

Técnicas Actuales de Programación 2022/2023

Convocatoria extraordinaria adelantada 7/3/2023

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

Además de que el código sea razonablemente correcto y dé los resultados esperados, se valorará que esté limpio de restos de pruebas y de código anulado con comentarios, que esté bien formateado (Nebeans ayuda), y que incluya algún comentario donde se considere necesario. En el GUI se valorará el redimensionamiento razonable.

El servicio web IMDB (Internet Movie Data Base) proporciona una API para desarrolladores que permite hacer aplicaciones que exploten sus datos (de acuerdo a diferentes niveles de acuerdos). Para hacer pruebas más limitadas también pone libremente a disposición de la comunidad una serie de ficheros que contienen información parcial de las tablas de su base de datos. Podemos cargar estas tablas en nuestro propio motor de bases de datos y hacer algunas pruebas. Aunque esto no es complicado, no está a nuestro alcance por varias limitaciones, pero sí que vamos a usar parcialmente esa información para hacer un pequeño ejercicio.

Usaremos sólo dos ficheros, accediendo a ellos de modo remoto en las direcciones

<http://gtts.ehu.es/German/Docencia/2223/TAP/examenes/title.basics.tsv>

<http://gtts.ehu.es/German/Docencia/2223/TAP/examenes/name.basics.tsv>

Estos ficheros presentan respectivamente datos de títulos de productos audiovisuales y personas que participan en su producción. La primera fila es una cabecera que indica qué información aporta cada columna en el resto del fichero. Las columnas están separadas por un tabulador (en alguna ocasión una columna puede estar vacía por lo que hay dos separadores contiguos, aunque normalmente, si no se dispone de la información se indica con “\N”

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
tt0000001	short	Carmencita	Carmencita	0	1894	\N	1	Documentary, Short
tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5	Animation, Short
tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4	Animation, Comedy, Romance
...								

nconst	primaryName	birthYear	deathYear	primaryProfession	knownForTitles
nm0000001	Fred Astaire	1899	1987	soundtrack, actor, miscellaneous	tt0072308, tt0053137, tt0050419, tt0045537
nm0000002	Lauren Bacall	1924	2014	actress, soundtrack	tt0038355, tt0071877, tt0117057, tt0037382
nm0000003	Brigitte Bardot	1934	\N	actress, soundtrack, music_department	tt0056404, tt0054452, tt0049189, tt0057345
...					

Lo que haremos será recoger parte de esta información en dos clases (cada una relacionada con uno de los ficheros) y fabricar un GUI que nos permita acceder limitadamente a los datos. Ambas clases serán muy parecidas, por lo que será adecuado desarrollar una y generar la segunda como una copia de la primera para posteriormente adaptarla a la adquisición del nuevo fichero.

Ambos ficheros son bastante grandes (9.656.026 líneas el primero y 12.339.153 el segundo) por lo que la aplicación admitirá como parámetro un número máximo de líneas a leer (150.000 es un valor adecuado para las capacidades de los equipos del laboratorio)

Clase IMDBException

Dispondremos una excepción propia para ser usada en algunos puntos del resto del código a desarrollar.

Clase Main

Mediante esta clase de control nos limitaremos a poner en marcha la aplicación leyendo cada uno de los ficheros y arrancado posteriormente el GUI. En fase de desarrollo puede utilizarse para comprobar que

cada paso funciona correctamente. Como ya se ha indicado, admitirá un parámetro entero, que transmitirá a las lecturas de ficheros como valor máximo de líneas a leer.

Clase Titulo

La primera tabla se recogerá mediante la clase Titulo. Para ello dispondrá de una lista estática que almacenará objetos de clase Titulo, y cada objeto de esta clase dispondrá de los campos correspondientes a las columnas del fichero, a excepción del "titleType" ya que **sólo se admitirán las líneas con valor "movie" en este campo**. Los campos serán accesibles desde el exterior sólo en lectura mediante getters.

Será necesario implementar los siguientes métodos:

[Constructor] Titulo(String linea)

Hará el split de la línea y si titleType es "movie" asignará los valores correspondientes a los campos. En caso contrario generará una excepción de clase IMDbException

void readAll(BufferedReader br, int max)

Recibirá un Reader para el fichero correspondiente y lo leerá línea a línea (hasta un máximo dado por "max") instanciando los objetos Titulo y almacenándolos en la lista estática (si no es un "movie" y se produce la excepción, es ignorada)

List<Titulo> select(String buscando)

Devolverá una lista de objetos título que contengan la String buscada en el campo primaryTitle con las siguientes condiciones:

- Será insensible a la capitalización
- Si la String buscada comienza con un "=", se requerirá que ambas cadenas sean idénticas

Titulo getByid(String id)

Devolverá el objeto Titulo que contenga la String buscada en el campo tconst

Clase Nombre

La segunda tabla se recogerá mediante la clase Nombre. Para ello dispondrá de una lista estática que almacenará objetos de clase Nombre, y cada objeto de esta clase dispondrá de los campos correspondientes a las columnas del fichero. **Sólo se tendrán en consideración aquellas líneas que contengan en el campo "primaryProfession" el texto "director", o "actor", o "actress". En todo caso no se tendrán en cuenta líneas que presenten el valor "\N" en los tres campos birthYear, deathYear, y knownForTitles**. Los campos serán accesibles desde el exterior sólo en lectura mediante getters.

Será necesario implementar los siguientes métodos:

[Constructor] Nombre(String linea)

Hará el split de la línea y si se dan las condiciones indicadas asignará los valores correspondientes a los campos. En caso contrario generará una excepción de clase IMDbException

void readAll(BufferedReader br, int max)

Recibirá un Reader para el fichero correspondiente y lo leerá línea a línea (hasta un máximo dado por "max") instanciando los objetos Nombre y almacenándolos en la lista estática (si no se cumplen las condiciones y se produce la excepción, es ignorada)

List<Nombre> select(String buscando)

Devolverá una lista de objetos Nombre que contengan la String buscada en el campo primaryName con las siguientes condiciones:

- Será insensible a la capitalización
- Si la String buscada comienza con un "=", se requerirá que ambas cadenas sean idénticas

(Observese que, a diferencia de la clase anterior, en este caso no escribiremos un método "getById")

Clase IMDBGUI

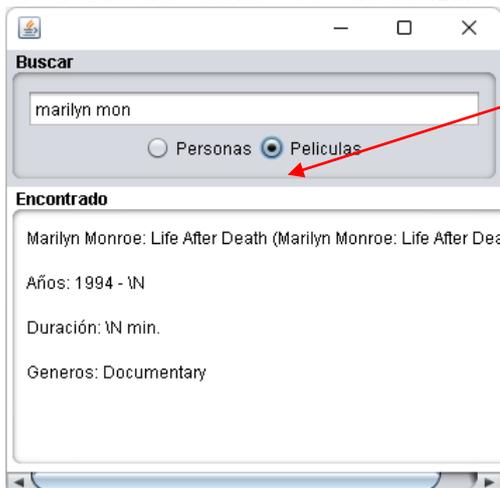
El interfaz gráfico tendrá un aspecto similar a la imagen adjunta.

En la parte superior tendremos un campo para introducir texto a buscar y dos "RadioButtons" para indicar si buscaremos en los datos de nombres (de personas) o de títulos (de películas). En la parte inferior, un área de texto proporcionará la información pertinente en cada caso (inicialmente puede mostrar el número de personas y películas recogidas).

Al introducir un texto y **pulsar la tecla Intro**, se realizará la búsqueda correspondiente. En caso de que haya más de un título o nombre seleccionados, se mostrará la lista. En caso de que haya sólo uno, se mostrará la información correspondiente.

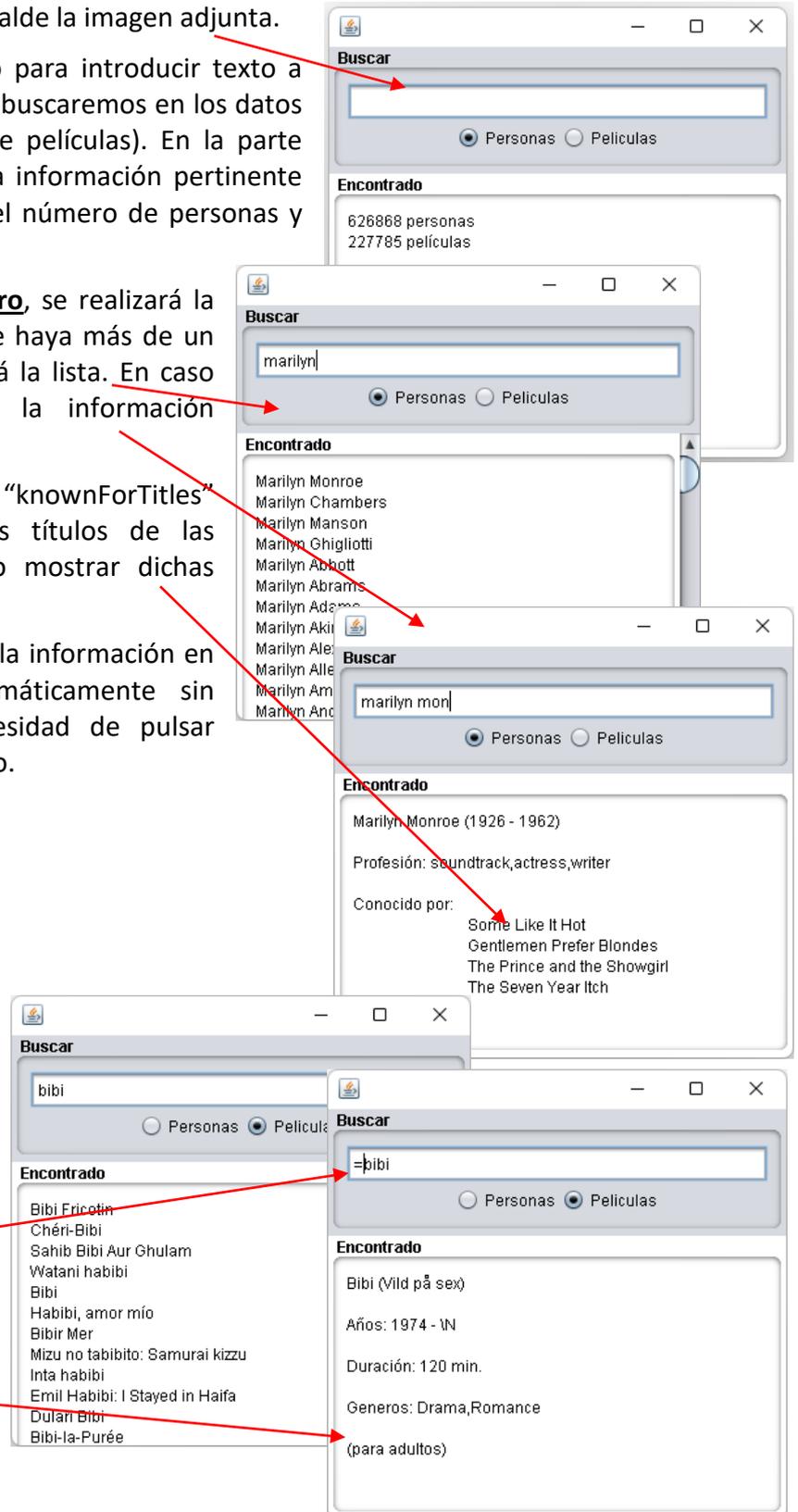
En el caso de buscar personas, el campo "knownForTitles" proporciona índices de referencia a los títulos de las películas, que deberán buscarse para no mostrar dichas referencias sino los títulos.

Al cambiar la selección Personas/Películas, la información en el área de texto se actualizará automáticamente sin necesidad de pulsar Intro.



Comenzando el texto buscado con el carácter "=" se forzará la búsqueda del término exacto.

Al mostrar información de películas, si el campo "isAdult" es "0" no se mostrará nada y si es "1" aparecerá el texto (para adultos)

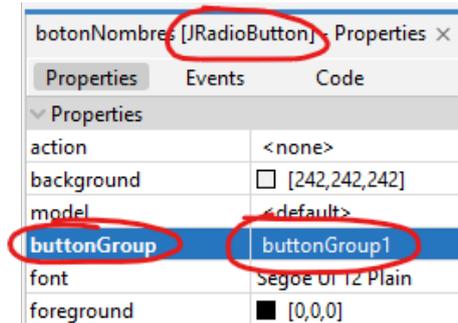


Sobre los JRadioButton:

Para que un conjunto de JRadioButtons se comporten como un grupo (que al seleccionar uno se deselectione automáticamente el anterior) debe generarse también un ButtonGroup (dejándolo en OtherComponents, es decir fuera de la zona visible del JFrame)



Cada JRadioButton deberá indicar en sus propiedades que pertenece a dicho ButtonGroup



Con esto el comportamiento como grupo es automático, y podemos limitarnos a chequear el estado de cada JRadioButton sabiendo que sólo uno estará seleccionado.

Sobre el argumento a pasar al main:

Para ejecutar dentro de Netbeans con paso de argumentos al main, deben ponerse estos en el campo correspondiente de las propiedades del proyecto

