

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

Indice

1. Introduzione	3
1.1 Struttura generale delle classi.....	3
2. Main Classes	5
2.1 Canzoni.....	5
2.2 Emozioni	5
2.3 EmozioniCanzone.....	6
2.4 InterfacciaServizio.....	7
2.5 MediaEmozioni	7
2.6 Playlist	8
2.7 UtentiRegistrati	9
3. GUI Classes	10
4. Exception Classes	11

1. 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.

1.1 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

- Exception classes
 - CanzoneEsistente
 - ChiaveDuplicata
 - DatiNonValidi
 - EmozioniInesistenti
 - MyServerException
 - PasswordErrata
 - PlaylistInesistenti
 - UtenteInesistente

2. Main Classes

2.1 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.

2.2 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.
- 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 metodisetter in quanto non necessari per la realizzazione del progetto.

2.3 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 $O(1)$.

2.4 InterfacciaServizio

La classe InterfacciaServizio rappresenta l'interfaccia per la gestione delle operazioni remote, per questo motivo conterrà 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
- GetEmozioniFromBranco
- GetCanzoniForPlaylist
- CreatePlaylist
- GetPlaylist
- GetCanzoniFromPlaylist
- InserisciEmozione

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

2.5 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".

Complessità stimate e strategie progettuali

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

2.6 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 $O(1)$.

2.7 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.

3. 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

4. Exception Classes

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

4.1 CanzoneInesistente

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

4.2 ChiaveDuplicata

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

4.3 DatiNonValidi

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

4.4 EmozioniInesistenti

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

4.5 MyServerException

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

4.6 PasswordErrata

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

4.7 PlaylistInesistenti

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

4.8 UtenteInesistente

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