

# Università degli Studi di Torino

Dipartimento di informatica



Progetto IUM-TWEB

## Progettazione e sviluppo di un applicativo web cinematografico

Gruppo:

**Studenti:** Federico Verra, Federico Raso

---

ANNO ACCADEMICO 2024/2025

# Indice

|          |                                      |          |
|----------|--------------------------------------|----------|
| <b>1</b> | <b>Introduzione</b>                  | <b>3</b> |
| 1.1      | Filmoteca . . . . .                  | 3        |
| <b>2</b> | <b>Implementazione</b>               | <b>4</b> |
| 2.1      | Interfaccia . . . . .                | 4        |
| 2.2      | Database . . . . .                   | 4        |
| 2.2.1    | PostgreSQL . . . . .                 | 4        |
| 2.2.2    | MongoDB . . . . .                    | 5        |
| 2.2.3    | Caricamento dei dati . . . . .       | 5        |
| 2.2.4    | Comunicazione tra i Server . . . . . | 5        |
| 2.3      | Socket.io . . . . .                  | 6        |
| <b>3</b> | <b>Riflessioni sul lavoro svolto</b> | <b>7</b> |
| 3.1      | Divisione del Lavoro . . . . .       | 7        |
| 3.2      | Considerazioni finali . . . . .      | 7        |

# Capitolo 1

## Introduzione

### 1.1 Filmoteca

Il progetto Filmoteca nasce con l'obiettivo di progettare e sviluppare un'applicazione web dedicata al mondo del cinema, con la duplice finalità di realizzare un prodotto funzionale e di approfondire le nostre competenze nell'ambito dello sviluppo web. Il lavoro si è sviluppato a partire da un'idea chiara e ben definita, che ci ha guidato fin dalle prime fasi di progettazione e ha reso più agevole l'implementazione delle varie componenti del sito.

Nella fase iniziale ci siamo concentrati sull'analisi della struttura generale del progetto. Questo momento di riflessione e organizzazione si è rivelato cruciale per poter procedere con una visione chiara degli obiettivi da raggiungere. Il sito è stato pensato per articolarsi in due viste principali: una homepage, che offre una panoramica iniziale e consente la ricerca dei film tramite una barra di navigazione, e una pagina di dettaglio dedicata a ciascun film. Quest'ultima presenta tutte le informazioni specifiche sull'opera cinematografica e include una cronologia delle recensioni lasciate dagli utenti. Inoltre, ogni film è associato a una chat dedicata, che permette agli utenti di interagire tra loro e condividere opinioni e passioni comuni.

Dal punto di vista strutturale, ogni pagina del sito è composta da una navbar superiore e da un footer, elementi costanti che favoriscono la navigazione e l'uniformità grafica. Il corpo centrale della pagina è invece riservato alla vista principale, che varia in base al contenuto selezionato, offrendo così un'esperienza dinamica e focalizzata.

Nelle pagine che seguono verranno analizzati nel dettaglio i diversi aspetti che hanno reso possibile la realizzazione del progetto, attraverso un percorso articolato in più sezioni, ciascuna dedicata a un'area specifica di progettazione o sviluppo. L'intento è quello di offrire una panoramica completa, coerente e approfondita del lavoro svolto, mettendo in luce l'approccio teorico, le scelte tecniche adottate, e le difficoltà incontrate lungo il percorso.

# Capitolo 2

## Implementazione

### 2.1 Interfaccia

Ogni pagina del sito è stata progettata secondo una struttura coerente, composta da una barra di navigazione (navbar) nella parte superiore e da un footer in chiusura, elementi costanti che garantiscono uniformità e facilità di navigazione. Il corpo centrale della pagina, invece, è dinamico e si adatta ai contenuti specifici visualizzati, variando in base alla sezione a cui si accede. Per lo sviluppo dell'interfaccia e delle funzionalità sono stati utilizzati principalmente quattro strumenti: **HTML**, che ha fornito la struttura base delle pagine; **CSS**, che ha permesso di definire lo stile e l'aspetto grafico degli elementi; **Bootstrap** ([1]), il framework che ha facilitato la creazione di un layout responsive e ordinato; e infine **JavaScript**, essenziale per introdurre interazioni dinamiche all'interno del sito, migliorando l'esperienza utente attraverso comportamenti e funzioni attive sulle diverse componenti della pagina.([3])

### 2.2 Database

Nel progetto sono stati implementati due database distinti: uno basato su PostgreSQL e l'altro su MongoDB, ciascuno con un ruolo specifico in base alla natura dei dati da gestire.

#### 2.2.1 PostgreSQL

Il database relazionale PostgreSQL è stato utilizzato, conformemente ai requisiti di progetto, per l'archiviazione di dati meno dinamici. In particolare, esso contiene informazioni relative ad attori, paesi, membri della crew, generi, film, premi Oscar, poster e date di uscita.

Le tabelle sono attualmente indipendenti tra loro, in quanto non sono state definite relazioni esplicite mediante chiavi esterne. Sebbene sia stata inizialmente considerata l'integrazione tramite *foreign key*, si sono manifestate criticità legate alla gestione degli identificativi scelti, motivo per cui si è optato per l'utilizzo di query più articolate, mantenendo le tabelle separate.

Alcune tabelle, come quella dedicata alle pubblicazioni dei film, non sono state effettivamente utilizzate durante lo sviluppo, ma sono state comunque mantenute nel database in previsione di possibili estensioni future.

L'accesso al database avviene attraverso un server sviluppato in Java con il framework Spring Boot, che si occupa della gestione delle richieste provenienti dal server principale (*MainServer*). Il codice del server è organizzato in *package* distinti, ciascuno associato a una specifica entità del database. Le classi *Service* e *Controller* sono presenti unicamente nei package relativi alle tabelle effettivamente utilizzate.

### 2.2.2 MongoDB

Il database non relazionale MongoDB è stato impiegato, come previsto dai requisiti, per la gestione dei dati a carattere maggiormente dinamico, come le recensioni e i messaggi delle chat.

L'accesso a questo database avviene tramite un server sviluppato con Express.js, responsabile dell'elaborazione delle richieste provenienti dal server principale. I modelli delle collezioni sono organizzati all'interno di una cartella dedicata denominata *Models*.

### 2.2.3 Caricamento dei dati

Il caricamento dei dati a partire dai file csv forniti è avvenuto tramite un jupyter notebook in seguito a una rapida pulizia di questi ultimi. Il file di caricamento è chiamato "PuliziaCaricamento.ipynb".

### 2.2.4 Comunicazione tra i Server

La comunicazione tra i diversi server avviene esclusivamente tramite l'utilizzo della libreria *Axios*. In particolare, è il *MainServer* ad avere il compito esclusivo di inviare richieste agli altri server, i quali si limitano a rispondere, svolgendo pertanto una funzione passiva all'interno dell'architettura del sistema.

Non è stato definito un protocollo di comunicazione strutturato e uniforme: le modalità di interazione variano in base al contesto e alle specifiche esigenze del caso d'uso.

La sezione dell'applicazione che prevede il maggior numero di interazioni tra i server è la pagina di dettaglio di un film. Questa richiede l'aggregazione di molteplici informazioni distribuite tra diversi server, a differenza della homepage, che si limita a una visualizzazione più sintetica.

In fase di progettazione era stata ipotizzata una soluzione basata sull'acquisizione preventiva di tutti i dati necessari, prima del rendering della pagina, per poi passarli in blocco alla vista. Tuttavia, tale approccio si è rivelato impraticabile, in quanto non è stato possibile individuare un metodo efficace per il passaggio di dati complessi alla vista se non tramite la *template visualization* offerta da Handlebars, la quale presenta limiti significativi nella rappresentazione di strutture dati articolate.

Pertanto, si è optato per una strategia alternativa: parte dei dati viene richiesta dinamicamente dopo il caricamento della pagina, tramite chiamate *Axios* inizialmente indirizzate al *MainServer*, il quale si occupa a sua volta di inoltrare la richiesta al server competente.

Questo approccio comporta un leggero aumento del carico computazionale lato client, in quanto rende lo script della pagina più complesso e attivo. Tuttavia, dal punto di vista operativo, si è rivelato funzionale e flessibile. Inoltre, questa

soluzione presenta un vantaggio anche in termini di gestione della massa di dati: vengono richieste e caricate solo le informazioni effettivamente necessarie, evitando il trasferimento e il trattamento di dati ridondanti o inutilizzati.

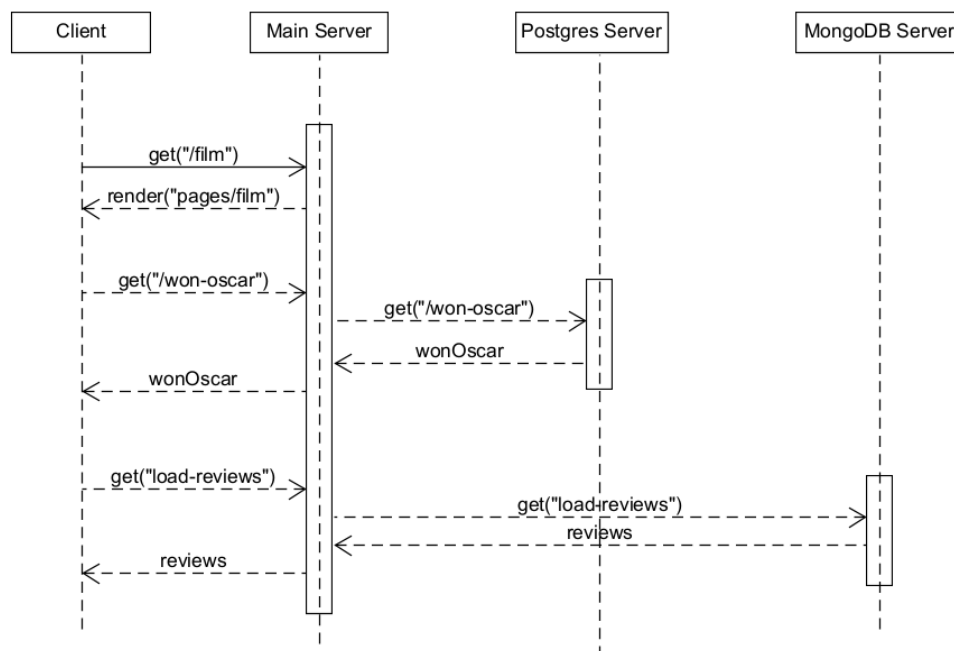


Figura 2.1: Schema UML della comunicazione tra i server per la visualizzazione della pagina in dettaglio dei film

## 2.3 Socket.io

All'interno del progetto è stata integrata la libreria Socket.IO al fine di gestire in tempo reale la funzionalità di chat dedicata a ciascun film. Ogni utente ha la possibilità di inserire il proprio nome e accedere a una stanza di chat identificata univocamente dal titolo del film, creando così un canale di comunicazione separato per ogni opera. Al momento della connessione, il client emette un evento al server per richiedere la creazione o la connessione a una stanza (create or join), e riceve a sua volta notifiche relative alla presenza di altri utenti nella stessa stanza tramite l'evento joined. I messaggi inviati vengono gestiti tramite l'emissione dell'evento chat, che trasporta il contenuto del messaggio, il nome dell'utente e la stanza di appartenenza. Questi dati vengono poi elaborati per visualizzare dinamicamente i messaggi in base al mittente, distinguendo graficamente tra quelli inviati e quelli ricevuti. Inoltre, per garantire la persistenza delle conversazioni, ogni messaggio viene salvato nel database e ricaricato alla riapertura della pagina tramite una chiamata asincrona. Questo approccio ha permesso di realizzare una chat efficiente, fluida e contestuale per ciascun film presente nel sito. ([2])

# Capitolo 3

## Riflessioni sul lavoro svolto

### 3.1 Divisione del Lavoro

L'attività di sviluppo è stata condotta in modo equilibrato e collaborativo tra i due membri del gruppo. Entrambi hanno contribuito attivamente a tutte le fasi del progetto, sia per quanto riguarda l'ideazione e la progettazione, sia per l'implementazione tecnica. In numerose occasioni il codice è stato scritto in presenza, lavorando fianco a fianco, il che ha favorito un confronto continuo e un'evoluzione condivisa delle soluzioni adottate.

Nei momenti in cui non è stato possibile lavorare insieme, le attività sono state suddivise con attenzione, cercando di frammentare il lavoro in unità funzionali il più possibile atomiche. Questo approccio ha permesso una distribuzione efficiente dei compiti: ciascun membro si è occupato, ad esempio, dell'implementazione di una funzione specifica o della visualizzazione di un determinato set di dati. Tale modalità ha garantito coerenza nello stile di programmazione e una buona integrazione tra le diverse componenti sviluppate separatamente.

### 3.2 Considerazioni finali

La realizzazione di questo progetto ha richiesto un impegno significativo, sia in termini di tempo che di dedizione. Fin dalle fasi iniziali ci siamo orientati verso un approccio di sviluppo incrementale, che si è rivelato estremamente efficace per affrontare in modo strutturato le diverse complessità emerse nel corso del lavoro.

Abbiamo iniziato definendo l'aspetto generale del sito e individuando, oltre ai requisiti tecnici forniti, anche le informazioni e le funzionalità che, dal punto di vista dell'utente finale, avrebbero potuto risultare più utili e interessanti. Ci siamo quindi messi nei panni degli utenti, cercando di comprendere quali contenuti avrebbero desiderato trovare e con quale modalità di interazione.

Successivamente, abbiamo sviluppato una prima bozza della homepage, intesa come base visiva e funzionale da cui partire. Da quel punto in poi, l'implementazione è avvenuta progressivamente, aggiungendo un elemento alla volta e testandone l'integrazione con le componenti già esistenti. Questo metodo ci ha permesso di mantenere un elevato livello di controllo sulla stabilità del progetto, facilitando l'identificazione e la risoluzione tempestiva di eventuali problemi.

Nel complesso, l'esperienza è stata estremamente formativa: non solo ci ha permesso di approfondire competenze tecniche, ma ci ha anche aiutato a migliorare la

---

nostra capacità di lavorare in team e affrontare lo sviluppo di un progetto complesso in modo organizzato e sostenibile.



# Bibliografia

- [1] Bootstrap Team. Bootstrap 5. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>, 2021.
- [2] Socket.IO Team. Socket.io v4 - introduction tutorial. <https://socket.io/docs/v4/tutorial/introduction/>, 2024.
- [3] W3Schools. W3schools - tutorials for web development. <https://www.w3schools.com/>, 2025.