

Ejercicio 8

Sea una empresa de cable *on demand* que entrega decodificadores a sus clientes para que miren las películas que ofrece. El decodificador muestra la grilla de películas y también sugiere películas.

Usted debe implementar la aplicación para que el decodificador sugiera películas. El decodificador conoce la grilla de películas (lista completa que ofrece la empresa), como así también las películas que reproduce. De cada película se conoce título, año de estreno, películas similares y puntaje. La similaridad establece una relación recíproca entre dos películas, por lo que si A es similar a B entonces también B es similar a A.

Cada decodificador puede ser configurado para que sugiera 3 películas (que no haya reproducido) por alguno de los siguientes criterios:

- (i) novedad: las películas más recientes.
- (ii) similaridad: las películas similares a alguna película que reprodujo, ordenadas de más a menos reciente.
- (iii) puntaje: las películas de mayor puntaje, para igual puntaje considera las más recientes.

Tenga en cuenta que la configuración del criterio de sugerencia del decodificador no es fija, sino que el usuario la debe poder cambiar en cualquier momento. El sistema debe soportar agregar nuevos tipos de sugerencias aparte de las tres mencionadas.

Sea un decodificador que reprodujo Thor y Rocky, y posee la siguiente lista de películas:

Thor, 7.9, 2007 (Similar a Capitan America, Iron Man)
Capitan America, 7.8, 2016 (Similar a Thor, Iron Man)
Iron man, 7.9, 2010 (Similar a Thor, Capitan America)
Dunkirk, 7.9, 2017
Rocky, 8.1, 1976 (Similar a Rambo)
Rambo, 7.8, 1979 (Similar a Rocky)

Las películas que debería sugerir son:

- (i) Dunkirk, Capitan America, Iron man
- (ii) Capitán América, Iron man, Rambo
- (iii) Dunkirk, Iron man, Capitan America

Nota: si existen más de 3 películas con el mismo criterio, retorna 3 de ellas sin importar cuales. Por ejemplo, si las 6 películas son del 2018, el criterio (i) retorna 3 cualquiera.

1. Realice el diseño de una correcta solución orientada a objetos con un diagrama UML de clases.
2. Si utiliza patrones de diseño indique cuáles y también indique los participantes de esos patrones en su solución según el libro de Gamma et al.
3. Escriba un test case que incluya estos pasos, con los ejemplos mencionados anteriormente:
 - configure al decodificador para que sugiera por similaridad (ii)
 - solicite al mismo decodificador las sugerencias
 - configure al mismo decodificador para que sugiera por puntaje (iii)
 - solicite al mismo decodificador las sugerencias
4. Programe su solución en Java. Debe implementarse respetando todas las buenas prácticas de diseño y programación de POO.