

SCUOLA DI SCIENZE
Corso di Laurea in Informatica per il Management

**APPLICAZIONE MOBILE PER LA
GESTIONE E L'ANALISI DI
LISTE CINEMATOGRAFICHE**

Relatore:
Dott. FEDERICO
MONTORI

Presentata da:
MATTEO
BOSCHERINI

Correlatore:
Dott. IVAN DIMITRY
RIBEIRO ZYRIANOFF

**II Sessione
Anno Accademico 2024/25**

*A me stesso,
per avercela fatta.*

Indice

Introduzione	1
1 Stato dell'Arte	5
1.1 Panoramica del Settore	5
1.2 IMDb	5
1.3 Letterboxd	6
1.4 TMDB	7
1.5 JustWatch	7
1.6 Analisi Comparativa	8
1.7 Posizionamento di MovieApp	8
2 Progettazione del Sistema	11
2.1 Architettura Generale	11
2.2 Backend	12
2.2.1 Tecnologie Utilizzate	12
2.2.2 Architettura	12
2.2.3 Modello dei Dati	14
2.2.4 Autenticazione e Sicurezza	17
2.3 Client Android	18
2.3.1 Tecnologie Utilizzate	18
2.3.2 Architettura MVVM	18
2.4 Processo di Importazione e Arricchimento Dati	19
2.5 Flusso di Lavoro Principale	20
2.6 Design dell'Interfaccia Utente	21
3 Implementazione	25
3.1 Schermata Home	25
3.1.1 Importazione dei File CSV	25
3.1.2 Popolamento Automatico del Database	28
3.2 Schermata Liste	30
3.2.1 Creazione e Gestione Liste	30

3.2.2	Interfaccia Gestione Liste	32
3.2.3	Esplorazione Liste	32
3.2.4	Follow e Copia Liste	33
3.2.5	Filtraggio Avanzato	34
3.2.6	Notifiche Intelligenti	35
3.3	Schermata Statistiche	36
3.3.1	Principi di Design dei Grafici	36
3.3.2	Descrizione Dettagliata dei 12 Grafici	37
3.4	Sicurezza	46
3.4.1	Autenticazione e Autorizzazione	46
3.4.2	Protezione dei Dati Sensibili	46
4	Valutazione	49
4.1	Il Questionario PSSUQ	49
4.2	Metodologia di Valutazione	49
4.2.1	Partecipanti	49
4.2.2	Strumento di Valutazione	50
4.3	Task di Valutazione	51
4.3.1	Task #1: Caricamento e Analisi Dati	51
4.3.2	Task #2: Creazione Liste	51
4.3.3	Task #3: Funzionalità Social	51
4.4	Risultati Quantitativi	51
4.4.1	Punteggi Domande 1-16	51
4.4.2	Riepilogo Punteggi Medi	54
4.5	Feedback Qualitativo	54
4.5.1	Aspetti Apprezzati	54
4.5.2	Funzionalità Future	55
4.5.3	Ulteriori Commenti e Suggerimenti	55
5	Conclusioni e Sviluppi Futuri	57
5.1	Riepilogo del Lavoro Svolto	57
5.2	Punti di Forza	57
5.3	Limitazioni	58
5.4	Sviluppi Futuri	58
5.5	Conclusione	59
Bibliografia		61
Ringraziamenti		63

Elenco delle figure

1.1	Logo di IMDb	5
1.2	Logo di Letterboxd	6
1.3	Logo di TMDB	7
1.4	Logo di JustWatch	7
2.1	Architettura del sistema	12
2.2	Architettura del backend	14
2.3	Diagramma E-R del database	17
2.4	Pattern architetturale Model-View-ViewModel	19
2.5	Flusso del processo di importazione di un file CSV	19
2.6	Schermata Home dell'applicazione	21
2.7	Schermata Liste dell'applicazione	22
2.8	Schermata Statistiche dell'applicazione	23
3.1	Schermata per importare i file CSV nell'applicazione	25
3.2	Interfaccia di creazione di una lista di film	31
3.3	Configurazione notifiche intelligenti per liste di film	32
3.4	Lista privata nella sezione "Le Mie Liste"	33
3.5	Lista pubblica nella sezione "Liste Pubbliche"	34
3.6	Schermata di dettaglio di una lista con filtri	35
3.7	Notifica intelligente di una lista per un utente	36
3.8	Riepilogo numero di film importati	36
3.9	Grafico Distribuzione Generi	37
3.10	Grafico Distribuzione Film per Anno	38
3.11	Grafico Top Registi	39
3.12	Grafico Top Attori	40
3.13	Grafico Distribuzione Valutazioni TMDB	40
3.14	Grafico Paesi di Produzione	41
3.15	Grafico Film per Decennio	42
3.16	Grafico Combinazioni di Generi	43
3.17	Grafico Distribuzione Durata Film	44

3.18	Grafico Lingue Originali	44
3.19	Grafico Popolarità vs Qualità	45
3.20	Grafico Evoluzione Popolarità nel Tempo	46
4.1	Grafico valutazione domande PSSUQ	53
4.2	Grafico boxplot domande PSSUQ	53
4.3	Grafico categorie PSSUQ per esperienza utenti	54

Elenco delle tabelle

1.1	Confronto tra le principali piattaforme cinematografiche	8
4.1	Punteggi delle sedici domande PSSUQ	52
4.2	Riepilogo Punteggi Medi PSSUQ	54

Listings

3.1	Ricerca film su TMDB (tmdb.service.ts)	26
3.2	Recupero metadati film completi (tmdb.service.ts)	27
3.3	Inizializzazione del backend (movies.service.ts)	28
3.4	Sincronizzazione film popolari TMDB (tmdb.service.ts)	29
3.5	Verifica password nel login (auth.service.ts)	46

Introduzione

Il presente elaborato descrive la progettazione e l'implementazione di MOVIEAPP, un'applicazione mobile creata per la gestione avanzata delle collezioni cinematografiche. L'obiettivo del progetto è quello di offrire agli appassionati di cinema uno strumento moderno e completo, che consenta di importare collezioni di film da piattaforme terze, arricchirli automaticamente con metadati, analizzare le proprie abitudini tramite grafici interattivi e creare liste personalizzate con funzionalità social.

Motivazioni

Negli ultimi anni guardare film è diventata un'abitudine diffusa, favorendo quindi l'utilizzo di piattaforme streaming. Parallelamente è cresciuto l'interesse degli utenti a scoprire nuovi film attraverso strumenti che permettano non solo di catalogare i film visti e film che gli utenti vorrebbero vedere, ma anche di analizzare le proprie preferenze, creare liste personalizzate e condividerle con altri appassionati.

Piattaforme molto diffuse e conosciute, come IMDb e Letterboxd, offrono funzionalità di tracciamento e social networking, ma presentano alcune limitazioni. Entrambe le piattaforme consentono agli utenti di scaricare i propri dati personali, che contengono le liste dei film guardati e dei film che vorrebbero vedere. Tuttavia, IMDb attualmente non fornisce strumenti di analisi visiva e statistica, mentre Letterboxd si concentra maggiormente sull'aspetto social e sulla scrittura di valutazioni e recensioni, ma non offre grafici interattivi dettagliati né la possibilità di esplorare i dati in profondità.

MOVIEAPP nasce proponendo un'esperienza che integri l'importazione di dati da piattaforme esistenti con un sistema di analisi grafica avanzato, composto da dodici grafici interattivi, e funzionalità social che permettono di creare, seguire e copiare delle liste, ovvero delle raccolte di film da guardare.

Obiettivi del Progetto

L'applicazione si articola in due componenti principali.

Importazione, Arricchimento e Analisi

La prima riguarda l'importazione tramite file CSV dei dati provenienti da IMDb e Letterboxd che vengono caricati, interpretati e inviati al backend per l'arricchimento dei film tramite **l'API di The Movie Database (TMDB)** [1]. Una volta arricchiti, i dati creano un sistema di analisi che offre dodici grafici interattivi, progettati per consentire un'esplorazione intuitiva e approfondita delle preferenze cinematografiche dell'utente.

Funzionalità Social

La seconda introduce un aspetto social attraverso la creazione di liste di film da guardare, che possono essere di natura pubblica o privata. Le liste permettono agli utenti di organizzare collezioni tematiche di film e di pianificare la propria esperienza cinematografica. Le liste pubbliche sono visibili a tutti gli utenti del sistema, mentre quelle private sono accessibili solo al creatore. Gli utenti iscritti possono seguire liste di altri utenti e copiare film di questi ultimi nella propria collezione personale. È inoltre implementato un sistema di notifiche intelligenti per la visualizzazione dei film di una lista che supportano obiettivi periodici scelti dall'utente, come ad esempio vedere un film al mese entro una data limite scelta dall'utente.

Struttura della Tesi

Il presente elaborato è strutturato come segue:

- **Capitolo 1 – Stato dell’Arte:** panorama delle applicazioni esistenti per la gestione e l’analisi di collezioni di film, analizzando le piattaforme IMDb, Letterboxd, TMDB e JustWatch.
- **Capitolo 2 – Progettazione del Sistema:** architettura generale di MOVIEAPP, le scelte tecnologiche adottate per la progettazione del backend e la parte Android.
- **Capitolo 3 – Implementazione:** dettagli implementativi delle funzionalità principali di MOVIEAPP con alcuni estratti di codice, includendo la spiegazione del processo di importazione e arricchimento dei dati tramite TMDB e del popolamento del database automatico con circa 10000 film.
- **Capitolo 4 – Valutazione:** risultati della valutazione dell’applicazione mediante un questionario Modulo Google, eseguito su un campione di 35 utenti di diverse età attraverso tre task da svolgere delle funzionalità principali.

- **Capitolo 5 – Conclusioni e Sviluppi Futuri:** riassunto dei risultati ottenuti, evidenziando punti di forza, limitazioni dell'applicazione e proponendo possibili sviluppi futuri.

Contributi Principali

I principali contributi di questo lavoro possono essere riassunti nei seguenti punti:

- Progettazione di un'applicazione mobile Android per la gestione e l'analisi di collezioni cinematografiche.
- Implementazione di un backend scalabile in NestJS con database PostgreSQL, in grado di gestire l'importazione di file CSV di grandi dimensioni e l'arricchimento automatico tramite l'API di TMDB.
- Realizzazione di un sistema di analisi tramite dodici grafici interattivi, progettati per esplorare preferenze cinematografiche attraverso diverse dimensioni (generi di film, film per anno, registi, attori, valutazione TMDB, paesi di produzione, film per decennio, combinazione di generi, durata dei film, lingue originali, relazione qualità-popolarità e popolarità dei film nei decenni).
- Funzionalità social per la creazione, condivisione e gestione di liste pubbliche e private, con un sistema follower e copia delle liste.
- Valutazione dell'applicazione attraverso un questionario sottoposto ad un gruppo di 35 persone di età differenti.

Capitolo 1

Stato dell'Arte

In questo capitolo vengono analizzate le principali piattaforme e applicazioni esistenti per la gestione e l'analisi di liste cinematografiche evidenziando i loro punti di forza e le loro limitazioni.

1.1 Panoramica del Settore

Negli ultimi anni si è osservata una crescita significativa nell'utilizzo di applicazioni per la gestione di collezioni di film, con l'emergere di diverse piattaforme che offrono funzionalità di catalogazione e social networking per gli appassionati di cinema.

1.2 IMDb



Figura 1.1: Logo di IMDb

IMDb [2] è un database cinematografico di proprietà Amazon dal 1998. Fondato nel 1990, rappresenta una fonte di informazioni sui film, con un database che conta milioni di titoli catalogati.

La piattaforma offre un sistema di valutazione basato sui voti degli utenti tramite una media ponderata, permettendo di creare una watchlist personale e di esportare le

proprie liste di film in formato CSV. IMDb si distingue per la completezza delle informazioni tecniche e produttive anche se presenta limitazioni per quanto riguarda le funzionalità di analisi e di visualizzazione statistica dei propri gusti personali, non offrendo grafici per esplorare la propria collezione. Inoltre le funzionalità social sono limitate, con poca possibilità di interazione tra utenti e condivisione di liste personalizzate.

1.3 Letterboxd



Figura 1.2: Logo di Letterboxd

Letterboxd [3] è una piattaforma social dedicata agli appassionati di cinema, lanciata nel 2011 in Nuova Zelanda. Si distingue per il suo design moderno ed elegante ed una particolare attenzione sull'aspetto social.

La piattaforma permette di visualizzare un diario cinematografico (con relative date di visione ed eventuali valutazioni personali), creare liste e scrivere recensioni dei film. Il sistema social include follow, like e commenti, creando una community attiva di appassionati. Anche Letterboxd offre, come IMDb, l'esportazione dei propri dati in formato CSV.

Nonostante i suoi punti di forza nel design e nelle funzionalità social, Letterboxd presenta limitazioni per quanto riguarda l'analisi dettagliata dei propri gusti cinematografici. Inoltre alcune funzionalità avanzate richiedono un abbonamento PRO a pagamento, che rappresenta un limite per utenti che desiderano utilizzare strumenti di analisi senza costi aggiuntivi.

1.4 TMDB



Figura 1.3: Logo di TMDB

TMDB [1] è un database cinematografico community-driven fondato nel 2008. Si distingue principalmente per la sua API ben documentata, che lo rende la scelta preferita degli sviluppatori per creare applicazioni cinematografiche di terze parti.

La piattaforma offre metadati ricchi e dettagliati sulle pellicole, come regista, cast, genere, trama, immagini e trailer, con supporto multilingua per oltre 39 lingue. Il database è costantemente aggiornato e verificato dalla community di utenti, garantendo una grandissima copertura internazionale di quasi tutti i film.

Tuttavia TMDB è progettato principalmente come database e risorsa per sviluppatori che costruiscono un'applicazione, piuttosto che come piattaforma per utenti finali. L'interfaccia web è orientata alla consultazione di film, manca completamente la funzionalità social tra utenti e non sono disponibili strumenti di analisi personale, come statistiche per esplorare i propri gusti cinematografici.

1.5 JustWatch



Figura 1.4: Logo di JustWatch

JustWatch [4] è una piattaforma lanciata nel 2014, che aggredisce informazioni sulla disponibilità di film su servizi di streaming, noleggio e acquisto digitale. Il suo obiettivo

principale è aiutare gli utenti a trovare rapidamente dove poter guardare un contenuto specifico tra oltre 100 servizi diversi.

La piattaforma offre un motore di ricerca avanzato tramite titolo dei film, e propone quelli in tendenza. JustWatch permette di creare watchlist personali, di seguire watchlist di altri utenti e invia notifiche quando un film diventa disponibile.

Tuttavia JustWatch si concentra esclusivamente sulla scoperta e disponibilità dei contenuti, senza offrire funzionalità di statistiche sui propri gusti.

1.6 Analisi Comparativa

La Tabella 1.1 riassume le caratteristiche principali delle piattaforme analizzate, evidenziandone punti di forza e limitazioni.

Tabella 1.1: Confronto tra le principali piattaforme cinematografiche

Caratteristica	IMDb	Letterboxd	TMDB	JustWatch
Database esteso	VV	V	VV	V
Import CSV	V	V	X	X
Export CSV	V	V	X	X
Liste personalizzate	V	VV	V	V
Statistiche	X	V	X	X
Grafici interattivi	X	X	X	X
Recensioni	V	VV	X	X
Streaming info	V	V	X	VV
API gratuita	X	X	VV	X

Legenda: VV = molto buono, V = buono, X = assente

1.7 Posizionamento di MovieApp

MOVIEAPP si posiziona come una **piattaforma di gestione e analisi di liste cinematografiche**, offrendo:

- **Import Facilitato:** import di film dalle collezioni IMDb e Letterboxd in formato CSV, con arricchimento automatico tramite TMDB.
- **Grafici Avanzati:** dodici grafici interattivi per esplorare la propria collezione.
- **Liste Personalizzate:** sistema di liste personalizzate con notifiche basate su obiettivi temporali prefissati dall'utente.
- **Funzionalità Social:** follow, copia liste ed esplorazione liste pubbliche di altri utenti.

- **Gratuito:** tutte le funzionalità disponibili sono gratuite.

MOVIEAPP non mira a sostituire IMDb o Letterboxd, ma ad offrire un'alternativa complementare, focalizzata sull'analisi personale e sulla gestione personalizzata delle proprie liste cinematografiche, con particolare attenzione agli utenti che desiderano visualizzare e comprendere i propri gusti attraverso grafici e statistiche dettagliate.

Capitolo 2

Progettazione del Sistema

In questo capitolo viene illustrata la progettazione di MOVIEAPP, descrivendo l'architettura generale del sistema, le scelte tecnologiche adottate, il modello del database e il design dell'interfaccia utente.

2.1 Architettura Generale

MOVIEAPP permette agli utenti di importare le proprie collezioni cinematografiche tramite file CSV, esportati da IMDb e Letterboxd, includendo sia i film visti (watched) che quelli da vedere (watchlist). Una volta caricati, il sistema elabora questi dati e calcola diverse statistiche sulle abitudini cinematografiche dell'utente. Inoltre, gli utenti possono creare liste personalizzate di film da guardare, specificando una pianificazione temporale per la quale l'applicazione invia notifiche di promemoria. Le liste hanno una funzionalità social, poiché gli utenti possono seguire le liste pubbliche create da altri utenti, favorendo la condivisione e la scoperta di nuovi film. L'applicazione è basata su un'architettura **client-server distribuita**, con una chiara separazione delle responsabilità tra le due componenti principali:

- **Client Android:** responsabile dell'interfaccia utente, dell'esperienza utente e della visualizzazione interattiva dei dati.
- **Backend:** responsabile dell'elaborazione dei dati ricevuti dal client Android, dell'arricchimento delle informazioni sui film tramite TMDB, della gestione delle liste e della persistenza dei dati nel database PostgreSQL.

La Figura 2.1 mostra l'architettura dell'applicazione.



Figura 2.1: Architettura del sistema

La comunicazione tra client e server avviene attraverso due canali principali:

- **API RESTful**: per operazioni CRUD (Create Read Update Delete) su film, utenti e liste.
- **WebSocket (Socket.IO)**: per aggiornamenti in tempo reale durante le operazioni più lunghe, quali l'importazione e l'arricchimento dei film.

2.2 Backend

2.2.1 Tecnologie Utilizzate

Il backend è stato sviluppato utilizzando le seguenti tecnologie:

- **TypeScript**: linguaggio di programmazione tipizzato che estende JavaScript.
- **NestJS** [5]: framework Node.js basato su TypeScript che segue i principi dell'architettura modulare.
- **PostgreSQL**: database relazionale open-source, è stato scelto per il supporto nativo di tipi di dati avanzati come gli array, particolarmente utili per rappresentare metadati complessi provenienti da TMDB (ad esempio array del cast di un film).
- **TypeORM**: semplifica l'interazione con il database, gestendo automaticamente la costruzione delle query SQL e le relazioni tra entità.
- **Socket.IO**: libreria per la comunicazione bidirezionale in tempo reale tra client e server ed è utilizzata per inviare aggiornamenti sullo stato di avanzamento delle operazioni di importazione e arricchimento.

2.2.2 Architettura

Il backend segue un'architettura modulare, dove ogni modulo è responsabile di una specifica funzionalità. Ogni modulo è composto da:

- **Controller:** gestisce le richieste HTTP, valida i dati ricevuti e delega la logica di business ai service. I controller espongono gli endpoint REST dell'API.
- **Service:** contiene la logica di business dell'applicazione. I service interagiscono con il database tramite il repository e eseguono operazioni complesse delegate dai controller.
- **Module:** definisce le associazioni tra service e controller.

I principali moduli del backend sono:

- **auth:** gestisce l'autenticazione degli utenti tramite JWT (JSON Web Token), un token sicuro che contiene informazioni sull'identità dell'utente e permette l'autenticazione stateless, consentendo la registrazione e il login di un utente.
- **movies:** gestisce il catalogo dei film, l'import da file CSV e l'associazione tra film e utenti (user-movies).
- **tmdb:** si occupa dell'integrazione con l'API di TMDB per la ricerca e l'arricchimento dei film.
- **lists:** gestisce la creazione, modifica e condivisione con tutti gli utenti della piattaforma di liste pubbliche e private.
- **websocket:** gestisce le connessioni WebSocket per l'invio di notifiche in tempo reale.
- **analytics:** genera le statistiche e i dati necessari per creare i grafici interattivi.
- **database:** fornisce i servizi di accesso al database.

La Figura 2.2 mostra lo schema dell'architettura del backend.

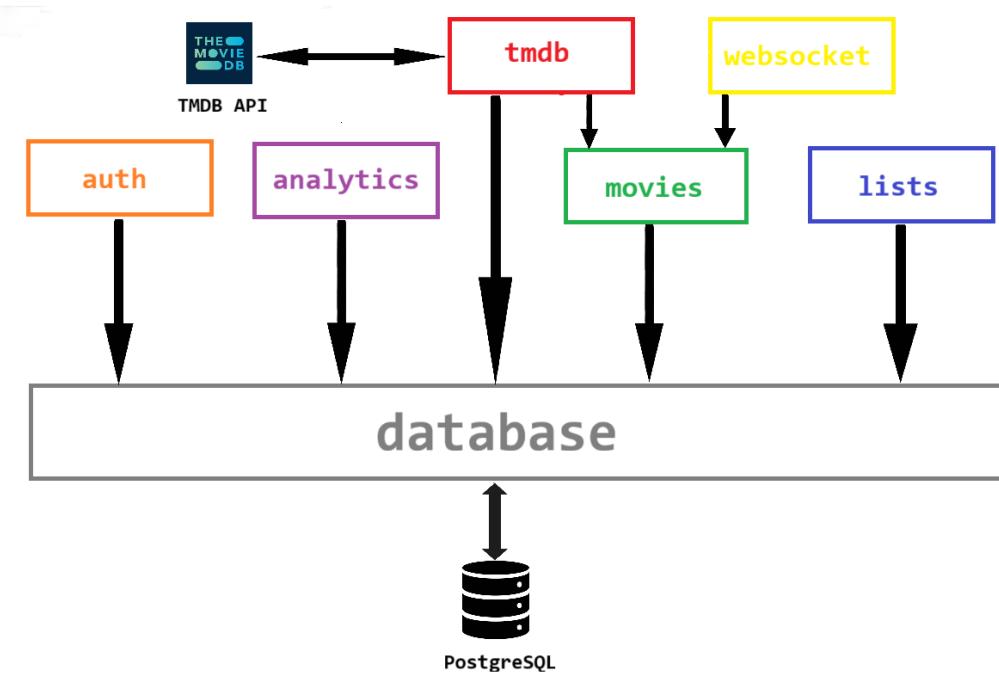


Figura 2.2: Architettura del backend

2.2.3 Modello dei Dati

Il database PostgreSQL è composto da quattro tabelle, progettate per minimizzare la duplicazione dei dati e garantire l'integrità referenziale.

Tabella users Contiene le informazioni sugli utenti registrati all'applicazione. I campi principali della tabella sono:

- `id`: identificatore univoco dell'utente.
- `is_active`: campo booleano che indica se l'account è attivo.
- `last_login`: data e ora dell'ultimo accesso all'applicazione.
- `created_at`: data e ora di creazione dell'account.
- `updated_at`: data e ora dell'ultimo aggiornamento dell'account.
- `email`: indirizzo email univoco dell'utente.
- `password`: password dell'utente hashata tramite bcrypt.
- `username`: username dell'utente opzionale.

Tabella movies Rappresenta tutti i film con i metadati completi provenienti da TMDB. Ogni film è registrato una sola volta nel sistema, evitando duplicazioni. I campi principali della tabella sono:

- **id**: identificatore univoco del film.
- **title**: titolo del film.
- **year**: anno di uscita del film.
- **source**: sorgente di importazione (TMDB, IMDb o Letterboxd).
- **tmdb_id**: id del film su TMDB.
- **is_enriched**: flag booleano che indica se il film è stato arricchito con i metadati TMDB.
- **genres**: array contenente i generi del film.
- **director**: nome del regista.
- **actors**: array contenente il cast del film.
- **overview**: trama del film.
- **runtime**: durata del film in minuti.
- **tmdb_rating**: valutazione media del film su TMDB.
- **vote_count**: numero di voti ricevuti dal film.
- **popularity**: valore numerico di popolarità del film.
- **budget**: budget di produzione del film.
- **revenue**: incasso totale del film.
- **production_companies**: array contenente le case di produzione del film.
- **production_countries**: array contenente i paesi di produzione del film.
- **original_language**: lingua originale del film.
- **adult**: flag booleano che indica se il film è per adulti.
- **trailer_url**: URL del trailer del film.
- **created_at**: data di aggiunta del film nel database.

- `updated_at`: data di eventuale modifica del film nel database.
- `poster_url`: URL dell'immagine del poster del film.
- `backdrop_url`: URL dell'immagine di sfondo del film.

Tabella `user_movies` Questa tabella implementa la relazione N-N tra utenti e film tramite le associazioni di un utente con un film. I campi principali sono:

- `id`: identificatore univoco della relazione.
- `user_id`: chiave esterna con l'id dell'utente dalla tabella `users`.
- `movie_id`: chiave esterna con l'id del film dalla tabella `movies`.
- `status`: stato del film (watched o watchlist).
- `user_rating`: valutazione opzionale dell'utente del film.
- `watched_date`: data di visione del film.
- `created_at`: data di creazione della relazione tra utente e film nel database.
- `updated_at`: data di modifica della relazione tra utente e film nel database.

Tabella `movie_lists` Contiene le liste di film create dagli utenti. I campi principali della tabella sono:

- `id`: identificatore univoco della lista.
- `user_id`: chiave esterna con l'id dell'utente dalla tabella `users`.
- `name`: nome della lista.
- `description`: descrizione opzionale della lista.
- `movie_ids`: array contenente gli id dei film presenti nella lista.
- `target_date`: data e ora della deadline per la visione dei film nella lista.
- `frequency`: frequenza delle notifiche di tipo giornaliera, settimanale e mensile.
- `notifications_enabled`: flag booleano che indica se la lista può ricevere notifiche intelligenti.
- `is_public`: flag booleano che indica se la lista è pubblica o privata.
- `followers_count`: numero di followers della lista.

- `followers_ids`: array contenente gli id degli utenti che seguono la lista.
- `created_at`: data di creazione della lista.
- `updated_at`: data dell'ultima modifica della lista.

La Figura 2.3 mostra il diagramma Entità-Relazione del database `movieapp`.

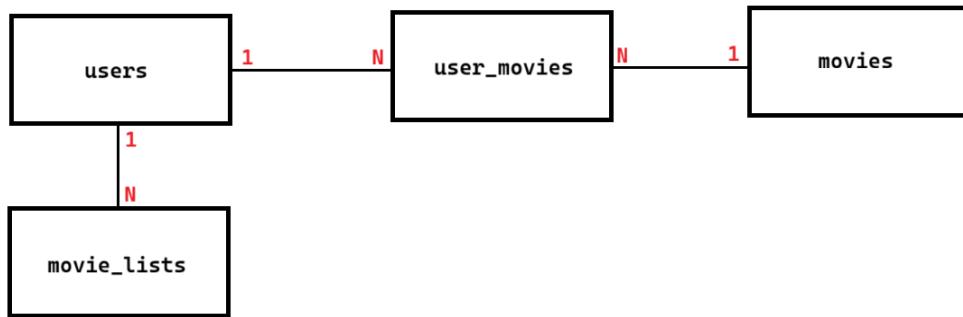


Figura 2.3: Diagramma E-R del database

2.2.4 Autenticazione e Sicurezza

L'autenticazione degli utenti si basa su **JWT**, un metodo che consente di gestire sessioni utente senza memorizzare informazioni sul server. Il processo di autenticazione funziona come segue:

1. L'utente invia le credenziali (email e password) all'endpoint `/api/v1/auth/login`.
2. Il backend verifica le credenziali confrontando l'hash della password con quello salvato nel database.
3. Se le credenziali sono corrette, il backend genera un JWT contenente l'ID dell'utente e una data di scadenza, con validità di 7 giorni.
4. Il client memorizza il JWT e lo include nell'header di ogni richiesta successiva.
5. Per ogni richiesta protetta, il backend valida il JWT e autorizza l'accesso alle risorse dell'utente.

Le password sono hashate utilizzando **bcrypt**, un algoritmo crittografico progettato per essere resistente agli attacchi brute-force.

2.3 Client Android

2.3.1 Tecnologie Utilizzate

Il client Android è stato sviluppato utilizzando:

- **Kotlin**: linguaggio di programmazione per lo sviluppo Android, studiato nel corso di Laboratorio di Applicazioni Mobili.
- **Android Jetpack**: insieme di librerie e strumenti per lo sviluppo Android.
- **Retrofit**: libreria HTTP per Android, utilizzata per effettuare chiamate REST al backend.
- **Socket.IO**: libreria per gestire le connessioni WebSocket e ricevere aggiornamenti in tempo reale.
- **MPAndroidChart**: libreria per la creazione di grafici interattivi (pie chart, bar chart, scatter plot e line chart).
- **Material Design 3**: sistema di design per la creazione di interfacce moderne, coerenti e accessibili agli utenti.

2.3.2 Architettura MVVM

Il client Android adotta il pattern architettonico **MVVM (Model-View-ViewModel)**. I componenti sono:

View La View è rappresentata dai Fragment, che implementano l'interfaccia grafica e gestiscono l'interazione con l'utente. Non contengono la logica di business, ma si limitano ad osservare i dati esposti dai ViewModel tramite LiveData, reagire ai cambiamenti di stato aggiornando l'interfaccia e delegare le azioni dell'utente ai ViewModel.

ViewModel I ViewModel gestiscono la logica di business e mantengono lo stato dell'interfaccia, sopravvivono ai cambiamenti di configurazione evitando, ad esempio, la perdita di stato durante la rotazione dello schermo.

Model Il Model è rappresentato dai Repository, che fungono da intermediari tra i ViewModel e le sorgenti dati.

La Figura 2.4 illustra il pattern MVVM adottato nel client Android.

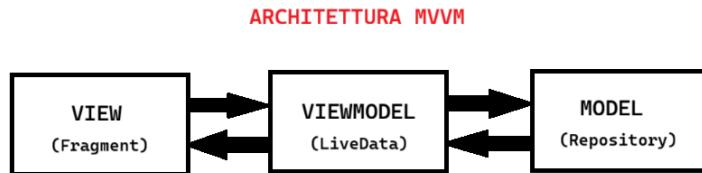


Figura 2.4: Pattern architetturale Model-View-ViewModel

2.4 Processo di Importazione e Arricchimento Dati

Il processo di importazione e l'arricchimento dei dati cinematografici rappresenta il cuore funzionale di MOVIEAPP. La Figura 2.5 illustra le sei fasi principali che compongono questo flusso:

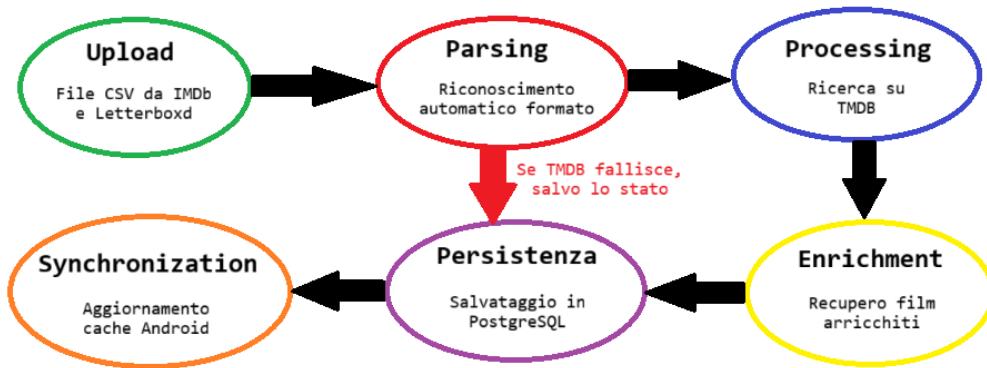


Figura 2.5: Flusso del processo di importazione di un file CSV

1. **Upload:** l'utente seleziona un file CSV, esportato dalle piattaforme esistenti da IMDb e Letterboxd, contenente la propria collezione cinematografica che desidera importare all'interno dell'applicazione.
2. **Parsing:** il sistema riconosce automaticamente il formato del file (se proviene da IMDb o Letterboxd) e ne estrae i dati essenziali (titolo e anno). Se dovesse fallire l'import di un film è possibile procedere allo stesso procedimento per il film successivo.
3. **Processing:** i film estratti dall'applicazione vengono inviati al backend, che effettua una ricerca su TMDB per identificarli e associarli al corrispondente `tmdb_id`, tramite nome e anno di uscita.
4. **Enrichment:** per ogni film identificato, il backend li arricchisce con metadati provenienti da TMDB.

5. **Persistenza:** i film arricchiti vengono salvati nel database PostgreSQL. Questa strategia evita chiamate ripetute all'API di TMDB per film già arricchiti, in quanto sono già salvati nel database; c'è un flag all'interno della tabella `movies` chiamato `is_enriched` che viene arricchito solamente se il valore è `false`, se il valore è `true` significa che il film è già stato arricchito.
6. **Synchronization:** al termine dell'elaborazione del backend e il salvataggio tramite il database PostgreSQL, il client Android sincronizza automaticamente la propria cache locale con il database.

2.5 Flusso di Lavoro Principale

Il flusso di lavoro principale dell'applicazione può essere riassunto nei seguenti passi:

1. **Registrazione e Login:** L'utente crea un account inserendo email e password, oppure effettua il login se è già registrato.
2. **Popolamento Automatico:** Al primo accesso il database locale viene popolato con circa 10000 film popolari da TMDB, permettendo all'utente di creare liste anche senza aver importato i propri film da IMDb e Letterboxd.
3. **Importazione CSV:** L'utente importa i propri film da IMDb o Letterboxd selezionando un file CSV. Il backend elabora il file, cerca i film su TMDB e li arricchisce con metadati completi.
4. **Visualizzazione Statistiche:** Una volta completata l'importazione, l'utente può esplorare i propri gusti e preferenze cinematografiche attraverso dodici grafici interattivi.
5. **Creazione Liste:** L'utente può creare liste, aggiungere e rimuovere film, impostare la visibilità e condividerle con altri utenti.
6. **Funzionalità Social:** L'utente può scoprire liste pubbliche create da altri, seguirle o copiarle nella propria collezione.
7. **Notifiche Intelligenti:** Durante la creazione di una lista, l'utente ha la possibilità di attivare notifiche intelligenti, ricevendo promemoria per guardare i film presenti nella lista entro un determinato arco temporale.

2.6 Design dell'Interfaccia Utente

L'interfaccia di MOVIEAPP è progettata seguendo i principi del **Material Design 3** ed è organizzata in tre sezioni principali, accessibili tramite una **Bottom Navigation Bar**:

Home La sezione **Home** permette all'utente di:

- Visualizzare un riepilogo con il numero dei film importati (numero di film totali, numero di film visti e numero di film da vedere).
- Importare file CSV da IMDb e Letterboxd.

La Figura 2.6 mostra la schermata **Home** dell'applicazione.

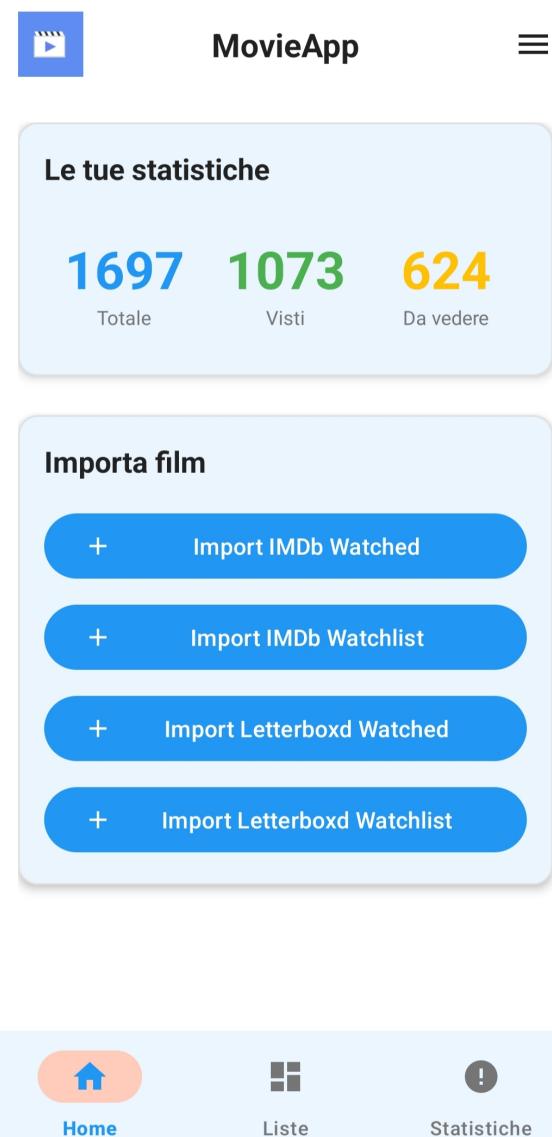


Figura 2.6: Schermata Home dell'applicazione

Liste La sezione **Liste** consente di:

- Creare e modificare nuove liste (pubbliche o private).
- Aggiungere o rimuovere dei film.
- Visualizzare le liste pubbliche create da altri utenti.
- Seguire liste pubbliche di altri utenti e copiarle nella propria collezione.
- Filtrare e ordinare le liste per nome, ordine alfabetico A-Z e Z-A, popolarità e visibilità della lista (pubblica o privata).

La Figura 2.7 mostra la schermata **Liste** dell'applicazione.

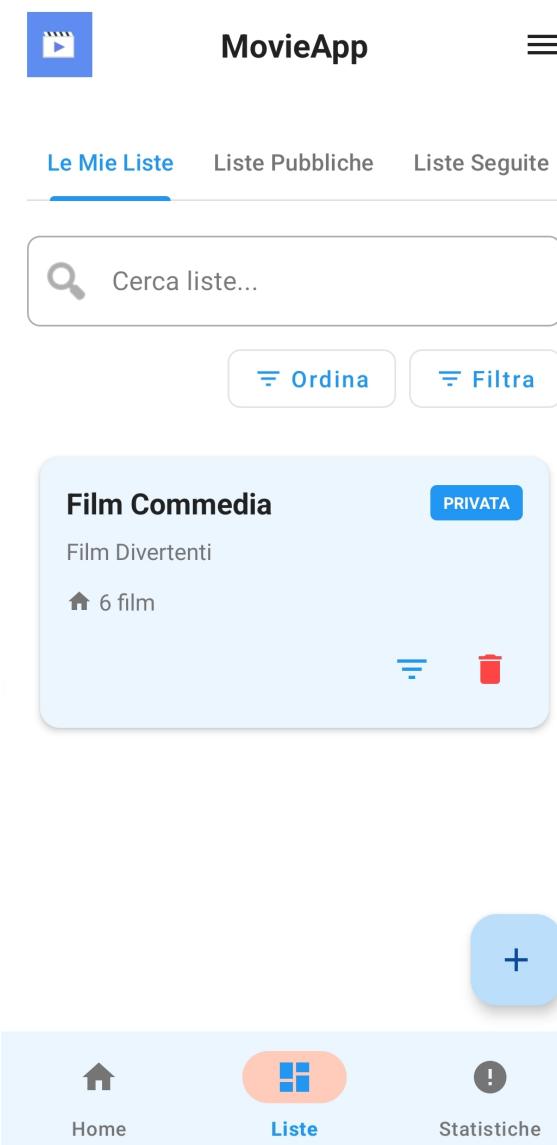


Figura 2.7: Schermata Liste dell'applicazione

Statistiche La sezione **Statistiche** offre:

- Dodici grafici interattivi per esplorare le preferenze cinematografiche dell'utente.
- Statistiche sintetiche (genere più visto dall'utente, film più lungo caricato dall'utente, etc..).
- Possibilità di esplorare i grafici cliccandoli e zoomandoli.

La Figura 2.8 mostra la schermata **Statistiche** dell'applicazione.

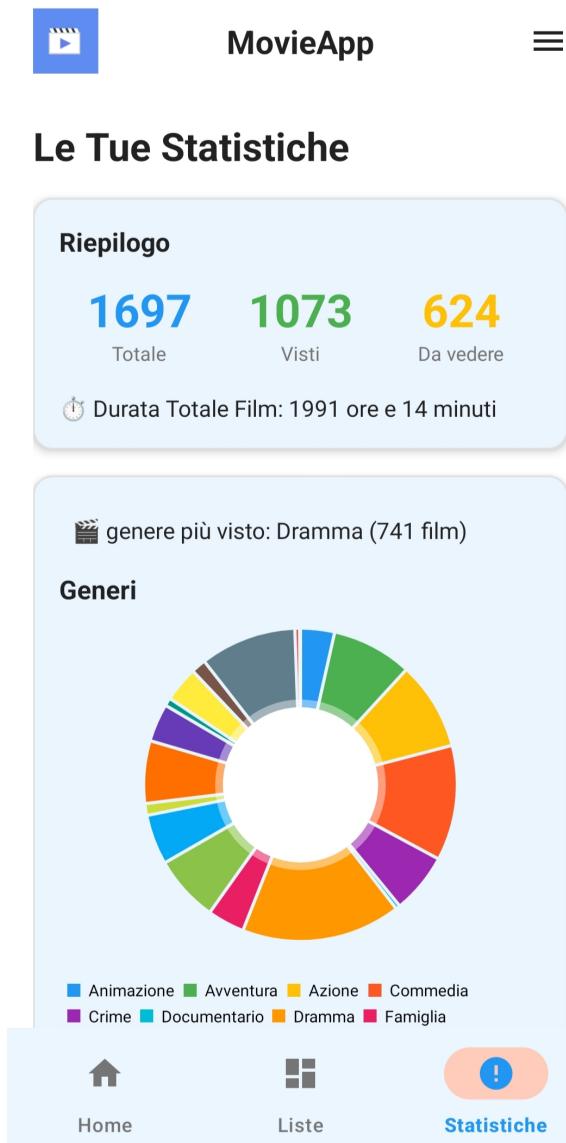


Figura 2.8: Schermata Statistiche dell'applicazione

Capitolo 3

Implementazione

In questo capitolo vengono descritti i dettagli implementativi di MOVIEAPP, analizzando le funzionalità principali, l'integrazione con l'API di TMDB per arricchire i film, il sistema di visualizzazione interattiva e la gestione delle liste. Vengono presentati alcuni estratti di codice per chiarirne la spiegazione.

3.1 Schermata Home

La schermata **Home** dell'applicazione permette l'importazione dei film da file CSV.

3.1.1 Importazione dei File CSV

L'applicazione supporta l'import di file CSV di qualunque dimensione generati da IMDb e Letterboxd. Il processo di importazione è articolato in diverse fasi, ciascuna gestita da componenti presenti sia sul client Android che sul backend. Nella Figura 3.1 è mostrata la schermata Home dove è possibile cliccare i bottoni per importare i film.



Figura 3.1: Schermata per importare i file CSV nell'applicazione

Lato Client Android

Sul client Android, il processo inizia quando l'utente seleziona un file CSV tramite un *file picker*. L'applicazione riconosce automaticamente la piattaforma di origine, e se il file proviene da una watched (film visti) o da una watchlist (film da vedere). Il parsing dei file CSV avviene localmente sul dispositivo tramite la classe `CsvProcessor`. Una volta completato il parsing locale i film vengono inviati al backend tramite `ImportWorker`, che gestisce l'upload in background dei film.

Lato Backend

Una volta ricevuti i film dal client, il backend avvia un processo asincrono articolato in due fasi.

Fase 1: Ricerca su TMDB Per ogni film ricevuto, il backend effettua una ricerca su TMDB utilizzando il titolo e l'anno di uscita del film come parametri.

Il seguente estratto di codice mostra la logica di ricerca tramite il metodo `searchByTitle` nel file `tmdb.service`. Il metodo invia una richiesta GET all'endpoint `/search/movie` di TMDB con titolo e anno del film. L'API restituisce una lista di risultati ordinati per rilevanza, viene estratto il primo risultato e utilizzato per recuperare i dettagli completi del film tramite il metodo `getMovieDetails`. In caso di errore, viene registrato nel log e restituito `null`.

```

1  async searchByTitle(
2      title: string,
3      year: number | undefined
4  ): Promise<TmdbMovieDetails | null> {
5      try {
6          const url = `${this.baseUrl}/search/movie`;
7          const response = await firstValueFrom(
8              this.httpService.get<TmdbSearchResponse>(url, {
9                  params: {
10                      api_key: this.apiKey,
11                      query: title,
12                      year: year,
13                      language: 'it-IT'
14                  }
15              })
16          );
17
18          // prendi primo risultato (best match)
19          const bestMatch = response.data.results[0];

```

```

20         return await this.getMovieDetails(bestMatch.id);
21     }
22     catch (error) {
23         this.logger.error(`errore ricerca titolo ${title}: ${
24             error.message}`);
25         return null;
26     }
27 }
```

Listing 3.1: Ricerca film su TMDB (tmdb.service.ts)

Fase 2: Recupero Metadati Completi Una volta identificato il `tmdb_id`, il backend recupera i metadati completi arricchiti sul film.

Il seguente estratto di codice mostra il procedimento di arricchimento tramite il metodo `getMovieDetails` nel file `tmdb.service`. Il metodo effettua una richiesta GET all'endpoint specifico del film utilizzando il suo `tmdb_id`. Il parametro `append_to_response` permette di recuperare i metadati completi con una singola chiamata API. In caso di successo, i dati vengono restituiti, mentre in caso di errore, viene registrato nel log e restituito `null`.

```

1  async getMovieDetails(tmdbId: number): Promise<
2     TmdbMovieDetails | null> {
3     try {
4         const url = `${this.baseUrl}/movie/${tmdbId}`;
5         const response = await firstValueFrom(
6             this.httpService.get<TmdbMovieDetails>(url, {
7                 params: {
8                     api_key: this.apiKey,
9                     language: 'it-IT',
10                    append_to_response: 'credits,keywords,
11                                     videos,releases'
12                }
13            })
14        );
15        return response.data;
16    }
17    catch (error) {
18        this.logger.error(`errore dettagli film ${tmdbId}: ${
19             error.message}`);
20        return null;
21    }
22 }
```

Listing 3.2: Recupero metadati film completi (tmdb.service.ts)

I metadati recuperati vengono salvati nella tabella `movies` del database PostgreSQL. Per evitare duplicazioni, il backend verifica prima se il film esiste già tramite `tmdb_id` prima di fare la seconda chiamata all'API TMDB.

3.1.2 Popolamento Automatico del Database

Al primo login dell'utente sull'applicazione, per consentirgli di creare liste personalizzate e aggiungere film anche in assenza di un import da file CSV, il sistema effettua automaticamente il popolamento del database locale con circa 10000 film caratterizzati da un elevato indice di popolarità su TMDB, in quanto si presume che la maggior parte dei film che vogliono vedere gli utenti provenga da film presenti in tali elenchi.

Il seguente estratto di codice mostra il metodo `initializeApp` nel file `movies.service` che verifica lo stato del database. Il metodo conta quanti film sono già presenti nel database. Se il conteggio è zero (primo avvio), viene avviato in background il processo di sincronizzazione che scarica 10000 film popolari da TMDB tramite il metodo `syncPopularMovies`. Se invece il database contiene già dei film, il metodo restituisce semplicemente una conferma che l'inizializzazione è stata già completata.

```

1  async initializeApp(): Promise<{
2      needsSync: boolean;
3      message: string;
4      stats?: any;
5  }> {
6      const movieCount = await this.movieRepository.count();
7
8      // primo avvio: scarica 10k film popolari
9      if (movieCount === 0) {
10          // avvia sync in background
11          this.tmdbService.syncPopularMovies(10000)
12
13          return {
14              needsSync: true,
15              message: 'scaricamento 10000 film popolari',
16          };
17      }
18
19      return {
20          needsSync: false,
21          message: 'database già inizializzato'
22      };
23  }

```

```

22     };
23 }
```

Listing 3.3: Inizializzazione del backend (movies.service.ts)

Il download effettivo dei film con l'indice `tmdb_id` avviene tramite chiamate all'API di TMDB ricevendo solo il nome e l'anno della pellicola.

Il seguente estratto di codice mostra il metodo `syncPopularMovies` nel file `tmdb.service` che gestisce la sincronizzazione. Il metodo itera attraverso le pagine dei risultati dell'endpoint `/movie/popular` di TMDB, recuperando i film più popolari. Per ogni pagina, vengono estratti i film e per ciascuno viene creato un oggetto `Movie` contenente le informazioni essenziali: `tmdb_id`, titolo e anno di uscita. Ogni film viene salvato nel database con il flag `is_enriched = false`, che indica che i metadati completi non sono ancora stati scaricati per velocizzare il processo iniziale.

```

1  async syncPopularMovies(limit: number = 10000): Promise<{
2     synced: number; errors: number > {
3         let synced = 0;
4         let errors = 0;
5
6         for (let page = 1; page <= totalPages; page++) {
7             const response = await firstValueFrom(
8                 this.httpService.get<TmdbSearchResponse>(`${this.
9                     baseUrl}/movie/popular`, {
10                     params: {
11                         api_key: this.apiKey,
12                         language: 'it-IT',
13                         page: page
14                     }
15                 )
16             );
17
18             for (const tmdbMovie of response.data.results) {
19                 const movie: Movie = {
20                     id: 'tmdb_${tmdbMovie.id}',
21                     title: tmdbMovie.title,
22                     year: tmdbMovie.release_date
23                         ? new Date(tmdbMovie.release_date).
24                             getFullYear()
25                         : undefined,
26                     source: 'TMDB',
27                     tmdb_id: tmdbMovie.id,
28                     is_enriched: false
29                 };
30             }
31         }
32     }
33 }
```

```

26         } ;
27
28         await this.databaseService.saveMovie(movie) ;
29         synced++ ;
30     }
31 }
32
33     return { synced, errors } ;
34 }

```

Listing 3.4: Sincronizzazione film popolari TMDB (tmdb.service.ts)

Questi film vengono salvati con il flag `is_enriched = false`, permettendo di scaricare velocemente i 10000 film popolari da TMDB, ma rimandando il download dei metadati completi al momento effettivo dell'inserimento del film nella lista o dell'inserimento del film tramite file CSV.

3.2 Schermata Liste

La schermata **Liste** è uno strumento di condivisione di liste di film e introduce un livello social nell'applicazione.

3.2.1 Creazione e Gestione Liste

Le liste sono create dagli utenti tramite un'interfaccia semplice basata su **Bottom-SheetDialog**. Ogni lista include:

- **Nome:** titolo della lista
- **Descrizione:** descrizione opzionale
- **Visibilità:** lista pubblica (visibile a tutti) o lista privata (visibile solo al proprietario).
- **Notifiche:** se abilitato il sistema di notifiche intelligenti, l'utente ha la possibilità di essere invitato a vedere i film presenti nella lista, in un determinato periodo temporale scelto.

La Figura 3.2 mostra l'interfaccia di creazione di una nuova lista per un utente.

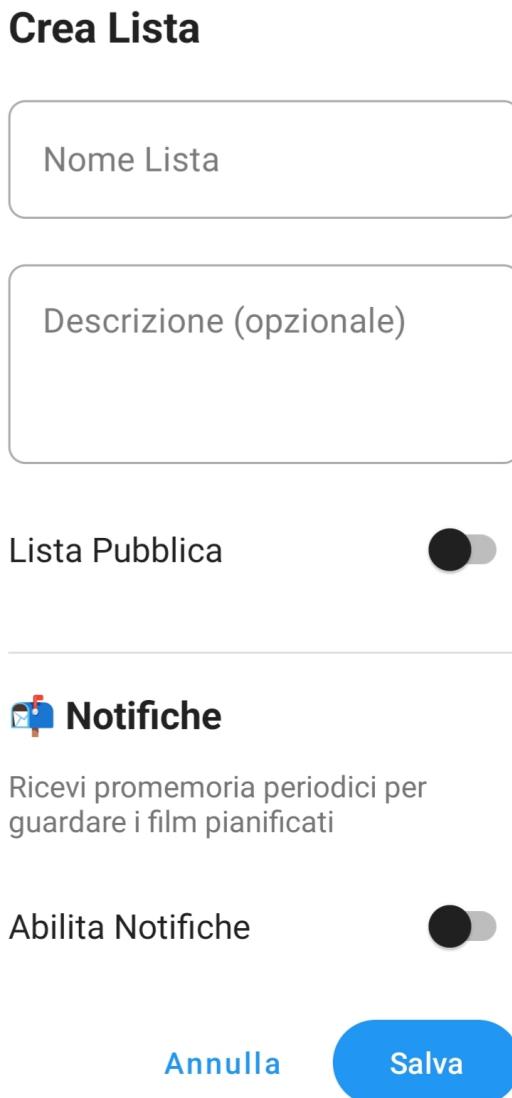


Figura 3.2: Interfaccia di creazione di una lista di film

Sistema di Notifiche Intelligenti

Il sistema di notifiche intelligenti permette all’utente di impostare dei promemoria periodici per completare la visione dei film presenti nelle liste personalizzate. La figura 3.3 illustra la configurazione delle notifiche intelligenti. L’utente può configurare:

- **Data Obiettivo:** termine entro cui completare la visione dei film all’interno della lista.
- **Frequenza Notifiche:** cadenza con cui ricevere i promemoria che può essere giornaliera, settimanale o mensile.

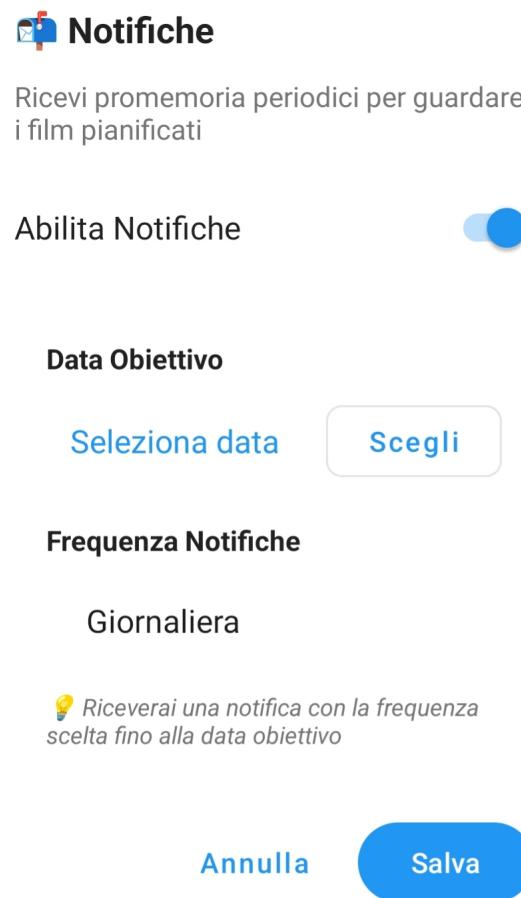


Figura 3.3: Configurazione notifiche intelligenti per liste di film

Il backend calcola automaticamente il ritmo di visione consigliato in base al numero di film rimanenti e ai giorni disponibili, inviando notifiche all’utente nei periodi scelti.

3.2.2 Interfaccia Gestione Liste

L’applicazione organizza le liste in tre tab principali:

- **Le Mie Liste:** contiene tutte le liste create dall’utente (pubbliche e private).
- **Liste Pubbliche:** mostra le liste pubbliche create da altri utenti.
- **Liste Seguite:** raccoglie le liste che l’utente segue.

3.2.3 Esplorazione Liste

Le liste sono presentate in una griglia. L’utente può applicare filtri alfabetici, per popolarità, tramite nome della lista e per visibilità.

La figura 3.4 rappresenta una lista e le informazioni che si visualizzano.

- Nome e descrizione
- Numero di film contenuti
- Numero di follower se è una lista pubblica
- Badge "PUBBLICA" o "PRIVATA"
- Bottone per modificare una lista
- Bottone per eliminare una lista



Figura 3.4: Lista privata nella sezione "Le Mie Liste"

3.2.4 Follow e Copia Liste

L'utente può scegliere di:

Seguire una lista Cliccando sul pulsante "+ segui", la lista viene aggiunta alla sezione delle "Liste Seguite".

Copiare una lista Cliccando sul pulsante "+ Copia", l'applicazione crea una copia della lista nella collezione personale dell'utente, di visibilità privata, dove l'utente potrà a sua volta inserire altri film.

La figura 3.5 rappresenta una lista pubblica che un utente può seguire e copiare.

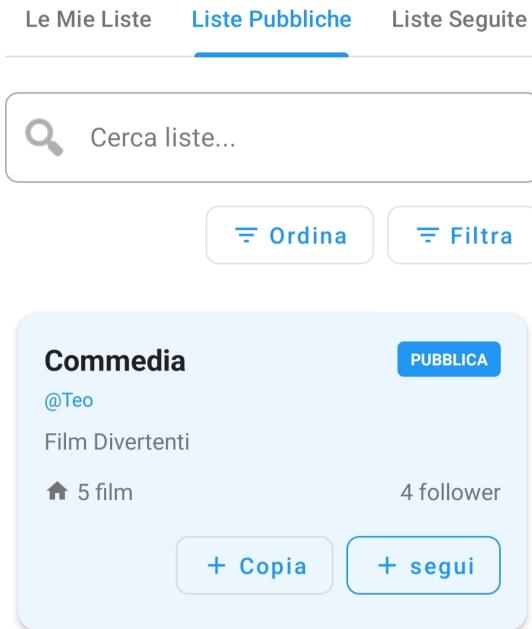


Figura 3.5: Lista pubblica nella sezione "Liste Pubbliche"

3.2.5 Filtraggio Avanzato

All'interno della schermata di dettaglio di ciascuna lista, è possibile filtrare i film per:

- **Nome:** ricerca testuale
- **Anno:** selezionare un'annata di uscita (l'anno è preso dai film della lista)
- **Regista:** selezionare uno specifico regista (il nome e cognome del regista è preso dai film della lista)
- **Genere:** selezionare il genere (il genere è preso dai film della lista)

La schermata di dettaglio di una lista, mostrata nella Figura 3.6, permette all'utente di visualizzare tutti i filtri possibili.

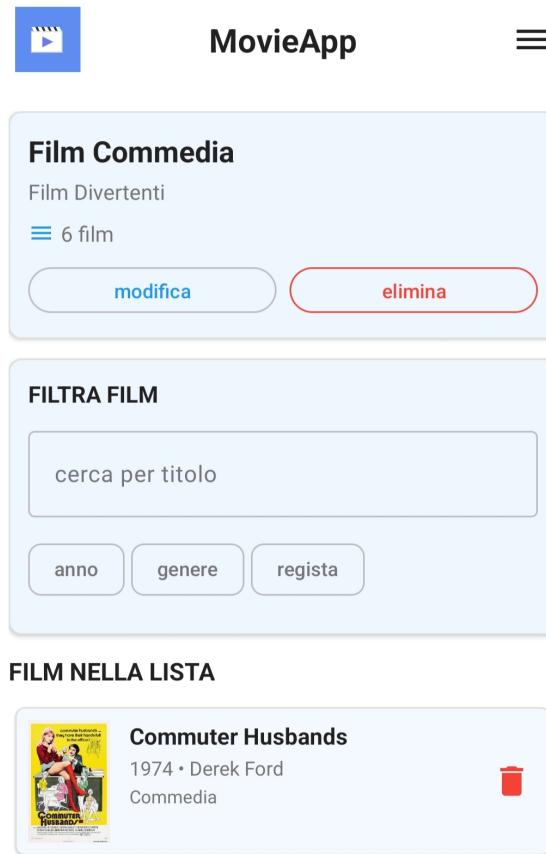


Figura 3.6: Schermata di dettaglio di una lista con filtri

I filtri si combinano in AND logico e l'UI si aggiorna in tempo reale tramite LiveData.

3.2.6 Notifiche Intelligenti

Le notifiche vengono inviate secondo la frequenza scelta dall'utente (giornaliera, settimanale o mensile).

La notifica, come mostrato anche nella Figura 3.7, include:

- Nome della lista
- Giorni mancanti alla data obiettivo
- Numero di film rimanenti da vedere
- Ritmo consigliato per finire di vedere i film nella lista entro la data obiettivo

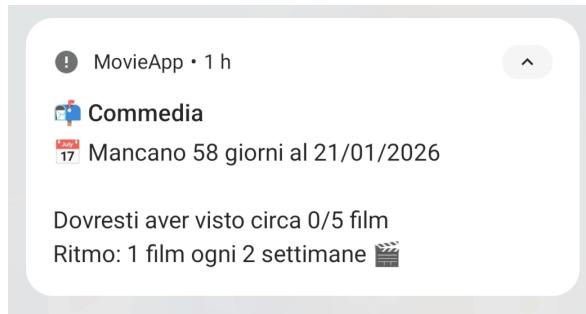


Figura 3.7: Notifica intelligente di una lista per un utente

3.3 Schermata Statistiche

La schermata **Statistiche** dell'applicazione fornisce dodici visualizzazioni interattive realizzate tramite **MPAndroidChart**, una libreria per la creazione di grafici su Android.

3.3.1 Principi di Design dei Grafici

Tutti i grafici condividono l'**interattività**, supportata da ciascuno tramite **tap**, ovvero toccare un elemento che mostra un **dialog** con la lista dei film correlati in quel grafico, **pinch**, ovvero lo zoom per esplorare dati dettagliati, e **pan**, ovvero lo scorrimento orizzontale per grafici con molti dati. I grafici condividono anche **statistiche sintetiche**, infatti ciascuno di essi è accompagnato da indicatori chiave come il genere più visto o il film di durata maggiore importato.

La schermata presenta un riepilogo generale della collezione, come illustrato nella Figura 3.8. Questo riepilogo mostra il numero di **film totali** importati nell'applicazione, il numero di **film visti**, il numero di **film da vedere** e la **durata totale** di tutti i film visti, espressa in ore e minuti.



Figura 3.8: Riepilogo numero di film importati

Questa panoramica consente all'utente di avere un quadro immediato della propria collezione cinematografica prima di esplorare le analisi dettagliate offerte dai grafici successivi.

3.3.2 Descrizione Dettagliata dei 12 Grafici

Grafico 1 – Distribuzione dei Generi (Donut Pie Chart)

Rappresenta la distribuzione percentuale dei generi cinematografici della collezione personale. Ogni fetta è selezionabile per visualizzare l'elenco dei film appartenenti a quel genere. La Figura 3.9 illustra il grafico dei generi.

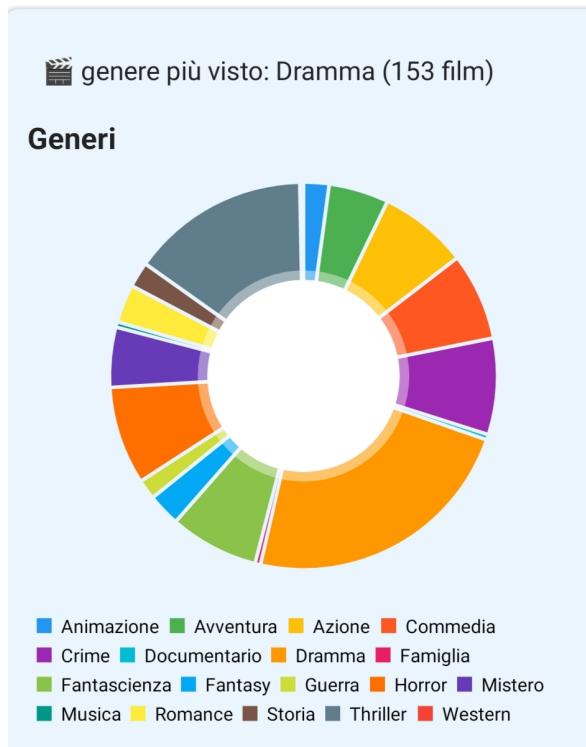


Figura 3.9: Grafico Distribuzione Generi

Grafico 2 – Distribuzione Film per Anno di Uscita (Bar Chart Verticale)

Mostra quanti film sono stati prodotti in ciascun anno. La visualizzazione è scrollabile orizzontalmente, utile per individuare pattern temporali ricorrenti. La Figura 3.10 mostra la distribuzione temporale dei film.

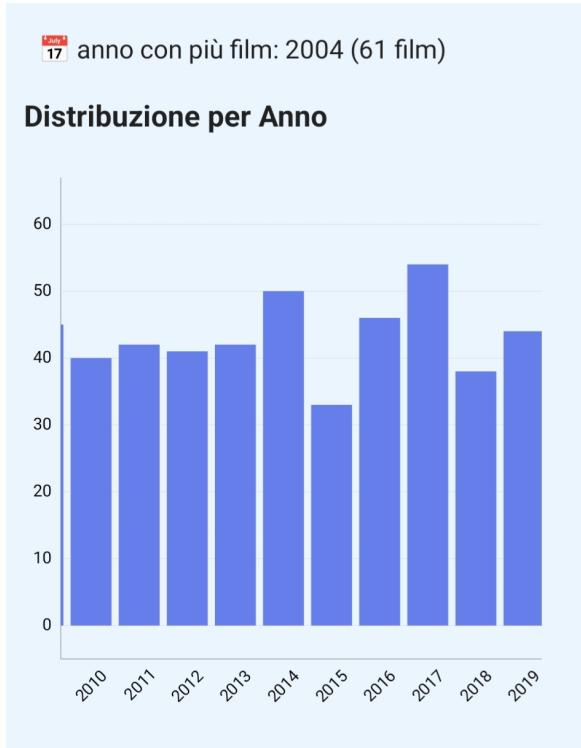


Figura 3.10: Grafico Distribuzione Film per Anno

Grafico 3 – Top Registi (Bar Chart Verticale)

Elenca i registi rappresentati nella collezione e mostra, per ciascuno, il numero di film diretti. Selezionando una barra viene mostrata la relativa filmografia presente nell'archivio personale. La Figura 3.11 presenta i registi più rappresentati nella collezione.

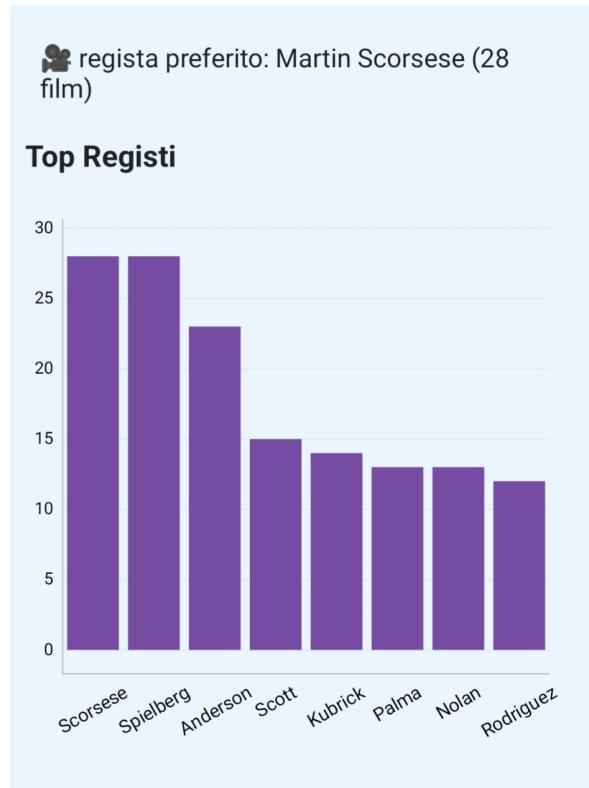


Figura 3.11: Grafico Top Registi

Grafico 4 – Top Attori (Bar Chart Verticale)

Elenca gli attori rappresentati nella collezione e mostra, per ciascuno, il numero di film a cui hanno partecipato. Selezionando una barra viene mostrata la relativa filmografia presente nell'archivio personale. La Figura 3.12 evidenzia gli attori più presenti nella collezione.

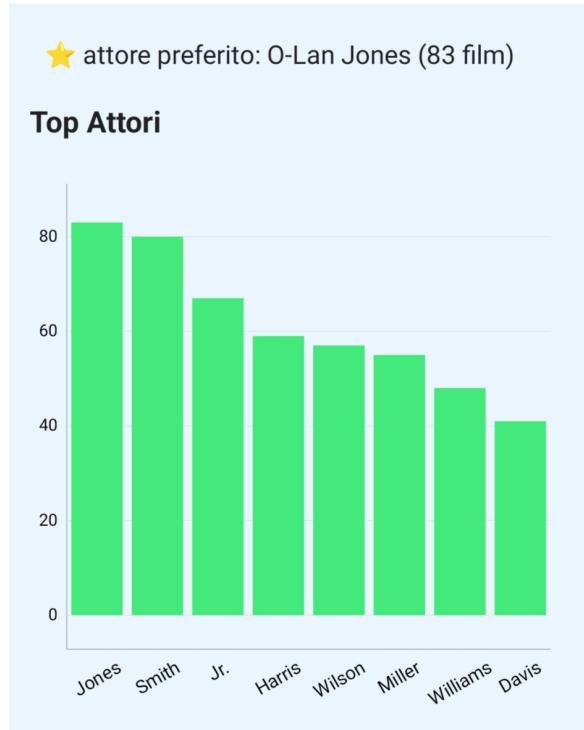


Figura 3.12: Grafico Top Attori

Grafico 5 – Distribuzione delle Valutazioni TMDB (Bar Chart Verticale)

Iistogramma con la quantità di film per fascia di voto (1–10). Permette una lettura immediata della qualità media della collezione. Selezionando una barra si accede ai film con quella valutazione. La Figura 3.13 mostra la distribuzione dei rating TMDB.

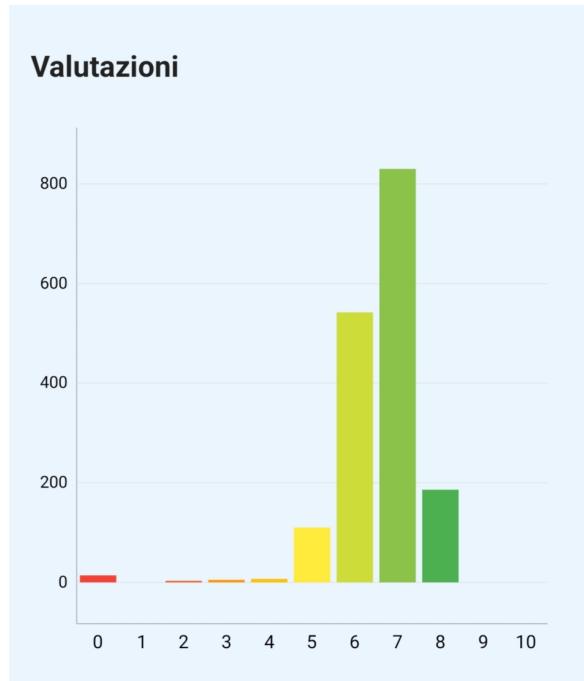


Figura 3.13: Grafico Distribuzione Valutazioni TMDB

Grafico 6 – Paesi di Produzione (Bar Chart Verticale)

Visualizza la distribuzione dei film per paese di produzione, evidenziando la varietà geografica della collezione. La Figura 3.14 illustra i principali paesi produttori.

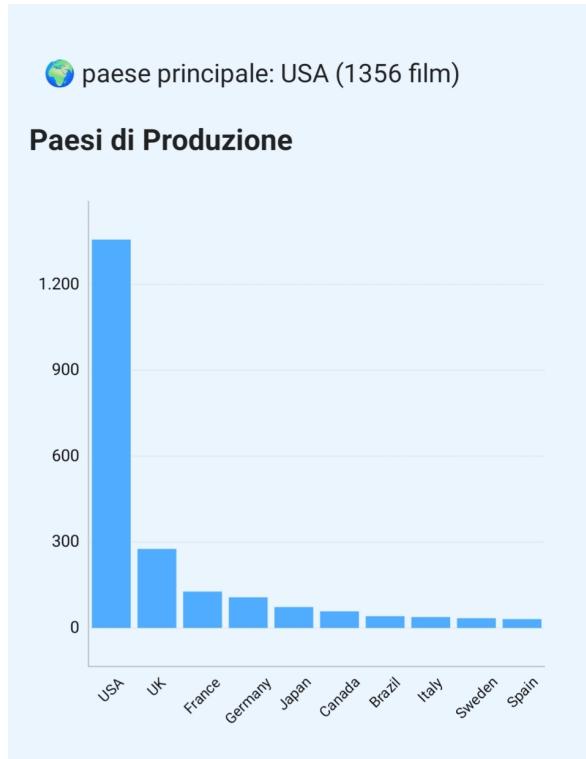


Figura 3.14: Grafico Paesi di Produzione

Grafico 7 – Film per Decennio (Bar Chart Verticale)

Aggrega i dati del grafico 2 su base decennale (per esempio 1990s, 2000s, 2010s). Fornisce una visione macroscopica delle preferenze temporali dell'utente. La Figura 3.15 presenta la distribuzione per decade.

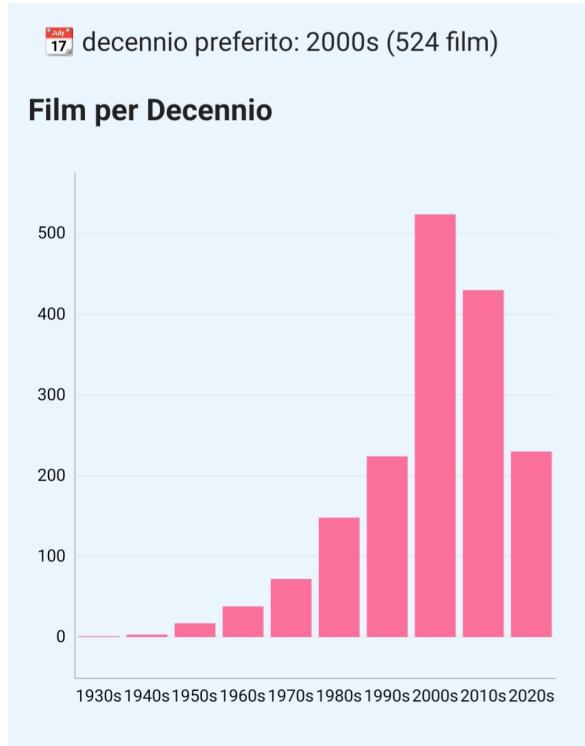


Figura 3.15: Grafico Film per Decennio

Grafico 8 – Combinazioni di Generi (Bar Chart Verticale)

Mostra le coppie di generi che ricorrono più spesso. È utile per identificare trend ricorrenti nelle preferenze dell'utente. La Figura 3.16 evidenzia le combinazioni più frequenti.



Figura 3.16: Grafico Combinazioni di Generi

Grafico 9 – Distribuzione della Durata dei Film (Bar Chart Verticale)

Raggruppa i film in fasce di durata (0–60, 60–90, 90–120, 120–150, 150–180 e 180+ minuti). Consente anche, tra i film importati, di individuarne quello di maggiore durata. La Figura 3.17 mostra la distribuzione delle durate dei film importati.

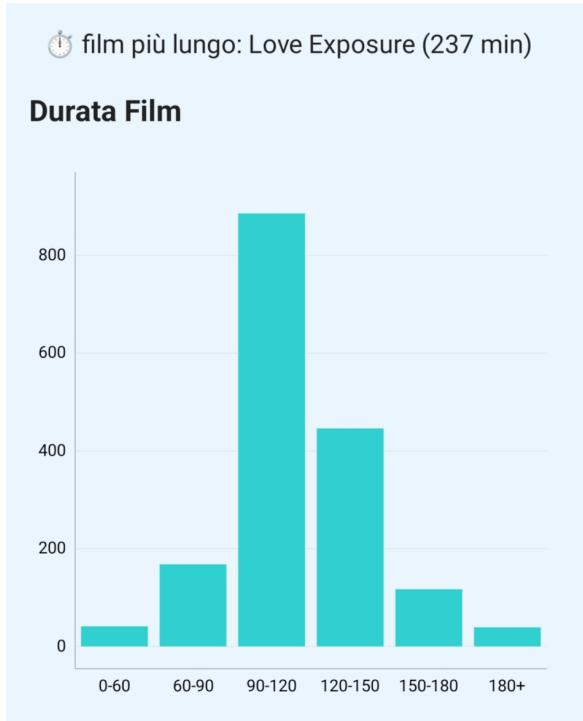


Figura 3.17: Grafico Distribuzione Durata Film

Grafico 10 – Lingue Originali dei Film (Donut Pie Chart)

Rappresenta la distribuzione della lingua originale di produzione, evidenziando la diversità linguistica della collezione. La Figura 3.18 illustra le lingue più rappresentate.

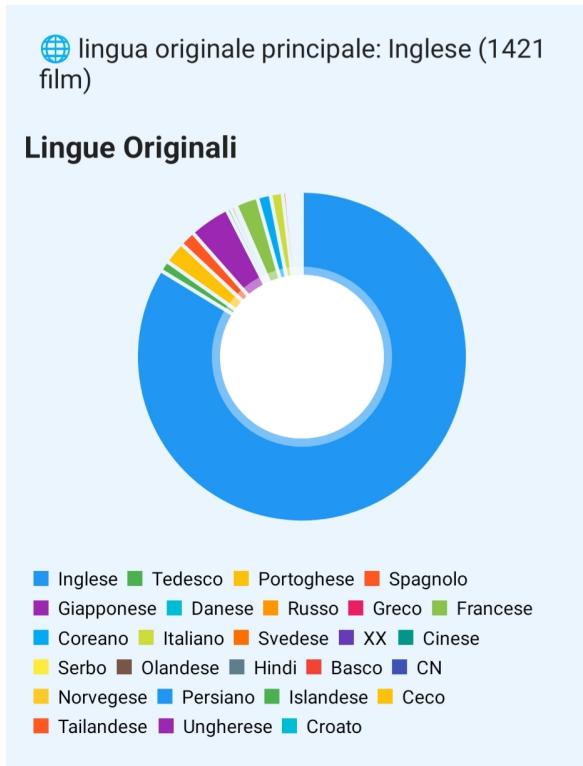


Figura 3.18: Grafico Lingue Originali

Grafico 11 – Film Popolari vs Film di Qualità (Scatter Plot)

Il grafico rappresenta ciascun film come un punto all'interno di un piano cartesiano, in cui l'asse X indica il valore di popolarità e l'asse Y il rating medio. La visualizzazione consente di analizzare la relazione tra successo commerciale e l'apprezzamento critico. La Figura 3.19 mostra lo scatter plot popolarità-qualità.

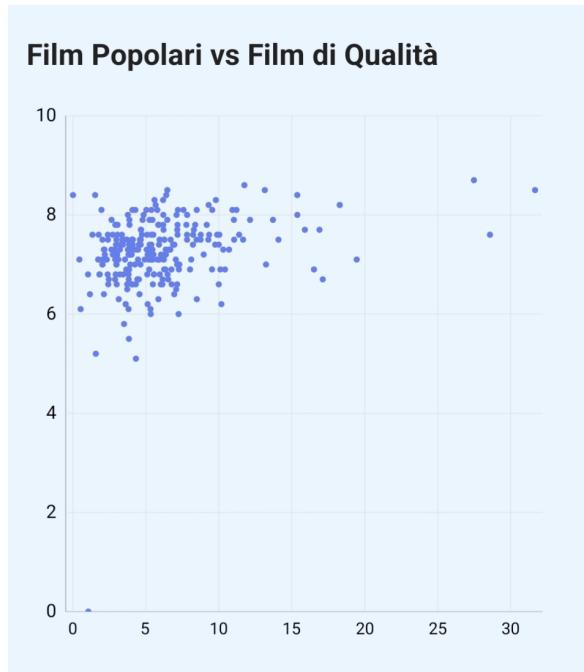


Figura 3.19: Grafico Popolarità vs Qualità

Grafico 12 – Evoluzione Popolarità nel Tempo (Line Chart)

Mostra l'evoluzione della popolarità media attuale su TMDB dei film presenti nella collezione dell'utente, raggruppati per decade di uscita. Ogni punto del grafico rappresenta la media della popolarità dei film di quel decennio presenti nella collezione. Ad esempio, se i film del decennio 1990s mostrano un valore elevato, significa che quei film sono attualmente molto ricercati e popolari tra gli utenti di TMDB. La Figura 3.20 presenta l'andamento temporale della popolarità.



Figura 3.20: Grafico Evoluzione Popolarità nel Tempo

3.4 Sicurezza

3.4.1 Autenticazione e Autorizzazione

L'applicazione implementa un sistema di autenticazione basata su JWT per garantire sicurezza nelle comunicazioni.

3.4.2 Protezione dei Dati Sensibili

Le password degli utenti non vengono mai salvate in chiaro nel database. Il backend utilizza la libreria `bcrypt` per generare hash sicuri. Quando un utente si registra o effettua il login, il backend verifica le credenziali confrontando l'hash della password con quello salvato nel database.

Il seguente estratto di codice mostra il metodo `login` nel file `auth.service`. Il metodo cerca l'utente nel database tramite `email`: se non viene trovato, solleva un'eccezione di credenziali non valide. Successivamente confronta la password fornita con l'hash salvato utilizzando `bcrypt.compare`, che verifica se la password inserita è corretta. Solo se la verifica ha successo, viene generato un token JWT tramite il metodo `generateToken` e restituito insieme ai dati dell'utente.

```

1  async login(dto: LoginDto): Promise<AuthResponse> {
2      // cerca utente per email
3      const user = await this.userRepository.findOne({

```

```
4         where: { email: dto.email }
5     );
6
7     if (!user) {
8         throw new UnauthorizedException('credenziali non
9             valide');
10    }
11
12    // verifica password confrontando hash bcrypt
13    const isPasswordValid = await bcrypt.compare(dto.password
14        , user.password);
15
16    if (!isPasswordValid) {
17        throw new UnauthorizedException('credenziali non
18            valide');
19    }
20
21    // genera jwt token per sessione
22    const token = this.generateToken(user);
23
24
25    return { access_token: token, user: user };
26 }
```

Listing 3.5: Verifica password nel login (auth.service.ts)

Questo approccio garantisce, che in caso di violazione del database, le password degli utenti rimangano protette grazie all'uso di algoritmi di hashing.

Capitolo 4

Valutazione

In questo capitolo viene presentata la valutazione dell’usabilità di MOVIEAPP attraverso un questionario somministrato a 35 utenti di differente età. La valutazione si basa sul questionario **PSSUQ** [6] (**Post-Study System Usability Questionnaire**). L’obiettivo è misurare il grado di soddisfazione degli utenti, identificare i punti di forza dell’applicazione e raccogliere suggerimenti per miglioramenti futuri.

4.1 Il Questionario PSSUQ

Il **PSSUQ** [6] è uno strumento di valutazione standardizzato sviluppato da IBM nel 1995, specificamente per misurare la soddisfazione degli utenti, dopo l’utilizzo di un sistema o applicazione. Questo questionario è stato progettato per fornire una misurazione affidabile dell’usabilità percepita attraverso tre categorie principali: l’utilità del sistema (System Usefulness), la qualità delle informazioni fornite (Information Quality) e la qualità dell’interfaccia (Interface Quality).

4.2 Metodologia di Valutazione

4.2.1 Partecipanti

La valutazione è stata condotta tramite un Modulo Google a **35 utenti**. I partecipanti che hanno svolto il modulo presentano le seguenti caratteristiche demografiche:

Distribuzione età utenti

- Sotto 18 anni: 8 utenti (23%)
- 18-25 anni: 13 utenti (37%)
- 26-35 anni: 5 utenti (14%)

- 36-45 anni: 3 utenti (9%)
- Oltre 45 anni: 6 utenti (17%)

Esperienza applicazioni simili

- Utilizzano IMDb: 8 utenti (23%)
- Utilizzano Letterboxd: 5 utenti (14%)
- Utilizzano altre applicazioni: 10 utenti (29%)
- Non utilizzano applicazioni per gestire film: 12 utenti (34%)

Il campione risulta quindi **eterogeneo**, con una significativa presenza sia di utenti esperti di applicazioni cinematografiche (66%) sia di utenti nuovi al settore (34%).

Limitazioni della valutazione È importante sottolineare che i valutatori erano principalmente conoscenti e compagni di corso dello sviluppatore, consapevoli della natura del progetto come elaborato di tesi.

4.2.2 Strumento di Valutazione

Il questionario si basa su **PSSUQ**, un questionario standardizzato sviluppato per misurare la soddisfazione degli utenti, composto da 16 domande suddivise in tre categorie:

1. **System Usefulness**: domande 1–6.
2. **Information Quality**: domande 7–12.
3. **Interface Quality**: domande 13–16

Ogni domanda utilizza una **scala Likert** a 5 punti:

- 1 = Fortemente in disaccordo
- 2 = In disaccordo
- 3 = Neutrale
- 4 = D'accordo
- 5 = Fortemente d'accordo

Gli utenti hanno eseguito questa procedura prima di poter procedere alla valutazione di MOVIEAPP:

1. **Installazione dell'applicazione:** download e installazione di MOVIEAPP sul personale dispositivo Android.
2. **Esecuzione dei task:** completamento di tre task predefiniti che coprono le funzionalità principali dell'applicazione.
3. **Compilazione del questionario:** valutazione dell'esperienza tramite 16 domande PSSUQ e domande aperte per fornire un feedback qualitativo dell'applicazione.

4.3 Task di Valutazione

Sono stati definiti tre task delle funzionalità principali di MOVIEAPP.

4.3.1 Task #1: Caricamento e Analisi Dati

Importare la tua lista di film visti (watched) e da vedere (watchlist) tramite file CSV provenienti dalle piattaforme IMDb o Letterboxd e navigare nella sezione "Statistiche" esplorando i grafici e le analisi generate.

4.3.2 Task #2: Creazione Liste

Nella sezione "Liste" creare una lista personalizzata dei propri film preferiti aggiungendoci almeno 3 film preferiti.

4.3.3 Task #3: Funzionalità Social

Esplorare le "Liste Pubbliche" create da altri utenti all'interno della sezione "Liste", iniziare a seguire una lista di interesse e copiarla nella propria lista personale. Aprire la lista copiata e aggiungerci almeno un altro film.

4.4 Risultati Quantitativi

4.4.1 Punteggi Domande 1-16

La Tabella 4.1 mostra il punteggio medio alle sedici domande del questionario.

Tabella 4.1: Punteggi delle sedici domande PSSUQ

Domanda	Media
1. Nel complesso, sono soddisfatto di quanto sia facile usare questo sistema.	4.34
2. È stato semplice usare questo sistema.	4.34
3. Sono stato in grado di completare i task e gli scenari rapidamente usando questo sistema.	4.43
4. Mi sono sentito a mio agio nell'usare questo sistema.	4.46
5. È stato facile imparare a usare questo sistema.	4.51
6. Credo che potrei diventare produttivo rapidamente usando questo sistema.	4.31
7. Il sistema ha fornito messaggi di errore che mi hanno detto chiaramente come risolvere i problemi.	3.89
8. Ogni volta che ho fatto un errore usando il sistema, ho potuto recuperare facilmente e rapidamente.	4.14
9. Le informazioni (come messaggi sullo schermo) fornite con questo sistema erano chiare.	4.46
10. Era facile trovare le informazioni di cui avevo bisogno.	4.31
11. Le informazioni sono state efficaci nell'aiutarmi a completare i task e gli scenari.	4.31
12. L'organizzazione delle informazioni sullo schermo del sistema era chiara.	4.49
13. L'interfaccia di questo sistema era piacevole.	4.49
14. Mi è piaciuto usare l'interfaccia di questo sistema.	4.51
15. Questo sistema ha tutte le funzioni e le capacità che mi aspetto abbia.	4.23
16. Nel complesso, sono soddisfatto di questo sistema.	4.40

Il punteggio leggermente inferiore nella domanda 7 (valutazione media 3.89) ha suggerito un margine di miglioramento nella gestione e nella comunicazione degli errori, sebbene il valore sia comunque positivo. Le domande 13 e 14 ottengono entrambe valutazioni eccellenti (4.49 e 4.51), confermando l'efficacia delle scelte grafiche effettuate.

La Figura 4.1 mostra graficamente i punteggi ottenuti per ciascuna domanda.

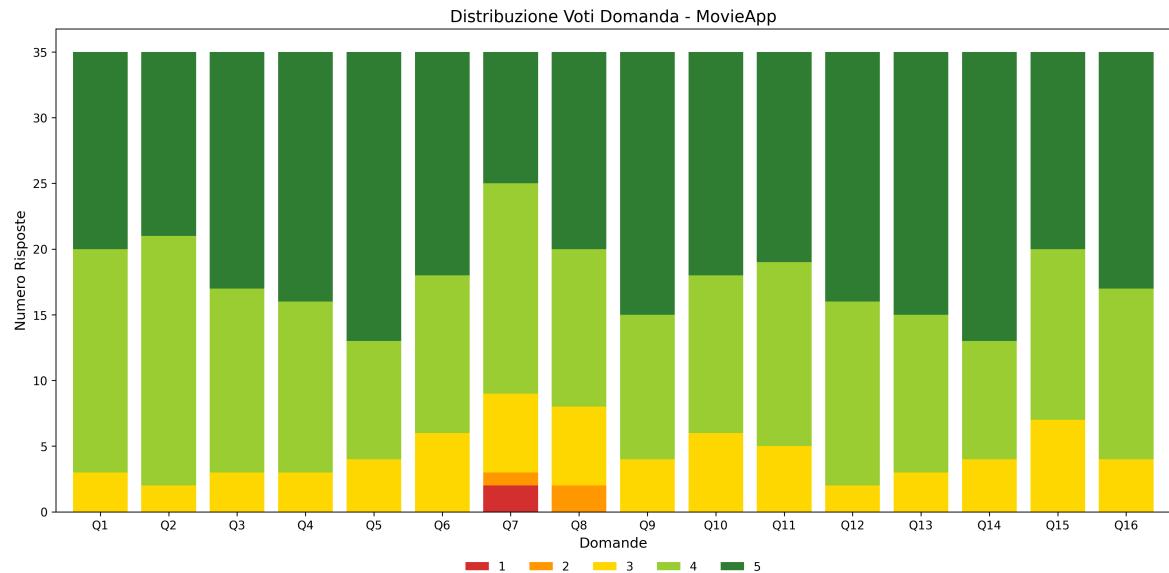


Figura 4.1: Grafico valutazione domande PSSUQ

La Figura 4.2 mostra graficamente i boxplot con il range di valutazione ottenuto per ciascuna domanda.

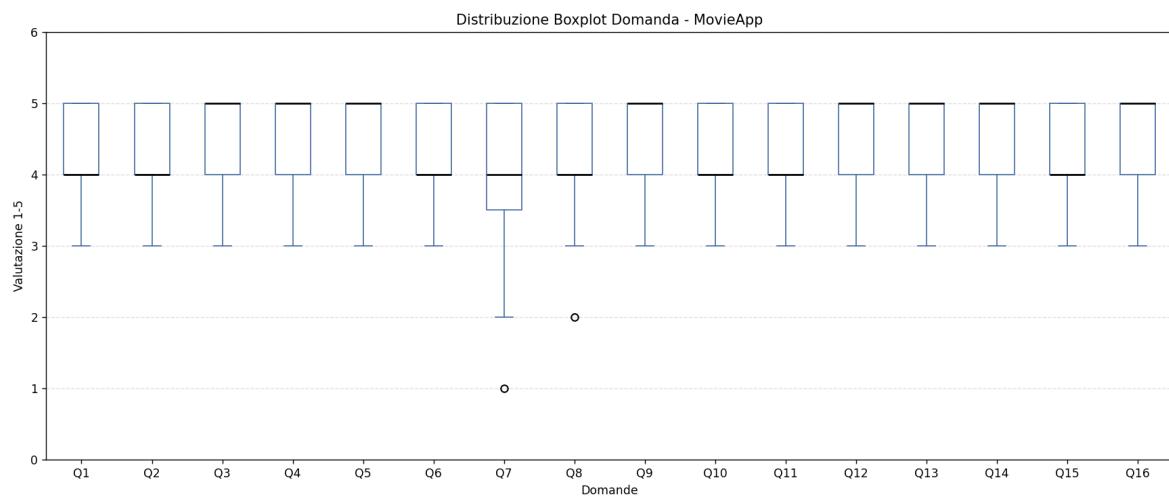


Figura 4.2: Grafico boxplot domande PSSUQ

La Figura 4.3 mostra graficamente i punteggi medi PSSUQ, aggregati per le tre categorie, suddivisi per tipologia di utente. Gli utenti sono stati classificati in base alla loro esperienza: utenti IMDb, utenti Letterboxd, utenti con esperienza di altre applicazioni e utenti senza esperienza.

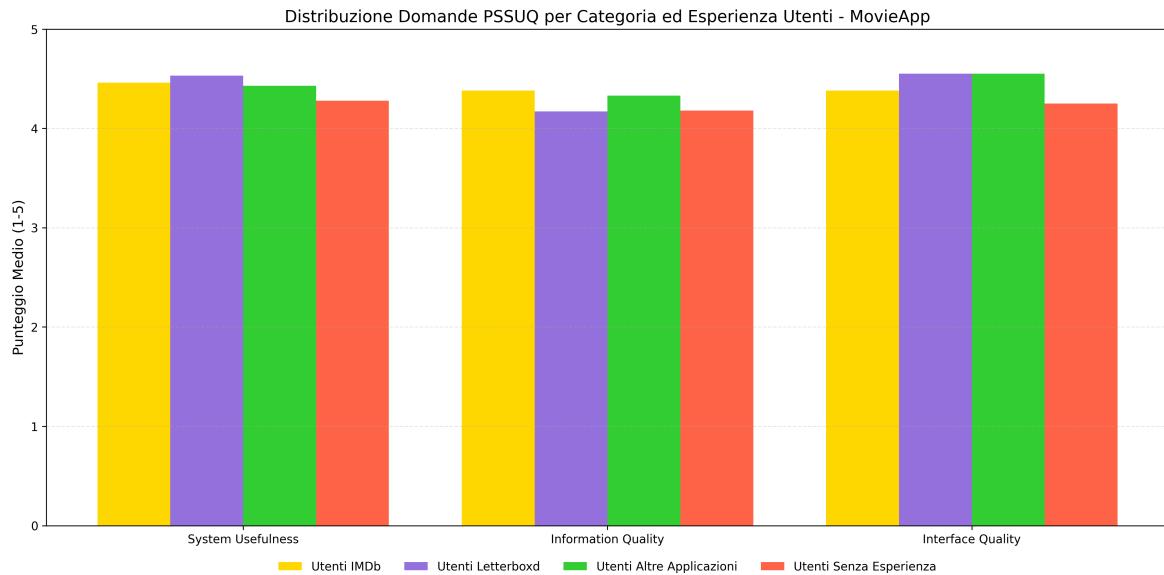


Figura 4.3: Grafico categorie PSSUQ per esperienza utenti

4.4.2 Riepilogo Punteggi Medi

La Tabella 4.2 riassume i punteggi medi per ciascuna categoria.

Tabella 4.2: Riepilogo Punteggi Medi PSSUQ

Dimensione	Punteggio Medio
System Usefulness (1–6)	4.40
Information Quality (7–12)	4.27
Interface Quality (13–16)	4.41
Media Generale	4.36

Il punteggio medio generale di **4.36 su 5** indica un'ottima usabilità e soddisfazione utente complessiva.

4.5 Feedback Qualitativo

Oltre ai punteggi quantitativi, gli utenti hanno fornito un feedback qualitativo attraverso tre domande aperte.

4.5.1 Aspetti Apprezzati

Analizzando le risposte alla domanda "Cosa ti è piaciuto di più dell'app?", emergono le seguenti risposte.

La **semplicità e intuitività** dell'applicazione è l'aspetto più citato, con nove menzioni da parte degli utenti. Gli utenti hanno apprezzato particolarmente quanto sia facile importare i film e la grafica semplice e chiara. Come evidenziato da un utente,

l'app è *"di facile utilizzo ma al contempo è interessante e completa"*, mentre un altro utente sottolinea che *"l'applicazione può essere utilizzata da utenti di tutte le età"*.

Il **sistema di gestione liste** ha ricevuto otto menzioni. Gli utenti hanno apprezzato la possibilità di creare liste personalizzate in modo semplice e la possibilità di renderle pubbliche, trovando molto interessante l'aspetto social dell'applicazione.

La **sezione statistiche e i grafici interattivi** rappresentano un aspetto citato con sette menzioni. Un utente ha commentato che *"i grafici sono dettagliati e mi hanno fatto scoprire cose sui miei gusti che non sapevo"*, mentre un altro utente ha evidenziato come questa sia una funzionalità *"che non avevo mai visto in altre applicazioni simili"*, confermando una caratteristica particolare legata a MOVIEAPP.

Infine, l'**interfaccia grafica** ha ricevuto tre menzioni positive, con utenti che hanno apprezzato il design piacevole dell'applicazione.

4.5.2 Funzionalità Future

Alla domanda "Quali funzionalità vorresti vedere aggiunte in futuro?", è emersa una richiesta molto interessante e univoca da più utenti: il **sistema di recensioni e commenti**, menzionato da otto utenti. Gli utenti vorrebbero poter scrivere recensioni dei film. Un utente ha suggerito la possibilità di *"creare dibattiti stile forum"*, evidenziando il desiderio di rafforzare l'aspetto social dell'applicazione.

Tre utenti desiderano l'**integrazione di contenuti multimediali**, come poter visualizzare il trailer dei film direttamente dall'applicazione, così da facilitare la scelta di quale film guardare.

Due utenti hanno consigliato la possibilità di implementare un **sistema di votazione personale** dei film visti, per creare successivamente un grafico basato sui punteggi di valutazione dei film di un determinato utente.

4.5.3 Ulteriori Commenti e Suggerimenti

Al quesito "Ulteriori commenti, suggerimenti o osservazioni" è stato chiesto un feedback libero, e la maggior parte dei commenti è stata positiva. Gli utenti hanno definito l'app *"molto semplice da utilizzare"* e *"bella e facile da usare anche per chi non è esperto"*.

Un utente ha segnalato un problema tecnico relativo alla renderizzazione dell'interfaccia su un dispositivo Android, dove la parte alta dell'applicazione entra in contrasto con il pannello delle notifiche. Questo feedback evidenzia la necessità di testare l'applicazione su una gamma più ampia di dispositivi Android, al fine di garantirne la compatibilità con diversi modelli e dimensioni.

Capitolo 5

Conclusioni e Sviluppi Futuri

In questo capitolo vengono presentate le considerazioni finali dell’elaborato, analizzando i risultati ottenuti, i punti di forza e le limitazioni di MOVIEAPP, proponendo infine possibili sviluppi futuri per estendere e migliorare questa applicazione.

5.1 Riepilogo del Lavoro Svolto

Il presente elaborato ha descritto la progettazione, l’implementazione e la valutazione di MOVIEAPP, un’applicazione mobile progettata solo su Android per la gestione e l’analisi di liste cinematografiche.

Il lavoro è articolato nelle seguenti fasi:

1. **Analisi dei requisiti e progettazione:** sono stati analizzati i requisiti funzionali e non funzionali, definendo un’architettura client-server distribuita con backend in NestJS, database tramite PostgreSQL e client Android basato sul pattern MVVM.
2. **Implementazione backend:** è stato sviluppato un backend scalabile e modulare, con gestione dell’autenticazione JWT, integrazione con l’API di TMDB, generazione statistiche e comunicazione real-time tramite WebSocket.
3. **Implementazione del client Android:** è stata realizzata un’interfaccia utente moderna con visualizzazione interattiva dei grafici tramite MPAndroidChart.
4. **Valutazione con utenti reali:** l’applicazione è stata testata da 35 utenti attraverso tre task e valutata tramite questionario PSSUQ, ottenendo risultati positivi.

5.2 Punti di Forza

I punti di forza di MOVIEAPP emersi durante lo sviluppo e la valutazione sono:

- **Analisi Visiva:** i dodici grafici interattivi rappresentano una funzionalità che differenzia MOVIEAPP dalle piattaforme già esistenti, come IMDb e Letterboxd, che non offrono strumenti di analisi visiva così avanzati.
- **Integrazione Piattaforme Esistenti:** l'applicazione si integra in modo efficace con IMDb e Letterboxd, permettendo agli utenti di importare facilmente i propri dati senza dover ripartire da zero.
- **Design Moderno:** l'interfaccia basata su Material Design 3 è esteticamente gradevole ed è accessibile anche ad utenti con esperienze diverse.
- **Architettura solida e manutenibile:** l'uso di pattern consolidati MVVM e di tecnologie come NestJS e TypeORM facilita la manutenzione e l'estensione del codice.

5.3 Limitazioni

Nonostante i risultati positivi, MOVIEAPP presenta alcune limitazioni che rappresentano opportunità di miglioramento futuro:

- **Dipendenza da TMDB:** l'arricchimento dei film dipende interamente dalla disponibilità e dalla qualità dei dati su TMDB. Il processo di arricchimento dei film si rivela molto lento, in quanto è presente un rate-limiting imposto a 40 richieste ogni secondo. Un altro limite è rappresentato dall'impossibilità di arricchire i film in assenza di connessione ad internet.
- **Funzionalità Social Limitate:** le funzionalità social attuali (follow e copia liste) sono semplici. Non è possibile commentare le liste, recensire i film al suo interno e mettere like ad una lista.
- **Solo Android:** l'applicazione al momento è disponibile solo per Android. Il vantaggio di aver scelto questa struttura di backend è che, in futuro, sarà possibile estendere l'applicazione anche per iOS e WEB, in quanto il backend è univoco.

5.4 Sviluppi Futuri

Sulla base dei risultati ottenuti, del feedback degli utenti e delle limitazioni identificate, sono stati individuati i seguenti sviluppi futuri:

- **Versione iOS e WEB:** sviluppare l'applicazione MOVIEAPP anche per iOS e WEB. Il backend è già pronto per supportare client multipli.

- **Applicazione Multilingua:** implementare il supporto multilingua in modo che l'applicazione possa essere utilizzata da persone di diverse nazionalità.
- **Recensioni e Commenti:** permettere agli utenti di scrivere recensioni testuali sui film e commentare le liste pubbliche, aggiungendo anche la possibilità all'utente di segnalare la visualizzazione del film di una lista con una spunta.
- **Esportazione Dati:** implementare la funzionalità di esportare le statistiche in formato PDF o come immagini PNG/JPG.

5.5 Conclusione

MOVIEAPP ha raggiunto gli obiettivi prefissati, offrendo agli appassionati di cinema uno strumento moderno per gestire e analizzare le proprie collezioni cinematografiche. La valutazione con 35 utenti ha confermato un alto livello di facilità di utilizzo e una soddisfazione generale. Il progetto dimostra come tecnologie moderne e un'attenzione costante all'esperienza utente possano generare un'applicazione di qualità, che offre un valore concreto agli utenti finali, e che possa competere con piattaforme affermate nel mercato.

Bibliografia

- [1] TMDB. The movie database (tmdb). URL <https://www.themoviedb.org/>. Accessed: 28-11-2025.
- [2] IMDb. Internet movie database (imdb). URL <https://www.imdb.com/it/>. Accessed: 28-11-2025.
- [3] Letterboxd. Letterboxd - social film discovery. URL <https://letterboxd.com/>. Accessed: 28-11-2025.
- [4] JustWatch. Justwatch - the streaming guide. URL <https://www.justwatch.com/it>. Accessed: 28-11-2025.
- [5] NestJS. Nestjs - a progressive node.js framework. URL <https://nestjs.com/>. Accessed: 02-12-2025.
- [6] Trymata. The pssuq and usability testing, 2023. URL <https://trymata.com/learn/pssuq/>. Accessed: 26-11-2025.

Ringraziamenti

Giunto al termine di questo percorso, desidero esprimere la mia gratitudine nei confronti di tutte le persone che hanno contribuito, in modi diversi, alla realizzazione di questo mio traguardo.

Un sentito ringraziamento va ai miei relatori, in particolare a Ivan Dmitry Ribeiro Zyrianoff che ogni lunedì mi ha supportato con un meeting settimanale verso il raggiungimento dell'obiettivo, e al Prof. Federico Montori per la sua disponibilità.

Ringrazio di cuore i miei genitori, Patrizia e Stefano, che mi sono sempre stati accanto, offrendomi supporto nei momenti difficili e permettendomi di arrivare a questo traguardo. Un ringraziamento speciale va anche ai miei nonni, Amerigo, Miranda e Franca, per aver sempre creduto in me.

Un grazie ai miei amici più stretti, con i quali ho condiviso momenti indimenticabili. Un particolare ringraziamento va a Matteo, il mio confidente, la persona su cui so di poter sempre contare.

Grazie anche alla mia seconda famiglia: gli arbitri di calcio, in particolare a tutti i membri della FAF, con cui ho condiviso momenti che hanno arricchito la mia vita dentro e fuori dal terreno di gioco.

Infine, il ringraziamento più importante va a Chiara, la persona che ha reso questo percorso più leggero. Chiara è sempre stata presente nei momenti di difficoltà, pronta ad incoraggiarmi a non mollare mai. Chiara è la mia metà, il mio raggio di luce e il mio rifugio quando tutto sembra difficile. Per tutto questo, per ogni sorriso, per ogni parola di conforto e per essere semplicemente te stessa, grazie dal profondo del mio cuore.