

Taller De Programación Orientada a Objetos (POO)

Alejandro Ducuara Satizabal

Ficha: 3192191

Docente: Mairon Salazar

1. ¿Que es la programación orientada a objetos (POO) y cuáles son sus ventajas frente a la programación estructurada?

R// La programación orientada a objetos es un modelo que organiza el código a travez de objetos pensando el problema como una relación de cosas, en este modelo se juntan datos (atributos) y comportamientos(métodos) en clases para separar el código en piezas simples y reutilizables. Este modelo es mejor que la programación estructurada ya que nos ofrece más orden al realizar un programa ya que pensamos en relaciones y no en solo funcionamiento, ayudándonos mucho en programas grandes

2. ¿Que es una clase? ¿Qué es un objeto? Explica con tus propias palabras y da un ejemplo

R// Una clase es el elemento principal, como el título de una tabla, de dónde se sacan las características(atributos) y sus funciones (métodos) un ejemplo de esto sería un carro con sus atributos como color, marca, tipo, etc. y sus funciones como acelerar, avanzar, retroceder y demás. Un objeto es como la representación en la realidad de la clase, es como los registros en una tabla, por ejemplo, continuando con el carro, el objeto sería el carro1 de color rojo, marca Mercedes, de tipo convertible, que puede acelerar, avanzar y retroceder.

3. Define los conceptos atributos, método y constructor en POO

R// Un atributo son las características de la clase a la que pertenece y almacenan los datos de los registros que se ingresen. Los métodos son las funciones o acciones que pueden realizar la clase, definiendo el comportamiento de los objetos interactuando con otros objetos o realizando tareas. El constructor es un método que se activa automáticamente cuando se crea un objeto de una clase, con el propósito de inicializar los atributos del objeto, estableciendo valores iniciales para que el objeto esté listo para ser usado

4. Explica que significa encapsulación, herencia, polimorfismo y abstracción

R// La encapsulación es hacer que los atributos o métodos de una clase nos se puedan acceder ni modificar desde afuera de la clase, solo el mismo objeto puede acceder a ellos. La herencia es una forma en que una clase puede heredar los atributos y métodos de otra, definiéndose nuevas clases basadas en unas clases ya existentes reutilizando código y generando una jerarquía. El polimorfismo es la capacidad de un objeto de tomar más formas, significando que dos clases distintas se les puede acceder utilizando el mismo interfaz y mostrar un comportamiento diferente según como se les acceda.

5. ¿Que es UML y para qué sirven los diagramas de clases, de objetos, de secuencia y de actividad?

R// El lenguaje unificado moderado (UML) es un lenguaje de modelado visual que usa diagramas para ver y construir sistemas, representando aspectos estáticos o estructurales del sistema.

Diagrama de clases: Representa la estructura de un sistema mostrando sus clases, atributos, métodos y las relaciones entre ellos. Ayuda a comprender como está organizado el sistema y como interactúan sus componentes.

Diagrama de objetos: Muestra la relación entre objetos por medio de ejemplos del mundo real e ilustra como se verá un sistema en un momento dado.

Diagrama de secuencia: Muestra como los objetos interactúan entre sí y el orden de la ocurrencia, representan interacciones para un escenario concreto.

Diagrama de actividad: Es una representación de las acciones, restricciones y requisitos que intervienen en la realización de las tareas, mostrando cada paso y punto de decisión en un proceso.

6. ¿Cuáles son las diferencias principales al implementar POO en Python, Java y C++?

R// Las diferencias se centran sobre todo en el enfoque. Python es un lenguaje multiparadigma lo que quiere decir que soporta POO y la programación funcional, adaptándose a diversas tareas, aunque no exigen que todo esté en una clase. En java se enfoca a objetos principalmente definiendo sus variables y funciones en clases, siendo un lenguaje más rígido para POO. En C++ también se centra en el POO, Pero no es tan estricto como Java siendo más flexible entre POO y el lenguaje funcional.

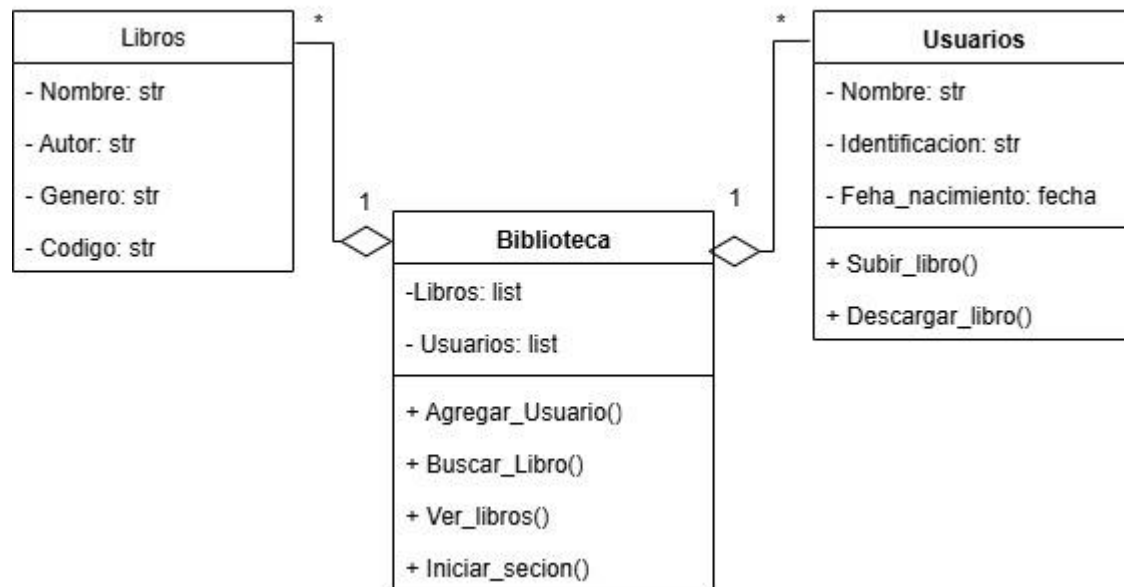
Parte 2

7. Elige un mini proyecto (ejemplo: biblioteca, gestión de cursos o tienda online) y responde: ¿Que clases identificas en el problema? ¿Qué atributos y métodos tendría cada clase? ¿Qué relaciones existen entre ellas?

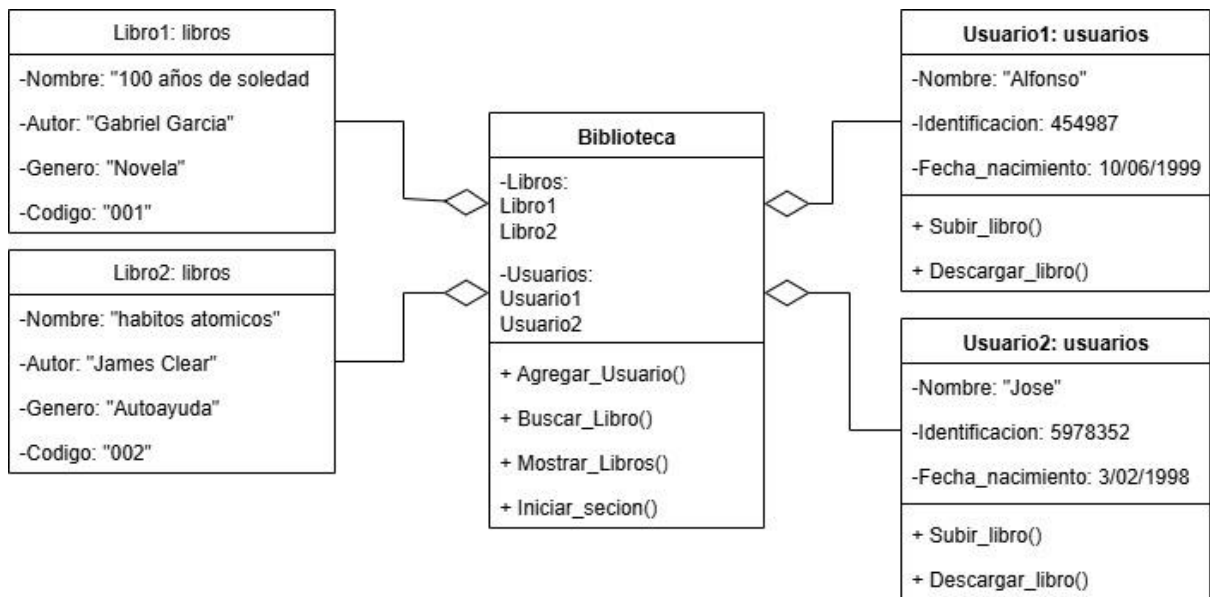
R// Elegí el mini proyecto de una biblioteca virtual, dónde se pueden subir libros, descargar libros, registrarse, iniciar sesión, buscar libros por género, autor, nombre y código y mostrar la información de los libros. Identifiqué 3 clases, la primera son los usuarios que tienen de atributos el nombre, identificación y la fecha de nacimiento, de métodos tienen subir libros y descargarlos. luego está la clase libros que tienen un autor, el género, un código y un nombre, esta clase se utiliza para guardar la información de los libros sin tener métodos. Por último, está la clase biblioteca que es

una lista de los libros y los usuarios, tienen la función de permitir registrar e iniciar sesión, permite buscar los libros por nombre, autor, género o por código, además de mostrar la información de los libros. Las relaciones que vi fue que la clase biblioteca contiene las otras dos, usuarios y libros

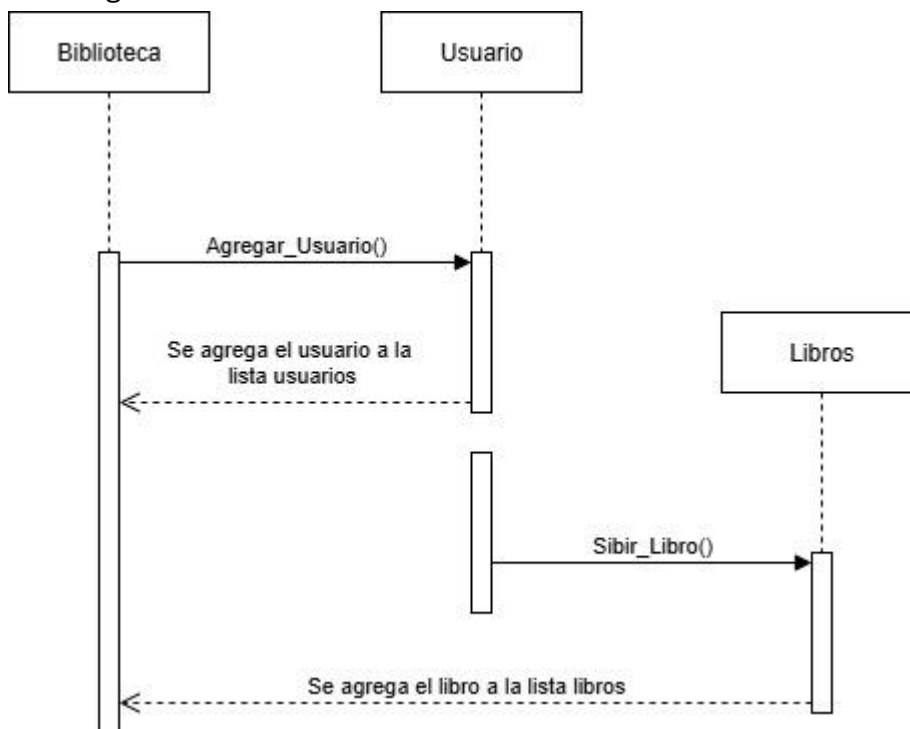
8. Diagrama de clases



9. Diagrama de objetos



10. Diagrama de secuencia



11. Diagrama de actividad

