

Emotional Songs

Manuale tecnico

**Versione 2.0 26/06/2023**

Università degli Studi dell’Insubria – Laurea Triennale in Informatica Progetto Laboratorio B: Emotional Songs

Sviluppato da: Erba Lorenzo, matricola 748702, Cacciarino Matteo, matricola 748231, Ferialdo Elezi, matricola 749721, Zancanella Alessandro, matricola 751494

Versione: 2.0 Emotional Songs – Manuale Utente

**Indice**

1. [Introduzione 3](#_bookmark0)
   1. [Struttura generale delle classi 3](#_bookmark1)
2. [Main Classes](#_bookmark2) 5
   1. [Canzoni](#_bookmark3) 5
   2. [Emozioni](#_bookmark4) 5
   3. [EmozioniCanzone 6](#_EmozioniCanzone)
   4. [InterfacciaServizio](#_InterfacciaServizio) 7
   5. [MediaEmozioni](#_MediaEmozioni) 7
   6. [Playlist 8](#_Playlist)
   7. [UtentiRegistrati 9](#_UtentiRegistrati)
3. [GUI Classes](#_bookmark11) 10
4. [Exception Classes 11](#_Exception_Classes)

Versione: 2.0 Emotional Songs – Manuale Utente



# Introduzione

Emotional Songs un applicativo sviluppato nell’ambito del progetto di Laboratorio B per il corso di Informatica dell’Università degli Studi dell’Insubria. Il progetto è sviluppato in Java 16.0.1, utilizza un’interfaccia grafica basata su JavaSwing ed è stato effettuato il testing sul sistema operativo Windows 10 e 11. Per quanto riguarda la gestione dei dati il software si appoggia al DBMS PostGreSQL mentre il DB viene hostato da AWS.

## Struttura generale delle classi

### Il progetto si basa su 3 principali macrogruppi di classi, le main classes, le gui classes e le exception classes. Il primo gruppo fa riferimento a tutte le classi utilizzate per l’elaborazione dati back-end, come lettura e scrittura su database, mentre il secondo gruppo gestisce l’interazione front-end tra l’utente e la parte grafica dell’applicativo. Il terzo gruppo viene utilizzato per gestire in maniera user-friendly le eccezioni sollevate durante l’esecuzione dell’applicativo.

### Main classes

### Canzoni

### Emozioni

### EmozioniCanzone

### InterfacciaServizio

### MediaEmozioni

### Playlist

### UtentiRegistrati

### Gui classes

### EmotionalSongs

### emotionsSummary\_Gui

### GUI

### Gui\_CreaPlaylist

### Gui\_ElencoBrani

### GUI\_login

### Gui\_Playlist

### Gui\_VisualizzaPlaylist

### notes\_gui

### PannelloEmozioni

### repositoryChoice\_Gui

### ConsoleFrame

### ServerFrame

Pag. **3** a **10**

Pag. **3** a **10**

### 

### Exception classes

### CanzoneEsistente

### ChiaveDuplicata

### DatiNonValidi

### EmozioniInesistenti

### MyServerException

### PasswordErrata

### PlaylistInesistenti

### UtenteInesistente

Pag. **4** a **10**

Pag. **3** a **10**



# Main Classes

## Canzoni

La classe Canzoni permette la gestione di oggetti rappresentanti il brano presente nel database. Per tale motivo la classe Canzone prevede i seguenti attributi:

* Titolo (String), rappresentante il titolo della canzone.
* Autore (String), rappresentante l’autore della canzone.
* Anno (String), rappresentante l’anno di uscita del brano

**Complessità stimate e strategie progettuali**

La classe Canzoni è composta da soli metodi getter and setter i quali hanno complessità O(1). Durante la progettazione della classe Canzoni si è deciso di non tenere traccia del genere, dell’album e della durata in quanto non sono stati richiesti come criteri di filtraggio durante la ricerca delle canzoni.

* 1. **Emozioni**

La classe Emozioni permette la gestione di oggetti di tipo EmozioniCanzone e quindi di tenere traccia delle emozioni inserite dall’utente. Per tale motivo la classe Emozioni prevede i seguenti attributi:

* EmozioniCanzoni (ArrayList<EmozioniCanzone>), rappresentante la lista di oggetti di tipo EmozioniCanzone.

o Media, oggetto di tipo MediaEmozioni, contenente la media delle emozioni.

**Complessità stimate e strategie progettuali**

La classe Emozioni è composta da soli metodi getter e il metodo per inserire le emozioni i quali hanno complessità O(1). Durante la progettazione della classe Emozioni si è deciso di non implementare metodi setter in quanto non necessari per la realizzazione del progetto.

Pag. **5** a **10**



## EmozioniCanzone

La classe EmozioniCanzone permette la gestione delle valutazioni delle emozioni da parte di utenti su brani. Per tale motivo la classe EmozioniCanzone possiede i seguenti attributi:

* idValutazione(int), rappresentante l’identificatore della valutazione.
* Titolo (String), rappresentante il nome del brano.
* Autore (String), rappresentante il nome dell’autore del brano.
* Anno(int), rappresentante l’anno di pubblicazione del brano.
* Amazement (int), rappresentante la valutazione dell’emozione “Amazement”.
* Solemnity (int), rappresentante la valutazione dell’emozione “Solemnity”.
* Tenderness (int), rappresentante la valutazione dell’emozione “Tenderness”.
* Nostalgia (int), rappresentante la valutazione dell’emozione “Nostalgia.
* Calmness (int), rappresentante la valutazione dell’emozione “Calmness”.
* Power (int), rappresentante la valutazione dell’emozione “Power.
* Joy (int), rappresentante la valutazione dell’emozione “Joy”.
* Tension (int), rappresentante la valutazione dell’emozione “Tension”.
* Sadness (int), rappresentante la valutazione dell’emozione “Sadness”.
* Amazement\_notes (String), rappresentante le note dell’emozione “Amazement”.
* Solemnity\_notes (String), rappresentante le note dell’emozione “Solemnity”.
* Tenderness\_notes (String), rappresentante le note dell’emozione “Tenderness”.
* Nostalgia\_notes (String), rappresentante le note dell’emozione “Nostalgia.
* Calmness\_notes (String), rappresentante le note dell’emozione “Calmness”.
* Power\_notes (String), rappresentante le note dell’emozione “Power.
* Joy\_notes (String), rappresentante le note dell’emozione “Joy”.
* Tension\_notes (String), rappresentante le note dell’emozione “Tension”.
* Sadness\_notes (String), rappresentante le note dell’emozione “Sadness”.

**Complessità stimate e strategie progettuali**

La classe EmozioniCanzone è composta da metodi getter e setter i quali hanno complessità O(1).

Pag. 6a **10**



## 

## InterfacciaServizio

La classe InterfacciaServizio rappresenta l’interfaccia per la gestione delle operazioni remote, per questo motivo conterra’ solo il prototipo dei metodi da implementare e nessun attributo.

**Complessità stimate e strategie progettuali**

La classe InterfacciaServizio possiede i seguenti metodi:

* + Login
  + Registrazione
  + FiltraPerTitolo
  + FiltraPerAutoreAnno
  + GetEmozioniFromBrano
  + GetCanzoniForPlaylist
  + CreatePlaylist
  + GetPlaylist
  + GetCanzoniFromPlaylist
  + InserisciEmozione

Per maggiori informazioni sulla complessita’ e sullo scopo dei metodi consultare la classe ConsoleFrame che implementa InterfacciaServizio.

## MediaEmozioni

La classe MediaEmozioni permette la gestione delle medie delle valutazioni delle emozioni. Per tale motivo la classe MediaEmozioni possiede i seguenti attributi:

* Avg\_Amazement (int), rappresentante la media dell’emozione “Amazement”.
* Avg\_Solemnity (int), rappresentante la media dell’emozione “Solemnity”.
* Avg\_Tenderness (int), rappresentante la media dell’emozione “Tenderness”.
* Avg\_Nostalgia (int), rappresentante la media dell’emozione “Nostalgia.
* Avg\_Calmness (int), rappresentante la media dell’emozione “Calmness”.
* Avg\_Power (int), rappresentante la media dell’emozione “Power.
* Avg\_Joy (int), rappresentante la media dell’emozione “Joy”.
* Avg\_Tension (int), rappresentante la media dell’emozione “Tension”.
* Avg\_Sadness (int), rappresentante la media dell’emozione “Sadness”.

Pag. **7** a **10**



**Complessità stimate e strategie progettuali**

La classe EmozioniCanzone è composta da metodi getter i quali hanno complessità O(1).

## Playlist

La classe Playlist permette la gestione di oggetti rappresentanti la playlist dell’utente. Per tale motivo la classe Playlist presenta i seguenti attributi:

* idPlaylist (int), rappresentante l’identificatore della playlist.
* Nomeplaylist (String), rappresentante il nome della playlist.
* CodiceFiscale (String), rappresentante il codice fiscale del proprietario della playlist.

**Complessità stimate e strategie progettuali**

La classe Playlist è composta da metodi getter e setter con complessità O(1).

Pag. **8** a **10**



## UtentiRegistrati

La classe UtentiRegistrati permette la gestione di oggetti rappresentanti l’utente che ha effettuato la registrazione. Per tale motivo la classe Utente presenta i seguenti attributi:

* Nome (String), rappresentante il nome dell’utente.
* Cognome (String), rappresentante il cognome dell’utente.
* CodiceFiscale (String), rappresentante il codice fiscale dell’utente.
* Citta (String), rappresentante la citta’ di residenza dell’utente.
* Cap (int), rappresentante il cap della citta’.
* Via (String), rappresentante la via di residenza dell’utente.
* Civico (int), rappresentante il civico di residenza dell’utente.
* Email (String), rappresentante l’email dell’utente.
* Password (String), rappresentante la password dell’utente.

**Complessità stimate e strategie progettuali**

La classe Utente è composta da metodi con complessità O(1). Durante la progettazione della classe Utente si è deciso di utilizzare il codice fiscale come identificatore univoco dell’utente. Inoltre si è deciso di non impostare particolari requisiti per la creazione della password.

Pag. **9** a **10**



# GUI Classes

### Per la realizzazione dell’interfaccia grafica è stato utilizzato il linguaggio Java\_Swing, una particolare estensione del linguaggio Java che fa utilizzo di costrutti grafici per la realizzazione di progetti con grafica utente. Fra i vari oggetti grafici utilizzati troviamo:

* JButton
* JPanel
* JFrame
* JLabel
* JTextField
* JTable
* JTextPane
* JTabbedPane
* JOptionPane

Pag. **10** a **12**



# Exception Classes

Le seguenti classi rappresentano le eccezioni lanciate all’interno del programma.

# CanzoneInesistente

# Classe rappresentante l'eccezzione CanzoneInesistente sollevata qual'ora la canzone ricercata non fosse presente nella base di dati.

# ChiaveDuplicata

# Classe rappresentante l'eccezzione ChiaveDuplicata sollevata qual ora la chiave inserita sia duplicata.

# DatiNonValidi

# Classe rappresentante l'eccezzione DatiNonValidi sollevata qual'ora i dati inseriti non rispettano i vincoli di integrità.

# EmozioniInesistenti

# Classe rappresentante l'eccezzione EmozioniInesistenti sollevata qual'ora l'emozione di una canzone da ricercare non è presente.

# MyServerException

# Classe rappresentante l'eccezzione MyServerException sollevata qual'ora si verifichi una SQLException.

# PasswordErrata

# Classe rappresentante l'eccezzione PasswordErrata sollevata qual'ora la password usata per la fase di login sia errata.

# PlaylistInesistenti

# Classe rappresentante l'eccezzione PlaylistInesistenti sollevata qual'ora l'utente non abbia nessuna playlist associata al suo account.

Pag. **11** a **12**

# 

# UtenteInesistente

# Classe rappresentante l'eccezzione UtenteInesistente sollevata qual'ora l'utente provi a registrarsi, essendo precedentemente già registrato al sistema.

Pag. **12** a **12**