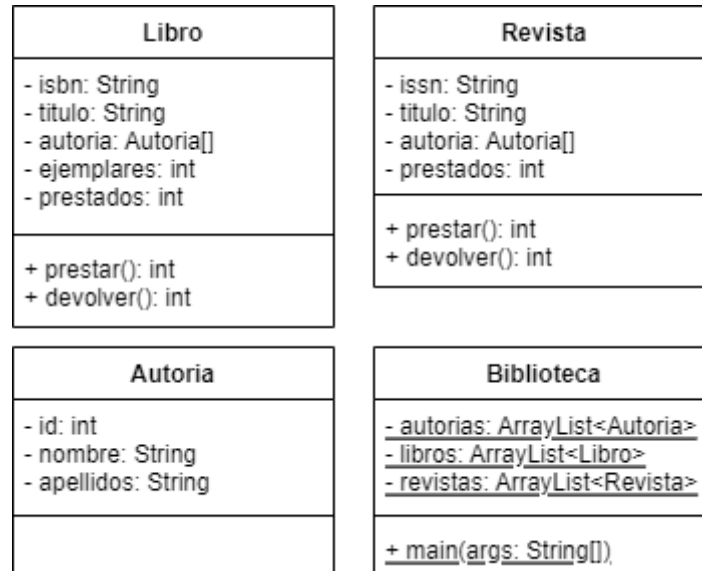


## Práctica 2 de PROGRAMACIÓN: Clases y arrays

La tarea se trata de hacer un sencillo **sistema de gestión de una biblioteca** en la que se pueden prestar libros y revistas. Cada libro y revista tiene asociado una o varias autorías.



La **clase Libro** tendrá los siguientes atributos:

- ISBN: Identificador único de cada libro.
- Título: de tipo *String*.
- Autoría: de tipo array de *Autoria*.
- Ejemplares: de tipo entero. Contiene el número total de ejemplares de dicho libro en la biblioteca, estén en préstamo o no.
- Prestados: de tipo entero. Contiene el número de ejemplares prestados de dicho libro.

Tendrá dos métodos:

- Prestar: Comprueba que hay libros disponibles para poder hacer el préstamo. El método devuelve un número entero con el número de libros disponibles en la biblioteca tras el préstamo, o -1 en caso de que no se haya podido hacer (todos los libros estaban prestados).
- Devolver: Comprueba que había ejemplares prestados para hacer la devolución. El método devuelve un número entero con el número de libros disponibles en la biblioteca tras esta devolución, o -1 en caso de que no se haya podido hacer (no había libros prestados).

La **clase Revista** tendrá los siguientes atributos:

- ISSN: Identificador único de cada revista.
- Título: de tipo *String*.
- Autoría: de tipo array de *Autoria*.
- Prestados: de tipo entero.

Para las revistas suponemos que hay ejemplares ilimitados. Tiene los métodos prestar y devolver, con el mismo funcionamiento que en la clase Libro. Se llevará la cuenta de los ejemplares prestados con el correspondiente atributo.

La **clase Autoría** tendrá tres atributos: el id (identificador único de cada autora o autor, de tipo entero) y el nombre y los apellidos (de tipo *String*). No tiene métodos.

La **clase Biblioteca** tendrá el *main*. Además, tiene tres atributos que serán un *ArrayList* de autores, libros y revistas, siendo todos ellos atributos estáticos.

### Programación de las clases

Programa las clases en Java, todas ellas dentro de un paquete que se llame “bibliotecaXxxxYyyy”, siendo Xxxx tu nombre, e Yyyy tu apellido.

De cada clase, **crea solamente los constructores, *getters* y *setters* que necesites**, de manera que el funcionamiento de tus clases sea lo más sencillo posible. Argumenta a través de comentarios en el código por qué has elegido dichos *getters*, *setters* y constructores. Piensa también si necesitas poner algún requisito en los *setters* para evitar la corrupción de datos.

Todas las clases sobrescribirán el método **toString()**, imprimiendo la información de sus atributos.

Libro, Autoria y Revista no tendrán salidas ni entradas por pantalla (ningún *System.out* ni *Scanner*).

### Programación del *main*

El *main*, que está dentro de la clase **Biblioteca**, tendrá el siguiente funcionamiento:

Mostrará por pantalla el siguiente menú:

1. Crear autor/a
2. Ver autoras/es
3. Crea el libro
4. Mostrar libros
5. Presta un libro
6. Devuelve un libro
7. Crea la revista
8. Mostrar revistas
9. Presta una revista
10. Devuelve una revista
0. Salir

Con la **opción 1** se podrá crear una autora o autor. Para ello, el programa pedirá por pantalla introducir el id de la nueva o nuevo autor (teniendo en cuenta que no se pueden repetir los id), el nombre y los apellidos. Con esta información, se añadirá al *ArrayList* *autorias*.

Con la **opción 2** se mostrarán por pantalla todas y todos los autores del *ArrayList*.

En caso de elegir la **opción 3**, se creará un objeto de la clase Libro que se añadirá al *ArrayList* *libros*. Se pedirá por pantalla que se introduzcan los siguientes campos en el siguiente orden:

1. ISBN (Se tiene que verificar que no existía ya un libro con dicho ISBN)
2. Título
3. Número de ejemplares
4. Número de autoras/es que tiene el libro

5. Por cada una de las autorías se pedirá que se introduzca el *id* del autor o autora correspondiente, para añadirlo al array *Autoria* de dicho libro. Tendréis que comprobar que el *id* exista.

Con la **opción 4** se mostrará por pantalla todos los libros disponibles en la biblioteca, mostrando todos sus datos.

Con la **opción 5** se realizará el préstamo de un libro. Para ello, en primer lugar, se pedirá por pantalla que se introduzca el ISBN del libro a prestar. En caso de que exista dicho libro, se realizará el préstamo del mismo, mostrando por pantalla el número de ejemplares disponibles tras el préstamo, o un mensaje avisando que no quedaban libros disponibles para prestar. En caso de que dicho ISBN no exista, puedes decidir qué hacer (volver a pedir un ISBN válido, ir al menú principal, etc.).

La **opción 6** será semejante a la anterior: se pedirá el ISBN de un libro existente, se realizará su devolución y se mostrará por pantalla el número de ejemplares disponibles o un mensaje avisando de que no había ninguno prestado.

Las **opciones 7, 8, 9 y 10** son las mismas aplicadas a las revistas en lugar de a los libros.

El menú se repetirá constantemente hasta que se elija la **opción 0**. Con cualquier otro número que no exista en el menú, se repetirá por pantalla.

A tener en cuenta en general en toda la práctica:

- Importante para todo el programa la utilización de funciones. Es imprescindible que el **código esté perfectamente encapsulado en funciones** evitando al máximo la repetición de código. Cada opción del menú debería llevar a una función distinta, y además seguramente necesites otras muchas funciones (buscar libros y revistas, por ejemplo). Piensa bien los argumentos y tipos de retorno para que tu programa tenga sentido.
- Tendrás que realizar correctamente el **Javadoc de todas las funciones y clases**.
- Podéis cambiar los tipos que consideréis adecuados, manteniendo la lógica general, y justificando dichos cambios a través de comentarios en el código. Es necesario que utilicéis arrays primitivos (con corchetes) y de alguna de las clases de Java (ArrayList, HashMap, etc.).
- Si el programa entregado **tiene errores en el código se restará 3 puntos** a la nota final.
- Parte del alumnado **corregirá la práctica en clase**. Si se detecta que no se comprende su propio código o no se sabe hacer modificaciones sencillas **se podrá restar hasta 3 puntos de la nota**, además de anular los ítems correspondientes que no se conocen.
- No se aceptan entregas atrasadas.
- Para la **entrega** tendrás que adjuntar todos los ficheros .java que hayas creado. En principio serán Libro.java, Biblioteca.java, Revista.java y Autoria.java, a no ser que hayas utilizado alguna clase extra.