



# Ejercicio de Alta Dificultad: Sistema de Gestión de Biblioteca



Tiempo estimado: 2 horas



Dificultad: Alta



Temas que aplicarás:

- Sintaxis y estructuras básicas
  - Clases, objetos y encapsulamiento
  - Herencia y polimorfismo
  - Uso de `ArrayList`, `HashMap`, `HashSet`
  - Buen diseño de clases
- 



## Enunciado:

Vas a desarrollar un **sistema de gestión de una biblioteca** donde puedas:

1. Agregar libros a la biblioteca.
  2. Registrar usuarios (lectores).
  3. Permitir que los usuarios puedan **tomar prestado y devolver libros**.
  4. Llevar un registro de los libros prestados y disponibles.
  5. Buscar libros por título, autor o género.
- 



## Estructura sugerida:



### Clase `Libro`

- `String titulo`
- `String autor`
- `String genero`
- `String ISBN (único)`
- `boolean estaPrestado`



### Clase `Usuario`

- `String nombre`
- `String idUsuario (único)`
- `ArrayList<Libro> librosPrestados`



### Clase `Biblioteca`

- `HashMap<String, Libro> catalogo (clave = ISBN)`
- `HashMap<String, Usuario> usuarios (clave = idUsuario)`

### ✂ Funcionalidades requeridas:

- `void agregarLibro(Libro libro)`
  - `void registrarUsuario(Usuario usuario)`
  - `void prestarLibro(String isbn, String idUsuario)`
  - `void devolverLibro(String isbn, String idUsuario)`
  - `void mostrarLibrosDisponibles()`
  - `void buscarPorTitulo(String titulo)`
  - `void buscarPorAutor(String autor)`
  - `void buscarPorGenero(String genero)`
- 

### ✓ Requisitos adicionales:

- Usa encapsulamiento y modificadores de acceso correctamente (`private`, `public`, etc.).
- Implementa `toString()` en las clases `Libro` y `Usuario`.
- Controla que no se puedan prestar libros que ya están prestados.
- Controla que no se pueda devolver un libro que no ha sido prestado.
- Usa correctamente las colecciones (`ArrayList`, `HashMap`).