a correção lógica e a coerência musical. Adicionalmente, planeja-se coletar feedback de usuários (estudantes ou professores de música) para aferir a clareza das explicações e o potencial pedagógico da ferramenta.
2. **Reconhecimento das Limitações:** Será realizada uma análise crítica das fronteiras do trabalho. As principais limitações a serem discutidas incluem: a dependência do formalismo de Aragão, o escopo da base de conhecimento tonal, a natureza puramente simbólica da análise (sem processamento de áudio) e a ausência, nesta fase, de uma interface de usuário dedicada.

Finalmente, espera-se ter construído não apenas um sistema, mas um argumento convincente de que a abordagem de Aragão (2021), quando cuidadosamente implementada, oferece uma ferramenta poderosa e didática para desvendar aspectos da harmonia tonal. A avaliação criteriosa e o reconhecimento honesto das limitações permitirão, então, analisar os resultados com discernimento e vislumbrar os próximos passos com organização.

4.6 Cronograma

Para organizar a execução dos múltiplos passos metodológicos e considerando o calendário acadêmico do semestre letivo, foi elaborado um cronograma de atividades.

Tabela 4 – Cronograma de Atividades do TCC II

Fase / Atividade	Ago	Set	Out	Nov	Dez	Jan
1. Implementação e Refinamento do Sistema						
1.1. Refinamento do motor de análise	X	X				
1.2. Finalização da API e testes unitários		X	X			
1.3. Design da Interface (Frontend) e fluxo de interação	X					
1.4. Desenvolvimento da estrutura do Frontend e consumo da API		X	X			
1.5. Implementação da geração de diagramas no backend			X	X		
2. Testes e Validação do Sistema						
2.1. Coleta e preparação do corpus de análise	X	X				
2.2. Execução dos testes e análise de casos de borda				X		
2.3. Testes de integração Backend-Frontend e refinamento da UI				X	X	
3. Análise de Resultados e Escrita da Monografia						
3.1. Execução do sistema sobre o corpus e análise dos resultados					X	
3.2. Escrita do Capítulo de Resultados					X	
3.3. Escrita do Capítulo de Conclusões e Trabalhos Futuros					X	X
3.4. Revisão e consolidação completa do texto da monografia						X
4. Finalização e Defesa						
4.1. Revisão da versão preliminar pelo Orientador						X
4.2. Ajustes finais pós-feedback e formatação						X
4.3. Preparação do material de apresentação						X
4.4. Submissão da versão final e Defesa						X

Este planejamento distribui as tarefas ao longo de cinco meses, de agosto a janeiro. As atividades de implementação e refinamento do sistema concentram-se nos meses iniciais, permitindo que as fases de testes, validação e análise de resultados ocorram de forma sequencial e sem sobreposições críticas. Os meses finais foram reservados para a consolidação da escrita da monografia e para os preparativos da defesa, assegurando um período adequado para revisão e ajustes de qualidade.

5 PROPOSTA PRELIMINAR DO SISTEMA

Conforme delineado no capítulo anterior, a metodologia estabelece o plano de ação para a concepção e desenvolvimento do sistema Tonality. Este capítulo, por sua vez, dedica-se a materializar esse plano, apresentando o design técnico detalhado e os fundamentos formais que servem de alicerce para a implementação. O objetivo aqui é traduzir os conceitos teóricos em uma arquitetura de software concreta e em algoritmos específicos, servindo como o projeto que guiará a construção da ferramenta.

5.1 Modelagem Formal Detalhada

A capacidade do sistema Tonality de realizar análises harmônicas explicáveis reside em sua sólida modelagem formal, que aplica os princípios da semântica de mundos possíveis ao domínio da música tonal, com base no trabalho de Aragão (2021).

5.1.1 A Estrutura de Kripke como Modelo Semântico

Conforme adaptada por Aragão (2021) para a música tonal, uma Estrutura de Kripke K é formalmente definida como uma tupla de cinco componentes $K = \langle S, S_0, S_F, R, \mathbb{L} \rangle$. A Tabela 5 detalha cada componente e sua correspondência musical no sistema Tonality.

A Figura 3 ilustra visualmente o conceito de “mundos possíveis”. O mecanismo de múltiplas funções de rótulo (a coleção \mathbb{L}) é o que permite a modelagem de diferentes tonalidades e, por extensão, modulações. A figura demonstra como a mesma estrutura funcional subjacente (os mesmos estados s_t, s_d, s_{sd} e as mesmas transições R) pode gerar diferentes conjuntos de acordes permitidos, dependendo da “lente” tonal (a função de rótulo L_i) aplicada.

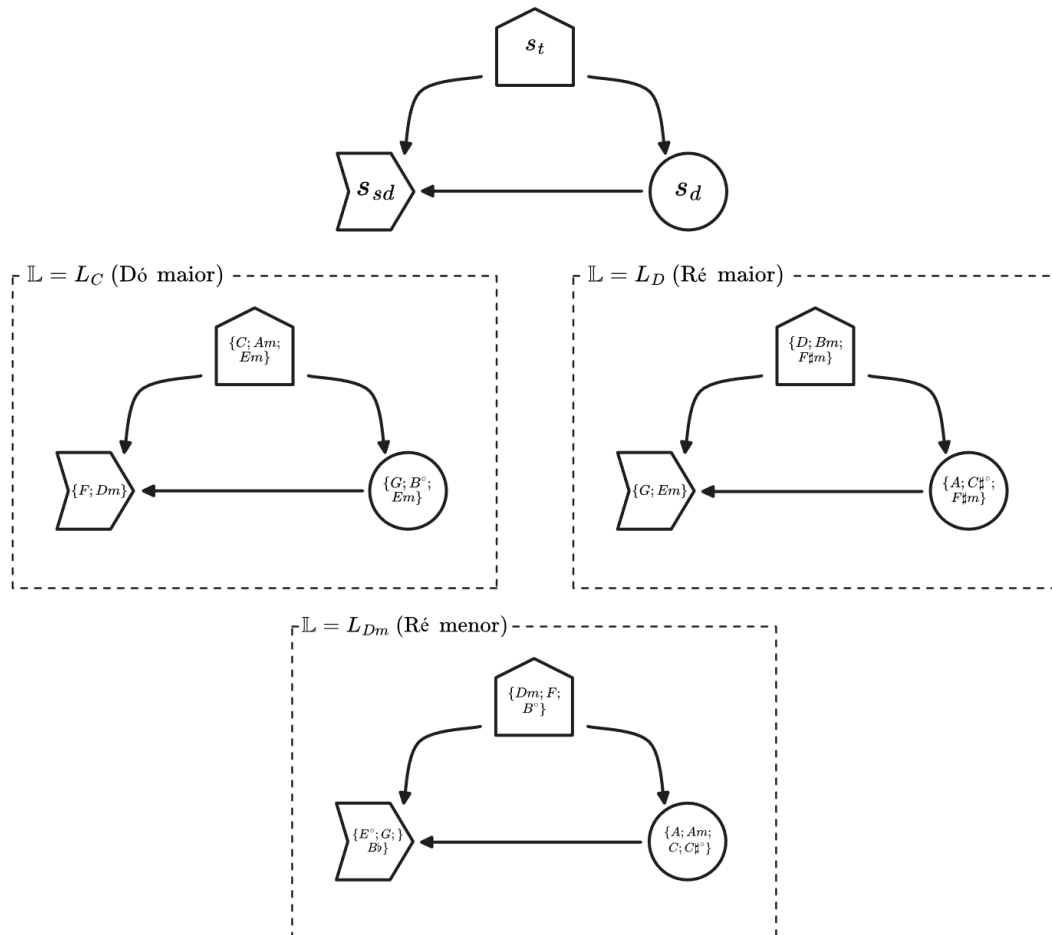
Tabela 5 – Componentes da Estrutura de Kripke para Análise Tonal (Definição 3 de Aragão (2021))

Componente	Descrição Formal Resumida	Interpretação Musical no Tonality
S	Um conjunto finito e não vazio de estados abstratos.	Cada estado $s \in S$ corresponde a uma função tonal fundamental (e.g., Tônica s_t , Dominante s_d , Subdominante s_{sd}). Estes são os “papéis” que os acordes podem desempenhar.
$S_0 \subseteq S$	Um subconjunto não vazio de S que representa os estados iniciais da progressão.	Ponto de partida para a análise da progressão (quando lida ao contrário). Aqui, $S_0 = \{s_t\}$, indicando que progressões irão acabar resolvendo na tônica, quando invertidas, irão começar com a mesma.
$S_F \subset S$	Um subconjunto próprio de S que representa os estados finais ou de aceitação.	O estado funcional de resolução final. Aqui, $S_F = \{s_{sd}, s_d\}$. A progressão (quando tonal) invertida vai terminar com uma desses estados.
$R \subseteq S \times S$	Uma relação de acessibilidade binária total sobre S especificando as transições permitidas entre os estados.	Define a “gramática” das progressões funcionais válidas (e.g., $\langle s_d, s_t \rangle \in R$ permite Dominante \rightarrow Tônica). Ser total garante que a análise sempre pode prosseguir.
\mathbb{L}	Uma coleção de funções de rótulo $L_i : S \rightarrow \mathcal{P}(\Pi)$, onde Π é o conjunto de todos os acordes.	Cada função $L_i \in \mathbb{L}$ representa um contexto tonal específico (uma tonalidade musical). $L_i(s)$ retorna o conjunto de acordes concretos que podem realizar a função s na tonalidade L_i .

Fonte: Adaptado de Aragão (2021)

Descrição: Uma tabela com colunas para “Componente”, “Descrição Formal Resumida” e “Interpretação Musical no Tonality”.

Figura 3 – Estrutura de Kripke com Múltiplas Funções de Rótulo



Fonte: Elaborado pelo autor, com base em Aragão (2021).

Um exemplo concreto de uma estrutura de Kripke $K = \langle S, S_0, S_F, R, \mathbb{L} \rangle$ para análise tonal é fornecido por Aragão (2021):

- $S = \{s_{sd}, s_d, s_t\}$ (representando Subdominante, Dominante e Tônica).
- $S_0 = \{s_t\}$.
- $S_F = \{s_{sd}, s_d\}$.
- R contém: $\langle s_t, s_{sd} \rangle, \langle s_t, s_d \rangle, \langle s_d, s_{sd} \rangle$.
- $\mathbb{L} = \{L_1, L_2, L_3, \dots\}$, onde, por exemplo:
 - $L_1(s_t) = \{C\}, L_1(s_{sd}) = \{F\}, L_1(s_d) = \{G\}$.
 - $L_5(s_t) = \{F, Dm\}, L_5(s_{sd}) = \{Bb, Gm\}, L_5(s_d) = \{C^7, E^\circ\}$.

Para efetivamente navegar por esta estrutura de Kripke e analisar sequências de acordes, o conceito de caminho (π) é fundamental. Conforme a Definição 4 de Aragão (2021), um caminho π em uma estrutura de Kripke K , a partir de um estado $s \in S$, é uma sequência finita de estados $\pi = s_0, s_1, \dots, s_n$ tal que o estado inicial do caminho s_0 é igual a s , e para todo índice k no intervalo $0, 1, \dots, n-1$, a transição do estado s_k para o estado s_{k+1} é permitida pela relação de acessibilidade, ou seja, $\langle s_k, s_{k+1} \rangle \in R$. Notações para prefixo ($\pi_{0,k}$) e sufixo ($\pi_{k,n}$) também são definidas. Um caminho representa, portanto, uma sequência de transições funcionais permitidas dentro da estrutura, e é ao longo desses caminhos que as progressões de acordes são avaliadas. Utilizando a estrutura de Kripke definida anteriormente, temos os seguintes caminhos¹:

- $\pi_A = [s_t, s_d, s_{sd}]$ (Tônica \rightarrow Dominante \rightarrow Subdominante)
- $\pi_B = [s_t, s_{sd}]$ (Tônica \rightarrow Subdominante)²
- $\pi_C = [s_t, s_d]$ (Tônica \rightarrow Dominante)³

5.1.2 A Relação de Satisfação Tonal

A Definição 5 (Satisfação) de Aragão (2021) estabelece: Seja $K = \langle S, S_0, S_F, R, \mathbb{L} \rangle$ uma estrutura de Kripke, sejam π e π' caminhos válidos em K , sejam L e L' funções de rótulo pertencentes à coleção \mathbb{L} , seja P um símbolo de acorde individual e seja ϕ uma sequência de acordes. Diz-se que K satisfaz ϕ sob L ao longo de π quando uma das seguintes condições é atendida:

¹ Deve ser notado que, como indicado anteriormente na tabela 5, a inversão da leitura das progressões resulta na inversão dos caminhos também.

² Conhecida como “Cadência Plagal”

³ Conhecida como “Cadência Perfeita”

- Satisfação de um único acorde (Equação 3 de Aragão (2021)):

$$K, L \models_{\pi} P \iff P \in L(\pi_0)$$

Esta regra base estabelece que um único acorde P é satisfeito no início do caminho π (ou seja, no estado π_0 , que é o primeiro estado do caminho π) sob o contexto da tonalidade L se, e somente se, o acorde P pertence ao conjunto de acordes que a função de rótulo L atribui ao estado funcional π_0 . Em termos simples, o acorde P deve ser um dos acordes que podem realizar a função tonal do estado π_0 na tonalidade L .

- A análise de uma sequência $P\phi$ onde P é o primeiro acorde e ϕ o restante) é mais complexa e permite a exploração de alternativas. A Equação 4 de Aragão é:

$$K, L \models_{\pi} (P\phi) \iff K, L \models_{\pi} P \text{ e } (K, L \models_{\pi_1, n} \phi \text{ ou } K, L \models_{\pi'} \phi)$$

Isso significa que $P\phi$ é satisfeita se P satisfaz o estado atual π_0 em L , **E** o restante ϕ é satisfeito continuando no mesmo caminho π (a partir de π_1) na mesma tonalidade L , **OU** ϕ é satisfeito em um novo caminho π' (ainda na mesma tonalidade L). Esta última disjunção permite explorar alternativas funcionais dentro da mesma tonalidade.

- A Equação 5 de Aragão introduz a mudança de contexto tonal:

$$K, L \models_{\pi} (P\phi) \iff K, L \models_{\pi_0} P \text{ e } K, L' \models_{\pi'_0} P \text{ e } K, L' \models_{\pi'_1, n} \phi$$

Esta formulação original parece descrever um acorde pivô específico que pertence a L e L' . No entanto, a implementação computacional no Tonology adota uma abordagem disjuntiva mais geral e exploratória que captura de forma mais ampla a natureza da análise harmônica e da modulação. A lógica recursiva implementada pode ser entendida assim:

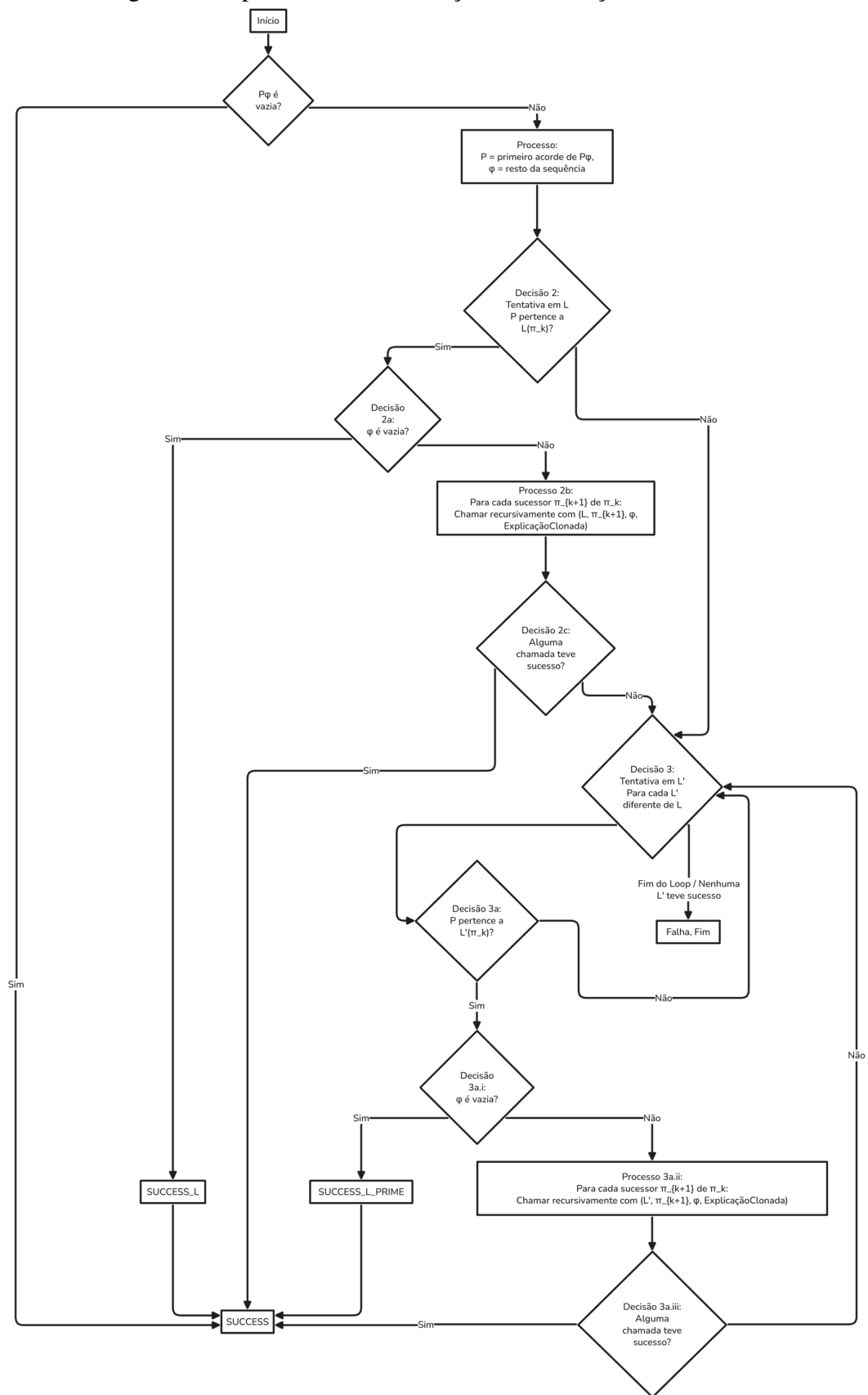
- Uma sequência $P\phi$ é satisfeita em K sob L no estado funcional atual π_k se:
 1. (P satisfaz π_k em L) **E** (ϕ é satisfeito recursivamente a partir de um sucessor de π_k em L), **OU**
 2. (P satisfaz π_k em uma tonalidade alternativa L') **E** (ϕ é satisfeito recursivamente a partir de um sucessor de π_k em L').

A análise harmônica humana é inerentemente exploratória; um músico considera múltiplas interpretações para uma passagem ambígua. A Definição 5, especialmente com a interpretação disjuntiva implementada, modela essa exploração. A recursão permite decompor a progressão em P (o acorde atual) e ϕ (o restante da progressão, dos acordes). As disjunções (“OU”) permitem que o sistema tente diferentes estratégias: continuar na mesma tonalidade, tentar um caminho alternativo na mesma tonalidade, ou tentar uma tonalidade diferente (L'). Essa capacidade de “tentar de novo” com L' é o que formalmente permite ao Tontology identificar modulações. O Tontology não segue um único caminho determinístico; ele explora uma árvore de possibilidades de análise. Se um ramo dessa árvore leva a uma satisfação completa da progressão, então essa é uma explicação válida. Isso permite ao sistema encontrar múltiplas análises válidas para progressões ambíguas, alinhando-se com o objetivo de Aragão (2021) de encontrar múltiplas explicações.

Uma convenção metodológica importante, estabelecida em Aragão (2021) e adotada integralmente pelo Tontology, é que a sequência de acordes de entrada é analisada do seu final para o seu começo. Nesta abordagem, π_0 corresponde ao último acorde da harmonia original. O ponto de maior estabilidade e definição tonal em uma progressão é tipicamente seu final, a resolução na tônica. O estado inicial da avaliação na estrutura de Kripke (π_0 para o primeiro acorde da sequência invertida) pode ser fixado como o estado funcional Tônica (s_t). Isso simplifica a condição inicial da recursão e guia a busca por progressões que convergem para essa tônica assumida, refletindo como os músicos muitas vezes entendem a harmonia – o final de uma frase lança luz sobre o significado do que veio antes e reduzindo o espaço de busca inicial.

A Figura 4 traduz a lógica recursiva e disjuntiva da Definição 5 em um formato visual passo a passo, auxiliando na compreensão didática de como o algoritmo explora as possibilidades de análise, destacando os pontos de decisão chave: satisfação do acorde atual, continuação na mesma tonalidade, e a tentativa de mudança de tonalidade.

Figura 4 – Fluxograma Simplificado da Verificação de Satisfação Recursiva



Fonte: Elaborado pelo autor.

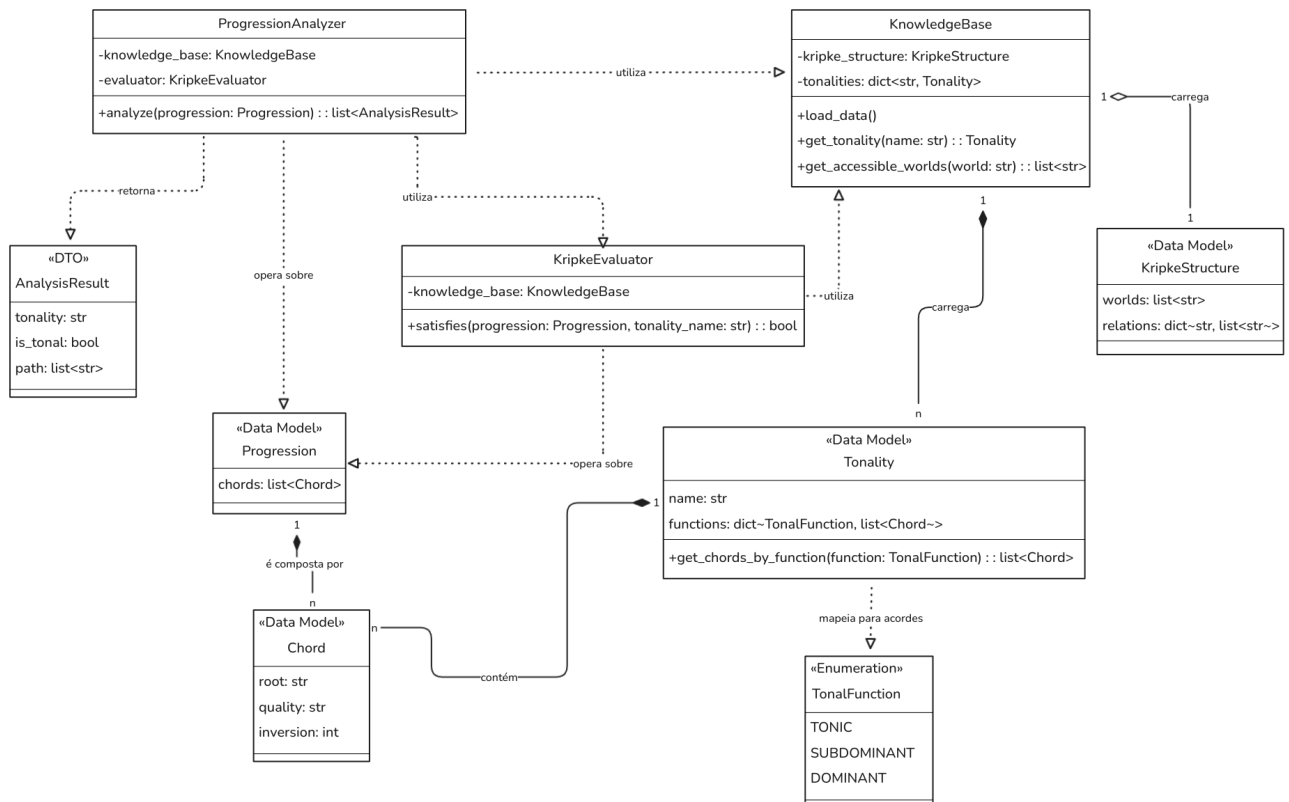
5.2 Arquitetura e Implementação Projetada

A materialização do modelo formal em software é realizada através de uma arquitetura modular e de um conjunto de componentes bem definidos, projetados para garantir clareza, manutenibilidade e fidelidade ao formalismo.

5.2.1 Arquitetura Geral do Sistema

O Tonality é projetado sobre uma arquitetura em camadas que separa a interface de comunicação (API), a lógica de orquestração (Serviço) e o núcleo de análise (Domínio). A Figura 5 apresenta o diagrama de classes UML simplificado, ilustrando os principais componentes de software e suas inter-relações.

Figura 5 – Diagrama de Classes UML do Sistema Tonality



Fonte: Elaborado pelo autor.

5.2.2 Componentes Centrais

O comportamento do sistema é definido por algoritmos implementados em seus componentes centrais. A Tabela 6 descreve a responsabilidade de cada classe principal.

Tabela 6 – Principais Classes de Domínio do Sistema Tonality e sua Relação com o Formalismo de Aragão (2021)

Nome da Classe	Responsabilidade Principal	Relação com o Formalismo de Aragão
Chord	Representar um acorde musical individual.	Proposição atômica P na linguagem formal.
TonalFunction	Definir as categorias funcionais tonais abstratas.	Categorias funcionais dos estados s_t, s_d, s_{sd} .
KripkeState	Modelar um estado $s \in S$ da estrutura de Kripke.	Estado s da Definição 3.
Tonality	Representar uma tonalidade musical (função de rótulo L_i).	Função de rótulo $L_i \in \mathbb{L}$ da Definição 3.
KripkeStructureConfig	Definir a parte estática $\langle S, S_0, S_F, R \rangle$ da estrutura de Kripke.	Componentes S, S_0, S_F, R da Definição 3.
Explanation	Coletar e estruturar os passos da derivação lógica da análise.	Estrutura para registrar a derivação da satisfação (Definição 5).
DetailedExplanationStep	Registrar detalhes de um passo específico da avaliação da satisfação.	Representa uma aplicação de uma regra da Definição 5.
ProgressionAnalyzer	Orquestrar o processo de análise de uma progressão harmônica.	Gerencia a aplicação da Definição 5 para a progressão e tonalidades.
SatisfactionEvaluator	Implementar a lógica recursiva da relação de satisfação (Definição 5).	Implementação direta da Definição 5 (Equações 3, 4, 5).
TonalKnowledgeBase	Carregar e fornecer acesso à KripkeStructureConfig e às definições de Tonality's (tonalidades).	Gerencia a instanciação dos componentes do formalismo a partir de fontes externas (e.g., JSON), permitindo a configuração de \mathbb{L} e da estrutura $\langle S, R \rangle$.

Fonte: Elaborado pelo autor.

O ProgressionAnalyzer atua como o orquestrador de alto nível, invertendo a progressão de entrada e invocando o motor de análise para cada tonalidade candidata. O coração do sistema é o SatisfactionEvaluator, que implementa a lógica recursiva da relação de satisfação. O plano é que seu algoritmo explore os diferentes ramos da árvore de análise, testando a satisfação na tonalidade atual (L) e em tonalidades alternativas (L').

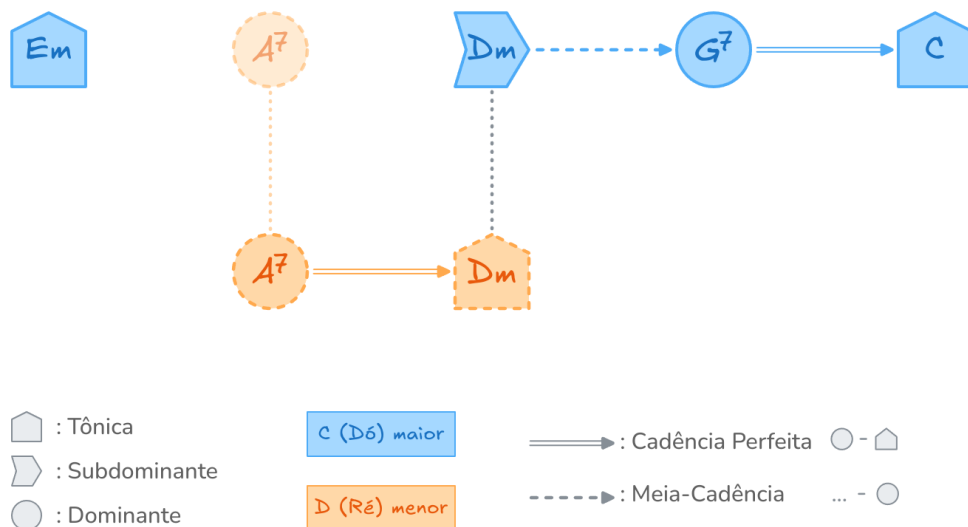
5.3 Visualização da Análise e Fluxo de Dados Projetado

Para cumprir o objetivo pedagógico do Tontology, o sistema é projetado não apenas para classificar uma progressão, mas também para fornecer uma explicação visual de seu raciocínio.

5.3.1 Módulo de Visualização da Análise

O Tontology irá incorporar um módulo de visualização que traduz a derivação lógica em um diagrama de fluxo intuitivo. Após uma análise bem-sucedida, o backend constrói uma representação do grafo de análise na linguagem DOT e utiliza a ferramenta Graphviz para renderizá-lo em uma imagem (e.g., SVG). A Figura 6 mostra um exemplo do diagrama de saída projetado para uma progressão com modulação.

Figura 6 – Exemplo de Diagrama de Saída para a Análise da Progressão [Em - A⁷ - D^m - G⁷ - C]

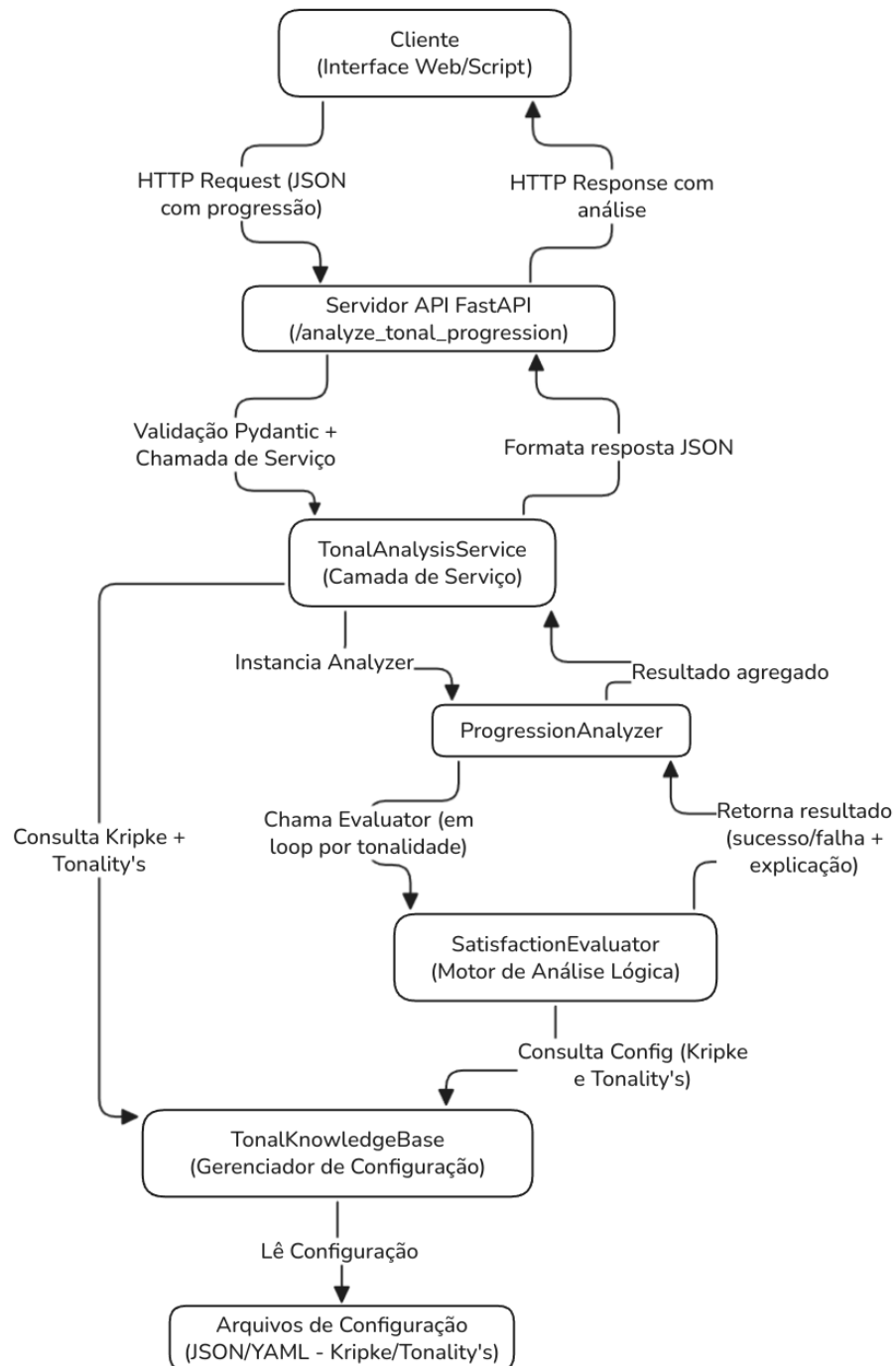


Fonte: Elaborado pelo autor.

5.3.2 Fluxo Operacional do Sistema

O processo completo, desde a requisição do usuário até a resposta da API, segue um fluxo de dados bem definido através dos componentes da arquitetura. A Figura 7 ilustra esta jornada, destacando a interação entre o cliente, o servidor API, a camada de serviço e os componentes do núcleo de análise e da base de conhecimento.

Figura 7 – Diagrama de Fluxo de Dados e Componentes do Sistema Tonality



Fonte: Elaborado pelo autor.

Foi, então, detalhada a proposta técnica do sistema Tonality, estabelecendo uma ponte clara entre o formalismo teórico de Aragão (2021) e um design de software concreto e implementável. A arquitetura e os algoritmos aqui apresentados formam o alicerce para o desenvolvimento e a subsequente avaliação da ferramenta, cujos resultados completos serão discutidos futuramente.

REFERÊNCIAS

- ALLEN, J. What is pydantic? validating data in python. **Prefect Blog**, February 2024. Disponível em: <<https://www.prefect.io/blog/what-is-pydantic-validating-data-in-python>>.
- ARAGÃO, F. E. F. d. Tonal progressions identification through kripke semantics. **MusiMat: Revista Brasileira de Matemática e Música**, Universidade Federal do Rio de Janeiro, v. 5, n. 1, p. 80–88, 2021. Orcid: 0000-0002-2237-4877. Disponível em: <<https://doi.org/10.46926/musmat.2021v5n1.80-88>>.
- BIZZARRI, M.; TOJO, S. Foundations of llcm: Labelled lambek calculus for music analysis. In: ENGLENESE, M.; JOHNSON, C.; ROMERO, J.; MARTINS, T. (Ed.). **Artificial Intelligence in Music, Sound, Art and Design: 17th European Conference, EvoMUSART 2024, Held as Part of EvoStar 2024, Aberystwyth, UK, April 3–5, 2024, Proceedings**. [S.l.]: Springer Nature Switzerland, 2024. (Lecture Notes in Computer Science, v. 14565), p. 17–32. ISBN 978-3-031-56806-2.
- BLACKBURN, P.; RIJKE, M. de; VENEMA, Y. **Modal Logic**. Cambridge: Cambridge University Press, 2001.
- CHEW, E. The spiral array: An algorithm for determining key boundaries. **Proceedings of the Second International Conference on Music and Artificial Intelligence**, p. 18–31, 2002.
- CLARKE, E.; GRUMBERG, O.; PELED, D. **Model Checking**. [S.l.: s.n.], 2001. ISBN 978-0-262-03270-4.
- D., N. Building a structured research automation system using pydantic. **Analytics Vidhya**, March 2025. Disponível em: <<https://www.analyticsvidhya.com/blog/2025/03/multi-agent-research-assistant-system-using-pydantic/>>.
- DALESSIO, G. **Functional Tonal Harmony 1**. 2020. Acesso em 03/06/2025. Disponível em: <https://youtu.be/qzzLj1tbVnA?si=rH49wwn5c61w_gxh>.
- KOSTKA, S.; PAYNE, D. **Tonal Harmony with an Introduction to Twentieth-Century Music**. 6th. ed. New York, NY: McGraw-Hill, 2008.
- KRIPKE, S. Semantical considerations on modal logic. **Acta Philosophica Fennica**, v. 16, p. 83–94, 1963.
- LAITZ, S. G. **The Complete Musician: An Integrated Approach to Tonal Theory, Analysis, and Listening**. New York: Oxford University Press, 2011.
- LATHKAR, M. **High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python**. 1st. ed. New York: Apress, 2023.
- LERDAHL, F.; JACKENDOFF, R. **A Generative Theory of Tonal Music**. Cambridge, MA: The MIT Press, 1983.
- PATIL, S. U.; PATIL, S. U. Python: A language for research. **International Journal for Multidisciplinary Research (IJFMR)**, v. 6, n. 3, 2024. Disponível em: <<https://www.ijfmr.com/papers/2024/3/23112.pdf>>.
- PETERS, T. **The Zen of Python**. 2004. PEP 20, Python Enhancement Proposals. Disponível em: <<https://peps.python.org/pep-0020/>>.

Pydantic Team. **Pydantic Documentation**. 2025. Acesso em: 02/06/2025. Disponível em: <<https://docs.pydantic.dev/latest/>>.

Python Software Foundation. **About Python**. 2025. Acesso em: 02/06/2025. Disponível em: <<https://www.python.org/about/>>.

Python Software Foundation. **Python 3.13.3 Documentation**. 2025. Acesso em: 02/06/2025. Disponível em: <<https://docs.python.org/3/>>.

RAMÍREZ, S. **FastAPI Documentation**. 2025. Acesso em: 02/06/2025. Disponível em: <<https://fastapi.tiangolo.com/>>.

RIEMANN, H. **Harmony Simplified: Or, the Theory of the Tonal Functions of Chords**. Leipzig: Breitkopf & Härtel, 1893. Reimpresso por diversos editores posteriormente.

ROADS, C. **The Computer Music Tutorial**. Cambridge, MA: MIT Press, 1996. ISBN 9780262680820.

SCHOENBERG, A. **Theory of Harmony**. Berkeley, CA: University of California Press, 1979.

Sociedade Brasileira de Computação. **Computação Musical**. 2025. Acesso em: 10 maio 2025. Disponível em: <<https://www.sbc.org.br/computacao-musical/>>.

TOJO, S. Modal logic for tonal music. In: ARAMAKI, M.; DERRIEN, O.; KRONLAND-MARTINET, R.; YSTAD, S. (Ed.). **Perception, Representations, Image, Sound, Music: 10th International Conference, CMMR 2013, Marseille, France, October 15-18, 2013 and 5th International Conference, ICMPC 2014, Seoul, Korea, August 4-8, 2014, Revised Selected Papers**. [S.l.]: Springer International Publishing, 2016, (Lecture Notes in Computer Science, v. 9617). p. 116–130. ISBN 978-3-319-41661-8.

VERBINA, E. Which is the best python web framework: Django, flask, or fastapi? **JetBrains PyCharm Blog**, February 2025. Disponível em: <<https://blog.jetbrains.com/pycharm/2025/02/django-flask-fastapi/>>.

VILAR, P. L. **AHO - Analisador Harmônico Otimista**. Dissertação (Mestrado) — Universidade Federal do Ceará, 2005.