



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2113 - Diseño Detallado de Software (2020-2)

Ejercicio I1

Indicaciones

- Pueden responder este ejercicio en C# o en Ruby. Recuerden que a lo menos uno de los dos ejercicios prácticos del semestre debe estar en C# (pueden responder una interrogación en Ruby y la otra en C#, o ambas en C#).
- Deben subir el o los archivos comprimidos en un .zip
- Todas las preguntas deben ser vía Issues de Github con el tag [Ejercicio I1].
- **El objetivo de este ejercicio es que tu solución sea modular, use buenas prácticas y sea entendible, además de funcional. El ejercicio consta de un total de 100 puntos, los cuales se reparten de la siguiente forma**
 - **Resuelve el problema correctamente [30 puntos]**
 - **El diseño de la aplicación obedece a los principios SOLID [40 puntos]**
 - **El diseño de la aplicación obedece a los principios fundamentales [10 puntos]**
 - **Uso de buenas prácticas [20 puntos]**

En caso de no resolver el problema (no compila o se cae en ejecución) la nota máxima a la que puede optar es 5.0.

Plazo de entrega: 17 de Septiembre a las 23:59 vía buzón de tareas en Canvas.

Tu mejor amigo se ganó un fondo de inversión por una idea revolucionaria en un fondo especial para proyectos tecnológicos con “machine learning, IA y blockchain”. Debido a que tiene el fondo, te contrató a ti como su primer desarrollador. Aceptaste feliz, porque puedes ayudar a un amigo y te pagarán bien. Además, *ya lo ayudaste antes a aterrizar su idea*, así que te sientes involucrado.

Su emprendimiento se llama Naděje. Es una aplicación que busca ser un repositorio de películas, hacer reviews y mostrar calificaciones. En una primera iteración, tú y tu amigo **buscan probar el tema de las calificaciones en base a *scripts* en el lenguaje que tu decidas (que para efectos de este ejercicio, se reducen a Ruby y C#)**.

Junto a este enunciado se subieron 3 archivos: imdb.json, rotten.json y metacritic.json, los cuales tienen información de algunas películas cada uno, con distintos datos sobre las mismas. Debes crear un programa que al ejecutarse, lea de estos archivos e imprima en pantalla la lista de todas las películas que se encuentran en los 3 archivos, mostrando solo el título y la calificación definitiva, enumeradas partiendo desde el 1 y ordenadas en base a la calificación definitiva. La consola deberá quedar esperando al input de un usuario. Este input debe ser un número desde el 1 hasta la última película. Al ingresar el número, se debe imprimir en pantalla el título de la película, la calificación definitiva, la calificación por archivo (imdb, rotten y metacritic; en caso de no existir poner n/a), la descripción más corta entre los 3 archivos (que no sean nulas), y la fecha de estreno (en caso de haber diferencias entre archivos, considerar la credibilidad en el orden: imdb, metacritics y finalmente rotten). Además deberá mostrar una lista con 3 elementos:

1. Calificar
2. Review
3. Volver

Si ingresa 1, entonces elige Calificar, se deberá pedir poner un número del 1 al 5. Este número deberá afectar directamente la calificación definitiva de la película.

Si ingresa 2, entonces deberá imprimir en pantalla “No implementado”.

Si ingresa 3, vuelve al “menú principal” que es la lista de películas con la calificación actualizada. El script se detiene si el usuario escribe “exit”.

¿Cómo se calcula la calificación definitiva?

Cada uno de los archivos usa una escala distinta para medir: imdb usa una escala del 1 al 10, rotten usa una escala del 1% al 100%, y metacritic usa una escala del 1 al 100. Además, el review del usuario dentro de Naděje va del 1 al 5. Por lo anterior, debe normalizar las 4 calificaciones antes de promediarlas. Se recomienda usar la siguiente fórmula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Donde X' es la calificación normalizada para cada fuente de datos. Luego para cada uno de los cuatro X' hace el promedio simple para tener la calificación definitiva.

Por facilidad, no es necesario persistir la información de las calificaciones (vale decir, pueden quedar en memoria).

Entregables: programa que se ejecute en consola que resuelva (en Ruby o C#).

Ejemplo del funcionamiento¹ al iniciar:

1. El padrino (0.96)
2. Sueños de libertad (0.91)
3. La lista de Schindler (0.89)
4. Pulp Fiction (0.88)
5. The Joker (0.85)
6. Star Wars: El imperio contraataca (0.84)
7. Matrix (0.83)
8. The Silent of The Lambs (0.81)
9. Volver al futuro (0.8)
10. Psycho (0.78)
11. Whiplash (0.77)

> 1

El padrino

Calificación definitiva: 0.96

IMDB: 9,2

Rotten Tomatoes: n/a

Metascore: 100

Mi calificación: n/a

Descripción: The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.

Año: 1972

1. Calificar

2. Review

3. Volver

> 1

¹ Este ejemplo no está usando la información real de los archivos json, por lo que su calificación definitiva no está siendo realmente calculada.

Ingrese un número del 1 al 5

> 4

El padrino

Calificación definitiva: 0.89

IMDB: 9,2

Rotten Tomatoes: n/a

Metascore: 100

Mi calificación: 4

Descripción: The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.

Año: 1972

1. Calificar

2. Review

3. Volver

> 3

1. Sueños de libertad (0.91)

2. La lista de Schindler (0.89)

2. El padrino (0.89)

4. Pulp Fiction (0.88)

5. The Joker (0.85)

6. Star Wars: El imperio contraataca (0.84)

7. Matrix (0.83)

8. The Silent of The Lambs (0.81)

9. Volver al futuro (0.8)

10. Psycho (0.78)

11. Whiplash (0.77)

> exit