

# Uma Exploração das Ontologias Musicais

Paulo Renato M.S. Gomes<sup>1</sup>

Instituto de Computação, UFRJ, Rio de Janeiro, BR

**Abstract.** Neste trabalho serão abordados tópicos de Web Semântica através das ontologias de representação musical, utilizando como referência o *dataset Isophonics Reference Annotations: The Beatles*. A ideia é conectar o conteúdo visto na disciplina através de uma análise dos mais recentes avanços e implementações que buscam conectar tanto metadados, quanto representações das músicas em si e seus elementos como grafos de conhecimento. Para tal, serão utilizadas ferramentas como GraphDB e SPARQL para elaboração de consultas para exploração dos dados. Além disso, ao final do trabalho buscarei sugerir avanços futuros para as ontologias, com ideias que acredito serem interessantes para aumentar seu potencial descritivo.

**Keywords:** OWL · Construção de Ontologias · Música.

## 1 Introdução

Na última década, a análise algorítmica de elementos musicais se tornou parte integral da produção, consumo e ensino de música. Embora grande parte dos esforços sejam destinados ao processamento de sinais sonoros e *machine learning*, também existe muito potencial para representação simbólica desses elementos através de lógicas descritivas.

Como podemos definir ontologicamente o que é uma música? Para o escopo do trabalho, idealizei uma definição que facilita a interpretação por máquina. Uma música é uma combinação padronizada de sons, composta em essência por três elementos básicos: Melodias (sequências de notas), Ritmos (relação temporal entre sons), Harmonias (sons simultâneos). A presença de todos os elementos ao mesmo tempo não é necessária.

Ao longo do trabalho poderemos observar que existem diversas abordagens de triplificação desses componentes, além de iniciativas que buscam conectar os diferentes recursos já existentes.

## 2 Base de dados

O *dataset* que utilizaremos como exemplo foi produzido pela tese de doutorado *Towards Automatic Extraction of Harmony Information from Music Signals*, onde foram geradas por máquina anotações musicais que contemplam os 11 álbuns principais da banda britânica *The Beatles*, lançados entre 1963 e 1970. Não foram incluídas músicas da discografia disponíveis fora desses álbuns.

Essas anotações foram disponibilizadas publicamente de forma triplificada, oferecendo conexões com diversas ontologias preexistentes. Além disso, foi definida na tese uma sintaxe de representação de acordes que serviu de base para a implementação posterior de outras ontologias.

### 3 Ontologias utilizadas

Antes de partirmos para a parte exploratória, é interessante abordar os diferentes *namespaces* utilizados na estruturação dos dados.

#### 3.1 FOAF, DC e OV

O vocabulário **FOAF** foi utilizado para classificar a anotação de cada faixa dos álbuns como um documento (*foaf:document*), e definir quem foi o artista principal (*foaf:maker* e *foaf:name*).

**DC** foi utilizado para definir metadados como a data de criação dos documentos (*dc:created*) e os títulos dos álbuns e faixas aos quais eles se referem (*dc:title*) e o **OV** para documentar a versão das anotações (*ov:version*) e um decimal numérico de confiabilidade do processamento de sinal obtido para cada elemento da faixa analisada (*ov:confidence*).

#### 3.2 Music Ontology

Essa ontologia é utilizada para representar metadados sobre os documentos (que são classificados como arquivos de áudio - *mo:AudioFile*) dentro do domínio musical, como artistas musicais, sinais sonoros, faixas de álbuns (*mo:Track*, *mo:Album*), lançamentos e gravações (*mo:Record*).

#### 3.3 Timeline, Event e Audio Feature Ontologies

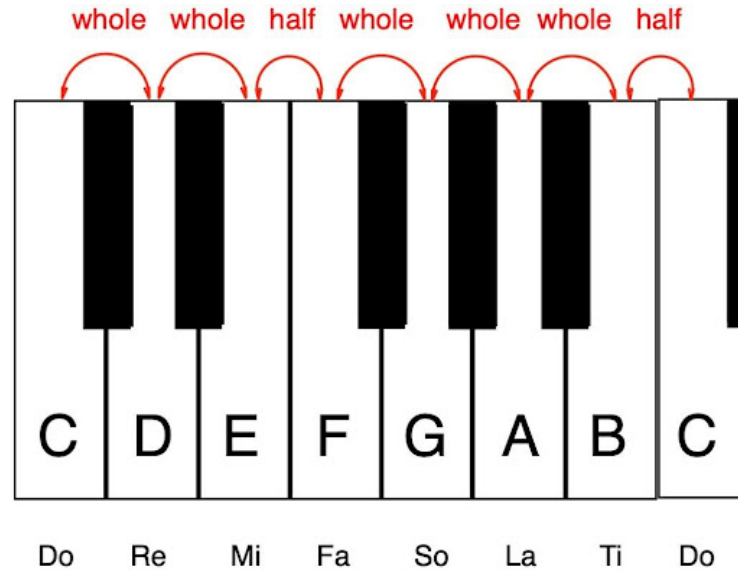
Essas três ontologias são interessantes por serem capazes de descrever e posicionar triplas no tempo (*tl:onTimeLine*). **Timeline** em conjunto com **Event** fornecem recursos para guardar intervalos de tempo (*tl:Interval*, *event:Time*) e o começo e duração de eventos (*tl:beginsAt*, *tl:duration*), os dados temporais podem ser manipulados em consultas.

A **Audio Feature Ontology** foi utilizada na base de dados para classificar os eventos no tempo como sonoros; acordes sendo tocados (*af:ChordSegment*), seções das músicas (*af:StructuralSegment*), mudanças de tom (*af:KeySegment*) e batidas dentro de um compasso (*af:Beat*).

## 4 Harmonia Musical na Web Semântica

O cerne da base de dados está na representação do elemento principal de harmonia musical (acordes) de forma legível por máquina. Foi definida uma sintaxe específica e para melhor compreensão será preciso abordar alguns tópicos básicos de teoria musical.

#### 4.1 Intervalos, Escalas e Tons



**Fig. 1.** Escala musical de dó maior em um piano

**Notas musicais** são frequências sonoras específicas que possuem relações matemáticas entre si, elas são representadas de forma simbólica por solfejo (do - re - mi...) ou por letras do alfabeto, como visto na figura 1.

**Intervalos** são nomes dados às "distâncias" de uma nota musical para outra, os menores intervalos possíveis são o semitom (*half-step*) e o tom (*whole-step* - dois semitons, distância entre duas notas pulando a que está entre elas). **Modificadores** podem ser utilizados para descer (bemol - b) ou subir (sustenido - # ou s) notas musicais e intervalos em 1 semitom.

**Escala musical** são um conjunto de notas que obedece um determinado padrão de intervalos entre si. (Ex.: uma *escala maior* sempre é formada por 2 tons, 1 semitom, 3 tons e 1 semitom, partindo de uma nota escolhida como base ou *tônica*). Os outros intervalos recebem nomes numéricos baseado na quantidade de notas da escala entre eles e a base (ex.: terça, quarta, quinta e etc).

Na língua portuguesa, a palavra **tom** adquire dois sentidos:

1. Representa um intervalo de dois semitons. (*whole-step*, em Inglês).
2. Também é utilizada para se referir à escala (conjunto de notas) que uma determinada obra musical se utiliza. (*key*, em Inglês).

As ontologias relacionadas e o *dataset* se utilizam do termo *key* para se referir ao que popularmente conhecemos como o *tom* de uma música.



## 5 Exploração dos Dados

Os dados extraídos estavam originalmente definidos em arquivos separados (cada faixa em um *.ttl* diferente), e para tratamento foram carregados em um grafo unificado na ferramenta GraphDB. Onde foram efetuadas consultas para exemplificar o potencial descritivo das ontologias presentes no *dataset*.

### Query 1

```
select ?songtitle where {
    ?s dc:title "Rubber Soul".
    ?s mo:track ?track.
    ?track dc:title ?songtitle.
} ORDER BY ?track
```

**Fig. 3.** Retornar de forma ordenada todas as faixas do álbum *Rubber Soul*.

### Query 2.1

```
SELECT ?segment ?chord
WHERE {
    ?_iri dc:title "Something".
    BIND (STRBEFORE(STR(?_iri),"track_") AS ?iri)
    ?segment_iri a af:StructuralSegment.
    ?segment_iri rdfs:label ?segment.
    ?segment_iri event:time / tl:beginsAt ?segmentBeginsAt.
    ?segment_iri event:time / tl:duration ?segmentDuration.
    ?chord_iri a af:ChordSegment.
    ?chord_iri rdfs:label ?chord.
    ?chord_iri event:time / tl:beginsAt ?chordBeginsAt.
    FILTER(?segmentBeginsAt < ?chordBeginsAt && ?chordBeginsAt < (?segmentBeginsAt + ?segmentDuration)).
    FILTER regex(str(?chord_iri), str(?iri)). FILTER regex(str(?segment_iri), str(?iri)).
}
```

**Fig. 4.** Retornar todos os acordes presentes na música *"Something"*, na ordem em que são tocados e com a seção ao qual pertencem.

### Query 2.2

```

SELECT ?key ?chord
WHERE {
  ?_iri dc:title "Something".
  BIND (STRBEFORE(STR(?_iri),"track_") AS ?iri)
  ?key_iri a af:KeySegment.
  ?key_iri rdfs:label ?key.
  ?key_iri event:time / tl:beginsAt ?keyBeginsAt.
  ?key_iri event:time / tl:duration ?keyDuration.
  ?chord_iri a af:ChordSegment.
  ?chord_iri rdfs:label ?chord.
  ?chord_iri event:time / tl:beginsAt ?chordBeginsAt.
  FILTER(?keyBeginsAt < ?chordBeginsAt && ?chordBeginsAt < (?keyBeginsAt + ?keyDuration))
  FILTER regex(str(?chord_iri), str(?iri)). FILTER regex(str(?key_iri), str(?iri)).
}

```

**Fig. 5.** Retornar todos os acordes presentes na música *"Something"*, na ordem em que são tocados e a qual tom eles pertencem.

### Query 3

```

select ?album_title (count(?track_iri) as ?tracks) where {
  ?album_iri a mo:Record.
  ?album_iri dc:title ?album_title.
  ?album_iri mo:track ?track_iri.
} group by ?album_title order by desc(?tracks)

```

**Fig. 6.** Ordenar os CDs presentes no dataset pela quantidade de faixas.

### Query 4

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX af: <http://purl.org/ontology/af/>

SELECT ?chord (COUNT(?chord) as ?chord_count)
WHERE {
  ?chord_iri a af:ChordSegment.
  ?chord_iri rdfs:label ?chord.
} group by ?chord order by desc(?chord_count) limit 10

```

**Fig. 7.** Retornar os 10 acordes mais utilizados nas músicas presentes no *dataset*.

### Query 5

```

PREFIX mo: <http://purl.org/ontology/mo/>
SELECT ?key (COUNT(distinct ?song_iri) as ?key_count)
WHERE {
  ?key_iri a af:KeySegment.
  ?key_iri rdfs:label ?key.
  ?track_iri a mo:Track.
  BIND (STRBEFORE(STR(?track_iri),"track_") AS ?song_iri).
  FILTER regex(str(?key_iri), str(?song_iri)).
} group by ?key order by desc(?key_count) limit 11

```

**Fig. 8.** Retornar os 10 tons mais utilizados nas músicas presentes no *dataset*.

### Query 6

```

SELECT ?album_title (COUNT(distinct ?song_iri) as ?key_count)
WHERE {
  ?album_iri a mo:Record.
  ?album_iri dc:title ?album_title.
  ?album_iri mo:track ?track_iri.
  BIND (STRBEFORE(STR(?track_iri),"track_") AS ?song_iri).
  ?key_iri a af:KeySegment.
  ?key_iri rdfs:label ?key.
  FILTER regex(str(?key), str("minor")).
  FILTER regex(str(?key_iri), str(?song_iri)).
} group by ?album_title order by desc(?key_count)

```

**Fig. 9.** Ordenar os álbuns pela quantidade de faixas que se utilizam de tons menores.

### Query 7

```

SELECT ?album_title (COUNT(distinct ?song_iri) as ?key_count)
WHERE {
  ?album_iri a mo:Record.
  ?album_iri dc:title ?album_title.
  ?album_iri mo:track ?track_iri.

  BIND (STRBEFORE(STR(?track_iri),"track_") AS ?song_iri).

  ?key_iri a af:KeySegment.
  ?key_iri rdfs:label "A".
  FILTER regex(str(?key_iri), str(?song_iri)).

} group by ?album_title order by desc(?key_count)

```

**Fig. 10.** Ordenar os álbuns pela quantidade de faixas que se utilizam do tom lá maior.

### Query 8

```

SELECT ?song_iri (COUNT(?key_iri) as ?key_count)
WHERE {
  ?key_iri a af:KeySegment.
  ?key_iri rdfs:label ?key.
  ?track_iri a mo:Track.
  BIND (STRBEFORE(STR(?track_iri), "track_") AS ?song_iri).
  FILTER regex(str(?key_iri), str(?song_iri)).
  FILTER NOT EXISTS{?key_iri rdfs:label "Silence"}.
} group by ?song_iri order by desc(?key_count) limit 10

```

**Fig. 11.** Retornar em ordem decrescente as 10 faixas que apresentam maior quantidade de mudanças de tom.

Apenas com os resultados dessas consultas de exemplo, conseguimos extrair muitas informações interessantes. Podemos ter acesso aos acordes, seções e tons de qualquer música e encaixá-los temporalmente entre si (*Query 2*).

O **acorde mais utilizado** foi **lá maior** ("A"), que aparece mais de 1500 vezes nas anotações (*Query 4*), e não por acaso o **tom mais utilizado** tem como tônica esse acorde, 35 músicas no total se utilizaram dessa escala (*Query 5*), sendo mais usada no álbum *Help!*, em 6 faixas (*Query 7*).

Dentre todos os álbuns, o que possui maior quantidade músicas que se utilizam de **tons menores** é o *Abbey Road*, com um total de 6 (*Query 6*). Coincidentemente, ele foi o último a ser gravado pela banda.

A música "I'll Be Back" é a que apresenta a maior quantidade de mudanças de tom, com um total de **18**. Aparentemente o que acontece são alterações entre os tons de lá maior e lá menor durante toda a sua duração.

## 6 Ontologias relacionadas

O artigo que deu a temática do trabalho, *Functional harmony ontology: Musical harmony analysis with Description Logics* implementa uma ontologia de análise de harmonia funcional, que é um campo teórico bem mais complexo dentro da música.

Minha intenção inicialmente era implementar um subconjunto da análise presente no artigo, focada em analisar semanticamente sequências de acordes, mas isso não foi possível devido à problemas computacionais, fugindo do escopo da disciplina. No entanto, através da pesquisa descobri uma iniciativa que busca implementar um *workflow* de interconexão dos *datasets* de harmonia musical dentro da Web Semântica chamada *ChoCo*.

Ela fornece uma API e SPARQL *endpoint* para acessar diversos *datasets* de anotação de acordes (incluindo o utilizado neste trabalho) além de códigos em outras linguagens para efetuar conversão entre os diversos tipos de notação.

Dentre as ontologias implementadas por essa iniciativa, duas casaram mais com a proposta inicial do trabalho: *Chord Ontology* (utilizada como exemplo anteriormente) e *Roman Chord Ontology*.



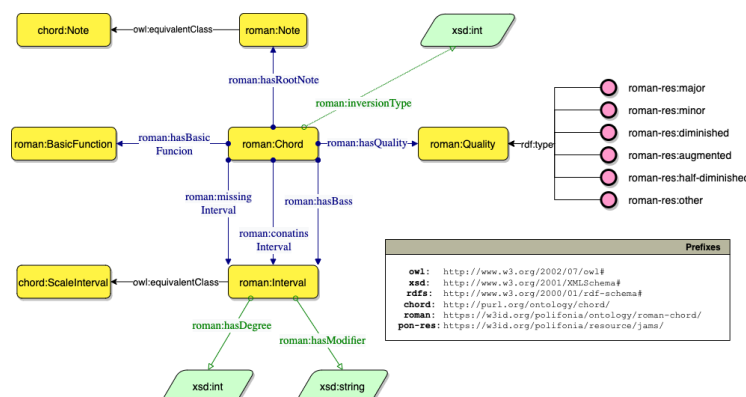


Fig. 12. Diagrama da ontologia Roman Chord

Basicamente, a ontologia descrita do diagrama acima representa acordes através da notação em numeração romana. Essa notação é interessante por ser capaz de representar a mesma progressão de acordes em diversos tons diferentes.

Na música popular, diversas músicas distintas se utilizam das mesmas progressões (sequências), o que é uma informação e conexão semântica muito interessante para estudantes e praticantes de música.

Ex.: C - Am - G - F (em dó maior) e D - Bm - A - G (em ré maior) se tratam da mesma progressão de acordes em numerais romanos (I - vi - V - IV).

## 7 Trabalhos futuros

Com as informações adquiridas no trabalho, seria interessante implementar uma conexão da *Roman Chord Ontology* com os outros *datasets* preexistentes e gerar versões triplicadas das progressões utilizadas nas faixas contidas neles.

Para isso, seria necessária a conversão da notação do *Chord Ontology* para a notação romana, de forma que seja possível automatizá-la para grandes volumes de dados, já existem na API códigos que fazem o caminho inverso.

Como esse é um tópico teórico mais avançado do que somente representar o acorde em si, seria necessária a conexão com outras áreas da computação. Algumas implementações fora da área da Web Semântica se utilizam de *parsing* (utilizado gramáticas livres de contexto) para efetuar a conversão.

Outra problemática encontrada no trabalho foi que, bases de dados geradas por máquina estão sujeitas à níveis variados de confiabilidade, e podem ocorrer anotações de acorde que destoam da forma que um ser humano interpretaria. Levando em conta o fato de que nesse domínio em específico, elementos semelhantes possuem diversas representações possíveis.

Uma outra abordagem seria através da raspagem de dados em sites que já possuem uma grande base de dados aberta (como *HookTheory* e *Ultimate Guitar*), anotada e curada por seres humanos ao longo do tempo, mas ainda presente

apenas na *Web* de documentos. Esses dados mais validados poderiam ser triplicados e então convertidos em progressões de acorde na notação desejada.

## 8 Conclusão

É possível observar que embora já existam diversas iniciativas e ontologias implementadas ao longo dos últimos anos na área musical, ainda existe muito a ser explorado e melhorado. A Web Semântica possui um potencial único de gerar conexões entre o conhecimento musical que foi construído coletivamente na rede.

O *dataset* e as *queries* realizadas podem ser encontrados no repositório: [https://github.com/paulo-renato13/Entrega\\_final\\_Web\\_Semantica](https://github.com/paulo-renato13/Entrega_final_Web_Semantica).

## References

1. S. Kantarelis, E. Dervakos, N. Kotsani et al: Functional harmony ontology: Musical harmony analysis with Description Logics. Journal of Web Semantics (2022)
2. Harte, C. (2010). Towards Automatic Extraction of Harmony Information from Music Signals [Doctoral Thesis, University of London].
3. J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, V. Presutti: ChoCo: a Chord Corpus and a Data Transformation Workflow for Musical Harmony Knowledge Graphs. Nature. (2023)
4. Basic Music Theory for Beginners – The Complete Guide, <https://www.iconcollective.edu/basic-music-theory/>, last accessed 2023/07/09
5. Reference Annotations: The Beatles, <http://isophonics.net/content/reference-annotations-beatles>, last accessed 2023/07/09
6. chordparser \* PyPI, <https://pypi.org/project/chordparser/>, last accessed 2023/07/09
7. HarmTrace: Harmony Analysis and Retrieval of Music, <https://hackage.haskell.org/package/HarmTrace>, last accessed 2023/07/09
8. Ultimate Guitar, <https://www.ultimate-guitar.com/>, last accessed 2023/07/09
9. Hooktheory, <https://www.hooktheory.com/>, last accessed 2023/07/09
10. FOAF Vocabulary Specification, <http://xmlns.com/foaf/spec/>, last accessed 2023/07/09
11. Open.vocab.org, <https://vocab.org/open/>, last accessed 2023/07/09
12. DCMI Metadata Terms, <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>, last accessed 2023/07/09
13. The Music Ontology - Specification, <http://musicontology.com/specification/>, last accessed 2023/07/09
14. Timeline Ontology, <https://motools.sourceforge.net/timeline/timeline.html>, last accessed 2023/07/09
15. Event Ontology, <https://motools.sourceforge.net/event/event.html>, last accessed 2023/07/09
16. Audio Features Ontology Specification, "https://motools.sourceforge.net/doc/audio\_features.html", last accessed 2023/07/09
17. The OMRAS2 Chord Ontology, [https://motools.sourceforge.net/chord\\_draft\\_1/chord.html](https://motools.sourceforge.net/chord_draft_1/chord.html), last accessed 2023/07/09
18. The Roman Chord Ontology, <https://raw.githubusercontent.com/polifonia-project/roman-chord-ontology/main/roman-chord.owl>, last accessed 2023/07/09