**Justificación del diseño del diagrama de casos de uso**

Este diagrama representa una primera versión del análisis funcional del sistema "Gestor de bibliotecas", centrado en los casos de uso más relevantes y frecuentes según los requisitos del proyecto. Se han identificado tres actores principales: **Socio**, **Bibliotecario** y **Administrador**, cada uno con un conjunto de funcionalidades diferenciadas según su rol en el sistema.

Los casos de uso seleccionados responden a la gestión básica de usuarios, libros, préstamos y reservas, incluyendo acciones como registrar devoluciones, generar informes o gestionar penalizaciones. Se ha incorporado la notación UML mediante relaciones <<include>> y <<extend>> para representar dependencias funcionales entre casos.

Al tratarse de una fase preliminar de diseño, se han priorizado los casos de uso fundamentales, dejando algunos detalles de implementación o funciones complementarias como futuras ampliaciones. El objetivo ha sido lograr una representación clara, coherente y funcional del sistema desde una perspectiva de estudiante en formación.

@startuml

title Gestor de bibliotecas

actor Socio

actor Bibliotecario

actor Administrador

usecase "Consultar catálogo" as UC1

usecase "Ver histórico de préstamos" as UC2

usecase "Reservar ejemplar" as UC3

usecase "Ver estado de reservas" as UC4

usecase "Cancelar reserva" as UC5

usecase "Registrar préstamo" as UC6

usecase "Registrar devolución" as UC7

usecase "Registrar o modificar usuario" as UC8

usecase "Modificar libro" as UC9

usecase "Ver incidencias" as UC10

usecase "Gestionar penalizaciones" as UC11

usecase "Generar informes" as UC12

Socio --> UC1

Socio --> UC2

Socio --> UC3

Socio --> UC4

Socio --> UC5

Bibliotecario --> UC1

Bibliotecario --> UC6

Bibliotecario --> UC7

Bibliotecario --> UC8

Bibliotecario --> UC9

Bibliotecario --> UC10

Bibliotecario --> UC11

Administrador --> UC1

Administrador --> UC10

Administrador --> UC12

UC2 .extends. UC1

UC3 .includes. UC1

UC4 .extends. UC3

UC5 .extends. UC3

@enduml

**Justificación del diseño del diagrama de clases**

Este diagrama de clases representa la estructura lógica del sistema "Gestor de Bibliotecas", construido según los requisitos planteados en el enunciado del proyecto. Se ha diseñado con orientación a objetos y notación UML, organizando las clases y relaciones para cubrir todas las funcionalidades esenciales del sistema.

Se parte de una clase base `Usuario`, de la que heredan `Socio`, `Bibliotecario` y `Administrador`, diferenciando claramente los roles del sistema. Esta jerarquía permite separar responsabilidades y adaptar los métodos a las funciones específicas de cada perfil. Por ejemplo, `Socio` puede solicitar préstamos y calcular penalizaciones, mientras que `Bibliotecario` se encarga de registrar préstamos y devoluciones, y `Administrador` de generar informes.

El sistema gestiona libros mediante las clases `Libro` y `Ejemplar`, diferenciando entre obra y copia física. La clase `Libro` incluye el método `buscarPorLibro()` como parte de la funcionalidad de consulta del catálogo. Las clases `Préstamo`, `Reserva` y `Penalización` reflejan operaciones clave para el control de ejemplares y usuarios, todas ellas con atributos y métodos adecuados.

Las relaciones están representadas mediante cardinalidades y conexiones lógicas entre entidades. Aunque algunas notaciones podrían mejorarse formalmente, el modelo cumple con los criterios de claridad, coherencia estructural y alineación con los casos de uso definidos.

En conclusión, el modelo proporciona una base sólida y bien organizada del sistema desde una perspectiva de análisis orientado a objetos.

@startuml

class Usuario {

+id: int

+nombre: string

+email: string

+contrasena: string

+iniciarSesion(): boolean

+cerrarSesion(): void

}

class Socio {

+fechaAlta: Date

+pedirPrestamo(): boolean

+calculoPena(): int

}

class Bibliotecario {

+regPrestamo(): void

+regDevolucion(): void

+modificarLibro(): void

}

class Administrador {

+fechaAlta: Date

+generarInforme(): string

}

class Libro {

+isbn: string

+titulo: string

+year: int

+genero: string

+buscarPorLibro(): int, string

}

class Ejemplar {

+codigoInventario: string

+estado: string

+ubicacion: string

+sePuedeReservar(): boolean

}

class Prestamo {

+idPrestamo: string

+fechaInicio: string

+registrar(): void

+devolver(): void

+vencido(): boolean

}

class Reserva {

+idReserva: string

+fechaReserva: Date

+estado: string

+crear(): void

+cancelar(): void

+pasarAprestamo(): void

}

class Penalizacion {

+motivoPena: string

+diasPena: int

+penaActiva: boolean

+estaActiva(): boolean

}

Usuario <|-- Socio

Usuario <|-- Bibliotecario

Usuario <|-- Administrador

Libro "1" \*-- "1..\*" Ejemplar

Socio "1" --> "0..\*" Prestamo

Prestamo "1" --> "1" Ejemplar

Socio "1" --> "0..\*" Reserva

Reserva "1" --> "1" Ejemplar

Socio "1" --> "0..\*" Penalizacion

@enduml