

## Técnicas Actuales de Programación 2021-2022

### Examen Final, convocatoria extraordinaria (1/7/2022)

Responda al examen con un proyecto que lleve su nombre (elija Java with Ant -> Java Application).

Una plataforma de streaming tipo Netflix, quiere analizar una manera sencilla de sugerir películas a sus usuarios. Ésta consiste en localizar a un usuario de gustos similares y proponer las películas que más le hayan gustado a este usuario similar que no haya visto ya el usuario al que se hacen las sugerencias.

Para esto disponen de las valoraciones que hacen los usuarios, que se encuentran en un fichero de datos que contiene una valoración por línea, consistente en el nombre de usuario, un tabulador, el título de la película, un tabulador, y la valoración numérica en el rango [1..5]

Ronstic	West Side Story	5
BarishAracne	Bella de día	3
Sparrowbar	Cinema Paradiso	3
Almadeestrella	Easy Rider	2
Ethernalflame	Los siete magníficos	3
DragoEldos	Desayuno con diamantes	1
Duca2	Las zapatillas rojas	5
Musadeazucar	Los 400 golpes	1
MariaVaria	Luna Nueva	3

La similitud entre usuarios se calcula como

$$sim(u, v) = 1 - \frac{1}{4|P_u \cup P_v|} \sum_{\forall p \in P_u \cup P_v} |s(u, p) - s(v, p)|$$

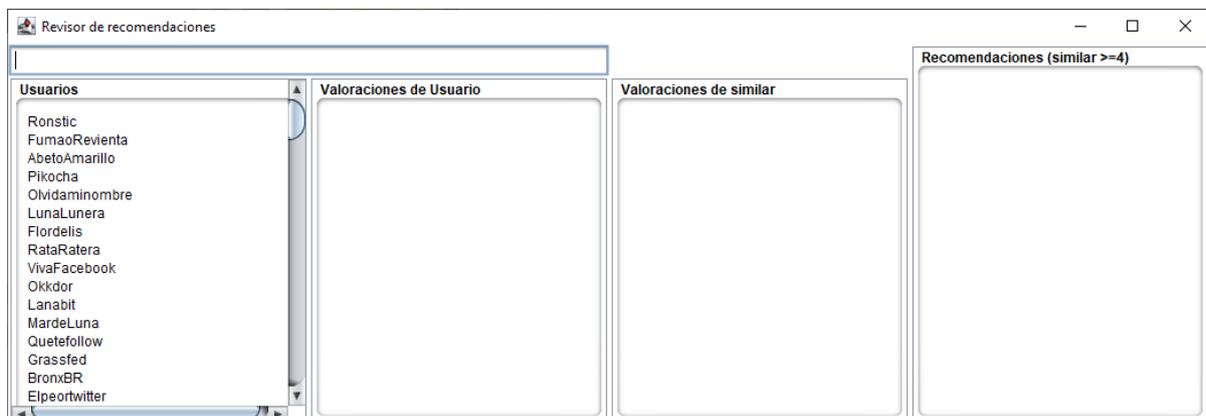
donde  $u, v$  son usuarios,  $p$  es una película,  $P_u$  es el conjunto de películas valoradas  $u$ , y  $s(u, p)$  es la valoración que el usuario  $u$  ha dado a la película  $p$ . Los usuarios valoran las películas con un entero de 1 a 5, y en el caso en que  $u$  no haya valorado  $p$  se considera  $s(u, p) \equiv 3$

Interpretación de la fórmula: La similitud es un valor en el rango [0..1] que se obtiene restando a 1 un valor en [0..1] que representa la disparidad entre dos usuarios. Ésta se calcula como la suma normalizada de todas sus diferencias de valoración extendida al conjunto de todas las películas valoradas entre ambos. El factor de normalización es el inverso al valor máximo de la suma, es decir, del número de películas tenidas en cuenta ( $|x|$  es el cardinal del conjunto  $x$ ) multiplicado por 4 (el valor máximo que puede tener la diferencia de valoraciones).

Para realizar este análisis **se le pide** una sencilla herramienta que permita explorar con un GUI de un modo “amigable”, las valoraciones dadas en casos concretos, mostrando los valores que se encuentran en el fichero y el resultado de aplicar la estrategia indicada. El **fichero de datos** se encuentra en <http://gtts.ehu.es/German/Docencia/2122/TAP/examenes/Tapflix.txt> con codificación UTF8.

**Se le recomienda** crear 1) la clase gráfica (JFrame) con funcionalidad y aspecto razonablemente ajustadas a lo indicado a continuación, y 2) una clase que recoja los datos y aporte funciones que faciliten la presentación en el GUI.

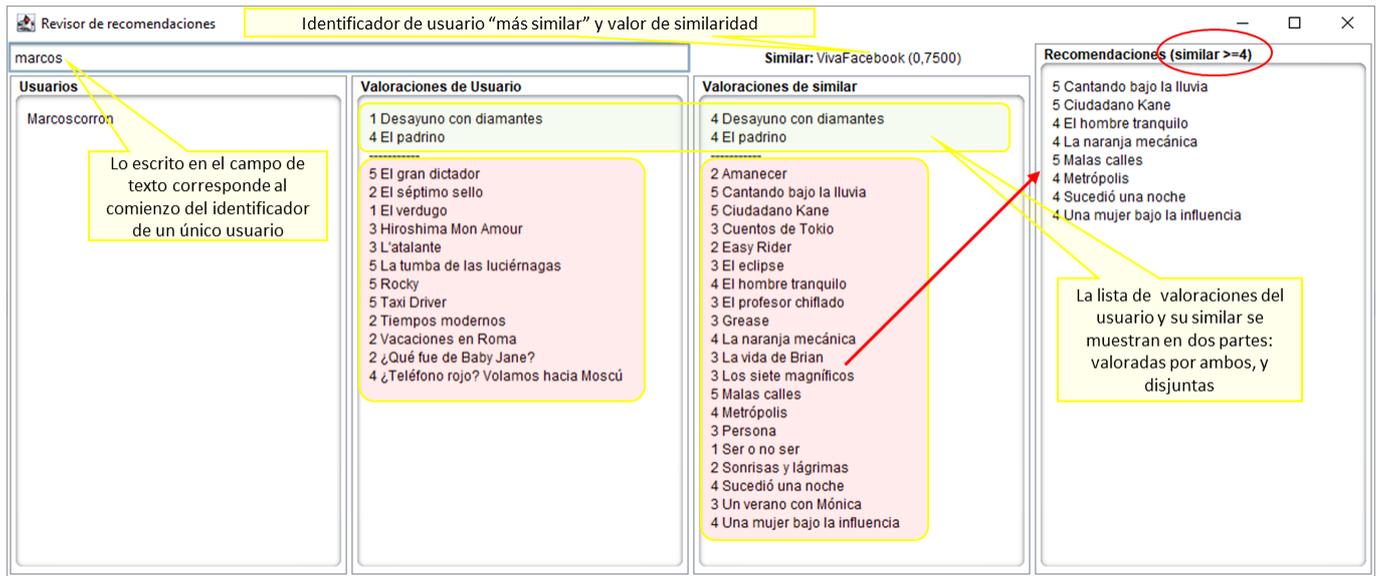
- 1) El **GUI** tendrá aproximadamente el siguiente aspecto en su arranque:



La funcionalidad ha de ser la siguiente: con lo que pongamos en el campo de texto buscaremos seleccionar un único usuario, es decir pondremos el identificador de uno de ellos, o al menos la parte inicial que sea suficiente para que sólo coincida con uno.

Una vez que se haya determinado un usuario concreto, se mostrará en los demás componentes la información que veremos en la imagen

La mejor gestión del campo de texto, con gran diferencia sobre las alternativas, puede hacerse carácter a carácter atendiendo al evento `keyTyped` (mostrando la lista de usuarios compatible con lo tecleado hasta el momento), pero se recomienda comenzar por un método más simple puesto que la gestión de la tecla de borrado no es evidente en principio y es posible que lleve tiempo resolverla. Las alternativas más sencillas (y menos agradables para el uso) son la atención al evento `actionPerformed` (que se lanza al pulsar la tecla enter), o la inclusión de un botón (o ambas).



## 2) La clase que recoge y proporciona los datos puede seguir las siguientes sugerencias:

Una estructura adecuada para los datos puede ser un mapa de nombres de usuario a sus valoraciones, donde éstas serán un mapa de películas a valoraciones. Si este segundo mapa está ordenado, la representación posterior en el GUI de los títulos de las películas se verá “automáticamente” ordenada.

Para llevar a cabo lo que se pretende con el GUI, un conjunto de métodos apropiados puede ser el siguiente;

- Métodos triviales que no necesitan explicación por su prototipo
  - `Set<String> getUsers()`
  - `Set<String> getUserMovies(String user)`
  - `Map<String, Integer> getUserRatings(String user)`
  - `int getRate(String user, String movie)`
- `public double similarity(String user1, String user2)` Calcula la función similaridad presentada más arriba.
- `String findMostSimilarUser(String user)` Aplica la función similaridad al usuario dado como parámetro frente a todos los demás y devuelve el identificador del usuario que obtiene el valor máximo (si hay más de uno, cualquiera de ellos)
- `Set<String> getCommonMovies(String user1, String user2)` Devuelve la lista de títulos de películas que han sido valoradas por los dos usuarios dados en los parámetros (es decir la intersección de ambos conjuntos).

Valoración del ejercicio.

Se valorará el resultado del ejercicio atendiendo a la consecución de diversos objetivos. A grandes rasgos serán los siguientes (con diferentes “pesos”).

- La corrección de las clases: elementos necesarios y suficientes, atributos (seguir sugerencias de Netbeans cuando sean razonables), etc.
- La lectura correcta de datos del fichero
- Atención a errores y excepciones
- El aspecto razonable del GUI (que contenga los elementos imprescindibles)
- El redimensionamiento razonable del GUI
- El funcionamiento global de la aplicación.
- La inclusión de comentarios. Particularmente los de documentación para clases y métodos (no es necesario que sean muy amplios, se trata de saber cómo se documenta).
- Legibilidad del código: La ausencia de “basura” (código de pruebas anulado convirtiéndolo en comentarios, importaciones innecesarias), el formateado razonablemente correcto (indentación, no exceso de líneas vacías, etc.)