**Práctica 09: Análisis y Diseño de una Biblioteca**

**Nombres:** Alex Travez – Mateo Oviedo

**Fecha de realización:** 18/12/2024

**Fecha de entrega:** 19/12/2024

**Objetivos**

* Comprender y aplicar los principios del análisis y diseño orientado a objetos.
* Diseñar un sistema usando UML, incluyendo diagramas de casos de uso, clases y secuencias.
* Implementar el diseño en código Java utilizando los conceptos de herencia, polimorfismo, encapsulamiento y abstracción.
* Manejar excepciones y realizar pruebas básicas del sistema diseñado.

**Análisis de Requisitos Funcionales**

1. **Gestionar Libros**
   * Permitir la adición, eliminación y búsqueda de libros.
2. **Gestionar Autores**
   * Permitir la adición y búsqueda de autores.
3. **Gestionar Préstamos**
   * Permitir a los usuarios realizar y devolver préstamos de libros.

**1. Diagrama de Casos de Uso**

**Propósito**

El diagrama de casos de uso representa visualmente la interacción entre los usuarios (actores) y el sistema. Muestra todas las funcionalidades principales que el sistema ofrece y quién puede acceder a cada una de ellas.

**Componentes Principales**

1. **Actores:** 
   * Administrador: Gestiona todo el sistema, incluyendo libros y autores.
   * Usuario: Interactúa con el sistema para préstamos y consultas.
2. **Casos de Uso Principales:** 
   * Gestión de Libros:
     + Agregar Libro
     + Eliminar Libro
     + Buscar Libro
     + Modificar Libro
   * **Gestión de Autores:** 
     + Agregar Autor
     + Buscar Autor
     + Actualizar Autor
   * **Gestión de Préstamos:** 
     + Realizar Préstamo
     + Devolver Libro
     + Consultar Préstamos

**Relaciones**

* Las relaciones incluyen e indica funcionalidades obligatorias.
* Las relaciones entre actores y casos de uso muestran permisos.
* Las notas proporcionan información adicional sobre responsabilidades.

Diagrama

Descripción generada automáticamente

Ilustración 1. Diagrama de casos de uso-Biblioteca

**2. Diagrama de Clases**

**Propósito**

El diagrama de clases muestra la estructura del sistema, definiendo las clases, sus atributos, métodos y las relaciones entre ellas. Representa el diseño orientado a objetos del sistema.

**Clases Principales**

1. **Biblioteca:** 
   * Clase central que gestiona todo el sistema
   * Contiene listas de libros, autores y préstamos
   * Métodos para todas las operaciones principales
2. **Libro:** 
   * Atributos: título, ISBN, autor, estado
   * Métodos para acceder y modificar información
   * Enum EstadoLibro para control de disponibilidad
3. **Autor:** 
   * Información del autor: nombre, nacionalidad
   * Relación con sus libros escritos
   * Métodos para gestionar información
4. **Préstamo:** 
   * Gestiona información de préstamos
   * Control de fechas y estados
   * Enum EstadoPrestamo para seguimiento

**Relaciones**

* Composición (rombo negro): Biblioteca contiene Libros, Autores y Préstamos
* Asociación (flecha): Préstamo referencia a Libro, Libro referencia a Autor
* Enumeraciones: EstadoLibro y EstadoPrestamo para control de estados

Diagrama

Descripción generada automáticamente

Ilustración 2. Diagrama de clases - Sistema de Biblioteca

**3. Diagrama de Secuencia - Realizar Préstamo**

**Propósito**

Muestra la interacción temporal entre los diferentes objetos del sistema durante el proceso de realizar un préstamo, incluyendo el flujo de mensajes y las validaciones necesarias.

**Flujo Principal**

1. **Inicio del Préstamo:** 
   * Usuario solicita realizar préstamo
   * Sistema solicita ISBN
   * Usuario proporciona ISBN
2. **Validaciones:** 
   * Verificación de existencia del libro
   * Verificación de disponibilidad
   * Validación de usuario
3. **Registro del Préstamo:** 
   * Creación del registro de préstamo
   * Actualización del estado del libro
   * Confirmación al usuario

**Flujos Alternativos**

* Manejo de libro no encontrado
* Manejo de libro no disponible
* Validaciones de usuario

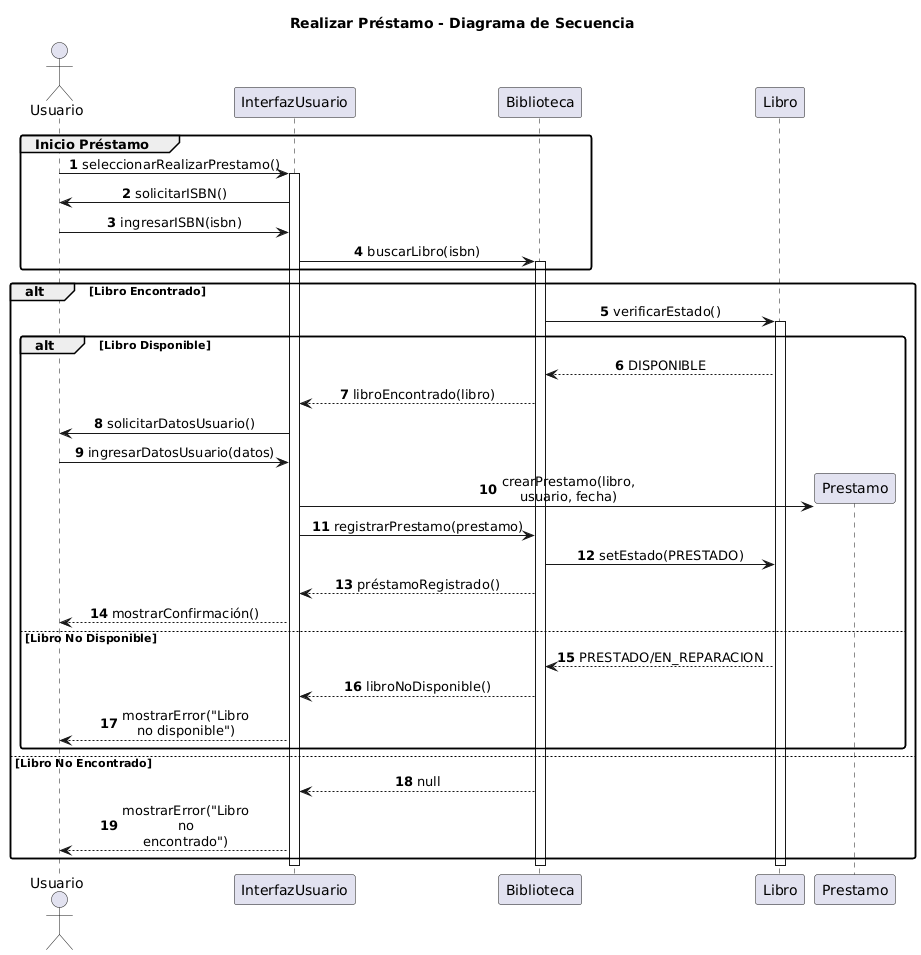


Ilustración 3. Diagrama de secuencias -Realizar Prestamo

**4. Diagrama de Secuencia - Agregar Libro**

**Propósito**

Detalla la interacción temporal durante el proceso de agregar un nuevo libro al sistema, incluyendo la validación y creación de autor si es necesario.

**Flujo Principal**

1. **Ingreso de Datos:** 
   * Administrador inicia proceso
   * Sistema solicita información
   * Validación de datos ingresados
2. **Validación de Autor:** 
   * Búsqueda de autor existente
   * Creación de nuevo autor si es necesario
3. **Registro de Libro:** 
   * Validación de ISBN único
   * Creación del libro
   * Confirmación del registro

Diagrama

Descripción generada automáticamente

Ilustración 4. Diagrama de secuencias - Agregar Libro

**5. Diagrama de Estados - Ciclo de Vida del Libro**

**Propósito**

Muestra los diferentes estados por los que puede pasar un libro en el sistema y las transiciones permitidas entre estos estados.

**Estados Principales**

1. **Registrado:** 
   * Disponible
   * En Préstamo
   * En Reparación
   * Dado de Baja
2. **En Préstamo:** 
   * Préstamo Normal
   * Préstamo Vencido
3. **Estados Finales:** 
   * Dado de Baja
   * Perdido

**Transiciones**

* Entre estados principales
* Condiciones para cambios de estado
* Estados terminales

**Uso del Sistema**

El sistema está diseñado para mantener un control riguroso de:

1. Inventario de libros y su estado actual
2. Información de autores y sus obras
3. Préstamos activos y su historial
4. Estados y disponibilidad de materiales

**Consideraciones de Implementación**

1. **Base de Datos:** 
   * Necesaria para persistencia de datos
   * Relaciones entre tablas según diagrama de clases
2. **Interfaz de Usuario:** 
   * Separada para administradores y usuarios
   * Validaciones en cliente y servidor
3. **Seguridad:** 
   * Control de acceso por rol
   * Validación de datos en cada operación
4. **Mantenimiento:** 
   * Logs de operaciones
   * Respaldo de información
   * Diagrama

     Descripción generada automáticamenteGestión de estados

Ilustración 5. Diagrama de Estados - Ciclo de vida del libro

**Conclusiones**

1. **Arquitectura Modular**
   * El sistema desarrollado demuestra una arquitectura modular bien estructurada, donde cada componente (Libro, Autor, Préstamo) está claramente definido con sus responsabilidades específicas.
   * La separación de responsabilidades facilita el mantenimiento y la escalabilidad futura del sistema.
2. **Gestión Eficiente**
   * Los diagramas UML desarrollados muestran un flujo de trabajo ordenado y eficiente para todas las operaciones del sistema.
   * La implementación de estados y validaciones asegura la integridad de los datos y las operaciones.
3. **Flexibilidad del Sistema**
   * El diseño permite la fácil incorporación de nuevas funcionalidades sin afectar las existentes.
   * Los patrones de diseño utilizados facilitan la adaptación del sistema a diferentes contextos bibliotecarios.

**Recomendaciones**

1. **Implementación de Seguridad**
   * Se recomienda implementar un sistema robusto de autenticación y autorización.
   * Incorporar registro de auditoría para todas las operaciones críticas del sistema.
   * Implementar encriptación para datos sensibles.
2. **Mejoras en la Interfaz de Usuario**
   * Desarrollar una interfaz web responsiva para mejor accesibilidad.
   * Implementar notificaciones automáticas para préstamos vencidos.
   * Agregar funcionalidades de búsqueda avanzada y filtros.
3. **Expansión de Funcionalidades**
   * Incorporar un sistema de reservas de libros.
   * Implementar un módulo de estadísticas y reportes.
   * Agregar integración con sistemas externos (catálogos online, otros sistemas bibliotecarios).

**Resultados**

1. **Sistema Integral**
   * Se logró desarrollar un sistema completo de gestión bibliotecaria con todos los diagramas UML necesarios.
   * El sistema permite gestionar eficientemente libros, autores y préstamos.
   * Se implementaron todas las funcionalidades requeridas en los casos de uso.
2. **Documentación Completa**
   * Se generaron diagramas detallados que facilitan la comprensión del sistema.
   * La documentación incluye casos de uso, diagramas de secuencia, clases y estados.
   * Se proporcionaron explicaciones detalladas de cada componente.
3. **Base para Desarrollo**
   * Los diagramas y documentación proporcionan una base sólida para la implementación.
   * El diseño modular facilita el desarrollo por etapas.
   * La estructura permite escalar el sistema según necesidades futuras.

**Bibliografía:**

1. **PlantUML**

* **Sitio oficial:** [**https://plantuml.com/**](https://plantuml.com/)
* **Documentación:** [**https://plantuml.com/guide**](https://plantuml.com/guide)

1. **Draw.io**

* **Editor online:** [**https://app.diagrams.net/**](https://app.diagrams.net/)
* **Guías:** [**https://www.diagrams.net/doc/**](https://www.diagrams.net/doc/)

1. **Lucidchart**

* **Plataforma:** [**https://www.lucidchart.com/**](https://www.lucidchart.com/)
* **Recursos UML:** [**https://www.lucidchart.com/pages/uml-diagram-tool**](https://www.lucidchart.com/pages/uml-diagram-tool)