**ACTIVIDAD 4 - PROYECTO: MODELAR LAS INTERACCIONES ENTRE LOS USUARIOS Y LOS LIBROS**

VALENTINA RAMIREZ REINA

**CÓDIGO ID:** 100195519

**DOCENTE:** JOSE RUIZ

**MATERIA:** ESTRUCTURA DE DATOS

CORPORACION UNIVERSITARIA IBEROAMERICANA

FACULTAD INGENIERÍA

**Desarrollo del Sistema de Biblioteca utilizando Árboles en Node.js**

**1. Comprensión y Diseño del Modelo**

El primer paso crucial en el desarrollo del sistema fue comprender completamente el modelo de la biblioteca. El sistema necesita gestionar varias operaciones relacionadas con los libros y los usuarios. A partir de esta necesidad, se identificaron los siguientes aspectos clave que el sistema debe cubrir:

* **Gestión de Libros:** Registrar libros de la base de datos. Los libros tienen atributos como id, título, autor, stock.
* **Gestión de Usuarios:** Los usuarios son aquellos que tomarán prestados los libros. Se debe gestionar su información personal, como nombre, identificación y libros actualmente prestados.
* **Préstamos y Devoluciones:** La biblioteca debe permitir el préstamo de libros y llevar un seguimiento de qué usuario tiene qué libro. Al devolver los libros, se debe actualizar el estado del libro en el sistema.

Para implementar este sistema, se decidió que las relaciones entre los elementos (usuarios, libros y préstamos) serían más fáciles de manejar utilizando estructuras de árbol. Esto se debe a que las operaciones principales (inserción y visualización) son bastante comunes en las bibliotecas, y los árboles ofrecen una eficiencia considerable en estos procesos.

2**. Elección de Árboles Adecuados**

Basándonos en las operaciones identificadas en el paso anterior, seleccionamos los tipos de árboles más adecuados para cada caso de uso:

* **Árbol Binario para gestionar usuarios:**  
  Para gestionar los usuarios, decidimos también utilizar un Árbol Binario, donde los usuarios se ordenan por su número de identificación o algún otro identificador único. Esto facilitará la búsqueda rápida de usuarios cuando se necesite asociar un préstamo a un usuario específico.
* **Listas de préstamos en cada usuario:**  
  Para mantener un seguimiento de los libros que cada usuario ha tomado prestados, se utilizarán listas enlazadas dentro de los nodos de los usuarios. Esta estructura permite gestionar fácilmente los préstamos, ya que cada vez que un libro es prestado, se agrega un nuevo nodo a la lista del usuario correspondiente.

3. **Implementación de Operaciones**

La implementación de las operaciones básicas de los árboles se realizó de acuerdo con las necesidades del sistema. A continuación, se describen las principales operaciones que se implementaron:

* **Inserción de Libros:**  
  Al agregar un nuevo libro a la biblioteca, se inserta un nodo en el Árbol Binario de Búsqueda. La clave de cada nodo será el identificador del libro (por ejemplo, el id o el título). Si el árbol ya contiene un libro con la misma clave, se evita la inserción para evitar duplicados.
* **Préstamos de Libros:**  
  Cuando un usuario solicita un préstamo, el sistema verifica la disponibilidad del libro en el árbol de libros. Si el libro está disponible, el sistema agrega un nuevo nodo en la lista de préstamos del usuario correspondiente. El estado del libro se marca como "prestado" y se actualiza en el árbol.
* **Devolución de Libros:**  
  Al devolver un libro, el sistema busca el préstamo correspondiente en la lista de préstamos del usuario y elimina el nodo de la lista. El estado del libro se marca como "disponible" y se actualiza en el árbol de libros.

**4. Integración y Pruebas**

Una vez que las operaciones básicas de los árboles fueron implementadas, el siguiente paso fue integrar todo el sistema utilizando Express y EJS para crear la interfaz web.

El servidor de Express se encargó de manejar las solicitudes HTTP, mientras que EJS se utilizó para renderizar las vistas. Las rutas principales que se implementaron fueron:

* /agregarLibro: Para agregar un libro al sistema.
* /agregarUsuario: Para agregar un libro al sistema.
* /prestarLibro: Para registrar un préstamo de libro a un usuario.
* /devolverLibro: Para devolver un libro y actualizar el estado en el sistema.

Se realizaron varias pruebas para asegurarse de que las operaciones funcionaran correctamente. Se probaron casos típicos (como agregar libros y registrar préstamos) y se realizaron pruebas de borde.

**5. Optimización y Eficiencia**

El uso de árboles binarios permitió que las operaciones de inserción fueran eficientes en comparación con una lista lineal. Sin embargo, al evaluar el rendimiento, se consideraron las siguientes mejoras:

* Persistencia de Datos: Inicialmente, el sistema no implementaba persistencia, por lo que se agregó la capacidad de guardar y cargar los árboles de libros y usuarios desde archivos JSON para asegurar que los datos no se pierdan cuando el servidor se reinicie.

**6. Documentación y Presentación**

Durante el desarrollo, se mantuvo una documentación detallada de cada paso. Esta documentación incluye:

* La estructura de cada árbol (qué datos se almacenan en cada nodo).
* Descripción de las operaciones implementadas (inserción, etc.).

En la presentación final, se explicó cómo el modelo de árbol ayuda a optimizar las operaciones de gestión de libros y usuarios. También se discutieron los desafíos encontrados, como la gestión de duplicados y el balanceo del árbol, y cómo se resolvieron.

**7. Autoevaluación y Aprendizaje**

Este proyecto no solo permitió aplicar conceptos de estructuras de datos, sino que también proporcionó una comprensión más profunda de cómo gestionar varias cantidades de datos de manera eficiente. Los principales aprendizajes fueron:

* La importancia de elegir la estructura de datos adecuada según las operaciones que se van a realizar.
* Cómo utilizar árboles binarios para optimizar la búsqueda y la inserción de elementos.
* La integración de estas estructuras con una aplicación web para gestionar un sistema real de biblioteca.

Reflexionar sobre este proyecto mostró cómo las estructuras de árboles no solo son útiles para problemas académicos, sino que son esenciales para la eficiencia en sistemas reales.

**Enlace git**

<https://github.com/vramirezreina/Sistema_Biblioteca.git>

**Enlace video funcionamiento del sistema**

[**SISTEMA\_BIBLIOTECA.mp4**](https://laiberocol-my.sharepoint.com/:v:/g/personal/vramir31_estudiante_ibero_edu_co/EQmo-zjVxBVJga5NiT71tpMBvthS_OShUrmiwFRfcp3fyg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=e6FbgP)

**BIBLIOGRAFÍA**

* Ruiz Rodríguez, R. (2009). Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java. El Cid Editor.
* Zohonero Martínez, I. y Joyanes Aguilar, L. (2008). Estructuras de datos en Java. McGraw-Hill. España.
* Fritelli, V. Guzman, A. & Tymoschuk, J. (2020). Algoritmos y estructuras de datos (2a. ed.). Jorge Sarmiento Editor - Universitas. (Págs. 311 - 341).
* Joyanes Aguilar, L. (2020). Fundamentos de programación: algoritmos, estructura de datos y objetos. (Págs. 492- 518).