

## ABP: Sistema de Biblioteca

### Contexto

La Biblioteca Municipal "El Saber" ha decidido modernizar su sistema de préstamos de libros, el cual hasta ahora se ha gestionado manualmente con fichas de papel. Esto ha ocasionado problemas como la pérdida de fichas, dificultad para realizar seguimientos a los préstamos, y la posibilidad de que un mismo libro sea prestado más de una vez al mismo tiempo.

El director de la biblioteca ha solicitado desarrollar un sistema de software que permita gestionar los libros disponibles, los usuarios que hacen uso de los servicios y los préstamos de libros. El objetivo es mejorar la organización de la biblioteca y evitar conflictos con el préstamo de ejemplares.

### Objetivo:

El objetivo es desarrollar un **programa orientado a objetos** que permita:

- Gestionar los préstamos y devoluciones de libros por parte de un usuario.
- Asegurar que un libro no pueda ser prestado a más de una persona a la vez.
- Mantener la privacidad de los datos sensibles de los usuarios (encapsulamiento).
- Permitir la colaboración entre distintas clases, como `Libro`, `Usuario` y `Prestamo`.

### Especificaciones del sistema:

1. **Libros:** Cada libro tiene un título, autor y un número de identificación único (ID).
2. **Usuarios:** Cada usuario tiene un nombre y un número de identificación único. Los usuarios deben poder solicitar préstamos y devolver libros.
3. **Préstamos:** Cada préstamo debe registrar qué usuario pidió un libro y en qué fecha. No se debe permitir prestar un libro si ya está en préstamo.
4. **Requerimientos funcionales:**
  - El programa debe permitir crear nuevos libros y usuarios.
  - Debe asegurarse que los datos de los usuarios y libros estén encapsulados, de manera que no puedan ser modificados directamente.
  - Permitir la devolución de libros.

### Pasos para abordar el problema:

#### 1. Análisis del problema:

- Discute cómo representarías los libros, usuarios y préstamos.
- ¿Qué atributos y métodos deben tener las clases `Libro`, `Usuario` y `Prestamo`?
- ¿Cómo asegurarías que un libro no pueda ser prestado más de una vez hasta que se devuelva?

#### 2. Definición de clases y atributos:

- Diseña las clases `Libro`, `Usuario` y `Prestamo`.
- Define los atributos y métodos de cada clase, asegurándote de encapsular los datos sensibles (como los detalles del libro y los datos del

usuario).

### 3. Implementación:

- Codifica las clases aplicando los conceptos de **abstracción**, **encapsulamiento** y **colaboración**.
- Asegúrate de que la clase `Prestamo` gestione las relaciones entre `Libro` y `Usuario`.

### 4. Validación:

- Asegúrate de que no sea posible prestar un libro que ya esté prestado.
- Implementa la lógica para devolver libros y actualizar el estado del préstamo.