Emotional Songs

Manuale Tecnico

Versione 2.0

Università degli Studi dell'Insubria – Laurea Triennale in Informatica

Progetto Laboratorio B: Emotional Songs.

Sviluppato da:

-Ashley Chudory, matricola 746423

-Bogdan Tsitsiurskyi, matricola 748685

-Maria Vittoria Ratti, matricola 748825

-Miguel Alfredo Valerio Aquino, matricola 748704

Indice

1. Requisiti del sistema

- Si intende sviluppare un sistema client-server per la gestione delle annotazioni delle emozioni, mediante l'uso di una scala valutativa da 1 a 5 per ciascuna emozione individuale. L'obiettivo è creare un'applicazione che consenta agli utenti di compilare annotazioni emozionali per singoli brani e di organizzarli in playlist personalizzate.

- L'applicazione sarà in grado di supportare la creazione di una o più playlist, all'interno delle quali gli utenti potranno per singolo brano fornire valutazioni emozionali individuali, basate su una scala di 9 stati emozionali standard. Questi stati emozionali includeranno: Amazement, Solemnity, Tenderness, Nostalgia, Calmness, Power, Joy, Tension,Sadness.

- Inoltre, l'applicazione dovrà fornire una panoramica aggregata delle etichette emozionali assegnate dagli utenti per ciascun brano all'interno delle proprie playlist. Questo resoconto riassuntivo mostrerà le emozioni medie per ogni stato emozionale specifico, consentendo agli utenti di comprendere meglio le reazioni emozionali associate a ciascun brano.

- Per raggiungere tali obiettivi, il sistema dovrà implementare un'interfaccia utente intuitiva per la gestione delle playlist, visione dei brani e l'assegnazione delle valutazioni emozionali. Inoltre, sarà necessario un backend per raccogliere, conservare e analizzare le informazioni emozionali assegnate dagli utenti, al fine di generare i resoconti aggregati desiderati.

- Il risultato finale sarà un'applicazione che consentirà agli utenti di esprimere e monitorare le proprie risposte emozionali fornendo al contempo una visione complessiva delle reazioni emozionali di tutti gli utenti verso i singoli brani all'interno delle playlist.

2. Suddivisione del lavoro

- La parte client è stata sviluppata da Ashley Chudory e Bogdan Tsitsiurskiy mentre la parte del server e DBCreator è stata sviluppata da Maria Vittoria Ratti e Miguel Alfredo Valerio Aquino.

3. Architettura del sistema

- L'applicazione è basata su un'architettura client-server, in cui il server espone metodi remoti attraverso RMI che i client possono chiamare per interagire con il sistema. RMI facilita la comunicazione tra oggetti distribuiti su macchine diverse, consentendo ai metodi di essere chiamati come se fossero metodi locali.

Componenti principali

Client

- Il client è un'applicazione Java che richiede servizi al server e fornisce l'interfaccia grafica per l'utente. Esso è responsabile dell'iniziazione delle chiamate remote ai metodi esposti dal server.

Server

- Il server è anch'esso un'applicazione Java, ma con il ruolo opposto. Espone metodi remoti che i client possono richiamare e interagisce con il database.

EmotSongsInterface

- Definisce l'interfaccia dei metodi remoti che il server espone.Il client utilizzerà questa interfaccia per chiamare i metodi remoti del server.

Database

- Il database è il sistema di archiviazione dati che memorizza e gestisce le informazioni dell'applicazione. Il server comunica con il database per recuperare o aggiornare i dati in base alle richieste dei client tramite la libreria JDBC (Java Database Connectivity)

4. Prerequisiti macchina

- Sistema operativo:
  - Windows
  - Linux
  - MacOS
- JVM
  - JDK 17 o versione successiva.

5. Linguaggio di Programmazione

- Java: è stato il linguaggio di programmazione scelto per l'intero progetto "Emotional Songs".
- SQL: Utilizzato per interagire con il database PostgreSQL all'interno del progetto server.

6. Librerie e framework

- JavaFX 17: Utilizzato per creare l'interfaccia grafica del Client e DBCreator.
- Abbiamo impiegato JavaFX Controls nei progetti client e DBCreator per creare un'interfaccia user-friendly con pulsanti, caselle di testo e tabelle interattive. L'uso del linguaggio FXML ha consentito di separare la logica dell'interfaccia dalla logica di programmazione.
- PostgreSQL JDBC Driver (postgresql-42.6.0.jar): Utilizzato per la connessione al database PostgreSQL.
- jBCrypt 0.4: Abbiamo integrato la libreria jBCrypt nel progetto per gestire l'hashing sicuro delle password degli utenti.

7. Strumenti e Tecnologie

- draw.io: Utilizzato per la creazione di vari tipi di diagrammi, tra cui quelli UML come diagrammi delle classi e di sequenza. Inoltre, per progettare in modo visuale lo schema Entità-Relazione (ER) del database "EmotionalSongs".
- Scene Builder: E’ stato impiegato per la progettazione dell'interfaccia grafica.
- pgAdmin 4: Abbiamo utilizzato pgAdmin 4 come strumento di amministrazione per il database PostgreSQL. Ci ha permesso di gestire in modo efficace il database "EmotionalSongs", compresi la creazione delle tabelle e l'esecuzione di query.
- Java.sql: package Java.sql per la gestione delle interazioni con il database PostgreSQL. Questo ci ha permesso di eseguire query, aggiornamenti e operazioni sul database in modo efficiente e sicuro.
- Java.rmi: Abbiamo implementato la tecnologia Java.rmi (Remote Method Invocation) per consentire la comunicazione tra il client e il server. Questo ha permesso al client di invocare metodi remoti sul server, facilitando lo scambio di dati e l'accesso alle funzionalità offerte dal server.

8. IDE e Strumento di Build

- IntelliJ IDEA: Utilizzato per lo sviluppo in tutti e tre progetti.
- Maven: Utilizzato per la build dei progetti e la gestione delle dipendenze. Maven semplifica il processo di compilazione e l'aggiunta di librerie esterne al progetto.

9. Descrizione Package

EmotionalSongs-Client

- Il package "Client" necessario per gli utenti intenzionati a fruire dell'applicazione. Esso racchiude una serie di elementi, tra cui i file destinati alla strutturazione dell'interfaccia utente, i quali sono caratterizzati dall'estensione "fxml". Inoltre, all'interno del package sono inclusi anche i controller necessari al controllo dei file fxml. In aggiunta, viene fornita l'interfaccia mediante la quale è possibile richiamare i metodi in modalità remota. Inoltre il package ospita ulteriori file Java indispensabili per assicurare il corretto funzionamento dell'applicazione.

EmotionalSongs-Server

- Il package “Server” è necessario per la configurazione del server. Contiene l’interfaccia in modo tale che gli utente connessi al server possono invocare i metodi da remoto e l’implementazione dei metodi dell’interfaccia stessa che permettono anche di stabilire la comunicazione con il database.

ESDBCreator

- Il package “ESDBCreator” è necessario per la creazione della base di dati. Composta da file per la creazione del database e file di controllo dell’interfaccia.

10. Scelte algoritmiche e progettuali

Singleton Pattern

Il design pattern Singleton è stato applicato all'interno del progetto "Emotional Songs" per garantire che esista un'unica istanza di determinate classi. Questo approccio è stato scelto per gestire in modo efficiente la connessione al server e la gestione del catalogo delle canzoni. Le classi coinvolte sono:

1. ConnessioneServer

- **Descrizione:** Questa classe gestisce la connessione al server RMI "EmotionalSongs" e fornisce un'istanza singleton della connessione per l'intero progetto.
- **Motivazione:** L'utilizzo del pattern Singleton garantisce che vi sia una sola istanza di ConnessioneServer durante l'esecuzione dell'applicazione, evitando connessioni multiple e incoerenze.
- **Implementazione:** La classe implementa un costruttore privato e un metodo statico getIstanza() per ottenere l'istanza unica.

2. SongManager

- **Descrizione:** Questo gestore delle canzoni utilizza il pattern Singleton per gestire l'elenco delle canzoni nella classe e comunicare con il server remoto.
- **Motivazione:** L'utilizzo del pattern Singleton garantisce che l'elenco delle canzoni sia gestito in modo coerente e che le operazioni di aggiunta e recupero siano effettuate su un'unica istanza.
- **Implementazione:** La classe implementa un metodo statico getIstanza() per ottenere l'istanza unica del gestore delle canzoni.

Algoritmo di Filtraggio delle Canzoni

Una delle funzionalità principali dell'applicazione "Emotional Songs" è la possibilità per gli utenti di cercare e filtrare rapidamente tra un vasto elenco di brani musicali direttamente dalla tabella di visualizzazione. Questa caratteristica è stata implementata utilizzando un sistema di filtraggio dinamico che permette all'utente di digitare del testo in un campo di ricerca, in modo da restringere l'elenco di canzoni visualizzate sulla tabella in tempo reale.

Implementazione

La classe TableUtil gestisce il processo di filtraggio e visualizzazione dei brani musicali all'interno della tabella. Il metodo cercaBranoMusicale di questa classe è responsabile per l'implementazione del filtraggio. Prende in input la tabella in cui visualizzare i brani musicali e i vari elementi grafici (TextField e TableColumn) che compongono la tabella.

Processo di Filtraggio Dinamico della Tabella Canzoni

1. **Recupero delle Canzoni:** Le canzoni vengono ottenute dal gestore delle canzoni (SongManager) utilizzando il metodo getAllSongs(). Queste canzoni vengono inizialmente memorizzate in una lista (canzoniOriginali) e convertite in un'ObservableList che verrà utilizzata come sorgente di dati per la tabella.

2. **Ordinamento delle Canzoni:** Una volta ottenuta l'ObservableList delle canzoni, questa viene ordinata in base all'ID del brano musicale.

3. **Creazione di una Lista Filtrata:** Viene creata una FilteredList a partire dall'ObservableList di canzoni. Questa lista filtrata serve come base su cui verranno applicati i criteri di ricerca.

4. **Aggiunta di un Listener:** Viene aggiunto un listener al campo di ricerca (textFieldNomeCanzone). Questo listener attiva il processo di filtraggio ogni volta che il testo nel campo di ricerca cambia.

5. **Impostazione del Predicato di Filtro:** All'interno del listener, viene impostato un predicato di filtro sulla FilteredList. Questo predicato confronta il testo di ricerca con il nome del brano, il nome dell'autore, l'autore e l'anno combinati, e l'anno di rilascio del brano.

6. **Aggiornamento della Tabella:** Infine, la FilteredList filtrata viene assegnata come sorgente di dati per la tabella. Ciò consente di mostrare solo le canzoni che soddisfano i criteri di ricerca dell'utente.

Gestione dei Dati e Coerenza dell'Elenco delle Canzoni

1. Caricamento Iniziale e Accesso Successivi

Durante la prima apertura della scena contenente l'elenco delle canzoni, l'ampio catalogo viene recuperato dal database “emotionalsongs”. Questo processo può richiedere un tempo iniziale di circa 5 secondi a causa del numero considerevole di dati da trasferire. Tuttavia, è importante evidenziare che questo ritardo si verifica solo durante il primo accesso.
Dopo il caricamento iniziale, l'elenco completo delle canzoni viene memorizzato nel "SongManager", consentendo accessi successivi notevolmente più veloci. Questo perché i dati vengono prelevati direttamente dalla memoria anziché essere recuperati nuovamente dal database remoto.

2. Garanzia di Coerenza e Aggiornamento Delle Informazioni

L'elenco delle canzoni memorizzato nel "SongManager" garantisce coerenza e integrità dei dati finché non si verifica un'operazione di modifica nella tabella delle canzoni all'interno del database “emotionalsongs”. Queste operazioni di modifica includono inserimenti, aggiornamenti o eliminazioni di singole canzoni.
Quando si verifica una modifica nella tabella delle canzoni del database, il "SongManager" rileva automaticamente il cambiamento e aggiorna l'elenco delle canzoni memorizzato. In questo modo, quando l'utente accede alla scena dell'elenco delle canzoni in future occasioni, viene presentata la versione più aggiornata e accurata dell'elenco delle canzoni.

Gestione dei Tag Emozionali

Questa sezione si concentra sulla possibilità per gli utenti di assegnare valutazioni emotive a una canzone selezionata, suddivisa in 9 emozioni. Gli utenti possono valutare ogni emozione utilizzando una scala da 1 a 5 tramite ComboBox e fornire anche commenti opzionali.

Classe ListaEmozioniController

Questa classe gestisce le funzionalità legate all'etichettatura emotiva e alla valutazione delle canzoni. Inizializza gli elementi dell'interfaccia e interagisce con il server per gestire emozioni, punteggi e commenti. Contiene i seguenti componenti chiave:

- **initialize:** Questo metodo inizializza il controller e gli elementi dell'interfaccia. Recupera i dati dal server e configura di conseguenza l'interfaccia utente.
- **inizializzaTable:** Questo metodo popola la tabella delle emozioni con dati predefiniti per ciascuna delle 9 emozioni. Configura le colonne per la categoria emotiva, la spiegazione, la selezione del punteggio (ComboBox) e l'inserimento del commento (TextArea).
- **inserisciEmozioniBrano:** Questo metodo viene attivato quando si fa clic sul pulsante "Salva". Elabora le valutazioni delle emozioni selezionate e i commenti opzionali, inviando i dati al server per l'archiviazione.
- **resetEmozione:** Questo metodo ripristina la tabella delle emozioni al suo stato iniziale, rimuovendo eventuali punteggi selezionati e commenti.
- **rimuoviEmotions:** Questo metodo rimuove le valutazioni emotive e i commenti associati alla canzone corrente per l'utente connesso.
- **tornaCanzoni:** Questo metodo naviga alla scena dell'elenco delle canzoni.

Calcolo delle Statistiche

L'applicazione Emotional Songs implementa un sistema per calcolare in modo accurato le statistiche inerenti alle emozioni associate alle singole canzoni. Questo processo di calcolo è eseguito mediante una serie di passaggi all'interno della classe StatsEmozioneBranoController, il controller responsabile della gestione delle statistiche delle emozioni per il brano selezionato. Queste statistiche includono la media dei punteggi emozionali, il numero di utenti che hanno assegnato almeno un tag emozionale e i commenti associati a ciascuna emozione.

L'algoritmo implementato per calcolare le statistiche delle emozioni si articola nei seguenti passaggi:

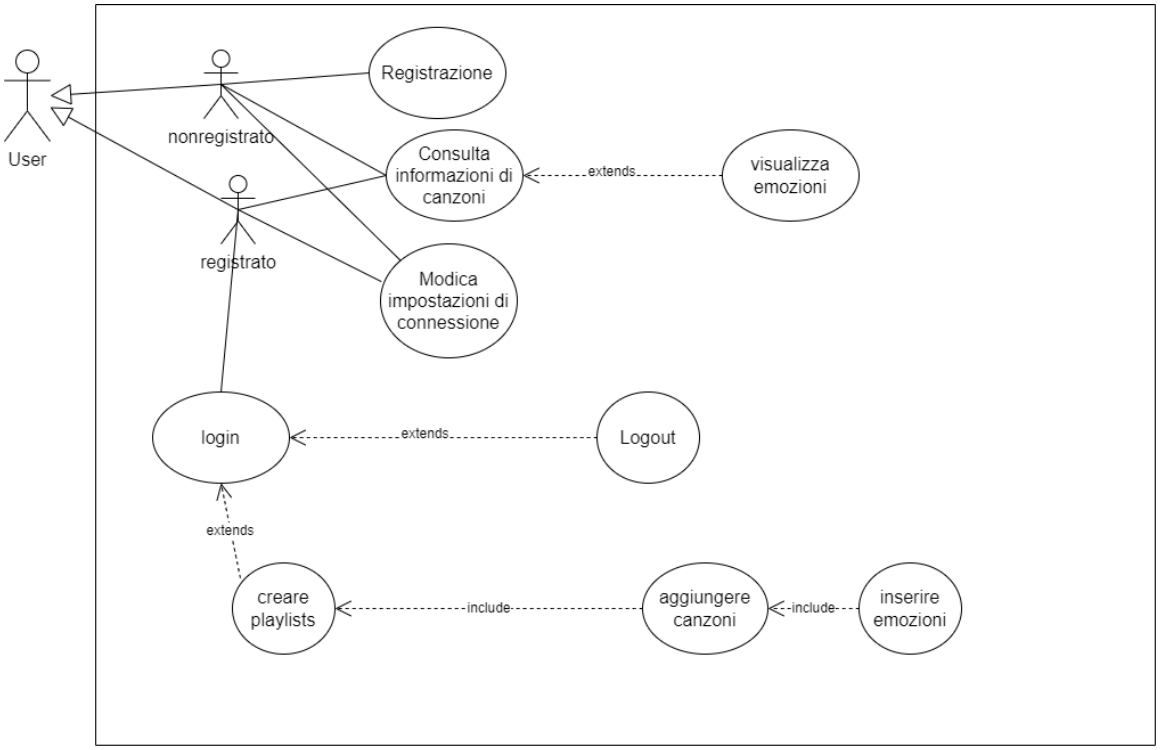
1. **Richiesta dei Dati dal Server (metodo richiediDati()):** Inizialmente, viene effettuata una richiesta al server tramite l'oggetto connectServer per acquisire i dati relativi alle emozioni associate alla canzone in esame. Questi dati includono i

punteggi totali delle varie emozioni, i punteggi assegnati da ciascun utente, il numero di utenti che hanno associato un determinato tag emozionale e i commenti degli utenti correlati alle emozioni. Tali informazioni costituiscono la base per il calcolo delle statistiche.

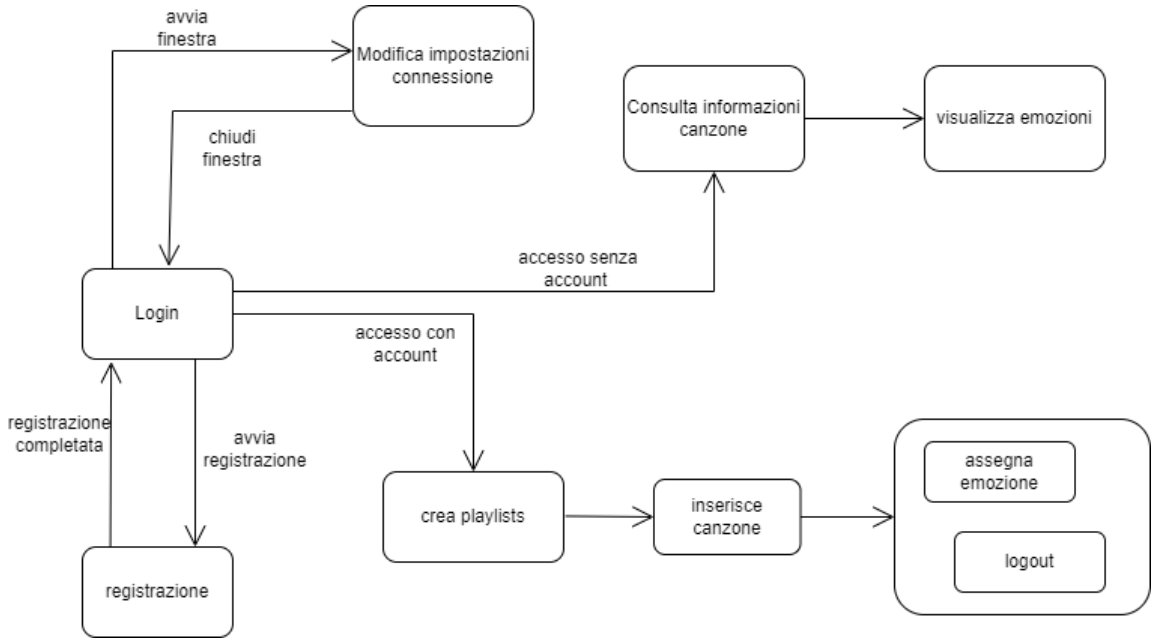
2. **Calcolo delle Medie delle Emozioni(metodo calcolaMediaEmozioni()):** Utilizzando i dati ottenuti dalla richiesta al server, il controller avvia il calcolo della media dei punteggi emozionali per ciascuna tipologia di emozione. Questo calcolo avviene attraverso la divisione del punteggio totale di un'emozione per il numero di utenti che hanno assegnato tale tag emozionale. Nel caso in cui il numero di utenti per una specifica emozione sia pari a zero, la relativa media viene impostata a zero per evitare divisioni per zero e riflette la mancanza di dati.

3. **Popolazione della Tabella e del Grafico (metodo popolaEmotionsStatsTable()):** Una volta ottenute le medie dei punteggi emozionali per ciascuna emozione, i risultati vengono utilizzati per popolare in modo dinamico la tabella delle statistiche e il grafico a barre delle emozioni. La tabella delle statistiche visualizza, per ogni riga, il nome dell'emozione, la sua media dei punteggi, il numero di utenti che l'hanno valutata e i commenti ad essa associati. Il grafico a barre, invece, rappresenta graficamente le emozioni come barre verticali, dove l'altezza di ciascuna barra rappresenta la media dei punteggi emozionali calcolata in precedenza.

11. Use Case Diagram



12. Diagramma Transizione Finestre

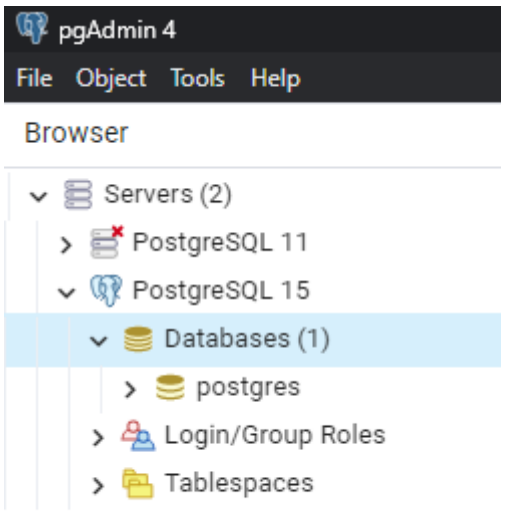


13. Configurazione ESDBCreator e Server

Questa configurazione è necessaria alla creazione della base dati “emotionalsongs” e all’avvio del Server.

Passo 1:

Prima di tutto bisogna aver installato pgAdmin preferibilmente la versione 6 visto che l’ultima versione in alcuni casi da problemi all’avvio, oltre a pgAdmin bisogna installare PostgreSQL, in questo caso si può scaricare anche l’ultima versione cioè la 15.La situazione che vi troverete su pgAdmin è la seguente:



Ovviamente una volta fatto partire il server PostgreSQL non bisogna chiuderlo.

Passo 2:

Per non avere problemi di permessi bisogna dare il seguente comando da terminale

- da Windows (sostituire "/percorso/del/file/" con il percorso completo del file csv)
  icacls "/percorso/del/file/canzoni.csv" /grant Everyone:(RX)

- da Linux e MacOS (sostituire "/percorso/del/file/" con il percorso completo del file csv)
  chmod a+rx /percorso/del/file/canzoni.csv

Oppure posizionare l'intero progetto o almeno la cartella "Server" in un'unità o directory diversa dal disco di sistema, in modo da poter importare con i permessi necessari tutti i dati dal file 'canzoni.csv'.

Passo 3:

Eseguire il file EmotionalSongs-DBCreator.jar premendo direttamente sull'icona dell'applicazione oppure da terminale dare il seguente comando:

- da Windows
  java -jar EmotionalSongs-DBCreator.jar

- da MacOS e Linux (sostituire "path_javafx" e "path_postgresql" con il percorso completo dei file)
  java --module-path="path_javafx/lib" --add-modules=ALL-MODULE-PATH -cp "path_postgresql/postgresql-42.6.0.jar" -jar EmotionalSongs-DBCreator.jar

Passo 4:

All'avvio dell'applicazione vanno compilati i seguenti campi: username del server PostgreSQL , la password e la porta su cui lavora il server PostgreSQL.Una volta compilato si può premere il tasto "Connetti al DB PostgreSQL" e se i dati sono stati inseriti correttamente verrà creato il database "emotionalsongs".

Se la creazione del database tramite l'applicazione manuale non ha funzionato si può creare il database manualmente facendo i seguenti passaggi:

1. Creare il database
   - Click destro su "Databases" e selezionare "Create" > "Database...". Assegnare il nome "emotionalsongs" al nuovo database e confermare.

2. Creare le tabelle
   Per creare tutte le tabelle necessarie è possibile farlo in due modi diversi:

   2.1. Utilizzando Query tool
   - Click destro sul database "emotionalsongs" e selezionare "Query Tool".
   - Copiare il contenuto del file "creaTabelle.sql" ed eseguire la query.

   2.2. Caricando direttamente il file "creaTabelle.sql" (solo su pgadmin4 v6.21)
   - Click destro sul database "emotionalsongs" e selezionare "Restore...".
   - Su "Filename" selezionare il file "creaTabelle.sql" per creare le tabelle e poi "Restore".

3. Importare le canzoni nel database
   Dopo aver creato tutte le tabelle necessarie, è possibile popolare la tabella "canzoni" utilizzando un file CSV. Ecco come farlo:

   3.1. Utilizzando pgadmin
   - Click destro sulla tabella "canzoni" e selezionare "Import/Export data...".
   - Su "Filename" specificare il percorso del file csv.

- Su "Options":
  Abilitare l'opzione "Header" e impostare "Delimiter" su ";".
- Su "Columns":
  Rimuovere la colonna "idCanzone" dalle colonne selezionate per importare solo le colonne necessarie (titolo, autore, anno, album, durata) dal file CSV.

3.2. Utilizzando il terminale SQL shell(psql)
Se si preferisce l'approccio da linea di comando, è possibile eseguire il seguente comando tramite il terminale SQL (psql). Assicurarsi di sostituire "path" con il percorso completo del file "canzoni.csv":

```
\copy Canzoni(titolo, autore, anno, album, durata) FROM 'path\canzoni.csv'
WITH (FORMAT csv, HEADER true, DELIMITER ';', ENCODING 'UTF-8');
```

4. Aggiungere il trigger
   - Click destro sul database "emotionalsongs" e selezionare "Query Tool".
   - Copiare il contenuto del file "trigger_canzoni.sql" ed eseguire la query.

Passo 5:

Una volta che il database è stato creato, è possibile procedere all'avvio del server. Tuttavia, è importante notare che i passaggi successivi differiscono a seconda di come il server viene avviato: tramite il file JAR o attraverso un IDE come IntelliJ.

Avvio tramite file JAR:

Se il server viene avviato utilizzando il file JAR, non è necessario apportare alcuna modifica al file "configServerPostgresSQL". Dopo aver creato con successo il database “emotionalsongs”, le credenziali necessarie vengono automaticamente salvate e utilizzate per stabilire la connessione al database.

Avvio tramite IDE (IntelliJ):

Se, invece, si sceglie di avviare il server tramite un IDE come IntelliJ, è necessario apportare alcune modifiche al file "configServerPostgresSQL.properties" prima di avviare il server. Questo file è posizionato nella cartella "ConfigServerPostgreSQL". In caso di necessità, è possibile modificare i seguenti campi all'interno del file:

- Password
- Porta nel campo URL
- Username

Assicurarsi di aggiornare queste informazioni in base alle credenziali corrette per il database PostgreSQL.

Passo 6:

A questo punto si può eseguire il file jar dal terminale stando nella cartella “Server” dove risiedono i file jar con il seguente comando:

- da Windows:
  java -jar EmotionalSongs-Server.jar

- da MacOS e Linux (sostituire "path_javafx" e "path_postgresql" con il percorso completo dei file):
  java --module-path="path_javafx/lib" --add-modules=ALL-MODULE-PATH -cp "path_postgresql/postgresql-42.6.0.jar" -jar EmotionalSongs-Server.jar

Alla fine se la configurazione del server è andata a buon fine nel terminale apparirà la scritta “Server ready”.