Taller #1

Programación Distribuida

INTEGRANTES:

Felipe Olaya Benítez

Juan Sebastián Restrepo Nieto

Profesor:

Daniel Andrey Villamizar Aranque



 Instituto Tecnológico Metropolitano – ITM

Medellín

2025-2

# Sistema de Gestión de Biblioteca

## 1. Caso de Negocio

### Problema

La biblioteca municipal "San José" actualmente gestiona sus préstamos de libros utilizando un sistema manual basado en papel y lápiz. Este método presenta varios inconvenientes: pérdida de información, errores en el registro de fechas de devolución, dificultad para consultar el estado de los libros, y falta de control sobre préstamos vencidos. La bibliotecaria requiere un sistema que le permita mantener un registro digital de usuarios, libros y préstamos de manera eficiente y confiable.

### Usuarios

* **Bibliotecario:** Personal encargado de registrar usuarios, libros, préstamos y devoluciones. Es el único usuario del sistema.
* **Usuario de Biblioteca:** Persona que solicita préstamos de libros (no interactúa directamente con el sistema).

### Valor Esperado

Un sistema de consola que permita al bibliotecario gestionar de manera digital y eficiente el catálogo de libros, el registro de usuarios, y el control de préstamos y devoluciones, eliminando los errores del sistema manual y proporcionando información actualizada sobre el estado de los libros y préstamos.

## 2. Historias de Usuario

### Historia 1: Registro de Usuarios

**Como bibliotecario, quiero registrar usuarios en el sistema para tener control de quién puede prestar libros.**

**Criterios de Aceptación:**

* Given un usuario nuevo con datos válidos
* When ingreso su ID único y nombre completo
* Then el usuario queda registrado en el sistema y disponible para préstamos

### Historia 2: Gestión de Catálogo

**Como bibliotecario, quiero agregar libros al catálogo para que estén disponibles para préstamo.**

**Criterios de Aceptación:**

* Given un libro nuevo con información completa
* When ingreso título, autor e ISBN único
* Then el libro queda disponible en el catálogo para préstamo

### Historia 3: Registro de Préstamos

**Como bibliotecario, quiero registrar un préstamo para controlar qué libros están prestados.**

**Criterios de Aceptación:**

* Given un usuario registrado y un libro disponible
* When registro el préstamo con fecha de devolución
* Then el libro queda marcado como prestado y no disponible

### Historia 4: Registro de Devoluciones

**Como bibliotecario, quiero registrar una devolución para liberar libros prestados.**

**Criterios de Aceptación:**

* Given un préstamo activo en el sistema
* When registro la devolución del libro
* Then el libro queda disponible nuevamente para préstamo

### Historia 5: Consulta de Libros Disponibles

**Como bibliotecario, quiero ver libros disponibles para saber qué puedo prestar.**

**Criterios de Aceptación:**

* Given que hay libros en el catálogo
* When consulto libros disponibles
* Then veo la lista de libros no prestados

### Historia 6: Control de Préstamos Vencidos

**Como bibliotecario, quiero ver préstamos vencidos para hacer seguimiento.**

**Criterios de Aceptación:**

* Given que hay préstamos activos en el sistema
* When consulto préstamos vencidos
* Then veo los préstamos que pasaron la fecha de devolución

## 3. Requerimientos de Negocio

1. **Gestión de Usuarios:** El sistema debe permitir registrar y mantener información de usuarios de la biblioteca.
2. **Catálogo de Libros:** Mantener un inventario digital de todos los libros disponibles en la biblioteca.
3. **Control de Préstamos:** Registrar préstamos con fechas de vencimiento y seguimiento de estado.
4. **Gestión de Devoluciones:** Procesar devoluciones y liberar libros para nuevos préstamos.
5. **Consultas y Reportes:** Proporcionar información sobre disponibilidad de libros y estado de préstamos.

## 4. Requerimientos Funcionales

1. **Registrar Usuarios:** Permitir el registro de usuarios con ID único y nombre completo.
2. **Agregar Libros:** Incorporar nuevos libros al catálogo con título, autor e ISBN único.
3. **Realizar Préstamos:** Crear préstamos asociando usuario, libro y fecha de vencimiento.
4. **Registrar Devoluciones:** Procesar devoluciones y marcar libros como disponibles.
5. **Listar Libros Disponibles:** Mostrar todos los libros que no están prestados.
6. **Listar Préstamos Activos:** Consultar todos los préstamos en curso.
7. **Listar Préstamos Vencidos:** Identificar préstamos que excedieron la fecha de devolución.
8. **Validar Disponibilidad:** Verificar que un libro no esté ya prestado antes de crear un préstamo.

## 5. Requerimientos No Funcionales

1. **Rendimiento:** El sistema debe responder a las consultas en menos de 1 segundo.
2. **Seguridad Básica:** Validar que los IDs de usuarios y libros sean únicos en el sistema.
3. **Logging:** Registrar todas las operaciones principales en la consola para auditoría.
4. **Mantenibilidad:** Código modular, bien documentado y siguiendo principios de POO.
5. **Testabilidad:** Métodos públicos que permitan la ejecución de pruebas unitarias.
6. **UX Consola:** Interfaz de menú simple, clara y con mensajes informativos para el usuario.

## 6. Modelo Conceptual

### Entidades Principales

**Usuario**

* ID (identificador único)
* Nombre (nombre completo)

**Libro**

* ID (identificador único)
* Título
* Autor
* ISBN (código único)
* Disponible (estado booleano)

**Préstamo**

* ID (identificador único)
* Usuario (referencia al usuario)
* Libro (referencia al libro)
* FechaPréstamo
* FechaVencimiento
* Activo (estado booleano)

### Relaciones

* Un Usuario puede tener múltiples Préstamos
* Un Libro puede tener múltiples Préstamos (pero solo uno activo a la vez)
* Un Préstamo relaciona exactamente un Usuario con un Libro

## 7. Diseño POO

### Aplicación de los 4 Pilares

**Abstracción:** Se implementa mediante la clase abstracta ItemBiblioteca que define la estructura común para elementos que pueden ser prestados, y la interfaz IPrestable que establece el contrato para objetos que pueden ser prestados, ocultando los detalles de implementación específicos.

**Herencia:** La clase Libro hereda de ItemBiblioteca, reutilizando propiedades comunes como ID y título, mientras que Prestamo hereda de Transaccion, aprovechando funcionalidad base para operaciones de la biblioteca.

**Polimorfismo:** Se manifiesta a través de la interfaz IPrestable que permite diferentes implementaciones de préstamo sin modificar el código cliente, y mediante la sobrescritura del método EsValida() en la clase Prestamo que extiende el comportamiento de la clase base Transaccion.

**Encapsulamiento:** Todos los atributos son privados y se acceden únicamente a través de métodos públicos, con validaciones en los setters para mantener la integridad de los datos y prevenir estados inconsistentes.

### Arquitectura del Sistema

El sistema implementa una **arquitectura de 3 capas** que separa claramente las responsabilidades:

**Capa de Dominio (Dominio/)**

* Contiene las entidades del negocio y lógica de dominio
* Implementa los 4 pilares de POO
* No tiene dependencias externas

**Capa de Servicios (Servicio/)**

* Contiene la lógica de aplicación
* Coordina las operaciones entre repositorios
* Implementa validaciones de negocio

**Capa de Infraestructura (Infraestructura/)**

* Implementa el patrón Repository
* Maneja la persistencia de datos
* Proporciona abstracciones para acceso a datos

## 8. Plan de Pruebas

|  |  |  |
| --- | --- | --- |
| Caso de Prueba | Entrada | Resultado Esperado |
| **Registrar Usuario** | ID: "U001", Nombre: "Juan Pérez" | Usuario creado correctamente, ID único validado |
| **Agregar Libro** | Título: "El Quijote", Autor: "Miguel de Cervantes", ISBN: "978-84-376-0494-7" | Libro agregado al catálogo, disponible para préstamo |
| **Realizar Préstamo** | Usuario: "U001", Libro: "L001", Días: 7 | Préstamo creado, libro marcado como no disponible |
| **Registrar Devolución** | Préstamo: "P001" | Libro marcado como disponible, préstamo inactivado |
| **Listar Disponibles** | Consulta de libros disponibles | Solo se muestran libros con estado disponible = true |
| **Préstamo Vencido** | Fecha actual > FechaVencimiento | Préstamo identificado como vencido en consulta |
| **Validar ID Duplicado** | ID: "U001" (ya existe) | Sistema rechaza registro con mensaje de error |
| **Préstamo Libro No Disponible** | Libro ya prestado | Sistema rechaza préstamo con mensaje informativo |

### Datos de Prueba

**Usuarios de Prueba:**

* U001: "María González"
* U002: "Carlos Rodríguez"
* U003: "Ana Martínez"

**Libros de Prueba:**

* L001: "Cien Años de Soledad", "Gabriel García Márquez", "978-84-376-0494-7"
* L002: "Don Quijote", "Miguel de Cervantes", "978-84-376-0495-4"
* L003: "El Principito", "Antoine de Saint-Exupéry", "978-84-376-0496-1"

**Escenarios de Préstamo:**

* Préstamo normal: 7 días de duración
* Préstamo largo: 15 días de duración
* Préstamo corto: 3 días de duración