

# Ontologia musicale

## 1. Motivazioni

Questa ontologia è sviluppata per strutturare e organizzare il dominio della musica, fornendo una rappresentazione chiara delle relazioni tra generi, artisti, album e brani. L'obiettivo è quello di creare un modello formale che consenta di categorizzare i diversi stili musicali e le loro connessioni, facilitando la comprensione delle influenze e delle evoluzioni nel panorama musicale.

Un aspetto fondamentale di questa ontologia è la suddivisione dei generi in categorie principali e sottogeneri. Ad esempio, il Rock è rappresentato con le sue varianti, come Punk Rock e Rock and Roll, per evidenziare le differenze stilistiche e le affinità storiche tra i vari movimenti musicali. Questa organizzazione permette di analizzare in modo più dettagliato il contesto musicale, supportando studi sulla musica, applicazioni di ricerca e piattaforme di gestione dei contenuti.

L'integrazione di elementi come artisti, album e testi delle canzoni arricchisce ulteriormente l'ontologia, rendendola un sistema utile per classificare e recuperare informazioni in modo più efficace rispetto ai tradizionali database musicali. Grazie a questa struttura, è possibile migliorare la navigazione tra i contenuti, sviluppare strumenti per la raccomandazione musicale e facilitare l'accesso a informazioni dettagliate sulle opere e sui loro autori.

---

## 2. Requirements per la creazione dell'ontologia

### Finalità

L'ontologia musicale mira a fornire una rappresentazione strutturata della musica, descrivendo generi, sottogeneri, artisti, album e canzoni. L'obiettivo è quello di migliorare la categorizzazione e la ricerca delle informazioni musicali attraverso un modello formale basato su relazioni e proprietà.

### Task specifici

- **Consultazione:** Permette di esplorare la musica in base a genere, artista, periodo storico o altre caratteristiche.
- **Reference:** Fornisce una base di conoscenza utile per studi musicologici e analisi storiche.

- **Verifica:** Consente di validare classificazioni musicali e confrontare caratteristiche di generi e sottogeneri.
- **Raccomandazione:** Può essere utilizzata per suggerire brani basati sulle preferenze dell'utente.

### Tipo di utenti

- Studenti e studiosi di musicologia.
  - Curatori di playlist e critici musicali.
  - Sviluppatori di sistemi di raccomandazione musicale.
  - Professionisti dell'industria musicale (case discografiche, servizi di streaming).
- 

## 3. Descrizione del dominio

**Dominio** L'ontologia è stata sviluppata per rappresentare il mondo della musica, includendo i Generi Musicali, gli Artisti, gli Album e i Brani. L'obiettivo è fornire una struttura chiara che permetta di comprendere e organizzare la varietà di stili musicali e le loro connessioni.

Le principali categorie e le loro relazioni possono essere descritte come segue:

- Un **Genere Musicale** può comprendere diversi **Sottogeneri**, ognuno con caratteristiche specifiche.
- Un **Artista** deve avere diversi **Brani** e può avere anche degli album.
- Un **Album** è una raccolta di **Brani**, che possono appartenere a uno o più generi.
- Un **Brano** ha una durata, un testo e può essere eseguito da uno o più artisti ed ha un proprio genere.

**Fonti** Per definire le categorie e le loro caratteristiche, ci si è basati su fonti autorevoli e database specializzati. Alcuni esempi includono:

- *MusicBrainz* (<https://musicbrainz.org/>): database collaborativo di informazioni musicali.
- *Discogs* (<https://www.discogs.com/>): archivio musicale con metadati strutturati.
- *DBpedia Music Ontology* (<http://dbpedia.org/ontology/MusicalWork>): ontologia di riferimento per entità musicali.

Infine, per i testi delle canzoni e la loro analisi, è stato preso in considerazione un database di liriche ufficiali:

- **Genius:** <https://genius.com/>

I dati sugli artisti, sugli album e sulle classifiche possono essere basati su fonti online o su esperienze personali, cercando di costruire un modello che rifletta la varietà del panorama musicale globale.

---

## 4. Competency questions

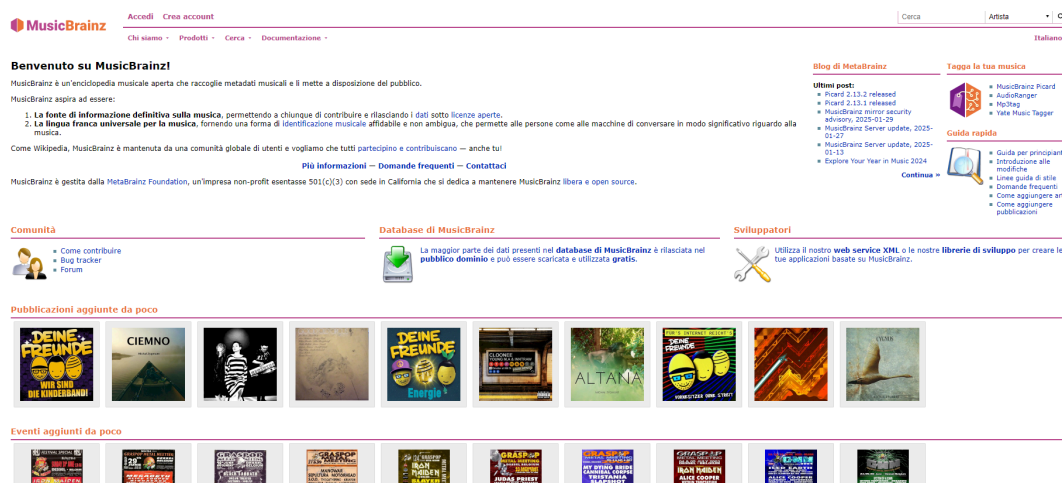
- Chi è l'artista di un determinato brano?
- Quali canzoni appartengono a un album specifico?
- Qual è il genere musicale di un dato brano?
- Quali artisti hanno pubblicato almeno un album?
- Quali canzoni hanno almeno un testo associato?

---

## 5. Documentazione sul dominio

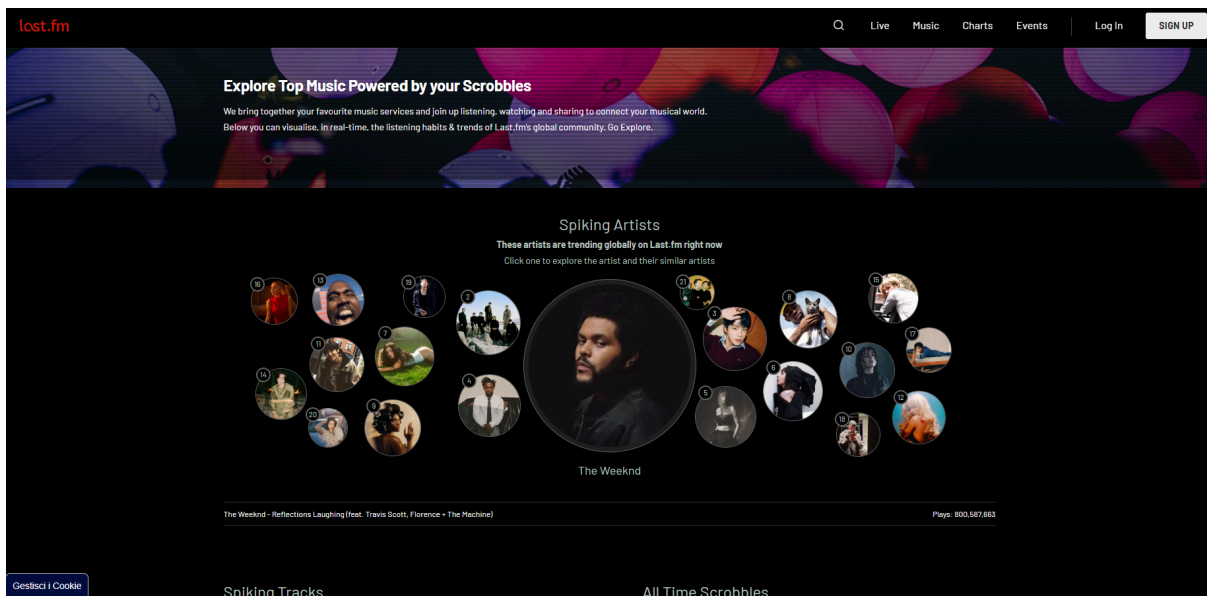
L'ontologia musicale si basa su modelli di classificazione già utilizzati nell'industria musicale. Diverse fonti strutturate forniscono dati dettagliati per comprendere e organizzare il panorama musicale in maniera efficace. Tra i database più utilizzati per catalogare artisti, album e brani ci sono:

- **MusicBrainz** (<https://musicbrainz.org/>): un database open-source che fornisce metadati su artisti, album e canzoni, utile per la normalizzazione delle informazioni musicali.



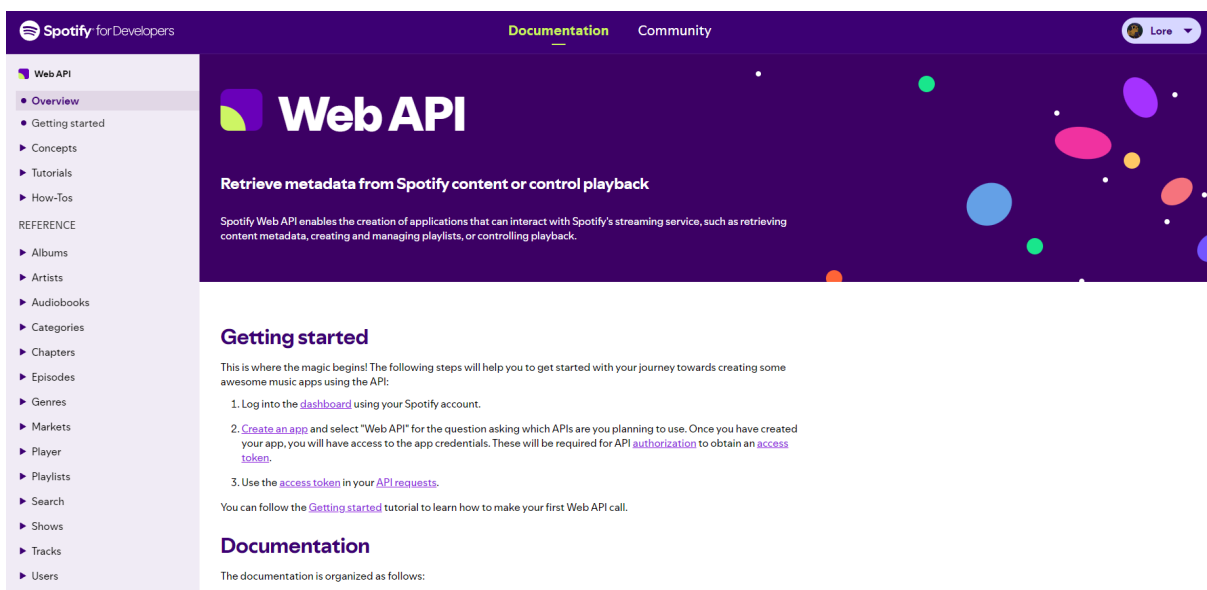
The screenshot shows the MusicBrainz website homepage. At the top, there's a header with the MusicBrainz logo and navigation links like 'Accedi', 'Crea account', 'Cerca', 'Artista', and 'C'. Below the header, the main content area is divided into several sections. On the left, there's a 'Benvenuto su MusicBrainz!' section with a welcome message and a list of recent posts from the MusicBrainz blog. In the center, there's a 'Database di MusicBrainz' section with a description of the database and a link to the 'Comunità' section. On the right, there's a 'Sviluppatori' section with a link to the 'Guida rapida' section. The bottom of the page features a grid of album covers and a list of recent events.

- Last.fm (<https://www.last.fm/>): raccoglie dati sugli ascolti degli utenti, fornendo insight sulle tendenze musicali e sulle correlazioni tra artisti.

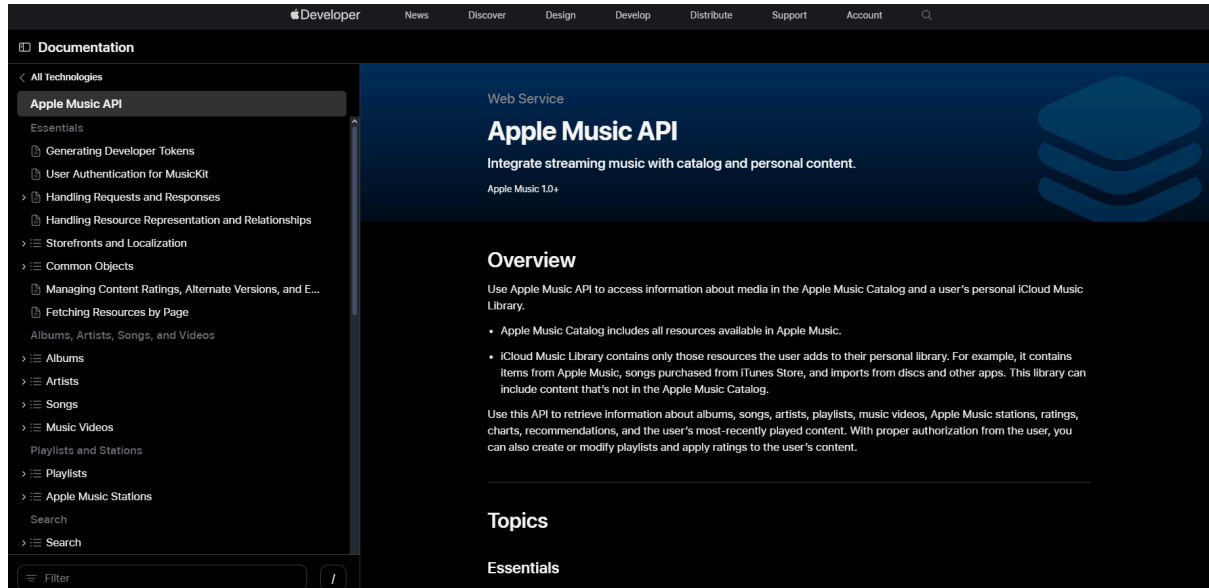


Oltre ai database pubblici, l'ontologia si allinea con standard esistenti, come le tassonomie musicali utilizzate dalle principali piattaforme di streaming. Servizi come:

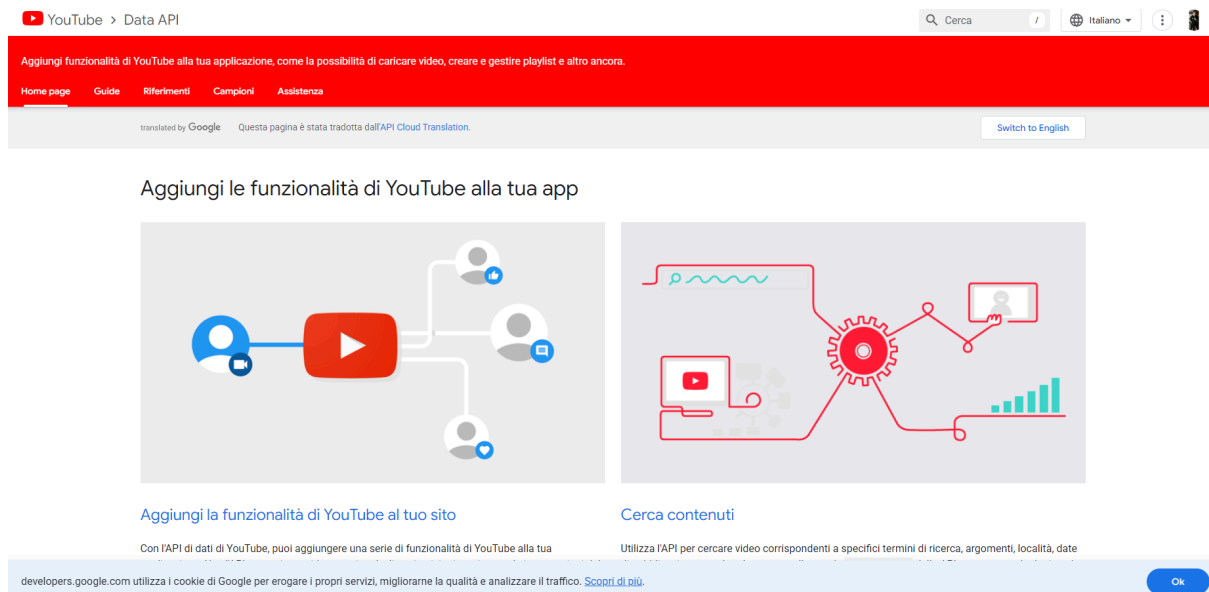
- Spotify API (<https://developer.spotify.com/documentation/web-api>)



- Apple Music  
(<https://developer.apple.com/documentation/applemusicapi>)



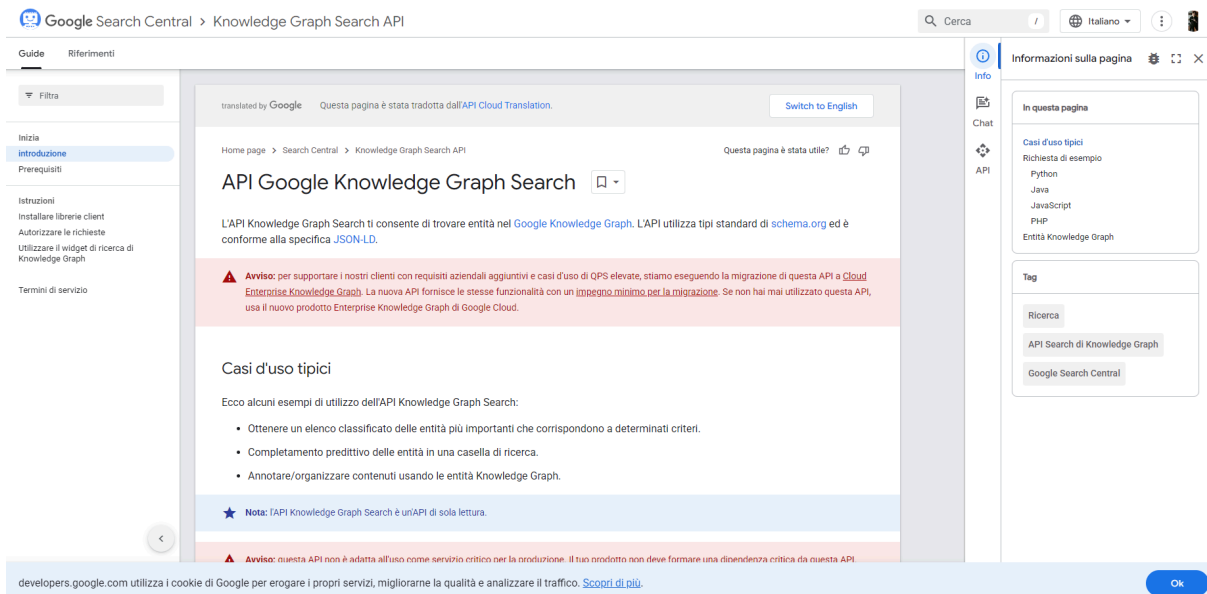
- YouTube Music (<https://developers.google.com/youtube/v3?hl=it>)



Questi esempi implementano algoritmi avanzati di categorizzazione per classificare la musica in base a genere, atmosfera e metriche di ascolto.

- Knowledge Graph di Google  
(<https://developers.google.com/knowledge-graph>): che associa generi musicali, artisti e album basandosi su relazioni semantiche e dati

strutturati. Questo sistema permette una ricerca più intuitiva e una migliore navigazione tra le informazioni musicali.



L'ontologia segue una logica simile a [Spotify](#) (6TIYQ3jFPwQSRmorSezPxX), permettendo un'organizzazione sistematica dei dati musicali e migliorando la fruizione delle informazioni da parte degli utenti. Grazie all'utilizzo di modelli semantici, è possibile interrogare i dati attraverso query strutturate, facilitando il recupero di informazioni precise e dettagliate.

## Esempio reale e dati di riferimento

Per dimostrare l'applicazione dell'ontologia, consideriamo il caso reale dell'artista **Machine Gun Kelly** e del suo album "Tickets to My Downfall". Nell'ontologia, questo caso può essere rappresentato attraverso le seguenti relazioni:

- L'artista **Machine Gun Kelly** ha pubblicato l'album **Tickets to My Downfall**.
- L'album **Tickets to My Downfall** contiene il brano **Bloody Valentine**.
- Il brano **Bloody Valentine** appartiene al genere **Pop Punk**.
- Il brano **Bloody Valentine** ha un testo associato.

## Allineamento con ontologie esistenti

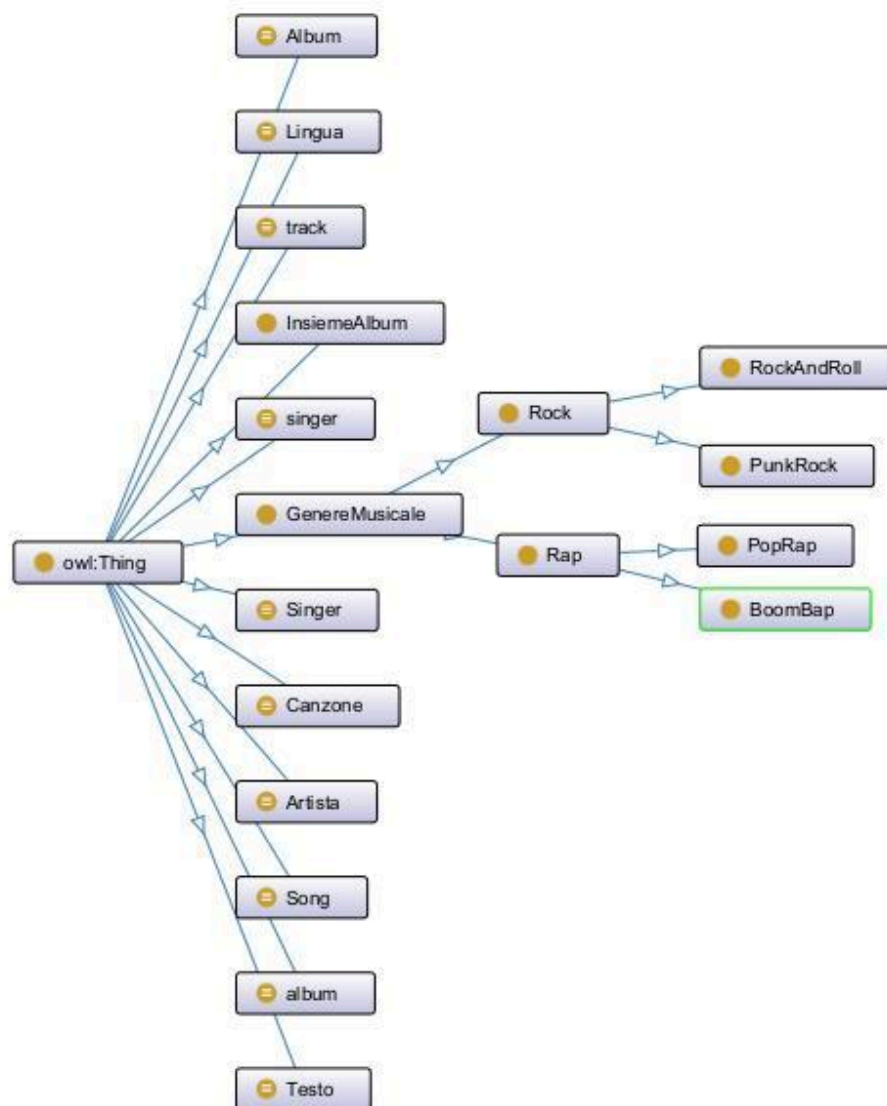
Per garantire la coerenza e l'integrazione con altri modelli semantici, l'ontologia è stata allineata con le seguenti ontologie di riferimento:

- Schema.org (<https://schema.org/MusicComposition>): utile per la descrizione delle entità musicali e delle relazioni tra artisti, brani e album.
- Music Ontology (<http://musicontology.com/>): un'ontologia sviluppata per descrivere le informazioni musicali con RDF e OWL.
- DBpedia (<https://wiki.dbpedia.org/>): utilizzata per l'allineamento delle entità musicali con dati estratti da Wikipedia.

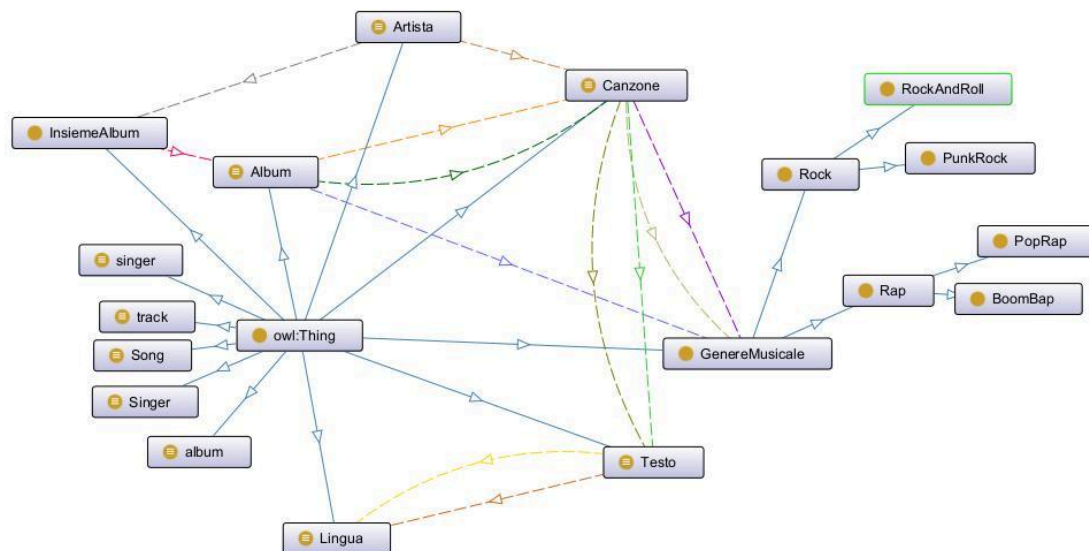
Questa documentazione fornisce una base solida per l'ontologia musicale, consentendo di rappresentare in modo dettagliato le relazioni tra i vari elementi del dominio musicale e facilitando l'integrazione con altri dataset semantici e knowledge graph.

## 7. Visualizzazione

L'ontologia include:



Con relazioni:



## Tassonomia delle classi

L'ontologia è organizzata in classi principali come:

- **Album** → contiene più **Brani**
- **Brano** → ha un **Genere**, può appartenere a un **Album** e ha un **Artista**
- **Artista** → può avere più **Album** e **Brani**

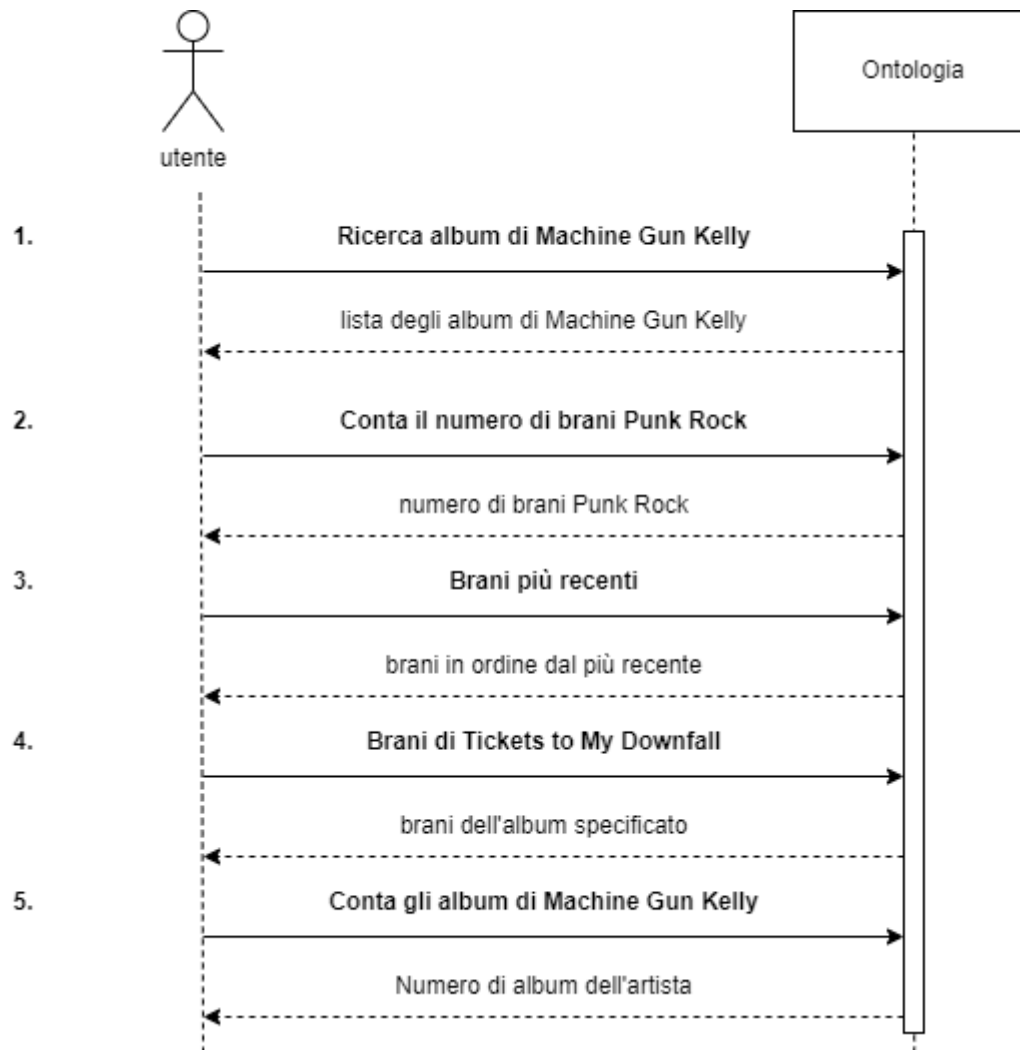
## Triple in tabella

Soggetto	Predicato	Soggetto
Machine Gun Kelly	haAlbum	Tickets to My Downfall
Tickets to My Downfall	contieneBrano	Title Track
Title Track	haGenere	PunkRock



## PARTE II

### QUERY



## Mockup

Cerca

Nome : \_\_\_\_\_

ARTISTA

ALBUM

CANZONE

V

Data : \_\_\_\_\_

Genere : \_\_\_\_\_

RISULTATI

### 1. Ricerca album di Machine Gun Kelly

Query:

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

*PREFIX music: <http://www.music.com/brani/>*

```
SELECT ?artista (COUNT(DISTINCT ?titoloAlbum) AS ?numeroAlbum)
(GROUP_CONCAT(DISTINCT ?titoloAlbum; SEPARATOR=", ") AS ?albumList)
WHERE {
  ?artista a music:Artista ;
    rdfs:label ?nomeArtista ;
    music:haInsiemeAlbum ?insiemeAlbum .
  ?insiemeAlbum music:contieneAlbum ?album .
```

*?album rdfs:label ?titoloAlbum .  
FILTER(?nomeArtista = "MachineGunKelly")  
} GROUP BY ?artista*

	artista	numeroAlbum	albumList
1	music:MachineGunKelly	"3"^^xsd:integer	"Hotel_Diablo, Mainstream_Sellout, Tickets_to_My_Downfall"

Descrizione: Questa query estrae tutti gli album pubblicati dall'artista "Machine Gun Kelly", ordinati per data di pubblicazione.

## 2. Conta il numero di brani Punk Rock

Query:

*PREFIX music: <http://www.music.com/brani/>*

*SELECT (COUNT(?canzone) AS ?numeroBraniPunkRock) WHERE {  
?canzone music:canzoneDiGenere music:PunkRock .  
}*

	numeroBraniPunkRock
1	"6"^^xsd:integer

Descrizione: Conta il numero totale di brani che appartengono al genere "Punk Rock".

## 3. Visualizza i brani più recenti

Query:

*PREFIX music: <http://www.music.com/brani/>*

*SELECT ?canzone ?dataUscita WHERE {  
?canzone music:dataDiUscita ?dataUscita .  
} ORDER BY DESC(?dataUscita) LIMIT 10*

	canzone	dataUscita
1	music:Eminem-Evil	"2024-07-12"
2	music:Eminem-Renaissance	"2024-07-12"
3	music:Eminem-Trouble	"2024-07-12"
4	music:MGK-Born_With_Horns	"2022-03-25"
5	music:MGK-Drunk_Face	"2020-09-25"
6	music:MGK-Lonely	"2020-09-25"
7	music:MGK-Make_Up_Sex	"2020-09-25"
8	music:MGK-Maybe	"2020-09-25"
9	music:MGK-Title_Track	"2020-09-25"
10	music:MGK-Burning_Memories	"2019-07-05"

Descrizione: Estrae gli ultimi 5 brani più recenti, ordinati per data di pubblicazione in ordine decrescente.

#### 4. Recupera i brani di Tickets to My Downfall

Query:

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

PREFIX music: <http://www.music.com/brani/>

```
SELECT ?album (COUNT(DISTINCT ?titoloCanzone) AS ?numeroCanzoni)
(GROUP_CONCAT(DISTINCT ?titoloCanzone; SEPARATOR=", ") AS ?canzoni)
WHERE {
    ?album a music:Album ;
        rdfs:label "Tickets_to_My_Downfall" ;
        music:contieneCanzone ?canzone .
    ?canzone rdfs:label ?titoloCanzone .
} GROUP BY ?album
```

	album	numeroCanzoni	canzoni
1	music:Tickets_to_My_Downfall	"3"^^xsd:integer	"MGK-Drunk_Face, MGK-Lonely, MGK-Title_Track"

Descrizione: Recupera tutti i brani contenuti nell'album "Tickets to My Downfall".

#### 5. Conta il numero di album per artista

Query:

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

PREFIX music: <http://www.music.com/brani/>

```
SELECT ?artista (SAMPLE(?nomeArtista) AS ?nomeUnico) (COUNT(DISTINCT
?album) AS ?numeroAlbum) WHERE {
    ?artista a music:Artista ;
        rdfs:label ?nomeArtista ;
        music:haInsiemeAlbum ?insiemeAlbum .
    ?insiemeAlbum music:contieneAlbum ?album .
} GROUP BY ?artista
```

	artista	numeroAlbum
1	music:Eminem	"3"^^xsd:integer
2	music:MachineGunKelly	"3"^^xsd:integer

Descrizione: Conta quanti album ha pubblicato ogni artista presente nel knowledge graph.

---

## Base di regole SWRL

### 1. Regola per assegnare automaticamente il genere "Rap" a brani con parole chiave nel titolo

Se una canzone ha un titolo contenente "Rap" o "HipHop", allora viene assegnata al genere "Rap"

```
Canzone(?c) ^ rdfs:label(?title) ^ swrlb:contains(?title, "Rap") -> canzoneDiGenere(?c, Rap)
Canzone(?c) ^ rdfs:label(?title) ^ swrlb:contains(?title, "HipHop") -> canzoneDiGenere(?c, Rap)
```

---

### 2. Regola per determinare la lingua della canzone in base alla lingua del testo

Se una canzone contiene un testo scritto in una lingua specifica, allora la canzone è in quella lingua

```
Canzone(?c) ^ contieneTesto(?c, ?t) ^ scrittoInLingua(?t, ?l) -> linguaCanzone(?c, ?l)
```

---

### 3. Regola per identificare gli album che contengono almeno due canzoni

Se un album contiene almeno due canzoni, allora appartiene alla classe albumConAlmenoDueCanzoni

```
Album(?a) ^ contieneCanzone(?a, ?c1) ^ contieneCanzone(?a, ?c2) ^ swrlb:notEqual(?c1, ?c2) -> albumConAlmenoDueCanzoni(?a)
```

---

### 4. Regola di mapping tra ontologie

# Se un individuo appartiene alla classe "https://dbpedia.org/ontology/Song", allora appartiene anche alla classe "Canzone" dell'ontologia locale

```
https://dbpedia.org/ontology/Song(?s) -> Canzone(?s)
```

---

## **5. Regola con operatore built-in per normalizzare i nomi degli artisti**

Se un artista ha un nome in lettere maiuscole, convertirlo in minuscole per uniformità

Artista(?a) ^ haNome(?a, ?n) ^ swrlb:upperCase(?upper, ?n) -> haNome(?a, ?upper)