

# Introducción a las ciencias de la computación:

## Proyecto 2

Alexander Adalid Castillo Romero

Diciembre 16, 2023

Nombre: Alexander Adalid Castilo Romero

Numero de cuenta: 316159023.

### **Proyecto gestor de colecciones de entretenimiento personal.**

La manera en que desarrollé el proyecto parte de generar primero en una carpeta para las clases del modelo que se llama *artículos*. En esta carpeta se incluye una superclase titulada *Articulo* que consta de 2 elementos que posee cada una de sus subclases, que son el titulo y el genero del articulo, además de sus métodos para obtener datos de un objeto o registrar datos en un objeto. Además de ello incluye un método que sirve para verificar si la clase ya posee la cantidad máxima de artículos indicada en los requerimientos del proyecto.

Esta clase se extiende a 3 subclases, que son Libro, Pelicula y Disco. Estas clases poseen los atributos faltantes para cada tipo de articulo en particular, en el caso de Libro sus dos atributos extras son de tipo **String**, mientras que pelicula y disco tienen un atributo **String** y uno de tipo **int**, que en el caso de las películas es el año de filmación y en el caso de los discos es el numero de pistas. Cada una de estas clases posee un método que a partir de los datos proporcionados genera una cadena de texto con cada uno de los datos separados por comas y ordenados de acuerdo a como se imprimen los elementos de cada colección.

También se cuenta con una carpeta para manejar las excepciones que son subclases de la clase *Exception*, las cuales incluyen los constructores y sus mensajes. Se posee otra carpeta que venia contenida en el empaquetado cuyo nombre es *files* y posee una clase llamada *ReaderWriter* que permite leer y escribir sobre archivos de tipo *.csv* que además cuenta con su manejo de excepciones para los errores que se puedan dar en la lectura o escritura de cadenas en los archivos.

Se creó una carpeta adicional para el gestor de búsqueda llamada *modeloBusqueda*, la cual posee una clase llamada *Busqueda*. Esta clase posee un constructor y 5 métodos. El primer método recibe el nombre de un archivo de texto y qué dato de cada articulo es el que se imprimirá, para con un ciclo **for** leer el archivo completo, con el método **StringTokenizer** separar cada linea en elementos delimitados por comas, los elementos se organizan y dependiendo el tipo de articulo es la etiqueta que se les asigna a la hora de imprimirlos, el ciclo

imprime todos los datos de un articulo particular y repite el proceso hasta que se repite la información de todos los artículos.

Los siguientes métodos funcionan de manera un poco similar, estos reciben 6 cadenas como parámetros, las primeras 5 cadenas son las mismas que el método para la colección completa, pero la sexta es un **String** que el usuario ingresa para hacer una búsqueda particular en torno a algún campo del articulo. El ciclo **for** separa cada una de las lineas utilizando **StringTokenizer** pero en lugar de imprimir todas las cadenas se utiliza un condicional que compara si el token de un dato particular coincide con el introducido por el usuario, si coincide entonces imprime todos los datos de ese articulo particular, si no hay ninguna coincidencia el programa continua con su ejecución. Los 4 metodos funcionan del mismo modo solo que cada uno hace una comparacion con alguno de los tokens en particular.

La clase Prueba es la que contiene al método main, esta empieza dando un mensaje de bienvenida que ocurre solo cuando se inicializa el programa, después se imprime un menú que permite seleccionar si se desean gestionar libros, películas o discos. el menú está implementado con un ciclo **do while** que permite que se siga ejecutando siempre y cuando la opción que se seleccione no sea 0, dentro del ciclo **do while** se encuentra un ciclo **switch** para cada una de las 4 opciones, que si recibe un **int** que no se encuentre entre 0 y 3 devuelve un mensaje que solicita una opción válida y en caso de recibir una entrada que no sea de tipo **int** se maneja con una excepción que de igual modo solicita una opción válida.

En el caso 1 se gestionan libros, dentro de ese caso primero se declara una variable que contiene el nombre del archivo *.csv* en donde se guardará la colección y se declara una variable de tipo **int** para un futuro ciclo **switch**. Después de ello se encuentra un ciclo **do while** que se ejecutará siempre que se seleccione una opción distinta de 0, lo que permite ejecutar otro menú implementado con un ciclo **switch** que utiliza como opción la variable declarada al inicio del caso. Esto permite que si se selecciona 0 en lugar de que el programa termine su ejecución, este menú anidado finalice y se permita regresar al menú inicial. Además el menú anidado envía un mensaje que solicita una opción válida si no se introduce un **int** entre 0 y 5 y si se recibe algún otro tipo se maneja la excepción.

El menú anidado posee 5 opciones que consisten en 3 consultas dado un dato particular, el despliegue de la colección completa y el registro de un nuevo elemento. Las primeras 3 opciones en primer lugar envían un mensaje que solicita al usuario que introduzca el dato a consultar, se almacena y se crea una nueva búsqueda a la cual se le asignan los parámetros correspondientes, es decir, el nombre del archivo declarado al inicio del caso, las etiquetas de los datos y el dato introducido por el usuario. El método particular a invocar depende de la posición que este tenga en el registro y en que token se almacenó. Si ningún elemento coincide con la busqueda no se imprime nada y se continua con la ejecución. Si uno o mas coinciden, se despliega su información con el formato solicitado. La opción de desplegar la colección completa genera un nuevo objeto de tipo *Busqueda* e invoca el método que despliega la información de todos los

elementos en el archivo, para ello solo requiere las etiquetas de cada token y el nombre del archivo a buscar.

La opción 5 primero declara una variable que servirá para verificar con un ciclo **for** si existe un elemento con el mismo título. Después genera un nuevo objeto de tipo **Libro** e invoca el método de la clase **Articulo** que permite identificar si la colección de archivos tiene más de 32 elementos registrados. Si de hecho ya están registrados 32 elementos imprime un mensaje que no permite que se registren nuevos elementos y termina la ejecución del registro de un nuevo libro. En caso de que aún se puedan registrar elementos procede a solicitar el título, con ello hace una búsqueda en los elementos almacenados, con un ciclo **for** y el método **read** obtiene el token de la posición del título y lo compara con el introducido por el usuario, si hay coincidencias se le indica al usuario que ya existe un elemento con ese título y termina el proceso de registro de un nuevo artículo. Cuando el título no está repetido se invoca el método **setTitulo** y se registra, para continuar con el resto de datos que no requieren comprobación de repetición.

Una vez obtenidos todos los datos, que serán registrados con sus respectivos métodos **set**, se invoca al método que concatena todos los datos en una sola cadena donde cada dato se separa con comas y este se guarda en el archivo correspondiente utilizando el método **write**. Al finalizar se regresa al menú de gestión de libros y el nuevo elemento se puede consultar ya sea en la colección completa o buscándolo por alguno de sus parámetros. Este ciclo correrá indefinidamente hasta que el usuario elija explícitamente regresar al menú anterior.

La gestión de películas y discos funciona de manera muy similar en la mayoría de los casos, excepto en los siguientes: los discos permiten hacer una búsqueda para los elementos que tengan más de 10 pistas. Este proceso se realiza como los otros métodos de búsqueda, pero como los archivos **.csv** y **StringTokenizer** devuelven cadenas se utiliza un método que permite pasar de **String** a **int**. Este proceso se repite con un ciclo **for** y para todos los elementos cuyo número de pistas sea mayor a 10 se despliega su información.

En el momento de registrar el objeto todo funciona como en el primer caso, solo que como para los discos uno de los elementos es de tipo **int** y se necesita en **String** se utiliza un método que lo convierte a cadena, se almacena, se le da formato con el método **newDisco()**; y se guarda. En caso de que el usuario ingrese una opción inválida se maneja con una excepción. Este mismo proceso se repite para registrar el año de filmación para el caso de las películas, por lo que el funcionamiento del gestor de películas es muy similar al del gestor de discos. Todos estos menús para gestionar cada tipo de artículo están anidados con un ciclo **do while** que permitirá regresar al menú inicial cuando el usuario lo pida.

Todas las clases cuentan con documentación realizada con **javadoc** y con comentarios para explicar que acciones realizan los métodos y cuál es la función de los ciclos y variables declaradas en ellos. La clase **main** cuenta con los comentarios que explican cuál es la función de los ciclos y métodos utilizados en ella.