

## Ejercicio sobre Bases de Datos Orientadas a Objetos

Supongamos que estás diseñando una base de datos orientada a objetos para una aplicación de gestión de una biblioteca. Debes modelar las entidades y las relaciones entre ellas utilizando los principios de las bases de datos orientadas a objetos. Utiliza un lenguaje de modelado de bases de datos orientadas a objetos, como ODMG (Object Database Management Group) o algún otro de tu elección.

1. Define las clases de objetos para representar los siguientes conceptos: Libro, Autor, Editorial y Prestamo. Cada libro tiene un autor y pertenece a una editorial. Cada préstamo está asociado a un libro y a un usuario.
2. Indica las relaciones entre las clases. Por ejemplo, un Libro puede tener una relación con un Autor y una relación con una Editorial.
3. Especifica los atributos relevantes para cada clase. Por ejemplo, un Libro podría tener atributos como título, ISBN, y añoPublicacion.
4. Agrega métodos a las clases si es necesario. Por ejemplo, un método en la clase Prestamo podría ser calcularDuracion.
5. Proporciona un ejemplo de instancias de estas clases, mostrando cómo podrían relacionarse entre sí en un escenario concreto.

Recuerda seguir los principios de la orientación a objetos, como la encapsulación, la herencia y el polimorfismo, al diseñar tu modelo de datos orientado a objetos.

El objetivo de esta actividad sobre bases de datos orientadas a objetos es diseñar y modelar una base de datos para una aplicación de gestión de una biblioteca utilizando los principios de la orientación a objetos. Aquí se detallan los objetivos específicos:

1. Definir Clases de Objetos:
  - Crear clases de objetos para representar conceptos específicos de la aplicación de gestión de la biblioteca, como Libro, Autor, Editorial y Prestamo. Cada clase debe encapsular sus propios datos y comportamientos.
2. Establecer Relaciones entre Clases:
  - Identificar y definir las relaciones entre las clases. Por ejemplo, establecer que un Libro tiene una relación con un Autor y una relación con una Editorial. Estas relaciones deben reflejar la asociación natural entre las entidades en el contexto de la biblioteca.
3. Especificar Atributos Relevantes:
  - Definir los atributos relevantes para cada clase. Por ejemplo, para la clase Libro, podrían incluirse atributos como título, ISBN y año de publicación. Cada atributo debe representar una característica significativa de la entidad.
4. Agregar Métodos a las Clases:

- Incluir métodos en las clases si es necesario para encapsular el comportamiento relacionado con la entidad. Por ejemplo, la clase Prestamo podría tener un método calcularDuracion para determinar la duración del préstamo.
5. Aplicar Principios de la Orientación a Objetos:
- Utilizar principios fundamentales de la orientación a objetos, como encapsulación, herencia y polimorfismo, en el diseño del modelo de datos. Garantizar que las clases sean cohesivas, tengan una interfaz clara y sigan buenas prácticas de diseño orientado a objetos.
6. Proporcionar Ejemplos de Instancias:
- Presentar ejemplos concretos de instancias de estas clases, demostrando cómo podrían relacionarse entre sí en un escenario específico de la aplicación. Esto ayuda a visualizar cómo se utilizarán y conectarán las clases en situaciones reales.