

# Programación Orientada a Objetos

## Semana 4 – Diagrama de Clases

### UML

### PORTADA

**Nombre del alumno: Cesar Zaid Reynoso Martinez**

**Materia: Programación Orientada a Objetos**

**Semana: 4**

**Fecha: 19/02/2026**

**Sistema elegido: Sistema de Biblioteca**

## **FASE 1 – ANÁLISIS DEL SISTEMA**

### Nombre del sistema:

Sistema de Biblioteca

### **Descripción general**

El sistema de biblioteca es una aplicación que permite administrar los libros, usuarios y préstamos dentro de una biblioteca escolar o pública. Su función principal es organizar la información de los libros disponibles, registrar a los usuarios y controlar los préstamos y devoluciones.

El sistema debe permitir saber qué libros están disponibles, quién los tiene prestados y cuándo deben devolverse. También debe almacenar información importante como título, autor, categoría y estado del libro.

Este sistema ayuda a mantener el orden, evitar pérdidas y facilitar la gestión de materiales dentro de la biblioteca.

### Entidades principales

Libro

Usuario

Préstamo

Biblioteca

### Información que maneja cada entidad

Libro

ID

Título

Autor

Categoría

Estado (Disponible / Prestado)

Usuario

ID

Nombre

Correo

Teléfono

Préstamo

ID

Fecha de préstamo

Fecha de devolución

Estado

Biblioteca

Nombre

Dirección

Lista de libros

#### **Funciones principales del sistema**

Registrar usuarios

Registrar libros

Realizar préstamos

Registrar devoluciones

Consultar disponibilidad de libros

#### **Lista preliminar de clases y justificación**

Libro → Representa cada libro físico de la biblioteca.

Usuario → Representa a las personas que pueden pedir libros prestados.

Prestamo → Representa la acción de prestar un libro a un usuario.

Biblioteca → Representa la institución que administra todo el sistema.

Se consideran clases porque cada una representa una entidad real del sistema con atributos y comportamientos propios.

## **FASE 2 – IDENTIFICACIÓN Y DEFINICIÓN DE CLASES**

## Clase: Libro

### Atributos

id : int

titulo : String

autor : String

categoria : String

estado : String

### Métodos

consultarDisponibilidad() : boolean

cambiarEstado(nuevoEstado : String) : void

### Explicación

La clase Libro almacena la información básica de cada libro y permite consultar si está disponible o cambiar su estado.

## Clase: Usuario

### Atributos

id : int

nombre : String

correo : String

telefono : String

### Métodos

registrarse() : void

solicitarPrestamo(libro : Libro) : boolean

### Explicación

La clase Usuario representa a las personas que pueden solicitar libros y registrarse en el sistema.

### Clase: Prestamo

### Atributos

id : int

fechaPrestamo : Date

fechaDevolucion : Date

estado : String

### **Métodos**

generarPrestamo(usuario : Usuario, libro : Libro) : boolean

registrarDevolucion() : void

### **Explicación**

La clase Prestamo controla la relación entre un usuario y un libro cuando se realiza un préstamo.

Clase: Biblioteca

Atributos

nombre : String

direccion : String

listaLibros : List<Libro>

### **Métodos**

agregarLibro(libro : Libro) : void

eliminarLibro(id : int) : boolean

### **Explicación**

La clase Biblioteca administra los libros disponibles dentro del sistema.

## **FASE 3 – RELACIONES ENTRE CLASES**

### **1 Asociación**

Usuario — Prestamo

Un Usuario puede tener varios préstamos.

Multiplicidad:

Usuario (1) ----- (0..\*) Prestamo

Un préstamo pertenece a un solo usuario.

### **2 Asociación**

Libro — Prestamo

Un Libro puede estar en varios préstamos a lo largo del tiempo.

Libro (1) ----- (0..\*) Prestamo

### **3 Agregación**

Biblioteca — Libro

Una biblioteca contiene muchos libros.

Biblioteca (1) ----- (1..\*) Libro

Es agregación porque los libros pueden existir fuera de la biblioteca.

