

Proyecto:

AutoBiblio

Grupo 1

Componentes:

Diego José Abengózar (z17m063)

Alejandro García Castellanos (z17m008)

Jorge Gallego (z17m069)

Jaime Vallejo (z17m002)

Índice

1. Objetivo: pág.2
 - 1.1. Resultados esperados
2. Primer planteamiento: pág.2
3. Planteamiento final: pág. 5
 - 3.1. Funciones Finales: pág.6
 - 3.2. Flujogramas Finales
4. Valoraciones: pág.14
 - 4.1. Valoración Grupal
 - 4.2. Valoración Jaime
 - 4.3. Valoración Diego
 - 4.4. Valoración Jorge
 - 4.5. Valoración Alejandro
5. Webgrafía: pág.16

1. Objetivo:

El objetivo en este proyecto es la creación de un programa con los conocimientos adquiridos en la asignatura de Programación I. Con este trabajo buscamos crear un programa simple y eficiente que facilite la utilización de una librería; tanto por parte de un cliente cualquiera, como al propio dueño de la tienda.

1.1. Resultados esperados:

Es por esto por lo que el programa debería de ser capaz de poder agregar y eliminar nuevos ejemplares. Al añadir un libro se añadiría el nombre y se tendría la posibilidad de añadir el género literario, la extensión, el autor y la página por la cual se va; de manera que se vincula a ese libro dichas características. Este paso es fundamental porque no queremos poner los libros únicamente por orden de entrada en nuestra librería, sino que también nos interesaría que muestre cómo sería la biblioteca ordenada según diferentes criterios.

2. Primer planteamiento:

La primera idea se fundamentó en una serie de funciones con objetivos básicos como almacenar los libros, cambiarle las características y, en resumidas cuentas, realizar un inventario de objetos. A continuación, se muestran dichas funcionalidades con sus cabeceras correspondientes:

1. Posibilidad de agregar y eliminar nuevos ejemplares.

public static void otroLibro (int a, String nombre, String[][] libreria)

{Esta función va a recibir el número (int) de la posición en el array (i), el nuevo valor de la posición (String) que será el título del libro, y el nombre del array que hemos definido (libreria)}

public static void eliminarLibro (int a, String nombre, String[][] libreria)

{Es igual que la anterior, pero asignará el valor nulo donde antes estaba el libro (null)}

2. Vincular a un libro unas características: Género, autor, extensión, valoración...

public static void otroAutor (int a, String nombre, String[][] libreria)

public static void otroGenero (int a, String nombre, String[][] libreria)

public static void otroExtension (int a, String nombre, String[][] libreria)

{Igual que la primera función, pero con el valor de la segunda coordenada (j) definida dentro de la función, dependiendo de si quieres añadir un género, autor, etc.}

3. Saber cuántos libros tienes en total en la biblioteca.

public static int contarLibros (String[][] libreria)

{Esta función contará mediante un bucle de incremento cuántas posiciones de la primera coordenada (los libros) es distinta de null}

4. Poder añadir una valoración a través de puntuación.

public static void otraValoracion (int a, String nombre, String[][] libreria)

{Igual que las otras características}

5. Añadir un marcapáginas virtual, es decir, que te muestre si te has leído un libro y en caso de que no lo hayas terminado decir por donde vas.

public static int marcador (int a, int m, String[][] libreria)

{La función te dice por qué punto vas en el libro (compara si el marcador es cero, en cuyo caso te dirá que no lo has empezado; si es igual a la extensión, te dirá que los has terminado; y si no te dirá por cuál vas) y posteriormente te dará la opción de cambiarlo}

6. Para ayudar a un lector a buscar libros según sus preferencias se podría hacer una función que te muestre los libros que sólo tienen unas características determinadas. Ya sea: “quiero que el libro sea de fantasía y que tenga menos de 500 páginas”.

public static void filtro (int a, String nombre, String[][] libreria)

{Coge la característica a filtrar (género, autor...), el valor de esa característica (fantasía, policiaca o J.K. Rowling, George. R.R. Martin), y compruebe con operadores lógicos si el valor coincide con el guardado en cada libro; y en caso positivo imprime todos los que cumplan la condición por pantalla}

7. Mostrar por pantalla la biblioteca ordenada según diferentes criterios:

- *Estándar*: almacenar por autor y dentro del autor por orden lexicográfico, es decir, primero diferencias grupos de libros por su autor y después dentro de cada grupo se tendría que ordenar alfabéticamente.
- *Valoraciones*: por orden de mayor a menor media de valoración
- *Género literario*

➤ Como sería muy tedioso tener que introducir todos los datos de los libros cada vez que se enciende el programa sería de gran utilidad poder guardarlos en un lugar.

Pero tras una reunión con el profesorado adaptamos las funcionalidades a los conocimientos generales de la clase para la entrega, de esta manera las funcionalidades con sus cabeceras correspondientes que tuvimos en cuenta a la hora de realizar nuestro programa en la primera etapa (sin la utilización de arrays) fueron:

public static int menu()

*{Pregunta al usuario que función quiere utilizar, y a partir de ahí ejecuta una de las funciones principales. Al final de cada una de las otras funciones se vuelve a llamar a esta función *menú*, de esta manera se puede seguir usando el programa hasta que no se quiera cambiar o consultar nada más.}*

public static boolean horaValida(int hora, int minutos)

{Comprueba que la hora está entre 0-24 y los minutos entre 0-60.}

public static boolean horarioApertura(int hora, int minutos)

{Compara hora/minutos con el horario (10-14:30 y 17:30-21 en este ejemplo).}

public static void horario()

*{Función principal horario (llama a *horaValida* hasta que se introduzca una hora válida y a *horarioApertura* para ver si está en el rango de horas correspondiente).}*

public static void contadorGeneros()

{Lee título y género de una cantidad de libros determinada por el usuario y cuenta cuántos hay de cada género.}

public static void descuento()

{Esta función permite al usuario comprobar sus descuentos disponibles según la edad, situación laboral, etc.}

*Las funciones *menu*, *horario*, *contadorGeneros* y *descuento* no usan parámetros ya que en la invocación de éstas no iba a ser necesaria la utilización de parámetros actuales, porque iban a hacer uso de variables que se crearían dentro de la función. *

3. Planteamiento final:

Tras aprender a utilizar arrays y aumentar nuestro manejo sobre los bucles, pudimos hacer una primera implementación de dichos conocimientos a nuestro programa con nuevas funciones que lo optimizaron, aumentando su eficacia y versatilidad. Las nuevas funciones con sus respectivas cabeceras fueron:

```
public static void recomienda (String[][]libreria, Scanner sc)  
{El usuario indica un género que le guste y la función le devuelve el ejemplar de ese género con mejor valoración}
```

```
public static void valorarLibro(String[][]libreria, Scanner sc)  
{Esta función permite al usuario poder añadir una valoración a través de puntuación.}
```

```
public static void anadirLibro(String[][]libreria,Scanner sc)  
{Añade un nuevo ejemplar a la librería con sus atributos (género, año, autor...)}
```

```
public static void verLibreria(String[][]libreria, Scanner sc)  
{Muestra la librería (en forma de tabla)}
```

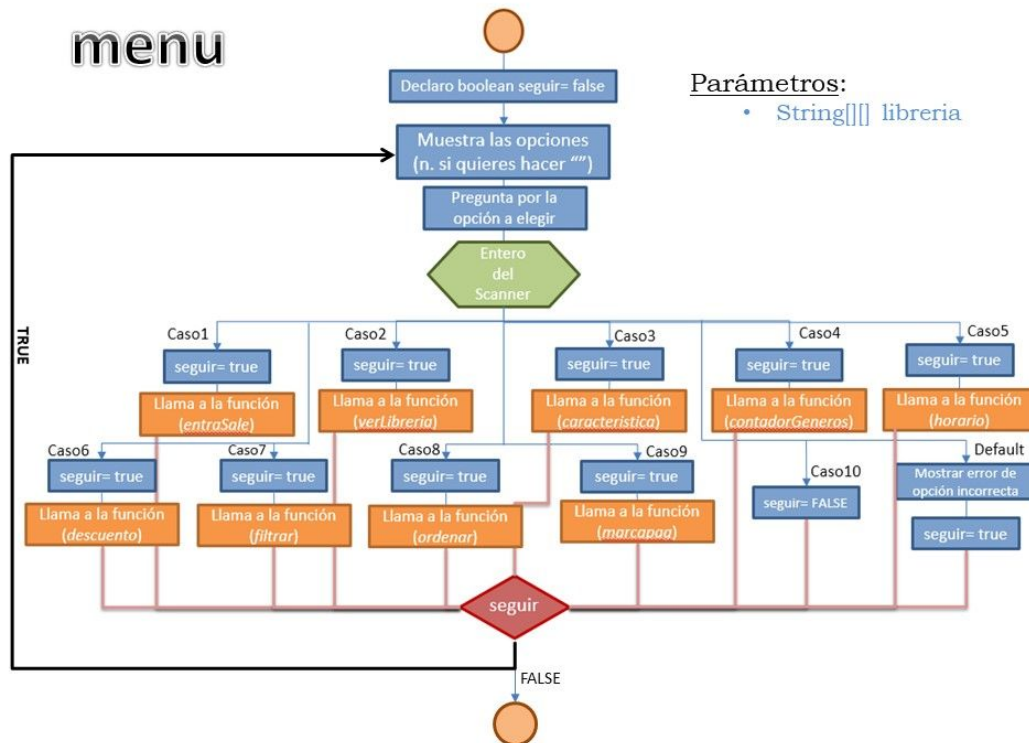
```
public static int str2int(String cadena)  
{Convierte un string que sea un número entero positivo en un int}
```

Gracias a la labor del crítico vimos como podíamos aumentar la utilidad de nuestro programa si implementábamos alguna de las funciones que habíamos pensado en primera instancia, tales como *marcador* o *filtro*; es por esto que para esta entrega final hemos añadido nuevas funciones, con las cuales nuestro programa ha conseguido cumplir el objetivo principal.

3.1. Funciones Finales:

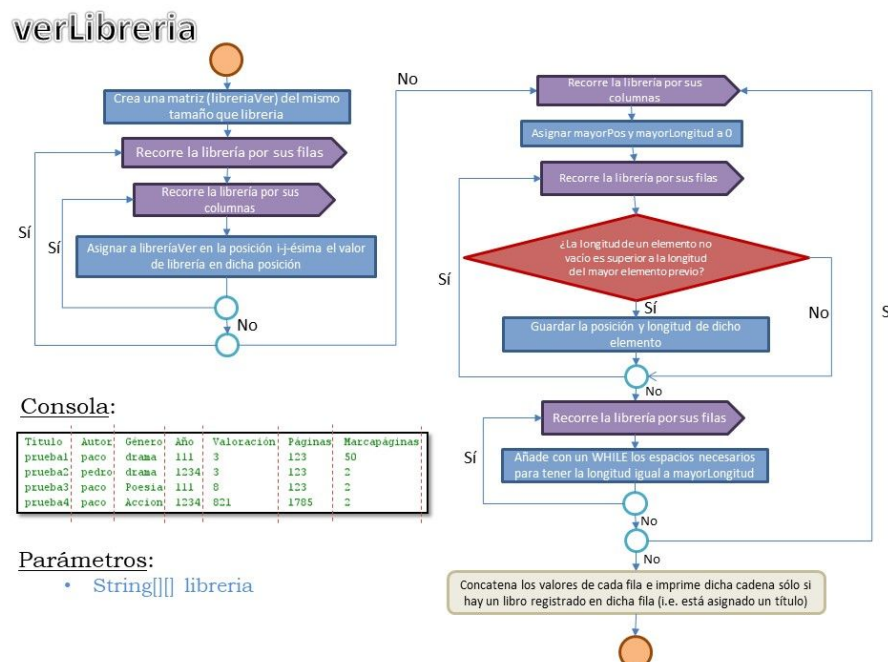
public static void menu(String[][] libreria)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}



public static void verLibreria(String[][] libreria)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

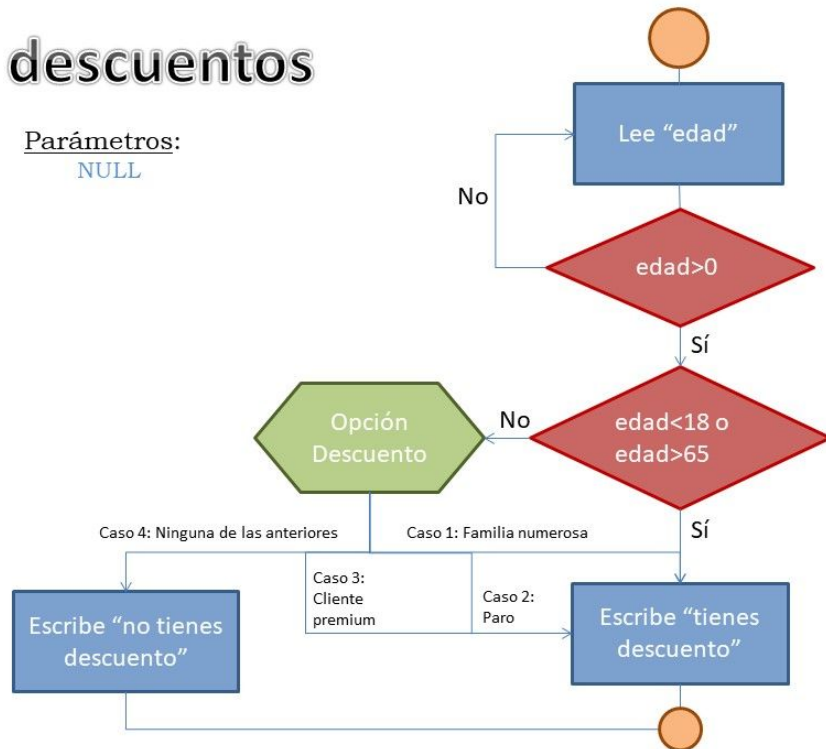


public static void descuento(String[][]libreria)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

descuentos

Parámetros:
NULL



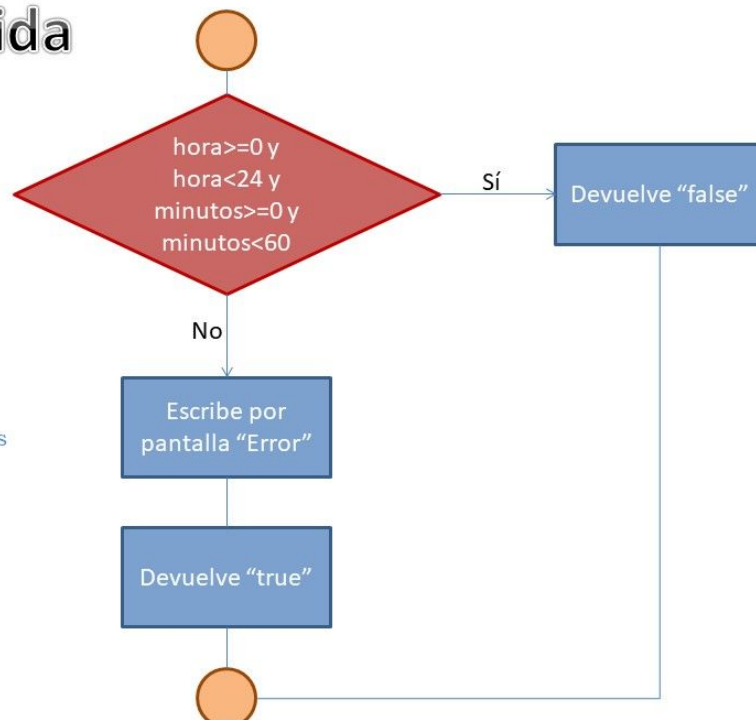
public static boolean horaValida(int hora, int minutos)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

horaValida

Parámetros:

- int hora
- int minutos



public static boolean horarioApertura(int hora, int minutos)

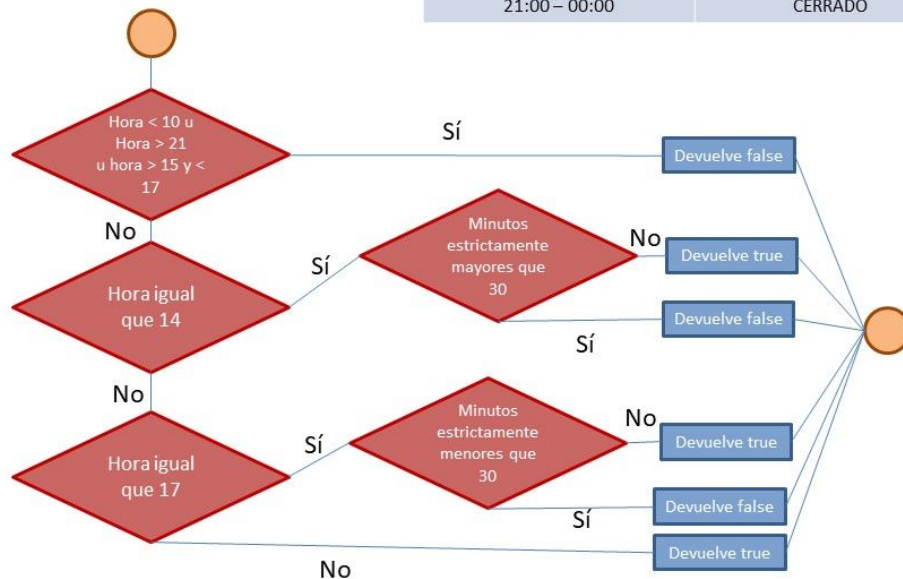
{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

horarioApertura

Parámetros:

- int hora
- int minutos

HORA	ESTADO
00:00 – 10:00	CERRADO
10:00 – 14:30	ABIERTO
14:30 – 17:30	CERRADO
17:30 – 21:00	ABIERTO
21:00 – 00:00	CERRADO



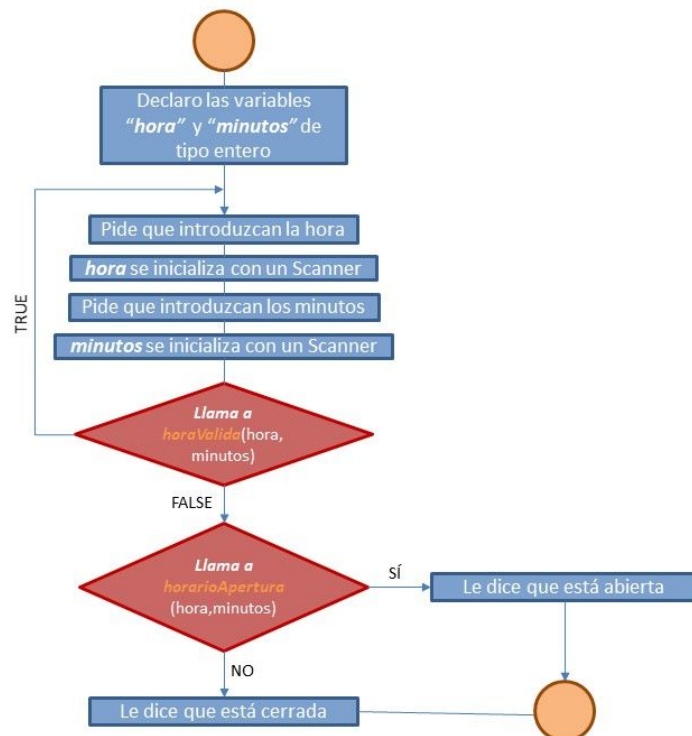
public static void horario(String[][]libreria)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

horario

Parámetros:

NULL



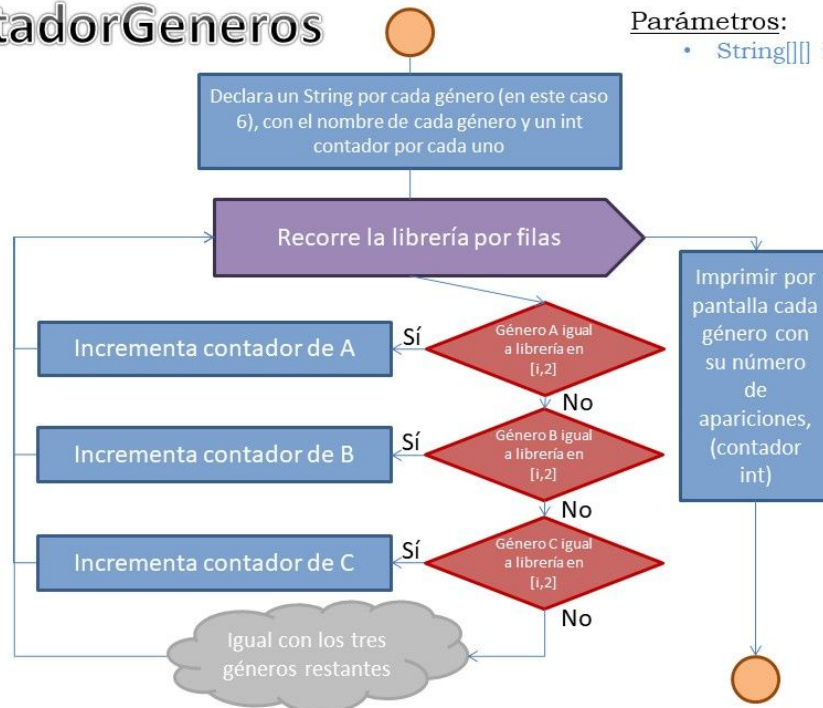
public static void contadorGeneros(String[][] libreria)

{La funcionalidad de esta función se ha mantenido igual que en anteriores entregas}

contadorGeneros

Parámetros:

- String[][] libreria



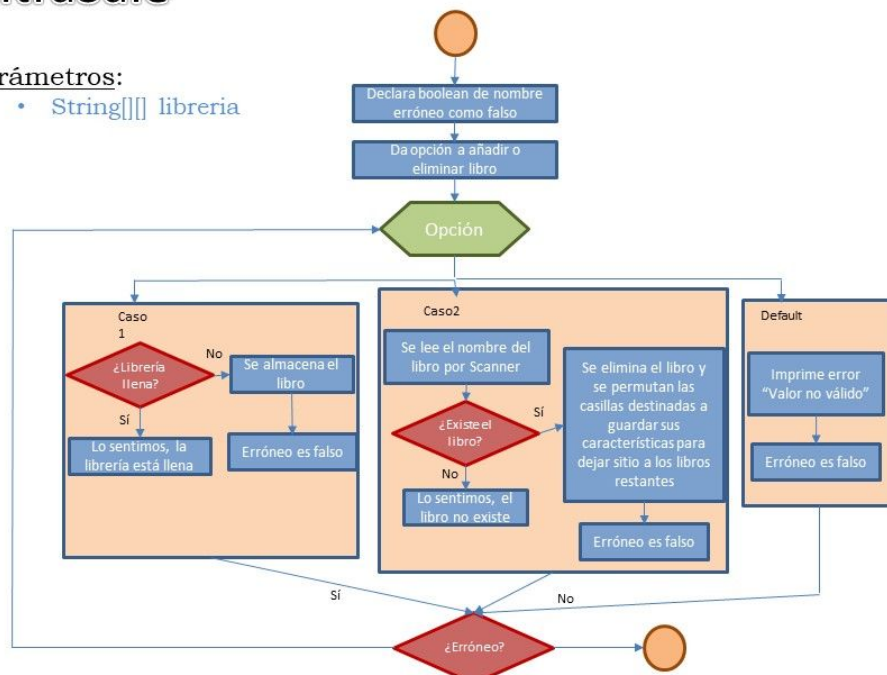
public static String[][] entraSale(String[][] libreria)

{Esta función es *anadirLibro* + *eliminarLibro* que se ha sintetizado con este nombre para que sea más simbólico}

entraSale

Parámetros:

- String[][] libreria



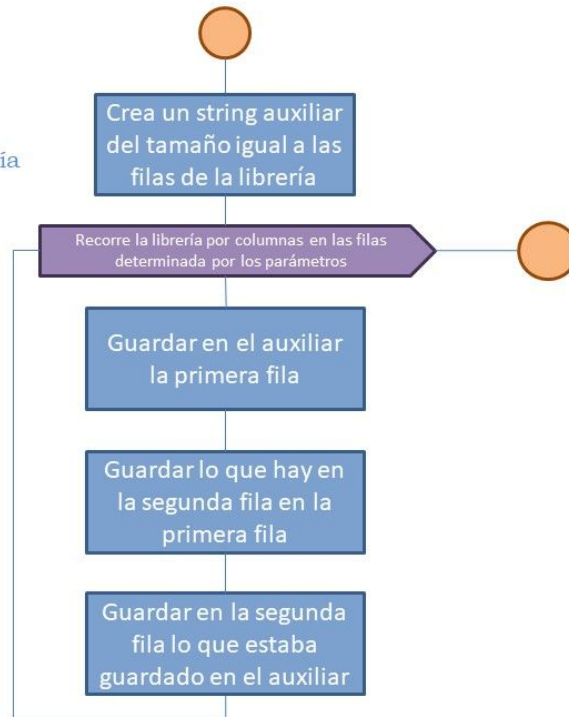
public static String[][] permutar(String[][] libreria, int i, int j)

{Esta función cambia todos los elementos de una fila por los de otra en la que coloca los elementos de la primera, es decir, permuta dos filas de la librería }

permutar

Parámetros:

- String[][] librería
- int i
- int j



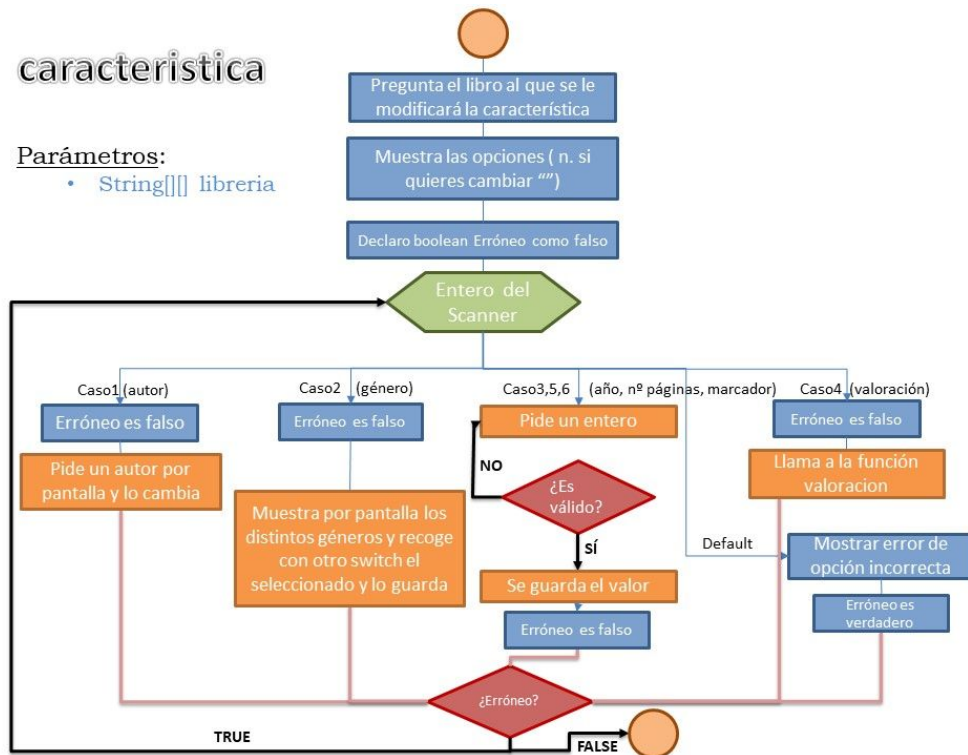
public static String[][] caracteristica(String[][] libreria)

{Esta función es una fusión de las anteriores funciones otroLibro, otroAutor, otraExtension, otroGenero...}

característica

Parámetros:

- `String[][] libreria`



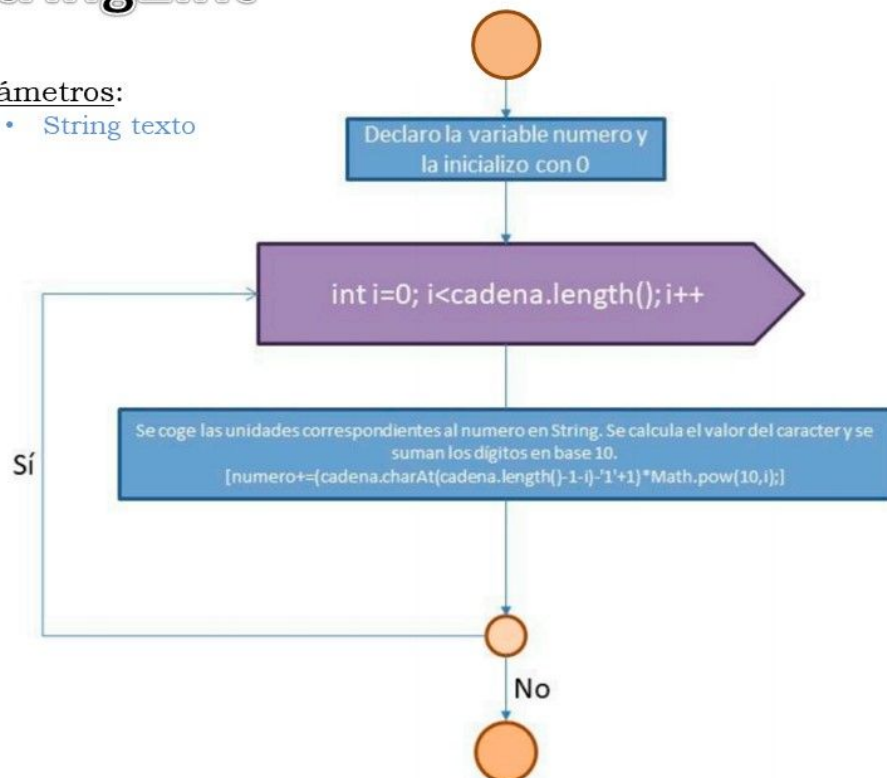
public static int string2int(String texto)

{Esta función es *str2int* con otro identificador}

string2int

Parámetros:

- `String texto`



```
public static int char2int(char character)
```

{Esta función es una variante de `string2int`}

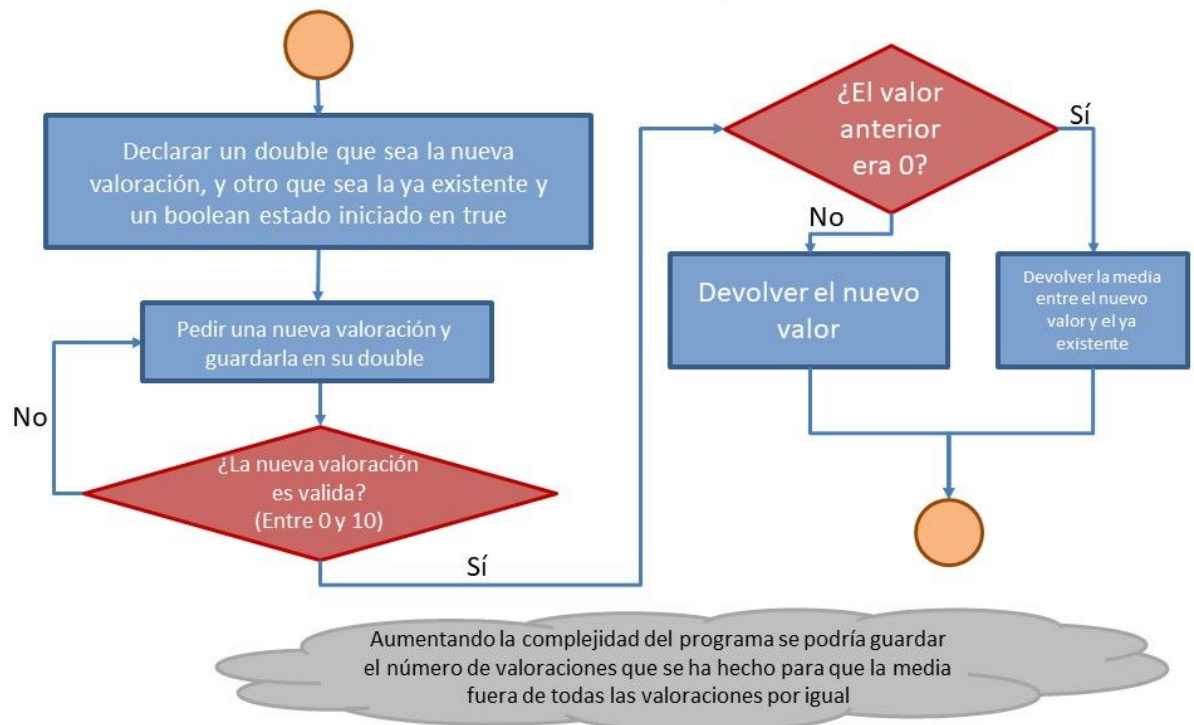
```
public static String valoracion(String valAnt)
```

{Esta función recoge una valoración y hace una media con lo anterior y la trunca a entero>}

valoracion

Parámetros:

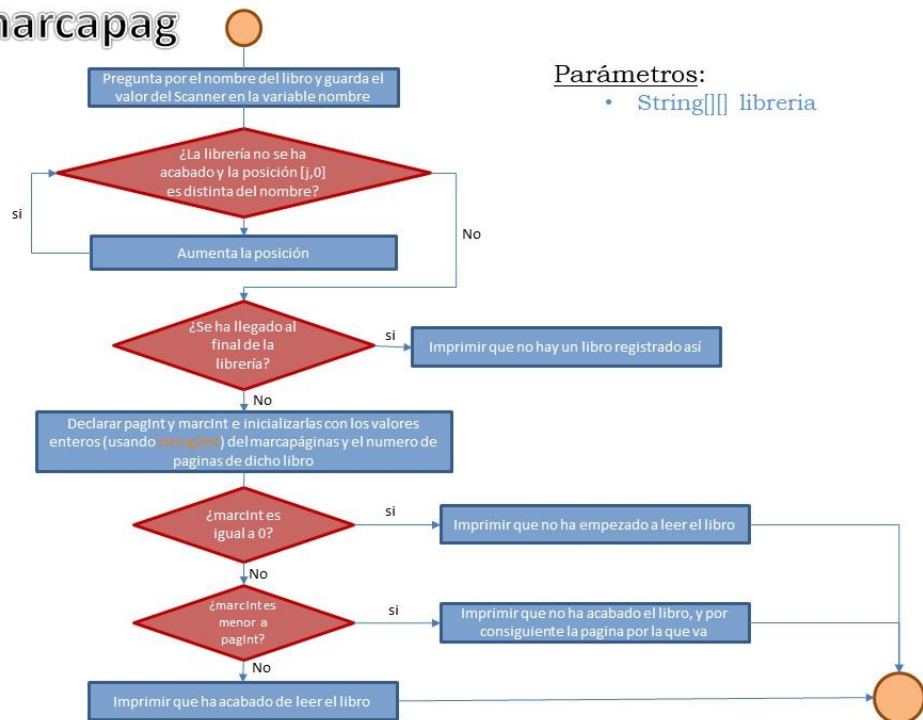
- String valAnt



```
public static void marcapag(String[][] libreria)
```

{Muestra si te has leído un libro y en caso de que no lo hayas terminado decir por donde vas}

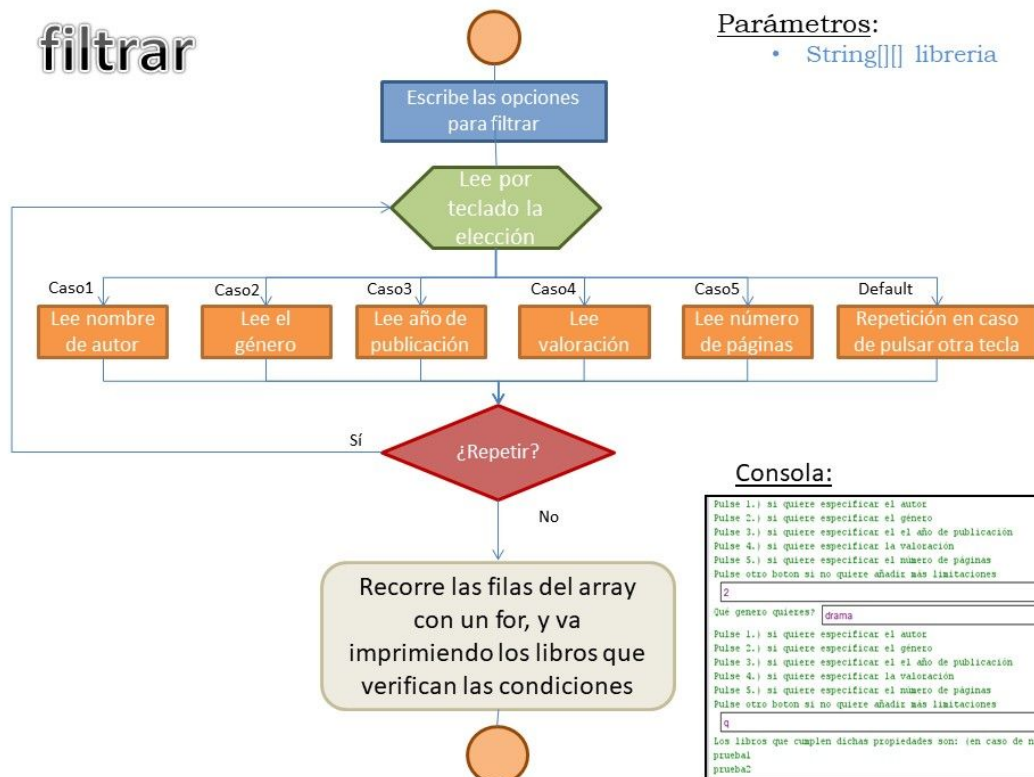
marcapag



public static void filtrar(String[][] libreria)

{Esta función selecciona los ejemplares con unas características que determine el usuario (imprime un conjunto por caracterización)}

filtrar



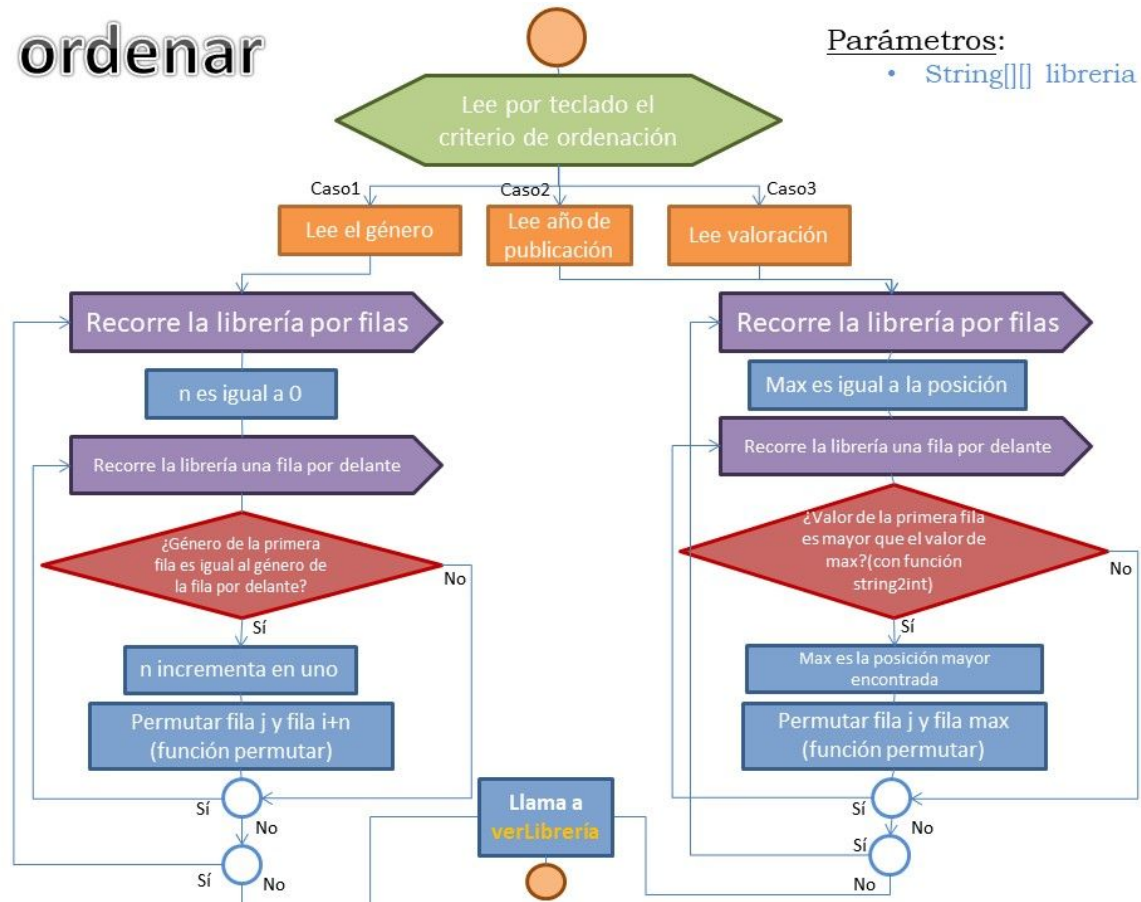
public static void ordenar(String[][] libreria)

{Esta función ordena los ejemplares respecto de las características que determine el usuario}

ordenar

Parámetros:

• String[][] libreria



4. Valoraciones:

4.1. Valoración Grupal:

En general nos ha parecido una experiencia que nos ha ayudado a asentar con más facilidad los conocimientos aprendidos en la asignatura y a trabajar de forma colaborativa. También nos gustaría mencionar brevemente la dinámica que hemos ido tomando a lo largo de este proyecto; así pues, hemos ido asignándonos diferentes roles periódicamente, lo cual nos ha facilitado el hecho de poder ser más autocríticos con nuestro proyecto, ya que al haber trabajado todos tanto de programadores, como diseñadores e incluso críticos, nos ha permitido conocer con gran detalle el progreso de nuestro proyecto desde diferentes puntos de vista. Cabe destacar que el uso dinámico y simultáneo de los archivos del proyecto mediante la plataforma GitHub ha

favorecido la participación de todos creando nuevo código, aportando ideas y corrigiendo errores del resto del equipo. De esta forma, gracias al uso de dichas tecnologías y dinámicas de trabajo hemos podido llegar al objetivo que nos habíamos propuesto a principio de esta asignatura.

4.2. Valoración Jaime:

Desde mi punto de vista, este proyecto ha sido enriquecedor no sólo por la oportunidad de llevar a cabo un reto más difícil que cualquier ejercicio usual de una o dos funciones, sino por el esfuerzo de imaginar nuevos retos y poder llevarlos a cabo con un equipo con el que me he sentido coordinado y de apoyo de todos sobre todos. Creo que podemos estar orgullosos del trabajo realizado y que ha sido una muy grata experiencia.

4.3. Valoración Diego:

En mi opinión ha sido un trabajo muy beneficioso para nuestro aprendizaje ya que se asimilan mucho mejor los conceptos de programación de forma práctica. Además, este tipo de proyectos nos ayudan a acercarnos un poquito a la realidad laboral. Ha supuesto un gran esfuerzo, pero el trabajo en equipo siempre añade un extra de motivación por lo que estamos muy satisfechos con los resultados obtenidos y los conocimientos adquiridos.

4.4. Valoración Jorge:

Gracias a este trabajo he podido ver una aplicación práctica y directa de los conocimientos que adquirimos en clase, además, creo que todos hemos aprovechado el hecho de trabajar en grupo tanto para reforzar nuestros conocimientos al discutir entre nosotros la mejor manera de resolver los problemas, como para, de vez en cuando, aprender cosas nuevas los unos de los otros. Respecto al objetivo del proyecto, creo que se han cumplido las expectativas con holgura, pero sobre todo personalmente me ha ayudado a descubrir la importancia del trabajo en grupo y a ver

el potencial que pueden tener proyectos futuros a medida que avancen nuestras habilidades.

4.5. Valoración Alejandro:

Desde mi punto de vista éste trabajo ha conseguido lograr los objetivos que nos habíamos impuesto desde el principio. Gracias a él hemos podido ir viendo aplicaciones más prácticas a los contenidos dados en la asignatura de Programación I, además de haber sido de gran ayuda a la hora de interiorizar en algunos puntos para solucionar los problemas que nos hemos topado. Sin embargo, creo que el potencial y la utilidad de nuestro programa podría aumentar a medida que nuestros conocimientos en programación, ya sea en bases de datos o programación con objetos, incrementen también.

5. Webgrafía:

Para el primer planteamiento, es decir, antes de dar en clase arrays utilizamos este recurso para visualizar con perspectiva las funciones que se ajustarían al nivel requerido de la asignatura:

- <http://www.homeandlearn.co.uk/java/java.html>
- <https://www.w3schools.com/js>
- Apuntes de clase y material de estudio del Moodle de Programación I