



Sistema de gestión de bibliotecas mediante algoritmos y estructuras de datos eficientes

Library management system using efficient algorithms and data structures

Brenda Vianey Hernandez Miramontes

brhernandezmi2697@uaemex.mx

ORCID: 0009-0006-1093-5717

Marco Alberto Mendoza Pérez¹

mamendozap@uaemex.mx

ORCID: 0000-0003-4911-4757

Resumen

El presente artículo presenta el desarrollo de un sistema de gestión de biblioteca (SGB) diseñado para optimizar las operaciones bibliotecarias, como la catalogación de libros, la consulta de usuarios, la gestión de préstamos, devoluciones, reservas, el cálculo de multas y la generación de reportes. Los procesos manuales o el uso de sistemas con funcionalidades limitadas generan dificultades en la administración de la información, además de errores y retrasos en su recuperación. Para atender esta problemática, se desarrolló un SGB utilizando Python y SQLite, integrando estructuras de datos y algoritmos que permiten administrar la información de manera eficiente, agilizar las consultas y mantener la integridad de las operaciones. El sistema se implementó mediante una arquitectura modular compuesta por módulos de gestión de libros, usuarios, préstamos y reportes. Asimismo, integra listas enlazadas para el historial de préstamos, tablas hash para la búsqueda de usuarios, árboles binarios de búsqueda para organizar el catálogo y colas FIFO para la administración de reservas. De igual forma, se incorporaron los algoritmos Merge Sort y búsqueda binaria para optimizar el ordenamiento y la localización de libros. El sistema fue evaluado mediante pruebas funcionales de las operaciones CRUD (Create, Read, Update y Delete) de cada módulo. Los resultados muestran que el sistema administra correctamente los libros, usuarios, préstamos, devoluciones, reservas y reportes, además de generar reportes en formatos PDF y Excel, manteniendo

¹ Ambos autores de Universidad Autónoma del Estado de Mexico.



la integridad y consistencia de los datos. En conjunto, los resultados demuestran que la integración de estructuras de datos y algoritmos contribuye a mejorar la eficiencia.

Palabras clave: Gestión de bibliotecas, Sistemas de información; Algoritmos; Bases de datos; Estructuras De Datos.

Abstract

This article presents the development of a Library Management System (LMS) designed to optimize library operations, including book cataloging, user management, loan processing, returns, reservations, fine calculation, and report generation. Manual processes or the use of systems with limited functionality often create difficulties in information management, as well as errors and delays in information retrieval. To address these challenges, an LMS was developed using Python and SQLite, integrating data structures and algorithms that enable efficient information management, streamline search operations, and preserve the integrity of library processes. The system was implemented through a modular architecture consisting of book management, user management, loan management, and reporting modules. In addition, it incorporates linked lists for loan history management, hash tables for user searches, binary search trees for catalog organization, and FIFO queues for reservation management. Merge Sort and binary search algorithms were also implemented to optimize book sorting and retrieval. The system was evaluated through functional testing of the CRUD (Create, Read, Update, and Delete) operations of each module. The results demonstrate that the system effectively manages books, users, loans, returns, reservations, and reports while generating reports in PDF and Excel formats and maintaining data integrity and consistency. Overall, the findings indicate that the integration of data structures and algorithms contributes to improving the efficiency of library management.

Keywords: Library Management, Information Systems; Algorithms; Databases; Data structures.

Fecha de recibido: 09 de julio 2026

Fecha de aprobación: 09 de julio 2026

Fecha de publicación: 20 de julio 2026



Introducción

En la actualidad, las bibliotecas desempeñan un papel fundamental acceso, organización y difusión de la información. Sin embargo, el constante crecimiento de los recursos bibliográficos y el aumento en el número de usuarios generan nuevos desafíos relacionados con la administración eficiente de libros, préstamos, devoluciones, reservas y registros de usuarios. Cuando dichos procesos se realizan de forma manual o a través de sistemas con funciones limitadas, pueden presentarse errores en el manejo de la información, retrasos en la localización, dificultades en el control de inventarios y una limitada capacidad para generar reportes que apoyen la toma de decisiones. La incorporación de sistemas informáticos en el ámbito bibliotecario ha permitido automatizar tareas administrativas, para mejorar la precisión y rapidez de dichos procesos. Estudios recientes sobre plataformas de servicios bibliotecarios señalan que estos sistemas cumplen funciones tanto administrativas como orientadas al servicio, aunque su implementación también enfrenta retos asociados con el manejo de grandes volúmenes de datos, la capacitación del personal y la disponibilidad de recursos tecnológicos (Liu & Shao, 2024). Asimismo, la adopción de tecnologías en bibliotecas académicas, como soluciones en la nube y sistemas automatizados, se han relacionado con beneficios como la flexibilidad, la escalabilidad, el ahorro de tiempo, la facilidad de acceso y la mejora en la gestión de servicios (Asim et al., 2024). No obstante, el desempeño de estos sistemas depende en gran parte de las estructuras de datos y algoritmos que son utilizados para almacenar, organizar y recuperar la información. En este contexto, los sistemas de gestión bibliotecaria han evolucionado a plataformas capaces de integrar procesos de catalogación, recuperación de información, administración de usuarios y análisis de datos, con el propósito de mejorar la calidad del servicio y la eficiencia operativa (Liu & Shao, 2024).

En los últimos años, investigaciones sobre sistemas de gestión de bibliotecas muestran una tendencia hacia plataformas integradas y digitales. Las Library Services Platforms han ampliado el alcance de los sistemas tradicionales al integrar la gestión de recursos físicos, electrónicos y digitales en entornos más flexibles (Liu & Shao, 2024). Además, el uso de tecnologías emergentes ha permitido mejorar procesos de acceso, control, seguimiento y administración de recursos bibliotecarios (Asim et al., 2024; Mammadov, 2025). No obstante, gran parte de estos desarrollos se enfocan en la funcionalidad general del sistema y no siempre explican cómo las estructuras de datos y los algoritmos influyen en el rendimiento computacional de las operaciones bibliotecarias.

Ante esta situación, surge la necesidad de desarrollar un Sistema de Gestión de Biblioteca que integre estructuras de datos como tablas hash, listas enlazadas, árboles binarios de búsqueda y colas,



así como algoritmos de ordenamiento y búsqueda eficientes, con el propósito de optimizar la administración de libros, usuarios, préstamos, reservas y reportes estadísticos. Con lo cual surge la siguiente pregunta de investigación: ¿Cómo puede un sistema de gestión de biblioteca basado en estructuras de datos y algoritmos de búsqueda y ordenamiento mejorar la eficiencia en la administración de libros, usuarios, préstamos, reservas y reportes dentro de una biblioteca? Lo que conlleva a que el objetivo de este artículo es desarrollar un Sistema de Gestión de Biblioteca que permita administrar de manera eficiente libros, usuarios, préstamos, devoluciones, reservas y reportes mediante la integración de estructuras de datos y algoritmos de búsqueda y ordenamiento, demostrando su aplicación práctica en la solución de problemas reales relacionados con la administración y recuperación de información bibliotecaria.

Buscando de forma específica:

- Almacenar libros mediante una lista enlazada y mantener atributos como título, autor, categoría y disponibilidad.
- Registrar y buscar usuarios mediante una tabla hash con matrícula como clave principal.
- Ordenar el catálogo con MergeSort y realizar búsquedas eficientes mediante búsqueda binaria.
- Organizar libros jerárquicamente mediante un árbol binario de búsqueda.
- Administrar reservas con una cola FIFO para respetar el orden de solicitud.
- Controlar préstamos, devoluciones y multas por retraso con validación de fechas.
- Generar reportes de libros disponibles, libros prestados, usuarios activos y estadísticas generales.

Ahora bien, a través de su desarrollo se implementan estructuras de datos como listas enlazadas, tablas hash, árboles binarios de búsqueda y colas, así como algoritmos de ordenamiento y búsqueda que ayudan a optimizar el manejo de la información. De acuerdo con Goodrich et al. (2014), las estructuras de datos son esenciales para organizar eficientemente los datos y mejorar el desempeño de los sistemas informáticos. Asimismo, Cormen et al. (2022) señalan que los algoritmos de búsqueda y ordenamiento permiten reducir significativamente los tiempos de procesamiento cuando se trabaja con grandes cantidades de información. Por ello, el desarrollo de este Sistema de Gestión de



Biblioteca aporta beneficios a la organización y control de la información bibliotecaria como por su contribución al aprendizaje y aplicación práctica de las estructuras de datos y los algoritmos.

2. Trabajos relacionados

2.1 Evolución de los sistemas de gestión bibliotecas

Los sistemas tradicionales de gestión bibliotecaria están siendo reemplazados por plataformas de servicios bibliotecarios (Library Services Platforms), las cuales integran procesos de catalogación, circulación, recuperación de información, administración de recursos electrónicos y generación de analíticas para apoyar la toma de decisiones. La revisión sistemática realizada por Liu y Shao (2024) muestra que estas plataformas representan la tendencia dominante en bibliotecas académicas debido a su capacidad para centralizar la administración de información y mejorar la interoperabilidad entre diferentes servicios bibliotecarios. Los autores destacan que el desarrollo actual se orienta hacia arquitecturas más flexibles, escalables y adaptables a las necesidades de las instituciones educativas.

2.2 Transformación digital y automatización de bibliotecas

Diversas investigaciones coinciden en que la transformación digital de las bibliotecas está estrechamente vinculada con la incorporación de nuevas tecnologías de información. Hussain y Ameen (2023), al analizar el nivel de automatización de bibliotecas universitarias, encontraron que la implementación de sistemas automatizados mejora la eficiencia administrativa, reduce errores en la gestión documental y optimiza los servicios ofrecidos a los usuarios. Sin embargo, también señalan que muchas instituciones continúan enfrentando limitaciones relacionadas con infraestructura tecnológica, interoperabilidad entre sistemas y actualización permanente de sus plataformas, aspectos que condicionan el desempeño de los procesos bibliotecarios.

2.3 Papel de las estructuras de datos y los algoritmos en los sistemas de información

Desde el punto de vista de las Ciencias de la Computación, diversos autores coinciden en que el desempeño de un sistema de información depende tanto del diseño de la base de datos como de las estructuras de datos y los algoritmos implementados. Goodrich et al. (2014) sostienen que la eficiencia de operaciones como inserción, búsqueda, eliminación y actualización está determinada



por la estructura utilizada para organizar la información. Asimismo, Cormen et al. (2022) demuestran que algoritmos de ordenamiento y búsqueda como Merge Sort y Búsqueda Binaria permiten reducir significativamente la complejidad computacional frente a métodos secuenciales cuando se administran grandes conjuntos de datos. De forma complementaria, Silberschatz et al. (2020) y Elmasri y Navathe (2021) destacan que un diseño adecuado de bases de datos garantiza la integridad, consistencia y disponibilidad de la información, elementos indispensables para el correcto funcionamiento de cualquier sistema de gestión.

2.4 Síntesis crítica y vacío de investigación

En conjunto, la evidencia disponible muestra una coincidencia clara respecto a la necesidad de automatizar los procesos bibliotecarios y desarrollar plataformas cada vez más robustas para administrar recursos. Sin embargo, también pone de manifiesto un vacío de investigación relacionado con la integración explícita de estructuras de datos clásicas como listas enlazadas, tablas hash, árboles binarios de búsqueda y colas FIFO junto con algoritmos eficientes de ordenamiento y búsqueda dentro de un mismo sistema de gestión bibliotecaria desarrollado con fines académicos y formativos. Este vacío resulta particularmente relevante porque limita la comprensión de cómo los principios del diseño algorítmico contribuyen a mejorar el rendimiento de aplicaciones utilizadas para resolver problemas reales de administración de información.

2.5 Aportación

En este contexto, el presente trabajo aporta una propuesta diferenciada al diseñar e implementar un Sistema de Gestión de Biblioteca desarrollado en Python y SQLite que integra estas estructuras de datos y algoritmos dentro de una arquitectura funcional para la administración de libros, usuarios, préstamos, devoluciones, reservas y reportes. A diferencia de los estudios previos, esta investigación no se limita a la automatización de procesos bibliotecarios, sino que demuestra de manera práctica cómo los principios del diseño algorítmico contribuyen a optimizar la organización, recuperación y procesamiento de la información, constituyendo un caso de aplicación que vincula los fundamentos de las Ciencias de la Computación con una problemática real de gestión bibliotecaria.



3. Materiales y métodos

El desarrollo del Sistema de Gestión de Biblioteca se realizó siguiendo un enfoque de ingeniería de software orientado al diseño e implementación de una aplicación capaz de administrar libros, usuarios, préstamos, devoluciones, reservas y reportes mediante el uso de estructuras de datos y algoritmos de búsqueda y ordenamiento. El proceso comprendió el análisis de requerimientos, el diseño de la arquitectura del sistema, la implementación de los módulos funcionales, la integración de algoritmos y estructuras de datos, y la validación funcional del software. Se empleó el lenguaje de programación Python y el sistema gestor de bases de datos SQLite para garantizar la persistencia de la información y la correcta ejecución de las operaciones CRUD. Las estructuras implementadas incluyeron listas enlazadas, tablas hash, árboles binarios de búsqueda, colas FIFO, Merge Sort y búsqueda binaria, seleccionadas de acuerdo con los requerimientos funcionales de cada módulo.

3.1 Diseño del estudio

Se empleó un diseño aplicado de desarrollo tecnológico con alcance descriptivo, orientado al diseño, implementación y validación funcional de un sistema de gestión bibliotecaria. Este tipo de diseño resulta adecuado cuando el objetivo consiste en desarrollar una solución informática para resolver un problema específico mediante la integración de métodos computacionales (Hernández-Sampieri et al., 2014).

El proceso metodológico se desarrolló en cinco etapas:

Etapa 1: Análisis de requerimientos, donde se identificaron las necesidades funcionales relacionadas con la administración de libros, usuarios, préstamos, devoluciones, reservas y generación de reportes.

Etapa 2: Diseño del sistema, en el que se definió la arquitectura modular, el modelo de base de datos relacional y la selección de las estructuras de datos y algoritmos más adecuados para cada proceso.

Etapa 3: Implementación, mediante el lenguaje Python y SQLite, incorporando operaciones CRUD, autenticación de usuarios, manejo de préstamos, reservas, multas y generación de reportes.



Etapla 4: Integración de algoritmos y estructuras de datos, implementando listas enlazadas para historiales de préstamos, tablas hash para la búsqueda de usuarios, árboles binarios de búsqueda para la representación jerárquica del catálogo, colas FIFO para reservas, Merge Sort para el ordenamiento de libros y búsqueda binaria para consultas eficientes.

Etapla 5: Validación funcional, mediante pruebas de caja negra orientadas a verificar el correcto funcionamiento de cada módulo, la consistencia de la base de datos y la correcta ejecución de las operaciones implementadas.

La validez interna del desarrollo se garantizó mediante pruebas funcionales sobre todas las operaciones CRUD, verificando el funcionamiento de la base de datos, validación de entradas del usuario y comprobación del funcionamiento de los algoritmos implementados antes de su integración al sistema.

3.2 Población y muestra

La población de interés estuvo conformada por usuarios con formación en Ciencias de la Computación e Ingeniería en Computación, considerados potenciales evaluadores del sistema desarrollado. Para la validación funcional del sistema se seleccionó una muestra no probabilística por conveniencia, integrada por tres participantes, debido a que el objetivo de esta etapa consistió en evaluar el funcionamiento de la aplicación e identificar oportunidades de mejora, más que realizar inferencias estadísticas sobre una población mayor (Hernández-Sampieri et al., 2014). La muestra estuvo conformada por un estudiante del Doctorado en Ciencias de la Computación, un estudiante de Ingeniería en Computación y un docente de la carrera de Ingeniería en Computación, perfiles que permitieron brindar una retroalimentación desde diferentes niveles de experiencia académica y técnica. Esta diversidad favoreció la evaluación del sistema tanto desde la perspectiva de un usuario con formación avanzada en ciencias de la computación como desde la experiencia de un usuario en formación y de un docente con conocimientos en desarrollo de software y estructuras de datos.

3.3 Tecnologías utilizadas y procedimiento de validación

El Sistema de Gestión de Biblioteca fue desarrollado utilizando el lenguaje de programación Python 3.13, la interfaz gráfica se implementó mediante la biblioteca Tkinter, mientras que el servidor de la aplicación y los servicios internos se desarrollaron con el microframework Flask. Para el



almacenamiento de la información se utilizó SQLite, un sistema gestor de bases de datos relacional ligero, adecuado para aplicaciones locales y de mediana complejidad. El desarrollo y depuración del software se realizaron en Visual Studio Code, empleando como entorno de ejecución Windows 11. La arquitectura del sistema se organizó de forma modular, integrando los componentes de autenticación, gestión de libros, gestión de usuarios, préstamos, devoluciones, reservas y generación de reportes como se muestra en la figura 1. Asimismo, se incorporaron diversas estructuras de datos y algoritmos con el propósito de optimizar las operaciones del sistema. Se implementaron listas enlazadas para la administración del historial de préstamos, tablas hash para la búsqueda eficiente de usuarios, árboles binarios de búsqueda (BST) para organizar jerárquicamente el catálogo bibliográfico y colas FIFO para gestionar el orden de las reservas. Además, se integraron los algoritmos Merge Sort para el ordenamiento del catálogo y Búsqueda Binaria para la recuperación eficiente de libros previamente ordenados.

3.4 Análisis de los datos

El análisis de los resultados se realizó mediante un enfoque descriptivo, considerando las observaciones obtenidas durante las pruebas funcionales y el comportamiento del sistema en cada uno de los escenarios de uso ejecutados. La información recopilada se organizó de acuerdo con los módulos evaluados (gestión de libros, usuarios, préstamos, reservas y reportes), verificando el cumplimiento de los requisitos funcionales establecidos durante el diseño del sistema. Asimismo, se analizaron las observaciones realizadas por los participantes con el fin de identificar oportunidades de mejora relacionadas con la interfaz de usuario, la navegación y la representación gráfica de las estructuras de datos y algoritmos implementados.

Debido al carácter exploratorio de la validación y al reducido número de participantes ($n = 3$) como se visualiza en la tabla 1, no se realizaron pruebas estadísticas inferenciales. El análisis se centró en la verificación del funcionamiento del sistema y en la descripción de los hallazgos derivados de la interacción de los participantes con la aplicación, siguiendo un enfoque de análisis descriptivo para estudios de desarrollo tecnológico (Hernández-Sampieri et al., 2014).

Tabla 1. Usuarios

Variable	Categoría	n	%
Perfil	Estudiante de Doctorado en Ciencias de la Computación	1	33.3
	Estudiante de Ingeniería en Computación	1	33.3
	Docente de Ingeniería en Computación	1	33.3
Total	Participantes	3	100

Fuente: Elaboración propia.

4. Resultados

4.1 Desarrollo e implementación del sistema

Se desarrolló un Sistema de Gestión de Biblioteca orientado a automatizar la administración de libros, usuarios, préstamos, devoluciones, reservas y reportes estadísticos. La aplicación fue implementada utilizando Python como lenguaje de programación y SQLite como gestor de base de datos, organizándose en una arquitectura modular compuesta por cuatro módulos principales: Gestión de Libros, Gestión de Usuarios, Gestión de Préstamos, Reservas, Reportes y Estadísticas. Esta organización permitió separar las funciones del sistema y facilitar la integración de estructuras de datos y algoritmos para cada proceso como se visualiza en la figura 2.

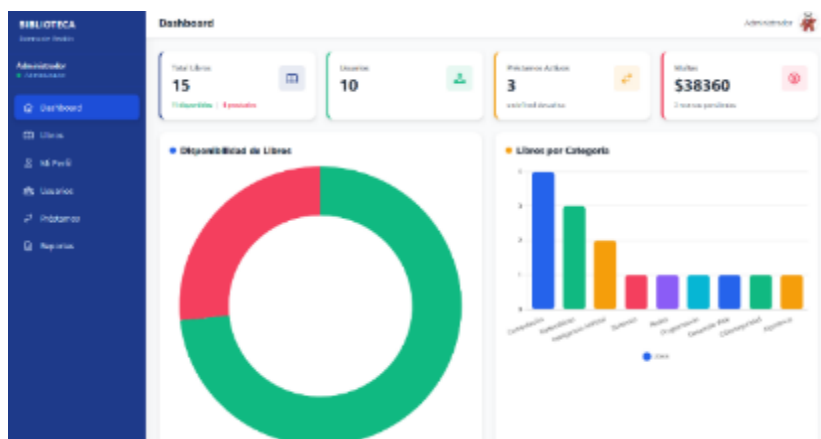


Figura 2. Sistema funcionando. Fuente: Elaboración propia.



4.2 Implementación de estructuras de datos y algoritmos

Durante las pruebas, las listas enlazadas almacenaron correctamente el historial de préstamos y se utilizaron para gestionar el historial de préstamos de los usuarios, las tablas hash permitieron realizar búsquedas rápidas de usuarios, mientras que los árboles binarios de búsqueda facilitaron la representación jerárquica del catálogo bibliográfico. Asimismo, las colas FIFO fueron empleadas para administrar el orden de atención de las reservas realizadas por los usuarios.

En cuanto a los algoritmos, Merge Sort se incorporó para ordenar el catálogo de libros previo a las consultas, mientras que la búsqueda binaria permitió localizar registros sobre listas previamente ordenadas con un menor número de comparaciones respecto a una búsqueda secuencial. La integración de estos algoritmos permitió mejorar la organización y recuperación de la información dentro del sistema.

4.3 Funcionalidad de los módulos

El módulo de gestión de libros permitió registrar, modificar, eliminar, consultar y ordenar el catálogo bibliográfico, además de visualizar la organización jerárquica mediante un árbol binario de búsqueda. El módulo de usuarios incorporó funciones para registrar, actualizar, eliminar y localizar usuarios mediante tablas hash, así como consultar su historial de préstamos mediante listas enlazadas. Por su parte, el módulo de préstamos gestionó préstamos, devoluciones, reservas, cálculo de multas y asignación automática de libros disponibles respetando el principio FIFO. El módulo de reportes generó información estadística relacionada con libros, usuarios, préstamos, reservas y multas, incluyendo la exportación automática de reportes en formato PDF.

4.4 Validación funcional

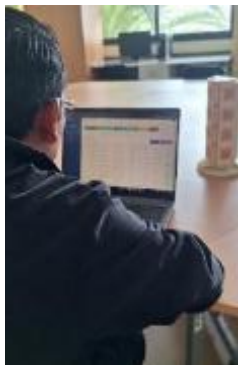


Figura 31. Prueba de usuario 1.
Fuente: Elaboración propia.



Figura 2. Prueba de usuario 2.
Fuente: Elaboración propia.



Figura 5. Prueba de usuario 3.
Fuente: Elaboración propia.

La validación funcional del Sistema de Gestión de Biblioteca se realizó con la participación de tres usuarios con diferentes perfiles académicos: un estudiante del Doctorado en Ciencias de la Computación, un estudiante de Ingeniería en Computación y un docente de Ingeniería en Computación. Cada participante ejecutó de manera individual los escenarios de uso para evaluar el funcionamiento de los módulos desarrollados y verificar la correcta integración de las estructuras de datos y algoritmos implementados. Las pruebas funcionales confirmaron el correcto funcionamiento de las operaciones CRUD en todos los módulos del sistema como se visualiza en la figura 3, 4 y 5.

Durante las pruebas se comprobó el funcionamiento de las operaciones de registro, consulta, actualización y eliminación (CRUD) de libros y usuarios, así como la gestión de préstamos, devoluciones, reservas y generación de reportes. En todos los escenarios evaluados, el sistema respondió de acuerdo con el comportamiento esperado, manteniendo la consistencia de la información almacenada en la base de datos y la sincronización entre la interfaz gráfica. La integración de las estructuras de datos también fue verificada durante las sesiones de prueba. Las listas enlazadas almacenaron correctamente el historial cronológico de préstamos de cada usuario; las tablas hash permitieron localizar usuarios de forma inmediata mediante su identificador; los árboles binarios de búsqueda organizaron el catálogo bibliográfico respetando su estructura jerárquica; y las colas FIFO administraron las reservas conservando el orden de llegada de las solicitudes. Asimismo, el algoritmo Merge Sort ordenó correctamente el catálogo antes de la ejecución de la Búsqueda Binaria, permitiendo recuperar los registros sin inconsistencias durante las consultas realizadas por los participantes como se muestra en las figuras 6 y 7.



Figura 6. Ordenamiento mediante Merge Sort. Fuente: Elaboración propia.



Figura 7. Búsqueda Binaria. Fuente: Elaboración propia.

Como resultado de la evaluación, no se identificaron errores funcionales críticos que impidieran la ejecución de las operaciones principales del sistema como se muestra en la tabla 2. No obstante, los participantes realizaron observaciones relacionadas con la experiencia de uso, principalmente en la representación gráfica del funcionamiento de los algoritmos, la organización de la selección de libros y la navegación entre algunos módulos. Estas observaciones fueron incorporadas durante la etapa final del desarrollo mediante ajustes en la interfaz de usuario y en la organización de los elementos visuales, obteniéndose una versión final con una interacción más clara e intuitiva.

Tabla 2. Resultados de la validación funcional del sistema.

Escenario evaluado	Resultado obtenido
Registro de libros	El sistema almacenó correctamente los registros en SQLite.
Consulta de libros	La búsqueda binaria recuperó correctamente los libros previamente ordenados.
Actualización de libros	Los cambios se reflejaron correctamente en la interfaz y la base de datos.



Eliminación de libros	Los registros fueron eliminados sin afectar la integridad de la base de datos.
Registro de usuarios	Las tablas hash permitieron localizar rápidamente los usuarios registrados.
Gestión de préstamos	Los préstamos actualizaron correctamente la disponibilidad de los libros.
Gestión de devoluciones	El sistema restauró correctamente la disponibilidad del material bibliográfico.
Gestión de reservas	Las colas FIFO conservaron el orden de atención de las solicitudes.
Generación de reportes	Se generaron correctamente reportes en formato PDF y Excel.
Integración de algoritmos	Merge Sort y Búsqueda Binaria funcionaron correctamente durante las consultas.

Fuente: Elaboración propia

Los resultados de la validación funcional evidencian que la integración de estructuras de datos específicas para cada proceso permitió administrar correctamente las operaciones de registro, consulta, actualización, eliminación, préstamos, devoluciones y reservas sin inconsistencias durante las pruebas realizadas con los tres participantes. Estos hallazgos muestran que la arquitectura implementada favorece la organización y recuperación eficiente de la información bibliotecaria, en concordancia con los principios descritos por Goodrich et al. (2014) y Cormen et al. (2022).

5. Discusión

Los resultados obtenidos muestran que la integración de estructuras de datos específicas para cada proceso contribuye a mejorar la organización y recuperación de la información bibliotecaria. La implementación de tablas hash, listas enlazadas, árboles binarios de búsqueda y colas FIFO permitió asignar a cada operación la estructura más adecuada según sus características funcionales, lo que coincide con lo señalado por Goodrich et al. (2014), quienes sostienen que la selección apropiada de estructuras de datos influye directamente en el rendimiento de las operaciones de inserción, búsqueda y actualización.

De manera similar, la incorporación de Merge Sort y búsqueda binaria confirma la importancia de utilizar algoritmos eficientes para el procesamiento de información. Cormen et al. (2022) indican que el rendimiento de los algoritmos de búsqueda depende en gran medida de que los datos se encuentren previamente ordenados, condición que fue satisfecha mediante la implementación de Merge Sort antes de ejecutar la búsqueda binaria. Esta integración permitió reproducir el



comportamiento esperado desde el punto de vista teórico y trasladarlo a un entorno de gestión bibliotecaria.

Asimismo, el uso de SQLite como sistema gestor de bases de datos favorece la persistencia e integridad de la información almacenada, aspecto que coincide con los principios descritos por Silberschatz et al. (2020) para el diseño de bases de datos relacionales. La estructura implementada permitió mantener la consistencia de los registros de libros, usuarios, préstamos y reservas durante las operaciones realizadas por el sistema.

Los resultados también son consistentes con investigaciones recientes sobre sistemas de recuperación de información. Liu y Shao (2024) señalan que la incorporación de estructuras de datos eficientes mejora la velocidad de acceso a los registros y optimiza la experiencia del usuario. En el sistema desarrollado, la combinación de tablas hash, búsqueda y árboles binarios favorece operaciones de búsqueda más eficientes de las operaciones de consulta favoreciendo una administración más eficiente del catálogo bibliográfico.

A diferencia de otros desarrollos de gestión bibliotecaria que se enfocan principalmente en la automatización administrativa, el presente trabajo integra explícitamente diversas estructuras de datos clásicas y algoritmos fundamentales como parte del diseño funcional del sistema. Esta característica constituye una aportación relevante desde el ámbito del diseño algorítmico, ya que demuestra cómo conceptos fundamentales de las Ciencias de la Computación pueden implementarse de manera conjunta para resolver un problema real de administración de información.

No obstante, el alcance del estudio se limita a la validación funcional del software. No se evaluó el desempeño del sistema mediante métricas experimentales de tiempo de ejecución, consumo de memoria o escalabilidad, ni se realizó una evaluación con usuarios finales para medir aspectos relacionados con la usabilidad o la experiencia de uso. En consecuencia, futuras investigaciones pueden incorporar análisis de rendimiento, pruebas de carga y evaluaciones con bibliotecarios y usuarios, con el propósito de cuantificar el impacto del sistema en escenarios de operación reales y fortalecer la evidencia sobre los beneficios de la integración de estructuras de datos y algoritmos en aplicaciones de gestión bibliotecaria.

6. Conclusiones y trabajos a futuro

El presente estudio permitió desarrollar e implementar un Sistema de Gestión de Biblioteca basado en Python y SQLite, integrando estructuras de datos y algoritmos fundamentales para optimizar la



administración de libros, usuarios, préstamos, devoluciones, reservas y reportes. La validación funcional confirmó el correcto funcionamiento de las funciones principales del sistema, validando la consistencia de la información almacenada y la adecuada integración entre la interfaz de usuario, la base de datos y los componentes algorítmicos. Asimismo, se comprobó que los algoritmos Merge Sort y Búsqueda Binaria facilitaron el ordenamiento y la recuperación eficiente de la información, mientras que las tablas hash, listas enlazadas, árboles binarios de búsqueda y colas FIFO respondieron adecuadamente a los requerimientos funcionales para la gestión bibliotecaria.

La aportación fue demostrar la aplicación integrada de estructuras de datos y algoritmos clásicos dentro de un sistema orientado a resolver una problemática real de administración bibliotecaria. A diferencia de otros desarrollos centrados únicamente en la automatización de procesos, esta propuesta demuestra cómo la selección adecuada de estructuras de datos influye en la organización, almacenamiento y recuperación de la información. Si bien, el estudio presenta algunas limitaciones. La validación se realizó mediante pruebas funcionales con una muestra reducida de participantes y estuvo orientada a verificar el correcto funcionamiento del sistema, por lo que no se evaluaron aspectos relacionados con el rendimiento computacional, el consumo de recursos, la escalabilidad del sistema ni la experiencia de uso mediante instrumentos estandarizados de usabilidad. En consecuencia, los resultados obtenidos corresponden a una validación funcional preliminar y no permiten establecer conclusiones sobre el desempeño del sistema en escenarios de operación con grandes volúmenes de información o un mayor número de usuarios.

Como trabajo futuro, se propone ampliar la evaluación del sistema a través de pruebas de rendimiento, escalabilidad y usabilidad con un mayor número de participantes y en contextos reales de operación bibliotecaria. Asimismo, se plantea integrar nuevas funcionalidades, como notificaciones automáticas por correo electrónico, autenticación mediante códigos QR o credenciales digitales, consulta del catálogo en línea, integración con aplicaciones móviles y medidas de seguridad. Estas mejoras contribuirán a incrementar la funcionalidad y capacidad de adaptación del sistema para su implementación en bibliotecas con mayores necesidades de gestión.



Referencias

- Asim, M., Arif, M., & Rafiq, M. (2024). Adoption and uses of cloud computing in academic libraries: A systematic literature review. *Journal of Information Science*.
<https://doi.org/10.1177/01655515241263272>
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
- Elmasri, R., & Navathe, S. B. (2021). *Fundamentals of database systems* (7th ed.). Pearson.
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *Data structures and algorithms in Java* (6th ed.). Wiley.
- Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, P. (2014). *Metodología de la investigación* (6.ª ed.). McGraw-Hill Education.
- Hussain, M., & Ameen, L. (2023). Exploring the automation landscape in university libraries of Khyber Pakhtunkhwa, Pakistan: Status, problems and implications. *The Journal of Academic Librarianship*, 49(6), 102787. <https://doi.org/10.1016/j.acalib.2023.102787>
- Liu, Z., & Shao, B. (2024). A systematic review of library services platforms research and research agenda. *Library & Information Science Research*, 46(4), 101325. <https://doi.org/10.1016/j.lisr.2024.101325>
- Mammadov, S. (2025). A user-centric smart library system: IoT-driven innovation. *Applied Sciences*, 15(7), 3792. <https://doi.org/10.3390/app15073792>
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database system concepts* (7th ed.). McGraw-Hill Education.