

## Proyecto 2: Sistema de recomendación de películas

### Estructuras Usadas:

Diccionario (std::unordered\_map)

- Es la base de datos, de esta estructura se crean las demás
- Clave = ID único de película (Entero)
- Valor = Objeto película con sus atributos.
- Clase Película = Atributos (ID, Título, Director, Género, Puntuación, Año de publicación, Duración en minutos)

Árbol AVL (Árbol de búsqueda binario autoequilibrado, implementación propia)

- Para cada nodo, la diferencia de altura entre sus subárboles izquierdo y derecho es como máximo 1 (factor de equilibrio) y esto se mantiene gracias a rotaciones específicas. Además, los valores en el subárbol izquierdo al nodo son menores que el valor de este, y todos los valores en su subárbol derecho son mayores.
- Nodo = Cada nodo tiene una puntuación o calificación película (0.0 a 10.0) asociada y un arreglo dinámico que guarda las películas con esa puntuación.
- El árbol se construye a partir del diccionario, si la película que se va a agregar tiene una puntuación existente en algún nodo del árbol, esta se agrega al arreglo alojado allí; si no hay nodo con esta puntuación se crea uno nuevo.
- Esta estructura permite filtrar las películas a través de su puntuación o calificación, se le da un intervalo de puntuación y ésta devuelve un conjunto con las películas dentro del intervalo.

Matriz

- Fue la estructura utilizada para manejar la relación entre las películas y los géneros. Dado que ya poseíamos un diccionario con las películas (una por una con los géneros como atributos), ya solo faltaba realizar otro diccionario por géneros para hacer las veces de filas, que tuviera a las películas como columnas, lo típico sería guardar el valor 1 ó 0 para diferenciar si una película pertenece a cierto género o no, pero al ser una matriz que solo contiene las celdas con información, resultaba redundante. Por tanto esta parte de la matriz resultó como un diccionario con conjuntos de películas en su interior, significando que dichas películas hacen parte del género en cuestión. Si bien lo común es trabajar matrices por medio de vectores, dado que no íbamos a realizar operaciones algebraicas tomamos el atrevimiento de manejarlo con diccionarios para agilizar la búsqueda.

Conjuntos (std::set)

- Colección ordenada(en el caso de std::set) de elementos únicos.
- Se usó para almacenar la colección de películas que cumplían con los criterios de intervalo del árbol y también las que cumplían con la búsqueda por género en la matriz . Esto facilita la intersección de estos dos filtros y la búsqueda por más de un género

### **Porcentaje de uso IA:**

Para el desarrollo del proyecto se usó en tres puntos cruciales la asistencia de la IA, en primera instancia para poder manejar el proyecto desde Visual Studio Code en varios archivos, esto con el fin de poder hacer una compilación y ejecución más profesional. La IA también fue de gran ayuda en el debugging, ya que al ser un programa con bastantes archivos y líneas de código en cada uno de estos, surgieron algunos errores difíciles de notar. Por último, para el desarrollo e integración de la interfaz con el BackEnd. En resumen, se puede decir que se usó un 60% de asistencia.