



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, pmoncan@alu.upo.es, 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 requeridos:

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.

Funcionalidades adicionales:

- Registro y eliminación de usuarios/libros/préstamos en LibrarySystem.
- Clasificación jerárquica de libros mediante la clase *Category* (categorías y subcategorías).
- Validación de operaciones según el estado actual del libro (ej: no prestar un libro dañado).

Requisitos técnicos:

- Encapsule las operaciones de notificación (ISubject, IObserver) y la lógica de préstamos.
- Aplique el principio de inversión de dependencias en la interfaz IBookState para los estados del libro.

Entregables:

- Código fuente en Java que refleje el diagrama UML.
- Pruebas unitarias para validar las transiciones de estado, notificaciones y cálculos de fechas de préstamo.
- Documentación breve explicando cómo se aplicó cada patrón de diseño en la implementación.