

# Relazione del Progetto di Raccomandazione di Canzoni

Giaconi Christian, Giacomo Rossi  
Matricola: 314045, 314671

3 gennaio 2025

## 1 Specifica del problema

Il progetto consiste nell'implementazione di un sistema di raccomandazione musicale in due linguaggi: Haskell e Prolog. L'obiettivo del programma è di suggerire una classifica di canzoni all'utente in base alle sue preferenze di genere musicale, utilizzando un sistema di punteggio ponderato per prioritizzare i risultati. Il contesto applicativo prevede l'uso del programma per scopi personali o di analisi musicale, caricando dati da file strutturati.

Le funzionalità principali includono:

- Caricamento di un elenco di canzoni da un file di testo.
- Specifica dei generi preferiti e di un peso numerico per valorizzarli.
- Calcolo dei punteggi ponderati delle canzoni.
- Generazione di una classifica ordinata in base ai punteggi calcolati.

## 2 Analisi del problema

### 2.1 Dati in ingresso

Il programma richiede:

- Un file contenente un elenco di canzoni, con ciascuna riga nel formato:

`Titolo,Artista,Genere,Punteggio`

- Una lista di generi musicali preferiti specificata dall'utente.
- Un valore numerico (peso) per i generi preferiti, che amplifica la rilevanza delle canzoni appartenenti a quei generi.

### 2.2 Dati in uscita

Il programma produce:

- Una lista di canzoni ordinata in base ai punteggi ponderati calcolati, stampata come una classifica numerata.
- Messaggi esplicativi per guidare l'utente nelle interazioni.

## 2.3 Relazioni tra i dati

- Il punteggio ponderato di ogni canzone è calcolato moltiplicando il suo punteggio originale per il peso associato al suo genere, se presente tra i generi preferiti.
- L'ordinamento della classifica avviene in ordine decrescente rispetto ai punteggi ponderati.

## 3 Progettazione dell'algoritmo

### 3.1 Scelte di progetto

- In Haskell, i dati delle canzoni sono rappresentati tramite un tipo di dato **Canzone** per migliorare leggibilità e modularità.
- In Prolog, le canzoni sono definite come fatti dinamici per consentire la manipolazione dei dati durante l'esecuzione.

### 3.2 Passi dell'algoritmo

1. Caricamento dei dati da file.
2. Analisi e parsing delle informazioni delle canzoni.
3. Inserimento delle preferenze dell'utente.
4. Calcolo dei punteggi ponderati.
5. Ordinamento delle canzoni e generazione della classifica.

## 4 Implementazione dell'algoritmo

### 4.1 Implementazione in Haskell

Il programma in Haskell utilizza funzioni pure per elaborare i dati e azioni IO per interagire con l'utente.

Esempio di calcolo dei punteggi ponderati:

```
1 arricchisci :: [String] -> Double -> [Canzone] -> [(Double, Canzone)]
2 arricchisci _ _ [] = []
3 arricchisci generiPreferiti peso (c:cs) =
4     let genereMinuscolo = map toLower (genere c)
5         punteggioPonderato = if genereMinuscolo `elem` generiPreferiti
6                               then fromIntegral (punteggio c) * peso
7                               else fromIntegral (punteggio c)
8     in (punteggioPonderato, c) : arricchisci generiPreferiti peso cs
```

Listing 1: Calcolo dei punteggi ponderati in Haskell

### 4.2 Implementazione in Prolog

Il programma in Prolog sfrutta predicati per rappresentare i dati e regole logiche per elaborarli.

Esempio di ordinamento delle canzoni:

```
1 classifica_ordinata(Ordinata) :-
2     findall(Punteggio-Titolo, punteggio_ponderato(Titolo, Punteggio),
3         Punteggi),
4     sort(1, @>=, Punteggi, Ordinata).
```

Listing 2: Ordinamento delle canzoni in Prolog

## 5 Testing

### 5.1 Esempi di input e output

Input di esempio (file `canzoni.txt`):

```
Despacito,Luis Fonsi,Reggaeton,8
Danza Kuduro,Don Omar,Reggaeton,9
Bachata Rosa,Juan Luis Guerra,Bachata,9
```

Output per generi preferiti **Reggaeton** con peso 1.5:

```
#1 - Danza Kuduro
  Artista: Don Omar
  Genere: Reggaeton
  Punteggio ponderato: 13.5
#2 - Despacito
  Artista: Luis Fonsi
  Genere: Reggaeton
  Punteggio ponderato: 12.0
```

### 5.2 Verifica dei risultati e limitazioni

Il programma funziona correttamente per file ben formattati. Estensioni future potrebbero includere:

- Validazione dei dati in ingresso.
- Supporto per più criteri di raccomandazione (es. anno, durata).