Università degli Studi di Salerno

Dipartimento di Informatica



**Progetto di Basi di Dati 2**

Analisi brani 1921-2020 estratti da Spotify Web API

**Professori: Studenti:**

Genoveffa Tortora Federico Ripoli 0522500921

Loredana Caruccio Tullio Pizzuti 0522500760

**Anno Accademico 2021/2022**

# Progetto

Il progetto consiste nella realizzazione di un applicativo web basato su Node.JS per l’interrogazione di un database NoSQL basato su documenti chiamato MongoDB.

Verranno estratte informazioni e statistiche sugli autori e sui brani, i quali hanno una data di rilascio che va dai primi anni ‘20 fino all’anno 2020.

Attraverso Node.JS sono state sviluppate delle REST API che restituiscono i risultati in formato JSON. Alcune parti delle pagine invece vengono combinate e preprocessate lato server dal sistema di template EJS, mentre le API vengono richiamate lato client attraverso richieste AJAX.

Lato client sono state utilizzati diversi plug-in JavaScript, tra cui JQuery per la semplificazione della stesura del codice e Chart.js per la costruzione di grafici. Per quanto riguarda lo stile delle pagine è stato utilizzato Bootstrap 5 e in particolare il template Mazer. È stato infine utilizzato Mongoose come object modeling per node.js.

# Dataset

Abbiamo prelevato il dataset dal sito Kaggle, il quale è stato costruito estraendo i dati dalla piattaforma Spotify attraverso le Spotify Web API. Si compone di due file in formato csv, artists.csv e tracks.csv, i quali contengono rispettivamente informazioni sugli artisti e sui brani. Le date di rilascio dei brani vanno dal 1921 al 2020. Il file artists.csv ha una dimensione di circa 65MB e contiene le informazioni di 1.162.095 artisti, mentre il file tracks.csv ha una dimensione di circa 112MB e contiene i dati di 586.052 brani.

## Features

Ogni file contiene diverse informazioni. Il file artists.csv presenta:

* Id: identificativo univoco di 22 caratteri
* Followers: numero di followers
* Genres: array contenente i generi
* Name: nome
* Popularity: percentuale di popolarità

Mentre il file tracks.csv:

* Id: identificativo univoco di 22 caratteri
* Name: nome
* Popularity: percentuale di popolarità
* Duration\_ms: durata espressa in millisecondi
* Explicit: booleano (0/1) che indica se il brano è esplicito o meno
* Artists: array contenente i nomi degli artisti che partecipano al brano
* Id\_artists: array contenente gli identificativi degli artisti che partecipano al brano
* Release\_date: data di rilascio del brano
* Danceability: percentuale che esprime quanto il brano si adatta al ballo
* Energy: percentuale che rappresenta una misura percettiva di intensità e attività
* Key: rappresenta la chiave in cui viene suonato il brano
* Loudness: esprime il volume complessivo del brano in decibel (dB)
* Mode: Mode indica la modalità (maggiore o minore) di un brano, il tipo di scala da cui deriva il suo contenuto melodico. Il maggiore è rappresentato da 1 e il minore è 0
* Speechiness: un valore percentuale che indica quanto il brano è simile al parlato
* Acousticness: una percentuale che indica quanto il brano sia acustico
* Instrumentalness: percentuale che indica la percentuale di voci nel brano
* Liveness: probabilità che il brano sia un live
* Valence: valore compreso tra 0 e 1 che indica quanto un brano risulta positivo (valori vicino all’1) o negativo (valori vicini allo 0)
* Tempo: indica il tempo complessivo stimato di un brano in battiti al minuto (BPM)
* Time\_signature: indica il tempo in chiave stimato (es. 3/4, 7/4, ecc.…)

Ulteriori informazioni sulle features sono espresse nella documentazione delle Spotify Web API.

## Import e trasformazioni

I file sono stati importati in MongoDB attraverso l’apposito comando shell mongoimport. Il database è stato chiamato Spotify mentre le collection artists e tracks rispettando così i nomi dei file.

Successivamente all’importazione vengono apportate delle modifiche ai dati.

Per la collection artists:

* Genres: il campo viene importato come stringa e non come array. Vengono quindi rimossi dalla stringa i caratteri speciali ‘,”,[, e ]. Successivamente viene effettuato uno split della stringa sul carattere “,”. Nel caso in cui nel campo sia presente la stringa vuota, viene rimpiazzata da un array vuoto.
* Followers: nel caso in cui il dato non sia presente, esso viene importato come una stringa vuota. In questo caso abbiamo sostituito la stringa vuota con lo 0.

Per la collection tracks:

* Artists: il campo viene importato come stringa e non come array. Vengono quindi rimossi dalla stringa i caratteri speciali ‘,”,[, e ]. Successivamente viene effettuato uno split della stringa sul carattere “,”.
* Id\_artists: il campo viene importato come stringa e non come array. Vengono quindi rimossi dalla stringa i caratteri speciali ‘,”,[, e ]. Successivamente viene effettuato uno split della stringa sul carattere “,”.
* Release\_date: viene importata come una stringa. Inoltre su alcuni brani viene espresso solo l’anno e non la data completa. In questo caso all’anno vengono aggiunti mese e anno trasformando la data in 01/01/yyyy. Successivamente viene convertita la stringa in oggetto Date.

## Indici

Sono stati aggiunti degli indici su entrambe le collection in modo da velocizzare le operazioni di ricerca, ordinamento e raggruppamento. Gli indici sono stati definiti in modo da migliorare le operazioni di lettura per determinate query che fanno uso di quei campi. Gli indici sono tutti di tipo ASC. Gli indici su Id e Id\_artists permettono di velocizzare l’operazione di lookup tra le due collection.

Artists:

* Id: 34.8MB
* Name: 21.4MB
* Followers: 5.6MB
* Popularity: 9.4MB

Tracks:

* Name: 12.7MB
* Id\_artists: 6.8MB
* Id\_artists.1 (sul secondo valore dell’array): 1.4MB
* Artists: 6.8MB
* Release\_date: 6.1MB
* Popularity: 2.9MB
* Duration\_ms: 4.2MB

## Query

Sono state costruite diversi tipi di query, ognuna delle quali viene utilizzata per mostrare i risultati in una pagina web apposita.

Abbiamo query di tipo generico, ovvero applicabili ad entrambe le collection:

* Find: ricerca, attraverso uno o più filtri, su uno o più attributi di una collection. È possibile inoltre ordinare sulla base di un campo e un tipo di ordinamento (ASC/DESC), limitare il numero di documenti restituiti e indicare un offset che scarta i primi n risultati
* Count: conteggio dei documenti presenti in una collection, sulla base di uno o più filtri sugli attributi di una collection
* Min/Max: ricerca del minimo/massimo di un attributo su una collection
* Duplicates: ricerca di duplicati, attraverso il group su uno o più attributi di una collection

Find e count sono utilizzate come supporta alla vista dei documenti tramite tabelle, min/max per il raggruppamento dei documenti a scaglioni sulla base di un attributo, mentre duplicates è utilizzata nell’apposita pagina web per valutare la bontà dei dati delle collections.

Abbiamo poi query specifiche per la collection artists:

* ArtistDetail: ricerca l’artista sulla base del suo id
* Find: filtrando sui campi name e genres e ordinando su name, popularity, genres e followers
* Duplicates: Utilizzando i campi name, genres, followers, popularity
* Group: utilizzata per raggruppare gli artisti in n scaglioni (costruiti basandosi sul min/max dell’attributo), limitata agli attributi followers e popularity
* ListGenres: estrae tutti i diversi tipi di generi presenti nell’attributo genres. È possibile filtrare sul nome del genere e imputare la mancanza di genere dell’artista come “Unspecified”, impostare un limite e un offset sul risultato e un ordinamento sui campi name, count, popularity e followers. La query restituisce anche il conteggio degli artisti che appartengono a quel genere, la media della popolarità del genere e il numero totale di followers del genere.
* GroupArtistsGenres: individua e conta tutti gli artisti che hanno un particolare genere. È possibile filtrare i risultati sulla base del numero di generi dell’artista, verificando se l’artista oltre al genere scelto ne ha associati altri. Il filtro è espresso in termini di minimo e massimo numero di generi associati.

Le query specifiche per la collection tracks invece sono:

* Find: limitata ai campi name e artists e ordinando su name, artists, release\_date, duration\_ms e popularity
* Duplicates: Utilizzando i campi name, popularity, duration\_ms, explicit, id\_artists, release\_date, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo e time\_signature
* FindByArtistId: restituisce i brani di un artista ricercandole attraverso il suo id. Permette di filtrare sul nome del brano, indicare un range sulla data di rilascio e sugli artisti che hanno partecipato al brano (solo, featuring o entrambe). Inoltre, è possibile ordinare sui campi name, artists, release\_date, duration\_ms e popularity ed indicare limit e offset
* CountTracksByYears: raggruppa per anno o anno/mese i brani di un artista restituendo il conteggio
* AvgPopularityByYear: raggruppa per anno o anno/mese i brani di un artista restituendo la media della popolarità
* GroupTracks: utilizzata per raggruppare i brani in n scaglioni (costruiti basandosi sul min/max dell’attributo), limitata agli attributi duration\_ms e popularity

Sono state poi costruite 3 query di lookup in modo da incrociare i dati sulle due collections basandosi sugli attributi id e id\_artists:

* Lookup: effettua il lookup tra le due collections andando a contare il numero di brani per ogni artista. Permette di filtrare sia il nome dell’artista che il nome del brano, limitare i risultati e definire il tipo di ordinamento sul conteggio dei brani
* LookupArtistsTracks: filtrando e raggruppando sugli id degli artisti, mostra le medie delle features danceability, energy, speechiness, acousticness, explicit, liveness, duration\_ms
* LookupGenresTracks: filtrando e raggruppando per genere, mostra le medie delle features danceability, energy, speechiness, acousticness, explicit, liveness, duration\_ms

# Applicativo

Per la realizzazione dell’applicativo si è scelto di utilizzare un approccio web-based in modo da poter sfruttare l’adattamento ai vari sistemi operativi, in quanto è possibile accedere all’applicazione da un normale browser, la facilità e velocità di costruzione delle interfacce oltre che l’adattabilità ai vari tipi di schermi attraverso tecnologie responsive e la facilità di integrazione tra il web server scelto (Node.JS) e MongoDB. Le pagine HTML sono costruite tramite tamplating, infatti ogni pagina è composta da 4 parti, di cui 3 fisse e una variabile in base alla pagina (body):

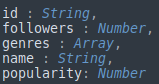
* Head: include i metadati delle pagine, i template css ed eventuali file JavaScript
* Footer: contiene i file JavaScript utilizzati da tutte le pagine
* Menu: contenente il menu di navigazione
* Body: è la parte variabile che viene definita pagina per pagina

I principali metodi di visualizzazione dei dati estratti, tabelle e grafici. Per le tabelle è stato utilizzato il plugin JQuery chiamato DataTables, mentre per i grafici è stata usata la libreria JavaScript Chart.Js.

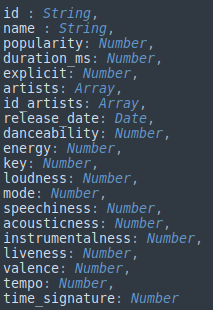
## Object modeling

Attraverso il modulo Mongoose di Node.JS abbiamo costruito gli object model per le due collections:

**Artists:**



**Tracks:**



## Routes

Per Routing si intende determinare come un’applicazione risponde a una richiesta client a un endpoint particolare, il quale è un URI (o percorso) e un metodo di richiesta HTTP specifico (GET, POST e così via). Il nostro applicativo, avendo metodi solo di lettura, risponde solo a richieste di tipo GET.

Abbiamo due routes principali:

* /api: questo path individua tutte le richieste di accesso ai dati attraverso richieste GET. Ad ogni path corrisponde un metodo in un controller che è specializzato nella risposta
* /... : Tutti gli altri path permetto di accedere a pagine in formato HTML

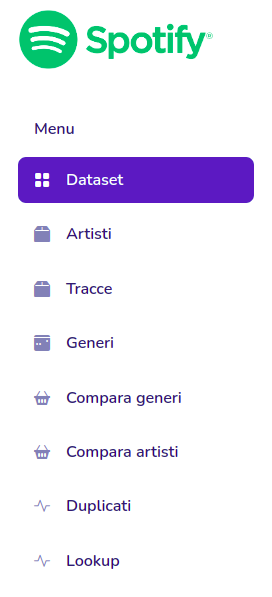
## Controller

Vi sono 3 tipi di controller che permettono l’accesso alle collections del database:

* CommonController: contiene le query comuni tra cui find, count, ecc.… Questo controller gestiste solo l’accesso al database e non l’analisi della richiesta HTTP
* ArtistsController, TracksController: gestiscono rispettivamente gli accessi alle collections. Ogni metodo del controller corrisponde ad uno specifico path e richiesta HTTP. Si occupano quindi di gestire i parametri ricevuti nella richiesta (path o query params) e della risposta in formato JSON. La risposta è costruita a seconda del tipo di visualizzazione dei dati, in formato tabellare o attraverso un grafico.

## Navigazione

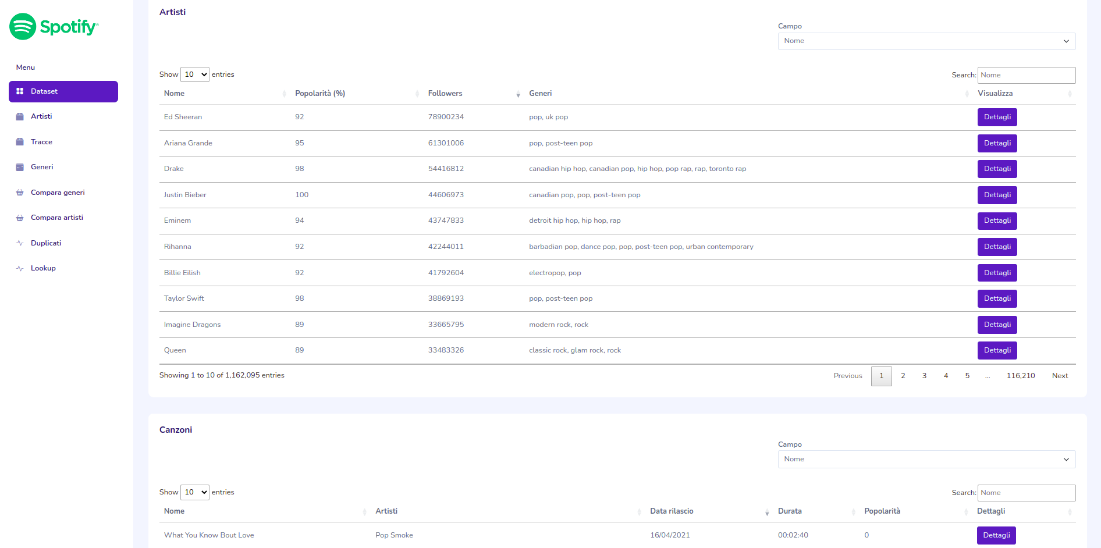
La navigazione dell’applicazione è stata gestita tramite un menu verticale posizionato sulla sinistra della pagina.



Verranno descritte di seguito le pagine e il loro funzionamento.

### Dataset

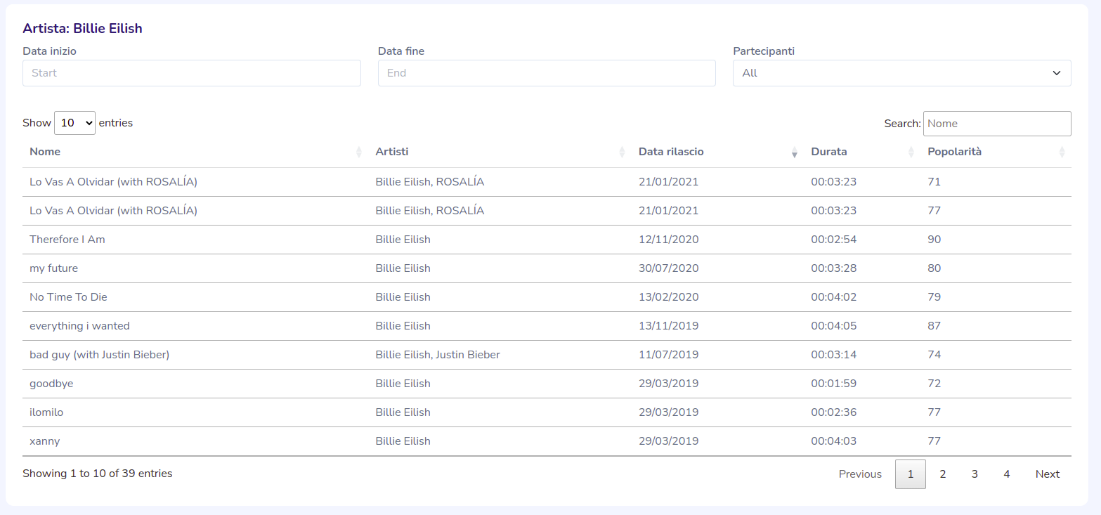
È la pagina principale dell’applicazione dalla quale è possibile ricercare all’interno delle collections, utilizzando anche i filtri di ricerca presenti sulle tabelle. Ogni tabella mostra il conteggio dei risultati estratti oltre che all’indice corrente. È possibile scorrere tra i risultati attraverso la paginazione posta in basso a destra della tabella. Inoltre c’è la possibilità di modificare l’ordinamento premendo sul titolo della colonna.

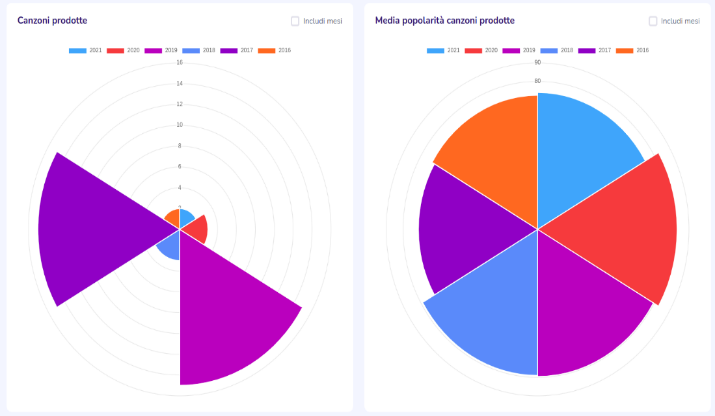


#### Dettaglio artista

Utilizzando l’apposito tasto presente alla fine di ogni riga nella tabella degli artisti viene mostrata la pagina di dettaglio dell’artista dove sono presenti:

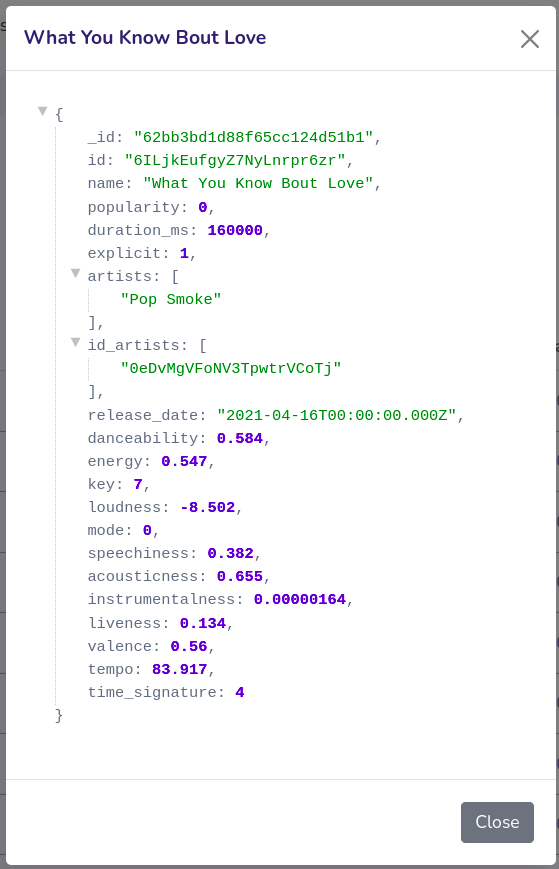
* Tabella dei brani prodotti dall’artista
* Grafico Polar Area che mostra il numero di brani rilasciati annualmente o mensilmente dall’artista
* Grafico Polar Area che mostra la media della popolarità dei brani rilasciati annualmente o mensilmente dall’artista
* Grafico che mostra l’andamento della popolarità dei brani dell’artista nel tempo





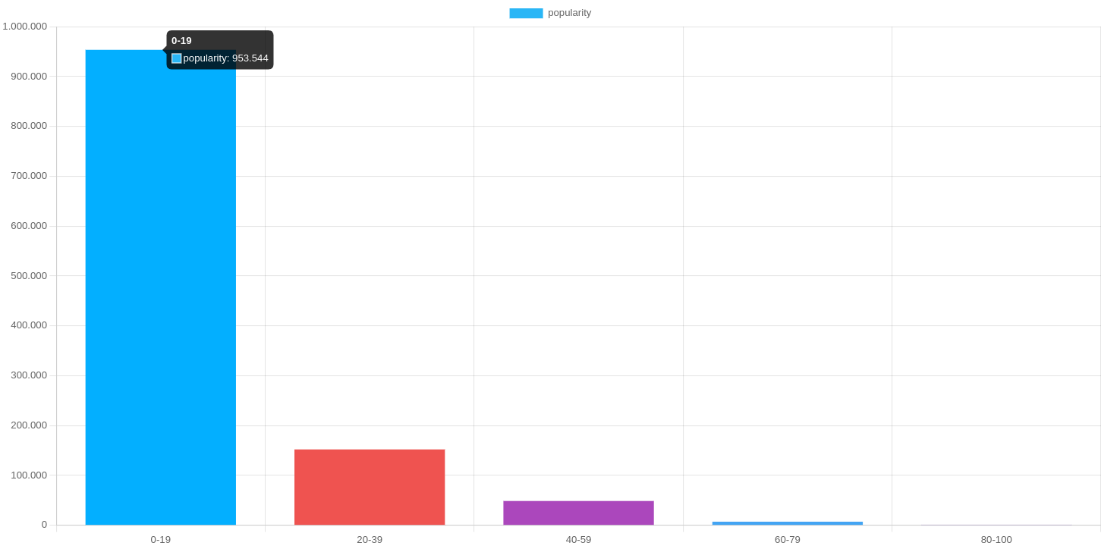
#### Dettaglio traccia

Utilizzando l’apposito tasto presente alla fine di ogni riga nella tabella delle tracce, è possibile mostrare gli attributi di una traccia presenti nella collection.



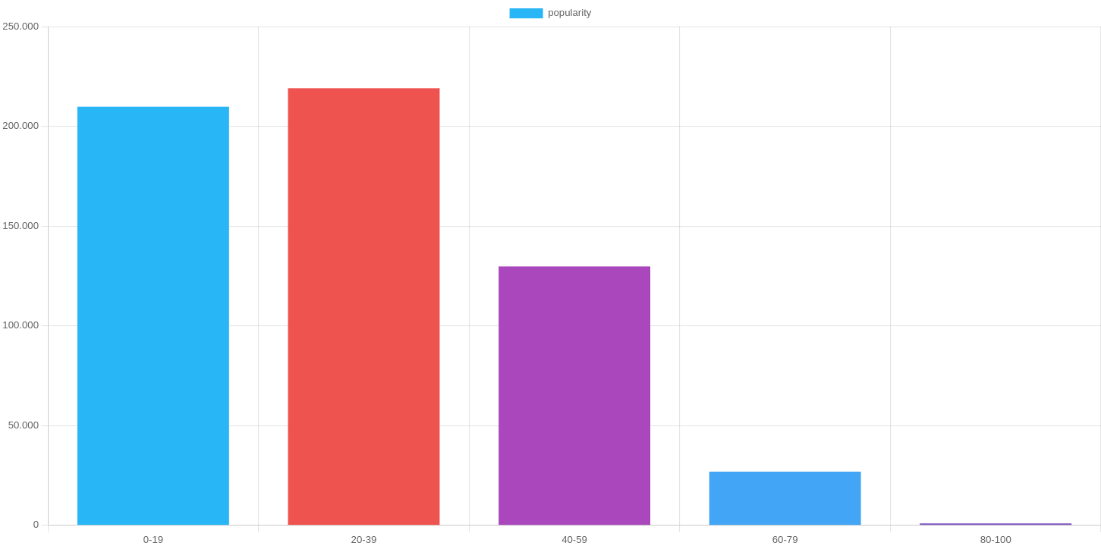
### Artisti

Mostra i grafici di raggruppamento per scaglioni degli artisti attraverso gli attributi followers e popolarità. È possibile modificare la taglia degli scaglioni attraverso il filtro posto in alto a destra di ogni grafico.



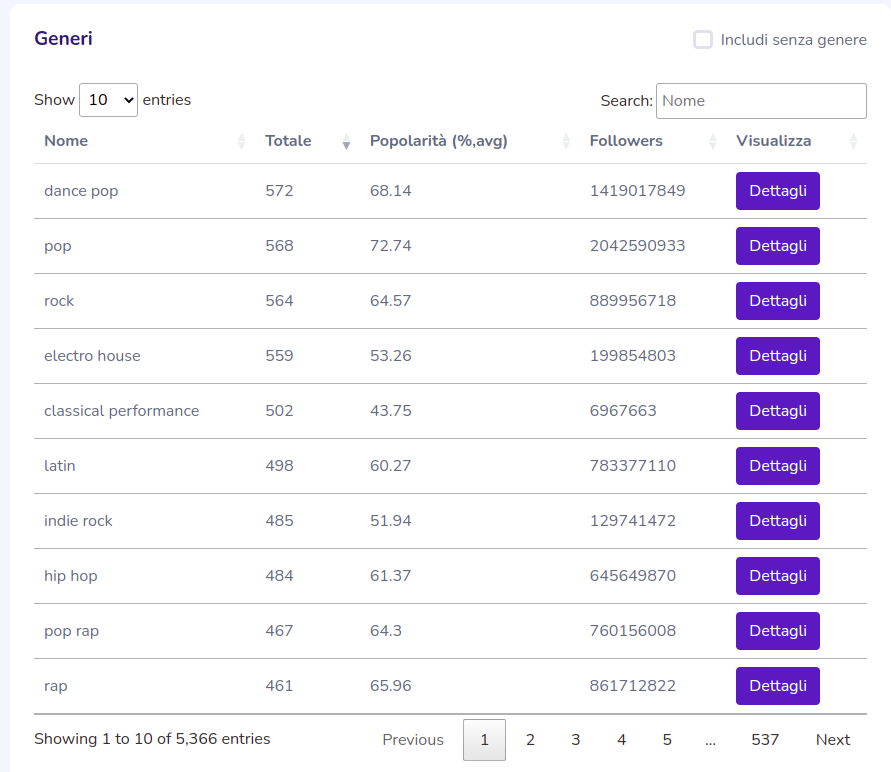
### Tracce

Mostra i grafici di raggruppamento per scaglioni dei brani attraverso gli attributi durata e popolarità. È possibile modificare la taglia degli scaglioni attraverso il filtro posto in alto a destra di ogni grafico.

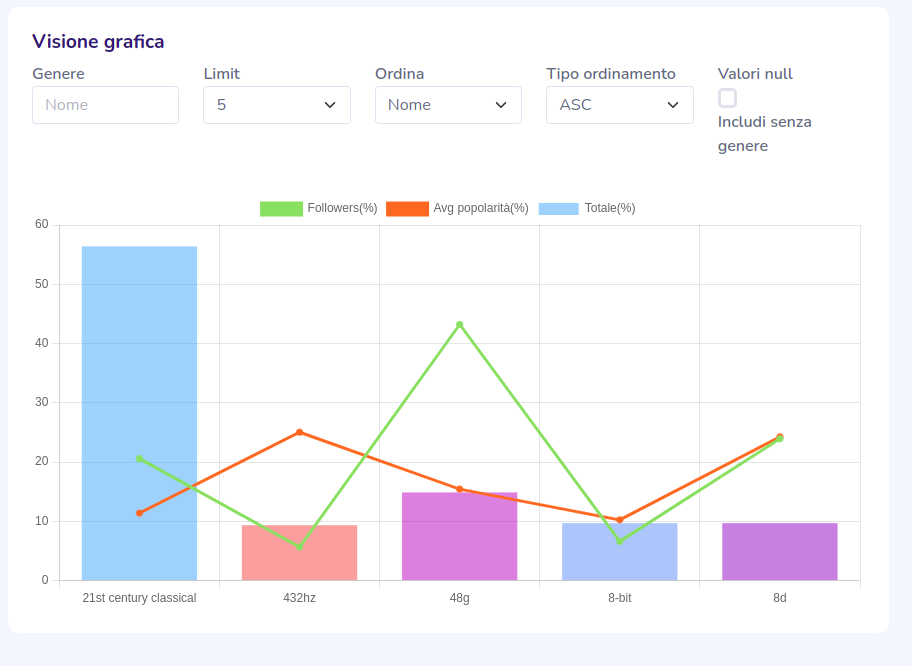


### Generi

Mostra la lista dei generi presenti nell’attributo genres della collection artists, visualizzando oltre al nome anche il numero totale degli artisti che hanno quel particolare genere, la media della popolarità e il numero di followers totali del genere. È possibile poi visualizzare in dettaglio il genere.

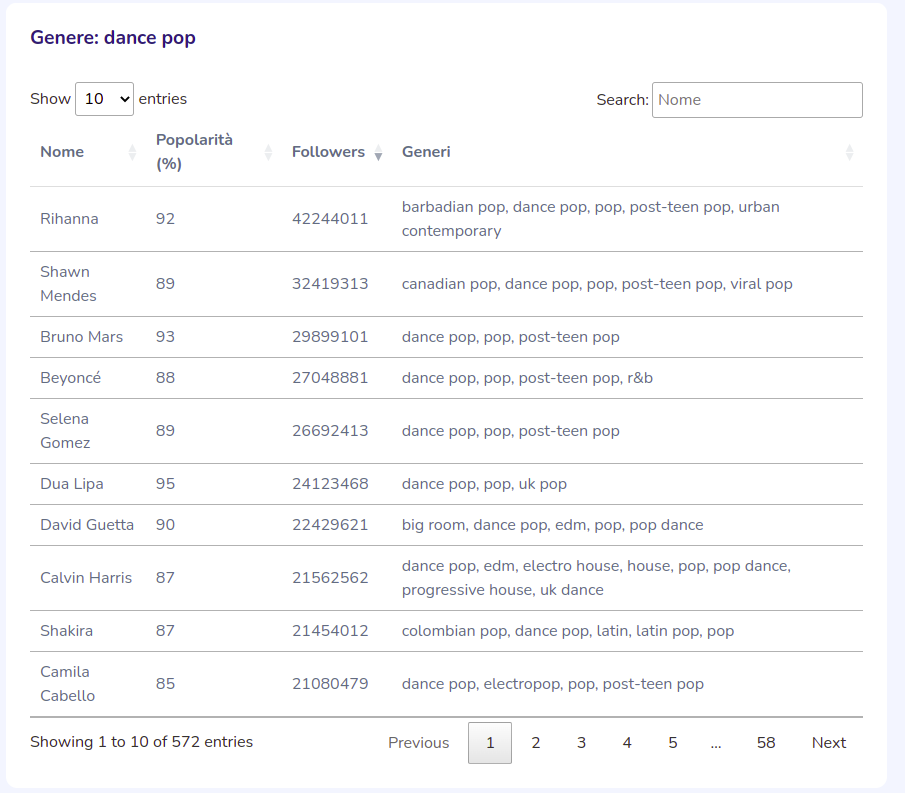


Oltre alla lista dei generi, viene anche mostrato un grafico con dati sovrapposti (followers, media popolarità e totale) per compare diversi tipi di generi. È possibile utilizzare i filtri di ricerca per specificare un particolare nome, un limite al risultato, l’ordinamento su uno specifico campo e il tipo di ordinamento. È possibile poi specificare di voler visualizzare anche il genere “Unspecified”, ovvero gli artisti che non hanno genere.

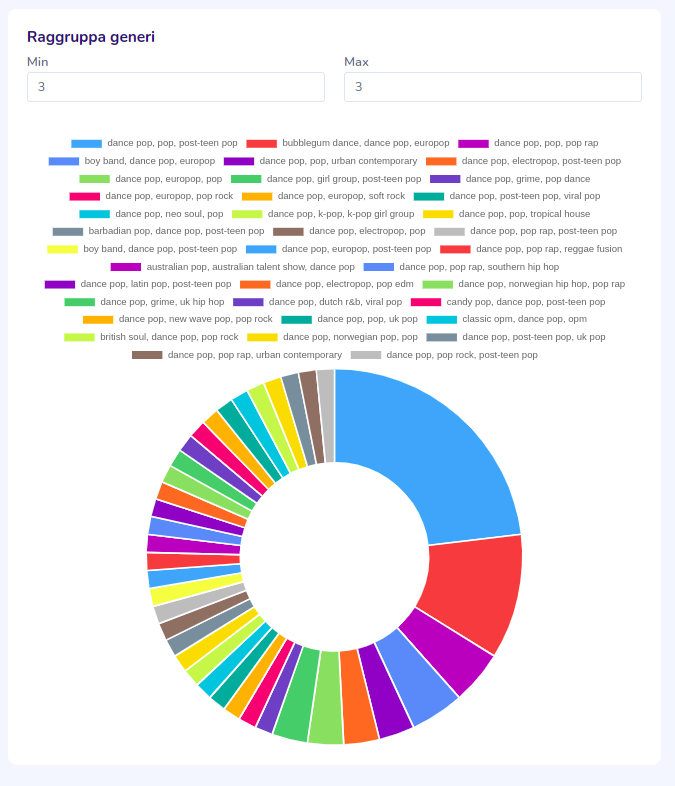


#### Dettaglio genere

Questa pagina permette di visualizzare gli artisti appartenenti a quel genere.

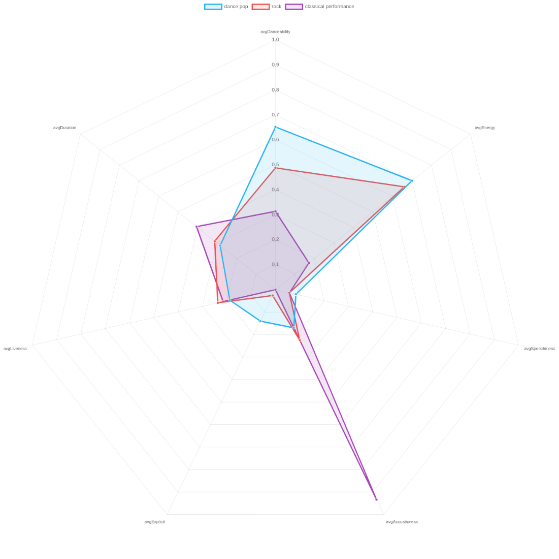


Inoltre, viene mostrato attraverso un grafico Doughnut il risultato della query GroupArtistsGenres.



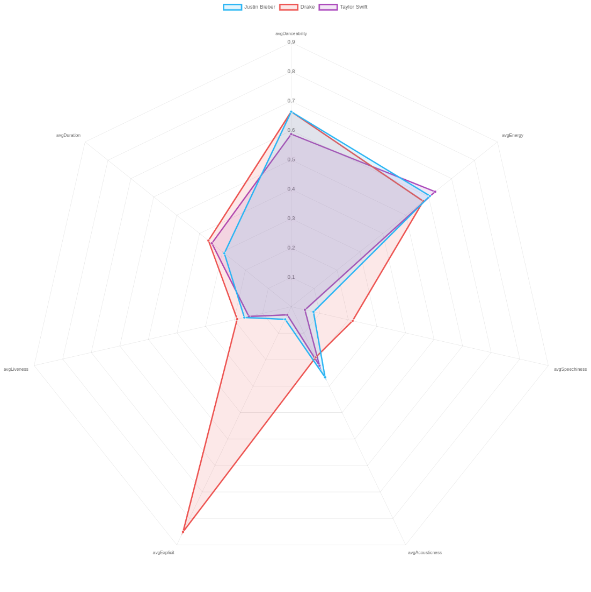
### Compara generi

Questa pagina permette di selezionare differenti generi da comparare. Permette di mettere in evidenza le differenze di durata dei brani, volume, energia, danzabilità, vitalità, acustica, esplicità e la discorsività attraverso un Radar Chart. La pagina è formata da due sezioni, una tabella in alto, attraverso la quale è possibile aggiungere o rimuovere i generi dal grafico presente nella parte sottostante.



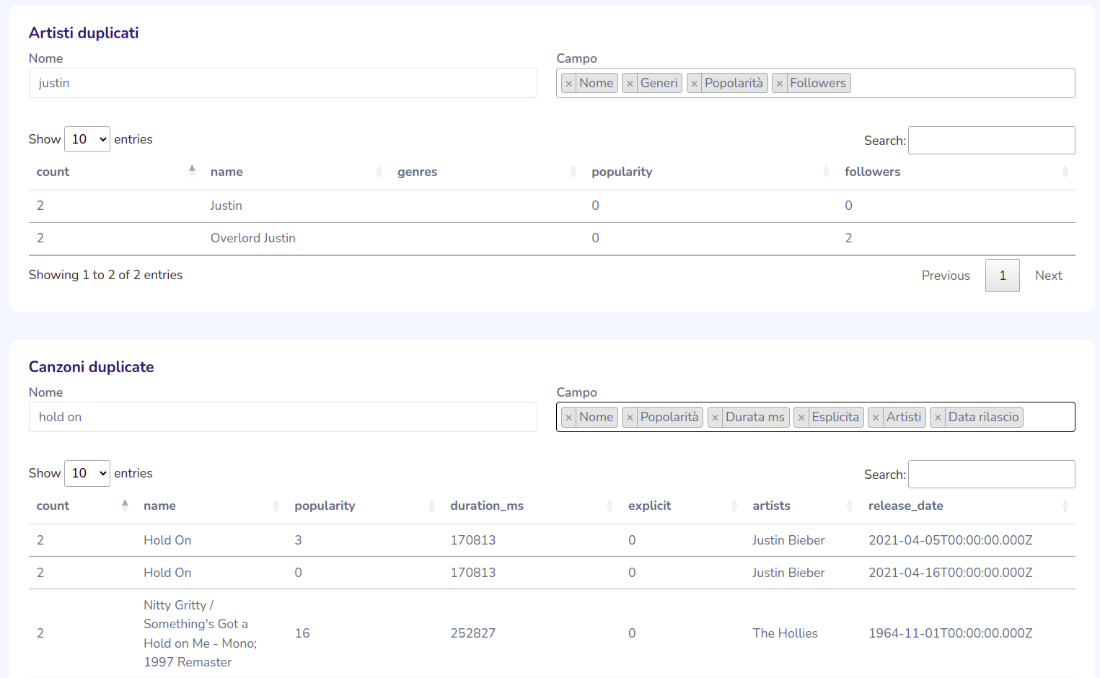
### Compara artisti

Ha la stessa funzionalità della pagina compara generi, tuttavia mette in relazione i singoli artisti.



### Duplicati

Questa pagina permette di esplorare i dataset alla ricerca di duplicati. Il suo funzionamento è quello di andare a raggruppare artisti/brani sulla base di uno o più attributi, ed è inoltre possibile specificare un filtro testuale sull’attributo name su entrambe le collections.



### Lookup

Questa funzione permette di andare ad effettuare una lookup dalla collection artists a tracks in modo da conteggiare quanti brani hanno gli artisti all’interno dell’altra collection. È possibile specificare un filtro testuale sul nome dell’artista e del brano, limitare il numero di risultati ottenuti e specificare il tipo di ordinamento sul conteggio dei brani. Tuttavia, è una funzione che impiega in media 1 minuti e 30 secondi se non vengono specificati filtri, per questo viene anche mostrato il tempo di inizio e di fine della richiesta.

