PRIMER REPASO INGENIERIA DE SOFTWARE

JUAN SEBASTIAN RODRIGUEZ BLANCO

Universidad De Cundinamarca, Extensión Chía

Facultad De Ingeniería

Ingeniería de Sistemas y Computación Arquitectura de Computadores

William Matallana

Chía, Cundinamarca

24 de octubre del 2025

INTRODUCCION

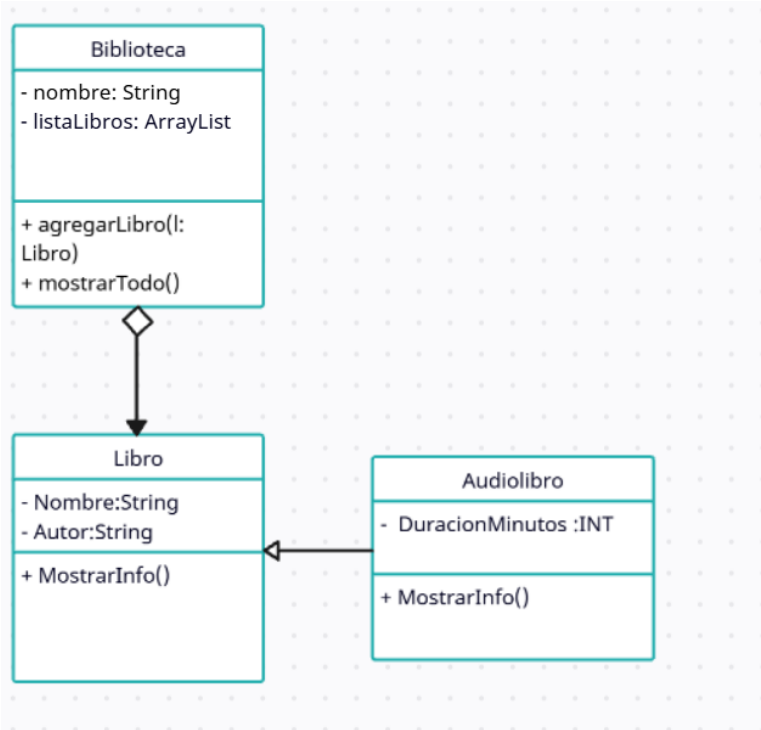
La Programación Orientada a Objetos (POO) es un paradigma diseñado para estructurar el software mediante clases, que actúan como plantillas, y objetos, que son las instancias reales de esas plantillas. Su potencia reside en la interacción de sus pilares: los atributos definen el estado, mientras que los métodos determinan el comportamiento; la herencia permite que nuevas clases hereden y extiendan funcionalidades de clases existentes para reutilizar código; el encapsulamiento protege la integridad de los datos mediante modificadores de acceso; y el polimorfismo otorga la flexibilidad de que una misma acción se ejecute de formas distintas según el objeto, logrando así sistemas escalables y fáciles de mantener.

EJERCICIO

Desarrollaremos un software diseñado para gestionar el inventario de una biblioteca de forma eficiente. El sistema se basa en una estructura de herencia donde definimos una clase padre llamada Libro, la cual centraliza los atributos comunes como títulos y autores. A partir de este molde, creamos la clase Audiolibro, que hereda automáticamente la información de nombre y autor, permitiéndonos añadir sus características propias (como duración) sin repetir código.

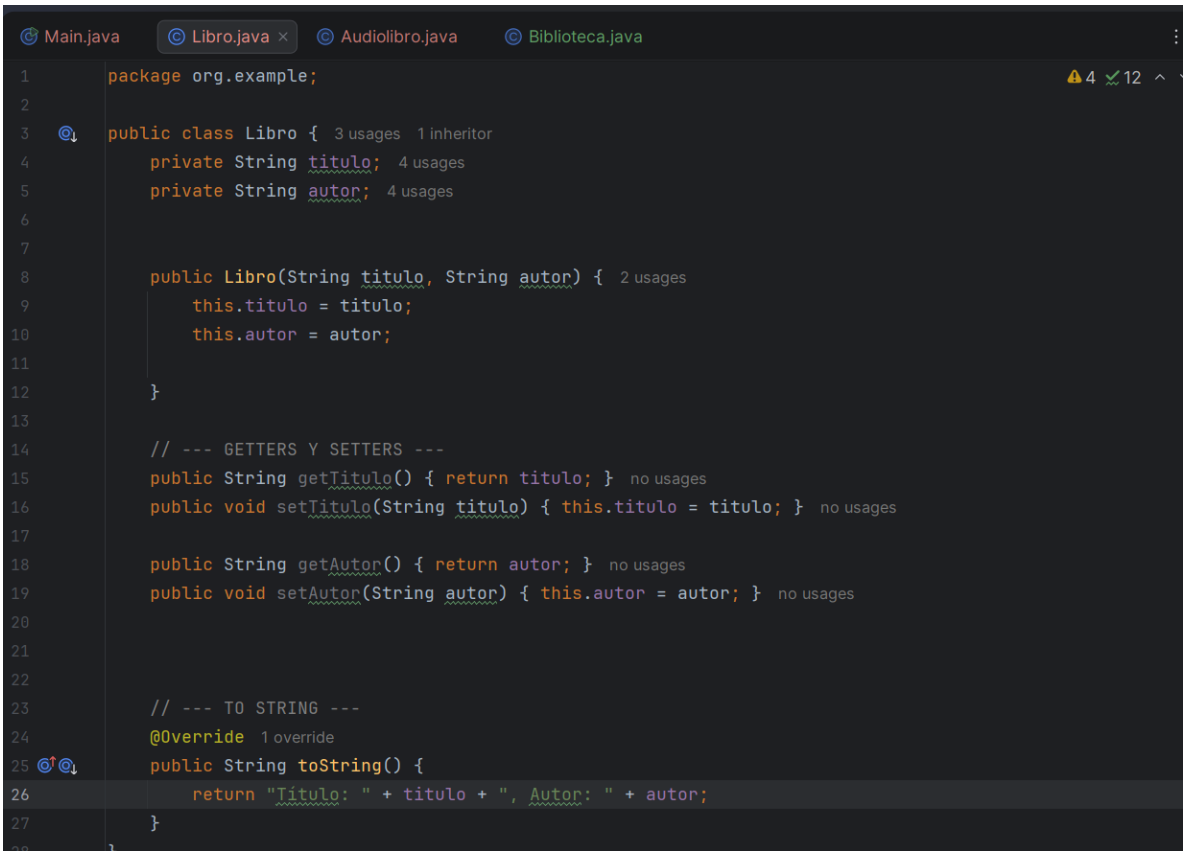
Además, el sistema integra una relación de agregación mediante la clase Biblioteca, la cual actúa como un contenedor dinámico que agrupa y gestiona todos los recursos. Esta organización nos permite aplicar los pilares de la POO: el encapsulamiento para proteger la información, el polimorfismo para manejar distintos formatos bajo una misma lista, y la abstracción para organizar el catálogo de forma profesional.

DIAGRAMA DE CLASES

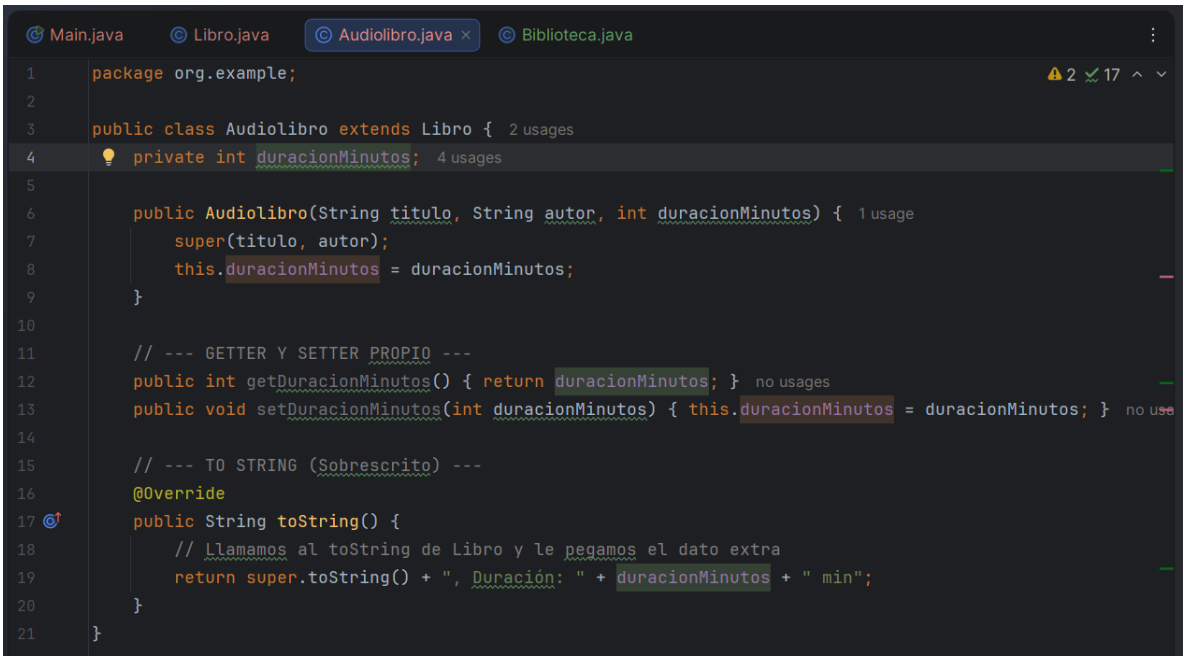


CODIGO

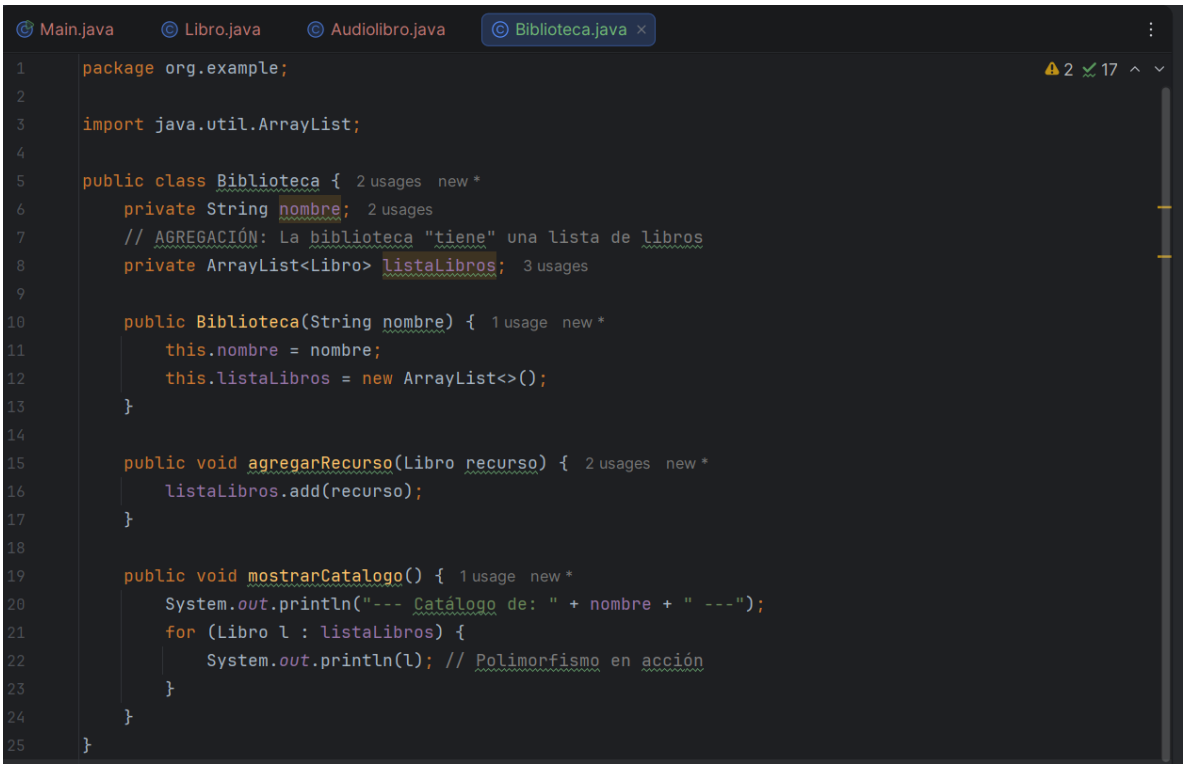
clase Libro



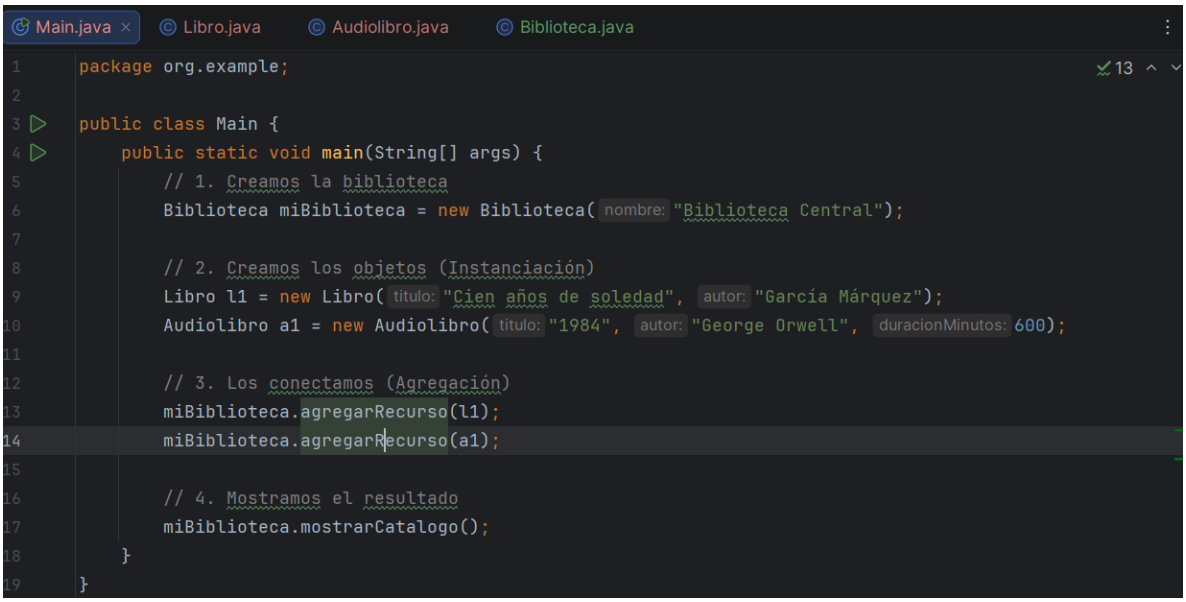
Clase Audiolibro



Clase Biblioteca



Clase Main



Muestra en consola

CONCLUSIONES

Se evidencio lo sencillo que es trabajar cuando ya tienes un diseño definido ya que al usar clases como plantillas, puedes crear todos los libros que necesites rápidamente, sabiendo que todos van a funcionar igual de bien y tendrán la misma estructura.

También se evidencio el ahorro de tiempo con la herencia al a plicar la herencia fue clave para no trabajar doble permitió reciclar lo que ya había escrito en la clase principal (Libro) y usarlo en la otra, logrando que el código se vea mucho más limpio, ordenado y, sobre todo, fácil de entender.

LINK REPOSITORIO

https://github.com/sjjrodriguez/repaso-1/tree/main

BIBLIOGRAFÍA

Moya, R. (8 de abril de 2014). *Herencia en Java, con ejemplos*. Jarroba. https://jarroba.com/herencia-en-la-programacion-orientada-a-objetos-ejemplo-en-java/