

EVALUACIÓN PARCIAL – RESOLUCIÓN DE CASOS

FACULTAD:	Tecnología Informática		
CARRERA:	Analista Programador (a Distancia)		
ALUMNO/A:	<b>GERARDO TORDOYA</b>		
SEDE:	Buenos Aires	LOCALIZACIÓN:	Distancia
ASIGNATURA:	Metodología de Desarrollo de Sistemas II		
CURSO:	T4-17-15 2-K-D	TURNO:	Distancia
PROFESOR:	Leonel Jiménez G.	FECHA:	27/05/2023
TIEMPO DE RESOLUCIÓN:	6 días	EXAMEN PARCIAL NRO.:	1
MODALIDAD DE RESOLUCIÓN:	Domiciliario – Individual.		
Resultados de aprendizaje:			
[Modela]+ [Casos de uso]+ [Para representar los requisitos funcionales] + [Utilizando el estándar UML]			
[Modela]+ [Diagramas de clases]+ [Para representar la estructura estática del sistema] + [Aplicando el estándar UML]			
[Modela]+ [Diagramas de interacción]+ [Para representar la interacción dinámica del sistema] + [Aplicando el estándar UML]			

Los estudiantes encontrará el examen habilitado en la plataforma ULTRA a partir de la fecha y hora indicada por el profesor.

Criterios de calificación: Para acreditar los saberes deberá obtener, al menos, el 60% de los aspectos conceptuales, además de, al menos, el 60% de los aspectos procedimentales. La calificación final se obtendrá luego de la defensa oral del trabajo presentado.

Criterios de evaluación: Se evaluará la claridad en el planteamiento de los aspectos conceptuales y procedimentales. La evaluación se hará a partir de la siguiente grilla:

Criterio	Calificación	Observaciones
Instancia oral		
Aspectos Conceptuales		
Pregunta 1		
Pregunta 2		
Pregunta 3		
Pregunta 4		
Pregunta 5		
Pregunta 6		
Pregunta 7		
Pregunta 8		
Aspectos procedimentales		
Pregunta 1		
Pregunta 2		
Pregunta 3		
Pregunta 4		
Pregunta 5		
Calificación final		

Forma de entrega del examen

- Se deberá resolver el examen sobre el mismo archivo del examen original.
- Se deberá subir al ULTRA el examen ya resuelto en un archivo tipo ZIP o RAR<sup>1</sup>.

<sup>1</sup> Profesor: usted indicó ayer (grupo de chat) que la entrega debía hacerse en DOC.

# ASPECTOS CONCEPTUALES

## 1. ¿Cuál es la diferencia entre análisis, diseño e implementación? [15 Puntos]

El **análisis** pone énfasis en la investigación del problema y los requisitos, en vez de ponerlo en la solución. Es decir, define el problema: En esta etapa se estudia y comprende el problema o la necesidad a resolver. Se recopila información, se identifican los requisitos y se analizan los procesos y las funcionalidades que el sistema debe cumplir. El objetivo es obtener una visión clara de lo que se espera del sistema y de las necesidades de los usuarios.

El **diseño** pone énfasis en una solución conceptual, que satisface los requisitos. Es decir, propone una solución al problema: En esta etapa se define la estructura y el funcionamiento del sistema (la solución informática) a partir de los requisitos establecidos en la etapa de análisis. Se determina cómo se organizarán los componentes del sistema, cómo interactuarán entre sí y cómo se presentará la interfaz de usuario.

La **implantación** es la traducción de la solución a un lenguaje de programación determinado. Es decir, construye la solución propuesta: En esta etapa se lleva a cabo la construcción del sistema de acuerdo con el diseño establecido. Se traducen las especificaciones y los diseños en código fuente utilizando un lenguaje de programación. También se realizan pruebas para verificar que el sistema funcione correctamente y se corrigen los errores encontrados. Finalmente, se documenta el código y se prepara el sistema para su despliegue y puesta en marcha.

## 2. ¿Qué tipos de requisitos conoce? Explique y Ejemplifique. [10 Puntos]

Distintos autores señalan distintos tipos de requisitos, y aunque hay divergencia, en general señalan los mismos que se mencionan los apuntes de cátedra:

**Requisitos de Usuarios.** Estos requisitos se centran en las necesidades y expectativas de los usuarios finales de la solución. Incluyen la funcionalidad que los usuarios desean tener, las tareas que deben realizar y las características que consideran importantes. Los requisitos de usuarios son recopilados a través de la interacción con los *stakeholders*<sup>2</sup> y suelen ser expresados en un lenguaje más comprensible para los usuarios finales.

**Requisitos de Sistema.** Los requisitos de sistema se refieren a las especificaciones generales del sistema en su conjunto. Es decir, los requisitos de sistema se refieren a las características y especificaciones que el sistema en su conjunto debe cumplir. Estos requisitos están relacionados con aspectos como la arquitectura del sistema, la infraestructura tecnológica, las interfaces con otros sistemas, los requisitos de rendimiento, la seguridad y la confiabilidad. Los requisitos de sistema definen cómo debe ser el sistema en términos generales para satisfacer las necesidades y expectativas de los usuarios y la organización.

**Requisitos Funcionales.** Los requisitos funcionales se centran en las funcionalidades específicas que el sistema debe proporcionar. Es decir, los requisitos funcionales se centran en las funcionalidades y comportamientos específicos que debe tener el sistema informático para cumplir con los objetivos y necesidades de los usuarios. Estos requisitos describen las acciones que el sistema debe ser capaz de realizar, cómo debe interactuar con los usuarios y cómo debe responder a las entradas y solicitudes. Los requisitos funcionales son más específicos y detallados, y se enfocan en las capacidades y características del sistema.

**Requisitos no funcionales.** Los requisitos no funcionales se centran en atributos específicos relacionados con la calidad y el rendimiento del sistema. Es decir, los requisitos no funcionales se centran en atributos y características específicas del sistema que no están relacionados directamente con las funcionalidades. Estos

---

<sup>2</sup> Las partes interesadas.

requisitos abordan aspectos como la usabilidad, la eficiencia, la escalabilidad, la interoperabilidad, la accesibilidad, la disponibilidad, la capacidad de respuesta y la mantenibilidad del sistema. Los requisitos no funcionales describen cómo debe ser el sistema en términos de calidad, rendimiento y experiencia del usuario.

\* \*\*

Es importante destacar que cada tipo de requisito puede contener subcategorías o detalles más específicos. Por ejemplo, los requisitos funcionales pueden desglosarse en requisitos de usuario y requisitos de sistema más detallados. Además, es fundamental tener en cuenta los requisitos legales, éticos y regulatorios que pueden ser aplicables al proyecto en cuestión.

3. Explique el concepto de escenario en el contexto de los casos de uso. Relacione este concepto con el uso de los diagramas de secuencia. [10 Puntos]

Presumo que se refiere a la especificación de casos de uso. Y teniendo en mente que un caso de uso es, ante todo, requisitos funcionales, entonces un escenario se refiere a una secuencia específica de acciones y eventos que ocurren durante la ejecución de un caso de uso.

Los casos de uso se utilizan para capturar los requisitos funcionales del sistema, describiendo las interacciones entre los actores (usuarios, sistemas externos) y el sistema en sí. Un caso de uso puede tener varios escenarios, y cada escenario representa una instancia particular de la ejecución del caso de uso. Cada escenario se compone de una secuencia de pasos y acciones que describen cómo interactúan los actores con el sistema para lograr un objetivo específico.

Los diagramas de secuencia, por otro lado, son diagramas UML que representan visualmente la interacción entre objetos en un sistema. Estos diagramas muestran la secuencia de mensajes que se intercambian entre los objetos a lo largo del tiempo para cumplir con un escenario o un caso de uso en particular.

La relación entre los escenarios y los diagramas de secuencia radica en que los escenarios proporcionan la base para crear los diagramas de secuencia. Los pasos y las acciones descritas en los escenarios se traducen en los mensajes intercambiados entre los objetos representados en el diagrama de secuencia. Así, un escenario **describe** la secuencia de acciones en lenguaje natural, mientras que el diagrama de secuencia **representa** (proporciona una representación visual) de esa secuencia utilizando objetos y mensajes. El diagrama de secuencia ayuda a visualizar el flujo de interacción entre los objetos involucrados en un caso de uso y proporciona una visión más detallada de cómo se lleva a cabo cada paso en el escenario.

4. ¿Cuál es la diferencia entre mensajes, operaciones y método? Ejemplifique [10 Puntos]

Los mensajes son solicitudes que se envían entre objetos, las operaciones son las acciones que los objetos pueden realizar y los métodos son las implementaciones específicas de esas operaciones. Los métodos están asociados con las clases y se ejecutan cuando se invocan las operaciones correspondientes a través de mensajes.

**Mensajes:** son solicitudes para que se lleve a cabo la operación indicada y se produzca el resultado. En un diagrama de secuencia los mensajes representan la comunicación o interacción entre los objetos participantes en un escenario o flujo de ejecución. Un mensaje en un diagrama de secuencia indica que un objeto envía un estímulo a otro objeto, solicitando que realice una acción o ejecute una operación específica.

**Operaciones:** son funciones o transformaciones que se aplican a un objeto de una clase. En un diagrama de clases las operaciones se representan en forma de métodos en las clases. Una operación es una acción o comportamiento definido en la interfaz de una clase y se representa mediante el nombre de la operación junto con los parámetros y el tipo de retorno si corresponde. Las operaciones indican las



funcionalidades disponibles en una clase, pero no especifican la implementación detallada de dichas funcionalidades.

**Métodos:** son las implementaciones de una operación. En un diagrama de clases los métodos se representan dentro de las clases como los bloques de construcción de la implementación concreta de una operación. Los métodos se indican mediante su nombre, parámetros, tipo de retorno y posiblemente otras características como la visibilidad y los modificadores. Los métodos proporcionan detalles sobre cómo se implementa una operación en una clase específica.

5. ¿Cuáles son las diferencias entre clase, clase abstracta e interfaz? [15 Puntos]

**Clase:** Una clase es una plantilla o molde para crear objetos. En ese sentido, puede considerarse como una estructura de datos que encapsula el estado y el comportamiento de los objetos y, por tanto, define las propiedades y comportamientos que tienen en común los objetos de ese tipo. Las clases se utilizan para crear instancias de objetos, es decir, son los planos a partir de los cuales se crean los objetos reales, es decir, las entidades (instancias) concretas e individuales que se crean a partir de ella, la clase.

**Clase abstracta:** Una clase abstracta es una clase que no puede ser instanciada directamente, sino que se utiliza como base para otras clases derivadas. Proporciona una abstracción común para un grupo de clases relacionadas. Una clase abstracta puede contener métodos con implementación y métodos abstractos. Los métodos abstractos son declarados en la clase abstracta pero no tienen una implementación definida. Las clases derivadas que heredan de una clase abstracta deben implementar los métodos abstractos definidos por esa clase. Las clases abstractas se utilizan para crear jerarquías de clases donde se comparten características comunes.

**Interfaz:** Una interfaz define un conjunto de métodos que una clase debe implementar. Es un contrato que una clase acepta cumplir al implementar los métodos definidos en la interfaz. Una interfaz proporciona una especificación de los métodos que una clase debe tener sin especificar su implementación. Una clase puede implementar múltiples interfaces, lo que permite la implementación de múltiples conjuntos de comportamientos. Las interfaces se utilizan para lograr la abstracción y el polimorfismo, permitiendo que los objetos se traten de manera genérica a través de interfaces comunes.

6. ¿Qué es el encapsulamiento? Explique cómo se implementa. Ejemplifique. [10 Puntos]

El encapsulamiento es uno de los principios fundamentales de la programación orientada a objetos. Consiste en ocultar los detalles internos de un objeto y exponer solo una interfaz que permita interactuar con él. El encapsulamiento combina datos y métodos relacionados en una sola entidad, lo que proporciona una capa de protección y facilita el mantenimiento y la modificación del código.

El encapsulamiento se implementa mediante el uso de modificadores de acceso en la definición de clases y miembros de clase. Estos modificadores de acceso, como "**public**", "**private**" y "**protected**", controlan la visibilidad de los miembros de una clase (es decir, quién puede acceder a ellos):

- **Public:** Un miembro público es accesible desde cualquier parte del programa. Puede ser accedido directamente tanto desde dentro de la clase como desde otras clases.
- **Private:** Un miembro privado es accesible solo desde dentro de la clase en la que se define. No se puede acceder a él directamente desde fuera de la clase.
- **Protected:** Un miembro protegido es accesible desde dentro de la clase en la que se define y desde sus subclases (clases que heredan de ella).

NOTA: Al definir una clase, se utilizan los modificadores de acceso para controlar la visibilidad de los atributos y métodos de la clase. Los atributos se suelen declarar como

privados y se accede a ellos mediante métodos públicos, conocidos como "métodos de acceso" (**getters** y **setters**), que proporcionan una interfaz controlada para leer y modificar los valores de los atributos.

Ejemplo de encapsulamiento:

```
public class CuentaBancaria {  
    private double saldo;  
  
    public void depositar(double cantidad) {  
        // Lógica para realizar el depósito  
        saldo += cantidad;  
    }  
  
    public void retirar(double cantidad) {  
        // Lógica para realizar el retiro  
        if (cantidad <= saldo) {  
            saldo -= cantidad;  
        } else {  
            System.out.println("Saldo insuficiente");  
        }  
    }  
  
    public double getSaldo() {  
        return saldo;  
    }  
}
```

En este ejemplo, la clase **CuentaBancaria** encapsula el saldo como un atributo privado. Los métodos depositar y retirar proporcionan la interfaz pública para interactuar con el objeto y modificar el saldo de manera controlada. El método **getSaldo** es un método de acceso que permite obtener el valor del saldo, pero no permite modificarlo directamente. De esta manera, el encapsulamiento asegura que el saldo solo pueda ser modificado a través de métodos específicos y protege la integridad de los datos de la clase.

7. Explique y relacione los siguientes conceptos: Herencia, Generalización, Especialización. [15 Puntos]

Estos tres conceptos tienen un denominador común: la relación "es-un". La herencia permite que una clase herede características y comportamientos de otra clase. La generalización implica la creación de una clase más general a partir de clases más específicas. La especialización consiste en crear clases más específicas a partir de una clase base más general.

**Herencia:** La herencia es un mecanismo mediante el cual una clase (llamada clase derivada o subclase) puede heredar características y comportamientos de otra clase (llamada clase base o superclase). La clase derivada tiene acceso a los atributos y métodos de la clase base y puede extenderlos o modificarlos según sea necesario. La herencia permite la reutilización de código y establece una relación de "es-un" entre las clases.

**Generalización:** La generalización es un concepto relacionado con la herencia y se refiere a la creación de una clase más general a partir de clases más específicas. En la generalización, se identifican características y comportamientos comunes en varias clases y se los extrae para formar una clase base más abstracta. Las clases específicas se convierten en subclases de la clase generalizada, heredando sus propiedades y comportamientos comunes. La generalización ayuda a establecer jerarquías de clases y a capturar la relación de "es-un" entre ellas.

**Especialización:** La especialización es el proceso opuesto a la generalización. Consiste en crear clases más específicas a partir de una clase base más general. Las clases especializadas (subclases) heredan las características y comportamientos de la clase base, pero también pueden tener sus propias características adicionales o modificar los comportamientos heredados. La especialización permite crear clases más concretas y específicas que amplían o adaptan la funcionalidad de la clase base.

8. ¿Qué es el polimorfismo? Explique y ejemplifique. [15 Puntos]

El polimorfismo es un principio de la programación orientada a objetos que permite tratar objetos de diferentes clases de manera genérica. Se basa en la capacidad de las clases derivadas de una clase base de proponer su propia implementación de los métodos heredados de la clase base. Esto significa que un método de la clase base puede tener diferentes comportamientos en las diferentes clases derivadas.

Un ejemplo común de polimorfismo es el uso de un método común, como "dibujar()", en diferentes clases que representan formas geométricas. Cada clase derivada de la clase base (por ejemplo, "Círculo", "Cuadrado" y "Triángulo") puede implementar su propia versión del método "dibujar()" de acuerdo con su forma específica:

```
public abstract class Forma {
    public abstract void dibujar();
}

public class Circulo extends Forma {
    @Override
    public void dibujar() {
        System.out.println("Dibujando un círculo");
    }
}

public class Cuadrado extends Forma {
    @Override
    public void dibujar() {
        System.out.println("Dibujando un cuadrado");
    }
}

public class Triangulo extends Forma {
    @Override
    public void dibujar() {
        System.out.println("Dibujando un triángulo");
    }
}

public class Main {
    public static void main(String[] args) {
        Forma forma1 = new Circulo();
        Forma forma2 = new Cuadrado();
        Forma forma3 = new Triangulo();

        forma1.dibujar(); // Salida: Dibujando un círculo
        forma2.dibujar(); // Salida: Dibujando un cuadrado
        forma3.dibujar(); // Salida: Dibujando un triángulo
    }
}
```

En este ejemplo, la clase abstracta **Forma** define el método abstracto `dibujar()`. Las clases derivadas **Circulo**, **Cuadrado** y **Triangulo** implementan este método de

manera específica para dibujar cada forma en particular. Al tratar los objetos de estas clases como objetos de tipo **Forma**, podemos llamar al método **dibujar()** de manera genérica y el polimorfismo se encargará de ejecutar la implementación específica en función del tipo de objeto en tiempo de ejecución.



# ASPECTOS PROCEDIMENTALES

Del siguiente enunciado que se indica al final, determinar:

1. Los Requerimientos Funcionales planteados. [10 Puntos]
2. Modelo de casos de uso de todo el sistema. [10 Puntos]
3. Especificar el Caso de Uso (Planilla de Especificación del Caso de Uso) más relevante. El CU principal desde el punto de vista de su análisis. [20 Puntos]
4. Elaborar el diagrama de secuencia para el caso de uso del punto anterior. [30 Puntos]
5. Elaborar el diagrama de clases de toda la solución. [30 Puntos]

Usted puede sumar los Atributos de Dato que considere sean necesarios para robustecer su diseño. También puede asumir alguna otra especificación y/o funcionalidad que considere conveniente.

## ENUNCIADO (BIBLIOTECA)

Una biblioteca contiene ejemplares de libros y películas en VHS. Cada ejemplar tiene un código que es distinto para todos y una descripción de su estado; además puede ser prestado a los socios de la misma. Los libros tienen un título, **n** autores, una editorial, un número de edición y una fecha de publicación. De las películas se conoce la editorial, la fecha de publicación y nombre y el tiempo en minutos. **En cambios<sup>3</sup>** están clasificados por temas (ciencia-ficción, historia, etc.). Además, existe un conjunto de socios de los que se guarda un número de socio, nombre, primer y segundo apellido, dirección, localidad, provincia y número de teléfono. Los socios pueden recibir sanciones económicas por un motivo dado, en una determinada fecha y con un determinado monto. Los socios pueden obtener préstamos de la biblioteca. Cada préstamo tiene como entrada la fecha del día como límite por defecto 10 días a contar a partir de esa fecha. Se mantendrá una lista de ejemplares que los socios piden pero que bien todavía no han sido adquiridos, o bien no hay suficientes ejemplares para satisfacer la demanda (ejemplares agotados) en la que se guardará una referencia al título demandado y la fecha en que se pidió. Aquellas referencias correspondientes a títulos de los que todavía no se han adquirido ejemplares, también serán apuntadas, por lo que se deberá de distinguir entre aquéllos de los que, si existen ejemplares, de los que no existen. Existe la posibilidad de hacer reservas de ejemplares que estén siendo utilizados por otros socios, a tal efecto se le facilitará al socio que va a hacer la reserva la fecha más próxima en la que se va a devolver un ejemplar del título deseado, quedando el ejemplar bloqueado para préstamo durante dos días, a la espera de que lo vaya a retirar el socio que lo ha reservado.

## Entidades principales del escenario:

1. BIBLIOTECA
  - Contiene EJEMPLARES de LIBROS y PELÍCULAS en VHS
2. EJEMPLAR
  - Código (distinto para cada EJEMPLAR)
  - Descripción de estado
  - Puede ser prestado a los SOCIOS
3. LIBRO
  - Título
  - Autores (n autores)

---

<sup>3</sup> Se usará la frase "los ejemplares" pues "en cambios" no tiene sentido.

- Editorial
- Número de edición
- Fecha de publicación
- 4. PELÍCULA
  - Editorial
  - Fecha de publicación
  - Nombre
  - Duración en minutos
- 5. TEMA
  - Clasificación de EJEMPLARES por temas (ciencia-ficción, historia, etc.)
- 6. SOCIO
  - Número de SOCIO
  - Nombre
  - Primer y segundo apellido
  - Dirección
  - Localidad
  - Provincia
  - Número de teléfono
- 7. SANCIONES
  - Motivo
  - Fecha
  - Monto económico
- 8. PRÉSTAMO
  - Fecha de préstamo (con límite por defecto de 10 días)
  - Relación entre SOCIO y EJEMPLAR
- 9. LISTA DE DEMANDA
  - Almacenamiento de EJEMPLARES solicitados pero no adquiridos o agotados
  - Referencia al título demandado y fecha de solicitud
- 10. RESERVA
  - Posibilidad de reservar EJEMPLARES en uso por otros SOCIOS
  - Fecha más próxima de devolución
  - Bloqueo del EJEMPLAR para préstamo durante dos días

## REQUERIMIENTOS FUNCIONALES

- 1. Gestión de ejemplares:**
  - Registrar ejemplares de libros y películas en VHS.
  - Asignar un código único a cada ejemplar.
  - Mantener una descripción del estado de cada ejemplar.
  - Permitir el préstamo de ejemplares a los socios.
  - Clasificar ejemplares por temas (ciencia-ficción, historia, etc.).
- 2. Gestión de libros:**
  - Registrar información de libros, como título, autores, editorial, número de edición y fecha de publicación.
- 3. Gestión de películas:**
  - Registrar información de películas, como editorial, fecha de publicación, nombre y duración en minutos.
- 4. Gestión de socios:**
  - Mantener un registro de socios con información como número de socio, nombre, apellidos, dirección, localidad, provincia y número de teléfono.
  - Aplicar sanciones económicas a los socios por motivos específicos, con una fecha y un monto determinados.
  - Permitir que los socios realicen préstamos de la biblioteca.
- 5. Gestión de préstamos:**
  - Registrar préstamos realizados por los socios.
  - Establecer una fecha límite de devolución por defecto de 10 días a partir de la fecha del préstamo.



- Mantener una lista de ejemplares solicitados pero aún no adquiridos o agotados.
- Registrar la referencia al título demandado y la fecha de solicitud.

#### **6. Gestión de reservas:**

- Permitir reservas de ejemplares que están siendo utilizados por otros socios.
- Asignar una fecha próxima para la devolución del ejemplar reservado.
- Bloquear el ejemplar para préstamo durante dos días, esperando que el socio que lo ha reservado lo retire.

### **CASOS DE USO DE TODO EL SISTEMA**

#### **1. Registrar ejemplar:**

- Descripción: Este caso de uso permite al personal de la biblioteca registrar un nuevo ejemplar de libro o película en el sistema.
- Actores principales: Personal de la biblioteca.
- Flujo básico:
  - El personal de la biblioteca ingresa los detalles del ejemplar, como el código, descripción del estado y tema.
  - Si el ejemplar es un libro, se ingresan el título, autores, editorial, número de edición y fecha de publicación. Si es una película, se ingresan la editorial, fecha de publicación, nombre y tiempo en minutos.
  - El sistema registra el ejemplar en la base de datos de la biblioteca.

#### **2. Registrar socio:**

- Descripción: Este caso de uso permite al personal de la biblioteca registrar a un nuevo socio en el sistema.
- Actores principales: Personal de la biblioteca.
- Flujo básico:
  - El personal de la biblioteca recopila la información del socio, como el número de socio, nombre, apellidos, dirección, localidad, provincia y número de teléfono.
  - El sistema registra al socio en la base de datos de la biblioteca.

#### **3. Realizar préstamo:**

- Descripción: Este caso de uso permite al personal de la biblioteca realizar un préstamo de un ejemplar a un socio.
- Actores principales: Personal de la biblioteca.
- Flujo básico:
  - El personal de la biblioteca busca al socio en el sistema utilizando su número de socio.
  - El personal de la biblioteca busca el ejemplar disponible para préstamo en la base de datos.
  - El personal de la biblioteca registra el préstamo, estableciendo la fecha de inicio y la fecha de vencimiento (10 días después de la fecha de inicio por defecto) en el sistema.
  - El sistema actualiza el estado del ejemplar como prestado y asocia el préstamo al socio.

#### **4. Realizar reserva:**

- Descripción: Este caso de uso permite a un socio realizar una reserva de un ejemplar que está siendo utilizado por otro socio.
- Actores principales: Socios de la biblioteca.
- Flujo básico:
  - El socio busca el título deseado en el sistema.
  - Si no hay ejemplares disponibles, el socio selecciona la opción de reserva.
  - El sistema muestra al socio la fecha más próxima en la que se espera que el ejemplar esté disponible.



- El socio confirma la reserva.
- El sistema bloquea el ejemplar para préstamo durante dos días, a la espera de que el socio que lo ha reservado lo retire.

## CASO DE USO MÁS RELEVANTE

Dentro del texto proporcionado, el caso de uso más relevante es **“Realizar Préstamo”**. Este caso de uso es fundamental para el funcionamiento del sistema, ya que permite a los socios de la biblioteca solicitar y obtener préstamos de ejemplares. Al realizar un préstamo, se establece una relación entre un socio y un ejemplar específico, y se registra la fecha de préstamo y la fecha límite de devolución.

Este caso de uso implica varias acciones importantes, como verificar la disponibilidad del ejemplar, actualizar el estado del ejemplar a "prestado", registrar la información del préstamo y establecer una fecha límite para su devolución. Además, el caso de uso podría involucrar validaciones adicionales, como verificar si el socio tiene alguna sanción económica pendiente o si ha alcanzado el límite máximo de préstamos permitidos.

## Especificación del Caso de Uso

<b>CU-03</b>	<b>Realizar Préstamo</b>	
<b>Versión</b>	1.0 (2023-05-28)	
<b>Estado</b>	Propuesto	
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>RF-01 Gestión de ejemplares</li> <li>RF-02 Gestión de libros</li> <li>RF-03 Gestión de películas</li> <li>RF-04 Gestión de socios</li> </ul>	
<b>Actores Principales</b>	<ul style="list-style-type: none"> <li>Personal de la Biblioteca</li> </ul>	
<b>Actores Secundarios</b>	<ul style="list-style-type: none"> <li>---</li> </ul>	
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El ejemplar y el socio deben estar registrados en el sistema.</li> <li>El ejemplar debe estar disponible para préstamo.</li> </ul>	
<b>Descripción</b>	El sistema deberá comportarse como se describe en el este caso de uso cuando el socio solicite al bibliotecario sacar ejemplares en préstamo.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El socio solicita un préstamo a lo cual el Personal de la biblioteca selecciona la opción "Realizar préstamo" en el sistema.
	2	El sistema muestra un formulario para buscar al socio.
	3	El Personal de la biblioteca ingresa el número de socio del socio que desea realizar el préstamo.
	4	El sistema busca al socio en la base de datos y muestra su información.
	5	El Personal de la biblioteca selecciona la opción para buscar el ejemplar disponible.
	6	El sistema muestra un formulario para buscar el ejemplar.
	7	El Personal de la biblioteca ingresa el código del ejemplar que desea prestar.

	8	El sistema busca el ejemplar en la base de datos y verifica si está disponible.	
	9	Si el ejemplar está disponible:	
		9.a	El Personal de la biblioteca registra el préstamo, ingresando la fecha de inicio y se establece la fecha de vencimiento (10 días después de la fecha de inicio por defecto) en el sistema.
		9.b	El sistema actualiza el estado del ejemplar como prestado y asocia el préstamo al socio.
		9.c	El sistema muestra un mensaje de éxito y el comprobante del préstamo.
	10	Si el ejemplar no está disponible:	
		10.a	El sistema muestra un mensaje de error indicando que el ejemplar no está disponible para préstamo.
11	Fin del caso de uso.		
Postcondiciones	<ul style="list-style-type: none"><li>El socio se lleva los ejemplares prestados.</li><li>El sistema ha registrado el préstamo de los mismos.</li></ul>		
Excepciones <sup>4</sup>	Paso	Acción	
	4	Si durante el proceso de realización del préstamo se detecta que el socio tiene registrada una penalización (sanción) vigente, se genera una excepción de socio con penalización. Esto significa que el socio no cumple con los requisitos para realizar un préstamo debido a una penalización pendiente.	
		4.1	El sistema muestra un mensaje de error al personal de la biblioteca, indicando que el socio tiene una penalización vigente y no puede realizar un préstamo en ese momento.
		4.2	El personal de la biblioteca debe informar al socio acerca de su penalización y las restricciones de préstamo asociadas.
		4.3	El flujo normal del caso de uso se interrumpe y se impide la continuación del proceso de préstamo.
Comentarios	El número máximo de préstamos simultáneos, la duración de los préstamos, y el importe de las sanciones depende de la política de la biblioteca y puede cambiar en el futuro. Ver las reglas de negocio RN-XX y RN-XX (no presentes en este trabajo).		

DIAGRAMA DE SECUENCIA PARA EL CASO DE USO “REALIZAR PRÉSTAMO”

Secuencia: Realizar préstamo

Actores: Personal de la biblioteca, Socio

Precondiciones: El ejemplar y el socio deben estar registrados en el sistema y el ejemplar debe estar disponible para préstamo.

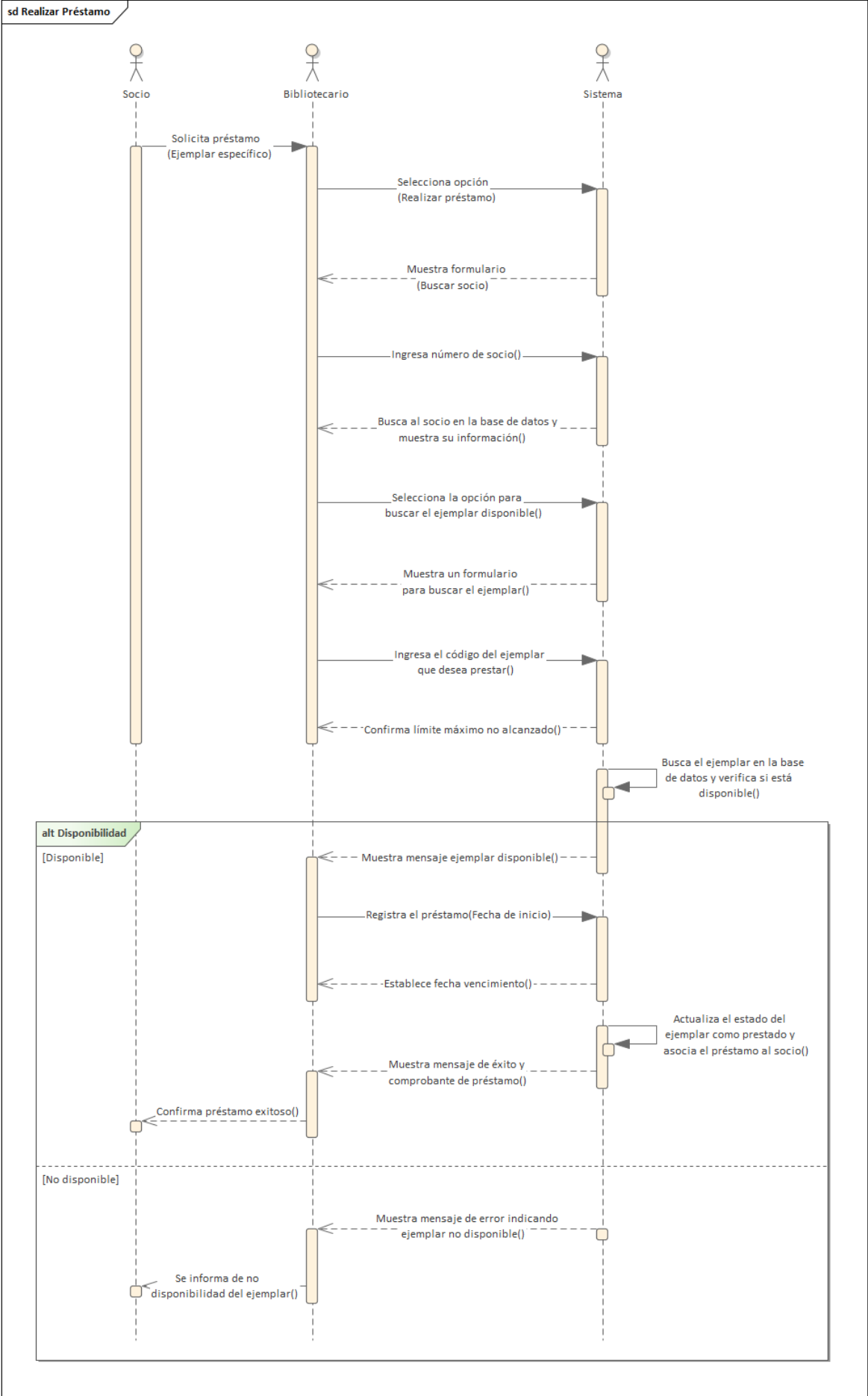
Flujo de eventos:

- El Personal de la biblioteca selecciona la opción de "Realizar préstamo" en el sistema.
  - El sistema muestra un formulario para buscar al socio.
  - El Personal de la biblioteca ingresa el número de socio del socio que desea realizar el préstamo.
  - El sistema busca al socio en la base de datos y muestra su información.

<sup>4</sup> A veces llamado “Flujo/s Alternativo/s”.

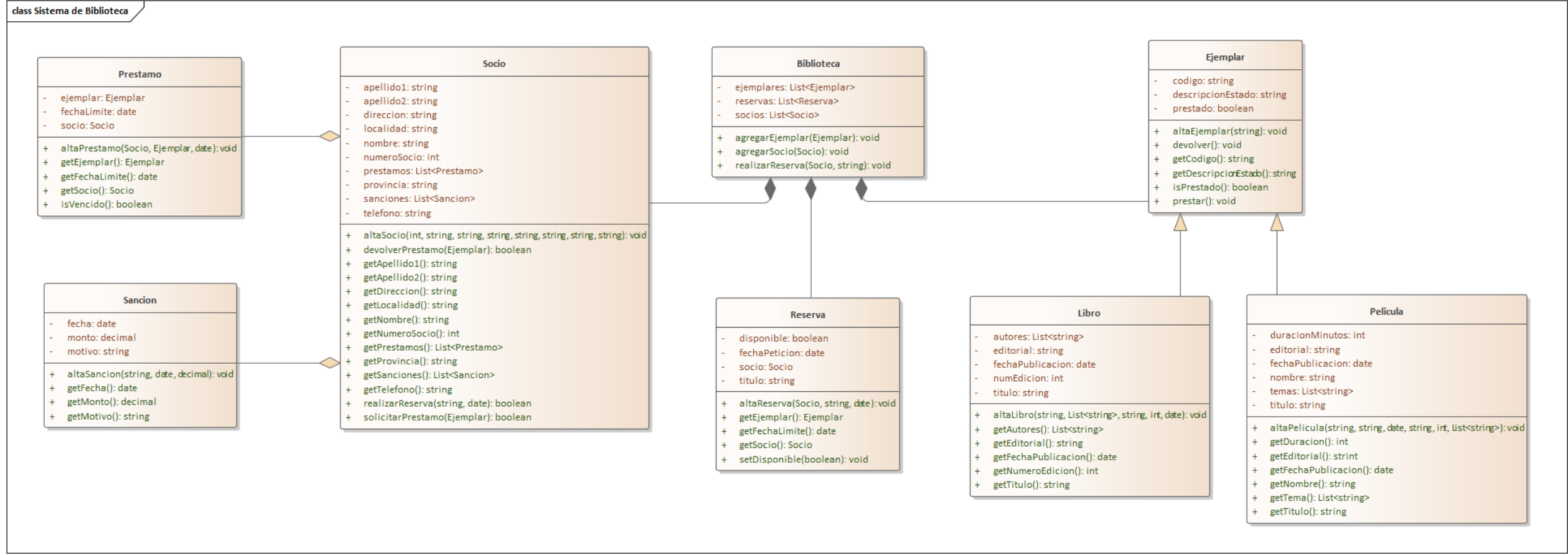


5. El Personal de la biblioteca selecciona la opción para buscar el ejemplar disponible.
6. El sistema muestra un formulario para buscar el ejemplar.
7. El Personal de la biblioteca ingresa el código del ejemplar que desea prestar.
8. El sistema busca el ejemplar en la base de datos y verifica si está disponible.
9. Si el ejemplar está disponible:
  - a. El Personal de la biblioteca registra el préstamo, ingresando la fecha de inicio y estableciendo la fecha de vencimiento (10 días después de la fecha de inicio por defecto).
  - b. El sistema actualiza el estado del ejemplar como prestado y asocia el préstamo al socio.
  - c. El sistema muestra un mensaje de éxito y el recibo del préstamo.
10. Si el ejemplar no está disponible:
  - a. El sistema muestra un mensaje de error indicando que el ejemplar no está disponible para préstamo.
11. Fin de la secuencia.



## DIAGRAMA DE CLASES DE TODA LA SOLUCIÓN

Este diagrama de clases representa las entidades principales del escenario: Biblioteca, Ejemplar, Libro, Película, Socio, Sanción, Préstamo y Reserva. Las relaciones entre las clases se definen mediante atributos y métodos para representar la interacción y las características de cada entidad en el sistema de la biblioteca.





Ésta es una versión más simplificada del diagrama (para mejor visibilidad)

