

Trabajo Práctico 1: Movies

Grupo 8: Integrantes

- Sofía Keceli
- Juan Luis Martinó

Introducción

El presente informe detalla el desarrollo de la aplicación web "MovieWeb", que permite a los usuarios buscar y explorar información sobre películas, actores y directores. Utilizando Node.js y una base de datos SQLite, la aplicación ofrece las opciones de búsqueda avanzada y visualización detallada de datos relacionados con el cine.

Objetivos

- Modificar el código existente para habilitar la búsqueda de películas, actores y directores en la página principal.
- Modificar la página de resultados de búsqueda para mostrar los resultados separados en secciones (películas, actores, directores).
- Desarrollar la funcionalidad para mostrar la lista de películas en las que una persona (actor o director) ha participado al hacer clic en su nombre. Página de actor y director.
- Mejorar la página de película para mostrar toda la información contenida en la base de datos "movies.db".
- Implementar un buscador de palabras clave que devuelva películas relacionadas. Crear las páginas del buscador y de resultados.
- Crear las tablas y archivos necesarios para permitir la creación, modificación y borrado de usuarios y las películas que vieron. Debe ser posible listar los usuarios, las películas asociadas a los usuarios, la puntuación y opinión de una película.

Esto se puede resumir como:

1. Búsqueda de películas, actores y directores.
2. Páginas de personas.
3. Información detallada de películas.
4. Búsqueda de películas por palabras clave.
5. Gestión de usuarios y reseñas.

Funcionalidades Implementadas

1. Búsqueda de Películas, Actores y Directores:

- Se modificó la página principal (index.ejs) para incluir un buscador que permite buscar no solo películas, sino también actores y directores.
- Se creó una página de resultados de búsqueda que muestra los resultados categorizados en secciones (resultado.ejs).

2. Páginas de Personas:

- Al hacer clic en un actor o director desde los resultados de búsqueda, se redirige a su respectiva página, donde se listan las películas en las que han participado.
- Se desarrolló una página llamada "person" (person.ejs) para cumplir esta funcionalidad. Posee la información de la persona como actor y como director cuando se hace click en ella.

3. Información Detallada de Películas:

- Se mejoró la página de películas (pelicula.ejs) para incluir toda la información de la base de datos. No solamente la que se encuentra en la tabla movies, sino que toda la información de las películas en la base de datos, como géneros, país de producción y otros datos relevantes.

4. Búsqueda de Películas por Palabras Clave:

- Se implementó un buscador de palabras clave, permitiendo a los usuarios encontrar películas relacionadas a palabras específicas. Esto fue implementado con una casilla de activación en la página de búsqueda principal (index.ejs).
- Se creó la página keywords_result.ejs para gestionar esta funcionalidad.

5. Gestión de Usuarios y Reseñas:

- Se desarrollaron las tablas¹ y funcionalidades necesarias para manejar la creación, modificación y eliminación de usuarios. Esto es a través de

¹ Users(user_id, user_username, user_name, user_email), User_login(user_id, password). La contraseña está encriptada dentro de la base de datos de movies.

usuarios individuales. Cada uno crea su propia cuenta, inicia sesión (auth_user.ejs) y así puede modificarla o eliminarla (account.ejs).

- También se desarrollaron funciones para permitir a los usuarios opinar sobre películas². Estas reseñas pueden ser modificadas o eliminadas por el usuario que las creó (account.ejs).
- Se habilitó la búsqueda de usuarios y películas reseñadas (users.ejs). Los resultados se muestran en users_result.ejs, listando usuarios y películas con sus reseñas correspondientes según la query. Al hacer clic en un usuario, se muestran las películas sobre las que opinó, junto con las puntuaciones y reseñas asignadas (user.ejs).

El código también ha sido documentado con comentarios descriptivos para facilitar la comprensión del mismo. Cada funcionalidad incluye explicaciones sobre su funcionamiento.

Desafíos Enfrentados

- **Integración de Funcionalidades:** Durante el desarrollo, se presentaron desafíos en la integración de diversas funcionalidades, especialmente en la correcta asociación de la información de las películas que no se encontraba en movie, sino en otras tablas del esquema.
- **Diseño del código front-end:** Otro desafío fue poder desarrollar y codificar de manera correcta el CSS, ya que a pesar de no ser el centro del trabajo práctico y su codificación no era avanzada, era crucial al momento de implementar el front-end.
- **Manejo de Errores:** La gestión de errores fue fundamental para asegurar la estabilidad de la aplicación. Se implementaron validaciones en los formularios de búsqueda y sus respectivas respuestas.
- **Coordinación en equipo:** Trabajando en un grupo, se volvió crucial coordinar las tareas para evitar conflictos. En particular, la superposición de cambios: utilizando la herramienta de live-sharing de VSCode, los cambios realizados por una persona pueden sobrescribir los de la otra, lo que causa confusiones y errores al ejecutar el código. Lo mismo aplica a trabajar sobre un repositorio en Github.
- **Comunicación constante:** Es necesario mantener una comunicación fluida sobre quién está trabajando en qué parte del proyecto para minimizar conflictos y asegurar

² Se creó la tabla Movie_user(movie_user_id, user_id, movie_id, rating, review), donde user_id y movie_id son foreign keys (con "on delete cascade" para consistencia) y exigiendo que el par fuera único (no más de una review por película por usuario).

la integración de los cambios. Por ejemplo, si un participante borraba archivos o secciones de código sin avisarle al otro, generaba problemas.

- **Conocimiento sobre herramientas:** Trabajar con herramientas nuevas, como la codificación de SQLite o cookies para inicio de sesión de usuarios trajo la necesidad de aprender sobre ellas y los problemas que no conocerlas bien implica. Un ejemplo fue el uso de foreign keys en Node.js, las cuales se tienen que activar para que funcionen correctamente.

Conclusión

El desarrollo de "MovieWeb" fue una experiencia valiosa que nos permitió trabajar en equipo, enfrentando y superando diversos desafíos, y aprender nuevas técnicas y habilidades en el camino. El hecho de implementar también parte de la interfaz nos llevó a experimentar con el CSS: A medida que ajustábamos los estilos, nos dimos cuenta de lo crucial que es una buena presentación para la experiencia del usuario.

Appendix

Creación de las tablas usadas para la funcionalidad de usuarios:

```
CREATE TABLE users (  
  
    user_id INTEGER PRIMARY KEY,  
  
    username TEXT UNIQUE NOT NULL,  
  
    name TEXT NOT NULL,  
  
    email TEXT UNIQUE NOT NULL  
  
);  
  
CREATE TABLE movie_user (  
  
    movie_user_id INTEGER PRIMARY KEY,  
  
    user_id INTEGER,  
  
    movie_id INTEGER,  
  
    rating REAL CHECK (rating between 1.0 and 5.0),  
  
    review TEXT,  
  
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE,  
  
    FOREIGN KEY (movie_id) REFERENCES movie(movie_id) ON DELETE CASCADE,  
  
    unique (user_id, movie_id)  
  
);  
  
CREATE TABLE user_login (  
  
    user_id INTEGER PRIMARY KEY,  
  
    password TEXT,  
  
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE  
  
);
```