

SISTEMA DE GESTIÓN

DE BIBLIOTECA

Proyecto Ingeniería del Software II

## Descripción breve

El Sistema de Gestión de Biblioteca permite a los usuarios buscar, reservar y devolver libros de manera eficiente. Implementa patrones de diseño para garantizar escalabilidad y mantenibilidad, además de integrar notificaciones y distintos roles de

usuario. Su desarrollo en Java lo hace modular y flexible.

## Pablo Carrasco Melero, Pablo Monda Cana, Pablo Aranda Cortés

[pcarmel@alu.upo.es,](mailto:pcarmel@alu.upo.es) [pmoncan@alu.upo.es,](mailto:pmoncan@alu.upo.es) [paracor@alu.upo.es](mailto:paracor@alu.upo.es)

# Enunciado:

Diseñe e implemente un sistema de gestión bibliotecaria basado en el siguiente diagrama UML, aplicando los patrones de diseño indicados. El sistema debe incluir las siguientes funcionalidades y requisitos:

# Clases y relaciones principales:

* Implementa las clases Book, LibrarySystem, Loan, User y sus subtipos (Student, Professor, Researcher).
* Gestiona los estados de los libros

(AvailableState, BorrowedState, ReservedState, DamagedState) mediante el **patrón State**, garantizando transiciones encapsuladas (ej: préstamo, devolución, reserva).

# Patrones de diseño:

***Singleton:*** Asegura que solo exista una instancia de LibrarySystem.

***Observer:*** Notificamos únicamente a los investigadores (Researcher) cuando un libro de su área de investigación esté disponible.

***Factory Method:*** SimpleUserFactory para crear usuarios académicos (estudiantes, profesores, investigadores).

***Strategy:*** Estrategias de préstamo

(StandardLoanStrategy, ExtendedLoanStrategy, RestrictedLoanStrategy) para calcular fechas de devolución (7, 14 o 30 días).

***Iterator:*** BookIterator, UserIterator y LoanIterator para recorrer colecciones de libros, usuarios y préstamos.

***State:*** AvaibleState, BorrowedState, ReservedState, DamagedState representan los diferentes tipos de estados que puede tener un libro, los cuales son implementados desde la interfaz IBookState.