

Sistema de Gestión de Biblioteca
Documento de Soporte Técnico



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

- Melisa Maldonado Melenge 20231020110 -

- Jean Pierre Mora Cepeda 20231020105 -

- Juan Felipe Guevara Olaya 20231020117 -

-Universidad Distrital Francisco José de Caldas-

-Ingeniería de Sistemas-

- Ciencias de la Computación -

-Luz Deicy Alvarado Nieto-

-2025-

Índice de Contenidos:

Estructura General del Proyecto.....	2
Archivos principales:.....	2
Arquitectura central.....	3
Módulos.....	5
ControlObras.h.....	5
ManejadorArchivos.h.....	5
ArbolRojiNegro.h.....	5
Componentes clave.....	5
Clase de control.....	5
Plantilla de lista.....	6
Matriz de operaciones del menú.....	7
Dependencias entre Archivos .h y .o.....	7
Estructuras de Datos Utilizadas.....	7
Árbol Rojo-Negro.....	7
¿Qué es un Árbol Rojo-Negro?.....	7
Ventajas clave:.....	8
¿Por qué se eligió sobre otras estructuras?.....	8
Multilista.....	9

Estructura General del Proyecto

El proyecto está dividido en múltiples módulos encabezados por archivos .h (cabecera) y .o (objetos compilados). Utiliza estructuras avanzadas para la representación y manipulación eficiente de datos bibliográficos.

Archivos principales:

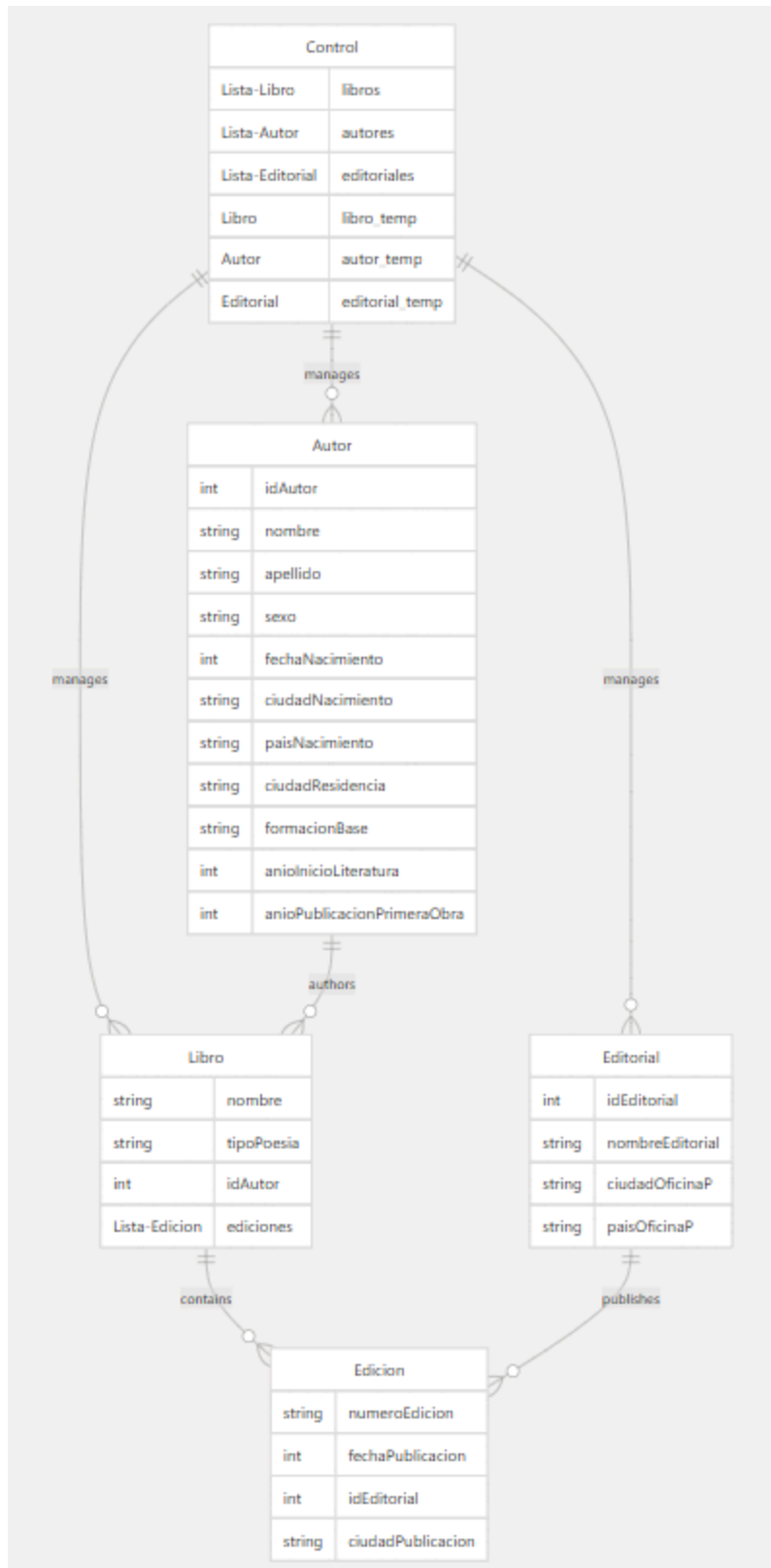
- ControlAutores.h, ControlEditoriales.h, ControlObras.h, ControlEdiciones.h: Gestión de entidades.
- ManejadorArchivos.h: Entrada y salida de archivos.
- ArbolRojiNegro.h: Estructura de datos para indexación.
- Multilista.h: Organización jerárquica de datos.
- MenuBiblioteca.h: Interfaz y navegación.
- GestionBiblioteca.h: Coordinador general de procesos.

Arquitectura central



Relaciones entre entidades

El sistema modela las relaciones editoriales literarias a través de cuatro entidades principales:



Módulos

ControlObras.h

- Gestiona la información de las obras literarias.
- Permite agregar, consultar, eliminar y modificar obras.
- Se apoya en listas enlazadas y referencias cruzadas a autores, ediciones y editoriales.

ManejadorArchivos.h

- Encargado de leer y escribir los datos persistentes en archivos planos.
- Cada tipo de entidad tiene su propio formato de almacenamiento.
- Usa funciones para parseo de líneas, escritura estructurada y validación.

ArbolRojiNegro.h

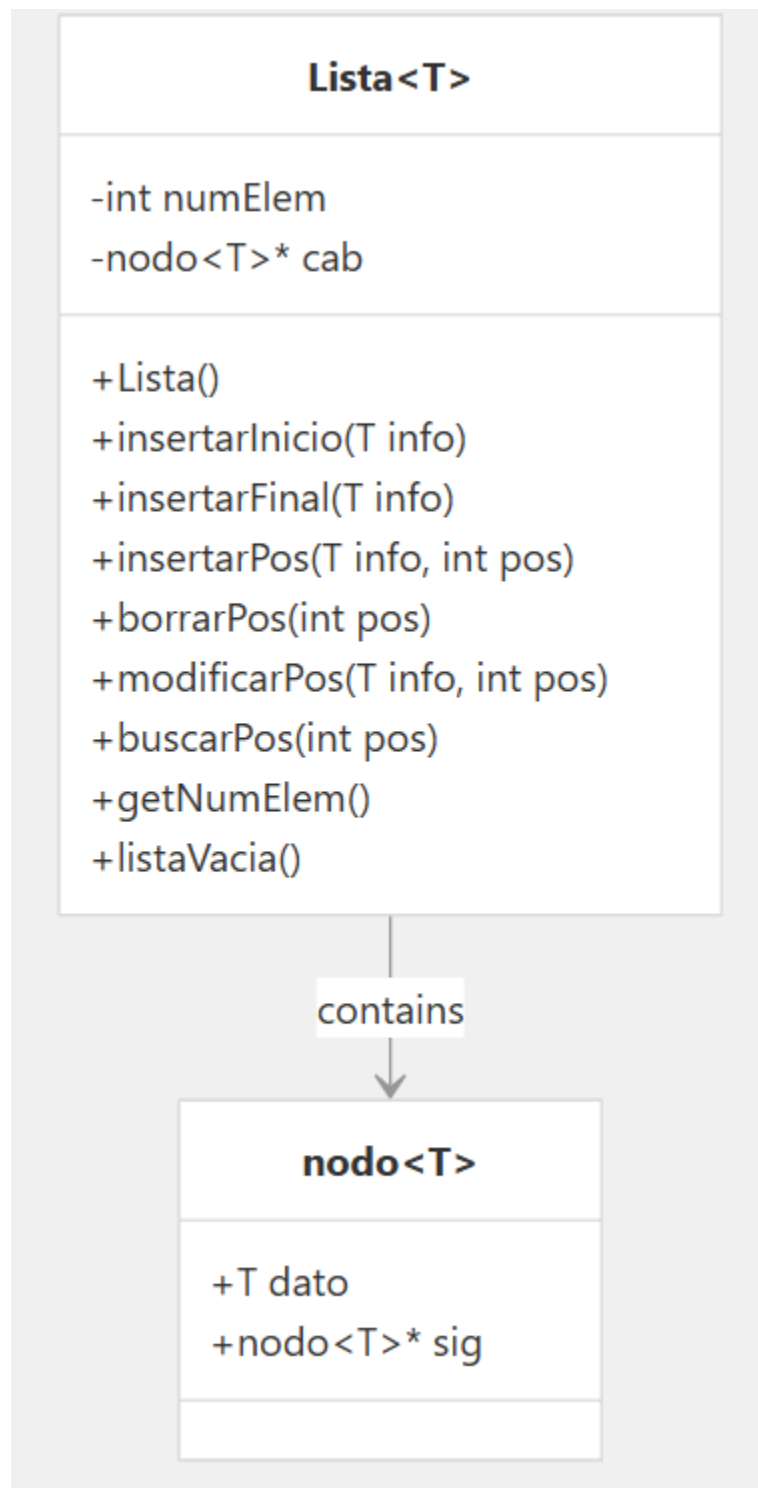
- Implementa un árbol rojo-negro, una estructura auto-balanceada que optimiza las búsquedas.
- Se utiliza para organizar autores, obras y ediciones con acceso eficiente.
- Las operaciones principales son inserción, búsqueda y rotaciones (balanceo).

Componentes clave

Clase de control

Componente	Objetivo	Métodos clave
Sistema de menú	Navegación de la interfaz de usuario	menuPrincipal(), menuAgregarLibro(), menuModificarAutor()
Recopilación de datos	Almacenamiento de entidades	Lista<Libro> libros, Lista<Autor> autores, Lista<Editorial> editoriales
Operaciones CRUD	Manipulación de datos	Operaciones de inserción, modificación y visualización para todas las entidades
Manejo de entrada	Interacción del usuario	rtaUsuario, rtaUsuarioStr para capturar la entrada del usuario

Plantilla de lista



Matriz de operaciones del menú

El sistema proporciona 16 operaciones principales organizadas por funcionalidad:

Categoría	Operaciones	Estado de implementación
Operaciones de consulta	1-7: Consultas literarias complejas	Implementado
Gestión de libros	8: Agregar, 9: Eliminar, 10: Modificar	Agregar/Modificar implementado, Eliminar marcador de posición
Gestión de autores	11: Agregar, 12: Eliminar, 13: Modificar	Agregar/Modificar implementado, Eliminar marcador de posición
Gestión de editoriales	14: Agregar, 15: Eliminar, 16: Modificar	Agregar/Modificar implementado, Eliminar marcador de posición

Dependencias entre Archivos .h y .o

- Control*.h → requiere ManejadorArchivos.h para guardar/cargar.
- GestionBiblioteca.h → importa todos los Control*.h y MenuBiblioteca.h.
- Archivos .o como MenuBiblioteca.o, ManejadorArchivos.o, etc., son productos de compilación y deben generarse antes de ejecutar el binario principal.

Estructuras de Datos Utilizadas

Árbol Rojo-Negro

El sistema utiliza un Árbol Rojo-Negro (ARN) como estructura central para la indexación eficiente de entidades clave, como autores, obras y ediciones. Esta decisión responde a criterios de eficiencia, flexibilidad y facilidad de mantenimiento frente a otras alternativas.

¿Qué es un Árbol Rojo-Negro?

Es un árbol binario de búsqueda balanceado, en el cual cada nodo tiene un color (rojo o negro) y se mantiene un conjunto de reglas que garantizan un equilibrio, lo que permite mantener la eficiencia sin realizar tantas rotaciones como en los AVL.

Ventajas clave:

- Tiempo de operación garantizado: las inserciones, eliminaciones y búsquedas se realizan en tiempo $O(\log n)$.
- Balanceo menos estricto que el AVL, lo que reduce el número de rotaciones, especialmente en inserciones consecutivas.
- Ideal para sistemas donde se realizan muchas inserciones y búsquedas, como en una base de datos de autores u obras.

¿Por qué se eligió sobre otras estructuras?

Árbol Binario de Búsqueda (ABB)

- Ventajas: más simple de implementar.
- Problema: en casos degenerados (datos ordenados), puede convertirse en una lista enlazada con operaciones $O(n)$, lo cual rompe la eficiencia.
- Veredicto: descartado por falta de balanceo automático.

Árbol AVL

- Muy eficiente: mantiene el árbol balanceado de forma más estricta.
- Sin embargo, realiza más rotaciones que un ARN, lo que lo hace menos adecuado para inserciones masivas rápidas.
- Veredicto: buena opción, pero más costosa en inserciones que el ARN.

Árbol B / B+

- Ideales para bases de datos o almacenamiento externo (disco).
- El proyecto se ejecuta en memoria, no necesita estructuras n-arias ni acceso por bloques.
- Veredicto: demasiado complejos para una aplicación en memoria.

Representaciones por Arreglos o Listas de Hijos

- Arreglos: útiles para árboles completos y estructuras predecibles.
- En árboles binarios de búsqueda dinámicos como los nuestros, usar arreglos es ineficiente en memoria y difícil de mantener.
- Veredicto: no adecuado para árboles binarios balanceados y dinámicos.

Árbol como Grafo

- Teóricamente viable.
- Añade una capa de complejidad innecesaria (más útil en representaciones generales de relaciones no jerárquicas).
- Veredicto: demasiado general para nuestro caso específico.

Conclusión sobre los Árboles Rojo-Negro

La elección del árbol rojo-negro combina:

- La eficiencia logarítmica garantizada.
- Un balanceo automático más liviano que el AVL.
- Simplicidad suficiente para implementar en memoria.
- Mantenimiento adecuado de los datos incluso con muchas inserciones y búsquedas.

Es la opción óptima para un sistema como el nuestro, donde se consultan e insertan grandes volúmenes de datos en tiempo real con estabilidad.

Implementación: codificado en ArbolRojiNegro.h.

Multilista

La multilista es una estructura empleada para representar relaciones múltiples entre entidades:

1. Usos:

- Obras enlazadas con autores y editoriales.
- Ediciones organizadas jerárquicamente por obra.
- Relaciones dinámicas entre nodos de diferentes tipos.

2. Justificación:

- Permite manejar nodos con múltiples campos de enlace.
- Soporta jerarquías y agrupamientos lógicos sin requerir estructuras rígidas como árboles n-arios.

Implementación: codificado en Multilista.h.