

UNIVERSITAT_{DE}
BARCELONA

Trabajo Final de Máster
en Humanidades Digitales

Curso: 2023-2024

Título: Diseño e Implementación de un Sistema de Gestión de
Bibliotecas basado en modelo MVC

Estudiante: Qigeng Peng

Xiaoping Wang

Tutor: Juan-José Boté

Resumen

A medida que se expanden las dimensiones de las bibliotecas, aumenta el número de obras almacenadas, y se amplió el círculo de lectores, también se intensifican las necesidades de gestión de la información. Especialmente desde el inicio de la era de alta informatización y la popularización de las computadoras, la gestión tradicional de bibliotecas ha quedado atrás frente a las tendencias de desarrollo futuras, haciendo que la creación de un sistema moderno y eficiente de gestión de bibliotecas sea un imperativo.

En respuesta a las necesidades de las bibliotecas en tiempos modernos y considerando el estado actual de desarrollo de los sistemas de gestión de información bibliotecaria, el presente artículo presenta el diseño y desarrollo de un sistema de gestión de bibliotecas (SGB). Este sistema se desarrolla a partir de los contenidos y características de las actividades de gestión bibliotecaria, alcanzando la automatización y digitalización de diversas tareas de la biblioteca.

Después de una comparativa de las arquitecturas C/S (Cliente/Servidor) y B/S (Navegador/Servidor), se ha decidido adoptar el esquema de diseño basado en la C/S para el SGB. El sistema emplea la estructura MVC (Modelo-Vista-Controlador) para lograr la separación de código. En cuanto a la programación, se ha utilizado el lenguaje Java, creando una base de datos de fondo con MySQL para almacenar la información del sistema. La plataforma de desarrollo seleccionada es IntelliJ IDEA. El diseño de la interfaz es intuitivo, el proceso de operación del sistema es sencillo y rápido, y cuenta con bajos costos de mantenimiento y operación.

El sistema ofrece módulos funcionales tales como gestión de información básica de usuarios, gestión de préstamos y devoluciones, gestión de libros, gestión de multas, etc. Los usuarios del sistema son lectores y administradores de biblioteca. Finalmente, este artículo emplea métodos de prueba funcional para perfeccionar el sistema.

Palabras clave: SGB, C/S, MVC, Java, MySQL

Abstract

As libraries expand in scale, the number of books stored increases, and the reader base broadens, the demands for information management have also escalated. Especially in the era of high-speed informatization and the widespread adoption of computers, traditional library management has become inadequate for future development trends, making it imperative to develop a novel and efficient library management system.

This paper, in response to the needs of libraries in the new era and considering the current development status of library management information systems, designs and develops a library management system (LMS). Originating from the content and characteristics of library management activities, this system realizes the automation and digitization of various operational tasks in the library.

After comparing and studying the C/S (Client/Server) and B/S (Browser/Server) structures of two management information systems, this paper adopts a design and development plan for the library management system based on the C/S architecture. The system utilizes the MVC (Model-View-Controller) model structure to achieve code separation. Software programming is conducted in Java, with MySQL used to establish a backend database for storing system data. The development platform selected is IntelliJ IDEA. The interface design is intuitive, and the system operation process is convenient and efficient, with low maintenance and operational costs.

The system implements functional modules such as user basic information management, borrowing and returning management, book management, and fine management. System users include readers and library administrators. Finally, we adopt functional testing methods to optimize the system.

Keywords: LMS, C/S, MVC, Java, MySQL

Tabla de correspondencia de abreviatura

Abreviaturas	Nombre completo en inglés	Nombre completo en español
SGB	Library Management System	Sistema de Gestión de Bibliotecas
MVC	Model-View-Controller	Modelo-Vista-Controlador
C/S	Client/Server	Cliente/Servidor
B/S	Browser/Server	Navegador/Servidor
GUI	Graphical User Interface	Interfaz Gráfica de Usuario
DAO	Data Access Object	Objeto de Acceso de Datos

Índice

1. Introducción	7
1.1 Antecedentes e Importancia del Estudio	7
1.2 Objetivos, Contenido y Estructura del Estudio.....	8
2. Revisión de Literatura.....	11
2.1 Identificación del objetivo investigador.....	11
2.2 Identificación de estudios relevantes	12
2.3 Selección de estudios	13
2.4 Mapeo de datos	14
2.5 Recopilación, síntesis y comunicación de los resultados.....	15
2.5.1 Características técnicas fundamentales de SGB	15
2.5.2 Módulos Funcionales en los SGB.....	16
2.5.3 Discusiones	17
2.6 Resumen.....	18
3. Resumen de herramientas de desarrollo de sistemas y tecnologías relacionadas....	19
3.1 Entorno de desarrollo IntelliJ IDEA	19
3.2 Introducción a tecnologías relacionadas	19
3.2.1 Comparación entre la arquitectura C/S y B/S.....	19
3.2.2 El patrón MVC.....	21
3.2.3 Java y la Interfaz Gráfica de Usuario (GUI) de Java	22
3.2.4 Tecnología de bases de datos MySQL	23
3.3 Resumen.....	24
4. Análisis de Necesidades	25
4.1 Estudio de Viabilidad	25
4.1.1 Viabilidad tecnológica	25
4.1.2 Viabilidad económica.....	26
4.1.3 Viabilidad operativa	26
4.2 Análisis de necesidades funcionales	27
4.2.1 Análisis de necesidades funcionales para administradores	27

4.2.2	Análisis de las necesidades funcionales de los usuarios	27
4.2.3	Presentación del Diagrama de Casos de Uso del Sistema	28
4.3	Análisis de requerimientos no funcionales	29
4.4	Resumen.....	30
5.	Diseño General.....	31
5.1	Diseño Global del Sistema.....	31
5.2	Diseño Detallado del Sistema	32
5.2.1	Módulo de Inicio de Sesión	32
5.2.2	Diseño de Módulos Funcionales del Sistema	33
5.2.3	Diseño de la Base de Datos del Sistema	39
5.3	Resumen.....	41
6.	Implementación del Sistema y Resultados	42
6.1	Estructura de Archivos del Sistema	42
6.2	Acceso a Datos.....	42
6.2.1	Implementación del DAO	43
6.3	Diseño e implementación de la interfaz de inicio de sesión	44
6.4	Realización de Módulos Funcionales	45
6.4.1	Implementación del Módulo de Gestión de Libros.....	45
6.5	Pruebas.....	48
6.5.1	Diseño de casos de prueba del sistema	48
6.6	Resumen.....	52
7.	Conclusión y Perspectivas	53
7.1	Conclusión	53
7.2	Perspectivas.....	54
	Bibliografía	55

1. Introducción

1.1 Antecedentes e Importancia del Estudio

Con el rápido avance de la tecnología informática, la automatización y la digitalización han pasado a ser factores determinantes en la innovación de diversos sectores. Las bibliotecas, como centros clave de almacenamiento y distribución de conocimiento, deben innovar en sus modelos de gestión para seguir el ritmo de la tecnología.

La biblioteca, como una entidad cultural de gran antigüedad, ha desempeñado desde tiempos un papel esencial en la preservación, difusión y acceso al conocimiento. Analizando sus funciones históricas, la biblioteca no solo tiene la misión de proporcionar a los usuarios una amplia gama de recursos de lectura, sino también promover el intercambio cultural, difundir el conocimiento y apoyar la investigación científica.

Sin embargo, el sistema tradicional de bibliotecas presenta desafíos. En primer lugar, el modelo de gestión manual tradicional se muestra ineficaz para cumplir su nueva función de procesar y actualizar información rápidamente. En segundo lugar, las colecciones de las bibliotecas tradicionales son limitadas, difícilmente llegando a satisfacer las necesidades de información diversa y personalizada de los usuarios. Además, las bibliotecas tradicionales tienen también limitaciones en cuanto a la colaboración y el intercambio de recursos.

Las características de las bibliotecas modernas abarcan la digitalización de recursos, la automatización de tareas y la prestación de servicios inteligentes mediante la incorporación de tecnologías avanzadas de la información. Entre ellas, el sistema de gestión de libros digitales destaca como un indicador relevante de modernización, pues permite gestionar préstamos, devoluciones y otras tareas habituales de manera automática, mejorando la eficiencia y evitando errores humanos (Zainal, 2019). Además, este sistema integra la gestión de recursos electrónicos y ofrece acceso

online a recursos digitales (Chowdhury, 2012), satisfaciendo a los usuarios modernos que buscan obtener información de manera eficiente y cómoda. En consecuencia, el sistema de gestión de libros digitales constituye una tendencia inevitable del desarrollo tecnológico, en línea de la era de la información.

1.2 Objetivos, Contenido y Estructura del Estudio

Confrontado al bajo rendimiento de los modelos de gestión tradicionales de bibliotecas, hemos recurrido a modelos similares para analizar y desarrollar un sistema de gestión de libros basado en el patrón MVC. MVC (Modelo-Vista-Controlador) es un patrón de diseño de software que divide una aplicación en tres partes interconectadas. La meta de este sistema es transformar la práctica de las operaciones manuales en bibliotecas tradicionales mediante la aplicación de tecnologías informáticas, apoyando así a los empleados en tareas cotidianas como la gestión de libros y archivos, el manejo de información de usuarios y el control de préstamos y devoluciones de libros. Este enfoque pretende aligerar el trabajo y presión de los empleados, al tiempo que aumenta la seguridad de los datos y la eficiencia del servicio de biblioteca, llevando a cabo las tareas de gestión sistemático y automáticamente. Por ello, durante el diseño y desarrollo de este sistema de gestión de libros, hemos realizado principalmente las siguientes actividades:

1. Mediante análisis exhaustivo, hemos adquirido un conocimiento completo sobre el sistema de gestión de libros basado en el patrón MVC.
2. A través de estudios de documentación, hemos explorado las bases de diseño y herramientas de implementación de sistemas similares, analizando características específicas del patrón MVC y comparando la aplicación práctica de las arquitecturas de red C/S y B/S.
3. Diseñado y desarrollado las diversas funcionalidades del sistema de gestión de libros. El sistema incluye dos interfaces: una para estudiantes y otra para

administradores. La interfaz estudiantil contiene módulos para inicio de sesión, búsqueda de libros, préstamo y devolución de libros; mientras que la administrativa incluye módulos para gestión de usuarios, gestión de libros, búsqueda de libros, gestión de administradores y gestión de multas por infracciones.

La organización y disposición del presente trabajo es la siguiente:

En el primer capítulo, Introducción. Se ofrece una breve exposición del contexto de investigación, destacando el florecimiento de los sistemas de gestión digital de libros impulsado por la construcción de bibliotecas modernas. Después de reconocer la importancia del estudio, se plantea el objetivo del sistema de gestión de libros que será diseñado y desarrollado, dando una visión general del contenido y estructura del artículo.

En el capítulo segundo, Revisión de Literatura. Mediante un análisis de alcance, se define el tema y alcance de investigación, estableciendo el marco teórico del trabajo y elaborando estrategias más científicas y eficaces.

En el capítulo tercero, Descripción de herramientas y tecnologías para el desarrollo. Se presenta el entorno y contexto tecnológico del sistema, destacando las herramientas como IntelliJ IDEA, así como tecnologías como la arquitectura C/S, el patrón MVC, el lenguaje Java, las bibliotecas Java GUI y la tecnología de bases de datos MySQL.

En el capítulo cuarto, Análisis de necesidades. Se lleva a cabo un análisis de viabilidad y necesidades del sistema, estableciendo una clasificación de usuarios, un flujo de trabajo, necesidades funcionales y necesidades no funcionales.

En el capítulo quinto, Diseño integral y detallado del sistema. Analizando los objetivos de diseño, se implementa la arquitectura funcional general del sistema, los módulos según las necesidades y el diseño de base de datos. Se utilizan diagramas de flujo, estructura y otros para visualizar el diseño.

En el capítulo sexto, Implementación y pruebas del sistema. Se describe la implementación de los distintos niveles del sistema de gestión de libros, mostrando imágenes y código de los módulos. También se llevan a cabo pruebas funcionales básicas del sistema.

En el último capítulo, Conclusión y futuro del sistema de gestión de biblioteca. Se resume el trabajo realizado, analizando problemas existentes y haciendo propuestas para mejorar el sistema en el futuro.

2. Revisión de Literatura

En la sección de revisión de literatura, hemos decidido emplear el método de revisión de ámbito (en inglés *scoping review*). Mediante esta revisión, estableceremos el marco teórico de la presente investigación, adquiriendo una comprensión básica del desarrollo teórico del SGB basado en el modelo MVC, la evolución de las herramientas de desarrollo y los puntos de mayor interés. De esta manera, tendremos una importante orientación para la construcción de la estructura y el contenido rellenado de nuestro trabajo.

La revisión de alcance constituye un método exhaustivo de análisis de literatura que permite a los investigadores describir los principales conceptos, tipos de evidencia y lagunas existentes en un área de estudio mediante la búsqueda sistemática, selección y consolidación de conocimientos previos. Este enfoque, caracterizado por su flexibilidad, sistematicidad y amplitud, resulta particularmente útil para aquellos que desean familiarizarse con un campo de investigación o rápidamente captar los avances alcanzados (Arksey y O'Malley, 2005). La revisión de alcance ha encontrado amplio empleo en diversas áreas de la informática, incluyendo el análisis de tendencias tecnológicas en desarrollo de software, la identificación de desafíos técnicos, la evaluación de herramientas y marcos de trabajo, el análisis de necesidades y funcionalidades de software, así como en estudios interdisciplinarios.

Una vez que hemos familiarizado con el concepto y las aplicaciones de la revisión de alcance, hemos adoptado el marco de cinco etapas propuesto por los mismos autores, quienes proporcionan una guía clara y sistemática para nuestro trabajo.

2.1 Identificación del objetivo investigador

Nuestro objetivo es realizar una revisión de alcance sobre el diseño y la implementación de sistemas de gestión de bibliotecas basados en la arquitectura MVC. En este proceso, prestaremos especial atención a las características del modelo MVC,

las formas de implementación de los sistemas de gestión de bibliotecas, los módulos funcionales del sistema y los métodos de prueba.

2.2 Identificación de estudios relevantes

Desarrollaremos una estrategia adecuada de búsqueda que incluye determinar la ruta de búsqueda y las palabras clave que nos ayudarán a encontrar estudios relacionados con nuestra pregunta de investigación.

En nuestro caso, hemos empleado dos grandes categorías de bases de datos: las bases de datos académicas específicas de la informática y la ingeniería de software, tales como IEEEExplore, ACM Digital Library y ScienceDirect, y los buscadores académicos integrales como CNKI (China National Knowledge Internet), Google Scholar y CQVIP (China Science and Technology Journal Database). Los métodos de búsqueda se han diseñado mediante la combinación de palabras clave específicas, términos libres, operadores booleanos, así como técnicas avanzadas de búsqueda estructurada proporcionadas por las bases de datos. Con esta estrategia, hemos utilizado las siguientes palabras clave aparte de las en inglés como en chino: *MVC*, *SGB*, *diseño de sistemas*, *implementación de sistemas*, *patrones arquitectónicos*, *desarrollo de software*, *programación orientada a objetos*, *diseño de bases de datos* y *experiencia de usuario*. El período de búsqueda abarca desde el inicio de las bases de datos hasta la fecha actual.

Dado que hemos aprovechado bases de datos internacionales y sitios web académicos chinos que incluyen bibliografía en múltiples idiomas, nuestra estructura teórica ha sido en gran medida influenciada por la perspectiva de autores de otros países, lo que nos permite observar las diversas opiniones multiculturales y tendencias de investigación existentes. Para ajustarnos a los distintos mecanismos de búsqueda ofrecidos por las diversas plataformas, hemos utilizado combinaciones similares de palabras clave, en concreto, se estructuran como abajo:

Tabla 2.1: Formas de búsqueda		
Bases de datos	Palabras de búsqueda	Forma de búsqueda
CNKI	Palabras clave: MVC “AND” Palabras clave: SGB	Campos de búsqueda estructurada
CQVIP	Palabras clave: MVC “AND” Palabras clave: SGB	Campos de búsqueda estructurada
ScienceDirect	Término: SGB “AND” Palabras clave: MVC	Campos de búsqueda estructurada
IEEEExplore	Todos los metadatos: MVC “AND” Todos los metadatos: SGB	Búsqueda avanzada
ACM Digital Library	Resumen: SGB “AND” Resumen: MVC	Campos de búsqueda estructurada
Google Académico	MVC, SGB, Java Swing	Palabras clave principales

2.3 Selección de estudios

Al seleccionar materiales de estudio, el paso fundamental es establecer criterios claros de inclusión y exclusión. En la siguiente tabla, podrá apreciar los estándares que deben cumplirse para la selección de diferentes documentos, así como las razones fundamentales por las que muchos de ellos fueron descartados:

Tabla 2.2: Criterios de elección	
Criterios de inclusión:	Criterios de exclusión:
Los objetos de estudio deben tratar específicamente sobre sistemas de gestión de bibliotecas basados en la arquitectura MVC.	Los objetos de estudio no mencionan explícitamente la arquitectura MVC o el SGB.
Los temas de investigación deben ser el diseño y la implementación de sistemas de gestión de bibliotecas.	Los contenidos de los documentos se centran principalmente en sistemas de gestión de bibliotecas que no utilizan la arquitectura MVC, o no están directamente relacionados con el diseño y la implementación de sistemas de gestión de bibliotecas basados en MVC.

Los documentos deben discutir detalladamente la aplicación, principios de diseño y tecnologías de implementación de la arquitectura MVC en sistemas de gestión de bibliotecas.	Los documentos cuyo texto completo no es disponible.
Los documentos deben ser estudios originales, incluyendo artículos de revistas, trabajos de congresos y tesis.	Los documentos publicados de manera redundante (seleccionar la versión más reciente o más completa).

Basados en estos criterios de inclusión y exclusión, hemos procedido a la selección de la literatura. La búsqueda inicial produjo 2.219 documentos, donde 30 provenían de CNKI, 15 de VIP, 32 de ScienceDirect, 32 de IEEE Xplore, 70 de ACM Digital Library y 2.040 de Google Académico. Tras leer los títulos y resúmenes, descartamos 2.131 documentos porque su tema no estaba relacionado, 38 no cumplían con el tipo de literatura esperado, y 13 eran sólo resúmenes de conferencias, dejando 37 documentos. Después de leer el contenido completo, eliminamos 15 documentos ajenos al objetivo del estudio, 3 por ser publicaciones duplicadas y 5 por no poderse acceder a su texto completo, llegando a incluir finalmente 14 documentos en nuestra investigación.

En el proceso de selección manual, constatamos que las bases de datos académicas de alta especialización ofrecen resultados más precisos aunque en menor cantidad, en comparación con los sitios web académicos integrales proporcionan una gran cantidad de recursos electrónicos poco relacionados, que requieren una selección manual reiterada.

2.4 Mapeo de datos

Durante la selección de materiales, también hemos realizado la extracción de información útil, que incluye: fecha de publicación, tipo de literatura, métodos de diseño y desarrollo del sistema, características funcionales del sistema, evaluación y mejoras, etc. Cabe señalar que, en algunos artículos, los autores no mencionan detalladamente todas las tecnologías y herramientas relacionadas con el desarrollo de software. Por lo tanto, hemos utilizado el código original que se escribe en papel, para inferir algunas de las tecnologías aplicadas desde una perspectiva práctica.

2.5 Recopilación, síntesis y comunicación de los resultados

2.5.1 Características técnicas fundamentales de SGB

Las características técnicas fundamentales de SGB provenientes de la literatura incluida son las siguientes:

Documento seleccionado	Patrón de diseño de software	Lenguaje de desarrollo	Sistema de bases de datos	Servidor web	IDE	Tecnologías relacionadas	Pruebas de rendimiento
Yang y Shi (2023)	B/S	JAVA	MariaDB	Tomcat	Spring Tools Suite 4	Java EE Web, CSS+JavaScript	✓
Luo y Zhu (2019)	B/S	ASP.net	SQL Server	-	Visual Studio	ADO.NET,ORM,JavaScrip	✓
Xian (2016)	B/S	Java	MySQL	Tomcat	-	JSP	✓
Jing (2015)	C/S	Java	MySQL	-	Myeclips	SSH	✓
Yao (2015)	B/S	Java	SQL Server	Tomcat	Myeclips	SSH,JSP, UML	✓
Chen (2014)	B/S	Java	MySQL	-	-	Java EE Web	✓
Li (2016)	B/S	ASP.net	SQL Server	-	-	ADO	✓
Zhang y Jiang (2016)	B/S	ASP.net	SQL Server	-	Visual Studio	Html+CSS+JavaScript	✓
Dai (2020)	B/S	-	MySQL	-	-	-	✓
Su et al. (2019)	B/S	Java	SQL Server	-	Myeclips	JSP	-
Wang (2010)	B/S	Java	-	-	-	SSH, JSP	-
Ning et al. (2008)	B/S	Java	-	-	-	JSP, Html+CSS+JavaScript	-
Granados y Millán (2012)	C/S	Java	PostgreSQL	-	NetBeans	MSF	✓
Gonzalez (2009)	C/S	PHP	PostgreSQL	Apache HTTP Server	Zend Studio	Symfony, UML, RUP	-

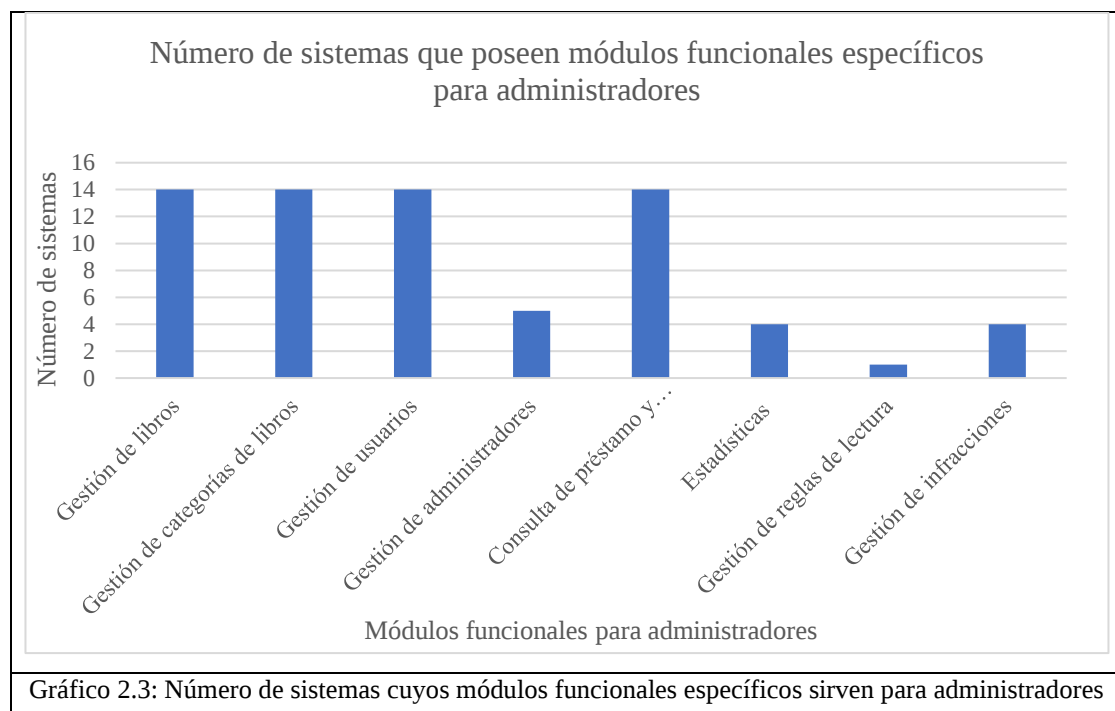
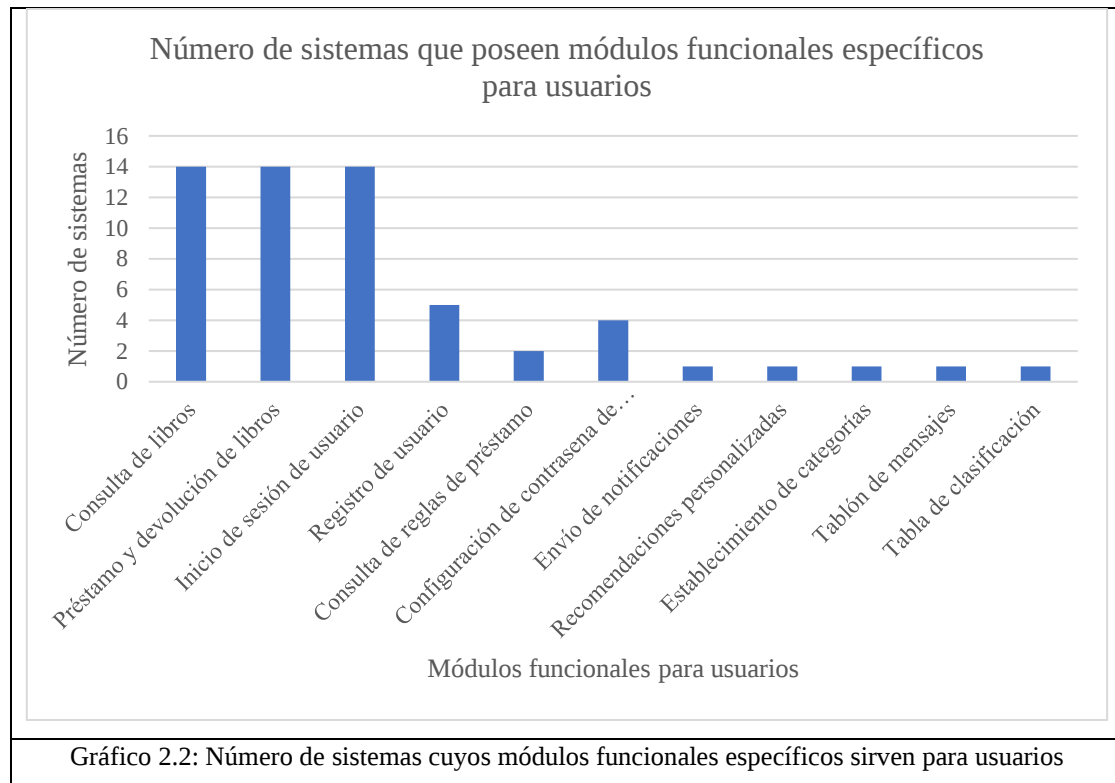
Gráfico 2.1: Características técnicas de los sistemas de gestión de bibliotecas

En la investigación realizada, se han considerado 14 documentos, de los cuales 11 SGB operan bajo el modelo de arquitectura de red B/S, mientras que los otros tres emplean el modelo C/S. Respecto a las herramientas utilizadas, 13 documentos especifican los lenguajes de desarrollo empleados, principalmente ASP.NET, PHP y Java. En cuanto a la elección del sistema de bases de datos, las principales opciones fueron SQL Server, MySQL, MariaDB (una variante de MySQL) y PostgreSQL. Cuatro de los estudios señalan que Tomcat y Apache pueden albergar sitios web y aplicaciones web. En relación a las herramientas de desarrollo de software o IDE (Entorno Integrado de Desarrollo), 8 documentos mencionan principalmente Spring Tools Suite 4, Visual Studio Myeclips, NetBeans y Zend Studio. Además de los principales elementos del desarrollo del sistema, también hay otras herramientas de front-end, back-end y bases de datos que se señalan claramente en el Gráfico 1. La mayoría de los documentos, 10 en total, hacen referencia detallada o breve a la existencia de pruebas del sistema, incluyendo los tipos de pruebas (por ejemplo, pruebas de rendimiento, de esfuerzo, funcionales) y las formas de realización de estas pruebas (como pruebas de caja negra, de caja blanca, unitarias, de integración y de sistema).

2.5.2 Módulos Funcionales en los SGB

Los SGB mencionados en la literatura revisada pueden ser estructurados principalmente en dos módulos: el módulo de usuario y el módulo de administrador.

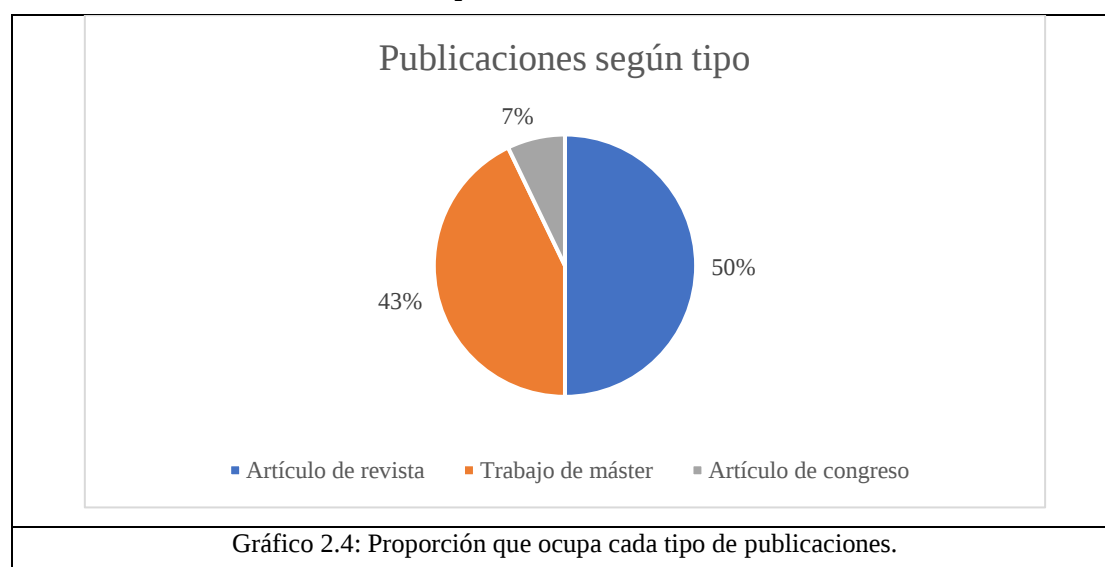
Las funcionalidades específicas por cada parte son las siguientes:



Al analizar los dos gráficos de barras, podemos llegar a las siguientes conclusiones. En primer lugar, las operaciones básicas que debe gestionar un sistema de gestión bibliotecaria son las siguientes: para los usuarios, son búsqueda de libros, préstamo y devolución de libros, y acceso al sistema; mientras que, para los administradores, hablan de gestión de libros, gestión de categorías de libros, gestión de usuarios, consultas de préstamos y devoluciones, etc. Además, en línea de las necesidades específicas de los participantes del sistema, así como el alcance técnico y de recursos disponibles para el desarrollo del software, también pueden incluirse funciones interactivas y de entretenimiento, tales como el envío de notificaciones, recomendaciones personalizadas, creación de categorías, tablón de anuncios, estadísticas, etc.

2.5.3 Discusiones

Una vez que hemos reunido y organizado el contenido de la literatura requerido, como se menciona por arriba, podemos aprovechar Excel para elaborar diversos gráficos que hagan visible la información y ofrecen una presentación más directa de los resultados. Además, mediante un análisis breve de los datos estadísticos, aseguramos que las preguntas de investigación abordadas han sido satisfactoriamente contestadas con todas las cinco etapas.



Respecto a la distribución por tipos de obras, observamos que tanto los artículos de revistas como los trabajos académicos ocupan conjuntamente un 93% del total, mientras que solo hay un artículo publicado en un congreso. En consecuencia, consideramos que la clara preponderancia cuantitativa de estos dos tipos de publicaciones revela la alta profesionalidad del contenido relacionado con el diseño y desarrollo de sistemas de gestión de bibliotecas, así como su enfoque en profundidad y sistematización de los estudios.



Aunque la cantidad final de literatura incluida es limitada y cubre el período 2009-2023, el gráfico de líneas no permite una descripción precisa de la tendencia del desarrollo de sistemas de gestión de bibliotecas basados en el modelo MVC. Sin embargo, aún se puede notar que desde 2009, este tema ha sido objeto de gran interés en el ámbito académico, con una actividad investigativa dinámica.

2.6 Resumen

Utilizando el método sistemático de revisión de alcance, hemos realizado una búsqueda, selección y síntesis exhaustiva de los SGB que emplean el patrón MVC. Al analizar y ordenar las evidencias existentes, hemos adquirido conocimiento sobre el pasado, el estado actual y las tendencias internacionales del campo, proporcionando además una sólida base para investigaciones posteriores.

3. Resumen de herramientas de desarrollo de sistemas y tecnologías relacionadas

3.1 Entorno de desarrollo IntelliJ IDEA

En la actual era de la información, los sistemas de gestión de bibliotecas se han convertido en un importante apoyo para la gestión moderna de libros. Después de un análisis exhaustivo de las necesidades y una comparación de las estrategias tecnológicas, hemos decidido adoptar la arquitectura MVC y utilizar IntelliJ IDEA como herramienta de desarrollo para el SGB, con Java y MySQL como lenguaje de desarrollo y tecnología de base de datos en segundo plano. Además de apoyar completamente la arquitectura MVC y la integración con bases de datos MySQL, IDEA ofrece una amplia gama de herramientas de diseño GUI, como Swing Designer, que ayudan a los desarrolladores a construir rápidamente interfaces gráficas de usuario amigables. En cuanto a la escritura de código, IDEA proporciona diversas herramientas de refactorización de código y también soporta múltiples métodos de depuración. IDEA se centra en la experiencia del usuario, proporcionando una interfaz operativa atractiva, sencilla y apoyada por numerosos accesos rápidos. Esto permite a los desarrolladores familiarizarse rápidamente con la herramienta y completar eficazmente las tareas de desarrollo. (JetBrains, 2023)

3.2 Introducción a tecnologías relacionadas

3.2.1 Comparación entre la arquitectura C/S y B/S

Modelo de arquitectura C/S (Cliente/Servidor):

La arquitectura C/S es un modelo que separa las funciones de una aplicación en dos partes distintas: el cliente y el servidor. En esta arquitectura, el cliente se encarga de ofrecer una interfaz amigable para el usuario y de gestionar la interacción con él, mientras que el servidor desempeña el papel de procesar datos y lógica de negocio. La

arquitectura C/S destaca por su elevado rendimiento, seguridad y experiencia de interacción del usuario, siendo especialmente adecuada para aplicaciones empresariales que requieren un elevado poder de cómputo y capacidad de procesamiento de datos, como sistemas de gestión de bases de datos.

Modelo de arquitectura B/S (Browser/Servidor):

La arquitectura B/S es un modelo de cómputo distribuido basado en tecnologías web, donde el cliente se conecta al servidor principalmente mediante un navegador web. En esta arquitectura, el servidor lleva a cabo el procesamiento de datos y la lógica de negocio, enviando los resultados al cliente en formato de páginas web. La arquitectura B/S se ha ganado gran popularidad por su capacidad de trabajar en múltiples plataformas, así como por su fácil despliegue y mantenimiento, siendo especialmente adecuada para aplicaciones web que necesitan un rápido desarrollo y despliegue, tales como sitios de comercio electrónico y sistemas de oficina en línea. El uso del navegador como único cliente simplifica en gran medida la complejidad y reduce el costo de mantenimiento en el lado del cliente.º

Comparación entre las arquitecturas C/S y B/S:

Analizando las características tecnológicas mencionadas de los sistemas de gestión de bibliotecas, se hace evidente que tanto la arquitectura C/S como la B/S pueden aplicarse exitosamente. Sin embargo, considerando las necesidades actuales y tendencias de desarrollo en los sistemas modernos de gestión de bibliotecas, la arquitectura B/S parece ser la más adecuada. Cabe señalar que la elección entre las dos arquitecturas para sistemas de gestión de bibliotecas debe realizarse teniendo en cuenta diversos factores. La decisión final dependerá específicamente de peticiones de biblioteca y de su alcance tecnológico.

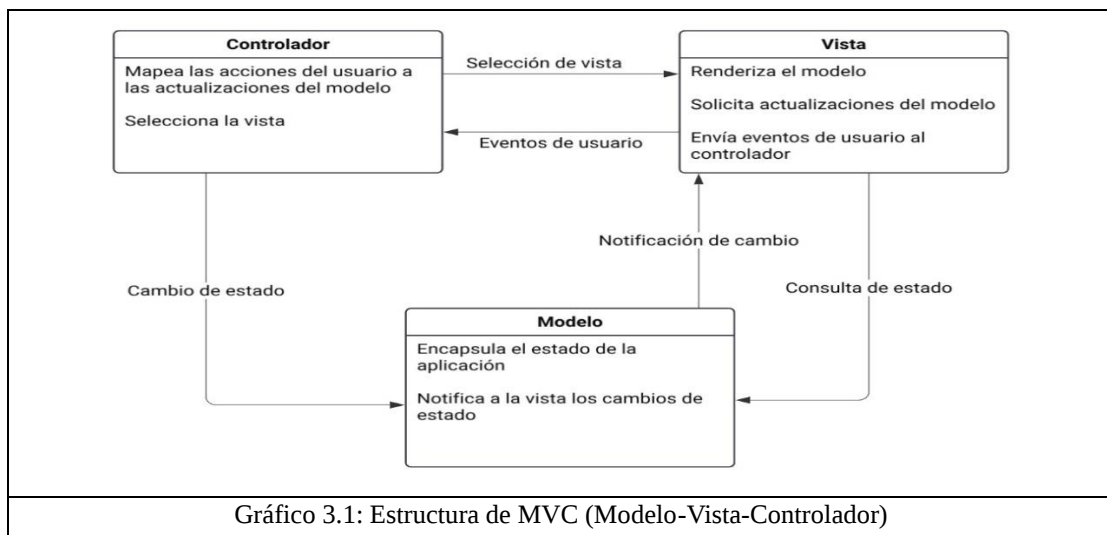
Para facilitar la comprensión, la Tabla 3.1 ofrece una detallada comparación de las ventajas y desventajas de ambas arquitecturas (Yao, 2016)."

Tabla 3.1: Comparativas de los modelos C/S y B/S		
Modelo	C/S	B/S
	Dado que el cliente soporta parte de la carga, el servidor de aplicaciones no necesita soportar todo el procesamiento de la lógica de negocio. Almacenamiento y gestión de datos sencillos.	El sistema requiere poco mantenimiento y es fácil de actualizar. Bajo coste.
Inconvenientes	Se necesitan diferentes versiones de software y los costes de desarrollo son relativamente altos.	Básicamente, el servidor lleva toda la lógica de negocio y está muy cargado.

3.2.2 El patrón MVC

MVC es un patrón de diseño para software que estructura las aplicaciones en tres partes clave: Modelo, Vista y Controlador. El Modelo encierra la fuente de datos y la lógica de negocio, permitiendo que múltiples Vistas puedan acceder a datos neutros sin necesidad de duplicar código. La Vista es responsable de mostrar los datos al usuario en una forma determinada y recibir las solicitudes de interacción. El Controlador, por su parte, gestiona la interacción del usuario con la aplicación, gestionando las solicitudes, ejecutando la lógica de negocio y actualizando la Vista según sea necesario. Este diseño garantiza que el Modelo puede alimentar múltiples Vistas, asegurando una rápida actualización de todas las Vistas afectadas cuando los datos cambian, mejorando la mantenibilidad y extensibilidad de las aplicaciones (Chen, 2014).

El mecanismo interno de MVC se ilustra gráficamente en el Gráfico 3.1.



En el marco del modelo MVC, el flujo de trabajo típico sigue los siguientes pasos:

Interacción del usuario:

El usuario realiza acciones como pulsar botones o introducir datos a través de la vista (View), estableciendo la interacción con la aplicación.

Solicitudes recibidas por el controlador:

Las acciones del usuario son interceptadas por el controlador, que se encargará de gestionarlas.

Actualización del modelo:

Según las solicitudes del usuario, el controlador invoca métodos del modelo, que pueden actualizar los datos de la aplicación o ejecutar otras lógicas de negocio.

Notificación y actualización de la vista:

Cuando el modelo experimenta cambios, notifica a la vista y la vista, a su vez, se actualiza utilizando los datos proporcionados por el modelo para representar la nueva interfaz de usuario.

Retroalimentación:

Finalmente, la vista actualizada se muestra al usuario, cerrando el ciclo de interacción.

3.2.3 Java y la Interfaz Gráfica de Usuario (GUI) de Java

Java

En nuestra revisión de literatura, hemos constatado que las principales lenguas de programación empleadas en sistemas son Java, PHP y ASP. Aunque cada lenguaje tiene sus propios campos de aplicación, Java se destaca especialmente en el desarrollo de sistemas de gestión de bibliotecas debido a sus características únicas.

Los programas Java pueden funcionar en diversos sistemas operativos gracias a la Máquina Virtual de Java (JVM). Como lenguaje de programación orientado a objetos, Java incorpora el concepto de clases y objetos, lo que la coloca en una posición ventajosa para manejar complejas lógicas y arquitecturas de sistemas. En un contexto donde la seguridad informática es cada vez más crítica, Java destaca por su

mecanismo de seguridad riguroso que protege eficazmente los programas de ataques maliciosos o virus. Además, Java ofrece un sólido apoyo a la programación multihilo, aprovechando mejor las ventajas de los procesadores multinúcleo y aumentando la eficiencia de ejecución de los programas, en comparación con PHP y ASP.

Java GUI

La GUI, que se refiere a la Interfaz Gráfica de Usuario, constituye uno de los principales medios de interacción entre el usuario y el programa. Java proporciona un conjunto completo de herramientas GUI que ayuda a los desarrolladores a crear aplicaciones visualmente atractivas y de fácil manejo. La programación de GUI es esencial para el desarrollo de una aplicación, dado que influye en cómo el usuario interactúa con el software, permitiéndole realizar operaciones mediante una interfaz gráfica intuitiva.

El núcleo de la programación de Java GUI es el manejo de eventos. En el desarrollo de GUI en Java, la vista representa la presentación de la interfaz de usuario, mientras que el manejo de eventos es el mecanismo central que responde a la interacción del usuario con estos componentes de la vista. La interacción del usuario con los componentes de la vista, como botones o cajas de texto, desencadena eventos que son capturados y gestionados por los correspondientes escuchadores de eventos. El proceso de manejo de eventos puede incluir la actualización del estado de los componentes de la vista, reflejando las acciones del usuario y permitiendo una interacción dinámica entre el usuario y la GUI.

3.2.4 Tecnología de bases de datos MySQL

La tecnología de bases de datos ocupa un lugar central en los sistemas informáticos, debido a su función clave de gestionar, organizar, integrar y procesar eficazmente datos de diversos tipos y estructuras (Yao, 2016).

En nuestra revisión de literatura, hemos encontrado que las bases de datos más utilizadas en sistemas de gestión de bibliotecas son MySQL, SQL Server, MariaDB y

PostgreSQL. Esto demuestra que la elección de una base de datos depende en muchos casos de los hábitos del usuario y de los requerimientos específicos del proyecto. Basándonos en nuestra experiencia y habilidades técnicas, hemos optado por utilizar el sistema de bases de datos MySQL, un sistema relacional de alto desempeño y sólido, que soporta múltiples motores de almacenamiento, tiene una curva de aprendizaje rápida, y además goza de un fuerte apoyo comunitario gratuitamente y de código abierto.

La conjunción del uso de MySQL con el lenguaje de programación Java ofrece una sólida capacidad de manejo de datos y un modo eficiente de interacción de datos. Los desarrolladores pueden conectarse y manipular fácilmente bases de datos MySQL mediante la API JDBC de Java y los frameworks ORM, realizando operaciones como agregar, eliminar, modificar y consultar datos. Esta combinación aumenta la eficiencia de desarrollo, asegurando la flexibilidad y escalabilidad de las aplicaciones, cumpliendo así con las necesidades tecnológicas del SGB.

3.3 Resumen

El presente capítulo destaca las tecnologías clave para el desarrollo del SGB. Primeramente, seleccionamos IntelliJ IDEA como herramienta de desarrollo, debido a su pleno apoyo para la arquitectura MVC y la integración con bases de datos MySQL, además de proporcionar una amplia gama de herramientas para diseño gráfico. En segundo lugar, analizamos las ventajas y desventajas de las arquitecturas cliente-servidor (C/S) y navegador-servidor (B/S). Posteriormente, detallamos el patrón de diseño MVC, destacando su importancia en mejorar la mantenibilidad y escalabilidad de las aplicaciones. Además, destacamos las ventajas de utilizar Java como lenguaje de desarrollo. Finalmente, señalamos la tecnología de bases de datos MySQL como la opción preferida para la base de datos del SGB. En general, este capítulo proporciona apoyo técnico integral y sugerencias de selección de opciones para el desarrollo del SGB.

4. Análisis de Necesidades

4.1 Estudio de Viabilidad

Antes de emprender cualquier proyecto, es necesario realizar un análisis de viabilidad para determinar si merece la pena llevarlo a cabo. Este estudio involucra examinar aspectos tales como la necesidad, la urgencia, la solidez científica del proyecto, y si su implementación dará los resultados esperados. El objetivo del análisis de viabilidad es determinar qué problemas pueden ser solucionados (Chen, 2014).

En el caso del SGB que estamos analizando, debemos considerar la viabilidad desde tres ángulos: técnico, económico y operativo.

4.1.1 Viabilidad tecnológica

La viabilidad tecnológica ocupa una puesta inicial en el desarrollo de este proyecto. Desde la perspectiva técnica, es necesario evaluar si la tecnología disponible permite llevar a cabo el desarrollo sin inconvenientes, analizando aspectos como la capacidad de los hardware y software para satisfacer las necesidades de uso.

Para este proyecto, se ha optado por desarrollar un sistema basado en el modelo C/S. Aunque la arquitectura B/S ha ganado popularidad en los últimos años, el modelo C/S ofrece ventajas significativas en seguridad, estabilidad y velocidad de respuesta. En particular, cuando se trata con grandes volúmenes de datos y complejas lógicas de negocio, el modelo C/S asegura una mayor estabilidad y eficiencia del sistema. En cuanto a la base de datos, MySQL será la opción elegida. Como un sistema gestor de bases de datos relacionales, MySQL ofrece características como alto rendimiento, escalabilidad y usabilidad, lo que permite cumplir con las necesidades de almacenamiento y consulta de datos del SGB. Con el rápido avance de la tecnología, las plataformas de hardware han mejorado tanto en confiabilidad como en relación calidad-precio, por lo que también pueden cumplir los requisitos del sistema.

4.1.2 Viabilidad económica

La viabilidad económica analiza principalmente la rentabilidad de la inversión y la relación costo-beneficio del proyecto. Aunque la inversión inicial en el lenguaje de desarrollo Java y la arquitectura de red C/S puede ser relativamente alta, dicha inversión es justificada dada su estabilidad a largo plazo, escalabilidad y el aumento en la eficiencia operativa y ahorro de costos que proporcionará. Además, el uso de MySQL como software de código abierto reduce los gastos asociados a la compra de software.

Adicionalmente, la aplicación del SGB reducirá los gastos relacionados con el personal, papel y otros recursos de la biblioteca, aumentando así la calidad y eficiencia de sus servicios. Por lo tanto, desde una perspectiva económica, este proyecto tiene una alta rentabilidad de la inversión y una sólida relación costo-beneficio.

4.1.3 Viabilidad operativa

La viabilidad operativa evalúa principalmente la facilidad de uso y la amigabilidad del sistema para el usuario. El diseño del sistema basado en el patrón MVC permite la separación entre la interfaz de usuario y la lógica de negocio, facilitando la adaptación y modificación de la interfaz. Las herramientas de diseño de interfaces gráficas de usuario (GUI) en Java ofrecen una amplia gama de elementos de interfaz y opciones de diseño flexibles para crear interfaces intuitivas y fáciles de utilizar.

El software cliente de la arquitectura C/S puede instalarse en las máquinas locales de los usuarios, proporcionando una experiencia de operación más fluida y estable. Además, mediante una formación adecuada y apoyo técnico, los usuarios podrán adquirir rápidamente las habilidades necesarias para utilizar el sistema.

4.2 Análisis de necesidades funcionales

El análisis funcional constituye tanto el núcleo como el punto de partida del diseño del sistema. El propósito de desarrollar un sistema en línea para la gestión y préstamo de libros es lograr una gestión sistemática e inteligente de la biblioteca, así como facilitar el préstamo y devolución de libros mediante la tecnología de red. Este sistema debe contemplar las necesidades tanto de los usuarios prestatarios como del personal administrativo encargado de la biblioteca. Desde el punto de vista de los roles, el sistema puede ser estructurado en dos módulos principales: uno para lectores comunes y otro para administradores.

4.2.1 Análisis de necesidades funcionales para administradores

En el módulo destinado a los administradores del sistema de gestión y préstamo de libros en línea, las principales funciones abarcan el registro, modificación, eliminación, clasificación, búsqueda y mantenimiento de libros. Una vez que el administrador se autentica en el sistema, podrá gestionar desde el backend la información detallada de los libros, incluyendo la posibilidad de consultar el estado de préstamos y reservas, verificar información sobre devoluciones y incumplimientos, gestionar las identidades y datos de los usuarios, así como consultar información relativa al sistema en general.

4.2.2 Análisis de las necesidades funcionales de los usuarios

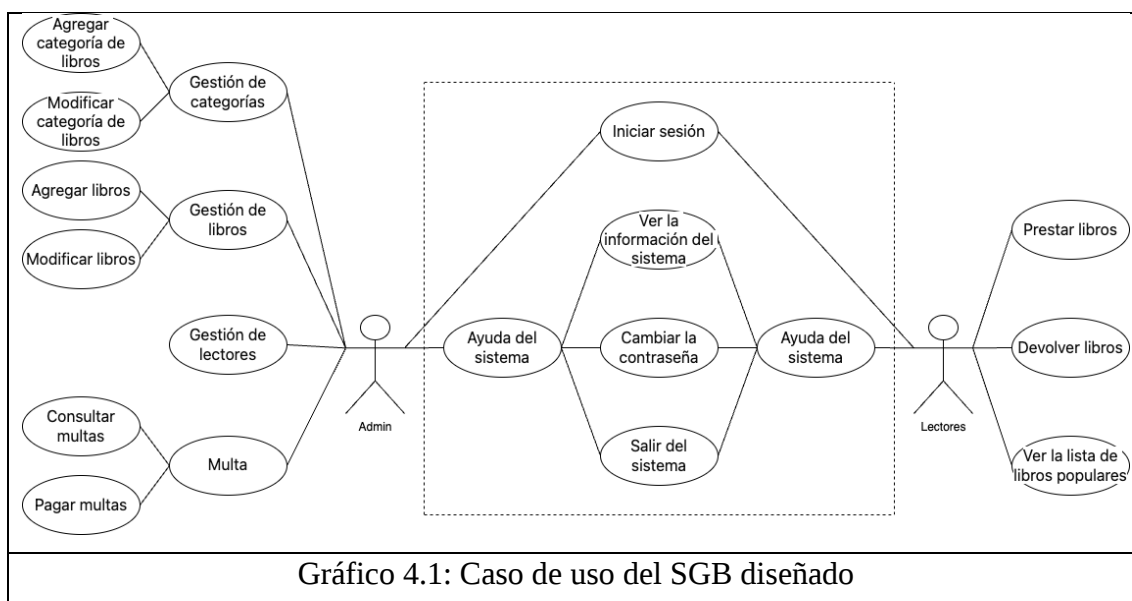
Los usuarios generales, que constituyen la mayor parte de la comunidad del sistema de préstamo de libros, deben poder registrarse y acceder a la plataforma mediante el inicio de sesión. Una vez dentro, pueden explorar fácilmente las categorías de libros, así como consultar las listas de libros más solicitados, las recomendaciones recientes y las incorporaciones más nuevas, proporcionando una visión rápida de la oferta disponible. Además, estos usuarios tendrán la capacidad de realizar búsquedas avanzadas para encontrar libros específicos, acceder a sus sinopsis

o resúmenes, y gestionar sus préstamos, devoluciones y prórrogas directamente desde la plataforma.

Un aspecto clave también incluye la posibilidad de que los usuarios puedan consultar sus propios registros de infracciones, permitiéndoles pagar las multas asociadas al momento de devolver los libros.

4.2.3 Presentación del Diagrama de Casos de Uso del Sistema

Los casos de uso se emplean para describir el comportamiento del sistema desde la perspectiva de los actores involucrados. Cada caso de uso representa una función completa y autónoma del sistema de gestión de libros. Por ejemplo, el rol de administrador interactúa con el sistema mediante casos de uso tales como inicio de sesión, búsqueda avanzada, gestión de lectores, administración de catálogo, así como el control de préstamos y devoluciones. Una vez definidos los actores y desarrollados los casos de uso, se procede a crear el Diagrama de Casos de Uso, que ofrece una visión clara y completa de las relaciones entre los diferentes componentes del sistema. Este diagrama, que sirve como una herramienta fundamental en el proceso de modelado del sistema, se muestra en el Gráfico 4.1.



4.3 Análisis de requerimientos no funcionales

Además de cumplir con las funcionalidades solicitadas, el sistema también debe satisfacer necesidades no funcionales que son características adicionales que complementan las funciones básicas. Estas necesidades no funcionales son cruciales porque influyen en la calidad y, en cierta medida, pueden afectar las funcionalidades requeridas. A continuación, analizaremos estas necesidades bajo los siguientes aspectos:

Confiabilidad. Dada la magnitud de datos que maneja el sistema de gestión de libros, es imperativo que los desarrolladores prevean cualquier posible conflicto desde el inicio, asegurando que las operaciones concurrentes no ocasionen problemas como inconsistencias de datos. Asimismo, el sistema debe ser capaz de detectar y manejar errores de operación por parte de los usuarios, garantizando así su confiabilidad.

Usabilidad. Debido a que no todos los usuarios del sistema son profesionales de informática, el diseño y operación deben tener en cuenta la realidad del uso. El sistema debe ser fácil de aprender y utilizar, acompañado de una amplia documentación y herramientas gráficas que apoyan a los usuarios en la navegación y uso del sistema.

Seguridad. El sistema debe manejar y almacenar una cantidad enorme de información, particularmente datos relacionados con usuarios y préstamos de lectores. La manipulación ilegal de estos datos podría ocasionar pérdidas significativas. Por esta razón, asegurar la integridad de los datos constituye un aspecto clave para elevar el nivel de seguridad del sistema.

Facilidad de mantenimiento. En el futuro, es probable que las bibliotecas aumentarán su tamaño y que se añadirán nuevas funcionalidades. Por lo tanto, es esencial que el sistema pueda actualizarse rápidamente para incorporar estas nuevas necesidades, garantizando la operación continua y estable del sistema.

Portabilidad. La portabilidad hace referencia a la capacidad de un software para funcionar correctamente cuando se transfiere de un entorno operativo a otro. La

independencia del sistema operativo es un rasgo destacado de la portabilidad de un software.

4.4 Resumen

En este capítulo se aborda el análisis de necesidades y la investigación de viabilidad del SGB. En primer lugar, mediante una profunda evaluación de la viabilidad técnica, económica y operativa, hemos asegurado que el proyecto es viable y cuenta con un alto valor de implementación. Luego, hemos analizado pormenorizadamente las principales necesidades funcionales del sistema, incluyendo las funciones esenciales de gestión de información de libros por parte de los bibliotecarios y las prácticas de préstamo y devolución de libros por parte de los usuarios habituales. Adicionalmente, mediante diagramas de casos de uso del sistema, hemos mostrado gráficamente la estructura funcional del sistema y las relaciones entre los participantes. Finalmente, hemos considerado cuidadosamente las necesidades no funcionales del sistema, tales como su confiabilidad, facilidad de uso, seguridad, capacidad de mantenimiento y portabilidad. Estos análisis servirán como guía clara para el diseño y desarrollo posterior del sistema.

5. Diseño General

El diseño general de este sistema tiene como objetivo proporcionar un software de administración eficaz y seguro para gestionar la información de las bibliotecas, cumpliendo con las necesidades de acceso y gestión de información de múltiples usuarios, mejorando así la eficiencia de la gestión de información en bibliotecas y logrando una excelente interacción humano-máquina (Yao, 2016). En función de las necesidades cotidianas de gestión de bibliotecas y los flujos de trabajo para el préstamo de libros, una vez implementado, el sistema debe cumplir con los siguientes objetivos:

Primero, ofrecer una interfaz de usuario atractiva y fácil de utilizar, así como garantizar la seguridad y la integridad de los datos almacenados.

Segundo, permitir una gestión eficiente y organizada de la información de libros y lectores.

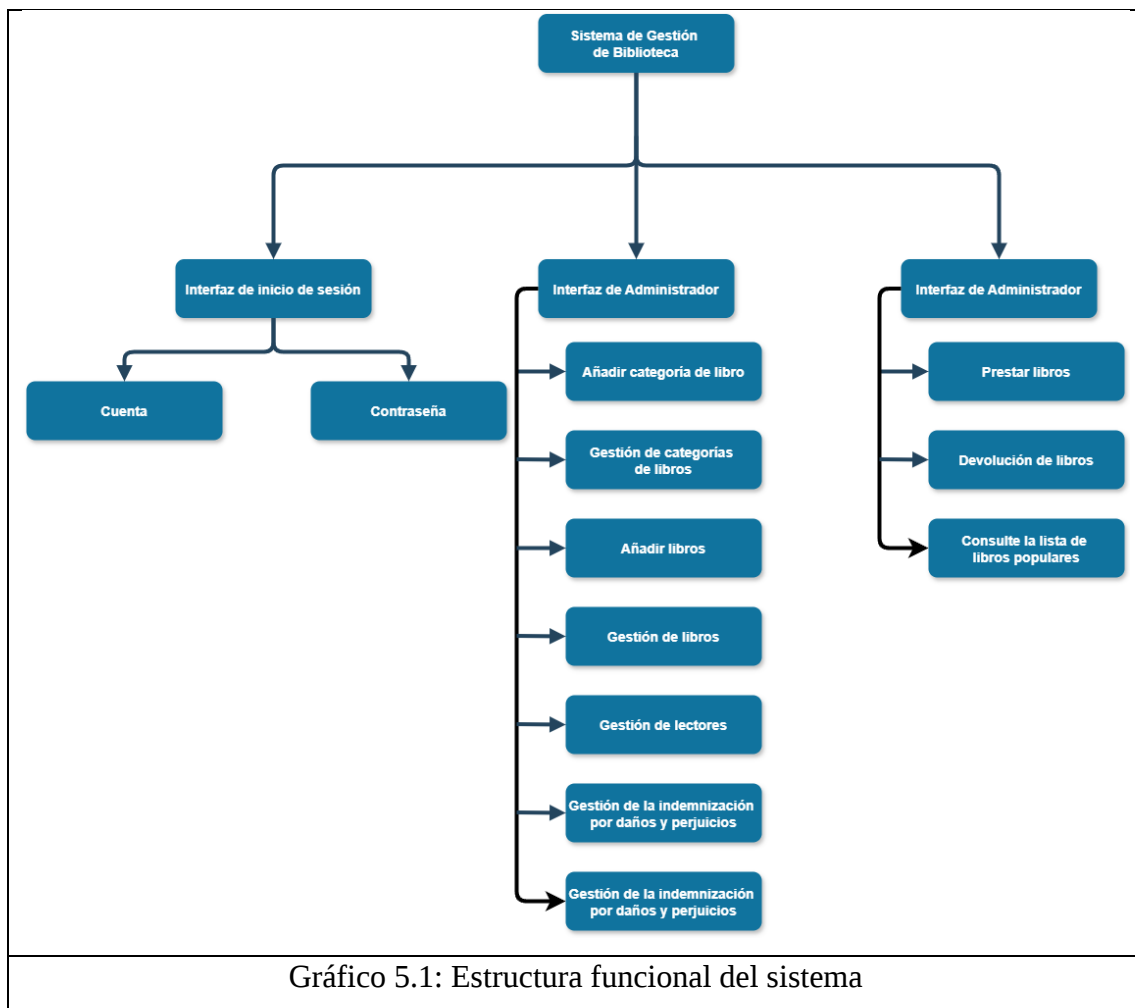
Tercero, contar con una potente función de búsqueda que permita consultar información de libros en línea y en tiempo real.

Cuarto, gestionar completamente el ciclo de préstamo, prórroga y devolución de libros, así como ofrecer una gestión transparente de los plazos de préstamo.

Quinto, permitir establecer listas de préstamos más populares, proporcionando información precisa para apoyar la gestión de bibliotecas.

5.1 Diseño Global del Sistema

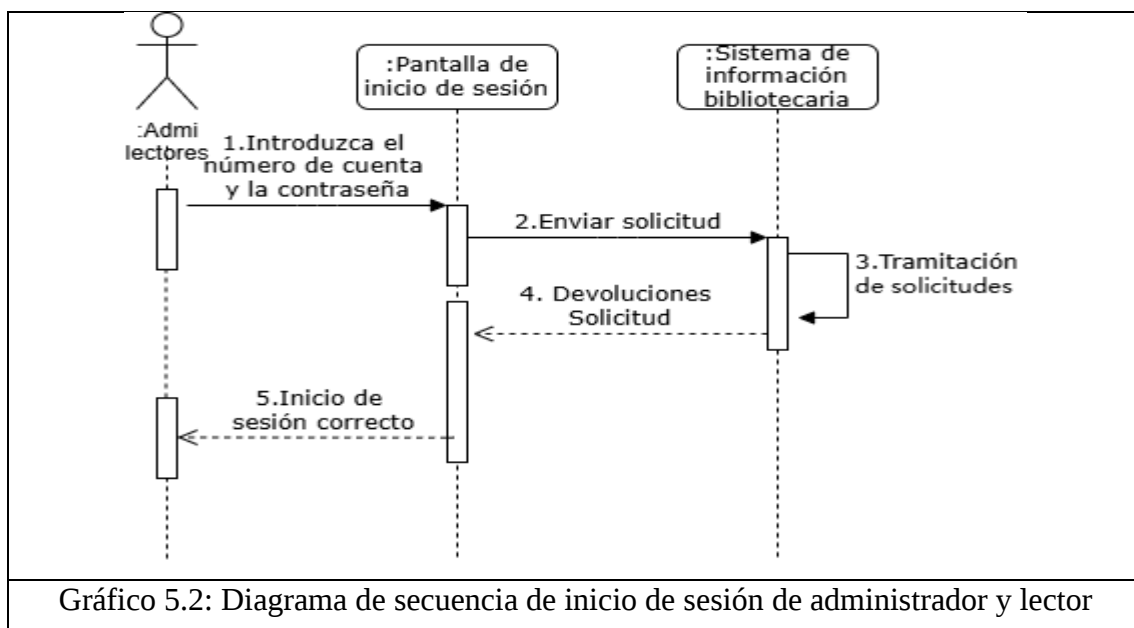
El modelo estructural funcional para la gestión de información de bibliotecas se ilustra en el Gráfico 5.1, incluyendo principalmente módulos tales como: el Módulo de Gestión de Lectores, el Módulo de Gestión de Libros, el Módulo de Consultas, el Módulo de Estadísticas y el Módulo de Gestión del Sistema.



5.2 Diseño Detallado del Sistema

5.2.1 Módulo de Inicio de Sesión

El principal propósito de la interfaz del módulo de autenticación es verificar la identidad y los privilegios del usuario, asegurando que solo los administradores o estudiantes puedan ingresar a los módulos correspondientes del SGB. En la pantalla de inicio de sesión, al introducir las condiciones pertinentes y realizar las correspondientes validaciones, se podrá confirmar la identidad y los derechos de acceso del usuario. Como ejemplo, consideremos el diagrama de flujo de casos de uso para el inicio de sesión de un administrador:

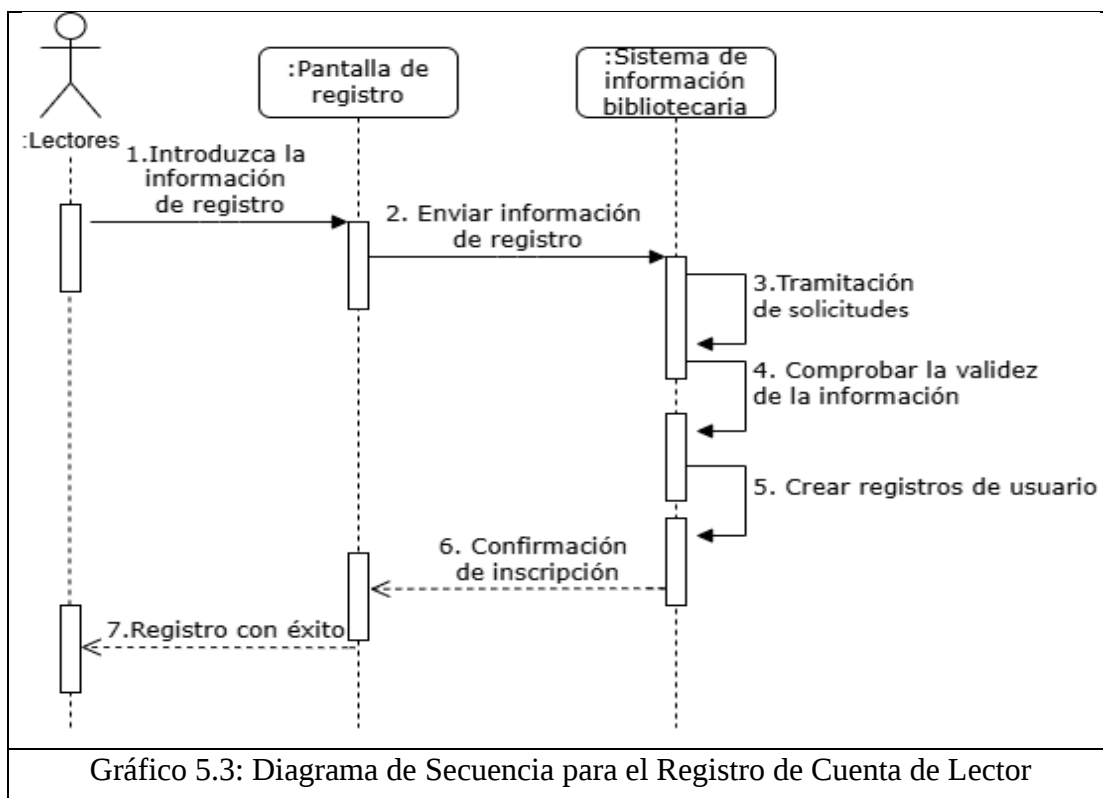


5.2.2 Diseño de Módulos Funcionales del Sistema

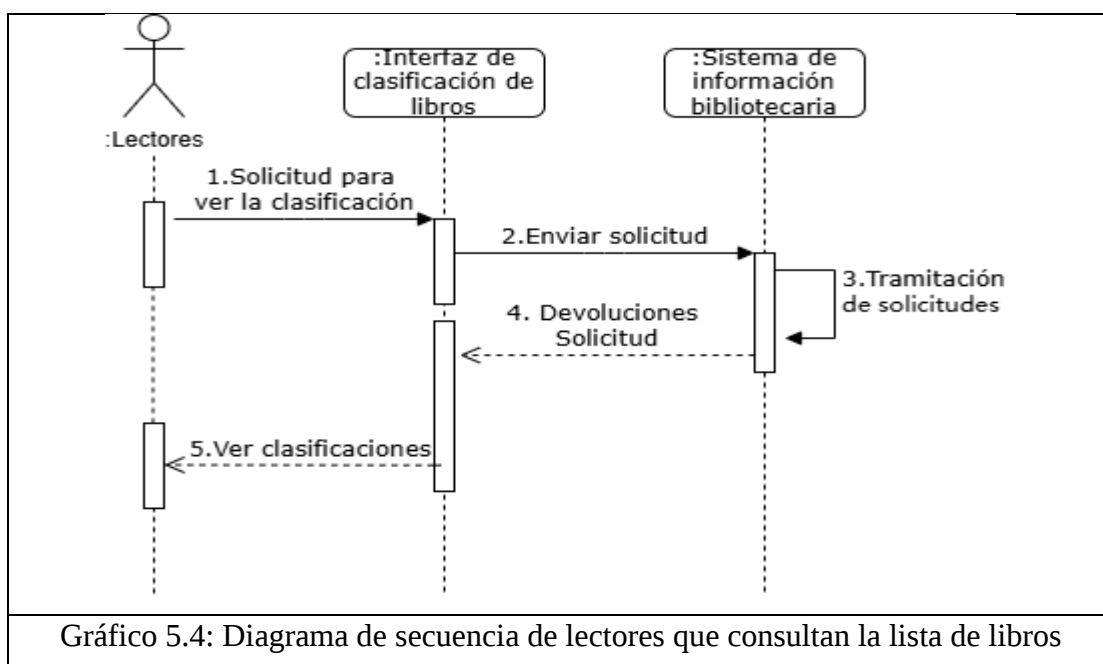
Módulo de Gestión de Lectores

El módulo de gestión de lectores tiene como función primordial la administración de la información relativa a los usuarios, incluyendo tanto la gestión y consulta de datos básicos personales como la gestión de las consultas sobre los préstamos de libros efectuados por los lectores. Los lectores pueden solicitar préstamos de libros mediante búsquedas por palabras clave.

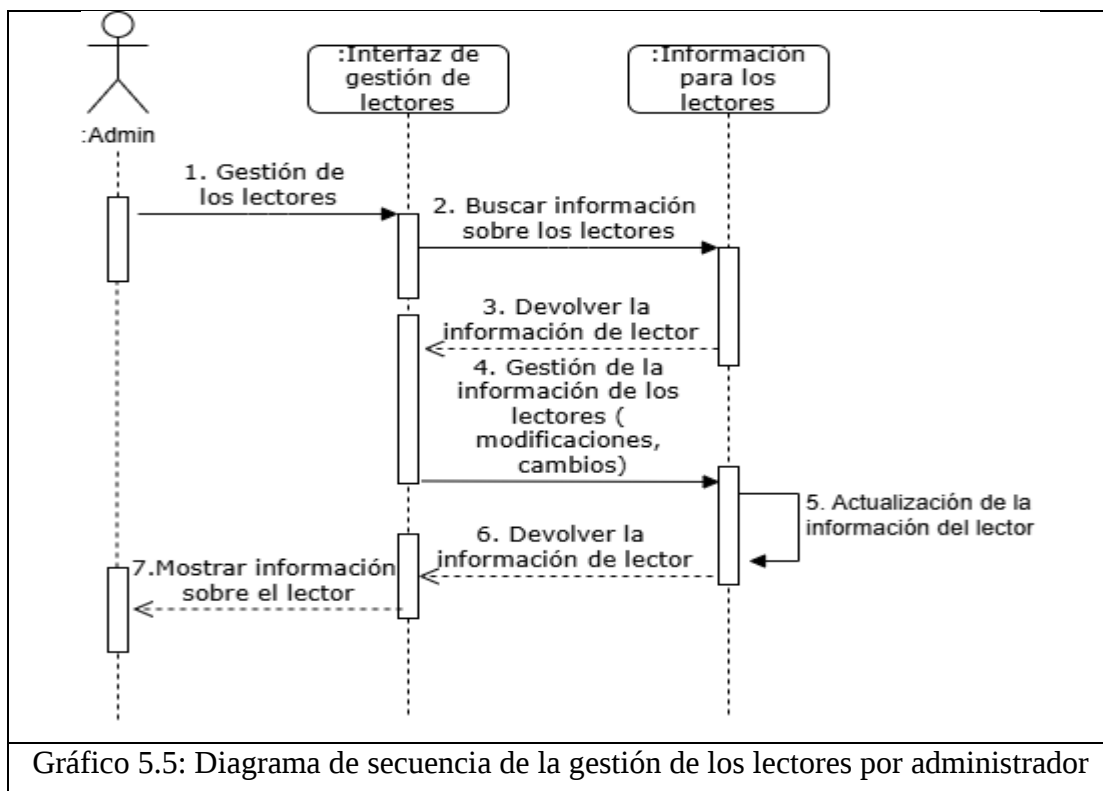
El diagrama de actividad de casos de uso para la gestión de lectores aparece en el Gráfico 5.3. En primer lugar, los lectores deben suministrar sus datos básicos personales. Posteriormente, el administrador del sistema verificará estos datos, asegurándose de su exhaustividad y precisión. En caso de que los datos no sean completos, el sistema notificará a los lectores que deben completarlos. Una vez que los lectores han proporcionado información completa, el sistema creará su perfil de usuario y almacenará estos datos en la base de datos, generando asimismo una tarjeta de préstamo.



El diagrama secuencial de casos de uso para la búsqueda de libros por lectores aparece en el Gráfico 5.4. Primero, los lectores deben iniciar sesión en el sistema y en la interfaz de búsqueda de libros ingresar los detalles del libro que desean consultar. Posteriormente, utilizando el número de libro como clave, el sistema busca en la base de datos y muestra la información correspondiente en la interfaz, permitiendo que los lectores puedan visualizar los detalles del libro.



Los lectores también pueden revisar su información personal una vez que han iniciado sesión en el sistema y seleccionado la interfaz de consulta de información personal. El sistema automáticamente busca y muestra la información personal relacionada con la cuenta de inicio de sesión del lector en la interfaz, lo que permite que los lectores puedan comprobar y visualizar sus datos personales. El diagrama secuencial para la búsqueda de información de lectores por parte de los administradores se encuentra en el Gráfico 5.5.

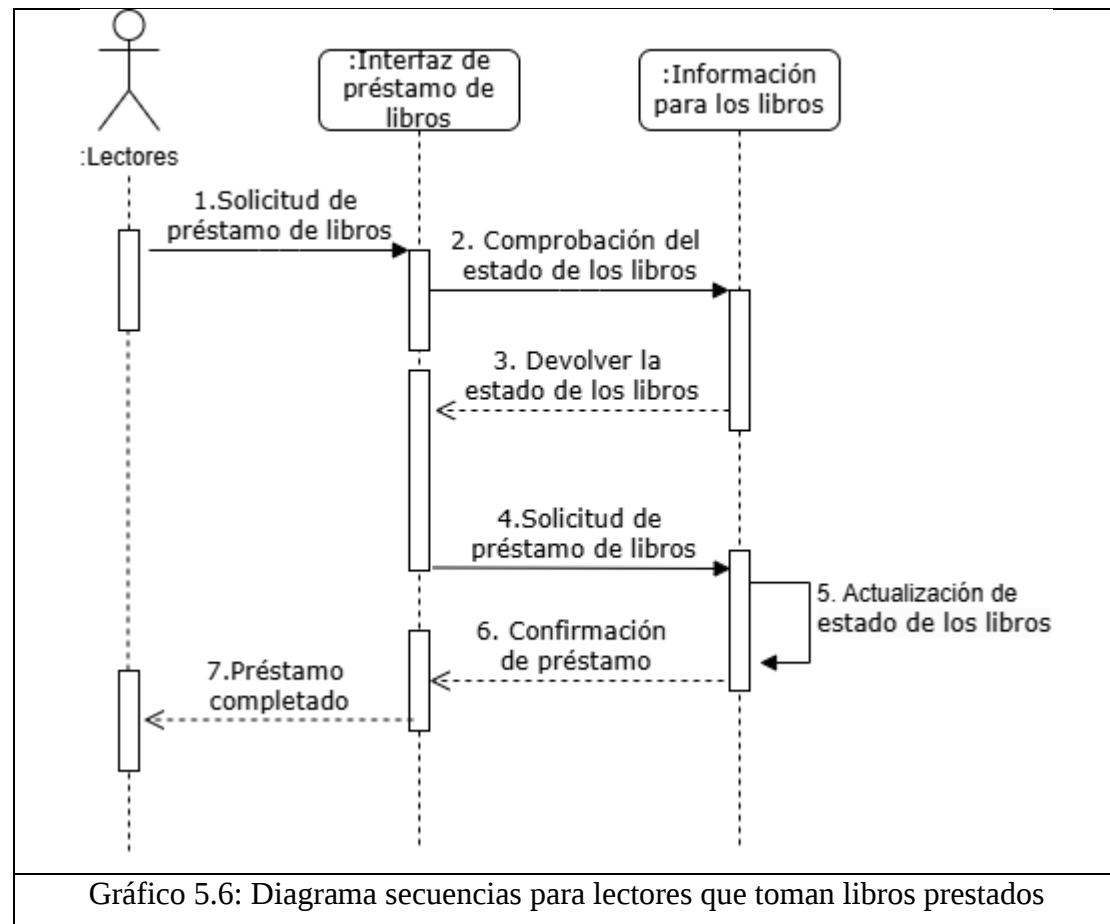


El Módulo de Gestión de Libros

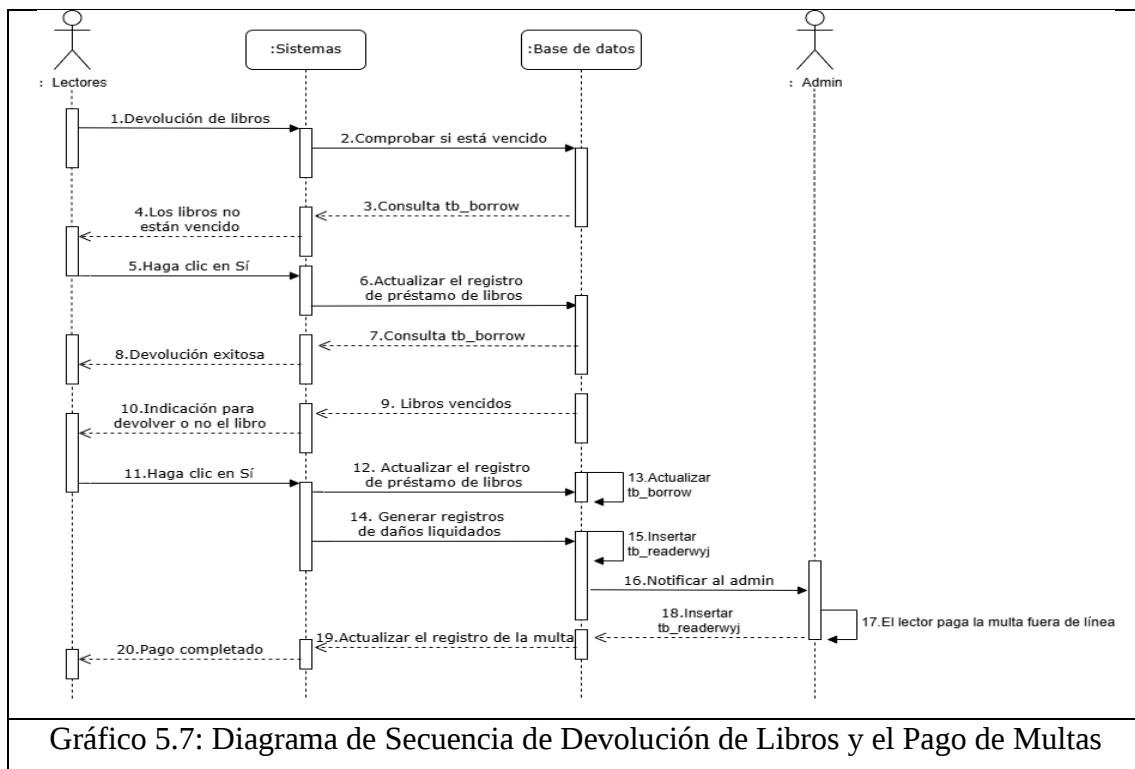
El módulo de gestión de libros comprende principalmente funciones tales como la administración de entrada de libros, la gestión de préstamos y la gestión de devoluciones.

El diagrama secuencial del caso de uso de préstamo de libros aparece en el Gráfico 5.6. Después de iniciar sesión en el sistema, el administrador de biblioteca consulta la información del lector en la función de préstamo, verificando si el lector tiene permiso para solicitar préstamos. Luego, se procede a revisar el historial de préstamos del lector, para determinar si es posible realizar un nuevo préstamo. Si así

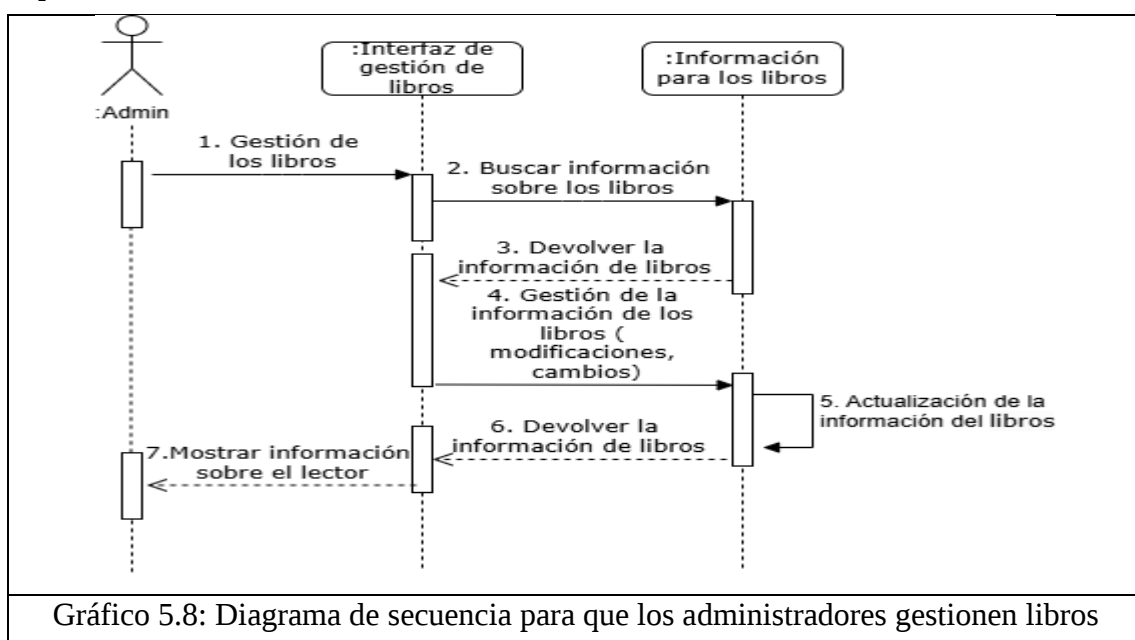
fuera, el sistema registra la información del libro solicitado por el lector, marca la operación como exitosa y muestra una notificación positiva en la interfaz.



El diagrama secuencial del caso de uso de devolución de libros aparece en el Gráfico 5.7. Primero, el administrador de biblioteca debe iniciar sesión en el sistema de gestión de libros. En la interfaz de devolución, ingresa el número de libro que debe ser devuelto. El sistema busca el registro de préstamo correspondiente, muestra la información en la interfaz y procede a la devolución del libro. Una vez realizada la devolución, el sistema actualiza automáticamente la información del libro, muestra una notificación de éxito en la interfaz y, finalmente, actualiza el historial de préstamos del lector.

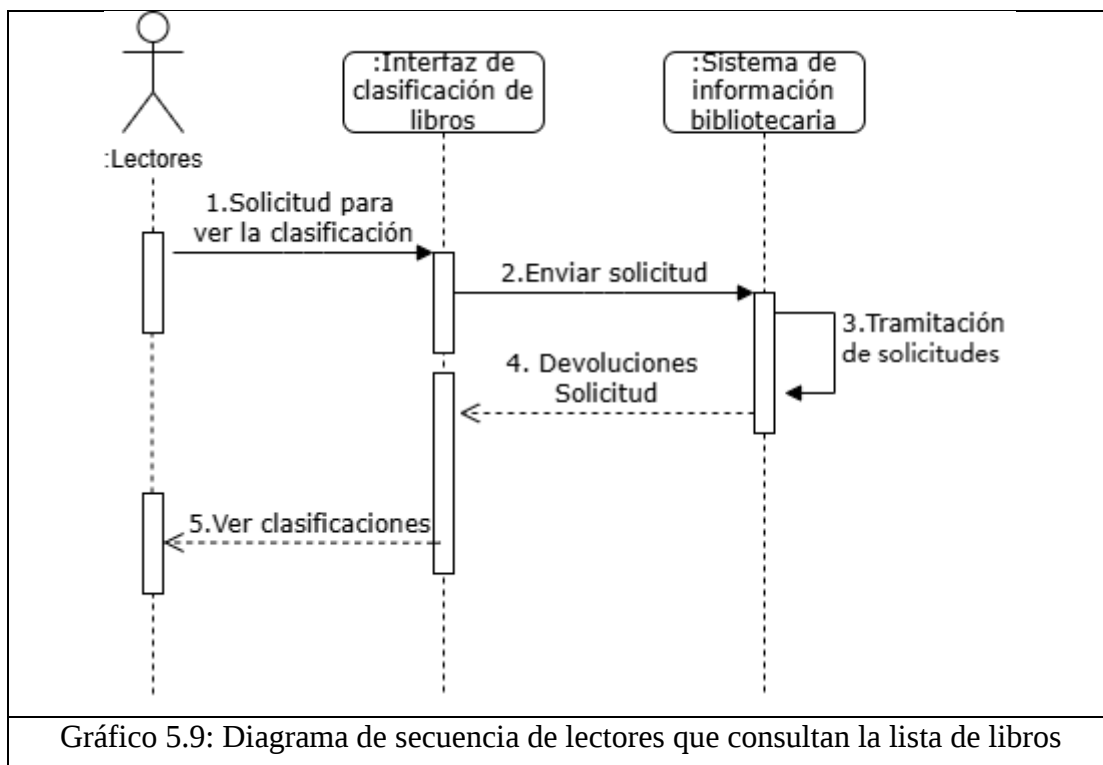


El Diagrama de Secuencia de Gestión de Libros presentado en el Gráfico 5.8 permite a los administradores ejecutar diversas operaciones tales como búsqueda, edición y borrado de libros. Tras iniciar sesión en el sistema, el administrador de biblioteca puede realizar operaciones de creación, eliminación, actualización y consulta sobre la información relacionada con los libros en el módulo de gestión. En caso de éxito, el sistema proporcionará una notificación positiva acerca de la operación realizada.



Módulo de Gestión de Búsquedas

El módulo de Gestión de Búsquedas se encarga principalmente de facilitar búsquedas sobre información almacenada en el sistema, incluyendo detalles de libros, datos básicos de lectores, registros de préstamos y devoluciones, así como información sobre la entrada de nuevos libros. Este módulo proporciona servicios auxiliares esenciales para el SGB, permitiendo que lectores y administradores realicen búsquedas mediante palabras clave o combinaciones de datos pertinentes. El Diagrama de Actividad de Búsquedas es ilustrado en el Gráfico 5.9.



Módulo de Administración de Multas de los Lectores

La administración de las multas por parte de los administradores de biblioteca comprende tanto la creación como la gestión de la información de estas. En presencia de infracciones tales como retrasos en las devoluciones de libros o daños causados a ellos, el sistema genera automáticamente información sobre las multas. Además, los administradores deben inspeccionar a los lectores y, en caso de infracciones graves, pueden aumentar la multa dentro de los límites establecidos. Lectores que acumulan varias multas pueden recibir una advertencia o incluso sufrir la congelación de su

cuenta.

Módulo de Administración del Sistema

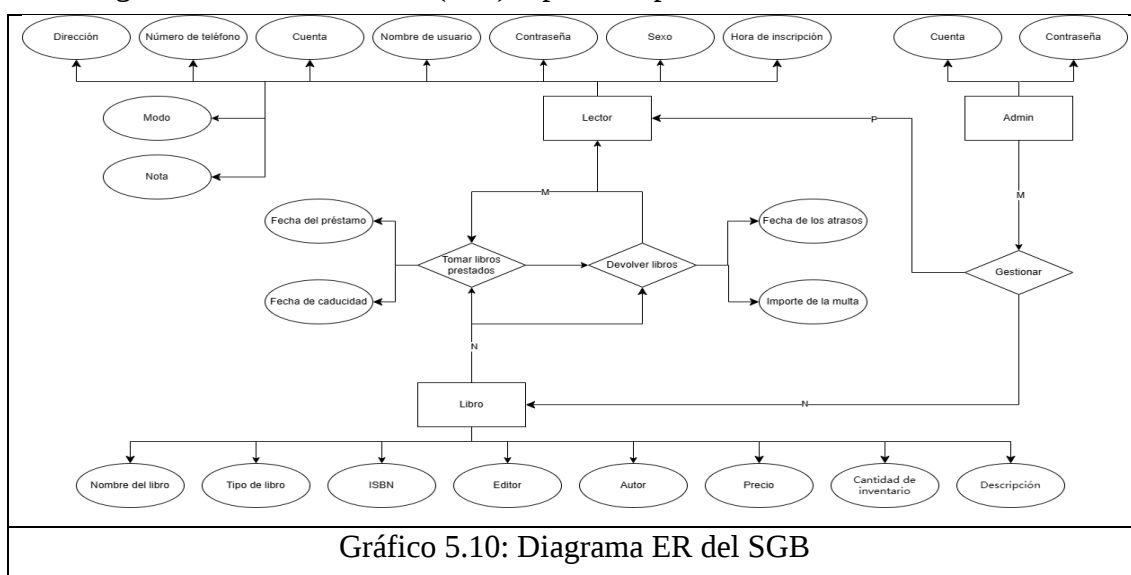
La función de administración del sistema incluye tanto la gestión de las normas de la biblioteca como la gestión de los registros del sistema. La gestión de las normas registra las políticas vigentes para la administración de libros y las normas de indemnización. Mientras tanto, la gestión de los registros del sistema rastrea todas las acciones realizadas en el sistema durante su operación. En caso de fallas, estos registros sirven como base para que el administrador del sistema pueda diagnosticar y solucionar los problemas.

5.2.3 Diseño de la Base de Datos del Sistema

Diseño Conceptual de la Base de Datos

El diseño conceptual de la base de datos ocupa un lugar clave en la creación de base de datos. A partir de las necesidades especificadas para el SGB, se desarrolla un modelo que contempla tanto las entidades propias de administradores y lectores como las relaciones lógicas entre ellas, siguiendo los principios de ingeniería de software. El Diagrama Entidad-Relación (E-R) constituye una herramienta esencial para el diseño de este modelo conceptual, representando entidades, relaciones y atributos (Li, 2016).

El análisis detallado de las relaciones entre entidades ha llevado a la construcción del Diagrama Entidad-Relación (E-R) específico para este sistema.



Diseño del Modelo Lógico

Para interactuar con la base de datos, el sistema emplea MySQL como sistema relacional de bases de datos, mediante una capa de acceso a datos (DAO), que vamos a explicarlo en la parte de realización de la base de datos. En el diseño de la estructura lógica, se ha considerado tanto la optimización de las consultas y modificaciones como la minimización de la redundancia entre las tablas. En este trabajo, se ha adoptado una convención de nomenclatura específica para el diseño de tablas, que incluye las siguientes entidades: información de libros, información de lectores, registros de préstamos y devoluciones, categorías de libros, y detalles de administradores. El diseño lógico de estas entidades principales se describe en detalle a continuación:

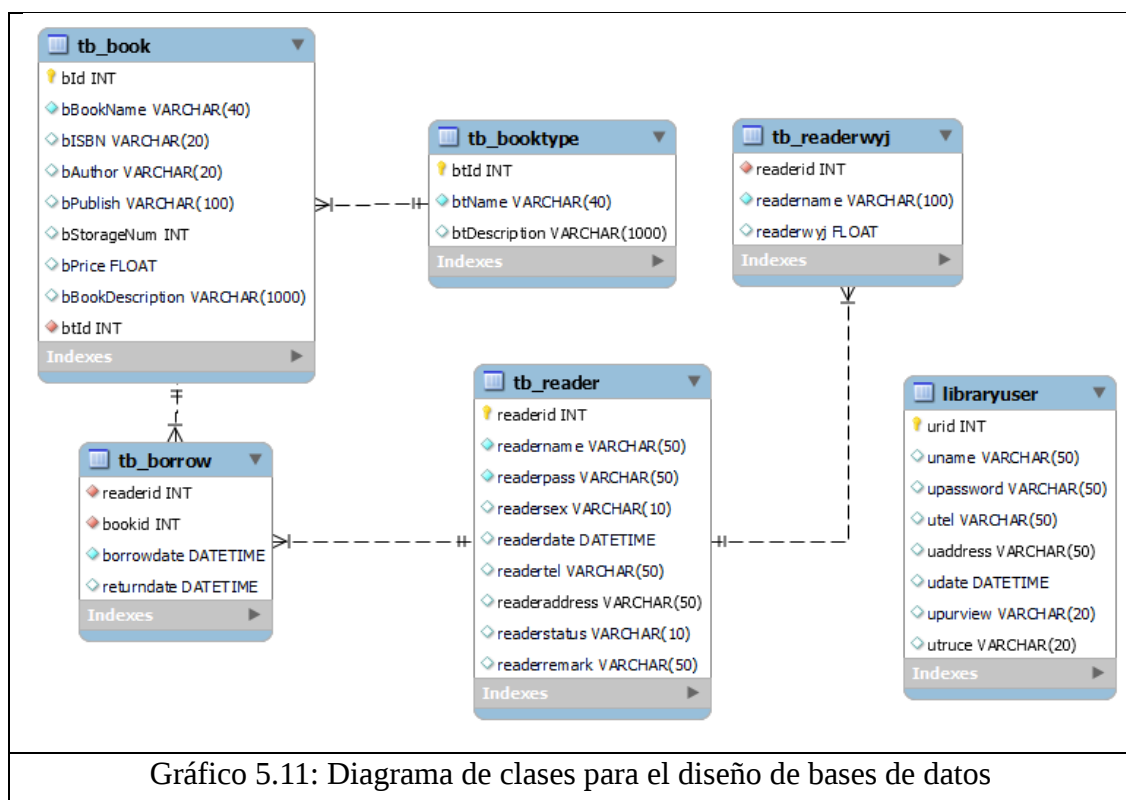


Gráfico 5.11: Diagrama de clases para el diseño de bases de datos

El Diagrama de clases ha sido diseñado para representar claramente las interacciones y asociaciones entre las distintas tablas del sistema, así como las características propias de cada una de ellas. Estas asociaciones son las siguientes:

tb_book está relacionado con tb_booktype por el campo btd, que indica la relación entre el libro y su tipo.

tb_book está relacionado con tb_borrow por el campo bid, que indica qué libros se han prestado.

tb_reader está asociado a tb_borrow por el campo readerid, que indica qué lectores han tomado prestado el libro.

tb_reader está asociado a tb_readerwyj mediante el campo readerid, que indica qué lectores se han requerido pago de multa.

5.3 Resumen

En esta sección se proporciona una visión panorámica del diseño integral del sistema de gestión de información de bibliotecas, que busca alcanzar la gestión eficiente y segura de la información de múltiples usuarios. El sistema cuenta con módulos dedicados a la gestión de lectores, libros, consultas, estadísticas y administración, entre otros, diseñados para satisfacer las necesidades cotidianas de las bibliotecas. Adicionalmente, destacamos la importancia del proceso de inicio de sesión y el diseño cuidadoso de la base de datos para garantizar la integralidad del sistema y la seguridad de los datos almacenados.

6. Implementación del Sistema y Resultados

6.1 Estructura de Archivos del Sistema

A continuación, se muestra la estructura del proyecto Java dentro del IDE, que proporciona una visión detallada del diseño de archivos y carpetas del SGB diseñado en esta investigación. La distribución de estos archivos y carpetas es descrita en la Tabla 6.1:

Tabla 6.1: Disposición de los archivos del SGB en IDEA	
Menú	Descripción
/BooksManagementSystem	Directorio base del sistema
/BooksManagementSystem/.idea	Contiene información de configuración del IDE
/BooksManagementSystem/bin	Archivos binarios compilados ejecutables
/BooksManagementSystem/lib	Archivos de biblioteca o dependencias adicionales
/BooksManagementSystem/META-INF	Metadatos centrales del proyecto
/BooksManagementSystem/src	Archivos fuente de código
.../src/Dao.ebook.com	Interfaces de acceso a base de datos
.../src/images	Archivos gráficos del proyecto
.../src/model.ebook.com	Clases de modelo
.../src/views.ebook.com	Interfaces de la capa de vista
.../src/tools	Clases de utilidad

6.2 Acceso a Datos

En el modelo MVC para arquitecturas C/S, la capa de DAO (Objeto de Acceso de Datos) desempeña un papel clave en encapsular y desacoplar las operaciones de acceso a la base de datos. En este tipo de arquitectura, donde el cliente y el servidor comunican directamente, el acceso a datos es esencial. La capa DAO ofrece una interfaz uniforme para las operaciones básicas de la base de datos, tales como creación, lectura, actualización y eliminación, permitiendo que las capas superiores se enfocen únicamente en la lógica de negocio. En el marco MVC, la capa DAO, como parte de la capa Modelo, interactúa con la capa de negocio mediante mecanismos como inyección de dependencias, fomentando una alta cohesión y baja dependencia

en las operaciones de acceso a datos, lo que mejora la reutilización y mantenibilidad del código.

6.2.1 Implementación del DAO

El patrón DAO proporciona una estructura clara para la interacción con la base de datos, separando las tareas de acceso a datos de la lógica de negocio. En nuestro sistema desarrollado en Java, hemos utilizado JDBC para acceder directamente a la base de datos. La estructura de archivos y sus funciones se detallan en la siguiente tabla 6.2:

Tabla 6.2: Estructura de Archivos DAO	
Carpeta bajo Dao.ebook.com	Descripción
BookDao.java:	Manejan las operaciones relacionadas con libros
BooktypeDao.java:	Gestionan las operaciones relacionadas con los tipos de libros.
BorrowDao.java:	Realiza las tareas relacionadas con los registros de préstamos.
CxwyjDao.java:	Encargada de las consultas y cálculos de multas.
RankingDao.java:	Gestiona las operaciones relacionadas con las listas de ranking.
ReaderDao.java:	Maneja las tareas relacionadas con los lectores.
ReaderwyjDao.java:	Realiza operaciones con respecto a las multas.
UserDao.java:	Encargada de las operaciones de administración

En el caso de BookDAO.java, esta clase principalmente implementa operaciones sobre la información de libros almacenada en una base de datos. A continuación, demostraremos y explicaremos algunos fragmentos de código.

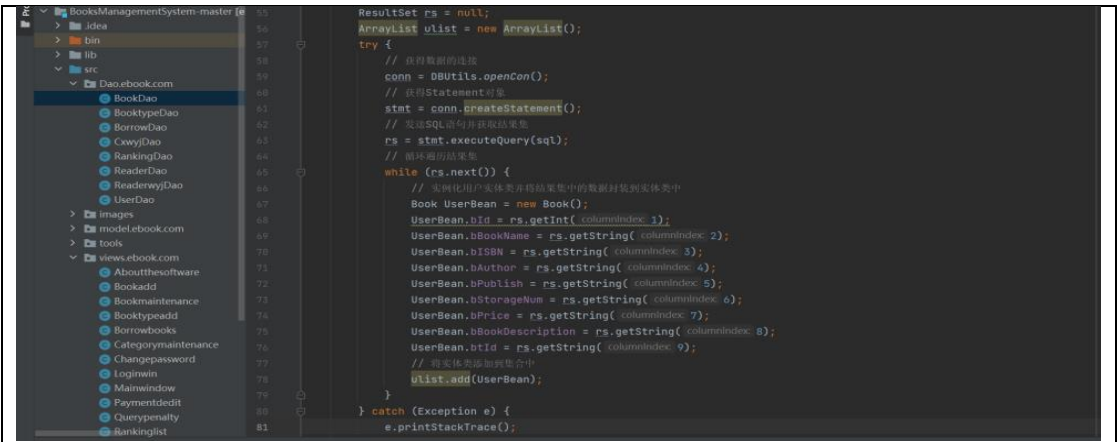


Gráfico 6.1: Consulta de la base de datos y envoltura del código

Seleccionamos parte del código para explicar a continuación:

```
private ConnDB conn = new ConnDB(); // Inicialización de la conexión a la base de datos
```

```
List<Booktype> bookTypeList = new ArrayList<>(); // Almacenamiento de la lista de objetos Booktype
```

```
stmt = conn.createStatement(); // Ejecución de una consulta SQL y obtención de un conjunto de resultados
```

```
bookType.setBtName(rs.getString("btName"));
```

```
bookType.setBtDescription(rs.getString("btDescription")); // Adición de objetos Booktype a la lista
```

6.3 Diseño e implementación de la interfaz de inicio de sesión

La interfaz de inicio de sesión es el punto de entrada principal para cualquier usuario que desee acceder al sistema. A través de esta interfaz, el usuario proporciona su nombre de usuario y contraseña para iniciar la sesión. Una vez que el sistema ha validado correctamente estas credenciales, procede a verificar la identidad del usuario y determinar el nivel de acceso que tendrá. Una vez completada la autenticación y la autorización, el usuario podrá acceder a las funcionalidades del sistema que corresponden a su perfil. De esta manera, el sistema garantiza la seguridad y la integridad de los datos, al tiempo que ofrece una experiencia de usuario adecuada para cada tipo de perfil.

El presente trabajo desarrolla un sistema de gestión de biblioteca que incorpora un proceso de inicio de sesión específico para administradores y estudiantes. En primer lugar, el usuario debe seleccionar la interfaz adecuada según su identidad (administrador o estudiante). Posteriormente, mediante la introducción del nombre de usuario y contraseña correspondientes, el sistema valida el inicio de sesión. La principal tarea de la interfaz de inicio de sesión es verificar la identidad y los privilegios del usuario, garantizando que solo los administradores o estudiantes

pueden acceder a los módulos pertinentes del sistema. La validación de los credenciales permite determinar los permisos de uso del usuario. La ilustración gráfica de la interfaz de inicio de sesión se presenta en el Gráfico 6.2.

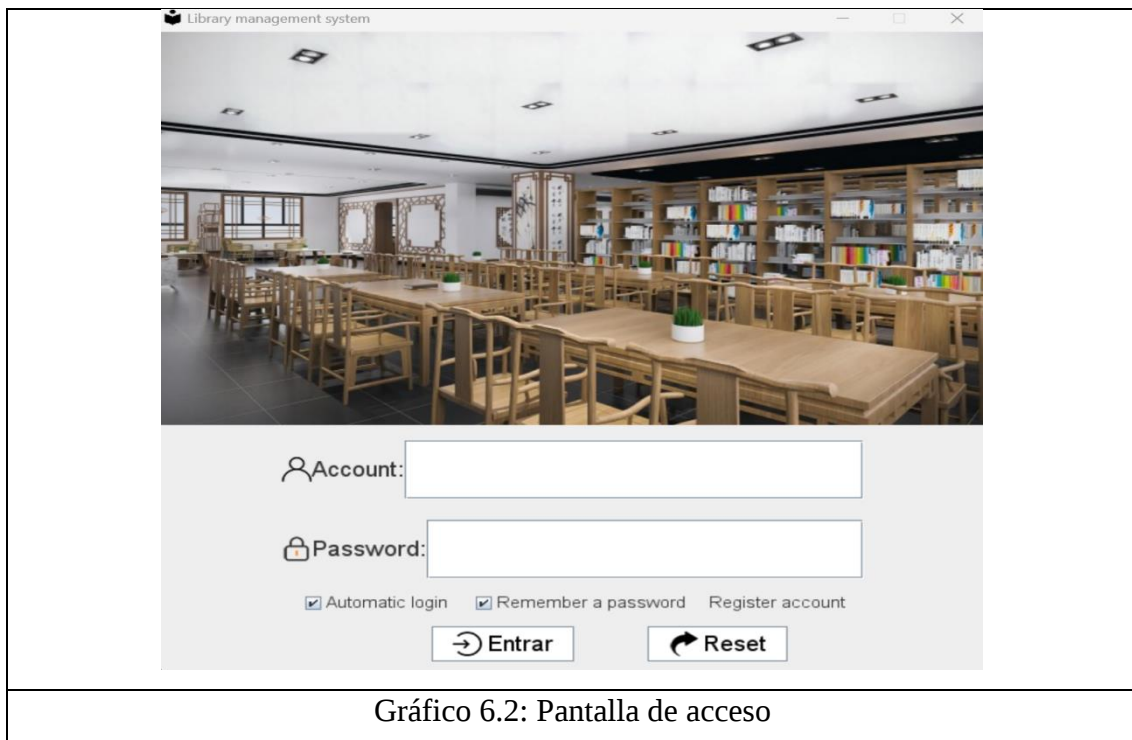


Gráfico 6.2: Pantalla de acceso

A continuación, se muestra parte del código que implementa la interfaz de inicio de sesión:

```
private JCheckBox jc1;  
public static String damin = ""; // Cuenta  
private ReaderDao readerdao = new ReaderDao();  
public static String readna = ""; // Nombre de usuario  
jc1 = new JCheckBox("Automatic login", true); // Iniciar sesión automáticamente  
jb21 = new JLabel("Register account"); // Registrarse nuevamente
```

6.4 Realización de Módulos Funcionales

6.4.1 Implementación del Módulo de Gestión de Libros

Como un componente de mayor importancia del SGB desarrollado en este trabajo, Este módulo abarca diversas funciones tales como la búsqueda, préstamo, reserva y devolución de libros, así como también la incorporación, eliminación y

actualización de información de los libros. En relación a la administración de libros, diferentes roles tienen distintos niveles de acceso y funcionalidad; el usuario lector solo puede realizar acciones como búsqueda, préstamo y devolución, mientras que el administrador tiene acceso completo a todas las funciones relacionadas con la administración de libros.

A partir del diseño detallado en el capítulo 4, la administración de libros incluye módulos como la gestión de información básica de libros, búsqueda de información, circulación de libros y gestión general de información. En particular, considerando el módulo de devolución de libros, se ofrecerá una descripción detallada de su implementación técnica y las operaciones que los usuarios pueden realizar:

Library management system Login Time:2024-05-15 18:08:35 Logout:Qigang Peng
Borrowing and returning books Book Ranking System Help

Readers return books

Title of the book: Set Reset

Number	Book name	Date of borrowing	Date of return of books
39	Modern Art: Impressions & Techn...	2024-05-15 18:08:55	2024-05-16 18:08:55

Number: Book name: Date of borrowing:

Book return date: Current time: Payment of liquidated damages

Return books Reset

Gráfico 6.3: Interfaz de devolución de libros

Cuando el lector se encuentra en la interfaz de devolución de libros, puede llevar a cabo las siguientes acciones:

1. Puede acceder a la interfaz haciendo clic en la barra de menú superior
2. Seleccione o infórmese sobre los libros que desea devolver
3. Pulse para volver
4. Devolución correcta si no se ha retrasado
5. Si está vencido se mostrará el importe de la tasa por impago que hay que pagar, haga clic en devolver el libro, el administrador sabrá cuánto hay que cobrar de tasa por impago y hay que pagarlo en la oficina del administrador
6. Si la cuota por defecto no se paga a tiempo, la cuenta del lector será desactivada por el administrador

7. Haga clic en Reiniciar para volver a seleccionar el libro que desea devolver

El siguiente fragmento de código representa la implementación para devolver libros:

```
JOptionPane.showMessageDialog(null, "¡Ningún libro seleccionado para devolver!", "Mensaje", JOptionPane.WARNING_MESSAGE);
```

```
int h = JOptionPane.showConfirmDialog(null, "¿Desea confirmar la devolución de este libro?", "Mensaje", JOptionPane.YES_OPTION);
```

```
JOptionPane.showMessageDialog(null, "¡Devolución realizada exitosamente!", "Mensaje", JOptionPane.INFORMATION_MESSAGE);
```

```
JOptionPane.showMessageDialog(null, "¡Fallo en la devolución!", "Mensaje", JOptionPane.ERROR_MESSAGE);
```

The screenshot shows a web application window titled "Payment of penalty". At the top, there are navigation tabs: "Category management", "Book management", "Reader management", "Penalty", and "System help". Below the tabs, there are input fields for "Reader ID:" and "Reader name:", followed by "Search" and "Reset" buttons. A table displays a list of readers with their details and penalties. The table has columns: Reader ID, Reader name, Sex, Telephone number, Address, and Penalty. The data rows are:

Reader ID	Reader name	Sex	Telephone number	Address	Penalty
1	pengqigeng	Male	17610073395	Rambla de Rrim 15...	12
100010016	xxx	Male	659234605	2401 Sunset Boule...	8
100010015	Manping Wang	Female	17610073395	4527 Oak Valley La...	2

Below the table, there is a form with the following fields: "Reader ID:" (value: 100010015), "Reader name:" (value: Manping Wang), "Sex:" (radio buttons for Male and Female, with Male selected), "Telephone number:" (value: 17610073395), "Address:" (value: illey Lane, Boston, MA 02124), and "Penalty:" (value: 2). At the bottom of the form are "Pay" and "Reset" buttons.

Gráfico 6.4: Liquidación de multa

En caso de que el libro se devuelve fuera del plazo establecido y debe pagarse una multa, cuando el lector llega a la biblioteca, el personal puede utilizar la interfaz administrativa para identificar y seleccionar a los lectores que tienen atrasos en las multas, llevando a cabo el proceso de recuperación del libro y pago de la multa correspondiente. La interfaz para consultar estas infracciones se muestra en la imagen de arriba.

6.5 Pruebas

La meta de las pruebas de software es verificar la calidad del producto, asegurando que no solo cumple con las expectativas preestablecidas sino que también funciona de manera correcta. Las pruebas deben incluir no solo el producto software sino también todo el ciclo de desarrollo. Los problemas que puedan surgir durante el uso, tales como problemas de memoria, interfaz o funcionalidad, pueden ser atribuidos a la falta de pruebas exhaustivas durante el desarrollo. Por ello, las pruebas de software tienen como objetivo garantizar un proceso de desarrollo de alta calidad y baja tasa de errores, así como ofrecer a los usuarios una experiencia óptima en un sistema que cuenta con funciones plenamente desarrolladas (Xian, 2018).

En este lugar, se hace hincapié en la prueba de funcionalidad del sistema de gestión de libros diseñado, para confirmar que el software cumple con las funciones esperadas. Nuestro plan es utilizar casos de prueba intuitivos y sencillos para llevarlas a cabo.

6.5.1 Diseño de casos de prueba del sistema

Pruebas de inicio de sesión del sistema

Los casos de prueba del módulo de inicio de sesión del usuario del sistema de gestión de libros se utilizan para verificar el módulo de inicio de sesión del sistema. Las pruebas de inicio de sesión se muestran en la Tabla 6.3. El Gráfico 6.5 es una captura de pantalla del interfaz de inicio de sesión del sistema.

Tabla 6.3: Casos de prueba del módulo de inicio de sesión			
Caso	Acción de prueba	Resultado esperado	Resultado obtenido
1	Ingresar manualmente el nombre de usuario "admi" y la contraseña "admi"	Mensaje de inicio de sesión exitoso	Aprobado
2	Ingresar manualmente el nombre de usuario "admi" sin contraseña	Mensaje indicando que el nombre de usuario y/o contraseña deben ser	Aprobado

		completados	
3	Ingresar manualmente el nombre de usuario "admi" y la contraseña "1234"	Mensaje indicando que el nombre de usuario o la contraseña son incorrectos	Aprobado



Pruebas del módulo de administración de libros

La integración de nuevos libros en el sistema bibliotecario constituye una tarea esencial para el mantenimiento de la base de datos. Durante el proceso de incorporación, el sistema debe verificar que la información proporcionada por el usuario es válida y completa. En caso de falta de datos o errores de entrada, el sistema deberá proporcionar una indicación adecuada. Los datos principales que deben ser introducidos incluyen el título del libro, la editorial y la categoría. Sin una información completa, la incorporación no podrá ser efectuada. En la Tabla 6.4 se detallan las pruebas diseñadas para evaluar el módulo de administración de libros, incluyendo cuatro casos de prueba, todos aprobados. El Gráfico 6.6 muestra una vista del módulo en funcionamiento, mientras que el Gráfico 6.7 muestra una captura de pantalla de la información detallada de un libro.

Tabla 6.4: Casos de prueba del módulo de administración de libros			
Caso	Acción de prueba	Resultado esperado	Resultado obtenido
1	No se ingresan datos	Se espera un mensaje de error	Aprobado
2	No se ingresa el título del libro	Se espera un mensaje de error	Aprobado
3	No se ingresa la editorial	Se espera un mensaje de error	Aprobado
4	No se ingresa la categoría del libro	Se espera un mensaje de error	Aprobado

Library management system · Hora de inicio de sesión: 15/05/2024 17:49:01 · Log in / profile / admin

Category management · Book management · Reader management · Penalty · System help

Add book

ISBN of book:

Publisher:

Author of the book:

Book price:

Book category:

Title of the book:

Number of additions:

Book description:

⊕ Add

⊗ Reset

Gráfico 6.6: Página de añadir libros

Library management system · Hora de inicio de sesión: 15/05/2024 18:03:51 · Log in / profile / admin

Category management · Book management · Reader management · Penalty · System help

Book maintenance

Title of book:

Author of book:

Book category:

Search

Reset

Number	Book name	Book ISBN	Book author	Publisher	Quantity in st...	Book prices	Book descripti...	Book category
36	Shadows ...	978-1451...	Emily Clar...	Gateway ...	12	22.0	A contem...	Literature ...
37	History of t...	978-03409...	Michael J...	University ...	40	30.0	This book ...	Non-Fiction
38	Economic...	978-04522...	Robert Ha...	Academic...	10	30.0	This book ...	Education ...
39	Modern Ar...	978-03754...	Lisa Young	ArtHouse ...	5	50.0	This book ...	Arts & De...
40	Introductio...	978-0262...	David Knox	TechMedia	15	22.0	This book ...	Arts & De...

Number:

Book name:

Book ISBN:

Publisher:

Prices:

Quantity in stock:

Book author:

Book category:

Descriptions:

✎ Revise

🗑 Delete

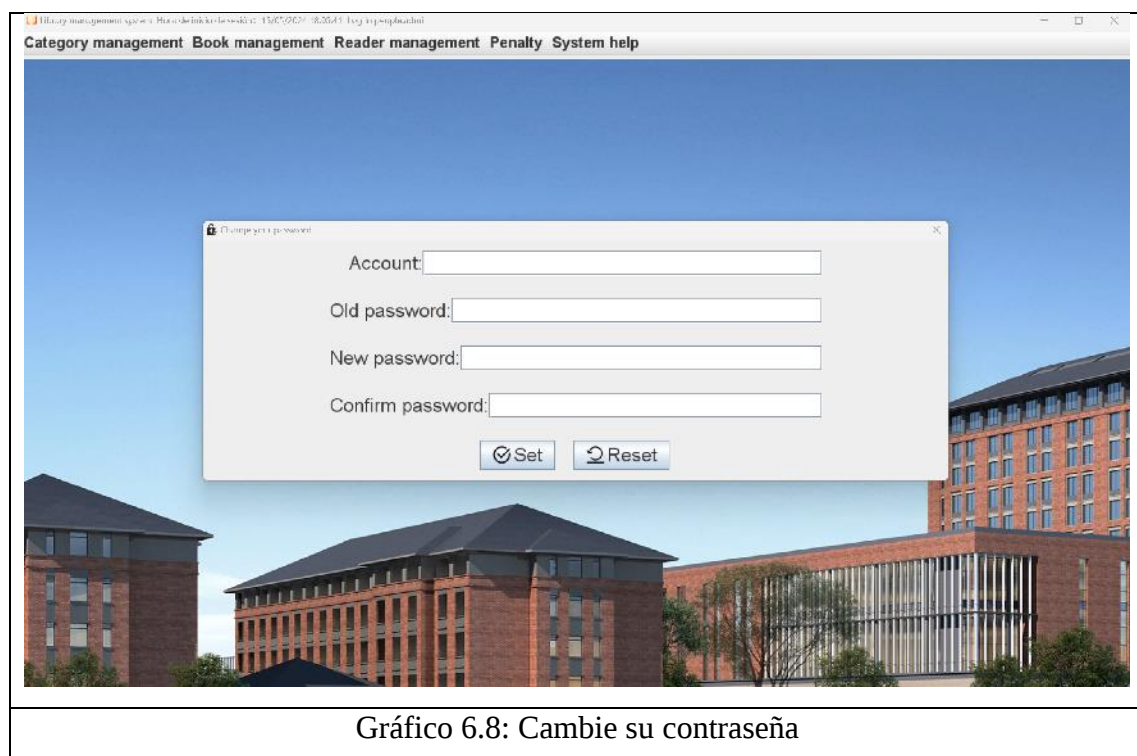
↺ Reset

Gráfico 6.7: Página de mantenimiento de libros

Pruebas de gestión de datos personales

El SGB permite a los usuarios, mediante el módulo de gestión de datos personales, actualizar su código de acceso. En caso de ingresar una contraseña anterior incorrecta, el proceso de actualización de la nueva contraseña será bloqueado. Sin embargo, si la contraseña proporcionada es válida, el sistema llevará a cabo automáticamente las verificaciones necesarias y actualizará la contraseña del usuario. Los casos de prueba diseñados para este proceso se detallan en la Tabla 6.5, donde cuatro casos fueron ejecutados y aprobados.

Tabla 6.5: Casos de prueba del módulo de gestión de datos personales			
Caso	Descripción de acción	Resultado esperado	Resultado obtenido
1	Ingresa contraseña correcta y nueva contraseña	Actualización exitosa	Aprobado
2	Ingresa contraseña correcta, pero sin nueva contraseña	Solicitud de ingreso de nueva contraseña	Aprobado
3	No ingresar contraseña correcta	Solicitud de ingreso de contraseña anterior, bloqueo de nueva contraseña	Aprobado
4	Ingresa contraseña incorrecta y nueva contraseña	Mensaje de error por cuenta o contraseña incorrecta	Aprobado



El SGB ha sido sometido a una serie de pruebas rigurosas, resultando en un estado de operación estable y funcionalidad básica adecuada.

6.6 Resumen

En este capítulo, hemos proporcionado una visión detallada de la implementación y resultados de pruebas del SGB. Se ha mostrado el diseño estructural del proyecto Java, destacando la organización de archivos y el flujo de trabajo. Asimismo, hemos profundizado en la implementación del acceso a la base de datos, haciendo énfasis en el papel de la capa DAO en el modelo MVC dentro de la arquitectura C/S. Se han descrito las características principales de la interfaz de inicio de sesión y el módulo de gestión de libros, destacando las diferencias en permisos según el rol de los usuarios. Finalmente, hemos destacado la importancia de las pruebas de software y presentado los casos de prueba diseñados para el sistema, proporcionando información valiosa para el desarrollo y optimización posteriores.

7. Conclusión y Perspectivas

7.1 Conclusión

A través de la detallada exposición y desarrollo presentado en este trabajo, hemos diseñado y desarrollado exitosamente un SGB basado en la arquitectura C/S. Este sistema toma como punto de partida las necesidades específicas de las actividades de gestión de bibliotecas, mejorando significativamente la eficiencia del trabajo mediante medios automatizados y digitalizados, alcanzando los estándares básicos requeridos por las bibliotecas modernas.

Este sistema tiene las siguientes ventajas:

Eficiencia y automatización: El sistema permite automatizar diversas tareas de gestión de bibliotecas, incluyendo la administración de datos de usuarios, la gestión de préstamos y devoluciones, la gestión de libros y de multas. Esto ha aumentado considerablemente la eficiencia en las tareas diarias.

Extensibilidad: La utilización de la estructura MVC permite separar el código, facilitando la futura ampliación y mantenimiento del sistema.

Facilidad de uso: El diseño intuitivo de la interfaz y la rápida y sencilla secuencia de operaciones hacen que el sistema sea fácil de usar, reduciendo el coste de aprendizaje para los usuarios.

Bajo coste: El sistema tiene un bajo coste de mantenimiento y operación, ahorrando costes operativos para las bibliotecas, comparado con sistemas tradicionales.

Sin embargo, en este sistema hay mucho por mejorar en el futuro. Primero, son limitaciones en las pruebas. Aunque hemos utilizado métodos de prueba funcional para optimizar el sistema, las técnicas empleadas fueron relativamente sencillas. Se requiere una prueba más exhaustiva para garantizar la estabilidad y confiabilidad del sistema.

Segundo, no incluye gestión de libros electrónicos. Con el aumento de la

demanda de libros electrónicos, el sistema aún no incluye esta funcionalidad, que será necesario incorporar en futuras versiones.

7.2 Perspectivas

Con respecto a las tendencias futuras del desarrollo en este SGB, tenemos las siguientes expectativas:

1. Gestión de libros electrónicos: Los sistemas futuros deberán integrar funciones de gestión de libros electrónicos, permitiendo la gestión unificada de libros físicos y electrónicos. Pueden que, por ejemplo, añadan los módulos funcionales como la compra de los libros electrónicos, así como su descarga.

2. Inteligencia y automatización avanzadas: Los sistemas incluirán funciones como búsquedas de texto completo, búsquedas inteligentes y personalización de servicios de información, satisfaciendo las necesidades más diversas de los usuarios.

3. Descubrimiento y transmisión de información: Los sistemas ofrecerán herramientas más avanzadas para ayudar a los usuarios a obtener la información requerida de manera más eficiente.

4. Actualización y optimización tecnológicas: Continuaremos actualizando y optimizando el sistema para garantizar su estabilidad y confiabilidad, mejorando simultáneamente la experiencia de los usuarios.

Bibliografía

Arksey, H., & O'Malley, L. (2005). Scoping studies: towards a methodological framework.

International Journal of Social Research Methodology, 8(1), 19–32.

<https://doi.org/10.1080/1364557032000119616>

Chen, L. F. (2014). Investigación y realización de un SGB basado en MVC (Tesis de maestría, Universidad de Yunnan).

https://kns.cnki.net/kcms2/article/abstract?v=mjz80qGfPOWA80KILELlExn82SXVPe5fKpiQ5RiW0n1ei4DTJdlD87rP1ZvN8yEzBRngtyKqAR6dxfHcm6FzKFn_ke8dtIgTcJZt5wU42epysUATfApKQFNdVHs25UCv63J7L9Z4_E0=&uniplatform=NZKPT&language=CHS

Chowdhury, F. Q. (2012). Database management systems and use of digital resources in some selected public university libraries of Bangladesh: An Overview. *Bangladesh Journal of Library & Information Science*, 2(1), 67–78.

<https://doi.org/10.3329/bjlis.v2i1.12923>

Dai, L. (2020). Análisis y diseño de un SGB basado en el patrón MVC. *Estilo Tecnológico*, 26, 101-102. <https://doi:10.19392/j.cnki.1671-7341.202026051>.

Gonzalez, Y. A. (2009). Sistema de gestion de Java Archives. *Universidad de las Ciencias Informáticas*.

Granados, H. T., & Millán, J. M. (2012). Estudio de Factibilidad, Desarrollo e Implementación del Sistema Integrado de Gestión Educativa aplicable a cualquier

nivel de educación (SIGA) Módulos de Gestión de Biblioteca y Gestión de Calidad.

<https://dspace.ups.edu.ec/bitstream/123456789/3162/1/UPS-GT000318.pdf>

JetBrains. (2023). Documentación de IntelliJ IDEA. *JetBrains*.

<https://www.jetbrains.com/help/idea/>

Jin, T. (2020). Diseño y realización del SGB. *Ingeniería de Sistemas de Información*, 07, 16-17.

Li, Y. (2016). Diseño y realización del sistema de información de gestión de bibliotecas de la Universidad de Radio y Televisión de Shanxi. *Revista de la Universidad de Radio y Televisión de Shanxi*, 02, 99-102.

Liu, J. C. (2013). Análisis de la situación actual de la ingeniería de software de computación. *Fabricación Electrónica*, 22, 75. <http://doi:10.16589/j.cnki.cn11-3571/tn.2013.22.216>.

Lu, K. J. (2024). Análisis de nuevas tecnologías de desarrollo de software de ingeniería en la era de "Internet+". *Mundo Digital de Comunicaciones*, 04, 95-97.

Luo, S., & Zhu, T. Y. (2019). Sistema de gestión de libros basado en el marco ASP.NET MVC. *Información y Computación (Versión Teórica)*, 05, 117-119.

Ning, W., Liming, L., Yanzhang, W., Yi-Bing, W., & Jing, W. (2008). Research on the web Information System development Platform based on MVC design pattern. 2009 *WASE International Conference on Information Engineering*.
<https://doi.org/10.1109/wiiat.2008.64>

Su, N. X., et al. (2019). Diseño de un sistema de préstamo y gestión de libros en línea basado en la arquitectura MVC. *Revista de la Universidad de Suihua*, 05, 135-138.

Wang, M. (2010). The building of library management system based on hibernate model.

2010 International Conference on Challenges in Environmental Science and Computer Engineering. <https://doi.org/10.1109/cesce.2010.255>

Xian, W. (2018). Diseño y realización del SGB del Instituto Politécnico de Petróleo y Química (Tesis de maestría, Universidad Electrónica y de Telecomunicaciones de Xi'an).

<https://kns.cnki.net/kcms2/article/abstract?v=mjz80qGfPOUSTr1CbxsbD9CRv9PtggATy33cZGxxqVqDOZbix1FoQWtLT2f4bt3ANPejSVnyBl2LFPeQT1k4e3hE6uaRMahXCGGTjBYCbYbiWswml09y9Ce7GYiipW3YrYs974UQRJE=&uniplatform=NZKPT&language=CHS>

Yang, X. L., & Shi, L. (2023). Resumen de la experiencia de transformación digital de pequeñas y micro bibliotecas de archivos. *Tecnología de Información Moderna*, 11, 133-137+142. <http://doi:10.19850/j.cnki.2096-4706.2023.11.030>.

Yao, Y. (2016). Diseño y realización del SGB de la Escuela de Economía Industrial de Anhui (Tesis de maestría, Universidad de Dalian Ligong).

https://kns.cnki.net/kcms2/article/abstract?v=mjz80qGfPOVGcCc779_0Ar1I7h1CD4SeGxRyT_MW0hcFCFCtrvIgH8tesi7qSH9Lxgqe5OnbpsM_qdexxW18USfnqI4pSIjQby9nY_nm7bUT_dDf0DPp59AUdUYtEavRBnNfjQKYok0=&uniplatform=NZKPT&language=CHS

Zainal, B. A. (2019). Developing a library management system for managing the books: a suggested model in the Central Library of Cihan University-Erbil. *International Journal of Advanced Science and Technology*, 28(13), 559–578.

Zhang, P., & Jiang, H. R. (2023). Diseño y realización de un SGB basado en MVC. *Nueva Tecnología Electrónica*, 03, 46-48.

Zhong, R. (2004). Overview of the research in model- view- controller pattern. *Application Research of Computers*, 10, 1–4. https://en.cnki.com.cn/Article_en/CJFDTOTAL-JSYJ200410001.htm