

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](#_bookmark0) 
   1. [Struttura generale delle classi](#_bookmark1)
2. [Main Classes](#_bookmark2) 
   1. [Canzoni](#_bookmark3)
   2. [Emozioni](#_bookmark4)
   3. [EmozioniCanzone](#_EmozioniCanzone)
   4. [InterfacciaServizio](#_InterfacciaServizio)
   5. [MediaEmozioni](#_MediaEmozioni)
   6. [Playlist](#_Playlist)
   7. [UtentiRegistrati](#_UtentiRegistrati)
3. [GUI Classes](#_bookmark11)
4. [Exception Classes](#_Exception_Classes)
5. [Documentazione del database ……………………………………………………….....](#_Documentazione_del_database)
   1. Analisi del Sistema……………………………………………………………………
   2. Schema ER……………………………………………………………………………
   3. Analisi Entità e Associazioni dello schema relazionale…………………………..
   4. Schema logico…………………………………………………………………………
   5. Analisi schema logico…………………………………………………………………
   6. Query significative……………………………………………………………………..

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

### 

### Exception classes

### CanzoneEsistente

### ChiaveDuplicata

### DatiNonValidi

### EmozioniInesistenti

### MyServerException

### PasswordErrata

### PlaylistInesistenti

### UtenteInesistente



# 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.



## 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).



## 

## 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”.



**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).



## 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.



# 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



# 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.

# 

# UtenteInesistente

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

# Documentazione del database

1. Analisi del sistema
   1. Obiettivi e scopo del prodotto

Il progetto descritto in questo documento ha come scopo la realizzazione di un sistema di annotazione di emozioni percepite durante l’ascolto di brani musicali. L’utente deve avere la possibilità di registrarsi al sistema, effettuare l’accesso al sistema tramite il quale sarà possibile:

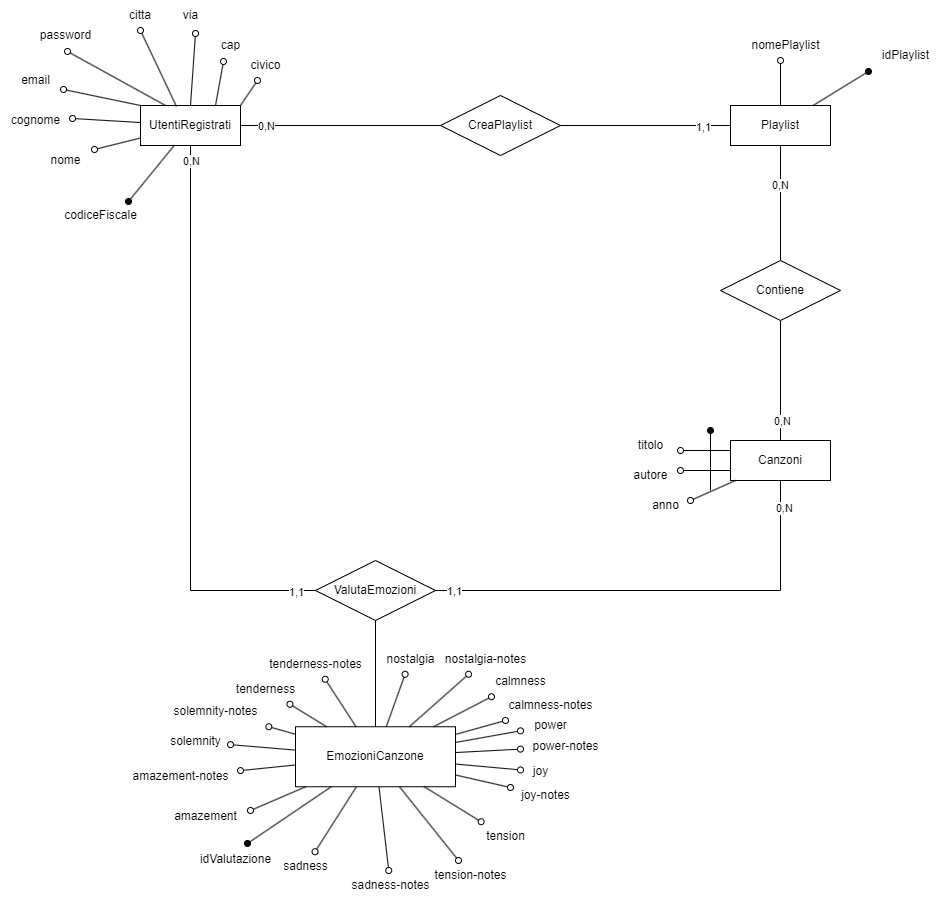
* Creare playlist.
* Inserire canzoni all’interno delle playlist.
* Filtrare le canzoni per titolo
* Filtrare le canzoni per autore e anno
* Inserire un punteggio relativo alle emozioni.
* Inserire note opzionali relative alle emozioni.

Il sistema dev’essere realizzato in modo tale da poter supportare l’esecuzione concorrente in ambito distribuito.

1. Schema ER

2.1 Metodologia di sviluppo dello schema ER

Lo sviluppo dello schema ER ha seguito un approccio top-down, dovuto soprattutto dal file con le richieste e le specifiche di sistema, il quale aumenta la propria specificità in seguito a letture successive.

**2.2** 

1. Analisi Entità e Associazioni dello schema relazionale
   1. Dizionario Entità

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Descrizione | Attributi | Identificatori |
| Canzoni | Canzoni utilizzate per l’ascolto da parte dell’utente. | Titolo, Autore, Anno | Titolo, Autore, Anno |
| UtentiRegistrati | Utenti che usufriscono dei servizi offerti dal sistema. | Nome, Cognome,  CodiceFiscale, Email, Password, cap, citta, civico, via | CodiceFiscale |
| Playlist | Playlist contenenti le canzoni inserite dall’utente. | idPlaylist, NomePlaylist | idPlaylist |
| EmozioniCanzoni | Emozioni suscitate dall’utente dopo l’ascolto di un brano specifico. | idValutazione, Amazement,  Amazement-Notes, Solemnity, Solemnity-Notes, Tenderness, Tenderness Notes, Nostalgia,  Nostalgia-Notes, Calmness,  Calmness-Notes, Power, Power-Notes, Joy, Joy-Notes, Tension, Tension-Notes, Sadness, Sadness-Notes. | idValutazione |



* 1. Dizionario delle Associazioni

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Descrizione | Attributi | Entità collegate |
| CreaPlaylist | Rappresenta la creazione della playlist da parte dell’utente. |  | UtentiRegistrati, Playlist |
| Contiene | Rappresenta l’inserimento delle canzoni nella playlist da parte dell’utente. |  | Playlist, Canzoni |
| ValutaEmozioni | Rappresenta la valutazione da parte dell’utente delle canzoni ascoltate. |  | UtentiRegistrati, Canzoni,  EmozioniCanzone |

* 1. Vincoli di integrità dello schema relazionale
     1. Vincoli di integrità di chiave primaria:
        + CodiceFiscale chiave primaria per l’entità UtentiRegistrati
        + IdPlaylist chiave primaria per l’entità Playlist
        + Titolo, Autore, Anno chiave primaria composta per l’entità Canzoni
        + IdValutazione chiave primaria per l’entità EmozioniCanzoni
     2. Vincoli di integrità referenziale
        + - Una playlist non può esistere senza una canzone associata.
          - Una playlist non può esistere senza un utente associato.
          - Una valutazione delle emozioni non può esistere senza un brano associato.
          - Una valutazione delle emozioni non può esistere senza un utente associato.
     3. Vincoli di integrità unique
* vincolo di integrità unique sulla tabella Playlist per cui un utente non può avere due playlist con stesso nome.



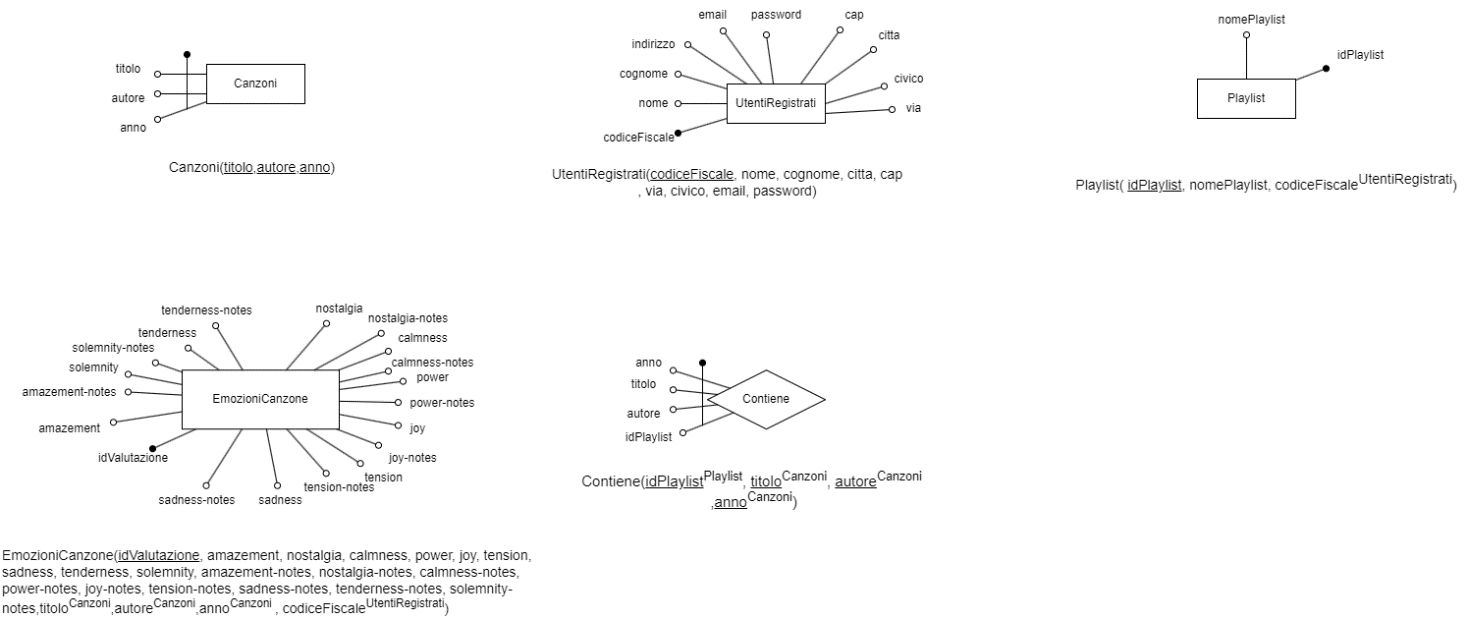
* 1. Motivazioni decisionali della progettazione dello schema concettuale

Durante la progettazione dello schema E/R si sono prese una serie di scelte giustificate dalle seguenti motivazioni:

* + L’entità Canzoni possiede un identificatore composto da 3 attributi, Titolo, Autore e Anno, in quanto è possibile che differenti autori pubblichino canzoni con lo stesso titolo ma anche che lo stesso autore pubblichi la stessa canzone in anni differenti (e.g. remix); non è però possibile che lo stesso autore pubblichi due canzoni identiche nello stesso anno.
  + L’entità UtentiRegistrati possiede come identificatore codiceFiscale e non email in quanto il codice fiscale rappresenta un’informazione maggiormente valida ed efficiente in termini di recupero di dati anagrafici dell’utente.
  + L’entità Playlist possiede come identificatore un identificativo generato automaticamente dal DBMS in quanto è consentito allo stesso utente di possedere due playlist omonime.
  + L’entità EmozioniCanzone possiede come identificatore un identificativo generato automaticamente dal DBMS, in modo da permettere, da parte dello stesso utente, di rilasciare più recensioni sulla stessa canzone.

1. Schema logico

4.1 Schema logico



1. Analisi schema logico
   1. Fase di ristrutturazione

La fase di ristrutturazione dello schema E/R ha prodotto il medesimo schema concettuale a causa dell’assenza di gerarchie di generalizzazione, attributi composti e attributi multi valore.

* 1. Fase di traduzione
  + Traduzione dell’entità Playlist: Playlist( idPlaylist, nomePlaylist codiceFiscaleUtentiRegistrati)
  + Traduzione dell’entità Canzoni: Canzoni(titolo, autore,anno)
  + Traduzione dell’entità UtentiRegistrati: UtentiRegistrati(codiceFiscale, nome, cognome, citta, cap, via, civico , email, password)
  + Traduzione dell’entità EmozioniCanzone: EmozioniCanzone(idValutazione, amazement, nostalgia, calmness, power, joy, tension, sadness, tenderness, solemnity, amazement-notes, nostalgia-notes, calmness-notes, power-notes, joy-notes, tension-notes, sadness-notes, tenderness-notes, solemnity-notes,titoloCanzoni,autoreCanzoni,annoCanzoni,codiceFiscaleUtentiRegistrati)



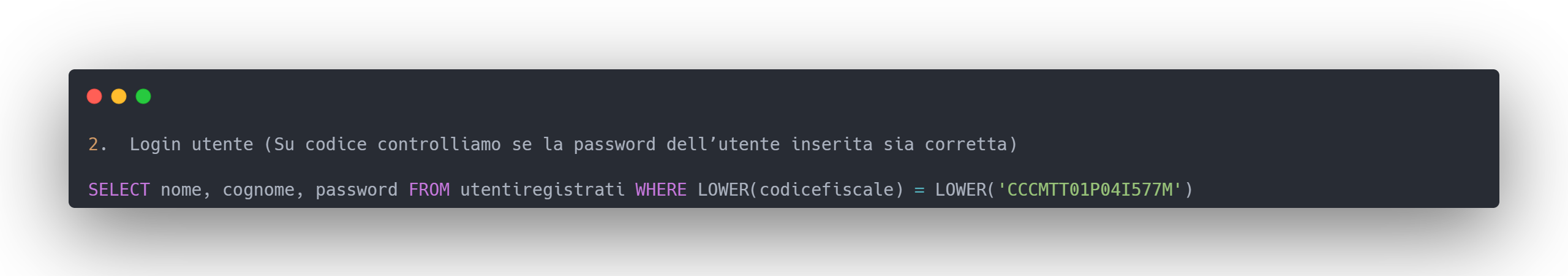
* 1. Motivazioni decisionali della traduzione dello schema concettuale

Non vi sono motivazioni legate alla fase di traduzione dello schema concettuale in quanto lo schema E/R fornito in input non presenta ambiguità.

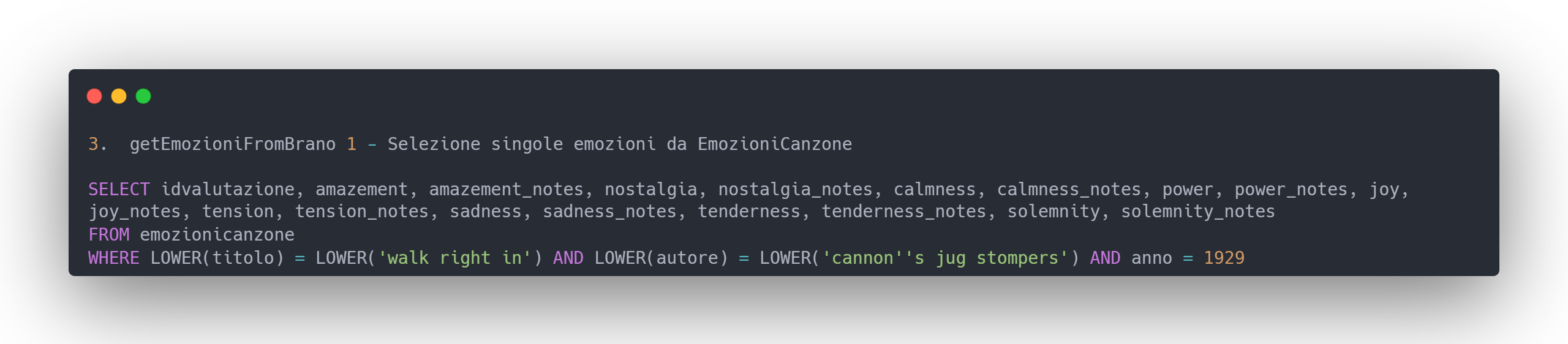
1. Query significative
   1. Registrazione utente

****

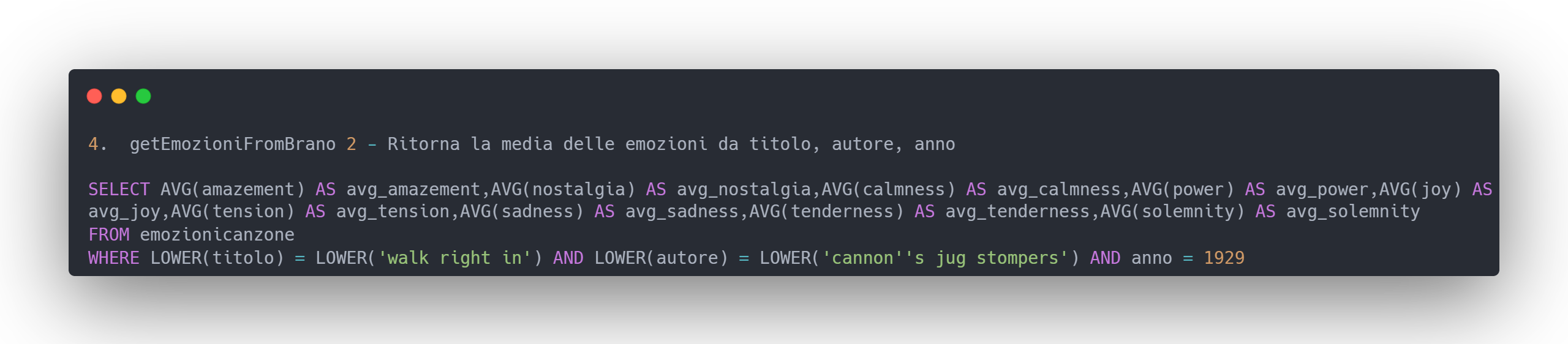
* 1. Login utente



* 1. getEmozioniFromBrano 1 – Restituisce le emozioni associate al brano specificato da un utente



* 1. getEmozioniFromBrano 2 – Restituisce le emozioni associate al brano specificato da un utente





* 1. Filtra per titolo – Restituisce le canzoni filtrate per il titolo specificato

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* 1. Filtra per autore anno – Restituisce le canzoni filtrate per autore e anno specificato

Immagine che contiene testo, schermata, Carattere, Software multimediale

Descrizione generata automaticamente

* 1. getCanzoniForPlaylist – Restituisce tutte le canzoni presenti nel database

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* 1. createPlaylist 1 - Creazione della playlist richiesta dall’utente

Immagine che contiene testo, schermata, Carattere

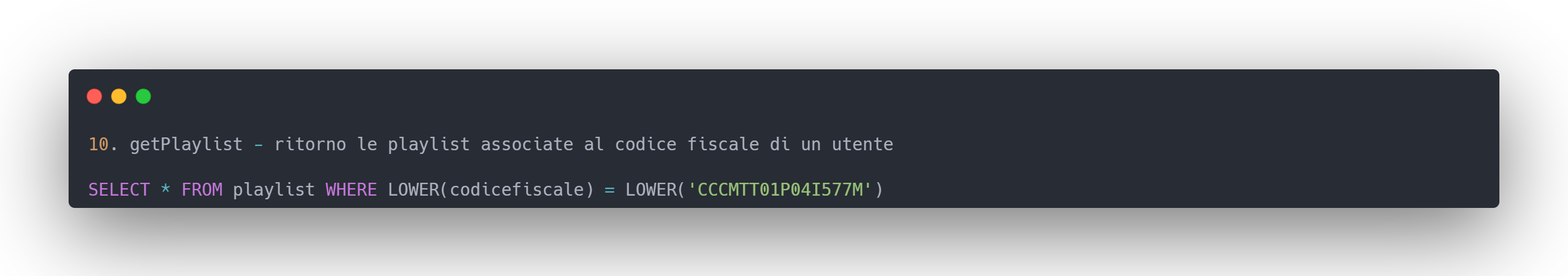
Descrizione generata automaticamente

* 1. createPlaylist 2 - Inserimento delle canzoni all'interno della playlist tramite inserimento nella relazione “contiene” Immagine che contiene testo, schermata, Carattere

     Descrizione generata automaticamente



* 1. getPlaylist - Restituisce le playlist associate al codice fiscale di un utente



* 1. getCanzoniFromPlaylist – Restituisce le canzoni da inserire nella playlist

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* 1. inserisciEmozione - Inserimento emozioni

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente