



**INSTITUTO POLITECNICO NACIONAL**

**CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS NÚMERO  
“BENITO JUAREZ”**



# “SISTEMA DE GESTIÓN BIBLIOTECARIO”

Presentan: Gracia Capetillo Mayte Guadalupe  
Del Olmo Ramírez Melissa  
Rangel Morales Grecia Odett



Asesora: Isaura Martínez Romero

PRESENTADO EL 05 DE JUNIO DE 2026

## Agradecimientos:

Quiero agradecer en primer lugar a mi mamá y a mi padrastro David por su gran esfuerzo y apoyo incondicional; su guía fue el motor de todo esto.

Asimismo, a las personas que marcaron la diferencia en este camino académico y personal:

A mi primo y a mi tío porque sus consejos y ejemplo influyeron más de lo que imaginan para que hoy forme parte del IPN.

A mi amigo Nath por el apoyo mutuo, la motivación y por hacer mucho más llevaderos los momentos difíciles para salir adelante.

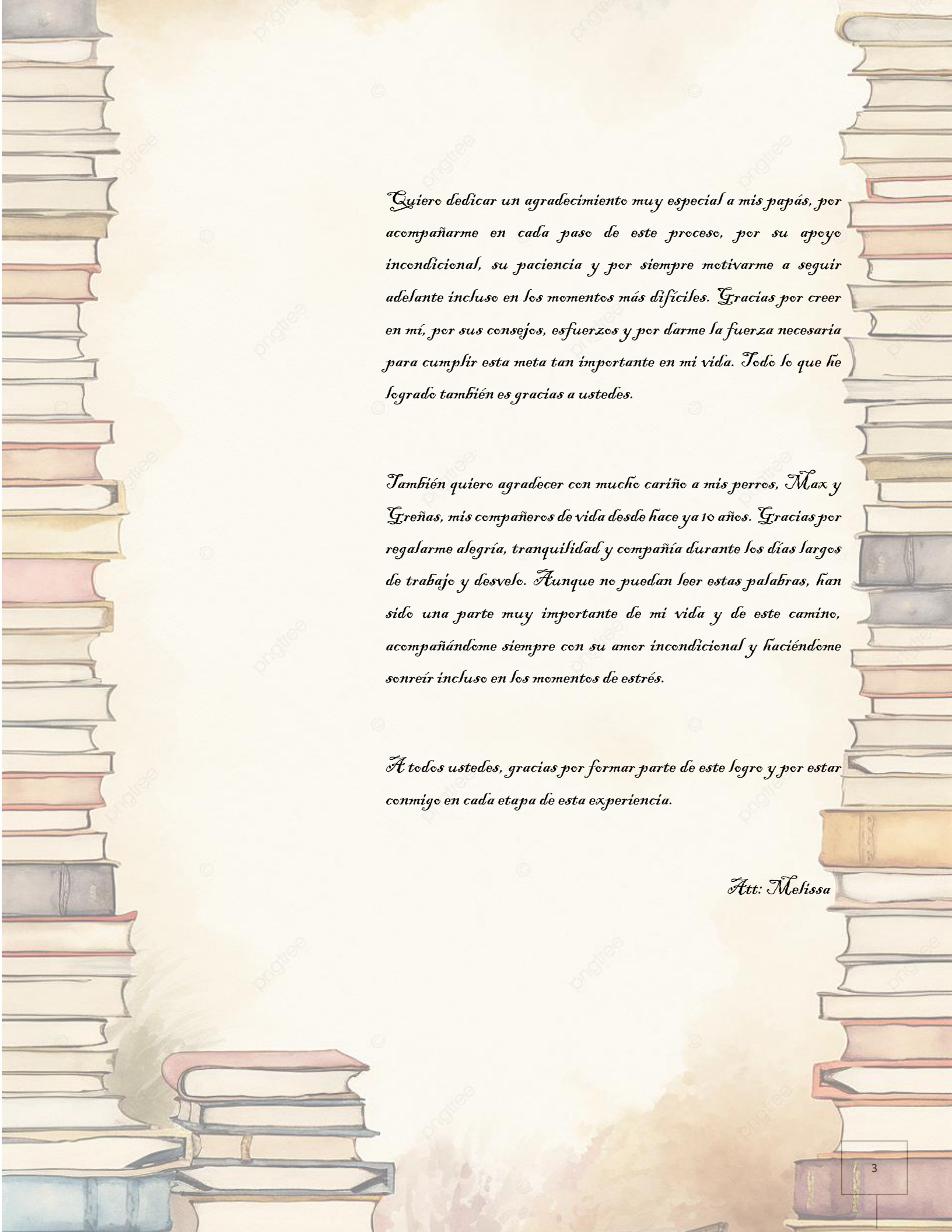
A mi amigo Jay por inspirarme a elegir la carrera de Informática y por darme el impulso para no rendirme cuando las cosas se complicaban.

A mi amiga Bri porque, aunque no lo sepa, pensar en ella me dio la fuerza para no dejarme vencer por las presiones y seguir adelante.

A mi hermano Brandon porque incluso con tus molestas, siempre logras hacerme reír y desestresarme en los momentos más pesados.

Finalmente, gracias a todos los que, de manera directa o indirecta, estuvieron presentes y me apoyaron para concluir con éxito este trabajo.

Att: Mayte



*Quiero dedicar un agradecimiento muy especial a mis papás, por acompañarme en cada paso de este proceso, por su apoyo incondicional, su paciencia y por siempre motivarme a seguir adelante incluso en los momentos más difíciles. Gracias por creer en mí, por sus consejos, esfuerzos y por darme la fuerza necesaria para cumplir esta meta tan importante en mi vida. Todo lo que he logrado también es gracias a ustedes.*

*También quiero agradecer con mucho cariño a mis perros, Max y Greñas, mis compañeros de vida desde hace ya 10 años. Gracias por regalarme alegría, tranquilidad y compañía durante los días largos de trabajo y desvelo. Aunque no puedan leer estas palabras, han sido una parte muy importante de mi vida y de este camino, acompañándome siempre con su amor incondicional y haciéndome sonreír incluso en los momentos de estrés.*

*A todos ustedes, gracias por formar parte de este logro y por estar conmigo en cada etapa de esta experiencia.*

*Att: Melissa*

*Quiero expresar mi más profundo agradecimiento a mis padres, por ser mi apoyo incondicional durante cada etapa de este camino. Gracias por su amor, paciencia, consejos y por impulsarme siempre a seguir adelante aun en los momentos más difíciles. Este logro también les pertenece a ustedes, porque sin su esfuerzo y confianza en mí, nada de esto habría sido posible.*

*A mis hermanos, gracias por acompañarme, apoyarme y enseñarme cosas tan valiosas a lo largo de mi vida. Cada experiencia, consejo y momento compartido ha dejado huellas importantes en mí y me ha ayudado a crecer como persona. Gracias por estar presentes y por motivarme de distintas maneras a nunca rendirme.*

*También quiero agradecer a todas las personas que han pasado por mi vida y que, de alguna forma, dejaron una enseñanza, un consejo, un recuerdo o un aprendizaje importante. Sería imposible mencionar a cada una, pero todas han contribuido a formar la persona que soy hoy. De cada experiencia y de cada persona he aprendido algo valioso que me acompaña en este logro.*

*Finalmente, agradezco a quienes estuvieron conmigo durante este proceso, brindándome apoyo, compañía y palabras de aliento. Gracias por formar parte de este momento tan importante en mi vida.*

*Att: Odett*

## TABLA DE CONTENIDO

<b>PLANEACIÓN</b>	<b>1</b>
<b>OBJETIVOS DEL SISTEMA</b>	<b>1</b>
Optimización Administrativa y Operativa	1
Análisis de Datos y Retroalimentación	1
<b>VIABILIDAD Y COSTOS DE IMPLEMENTACIÓN</b>	<b>2</b>
ALCANCE ESPERADO DEL PROYECTO	2
Alcance Funcional	3
Alcance Administrativo (Reporting)	3
Alcance Técnico y Soporte	3
<b>ANÁLISIS DEL SISTEMA</b>	<b>4</b>
<b>ANÁLISIS DE REQUERIMIENTOS PARA EL SISTEMA DE GESTIÓN DE BASE DE DATOS</b>	<b>4</b>
<b>IDENTIFICACIÓN DE NECESIDADES DEL USUARIO</b>	<b>4</b>
Técnicas para comprender las necesidades de los usuarios	5
Recolección de información	6
Entrevista	7
Resultados obtenidos	8
<b>REQUERIMIENTOS FUNCIONALES</b>	<b>10</b>
Características de los requerimientos funcionales	10
Tipos de requerimientos funcionales	10
Funciones principales del sistema de biblioteca	11
<b>REQUERIMIENTOS NO FUNCIONALES</b>	<b>11</b>
Principales requerimientos no funcionales	12
<b>IDENTIFICACIÓN DE ACTORES DEL SISTEMA</b>	<b>13</b>
Restricciones y condiciones del proyecto	14
<b>DISEÑO</b>	<b>16</b>
<b>Fundamentos del Análisis y Diseño de Sistemas (SAD)</b>	<b>16</b>
<b>El enfoque del Diseño Centrado en el Usuario (DCU)</b>	<b>18</b>
Fundamentos del Análisis y Diseño de Sistemas (SAD)	19
Principios de estructura, integridad y seguridad	19
Usabilidad y validación del sistema	20
<b>IDENTIDAD VISUAL</b>	<b>21</b>
DEFINIR PALETA DE COLORES	22
PAGINA WEB	22

LOGOTIPO _____	23
DEFINIR PARAMETROS DE LA APLICACIÓN. _____	25
<b>Arquitectura básica del sistema _____</b>	<b>27</b>
Proceso del sistema _____	
<b>Principios técnicos para la integridad de la información _____</b>	
<b>El Diseño Centrado en el Usuario (DCU) _____</b>	
<b><i>DESARROLLO</i> _____</b>	
<b>ARQUITECTURA DEL SISTEMA _____</b>	<b>--</b>
CAPA DE PRESENTACIÓN _____	33
CAPA LÓGICA O DE NEGOCIO _____	34
<b>CAPA DE DATOS _____</b>	<b>35</b>
Arquitectura Basada en Servicios (SOA) _____	35
Seguridad del sistema _____	36
<b>PROCESOS DEL SISTEMA _____</b>	<b>37</b>
Proceso de registro de usuarios _____	37
Proceso de gestión del catálogo _____	37
Proceso de búsqueda de libros _____	38
Proceso de autenticación y control de acceso _____	38
<b>Integración de procesos _____</b>	<b>39</b>
<b>IMPLEMENTACIÓN DE ENTRADAS Y SALIDAS _____</b>	<b>39</b>
ENTRADAS _____	39
SALIDAS _____	40
Informes Visuales _____	41
Consultas de Selección y Parámetros (Outputs en Pantalla) _____	43
ESTRUCTURA DE LA BASE DE DATOS _____	44
<b><i>IMPLEMENTACIÓN</i> _____</b>	<b>45</b>
<b>INTRODUCCIÓN A LA IMPLEMENTACIÓN _____</b>	<b>45</b>
<b>INSTALACIÓN DEL SISTEMA _____</b>	<b>48</b>
<b>Configuración y entorno (uso de Access y requisitos) _____</b>	<b>50</b>
<b>Capacitación del Usuario _____</b>	<b>52</b>
<b>USABILIDAD Y ACCESIBILIDAD _____</b>	<b>57</b>
AJUSTE Y REFACTORIZACIÓN _____	59
IMPACTO SOCIO-TÉCNICO _____	60
<b>PLAN DE MITIGACIÓN DE RIESGOS _____</b>	<b>60</b>
CONTROL DE CONCURRENCIA Y BLOQUEO TRANSACCIONAL _____	61
PROTOCOLO DE CAIDA DE RED _____	62
IMPORTANCIA DEL BLINDAJE DE IMPLEMENTACIÓN _____	63

<b>PRUEBAS</b>	<b>64</b>
<b>FUNDAMENTOS DE LAS PRUEBAS</b>	<b>64</b>
Evolución metodológica y cambio de paradigma	65
El Ciclo de Vida de las Pruebas de Software (STLC)	
<b>IMPORTANCIA DEL PROCESO DE PRUEBAS</b>	
Detección temprana y la ley de amplificación de costos	
Casos de estudio históricos: El costo del fallo de software	
Beneficios estratégicos del aseguramiento de calidad (SQA)	
<b>TIPOS DE PRUEBAS</b>	--
<b>Pruebas Funcionales</b>	69
Pruebas no funcionales	70
<b>CASOS DE PRUEBA</b>	<b>72</b>
PRUEBAS DE INTERFAZ Y USABILIDAD	72
PRUEBAS LOGICAS Y DE SEGURIDAD	72
PRUEBAS DE FUNCIONABILIDAD	73
<b>CAPA DE DATOS</b>	<b>73</b>
<b>ARQUITECTURA BASADA EN SERVICIOS (SOA)</b>	<b>74</b>
<b>SISTEMA DE SEGURIDAD</b>	<b>74</b>
<b>ESCALABILIDAD Y MANTENIMIENTO</b>	<b>75</b>
<b>DESCRIPCIÓN FUNCIONAL DEL SISTEMA</b>	<b>75</b>
GESTIÓN DE USUARIOS	75
GESTIÓN DE CATALOGO Y MOTOR DE BUSQUEDA	76
<b>RESULTADOS</b>	<b>76</b>
EFICIENCIA EN PROCESO DE DATOS	76
INTERFAZ Y USABILIDAD	76
SEGURIDAD Y CONTROL DE ACCESO	76
<b>ANÁLISIS DE ERRORES Y ESTRATEGIAS DE MEJORA CONTINUA.</b>	<b>77</b>
GESTIÓN ESTRATEGICA Y DEFINICIÓN DE OBJETIVOS	77
FACTOR HUMANO	77
DOCUMENTACIÓN Y CUMPLIMIENTO NARRATIVO	78
<b>MANTENIMIENTO</b>	<b>78</b>
<b>CONCEPTO DE MANTENIMIENTO DE SOFTWARE</b>	<b>78</b>
IMPORTANCIA DEL MANTENIMIENTO	79
<b>TIPOS DE MANTENIMIENTO DE SOFTWARE</b>	<b>79</b>
MANTENIMIENTO CORRECTIVO	79
MANTENIMIENTO ADAPTATIVO	79
MANTENIMIENTO PERFECTIVO	80
MANTENIMIENTO PREVENTIVO	80

<b>PROCESO DE MANTENIMIENTO</b>	<b>80</b>
DOCUMENTACIÓN Y RETROALIMENTACIÓN	81
RETOS Y SOLUCIONES EN EL MANTENIMIENTO	
<b>CORRECCIÓN DE ERRORES</b>	
<b>ACTUALIZACIÓN DEL SISTEMA</b>	
<b>SOPORTE AL USUARIO</b>	
SOPORTE INTERNO	
SOPORTE EXTERNO	85
BENEFICIOS DEL SOPORTE AL USUARIO	85
<b>MEJORAS EN EL SISTEMA</b>	<b>85</b>
FORTALECIMIENTO DE LA INTEGRIDAD REFERENCIAL	86
NORMALIZACIÓN Y ESTABILIDAD DE LA BASE DE DATOS	86
<b>OPTIMIZACIÓN DE LA EXPERIENCIA AL USUARIO</b>	<b>87</b>
<b>ARQUITECTURA MEJORADA</b>	<b>87</b>
<b>PROYECCIÓN A FUTURO Y MEJORA CONTINUA</b>	<b>88</b>
<b>RESPALDO DE INFORMACIÓN</b>	<b>88</b>
<b><i>BIBLIOGRAFIA</i></b>	<b>91</b>
<b><i>TABLA DE IMAGENES</i></b>	<b>94</b>

# INTRODUCCIÓN

En la era digital contemporánea, la implementación de sistemas informáticos ha dejado de ser una opción para convertirse en una necesidad estratégica fundamental. En el ámbito educativo y administrativo, estas herramientas permiten optimizar la organización de grandes volúmenes de datos, transformando procesos manuales obsoletos en flujos de trabajo dinámicos y precisos. Las bibliotecas escolares, en particular, funcionan como centros neurálgicos de información que gestionan catálogos extensos, registros de usuarios y un flujo constante de préstamos. Por ello, requieren de mecanismos tecnológicos que no solo agilicen la consulta, sino que garanticen la integridad y el control absoluto de su acervo.

El presente trabajo tiene como objetivo primordial el diseño, desarrollo e implementación de un Sistema de Gestión Bibliotecario robusto, utilizando la plataforma Microsoft Access como motor de base de datos relacional. La elección de esta herramienta responde a su capacidad para integrar interfaces de usuario intuitivas con una lógica de procesamiento eficiente, ideal para entornos administrativos que demandan rapidez sin comprometer la seguridad de la información. Este sistema ha sido concebido para automatizar tareas críticas que tradicionalmente consumen tiempo y son propensas al error humano, tales como la catalogación de nuevos ejemplares, el monitoreo de fechas de devolución y la administración detallada de los perfiles de los usuarios.

A lo largo del desarrollo de este proyecto, se han aplicado de manera integral conocimientos avanzados en ingeniería de software y gestión de datos. El enfoque no se limitó únicamente a la funcionalidad operativa, sino que abarcó aspectos determinantes como la normalización de tablas, el mantenimiento preventivo y correctivo del software, y el establecimiento de protocolos de respaldo para salvaguardar la continuidad de la información ante posibles contingencias técnicas. Cada fase del desarrollo fue guiada por un análisis de requerimientos que prioriza las necesidades

reales de los bibliotecarios y estudiantes, buscando reducir la brecha entre el usuario y la tecnología.

Más allá del despliegue técnico, esta tesina busca profundizar en la importancia de la modernización administrativa dentro de las instituciones educativas. A través de este desarrollo, se pretende evidenciar cómo la transición hacia procesos automatizados impacta positivamente en la productividad, disminuyendo drásticamente los errores en el manejo manual de datos y proporcionando herramientas de reporte que facilitan la toma de decisiones. En última instancia, este proyecto es un testimonio de cómo la innovación tecnológica, aplicada con rigor metodológico, puede transformar la gestión del conocimiento, aportando beneficios tangibles en términos de organización, velocidad de respuesta y seguridad informática para la comunidad académica.



# INSTITUTO POLITECNICO NACIONAL

## CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS NÚMERO “BENITO JUAREZ”



### PLANEACIÓN

## OBJETIVOS DEL SISTEMA

El propósito central de este proyecto consiste en el diseño, desarrollo e implementación de un Sistema Integral de Administración de Bibliotecas, cuya arquitectura tecnológica está orientada a optimizar la gestión, organización y control de los recursos bibliográficos. Mediante el uso de herramientas de vanguardia y el diseño de bases de datos relacionales robustas, se busca transformar la operatividad administrativa en un proceso automatizado y eficiente.

### Optimización Administrativa y Operativa

El sistema se ha estructurado para permitir el registro, almacenamiento, consulta y actualización digital de la información referente a ejemplares, perfiles de usuarios y el flujo dinámico de movimientos. Según los estándares de los sistemas integrados de gestión bibliotecaria (*Koha Community, s.f.*), este despliegue tecnológico persigue los siguientes fines:

Reducción de Errores: Mitigar las inconsistencias y fallas críticas inherentes al registro manual y la gestión física de archivos.

Eficiencia en la Búsqueda: Implementar algoritmos de clasificación y organización que permitan filtrar el acervo por metadatos específicos (título, autor, género, editorial), facilitando la localización inmediata de materiales y optimizando los tiempos de respuesta al usuario.

Seguridad de la Información: Centralizar los datos para garantizar su integridad y disponibilidad permanente, estableciendo mecanismos que eviten pérdidas accidentales y aseguren el resguardo de la propiedad intelectual de la institución.

### Análisis de Datos y Retroalimentación

Una función distintiva y de alto valor agregado en este sistema es la integración de módulos de valoración. A través de instrumentos de recolección de datos, como encuestas breves, los usuarios podrán calificar las obras y proporcionar comentarios cualitativos. Este enfoque permite que los administradores, apoyados en la inteligencia de datos (*Lucidea, 2025*), puedan:

- Identificar tendencias de lectura y determinar el ranking de popularidad de los títulos.
- Generar estadísticas descriptivas y reportes automáticos que sirvan como insumo para la toma de decisiones estratégicas.
- Optimizar la adquisición de nuevos títulos basándose en la demanda real y las preferencias detectadas en la comunidad académica.

## VIABILIDAD Y COSTOS DE IMPLEMENTACIÓN

Para garantizar la ejecución exitosa y la sostenibilidad del sistema, se han identificado cuatro pilares económicos y técnicos fundamentales, siguiendo modelos de gestión de costos para entidades tecnológicas (*Ernesto & Bolia, 2007*):

Hardware: Consideración de la infraestructura física necesaria, incluyendo terminales de cómputo para el personal y el alojamiento de la base de datos en un entorno de servidor que garantice la estabilidad del sistema.

Software: Evaluación de costos asociados al sistema de gestión de base de datos (SGBD). Se prioriza el uso de herramientas accesibles y potentes como Microsoft Access, analizando su costo-beneficio frente a opciones de código abierto (*Tramullas, 2020*).

Mantenimiento: Planificación presupuestaria para actualizaciones periódicas, corrección de errores (*debugging*) y optimización técnica continua, asegurando que el software evolucione junto a las necesidades de la biblioteca.

Capacitación: Recursos destinados al entrenamiento especializado del personal administrativo, asegurando que la adopción tecnológica sea fluida y el uso de la herramienta alcance su máximo potencial desde el día de su implementación.

## ALCANCE ESPERADO DEL PROYECTO

Con base en los requerimientos técnicos y funcionales planteados, el alcance del proyecto se define en tres dimensiones críticas:

## Alcance Funcional

Comprende la operatividad directa del software, incluyendo la gestión total del catálogo mediante búsqueda filtrada y una interfaz de control de circulación en tiempo real. Este módulo vincula cada préstamo y devolución a un perfil de usuario único, garantizando la trazabilidad de cada movimiento.

## Alcance Administrativo (Reporting)

El sistema actuará como una herramienta de inteligencia de negocios (BI) básica, entregando informes automáticos sobre el estado del inventario y la satisfacción del lector. Provee datos estructurados fundamentales para la planificación administrativa y la renovación del material bibliográfico.

## Alcance Técnico y Soporte

Se enfoca en el diseño de una arquitectura de datos relacional segura bajo principios de normalización de bases de datos (*Silberschatz et al., 2019*). Asimismo, incluye la transferencia de conocimiento mediante la entrega de manuales de usuario y sesiones de capacitación técnica, permitiendo que el personal del CECyT 5 opere el sistema de forma autónoma y eficiente.

# ANÁLISIS DEL SISTEMA

## ANÁLISIS DE REQUERIMIENTOS PARA EL SISTEMA DE GESTIÓN DE BASE DE DATOS

El análisis de requerimientos es una etapa fundamental dentro del desarrollo de sistemas de información, ya que permite comprender el problema que se desea resolver y definir los elementos que formarán parte de la solución técnica. En esta fase, se estudian minuciosamente las necesidades de los usuarios, el funcionamiento del sistema actual y las funciones que deberá cumplir la solución tecnológica para garantizar que el producto final responda a los problemas reales del entorno en el que se implementará (*Pressman & Maxim, 2020*).

En el contexto de una biblioteca, el análisis de requerimientos permite examinar cómo se administran los libros, cómo se registran los préstamos y devoluciones, y cuáles son las principales dificultades que enfrentan los usuarios al utilizar los servicios bibliotecarios. A partir de este análisis, es posible identificar las funciones necesarias para diseñar un sistema que mejore la organización de la información y facilite el acceso a los recursos disponibles mediante una estructura de datos lógica y eficiente (*Silberschatz et al., 2019*).

Durante esta etapa, se utilizan diversas técnicas de investigación para recopilar información, comprender las expectativas de los interesados y traducirlas en requerimientos funcionales y no funcionales. Estos servirán como la base técnica para el diseño, desarrollo y posteriores pruebas del sistema, asegurando que la arquitectura del software sea coherente con los objetivos institucionales (*Sommerville, 2016*). De esta manera, un análisis de requerimientos riguroso minimiza los riesgos de fallos en la implementación y garantiza una herramienta útil para la comunidad académica del CECyT 5.

## IDENTIFICACIÓN DE NECESIDADES DEL USUARIO

Dentro del análisis de requerimientos, la identificación de necesidades del usuario es una de las actividades más fundamentales, ya que permite comprender las deficiencias del sistema actual y las expectativas de quienes utilizan el servicio. Esta etapa es crucial para establecer las funciones que el sistema deberá incluir con el fin de optimizar el funcionamiento de la biblioteca (*Pressman & Maxim, 2020*).

En una biblioteca, los principales usuarios suelen ser los estudiantes, docentes y el personal encargado de administrar los recursos bibliográficos. Cada uno de estos grupos tiene necesidades específicas:

Estudiantes: Necesitan encontrar libros de manera rápida y conocer la disponibilidad de los materiales en tiempo real.

Docentes: Requieren consultar información bibliográfica precisa para apoyar sus actividades académicas o investigaciones.

Personal administrativo: Precisa llevar un control organizado del inventario, registrar préstamos y devoluciones, y mantener actualizada la base de datos de usuarios (*Sommerville, 2016*).

### Técnicas para comprender las necesidades de los usuarios

Para identificar correctamente estas necesidades, se aplican diversas técnicas de investigación que permiten analizar expectativas, comportamientos y dificultades en el servicio (*Mampel, 2025*):

- **Determinación de la disposición al cambio**: Es vital analizar si los usuarios están dispuestos a adoptar un nuevo sistema digital. Identificar las inconformidades con los métodos tradicionales (como registros manuales) facilita que los usuarios acepten la herramienta al percibir mejoras en la organización y búsqueda de materiales.
- **Escucha activa**: Esta técnica permite comprender las experiencias reales de los usuarios mediante la atención plena y el uso de preguntas abiertas, lo que ayuda a identificar problemas críticos en el flujo de trabajo actual.
- **Empatía con los usuarios**: Consiste en analizar la experiencia desde la perspectiva del usuario. Por ejemplo, entender la frustración de un estudiante que tarda demasiado en localizar un libro permite diseñar un sistema que facilite la búsqueda y mejore la experiencia de uso (Nielsen, 1994).

- Identificación de los motivos de uso: Comprender si el usuario busca investigación académica o material de apoyo personal permite organizar la información de manera más eficiente y facilitar el acceso a los recursos más utilizados.
- Uso de información del personal de atención: Debido a su interacción constante con el público, el personal de la biblioteca es una fuente valiosa para identificar problemas frecuentes y proponer mejoras en los procesos de gestión.
- Demostraciones y pruebas del sistema: Presentar prototipos o versiones preliminares permite observar la interacción real del usuario con la herramienta y recopilar opiniones para detectar posibles mejoras antes de la implementación final (Sommerville, 2016).
- Uso de estudios y datos del sector: Analizar investigaciones sobre la gestión de bibliotecas y sistemas de información permite conocer tendencias tecnológicas y estrategias exitosas utilizadas en otras instituciones educativas (*International Federation of Library Associations and Institutions, s. f.*).

### Recolección de información

La recolección de información es una actividad fundamental dentro del análisis de requerimientos, ya que permite obtener datos precisos sobre el funcionamiento actual de la biblioteca y las dificultades específicas que enfrentan los usuarios. Esta fase es esencial para asegurar que el diseño del sistema se base en hechos observables y necesidades reales, minimizando la ambigüedad en el desarrollo (*Pressman & Maxim, 2020*).

Para este proyecto, se han definido las siguientes técnicas de recolección de datos:

- Observación: Consiste en analizar directamente cómo se desarrollan las actividades cotidianas dentro de la biblioteca, tales como el proceso de búsqueda de libros o la dinámica del registro de préstamos. Esta técnica permite identificar cuellos de botella que los usuarios podrían no mencionar de forma verbal (*Sommerville, 2016*).
- Cuestionarios o encuestas: Permiten recopilar información de un mayor número de usuarios mediante preguntas estructuradas sobre el servicio bibliotecario. Es una herramienta eficaz

para obtener datos estadísticos sobre la satisfacción del usuario y las funciones más demandadas (Mampel, 2025).

- Focus group: Es una técnica cualitativa que reúne a un grupo representativo de usuarios (estudiantes y docentes) para discutir sus opiniones y percepciones sobre los servicios de la biblioteca. Esto ayuda a generar ideas para nuevas funciones del sistema basadas en la discusión colectiva.
- Entrevistas: Las entrevistas permiten obtener información detallada y profunda mediante conversaciones directas con los usuarios o el personal administrativo de la biblioteca. *Según Contacto (s.f)*, esta interacción personal es clave para explorar necesidades complejas que requieren una explicación detallada por parte de los actores del sistema.

## Entrevista

Para conocer las necesidades de los posibles usuarios del sistema de gestión de biblioteca privada, se realizó una entrevista con el propósito de identificar problemas, preferencias y funciones necesarias para el desarrollo del sistema.

### PREGUNTAS REALIZADAS:

1. Nombre
2. Correo electrónico
3. ¿Te gustaría consultar los libros con los que cuentas de forma remota, desde internet?
  - Si
  - No
4. ¿Qué función consideras más útil en un sistema de biblioteca?
  - Búsqueda de libros
  - Registro de préstamos
  - Reservación de libros
  - Notificaciones de devolución
5. ¿Crees que un sistema digital ayudaría a mejorar la organización de la biblioteca?
  - Si
  - No
  - Tal vez
6. ¿Qué problema encuentras con más frecuencia en la biblioteca?
  - Dificultad para encontrar libros
  - Libros no disponibles
  - Falta de organización

- Ningún problema
- 7. ¿Qué mejorarías del servicio de biblioteca?
  - Mejor organización de libros
  - Sistema digital más rápido
  - Mayor cantidad de libros
  - Atención al usuario
- 8. En general, ¿cómo calificarías el servicio de la biblioteca?
  - Excelente
  - Bueno
  - Regular
  - Malo
  - Otro:

### Resultados obtenidos

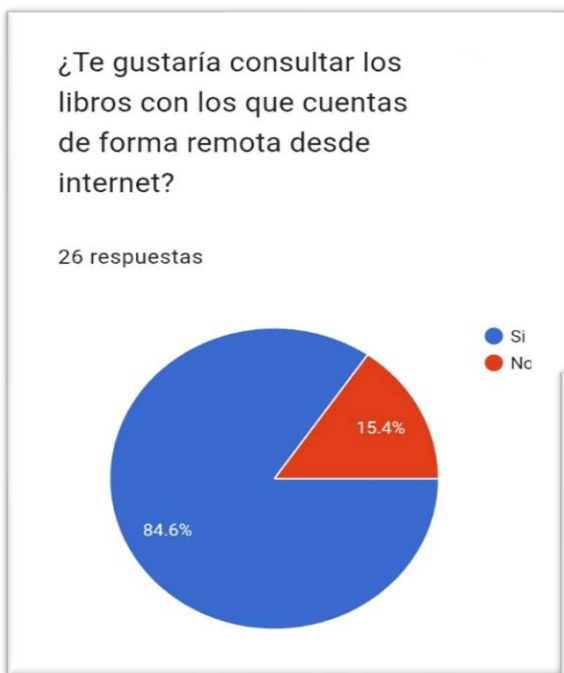


Tabla 1\_PREGUNTA1\_ENTREVISTA

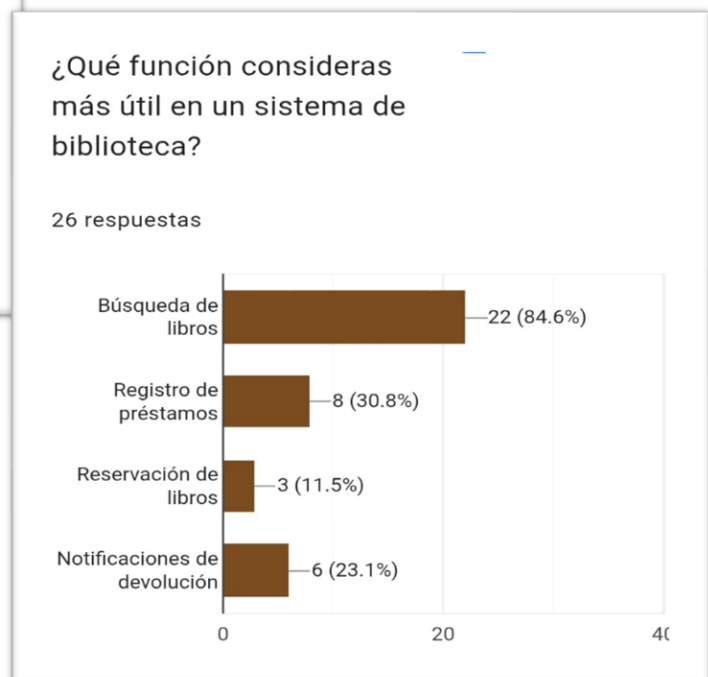


Tabla 2\_PREGUNTA2\_ENTREVISTA

¿Crees que un sistema digital ayudaría a mejorar la organización de la biblioteca?

26 respuestas

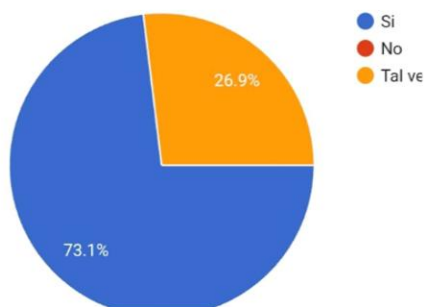


Tabla 3\_PREGUNTA3\_ENTREVISTA

¿Que problema encuentras con más frecuencia en la biblioteca?

26 respuestas

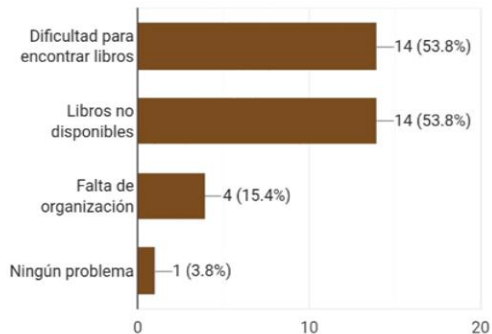


Tabla 9\_PREGUNTA4\_ENTREVISTA

Tabla 4\_PREGUNTA3\_ENTREVISTA  
Tabla 5\_PREGUNTA3\_ENTREVISTA

26 respuestas

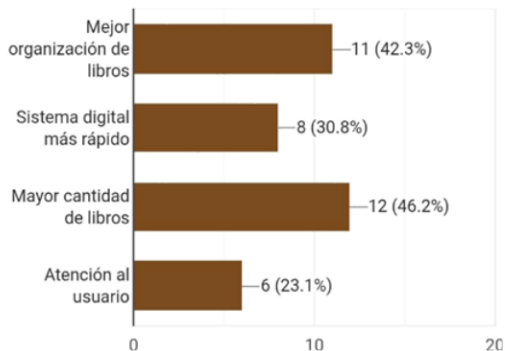


Tabla 12\_PREGUNTA5\_ENTREVISTA

Tabla 10\_PREGUNTA5\_ENTREVISTA  
Tabla 11\_PREGUNTA4\_ENTREVISTA

En general, ¿como calificarías el servicio de la biblioteca?

26 respuestas

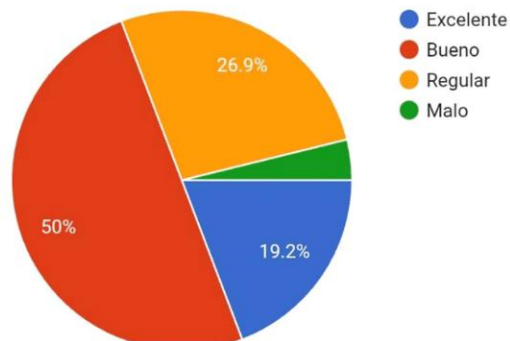


Tabla 15\_PREGUNTA6\_ENTREVISTA

Tabla 13\_PREGUNTA6\_ENTREVISTA  
Tabla 14\_PREGUNTA5\_ENTREVISTA

Tabla 16\_DIAGRAMA-DE-FLUJO\_Tables  
Tabla 17\_PREGUNTA6\_ENTREVISTA

# REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales constituyen la descripción de las funciones y servicios que el sistema debe proporcionar para satisfacer las necesidades de los usuarios y cumplir con los objetivos operativos institucionales. Estos requerimientos especifican las acciones que el software debe ejecutar, las respuestas ante determinadas entradas y la lógica con la que se gestiona la información. En la ingeniería de software, son fundamentales porque definen el comportamiento esperado del sistema y sirven como base para el diseño, el desarrollo y las fases de pruebas (Pressman & Maxim, 2020).

## Características de los requerimientos funcionales

Estos requerimientos se enfocan en describir "qué" debe hacer el sistema. Definen las funcionalidades específicas, la interacción del usuario con la interfaz y las operaciones necesarias para procesar datos y generar resultados. Según Sommerville (2016), para que un requerimiento funcional sea efectivo, debe poseer las siguientes características:

- Claridad y Especificidad: Deben evitar ambigüedades para que los desarrolladores comprendan exactamente la función a programar.
- Verificabilidad: Deben ser medibles para evaluar, mediante pruebas, si el sistema cumple con lo solicitado.
- Interacción de Componentes: Deben establecer cómo se comunican los diferentes módulos del sistema entre sí y con el usuario final.

## Tipos de requerimientos funcionales

La clasificación de estos requerimientos depende del tipo de operación que el sistema realiza sobre la información:

- Requerimientos de entrada: Describen los métodos, formularios y campos mediante los cuales los datos ingresan al sistema. Garantizan que la información sea correcta, completa y válida antes de ser procesada.
- Requerimientos de procesamiento: Definen la lógica y las operaciones de transformación sobre los datos (cálculos, validaciones y actualizaciones), convirtiendo las entradas en información de valor (Pressman & Maxim, 2020).

- Requerimientos de salida: Establecen los resultados que el sistema debe mostrar, ya sea en forma de reportes, pantallas informativas, notificaciones automáticas o documentos generados para el usuario (*Sommerville, 2016*).
- Requerimientos de gestión de datos: Se encargan de las operaciones de almacenamiento, consulta, actualización y eliminación de registros dentro del repositorio o base de datos, asegurando la accesibilidad y organización de la información (*Pressman & Maxim, 2020*).
- Requerimientos de seguridad: Implementan mecanismos de protección como el control de acceso, la autenticación de perfiles y la asignación de permisos, garantizando la confidencialidad e integridad de los datos almacenados (*Sommerville, 2016*).

### *Funciones principales del sistema de biblioteca*

El sistema diseñado para el CECyT 5 debe operar como una plataforma integral que facilite la administración de recursos y el acceso a los materiales mediante las siguientes funciones clave:

- Registro de libros en el catálogo digital: Permite el alta de materiales incluyendo metadatos como título, autor, editorial, categoría y existencias, manteniendo un inventario actualizado.
- Motor de búsqueda de materiales: Herramienta eficiente para localizar recursos mediante diversos criterios (título, autor, tema o palabras clave), optimizando los tiempos de consulta.
- Gestión de préstamos y devoluciones: Registro de movimientos vinculados a un usuario específico, con control de fechas y actualización automática de la disponibilidad del catálogo.
- Administración de usuarios: Registro y control de los perfiles de estudiantes, docentes y personal autorizado, junto con su historial de actividad.
- Control de retrasos y sanciones: Identificación automática de usuarios con entregas fuera de plazo para la generación de alertas, registro de multas o aplicación de restricciones en el servicio.
- Generación de reportes administrativos: Producción de estadísticas sobre el uso de la biblioteca, libros más solicitados y reportes de materiales pendientes, facilitando la toma de decisiones para la mejora del servicio bibliotecario.

## REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales describen las características y condiciones de calidad que debe poseer un sistema para operar de manera adecuada y eficiente. A diferencia de los requerimientos funcionales, que especifican las acciones o tareas que el sistema debe realizar, los requerimientos no funcionales se enfocan en los atributos del sistema y en los estándares de calidad

que debe cumplir. Estos aspectos incluyen dimensiones críticas como el rendimiento, la seguridad, la facilidad de uso y la confiabilidad operativa (*Sommerville, 2016*).

La correcta definición de estos requerimientos es esencial en el desarrollo de software, ya que influyen directamente en la experiencia del usuario y en el comportamiento del sistema bajo diferentes condiciones de carga o entorno. Además, garantizan que la arquitectura sea estable, segura y capaz de adaptarse a cambios o crecimientos futuros. Cuando estos parámetros se establecen con precisión, contribuyen significativamente a elevar la calidad del software y a reducir incidencias técnicas durante su implementación y fase de producción (*Pressman & Maxim, 2020*).

### Principales requerimientos no funcionales

Para asegurar que el sistema de gestión bibliotecaria funcione correctamente y cumpla con las expectativas institucionales, se han identificado los siguientes tipos de requerimientos no funcionales:

- **Usabilidad:** Se refiere a la facilidad con la que los usuarios pueden aprender a interactuar con el software. Un sistema con alta usabilidad debe presentar una interfaz clara e intuitiva que permita realizar tareas de manera ágil. Esto abarca la organización lógica de la información, la claridad en los menús y una navegación fluida dentro de la aplicación (*Sommerville, 2016*).
- **Seguridad:** Representa un pilar fundamental para proteger la integridad de los datos contra accesos no autorizados, pérdidas o alteraciones malintencionadas. El sistema debe implementar mecanismos de autenticación, uso de contraseñas y niveles de privilegios (control de acceso), asegurando que cada actor interactúe únicamente con las funciones permitidas según su perfil (*Pressman & Maxim, 2020*).
- **Rendimiento:** Determina la capacidad de respuesta del software ante las solicitudes del usuario. Un rendimiento óptimo implica que las búsquedas en el catálogo, la generación de reportes y los registros de préstamos se ejecuten en tiempos mínimos, evitando demoras que afecten la productividad administrativa o la satisfacción del lector (*Sommerville, 2016*).
- **Disponibilidad:** Indica que el sistema debe permanecer accesible y operativo en el horario en que los usuarios lo requieran. En un entorno educativo, es vital minimizar los tiempos de inactividad técnica para garantizar el acceso continuo a los recursos bibliográficos y la información de los ejemplares (*Pressman & Maxim, 2020*).
- **Escalabilidad:** Se define como la capacidad del sistema para adaptarse al incremento en el volumen de datos (crecimiento del catálogo) o de usuarios sin degradar su funcionamiento.

Un diseño escalable permite expandir la base de datos o añadir nuevos módulos funcionales sin necesidad de reconstruir la arquitectura base del sistema (*Sommerville, 2016*).

- Confiabilidad: Evalúa la capacidad del sistema para operar de manera correcta y sin fallos durante periodos prolongados. La confiabilidad asegura que la información almacenada sea consistente y que existan mecanismos de respaldo para recuperar datos ante posibles fallas técnicas o errores del sistema, resguardando la integridad institucional (*Pressman & Maxim, 2020*).

En conjunto, estos requerimientos no funcionales aseguran que el sistema no solo cumpla con sus objetivos operativos, sino que lo haga bajo estándares de eficiencia, seguridad y calidad, elementos determinantes para el éxito de cualquier proyecto de tecnologías de la información.

## IDENTIFICACIÓN DE ACTORES DEL SISTEMA

La identificación de actores es una etapa crítica dentro del análisis y diseño de sistemas de información. Los actores representan a las personas, organizaciones o sistemas externos que interactúan con el software para cumplir funciones determinadas, ya sea enviando información, solicitando servicios o recibiendo resultados. Identificarlos correctamente permite comprender el uso real del sistema y las necesidades que deben cubrirse durante su desarrollo (*Sommerville, 2016*).

En el caso del sistema de gestión bibliotecaria para el CECyT 5, los actores son las entidades que interactúan con la base de datos para administrar o consultar el acervo. Cada uno posee responsabilidades y niveles de acceso diferenciados según su tipo de interacción (*Pressman & Maxim, 2020*):

- Administrador de la biblioteca: Es el actor responsable de la gestión integral del sistema. Posee permisos avanzados para registrar nuevos libros, actualizar materiales, gestionar préstamos, administrar usuarios y generar reportes estadísticos. Su papel es fundamental para asegurar que la información almacenada sea correcta y esté disponible (*Sommerville, 2016*).
- Usuarios (Estudiantes o lectores): Son los principales beneficiarios del sistema. Utilizan la plataforma para realizar búsquedas en el catálogo, verificar la disponibilidad de títulos, solicitar préstamos y consultar su historial personal, lo que mejora significativamente su experiencia de acceso a la información (*Pressman & Maxim, 2020*).

- Docentes o investigadores: Interactúan con el sistema para localizar material bibliográfico específico que apoye sus proyectos o actividades académicas. El sistema les permite verificar la disponibilidad de artículos o documentos especializados, convirtiéndose en una herramienta de apoyo para la enseñanza y la investigación (Sommerville, 2016).

## Restricciones y condiciones del proyecto

El desarrollo del sistema requiere un análisis de factores que influyen en su diseño e implementación. Estos se dividen en restricciones (limitaciones del proyecto) y condiciones (requisitos para el funcionamiento correcto).

### Restricciones del proyecto

- Recursos tecnológicos: El software debe ser compatible con la infraestructura existente en la biblioteca (computadoras y servidores). Equipos obsoletos podrían limitar el rendimiento del sistema.
- Capacidad de almacenamiento: Es necesario garantizar espacio suficiente en la base de datos para el crecimiento futuro del catálogo y el registro de usuarios sin afectar la velocidad de consulta.
- Presupuesto: Los recursos limitados obligan a priorizar funciones o utilizar herramientas de software accesibles o de código abierto para evitar gastos excesivos.
- Tiempo: El proyecto debe cumplir con un cronograma estricto que abarque desde el análisis hasta la implementación y pruebas finales.
- Conocimiento técnico: El sistema debe desarrollarse con tecnologías que el equipo domine adecuadamente para reducir errores y facilitar el mantenimiento futuro.

### Condiciones del sistema

- Seguridad e Integridad: Implementación de contraseñas y niveles de acceso para proteger los datos contra usos no autorizados. Asimismo, se deben realizar copias de seguridad periódicas para prevenir la pérdida de información.
- Facilidad de uso: El software debe contar con una interfaz intuitiva que permita a los administradores realizar registros y búsquedas sin complicaciones técnicas.
- Validación de datos: Mecanismos que aseguren que la información ingresada sea correcta, evitando registros duplicados o campos obligatorios vacíos.

- Disponibilidad y Reportes: El sistema debe operar de manera estable y permitir la generación de reportes estadísticos (libros más prestados, autores consultados) para apoyar la toma de decisiones administrativas.
- Escalabilidad y Mantenimiento: El diseño debe ser flexible para incorporar nuevas funciones a futuro y permitir actualizaciones periódicas que corrijan errores o adapten el sistema a nuevas necesidades.

### Fundamentos del Análisis y Diseño de Sistemas (SAD)

El diseño de la base de datos para una biblioteca se fundamenta en los principios del Análisis y Diseño de Sistemas (SAD), disciplina que ha evolucionado desde mediados del siglo XX como respuesta a la creciente complejidad de los sistemas organizacionales (*Sommerville, 2016*). A medida que las instituciones comenzaron a manejar grandes volúmenes de información, surgió la necesidad metodológica de estructurarla de forma lógica, eficiente y segura mediante herramientas informáticas especializadas.

En este contexto, un sistema se define como un conjunto de elementos interrelacionados que operan de manera coordinada para alcanzar un objetivo específico (*Pressman & Maxim, 2020*). En el escenario de una biblioteca, este ecosistema no se limita exclusivamente al repositorio de datos, sino que engloba todos los procesos operativos que permiten su funcionamiento, tales como el registro de materiales bibliográficos, la administración de usuarios, el control transaccional de préstamos y devoluciones, así como la generación automatizada de reportes estadísticos y administrativos.

El diseño de la base de datos constituye una etapa crítica dentro del ciclo de vida del desarrollo de software, ya que en ella se determina la arquitectura lógica de la información, las relaciones entre las entidades y las reglas de negocio que rigen su funcionamiento. Una planificación inadecuada en esta fase puede derivar en problemas operativos significativos, como inconsistencias en los datos, redundancia destructiva de información, degradación en los tiempos de respuesta de las consultas o incluso la pérdida de integridad de registros relevantes (*Silberschatz et al., 2019*).

Para mitigar estos inconvenientes, el diseño conceptual y lógico se apoya en una serie de principios arquitectónicos fundamentales:

- Minimización de la redundancia de datos: Se evita la duplicación innecesaria de información a través de procesos de normalización. Esto permite organizar los datos en diferentes tablas según su naturaleza (tales como libros, usuarios, préstamos y sanciones), garantizando un almacenamiento óptimo y una mayor coherencia en la información (*Elmasri & Navathe, 2016*).

- Integridad de los datos: Asegura que la información almacenada en el back-end sea precisa, consistente y confiable a lo largo del tiempo. Esto se logra mediante la implementación de restricciones de dominio, claves primarias (PK) y claves foráneas (FK), así como reglas de validación en los formularios que evitan errores humanos en la captura y manipulación de los registros.
  
- Modelado de relaciones relacionales: Permite vincular las tablas de manera lógica y estructurada. Por ejemplo, la entidad transaccional de préstamos coexiste asociada simétricamente tanto a un usuario como a un libro, lo que facilita el seguimiento de los materiales, la identificación exhaustiva de los responsables y el control riguroso de las fechas de vencimiento (*Coronel & Morris, 2019*).
  
- Optimización del rendimiento: Consiste en estructurar la base de datos para permitir el acceso rápido y eficiente a la información, incluso ante un incremento en el volumen de los registros. Para ello, se emplean índices selectivos y sentencias de consulta optimizadas que reducen los tiempos de respuesta del sistema en pantalla.
  
- Escalabilidad y adaptabilidad: Capacidad de la arquitectura de software para absorber nuevas necesidades operativas sin requerir una reestructuración completa del modelo de datos. En el entorno bibliotecario, esto asegura la futura incorporación de recursos digitales, repositorios multimedia o la integración nativa con plataformas académicas externas (*Sommerville, 2016*).
  
- Seguridad y confidencialidad: Mecanismos orientados a proteger la información contra accesos no autorizados, modificaciones maliciosas o pérdidas accidentales. Involucra el diseño de controles de acceso basados en roles (RBAC), la planificación de respaldos periódicos (backups) y políticas estrictas para la gestión de datos sensibles.

Desde una perspectiva integral, la correcta estructuración de la base de datos se justifica plenamente porque permite cumplir con los objetivos estratégicos del sistema: organizar la información con rigor técnico, agilizar el acceso al acervo, optimizar los flujos de trabajo en ventanilla y proveer datos confiables para apoyar la toma de decisiones dentro de la institución.

## El enfoque del Diseño Centrado en el Usuario (DCU)

El diseño de la base de datos para la biblioteca está estrechamente vinculado con los requerimientos específicos de sus actores, adoptando los principios metodológicos del Diseño Centrado en el Usuario (DCU). Este enfoque de la ingeniería de software establece que el desarrollo de todo sistema de información debe ejecutarse considerando prioritariamente a las personas que interactuarán con la plataforma, alineando la arquitectura del software con sus objetivos, habilidades operativas, limitaciones y el contexto institucional en el que operan (*Sommerville, 2016*).

En el entorno de la biblioteca, los usuarios se clasifican en dos grupos o perfiles de rol principales, cada uno con requerimientos funcionales específicos que la base de datos debe resolver de manera diferenciada:

Personal Administrativo (Bibliotecarios): Sus necesidades se enfocan en el control, la gestión eficiente y la consistencia de la información. Requieren interfaces operativas que les permitan ejecutar transacciones de altas, bajas y modificaciones (ABM) de materiales, administrar los registros de usuarios, controlar el flujo de préstamos y devoluciones, y generar reportes analíticos para la toma de decisiones. El diseño del back-end responde a este perfil mediante una estructura relacional normalizada que simplifica la manipulación de los datos y mitiga la introducción de errores humanos o inconsistencias en el inventario (*Pressman & Maxim, 2020*). Asimismo, el personal requiere acceder de forma inmediata a vistas consolidadas, como el estado de circulación de los libros o las alertas de usuarios morosos, lo cual se resuelve mediante consultas de selección indexadas que optimizan la productividad del personal en ventanilla.

Usuarios Finales (Lectores, Estudiantes e Investigadores): Este grupo demanda principalmente usabilidad, rapidez y transparencia en el acceso a la información. Sus requerimientos se centran en la consulta interactiva del catálogo digital para verificar la disponibilidad matemática de las obras, identificar su ubicación topográfica exacta dentro de las estanterías, conocer el estado del ejemplar (disponible, prestado o en mantenimiento) y revisar de forma personalizada su estado de cuenta institucional. El diseño centrado en este perfil prioriza la velocidad de respuesta del motor de búsqueda y la claridad de los outputs en pantalla, elevando la experiencia de usuario dentro de la plataforma.

La convergencia de estas necesidades demuestra que una base de datos bien estructurada no solo actúa como un almacén estático de registros, sino como una herramienta dinámica que equilibra la eficiencia técnica del administrador con la satisfacción y accesibilidad del usuario final, cumpliendo con los estándares de calidad de las interfaces modernas (*Microsoft, 2023*).

## Fundamentos del Análisis y Diseño de Sistemas (SAD)

El modelado y la estructuración de una base de datos orientada a entornos bibliotecarios se rigen bajo los axiomas del Análisis y Diseño de Sistemas (SAD). Esta disciplina de la ingeniería de software se consolidó hacia la segunda mitad del siglo XX con el objetivo de proveer soluciones metodológicas ante la complejidad operacional de las organizaciones modernas y la necesidad de gestionar flujos masivos de datos sin perder el control de los activos informativos (*Sommerville, 2016*).

Bajo este marco teórico, un sistema de información se concibe como una infraestructura integrada por componentes tecnológicos, humanos y procesales interrelacionados, los cuales operan de forma sinérgica para alcanzar un fin institucional *determinado* (*Pressman & Maxim, 2020*). En el caso específico del CECyT 5, el sistema no se restringe únicamente al contenedor físico del back-end en Microsoft Access; por el contrario, abarca la totalidad de las reglas de negocio y flujos de trabajo automatizados que permiten la administración del catálogo, el control riguroso de préstamos y devoluciones, la gestión del padrón de usuarios y la generación estratégica de reportes ejecutivos.

Por lo tanto, el diseño de la base de datos representa una fase de máxima criticidad dentro del ciclo de vida del desarrollo clásico de sistemas. Es en esta etapa donde se transforma el análisis abstracto de requerimientos en especificaciones lógicas concretas: tablas, tipos de datos, restricciones de integridad y relaciones cardinales (*Elmasri & Navathe, 2016*).

Omitir una planificación rigurosa o prescindir de las metodologías de diseño en esta fase compromete el rendimiento total de la plataforma, provocando anomalías severas como la redundancia destructiva de registros, la pérdida de consistencia relacional (datos huérfanos), tiempos de respuesta elevados en las interfaces de consulta y vulnerabilidades en la seguridad del repositorio de datos (*Silberschatz et al., 2019*).

## Principios de estructura, integridad y seguridad

Para prevenir anomalías operacionales y fallos estructurales, el diseño conceptual del repositorio de datos se apoya en un conjunto de principios fundamentales de la ingeniería de software. Estos lineamientos garantizan que el sistema no solo funcione de manera óptima en el presente, sino que mantenga su estabilidad a lo largo de su ciclo de vida (*Pressman & Maxim, 2020*).

- **Estructuración y Mitigación de Redundancia:** En primer lugar, la arquitectura del sistema busca minimizar la redundancia de datos, evitando la duplicación innecesaria de cadenas informativas. Esto se logra formalmente mediante el proceso de normalización, el cual distribuye y organiza los atributos en tablas lógicas independientes de acuerdo con su naturaleza y su entidad (por ejemplo: libros, usuarios, préstamos y devoluciones), garantizando un uso eficiente del almacenamiento físico y una coherencia absoluta en todo el repositorio relacional (*Elmasri & Navathe, 2016*).
- **Integridad y Coherencia Referencial:** En segundo lugar, se prioriza la integridad de los datos, directriz que asegura que la información almacenada en el back-end permanezca precisa, consistente y confiable ante cualquier transacción. Su implementación técnica exige la configuración estricta de restricciones de dominio, asignación de claves primarias (PK) para la identificación unívoca de registros, definición de claves foráneas (FK) para el enlace relacional y el establecimiento de reglas de validación en la capa de interfaz. En el contexto de la biblioteca del CECyT 5, este factor es indispensable para asegurar el control preciso del acervo y erradicar inconsistencias administrativas (*Silberschatz et al., 2019*).
- **Modelado de Relaciones Cardinales:** Derivado de la integridad, la correcta definición de los vínculos entre tablas permite unificar la información de manera lógica. Al estructurar relaciones donde una entidad transaccional (como el préstamo) coexiste asociada simultáneamente a las entidades maestras de un usuario y un libro, se facilita el rastreo de los materiales, la identificación inmediata de los alumnos responsables y la auditoría automática de las fechas de vencimiento.
- **Optimización, Escalabilidad y Seguridad:** Finalmente, el diseño contempla la optimización del rendimiento en las consultas, permitiendo un acceso rápido a los índices aun cuando el volumen de registros incrementa *exponencialmente* (*Coronel & Morris, 2019*). Esta cualidad se complementa con la escalabilidad, la cual faculta a la base de datos para absorber futuras necesidades institucionales —como la adición de módulos para recursos digitales, audiolibros o repositorios multimedia— sin requerir una reestructuración de las tablas base. Asimismo, se integran políticas de seguridad lógica mediante el diseño de mecanismos de autenticación, control de acceso perimetral basado en roles y la planificación de respaldos periódicos (backups) contra pérdidas accidentales de información.

## Usabilidad y validación del sistema

El diseño centrado en el usuario exige un análisis metodológico del contexto de uso del sistema. Este proceso implica identificar con precisión los escenarios operativos en los que se

desplegará la plataforma, la frecuencia de las transacciones y las condiciones técnicas del entorno de hardware. Este diagnóstico exhaustivo permite deducir requisitos de software mucho más precisos y diseñar soluciones arquitectónicas que se alineen con las necesidades reales de la organización. Para garantizar la viabilidad del sistema, se promueve la participación activa de los actores a lo largo del ciclo de desarrollo mediante técnicas de recolección de datos como encuestas, entrevistas estructuradas y dinámicas de retroalimentación continua (Sommerville, 2016).

Bajo este enfoque, la usabilidad se consolida como un atributo de calidad de software crítico, ya que determina la facilidad, eficiencia y velocidad con la que los usuarios interactúan con la interfaz. Un sistema usable y ergonómico faculta la ejecución de tareas de manera intuitiva, incorporando principios universales de accesibilidad, legibilidad en la presentación visual de los datos y satisfacción general del operador. El aseguramiento de estos estándares se logra combinando técnicas de análisis de requerimientos, modelado conceptual de datos, diseño de prototipos rápidos de baja y alta fidelidad, y la ejecución sistemática de pruebas funcionales y de aceptación de usuario (UAT) (Pressman & Maxim, 2020).

## IDENTIDAD VISUAL

Para comenzar a definir los parámetros que utilizaríamos en el diseño de nuestra aplicación llevamos a cabo una reunión entre las personas que conforman el equipo y llegamos a las conclusiones de lo que consideramos más adecuado para la comodidad del usuario y que sería más cómodo y accesible para estos.

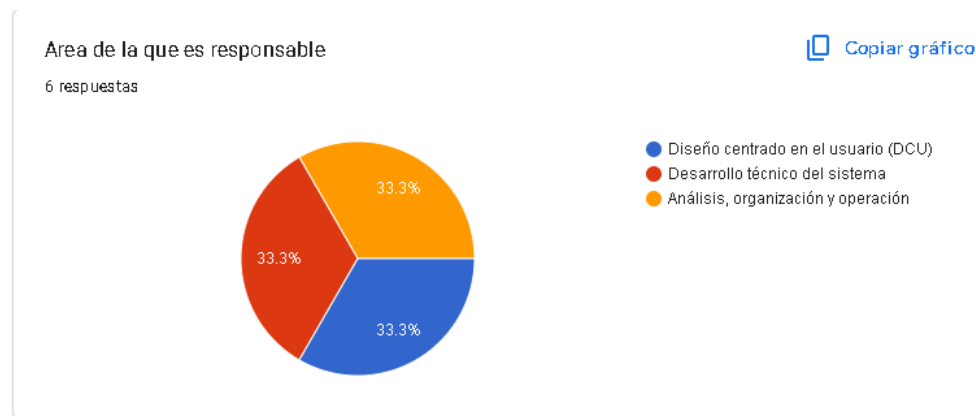


Tabla 18\_DefinirResponsibilidades

## DEFINIR PALETA DE COLORES

Los parámetros definidos para establecer la paleta de color se baso en lo que era más adecuado y relacionados con nuestro tema para crear una identidad visual con la que los usuarios se sintieran cómodos y relacionados.

- **Verde**

El color azul transmite salud, esperanza, te hace sentir conectado con la naturaleza y con la vida; lo cual genera comodidad.

- **Café y tonos dorados**

Estos colores representan los libros físicos, el conocimiento tradicional y la experiencia académica. Además, generan una combinación visual equilibrada entre lo clásico y lo tecnológico.

- **Blanco**

El fondo blanco simboliza claridad, simplicidad y limpieza visual, permitiendo que el logotipo tenga una apariencia moderna y fácil de identificar.

¿Qué colores consideras más apropiados para la aplicación? \*

Opción 1



Opción 2



Opción 3



Opción 4



Otr *Ilustración 1\_PaletaDeColores*

Al llevar a cabo la reunión el equipo concluyo que la paleta de color adecuada en esta ocasión eran verdes, blancos y cafés.

¿Qué colores consideras más apropiados para la aplicación?

6 respuestas

[Copiar gráfico](#)

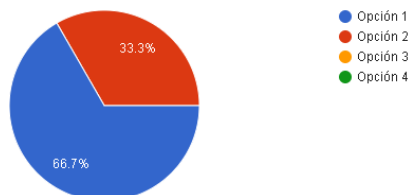
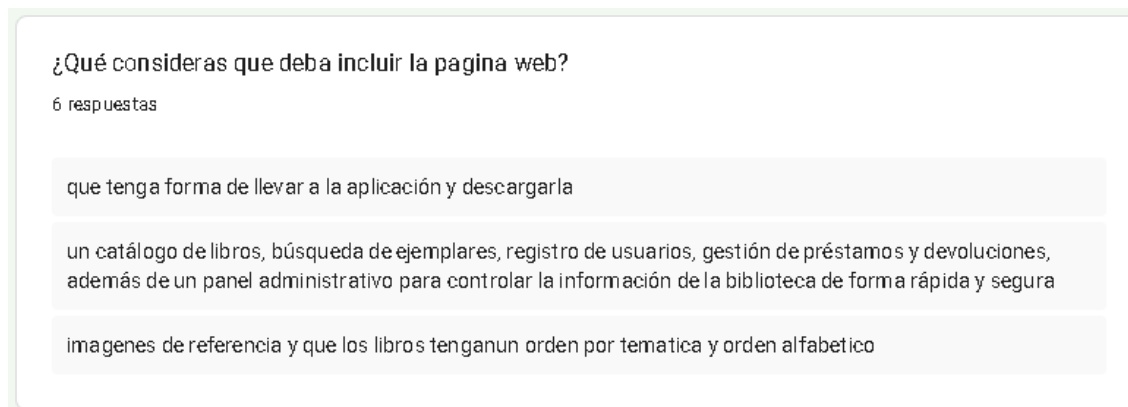


Tabla 19\_Colores

## PAGINA WEB

Después de reuniones se llego a la conclusión que la mejor forma de llegar a una mayor cantidad de personas era si creábamos una

página web, y al final concluimos que era lo más relevante a colocar en esta.



*Ilustración 2\_PaginaWeb*

## LOGOTIPO

Las opciones para nuestros logotipos fueron propuestas presentadas por los integrantes del equipo, que después de reuniones terminamos en dos opciones de la cual en la última reunión realizada logramos llegar a la conclusión de cuál de estos representaba mejor nuestra aplicación.

- **Libro abierto**

El libro abierto representa el conocimiento, el aprendizaje y el acceso a la información. Simboliza la función principal del sistema, que es almacenar y organizar contenido bibliográfico de manera digital.

- **Circuitos tecnológicos**

Los circuitos integrados alrededor del libro representan la tecnología, las bases de datos y la conectividad digital. Estos elementos reflejan el uso de herramientas informáticas para facilitar la búsqueda, administración y consulta de información.

- **Forma circular**

La estructura circular del logotipo simboliza organización, continuidad y unidad. También transmite estabilidad y un funcionamiento constante dentro del sistema de biblioteca digital.

- **Engrane inferior**

El engrane representa el funcionamiento interno del sistema y los procesos automatizados que permiten administrar correctamente los datos bibliográficos.

¿Qué logotipo consideras que va mejor con nuestro sistema? \*

Opción 1



Opción 2



Otros: .....

Ilustración 3\_Logotipo

Para finalizar en la conclusión de que la primera opción era la más relacionada con nuestra app.



Tabla 20\_Logotipo

Y ya con la información anterior pudimos crear las versiones finales de nuestro logotipo.



Ilustración 5\_LogotipoColor



Ilustración 4\_LogotipoBlancoyNegro

## DEFINIR PARAMETROS DE LA APLICACIÓN.

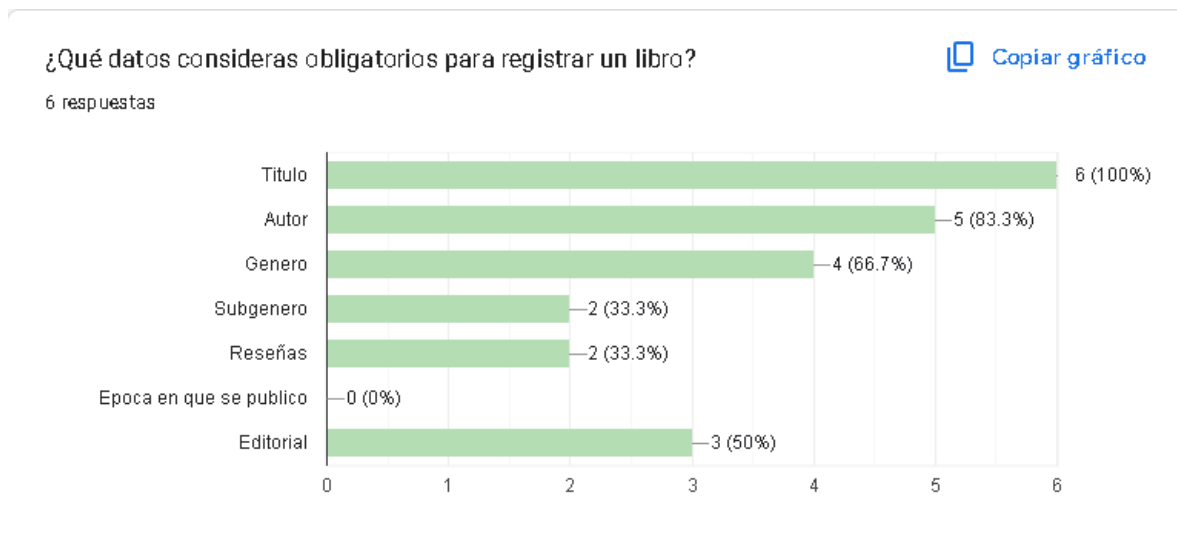


Tabla 21\_Registro

### ¿Cómo clasificas actualmente los libros?

 Copiar gráfico

6 respuestas

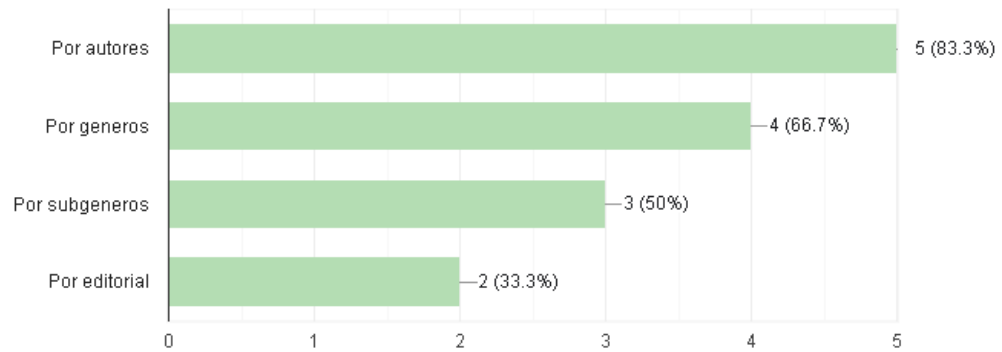


Tabla 22\_Clasificación

### ¿Qué filtros de búsqueda consideran indispensables para los usuarios?

6 respuestas

busqueda por autor y por generos

título del libro, autor, género, editorial y año de publicación, ya que permiten a los usuarios encontrar información de manera rápida y eficiente

por generos y por autores

por generos

Tabla 23\_Filtros

## Arquitectura básica del sistema

La arquitectura básica del sistema de administración de la biblioteca define la organización estructural, la distribución lógica y los protocolos de comunicación entre los diversos componentes de software para garantizar un funcionamiento óptimo, íntegro y fluido. Dentro de la etapa de diseño, la especificación arquitectónica constituye un pilar crítico, ya que establece los cimientos técnicos necesarios para facilitar las fases subsecuentes de codificación, mantenimiento correctivo y escalabilidad del *sistema* (Sommerville, 2016).

Para la solución planteada, se implementa un patrón de Arquitectura en Tres Capas (Presentation, Application, and Data Layers), modelo ampliamente utilizado en la ingeniería de software debido a su alta eficiencia en el acoplamiento débil y la clara separación de responsabilidades. Esta infraestructura divide las operaciones del sistema en tres niveles jerárquicos independientes:

- **Capa de Presentación (Interfaz de Usuario - UI)**: Constituye el componente periférico con el que interactúan directamente los diferentes perfiles de usuario. En el entorno de Microsoft Access, se compone de formularios interactivos, cuadros combinados y menús de navegación dinámicos diseñados bajo un enfoque intuitivo y ergonómico. A través de esta interfaz, el personal administrativo ejecuta transacciones operativas (altas de libros, control de inventario y emisión de reportes), mientras que los usuarios finales realizan búsquedas en el catálogo y verifican la disponibilidad de ejemplares en tiempo real.
- **Capa de Lógica de Negocio (Capa de Aplicación)**: Representa el núcleo algorítmico de la plataforma y se encarga de procesar las transacciones de acuerdo con las reglas de negocio preestablecidas. Desarrollada mediante módulos de programación en VBA (Visual Basic for Applications) y macros automatizadas, esta capa valida las credenciales de acceso, computa el estado dinámico de los libros (disponible/prestado), calcula las fechas límite de devolución y procesa las alertas del sistema. Actúa como un intermediario regulador entre la capa de presentación y el repositorio físico, asegurando que ninguna operación viole las restricciones operativas de la biblioteca (Pressman & Maxim, 2020).
- **Capa de Datos (Almacenamiento - Back-end)**: Es el nivel responsable del almacenamiento persistente de toda la información del sistema. Estructurada en un motor relacional normalizado (Microsoft Access), resguarda las tablas maestras y transaccionales relacionadas con el acervo, usuarios, préstamos, sanciones e históricos de movimientos. Su diseño se optimiza mediante la indexación de campos

clave para agilizar la velocidad de las consultas (SELECT) y garantizar la integridad referencial de los registros (Silberschatz et al., 2019).

- **Módulos Complementarios e Interoperabilidad:** Más allá de los tres niveles principales, la arquitectura de la plataforma integra módulos transversales periféricos para robustecer la estabilidad y la seguridad del sistema:
- **Módulo de Seguridad Lógica:** Gobierna los procesos de autenticación y control de acceso basados en roles (RBAC), mitigando los riesgos de manipulación de registros críticos por parte de actores no autorizados.
- **Módulo de Reportes y Analítica:** Ejecuta consultas de agregación y procesamiento de datos para la generación de informes formales y estadísticas descriptivas de movimientos útiles para la toma de decisiones.
- **Módulo de Respaldo y Contingencia:** Provee funciones para la clonación y generación de copias de seguridad periódicas (backups) del archivo de la base de datos, resguardando la continuidad del servicio ante fallas físicas o lógicas.

*De acuerdo con Elmasri y Navathe (2016)*, una arquitectura de software de alta calidad debe prever la escalabilidad horizontal y vertical. Por esta razón, el desacoplamiento en tres capas implementado asegura que, en una fase futura de desarrollo, el sistema pueda migrar de forma transparente hacia una arquitectura distribuida (como un servidor de bases de datos robusto tipo SQL Server) o expandirse para permitir el acceso remoto vía web y dispositivos móviles sin necesidad de reconstruir la lógica de programación o las interfaces del usuario. La fluidez en el intercambio de información entre estos componentes consolida una plataforma flexible, segura y de alta disponibilidad, diseñada bajo estándares profesionales.

### Proceso del sistema

Dentro de la etapa de diseño, la definición y modelado de los procesos del sistema especifican el comportamiento dinámico y las transiciones de estado que el software ejecutará de manera automatizada. El sistema propuesto para la biblioteca del CECyT 5 unifica las operaciones administrativas y de consulta en una sola plataforma digital, erradicando los vicios del registro manual tradicional y optimizando la integridad operativa a través de tres macroprocesos interconectados (Sommerville, 2016):

#### 1. Macroproceso de Gestión de Inventario y Catálogo

- **Registro Automatizado de Obras:** Controla la inserción de nuevos materiales en la tabla TBL\_Libros. Al capturar los atributos esenciales (Título, Autor, ISBN, Editorial, Año, Género y Existencias), la capa lógica dispara funciones

de validación en tiempo real para impedir la duplicidad de registros o la indexación de datos incompletos.

- **Mantenimiento y Actualización Lógica del Acervo:** Provee al administrador los mecanismos necesarios para modificar selectivamente los campos de las obras, actualizar el stock de ejemplares disponibles en sala o ejecutar la baja lógica de registros que han perdido relevancia o vigencia institucional, garantizando la confiabilidad del repositorio (*Silberschatz et al., 2019*).
- **Motor de Búsqueda Indexado:** Procesa consultas estructuradas permitiendo tanto a los administradores como a los estudiantes localizar libros de manera síncrona. El proceso ejecuta filtros avanzados por criterios múltiples (título, autor, género o palabras clave) y devuelve una matriz de resultados clara que indica la disponibilidad y ubicación física del material.

## 2. Macroproceso Transaccional y de Circulación

- La operación medular del flujo de la biblioteca, que comprende el control de salidas, retornos y la recolección de experiencias de lectura, se encuentra estandarizada visualmente bajo el siguiente modelo lógico de negocio:

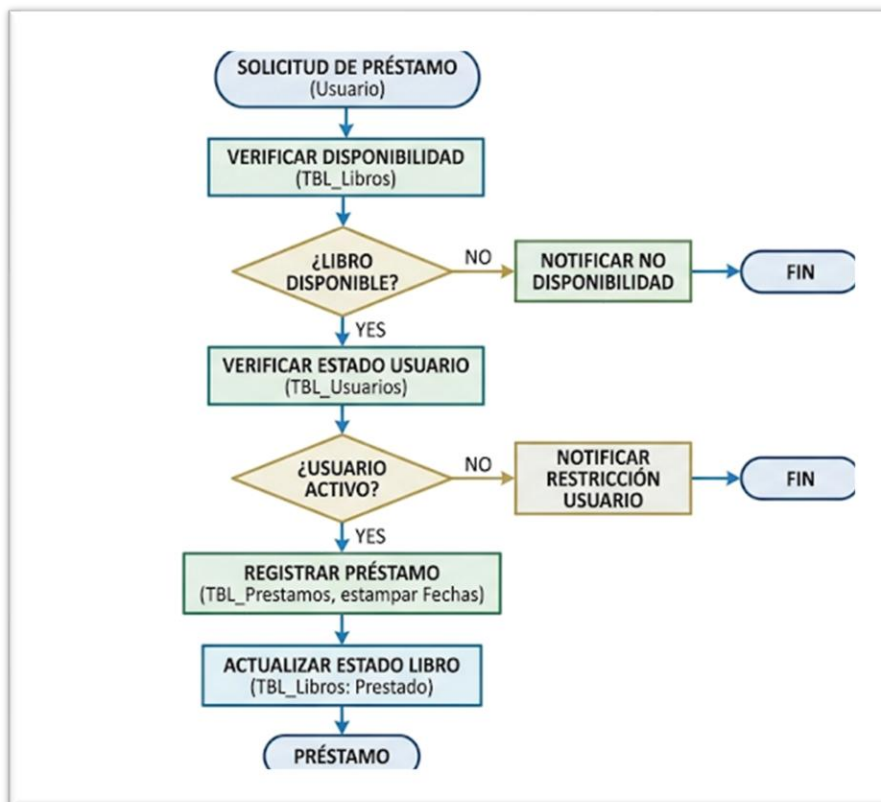


Figura X Diagrama de flujo lógico para el control de circulación y retroalimentación de usuarios. Nota. El esquema detalla las bifurcaciones y reglas de integridad que gobiernan los préstamos, las devoluciones y el almacenamiento de valoraciones en la base de datos.

Como se observa en el modelo, este macroproceso abarca:

Tabla 24\_DIAGRAMA-DE-FLUJO\_

- **Gestión y Validación de Préstamos**: Coordina la salida supervisada de los libros. Antes de autorizar cualquier movimiento, el sistema inspecciona las restricciones de integridad relacional: comprueba que el ejemplar no esté asignado a otro servicio y que el alumno no cuente con penalizaciones activas. Al aprobarse, se genera el registro en TBL\_Prestamos vinculando al usuario con la obra y fijando los plazos de entrega (*Coronel & Morris, 2019*).
- **Control de Devoluciones y Alertas**: Registra el reingreso de los materiales para actualizar de inmediato el inventario y permitir que el libro sea elegible para nuevas solicitudes. Este proceso audita las fechas límites de forma automatizada, calculando retrasos temporales y disparando alertas o recordatorios administrativos en caso de incidencias.
- **Procesamiento de Instrumentos de Valoración (Encuestas)**: Actúa como una característica distintiva de la plataforma. Tras concluir una devolución, el sistema despliega un formulario interactivo en la capa de presentación para que el usuario califique la obra y emita comentarios. Estos datos se almacenan de forma persistente en tablas dedicadas.
- **Análisis Cuantitativo de Preferencias**: El módulo analítico procesa las respuestas de las valoraciones para estructurar reportes estadísticos automatizados. Mediante consultas de agregación, identifica las tendencias del acervo (obras más populares y géneros más solicitados), proveyendo información estratégica para la adquisición de nuevos materiales y la optimización del servicio (*Pressman & Maxim, 2020*).

### 3. Macroproceso de Administración, Seguridad y Contingencia

- Registro y Padrón de Usuarios: Administra de forma centralizada las hojas de datos de la comunidad escolar (nombre, identificación, contacto e historial transaccional), facilitando la trazabilidad de las actividades dentro de la biblioteca.
- Control de Acceso Perimetral basado en Roles (RBAC): Mecanismo de seguridad que restringe el uso de las funciones del sistema de acuerdo con el perfil del usuario autenticado. Los bibliotecarios heredan permisos totales de edición, modificación y control de inventario, mientras que los lectores quedan limitados estrictamente a la consulta del catálogo y al llenado de valoraciones.
- Respaldo y Recuperación de Datos (Backup & Recovery): Ejecuta copias de seguridad redundantes y periódicas de la base de datos de Microsoft Access en segundo plano. Este proceso mitiga el impacto de fallas técnicas o lógicas en el hardware, asegurando la tolerancia a fallos, la protección de los activos de información y la continuidad del servicio bibliotecario a largo plazo (*Elmasri & Navathe, 2016*).

## Principios técnicos para la integridad de la información

El aseguramiento de un entorno transaccional eficiente, robusto y libre de anomalías operacionales dentro de la biblioteca del CECyT 5 exige que el modelo de datos se rija por directrices formales de la ingeniería de bases de datos. Estos principios garantizan la calidad, consistencia y persistencia de los activos de información a lo largo del ciclo de vida del software (*Pressman & Maxim, 2020*).

**Abstracción Algorítmica y Normalización:** La normalización constituye un proceso algebraico-estructural indispensable para transformar modelos conceptuales complejos en esquemas relacionales óptimos. Al segregar los atributos de la biblioteca en tablas independientes ordenadas por su naturaleza (como libros, usuarios y movimientos transaccionales), se minimiza drásticamente la redundancia de datos. Este aislamiento de entidades impide la aparición de anomalías durante los eventos de inserción, actualización y borrado, agilizando el mantenimiento predictivo del back-end (*Date, 2003*).

**Integridad Referencial y Restricciones de Dominio:** Este principio valida que el estado de la base de datos sea preciso, coherente y confiable en todo momento. Su aplicación práctica se gestiona mediante el acoplamiento estricto de claves primarias (Primary Keys) y claves foráneas (Foreign Keys). Bajo esta arquitectura, el motor relacional de Microsoft Access impide la ejecución de transacciones inválidas; por ejemplo, el sistema denegará la creación de un nuevo registro en la tabla de préstamos si el identificador del usuario o el ISBN del libro no coexisten previamente en sus respectivas tablas maestras, erradicando los registros huérfanos y los fallos de consistencia (*Silberschatz et al., 2019*).

**Optimización y Eficiencia de Consultas:** El diseño físico contempla la indexación estratégica de los campos de mayor concurrencia para acelerar los tiempos de respuesta del procesador de consultas. En sistemas de gestión documental con un flujo continuo de operaciones, estructurar sentencias SQL optimizadas previene la degradación del rendimiento de la memoria, garantizando que el usuario en ventanilla obtenga respuestas inmediatas en la interfaz de presentación (*Coronel & Morris, 2019*).

**Escalabilidad Estructural y Elasticidad:** Representa la capacidad intrínseca del modelo de datos para absorber nuevas demandas de almacenamiento o integraciones funcionales sin obligar a una reestructuración o rediseño de las tablas base. En el entorno del proyecto, este enfoque flexible prepara la arquitectura del software para asimilar la futura automatización de catálogos

digitales, repositorios multimedia independientes o la interconexión nativa con API de plataformas académicas externas (Sommerville, 2016).

## El Diseño Centrado en el Usuario (DCU)

Más allá de la rigidez de los parámetros estructurales y lógicos del *back-end*, la viabilidad y el éxito operacional de un sistema de información radican en su capacidad intrínseca para resolver las necesidades reales de su entorno. En este sentido, la metodología del Diseño Centrado en el Usuario (DCU) establece que la conceptualización y desarrollo de los sistemas informáticos deben pivotar en torno a las características cognitivas, los objetivos operacionales y las limitaciones contextuales de las personas que interactúan de manera directa con las interfaces de la aplicación (Norman, 2013).

En el ecosistema tecnológico de la biblioteca del CECyT 5, se han modelado e identificado dos perfiles de rol o grupos de interés primarios, cuyas interacciones con la base de datos demandan aproximaciones de diseño de interfaz (*User Interface - UI*) claramente diferenciadas:

- **Personal Administrativo (Bibliotecarios):** Representa el perfil operativo de alta concurrencia transaccional. Este grupo requiere herramientas especializadas de control que agilicen la manipulación del repositorio mediante interfaces de captura despejadas, menús de navegación simplificados y disparadores automáticos de validación que actúen como un filtro de seguridad contra el error humano o la inserción de registros duplicados (Pressman & Maxim, 2020). Asimismo, el personal administrativo demanda la capacidad de extraer de forma inmediata vistas consolidadas e indicadores críticos —tales como la disponibilidad matemática de un volumen o el estatus moroso de una cuenta estudiantil—. La base de datos resuelve este requerimiento mediante la estructuración de consultas de agregación y vistas de reporte optimizadas que sirvan como herramientas de soporte para la toma de decisiones en ventanilla.
- **Usuarios Finales (Lectores e Investigadores):** Constituye el perfil de consulta interactiva. Su interacción con el sistema está orientada a la autogestión de la información, demandando altos índices de usabilidad, transparencia y velocidad de respuesta. Sus necesidades funcionales implican el rastreo síncrono del catálogo digital para verificar la existencia de obras, la localización topográfica exacta de los textos dentro de los estantes físicos y el acceso autónomo a su historial personalizado de préstamos. Diseñar interfaces bajo los estándares de este perfil implica construir flujos interactivos e intuitivos que erradiquen la frustración digital y maximicen la experiencia de usuario (UX) (Sommerville, 2016).

Metodológicamente, el despliegue del DCU exige también un diagnóstico profundo del contexto de uso; es decir, evaluar las variables físicas, de hardware y de red bajo las cuales operará la plataforma (tales como terminales en mostradores de atención, computadoras de consulta pública en salas de lectura o futuras interfaces de acceso remoto). Este análisis perimetral garantiza que la

solución final no solo sea técnicamente robusta, sino ergonómica y perfectamente adaptada al flujo de trabajo real de la institución.

## DESARROLLO

### ARQUITECTURA DEL SISTEMA

La arquitectura del sistema de información para la gestión de la biblioteca ha sido diseñada bajo un enfoque moderno, escalable y orientado a servicios, con el propósito de garantizar eficiencia, seguridad, disponibilidad y facilidad de mantenimiento. *Según IBM (2025b)*, este sistema se concibe como una plataforma integral que permite administrar los recursos bibliográficos, los usuarios, los préstamos, las devoluciones, las reservas y los reportes administrativos, integrando múltiples componentes tecnológicos que trabajan de manera coordinada.

En su concepción general, el sistema adopta una arquitectura multicapa, la cual separa claramente las responsabilidades del sistema en diferentes niveles: la capa de presentación, la capa lógica o de negocio y la capa de datos. Esta división no solo facilita el desarrollo y mantenimiento del sistema, sino que también permite una mayor flexibilidad para futuras mejoras o ampliaciones, siguiendo las mejores prácticas de diseño de sistemas en ingeniería de software (*GeeksforGeeks, 2025b; Microsoft, 2025*).

#### CAPA DE PRESENTACIÓN

La capa de presentación constituye el punto de interacción entre el sistema y los usuarios finales. En el contexto del sistema de gestión de biblioteca, esta capa está compuesta por una interfaz web responsiva accesible desde navegadores modernos, así como por una posible aplicación móvil diseñada para dispositivos inteligentes (*MDN Web Docs, 2025*). Esta estructura permite que el acceso a la información sea multiplataforma y eficiente.

La interfaz ha sido diseñada bajo principios de usabilidad y experiencia de usuario, asegurando que tanto bibliotecarios como usuarios puedan navegar fácilmente por el sistema. Como señala *Shneiderman (2016)*, es crucial implementar estrategias efectivas de interacción

humano-computadora para reducir la curva de aprendizaje. En este sentido, se han integrado formularios dinámicos, menús intuitivos y paneles de control personalizados. Por ejemplo, los bibliotecarios tienen acceso a funciones administrativas como el registro de libros y control de préstamos, mientras que los usuarios finales se enfocan en la búsqueda y reserva de ejemplares (*TutorialsPoint, 2025*).

Esta capa se comunica con la capa de negocio mediante servicios web (*API REST*), utilizando protocolos seguros como HTTPS, lo que garantiza la protección de la información transmitida. Este método de comunicación asegura que la interfaz permanezca desacoplada de la lógica del servidor, permitiendo una mayor seguridad y escalabilidad en el intercambio de datos (*IBM, 2025a; Microsoft, 2023*).

## CAPA LÓGICA O DE NEGOCIO

La capa lógica constituye el núcleo del sistema, ya que en ella se implementan todas las reglas de negocio y procesos que definen el funcionamiento de la biblioteca. De acuerdo con IBM (2025b), esta capa actúa como intermediaria, procesando los comandos, realizando evaluaciones lógicas y moviendo datos entre las capas de presentación y de datos. Aquí se gestionan aspectos críticos como las políticas de préstamo, los límites de libros por usuario, las sanciones por retrasos, la disponibilidad de ejemplares y la validación de datos.

Esta capa está desarrollada utilizando lenguajes de programación robustos y marcos de trabajo empresariales que permiten estructurar el código de manera modular (*Microsoft, 2025*). La modularidad es esencial para la escalabilidad, permitiendo que cada funcionalidad del sistema esté encapsulada en componentes independientes, tales como:

- Módulo de gestión de usuarios: Control de perfiles y permisos.
- Módulo de catálogo bibliográfico: Organización según estándares de bases de datos (*Elmasri y Navathe, 2016*).
- Módulo de reportes y estadísticas: Basado en la visualización efectiva de información cuantitativa (*Tufte, 2001*).

La lógica del sistema asegura que todas las operaciones se realicen de manera consistente y conforme a las políticas definidas por la biblioteca. *Como explican Coronel y Morris (2019)*, la integridad de los procesos depende de reglas de validación estrictas; por ejemplo, antes de permitir un préstamo, el sistema verifica automáticamente si el usuario tiene sanciones activas, si ha alcanzado el límite de préstamos o si el libro está disponible en el inventario actual (*GeeksforGeeks, 2025a*).

### CAPA DE DATOS

La capa de datos es responsable del almacenamiento, recuperación y gestión de la información del sistema. Para este propósito, se ha implementado un sistema gestor de bases de datos relacional (RDBMS), como MySQL, PostgreSQL o Microsoft SQL Server, el cual permite manejar grandes volúmenes de información de manera estructurada y eficiente (*Coronel y Morris, 2019*). *Según IBM (2025b)*, esta capa es fundamental para asegurar que los datos persistan independientemente de los procesos lógicos del servidor.

La base de datos está diseñada siguiendo principios de normalización, lo que evita redundancias y asegura la integridad referencial de los datos. *Como explican Elmasri y Navathe (2016)*, el proceso de normalización es crítico para minimizar anomalías de actualización y optimizar el almacenamiento. Entre las principales tablas del sistema se encuentran:

- Usuarios
- Libros
- Ejemplares

Cada tabla está relacionada mediante claves primarias y foráneas, permitiendo establecer vínculos lógicos entre los diferentes elementos del sistema (*Microsoft, 2023*). Por ejemplo, un préstamo está asociado a un usuario y a un ejemplar específico de un libro, garantizando la trazabilidad de cada recurso bibliográfico.

Además, se han implementado mecanismos de respaldo automático y recuperación ante fallos, siguiendo las mejores prácticas de seguridad de la información. *Según Amazon Web Services (2025)*, estas estrategias de recuperación son vitales para garantizar la disponibilidad continua y proteger la integridad de la base de datos ante posibles contingencias técnicas.

El sistema incorpora principios de Arquitectura Orientada a Servicios (*Service-Oriented Architecture, SOA*), lo que significa que sus funcionalidades operan de manera modular y están expuestas como servicios independientes que pueden ser consumidos por diferentes clientes. Esta aproximación arquitectónica facilita la interoperabilidad y la integración con plataformas externas, tales como entornos virtuales de aprendizaje, sistemas de pago institucional o aplicaciones móviles (*Sommerville, 2016*).

Bajo este modelo, cada componente o servicio de la base de datos está diseñado para ser autónomo, reutilizable y escalable. Esto asegura un bajo acoplamiento entre los módulos del software, permitiendo que la plataforma evolucione, reciba mantenimiento o incorpore nuevas funciones de forma aislada sin afectar el funcionamiento global ni comprometer la disponibilidad del sistema bibliotecario (*Pressman & Maxim, 2020*).

### Seguridad del sistema

La arquitectura del sistema incluye múltiples mecanismos de seguridad diseñados para proteger la integridad de los datos, salvaguardar la confidencialidad de la información institucional y garantizar un acceso estrictamente controlado a la plataforma (*Silberschatz et al., 2019*). Entre los mecanismos implementados para mitigar riesgos y vulnerabilidades se encuentran los siguientes:

- Autenticación de usuarios: Validación de la identidad de cada actor mediante el ingreso de credenciales seguras antes de permitir el acceso al entorno del sistema.
- Autorización basada en roles (RBAC): Asignación de privilegios y restricciones diferenciadas según las responsabilidades del actor dentro de la institución, distinguiendo claramente los niveles de acceso para el administrador, el bibliotecario y el usuario lector (*Sommerville, 2016*).
- Encriptación de datos sensibles: Cifrado de campos críticos dentro del repositorio de datos para evitar que la información sea legible en caso de extracciones no autorizadas.
- Uso de protocolos seguros: Implementación de HTTPS para asegurar que la transferencia de datos entre los clientes y el servidor viaje de manera cifrada a través de la red.
- Protección contra ataques comunes: Incorporación de validaciones y filtros en las capas del software para prevenir amenazas críticas de seguridad, tales como la inyección de código malicioso (SQL Injection) y secuencias de comandos en sitios cruzados (*Cross-Site Scripting, XSS*) (*Pressman & Maxim, 2020*).

Además, el sistema incorpora un módulo de auditoría que registra de manera persistente todas las actividades relevantes en bitácoras de eventos (logs). Esta función permite realizar un seguimiento cronológico de las operaciones ejecutadas, facilitando la detección temprana de

anomalías operativas y garantizando un control de seguridad robusto dentro del entorno bibliotecario.

## PROCESOS DEL SISTEMA

El sistema de gestión de biblioteca integra una serie de procesos fundamentales que permiten el funcionamiento eficiente, consistente y coordinado de todas sus operaciones. Estos procesos han sido definidos metodológicamente con base en las necesidades reales y los flujos de trabajo de una biblioteca moderna, automatizando tareas críticas que tradicionalmente se realizaban de forma manual (*Sommerville, 2016*).

La correcta delimitación de estos procesos asegura que las transacciones lógicas dentro de la base de datos se ejecuten bajo reglas de negocio automatizadas. Esto disminuye drásticamente los tiempos de espera en ventanilla y erradica los errores humanos asociados al registro analógico en papel, cumpliendo con los estándares de calidad y optimización operativa en el desarrollo de sistemas de información (*Pressman & Maxim, 2020*).

### Proceso de registro de usuarios

El proceso de registro de usuarios permite dar de alta a nuevos miembros dentro del ecosistema de la biblioteca, asegurando el control y la trazabilidad de los accesos al acervo. Este proceso inicia formalmente cuando el usuario proporciona sus datos personales e institucionales, los cuales son sometidos a validaciones lógicas por parte del sistema antes de ser almacenados de manera persistente en la base de datos (*Silberschatz et al., 2019*).

Una vez completado el registro de manera exitosa, el usuario recibe credenciales de acceso que le permiten interactuar con las interfaces correspondientes e incorporar su perfil a los servicios del sistema. Este flujo operativo incluye verificaciones automáticas de integridad para evitar la duplicidad de registros (mediante el control de claves primarias únicas) y garantizar la calidad, consistencia y fiabilidad de la información almacenada en el repositorio institucional (*Pressman & Maxim, 2020*).

### Proceso de gestión del catálogo

El catálogo bibliográfico constituye el núcleo del sistema de información, ya que almacena de manera estructurada todos los metadatos y registros correspondientes a los libros disponibles en la institución. El proceso de gestión del catálogo otorga al personal bibliotecario las funciones necesarias para registrar nuevos libros (altas), actualizar información existente (modificaciones) y

depurar registros obsoletos (bajas), garantizando un control dinámico del acervo (*Sommerville, 2016*).

Bajo este flujo operativo, el sistema discrimina entre la entidad abstracta del libro y sus múltiples ejemplares físicos, lo que permite gestionar las existencias y la disponibilidad en estanterías de manera precisa dentro de la base de datos relacional. Asimismo, el proceso automatiza la clasificación de los materiales por categorías, autores, editoriales y palabras clave, estructurando los índices que posteriormente serán utilizados por el motor de búsqueda para agilizar las consultas de los usuarios (*Silberschatz et al., 2019*).

### Proceso de búsqueda de libros

El sistema incluye un motor de búsqueda avanzado que permite a los usuarios encontrar libros de manera rápida y eficiente dentro del catálogo digital. Este proceso operativo utiliza filtros indexados y criterios de selección específicos —tales como el título, autor, categoría o el International Standard Book Number (ISBN)— para realizar consultas directas sobre las tablas relacionales de la base de datos, optimizando los tiempos de recuperación de la información (*Silberschatz et al., 2019*).

Los resultados de la consulta se presentan al usuario de forma clara y estructurada a través de la interfaz gráfica. El sistema despliega dinámicamente los metadatos del libro junto con indicadores críticos en tiempo real, mostrando la disponibilidad matemática de los ejemplares físicos y su ubicación topográfica exacta dentro de las estanterías de la biblioteca, lo que mejora significativamente la experiencia del lector y reduce la carga operativa en los módulos de atención (*Sommerville, 2016*).

### Proceso de autenticación y control de acceso

Este proceso garantiza que solo los usuarios legítimamente autorizados puedan ingresar a la plataforma y manipular sus recursos. El flujo operativo contempla la validación estricta de credenciales de identidad en la capa de seguridad y la asignación dinámica de privilegios específicos en la base de datos de acuerdo con el rol institucional previamente configurado para el usuario. De este modo, se restringe la ejecución de operaciones críticas —como la modificación de catálogos o la eliminación de registros— únicamente al personal administrativo facultado para ello (*Pressman & Maxim, 2020*).

## Integración de procesos

Todos los procesos que integran el sistema de información se encuentran interconectados de manera relacional, lo que permite una gestión integral, centralizada y automatizada de la biblioteca. Ningún módulo opera de forma aislada; por ejemplo, la ejecución de un préstamo de material bibliográfico desencadena de manera simultánea verificaciones en la gestión de usuarios (validación de estado activo), consultas en el catálogo digital (reducción de existencias físicas del libro) y auditorías en el módulo de sanciones (comprobación de multas pendientes). Esta cohesión arquitectónica asegura la consistencia, integridad y coherencia de la información en todo el repositorio de datos (*Silberschatz et al., 2019*).

## IMPLEMENTACIÓN DE ENTRADAS Y SALIDAS

### ENTRADAS

La captura de datos constituye un elemento fundamental en cualquier sistema de información, ya que de su calidad depende la confiabilidad de los procesos y consultas posteriores. En el entorno de Microsoft Access, las entradas se controlan rigurosamente mediante formularios estructurados, reglas de validación en la capa de la base de datos y tipos de datos específicos, lo que permite minimizar errores y asegurar la integridad de la información desde el momento de su registro (*Silberschatz et al., 2019*). Con el fin de optimizar la experiencia del usuario y normalizar la estructura, las entradas se dividen en diferentes niveles según su propósito y frecuencia de uso:

Datos del Libro (Tabla Libros): En este nivel se almacena la información descriptiva principal del acervo. El campo ISBN, definido bajo el tipo de datos Texto corto, funciona como la clave primaria o un identificador único que impide la duplicidad de registros y acelera la indexación para la recuperación de información. Los campos Título y Autor se configuran con la propiedad Requerido: Sí, garantizando que ningún registro quede incompleto. Para el campo Género, se implementa un cuadro combinado (*ComboBox*) que restringe la entrada a una lista predefinida de valores; este control no solo unifica la consistencia de los datos, sino que agiliza la captura y mitiga los errores humanos de escritura (*Microsoft, 2023*). Finalmente, el campo Ubicación registra la posición topográfica exacta del ejemplar físico (estantería o sección) para su localización inmediata.

Datos de Control (Tabla Movimientos): Este nivel está orientado al seguimiento y control del estado dinámico del libro dentro del sistema. El campo Estado se configura como una lista de valores fijos (disponible, prestado, en mantenimiento) para monitorear el flujo de circulación. Las variables temporales, como la Fecha de adquisición o Fecha de movimiento, se registran mediante el tipo de datos Fecha/Hora acoplado a un selector de fecha nativo, asegurando un formato uniforme. Asimismo, el campo Valor estimado se define como tipo Moneda, lo que permite llevar un control financiero del inventario de la colección, representando los libros como activos de valor institucional (*Coronel & Morris, 2019*).

Captura Rápida mediante Automatización: Como estrategia de optimización orientada a la eficiencia operativa, el sistema integra el uso de lectores de códigos de barras USB para la entrada automatizada del ISBN. Debido a que estos dispositivos emulan la entrada de un teclado físico, en Microsoft Access se programan macros o procedimientos VBA en los eventos “Al cambiar” (OnChange) o “Después de actualizar” (AfterUpdate) del control de texto. Al accionarse, el sistema ejecuta de manera automática una consulta interna en las tablas para verificar la existencia del libro o activar el módulo de préstamos. Esta automatización reduce significativamente la intervención manual y disminuye los errores de captura por transcripción, garantizando la consistencia del repositorio de datos (*Elmasri & Navathe, 2016*).

## SALIDAS

Las salidas de información constituyen un componente fundamental en los sistemas de gestión de bases de datos, ya que permiten transformar los datos relacionales almacenados en conocimiento útil, significativo y estructurado. A través de estas interfaces de salida, los usuarios pueden interpretar la información de manera clara, facilitando la toma de decisiones, el análisis de tendencias y la generación de reportes estratégicos dentro de la institución (*Silberschatz et al., 2019*).

En el entorno de Microsoft Access, estas salidas se implementan principalmente mediante tres mecanismos específicos que cumplen funciones diferenciadas en el proceso de presentación de los datos:

- Consultas de Selección y Agregación: Permiten filtrar, ordenar, calcular y procesar grandes volúmenes de datos mediante criterios definidos por el usuario (operaciones en lenguaje SQL subyacente). Esto posibilita extraer de las tablas únicamente los registros precisos y relevantes en función de las necesidades operativas de la biblioteca.

- Informes (Reports): Ofrecen una estructura formal, estática y organizada de los datos, ideal para su presentación institucional, impresión o exportación a formatos como PDF. Los informes en Access aprovechan las propiedades de ordenación y agrupamiento, incorporando secciones estructuradas como encabezados de página, pies de informe, totales calculados y formatos condicionales personalizados (*Microsoft, 2023*).
- Elementos Visuales en Formularios: Los formularios no solo operan como interfaces de entrada, sino que funcionan como tableros de control (dashboards) que incluyen elementos visuales como subformularios dinámicos, gráficos de rendimiento e indicadores de estado. Estos componentes representan los datos de manera intuitiva, favoreciendo la interpretación inmediata de la información en tiempo real.

*De acuerdo con Sommerville (2016)*, un aspecto crítico en el diseño de las salidas de información es su capacidad de adaptación a los distintos niveles de usuario dentro de la organización. Mientras que los actores operativos (como el bibliotecario en ventanilla) requieren listados detallados, específicos y minuciosos de los movimientos diarios, los niveles directivos o de alta administración precisan de resúmenes ejecutivos, reportes consolidados y visualizaciones sintéticas que reúnan grandes cantidades de registros en indicadores clave de desempeño.

En conjunto, la correcta arquitectura de las salidas de información optimiza la comunicación interna y asegura la entrega de datos confiables y estructurados entre las áreas. De este modo, las salidas no representan simplemente el resultado final del procesamiento de datos, sino que se consolidan como una herramienta estratégica indispensable para convertir la información en un activo de valor para el control del inventario y la toma de decisiones basada en evidencia (*Pressman & Maxim, 2020*).

## Informes Visuales

Los informes (Reports) constituyen una de las herramientas más potentes dentro de Microsoft Access para la presentación formal, el filtrado avanzado y el análisis estructurado de datos. Su función principal es organizar el registro de transacciones en una estructura jerárquica y comprensible, permitiendo al usuario visualizar grandes volúmenes de datos de forma clara, ordenada y significativa. A través de la capa de diseño de los informes, es posible integrar agrupaciones lógicas, totales y subtotales calculados que enriquecen la interpretación de la información almacenada en el repositorio relacional (*Silberschatz et al., 2019*).

Una de las ventajas metodológicas de los informes es su capacidad para parametrizar y personalizar la presentación de los datos según los requerimientos del actor que los solicita. En el diseño de la interfaz del sistema de la biblioteca, esto se traduce en capacidades operativas específicas:

- Estructuración del Acervo: Permite generar un catálogo de libros agrupado dinámicamente por autor, editorial o género. En la sección Detalle del informe, se integran controles de imagen vinculados a las rutas de almacenamiento de las portadas para mejorar la presentación visual.
- Seguimiento de Flujos: Facilita el diseño de reportes específicos orientados a la auditoría del servicio, tales como listas de libros con devoluciones pendientes, registros cronológicos de préstamos o listados consolidados de usuarios sancionados (*Sommerville, 2016*).
- Módulos Estadísticos: Permite calcular de manera automatizada indicadores clave, como la frecuencia de uso de los materiales, la disponibilidad física de los ejemplares en las estanterías y las tendencias en los hábitos de lectura de la comunidad del CECyT 5.

Adicionalmente, el motor de informes de Microsoft Access permite aplicar reglas de formato condicional (por ejemplo, resaltar en color rojo los registros de usuarios con retrasos críticos), estructurar secciones fijas mediante encabezados y pies de página o de grupo, e integrar expresiones en lenguaje Access SQL o funciones integradas (como Suma o Cuenta) para el análisis cuantitativo. Esto los convierte en una herramienta clave para la interpretación de la información, ayudando a detectar patrones, variaciones institucionales y posibles anomalías en el inventario (*Microsoft, 2023*).

Desde una perspectiva de interoperabilidad, los informes de Access pueden ser exportados de forma nativa a diferentes formatos externos como PDF o archivos de hojas de cálculo de Excel. Esta flexibilidad facilita su distribución electrónica y su uso en distintos contextos, ya sea para auditorías administrativas, planeación académica o juntas de gestión. De esta manera, el objeto informe trasciende su función interna y se convierte en un medio de comunicación efectiva de la información procesada (*Pressman & Maxim, 2020*).

*De acuerdo con Tufte (2001)*, una visualización efectiva de datos debe priorizar de manera absoluta la claridad, evitar la incorporación de elementos innecesarios (chartjunk) y facilitar la interpretación directa de las métricas. En este sentido, la arquitectura visual de los informes en este proyecto se centra en la simplicidad, la coherencia en la paleta de colores y la organización lógica de los datos, evitando la sobrecarga de decoraciones gráficas que puedan dificultar la legibilidad de los metadatos bibliográficos por parte del administrador.

En conjunto, los informes representan una herramienta esencial dentro del ciclo de desarrollo de software para transformar los datos puros en información útil y estructurada, permitiendo no

solo su presentación estética, sino también su análisis analítico y su aprovechamiento estratégico en la toma de decisiones basada en evidencia.

### Consultas de Selección y Parámetros (Outputs en Pantalla)

Las consultas constituyen una de las herramientas de procesamiento y manipulación más potentes dentro de Microsoft Access, ya que permiten interactuar, filtrar y proyectar la información de forma dinámica y eficiente. A través de ellas, los usuarios y los módulos de la aplicación pueden explotar los datos almacenados sin necesidad de alterar directamente las tablas base o las relaciones subyacentes, lo que facilita el análisis, el aislamiento de registros y la exploración del acervo en tiempo real (*Silberschatz et al., 2019*).

En el contexto de las salidas (outputs) directas en pantalla, las consultas desempeñan un papel clave al proporcionar conjuntos de resultados inmediatos (recordsets) que se despliegan de forma interactiva en formularios, subformularios o vistas de hoja de datos. En el diseño de la interfaz de la biblioteca, esta funcionalidad se implementa mediante soluciones técnicas específicas:

- **Buscador Dinámico en Pantalla:** Consiste en un control de búsqueda rápida insertado en el formulario principal, el cual filtra los registros del catálogo bibliográfico conforme el usuario introduce texto en tiempo real. Este mecanismo reduce drásticamente el tiempo de localización de ejemplares y optimiza la experiencia del usuario al manipular grandes volúmenes de datos.
- **Automatización mediante Eventos:** Esta búsqueda dinámica se desarrolla mediante programación en VBA (Visual Basic for Applications) o a través de macros nativas de Access. Al vincular los procedimientos al evento “Al cambiar” (OnChange) del cuadro de texto, el sistema intercepta cada pulsación de tecla para actualizar automáticamente los criterios de una consulta SQL condicional (utilizando el operador LIKE), generando una interfaz fluida e intuitiva.
- **Consultas Parametrizadas:** Permiten que el sistema solicite al usuario una variable específica (como el número de cuenta de un alumno o el ISBN de un libro) justo antes de ejecutar la consulta, refinando los resultados en pantalla de manera exacta (Microsoft, 2023).

Además de su uso en la indexación de búsquedas, las consultas en pantalla permiten generar listas personalizadas, ordenar datos bajo esquemas multifactoriales y ejecutar operaciones aritméticas que derivan en indicadores útiles para el análisis administrativo de la biblioteca. Es importante destacar que mientras las consultas de selección y de parámetros recuperan y proyectan

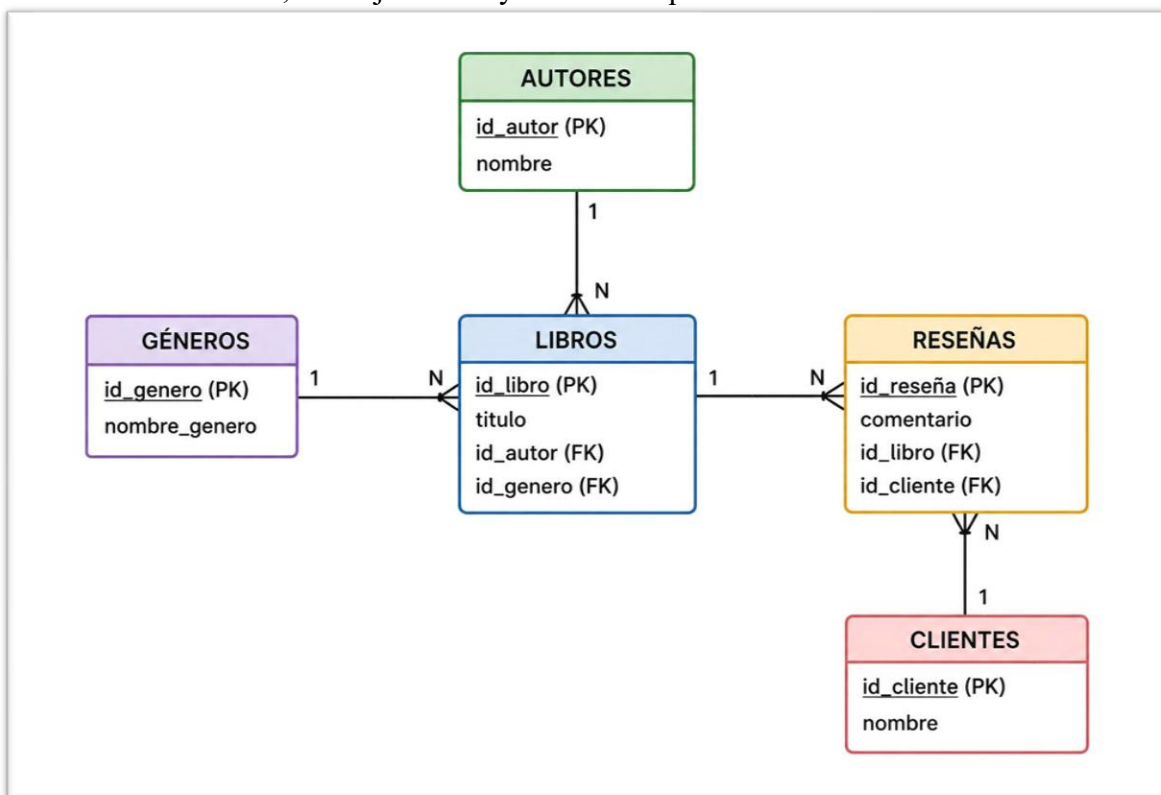
datos específicos en pantalla como salidas de información, el sistema también aprovecha las consultas de acción, las cuales se encargan de automatizar modificaciones masivas en el repositorio a través de transacciones controladas de actualización (UPDATE), eliminación (DELETE) o inserción (INSERT) de registros (Coronel & Morris, 2019).

Asimismo, estas consultas se integran de forma nativa como el origen de registro (RecordSource) para los formularios operativos y los informes institucionales, sirviendo como la base lógica para la generación de salidas de información más complejas y estructuradas. De acuerdo con Pressman y Maxim (2020), esta cohesión arquitectónica entre la lógica de las consultas y las interfaces de presentación contribuye a mantener la consistencia lógica del sistema y a optimizar el rendimiento de los flujos de trabajo de software.

En conjunto, las consultas en pantalla no solo simplifican el acceso inmediato a la información, sino que potencian las capacidades analíticas de la plataforma, consolidándose como un componente de software esencial para elevar la eficiencia operativa de la biblioteca y la toma de decisiones basada en datos confiables.

## ESTRUCTURA DE LA BASE DE DATOS

El proceso inicia con el acceso del usuario al sistema, donde posteriormente puede seleccionar distintas acciones, como registrar un nuevo libro o consultar información previamente almacenada. En caso de registrar un libro, el usuario introduce los datos correspondientes, los cuales son almacenados en la base de datos. De igual manera, el sistema permite agregar reseñas relacionadas con cada libro, facilitando la organización y consulta de opiniones personales. Finalmente, el flujo concluye una vez que el usuario termina la actividad deseada dentro del



# IMPLEMENTACIÓN

## INTRODUCCIÓN A LA IMPLEMENTACIÓN

En el desarrollo actual del proyecto de la base de datos para la biblioteca, la fase de implementación ya se encuentra en ejecución y se está abordando de manera estructurada, considerando cada una de sus etapas como componentes fundamentales para lograr un sistema funcional, eficiente y adoptado por los usuarios. La implementación no es una actividad única, sino un proceso integral compuesto por varias fases interrelacionadas que permiten materializar todo lo definido en el análisis y diseño del sistema (*Ramsdata, s.f.*).

A continuación, se describen de manera amplia las principales partes de la implementación, aplicadas al contexto del proyecto:

### • **Codificación (Desarrollo del sistema)**

La codificación constituye el núcleo técnico de la implementación, ya que en esta etapa se traducen las especificaciones del diseño en código funcional. En el proyecto de la biblioteca, esto implica la creación de la base de datos, el desarrollo de las tablas, relaciones, consultas y posibles interfaces que permitirán gestionar la información bibliográfica, los usuarios y los préstamos.

Este proceso no se limita a “programar”, sino que requiere aplicar buenas prácticas para garantizar la calidad del sistema. El código debe ser claro, legible, modular y correctamente documentado, lo que facilita su mantenimiento y futuras mejoras (*Robertosi3, 2014*). Además, se están utilizando estructuras de datos y algoritmos adecuados para asegurar un rendimiento eficiente, especialmente en consultas frecuentes como la búsqueda de libros o el historial de préstamos.

Otro aspecto importante es que la codificación se está desarrollando de forma paralela con las pruebas, lo que permite detectar errores desde etapas tempranas y evitar fallas mayores en fases posteriores.

### • **Pruebas (Validación del sistema)**

La fase de pruebas es esencial para garantizar que el sistema funcione correctamente antes de su uso definitivo. En el caso de la base de datos de la biblioteca, se están realizando pruebas sobre cada uno de los módulos, verificando que las operaciones como el registro de libros, préstamos, devoluciones y consultas funcionen sin errores.

Estas pruebas no solo se enfocan en detectar fallos técnicos, sino también en validar que el sistema cumpla con los requerimientos definidos durante el análisis. Es decir, se busca asegurar que la solución realmente resuelva las necesidades de la biblioteca.

Además, la ejecución de pruebas continuas permite mejorar la confiabilidad del sistema y garantizar la integridad de los datos, lo cual es especialmente importante en un entorno donde la información debe ser precisa y consistente.

#### • **Instalación (Implementación técnica en el entorno real)**

Una vez que el sistema ha sido desarrollado y probado, se procede a su instalación en el entorno donde será utilizado. En este proyecto, esto implica configurar el sistema gestor de bases de datos, implementar la base de datos en los equipos o servidores correspondientes y asegurar que todos los componentes tecnológicos funcionen correctamente.

La instalación también contempla la integración con otros recursos tecnológicos disponibles en la biblioteca, asegurando que el sistema pueda operar sin conflictos. Esta etapa es clave, ya que cualquier error en la configuración puede afectar el funcionamiento general del sistema.

#### • **Documentación (Soporte técnico y operativo)**

La documentación es una parte fundamental de la implementación, aunque muchas veces es subestimada. En este proyecto, se está generando documentación tanto técnica como de usuario.

Por un lado, la documentación técnica describe la estructura de la base de datos, el funcionamiento del sistema y los aspectos relevantes para su mantenimiento. Por otro lado, la documentación de usuario incluye manuales e instrucciones que facilitan el uso del sistema por parte del personal de la biblioteca.

Una buena documentación permite reducir la dependencia de los desarrolladores y facilita la resolución de problemas, además de servir como base para futuras actualizaciones o mejoras.

#### • **Capacitación (Preparación de los usuarios)**

Aunque se desarrolla como un punto independiente dentro del trabajo, la capacitación forma parte directa de la implementación. En esta etapa, los usuarios de la biblioteca están siendo entrenados para utilizar el sistema de manera adecuada.

Esto incluye aprender a registrar información, realizar consultas, gestionar préstamos y utilizar correctamente las funcionalidades del sistema. La capacitación es clave para garantizar la adopción del sistema, ya que un sistema técnicamente correcto puede fracasar si los usuarios no saben cómo utilizarlo (*Evaluando Software, s.f.*).

- **Puesta en marcha o conversión (Inicio de operaciones)**

La puesta en marcha representa el momento en que el sistema comienza a utilizarse en condiciones reales. En el caso del proyecto, esto implica que la base de datos de la biblioteca empieza a operar como el sistema principal para la gestión de información.

Este proceso puede realizarse de manera gradual (por ejemplo, mediante un piloto con un grupo de usuarios) o de forma directa, dependiendo de la estrategia definida. Durante esta etapa, es común que surjan ajustes menores, los cuales se corrigen conforme el sistema entra en operación.

- **Soporte y evaluación (Seguimiento del sistema)**

Una vez que el sistema está en funcionamiento, se inicia la fase de soporte y evaluación. En el proyecto de la biblioteca, esto implica brindar asistencia a los usuarios, resolver dudas, corregir errores y realizar mejoras continuas.

Además, se evalúa el desempeño del sistema en función de indicadores como la eficiencia en los procesos, la satisfacción de los usuarios y la calidad de la información generada. Esta evaluación permite determinar si el sistema cumple con los objetivos planteados y detectar oportunidades de mejora.

- **Integración de las etapas en el proyecto**

Todas estas fases no se desarrollan de manera aislada, sino que están interrelacionadas y se ejecutan de forma progresiva dentro del proyecto. La implementación del sistema de información de la biblioteca se está llevando a cabo como un proceso integral que combina aspectos técnicos, organizacionales y humanos.

Asimismo, es importante considerar que esta fase implica un cambio organizacional significativo, ya que modifica la forma en que los usuarios gestionan la información. Por ello, el éxito del proyecto no depende únicamente de la tecnología, sino también del nivel de adopción por parte de los usuarios, el apoyo de la dirección y la correcta gestión del cambio (Lucas, 1993)

## INSTALACIÓN DEL SISTEMA

La instalación del sistema, dentro de la etapa de implementación de un sistema de información, constituye un proceso fundamental que marca la transición entre el desarrollo teórico del sistema y su aplicación práctica en un entorno real. En el contexto de la gestión de una biblioteca, esta fase adquiere una relevancia particular, ya que implica garantizar que todos los componentes tecnológicos, humanos y organizativos funcionen de manera coordinada para lograr una administración eficiente de los recursos bibliográficos y de los servicios ofrecidos a los usuarios (*Laudon & Laudon, 2016*).

En términos generales, la instalación del sistema no debe entenderse como una simple acción técnica de copiar archivos o ejecutar programas, sino como un conjunto de actividades planificadas que aseguran que el sistema funcione correctamente en el entorno donde será utilizado. Esto implica considerar aspectos como la infraestructura tecnológica disponible, las características del personal que operará el sistema, la estructura organizativa de la biblioteca y los procesos internos que serán automatizados (*Kendall & Kendall, 2014*).

Uno de los primeros pasos en la instalación del sistema consiste en la evaluación del entorno físico donde se implementará. En una biblioteca, este entorno puede variar considerablemente dependiendo del tamaño y tipo de institución. Por ejemplo, una biblioteca escolar pequeña puede requerir un solo equipo de cómputo, mientras que una biblioteca universitaria puede necesitar múltiples estaciones de trabajo interconectadas mediante una red local. En ambos casos, es indispensable verificar que los equipos disponibles cumplen con los requisitos mínimos necesarios para ejecutar el sistema sin inconvenientes.

La evaluación del hardware incluye aspectos como la capacidad del procesador, la memoria RAM, el espacio de almacenamiento y la estabilidad del equipo. Un sistema de gestión bibliotecaria, aunque no suele ser excesivamente demandante en términos de recursos, sí requiere cierta estabilidad para evitar fallos durante operaciones críticas como el registro de préstamos o la actualización de datos. Asimismo, es importante considerar la disponibilidad de dispositivos periféricos, como impresoras para la emisión de reportes o lectores de códigos de barras en caso de que la biblioteca utilice este tipo de tecnología para agilizar los procesos de préstamo y devolución.

Una vez evaluado el entorno físico, se procede a la instalación del software base necesario para el funcionamiento del sistema. En este caso, al tratarse de un sistema desarrollado en Microsoft Access, es imprescindible contar con una versión compatible de este programa instalada en los equipos. Dependiendo de la versión del sistema, puede ser necesario instalar también componentes adicionales, como bibliotecas de ejecución o complementos específicos que permitan el correcto funcionamiento de formularios y consultas.

La instalación del sistema también implica la organización de los archivos que componen la aplicación. En muchos casos, especialmente cuando se trabaja con Access, se recomienda utilizar una arquitectura dividida en dos partes: el front-end y el back-end. El front-end contiene la interfaz de usuario, es decir, los formularios, menús y reportes, mientras que el back-end almacena los datos en forma de tablas. Esta separación permite mejorar el rendimiento del sistema y facilita su mantenimiento, ya que los cambios en la interfaz no afectan directamente a los datos (*Coronel & Morris, 2019*).

En una biblioteca donde múltiples usuarios acceden al sistema, el back-end suele ubicarse en un servidor o en una carpeta compartida dentro de una red local. Esto permite que todos los usuarios trabajen con la misma base de datos, evitando duplicidades y asegurando la coherencia de la información (*Date, 2004*). La instalación de este componente debe realizarse con especial cuidado, garantizando que la ubicación elegida sea segura, accesible y cuente con mecanismos de respaldo.

Otro aspecto crucial durante la instalación es la configuración de la red, en caso de que el sistema sea multiusuario. La red debe ser estable y tener suficiente capacidad para soportar el tráfico generado por las consultas y actualizaciones de la base de datos. Además, es importante configurar adecuadamente los permisos de acceso a la carpeta donde se encuentra el back-end, de manera que solo los usuarios autorizados puedan modificar la información.

La instalación también incluye la definición de perfiles de usuario y niveles de acceso. En una biblioteca, no todos los empleados tienen las mismas responsabilidades, por lo que el sistema debe reflejar esta realidad. Por ejemplo, el administrador del sistema puede tener acceso completo para modificar estructuras y configuraciones, mientras que el personal de atención al público puede limitarse a registrar préstamos y devoluciones. Esta diferenciación contribuye a la seguridad del sistema y reduce el riesgo de errores o manipulaciones indebidas (*Silberschatz et al., 2006*).

Durante esta fase, también es fundamental realizar pruebas de funcionamiento en el entorno real. Estas pruebas permiten verificar que el sistema responde adecuadamente a las operaciones cotidianas de la biblioteca. Se pueden simular escenarios como el registro de un nuevo usuario, el préstamo de un libro, la devolución de materiales y la generación de reportes. Estas pruebas ayudan a identificar posibles fallos que no fueron detectados durante el desarrollo y permiten realizar ajustes antes de que el sistema entre en operación definitiva (*Pressman, 2010*).

Otro elemento importante en la instalación del sistema es la migración de datos. Si la biblioteca ya contaba con registros previos, ya sea en formato digital o en papel, es necesario transferir esta información al nuevo sistema. Este proceso puede ser complejo, ya que implica la revisión, limpieza y organización de los datos para garantizar su calidad. Una migración mal realizada puede generar inconsistencias que afecten el funcionamiento del sistema.

Además, la instalación debe contemplar la creación de copias de seguridad desde el inicio. Esto implica definir una política de respaldos que asegure la recuperación de la información en

caso de fallos técnicos, errores humanos o eventos imprevistos. En una biblioteca, la pérdida de datos puede tener consecuencias graves, por lo que este aspecto no debe subestimarse (*Date, 2004*).

Finalmente, la instalación del sistema culmina con su puesta en marcha y la capacitación del personal. Este proceso implica enseñar a los usuarios cómo utilizar el sistema de manera correcta, familiarizarlos con la interfaz y explicarles los procedimientos básicos. Una buena capacitación facilita la adopción del sistema y reduce la resistencia al cambio, que es común cuando se introducen nuevas tecnologías en una organización (*Kendall & Kendall, 2014*).

## Configuración y entorno (uso de Access y requisitos)

La configuración del entorno y el uso de la herramienta tecnológica representan una fase esencial dentro de la implementación de un sistema de información. En el caso de una biblioteca que utiliza Microsoft Access como plataforma principal, este proceso implica adaptar el sistema a las condiciones específicas de operación, optimizar su rendimiento y asegurar que los usuarios puedan interactuar con él de manera eficiente, segura y coherente con los objetivos de la institución (*Laudon & Laudon, 2016*).

Microsoft Access es un sistema de gestión de bases de datos relacional (RDBMS) que permite almacenar, organizar y manipular información de manera estructurada. Su principal ventaja radica en su accesibilidad, ya que no requiere conocimientos avanzados de programación para desarrollar aplicaciones funcionales. Esto lo convierte en una herramienta ideal para entornos como bibliotecas, donde se necesita una solución práctica y adaptable para gestionar grandes volúmenes de información (*Coronel & Morris, 2019*).

El primer aspecto a considerar en la configuración del entorno es la definición de los requisitos técnicos. En términos de hardware, es necesario contar con equipos que tengan suficiente capacidad para ejecutar el sistema sin dificultades. Esto incluye procesadores modernos, una cantidad adecuada de memoria RAM y espacio de almacenamiento suficiente para albergar la base de datos. Aunque Access no es particularmente exigente, el rendimiento puede verse afectado si los recursos son limitados, especialmente en entornos multiusuario (*Kendall & Kendall, 2014*).

En cuanto al software, se requiere un sistema operativo compatible, generalmente Windows, y una versión de Microsoft Access que soporte las características del sistema desarrollado. También es recomendable mantener el software actualizado para evitar problemas de compatibilidad y aprovechar mejoras en seguridad y rendimiento.

La configuración del entorno de trabajo implica organizar adecuadamente los archivos y definir la estructura del sistema. Como se mencionó anteriormente, es recomendable dividir la base

de datos en front-end y back-end. Esta separación permite que los usuarios trabajen con interfaces locales mientras acceden a una base de datos centralizada, lo que mejora la eficiencia y reduce el riesgo de conflictos de red y corrupción de archivos (*Date, 2004*).

El uso de Access en la biblioteca se centra en la interacción a través de formularios. Estos formularios actúan como interfaces que facilitan la entrada y consulta de datos. Por ejemplo, un formulario de registro de libros puede incluir campos para el título, autor, editorial y año de publicación. Asimismo, un formulario de préstamos permite asociar un usuario con un libro y registrar la fecha de devolución.

La configuración de estos formularios es clave para la usabilidad del sistema. Deben ser intuitivos, claros y diseñados de manera que minimicen errores. Esto puede lograrse mediante validaciones de datos, listas desplegadas y mensajes informativos, aplicando principios fundamentales de la interacción humano-computadora (*Shneiderman et al., 2016*). Un buen diseño de formularios mejora la eficiencia del trabajo y reduce la necesidad de capacitación intensiva.

Otro componente importante son las consultas, que permiten recuperar información específica de la base de datos mediante el álgebra relacional subyacente. En una biblioteca, las consultas pueden utilizarse para identificar libros disponibles, préstamos atrasados o usuarios frecuentes. Estas consultas pueden ser simples o complejas, dependiendo de las necesidades de información (*Silberschatz et al., 2006*).

Los reportes, por su parte, permiten presentar la información de manera organizada y comprensible. Son especialmente útiles para la generación de informes administrativos, estadísticas y seguimiento de actividades. La configuración de reportes debe considerar aspectos como el formato, la claridad y la relevancia de la información presentada para soportar la toma de decisiones (*Kendall & Kendall, 2014*).

En un entorno multiusuario, la configuración adquiere mayor complejidad. Es necesario asegurar que múltiples usuarios puedan acceder al sistema simultáneamente sin generar conflictos de concurrencia. Esto implica configurar adecuadamente el bloqueo de registros, optimizar las consultas y garantizar la estabilidad de la red (*Date, 2004*). También es importante establecer normas de uso para evitar prácticas que puedan comprometer el sistema.

La seguridad es otro aspecto fundamental en la configuración del entorno. Esto incluye la protección de la base de datos mediante contraseñas, la definición de permisos de acceso y la implementación de controles internos. En una biblioteca, es importante proteger tanto la privacidad y la información confidencial de los usuarios como la integridad de los registros de los libros (*Silberschatz et al., 2006*).

La capacitación del personal forma parte integral de la configuración del entorno. Un sistema bien diseñado pierde su efectividad si los usuarios no saben cómo utilizarlo correctamente. Por ello, es necesario proporcionar formación adecuada que cubra desde el uso básico hasta la resolución de problemas comunes, disminuyendo la resistencia al cambio tecnológico (*Laudon & Laudon, 2016*).

El mantenimiento del sistema también debe considerarse dentro de la configuración. Esto incluye tareas críticas como la compactación y reparación periódica de la base de datos para recuperar espacio en disco, la actualización del software y la revisión del rendimiento. Estas acciones ayudan a prevenir la degradación de los archivos y prolongan la vida útil del sistema (*Coronel & Morris, 2019*).

Finalmente, la configuración del entorno debe ser flexible para adaptarse a cambios. Las bibliotecas evolucionan constantemente, por lo que el diseño conceptual del sistema debe poseer la escalabilidad necesaria para ajustarse a nuevas necesidades sin requerir una reconstrucción completa (*Pressman, 2010*).

## Capacitación del Usuario

En el contexto actual del proyecto de implementación de la base de datos de la biblioteca, la capacitación de usuarios se está desarrollando como una fase estratégica y continua, orientada a garantizar que el sistema no solo funcione correctamente desde el punto de vista técnico, sino que también sea utilizado de manera eficiente por todos los actores involucrados. Esta etapa resulta fundamental, ya que el éxito de un sistema de información depende en gran medida del nivel de conocimiento y adaptación por parte de los usuarios finales, un fenómeno ampliamente estudiado en los modelos de aceptación tecnológica (*Davis, 1989*).

Al igual que la implementación técnica, la capacitación no es una actividad aislada, sino un proceso estructurado que se divide en distintas etapas, las cuales se están ejecutando de manera progresiva dentro del proyecto de la biblioteca (*Kendall & Kendall, 2014*).

- Identificación de usuarios y necesidades de capacitación

En esta primera etapa, se ha realizado la identificación de los diferentes tipos de usuarios que interactuarán con el sistema. En el caso de la biblioteca, esto incluye personal administrativo, encargados de préstamos, catalogadores y posiblemente usuarios con funciones de consulta.

Cada grupo posee necesidades, responsabilidades y niveles de conocimiento distintos, por lo que la capacitación se está diseñando de forma diferenciada. Este enfoque permite que cada usuario reciba únicamente la información relevante para sus funciones, evitando la sobrecarga cognitiva y facilitando el aprendizaje constructivo (*Sweller, 1988*).

Asimismo, se están considerando los conocimientos previos del personal, ya que no todos poseen el mismo nivel de experiencia en el uso de sistemas informáticos. Esto influye directamente en el tiempo y la profundidad de la capacitación requerida.

- Planificación de la capacitación

Una vez identificados los usuarios, se ha procedido a estructurar un plan de capacitación alineado con el avance del proyecto. Este plan contempla objetivos claros, contenidos específicos, cronogramas y recursos necesarios para su ejecución (*Pressman, 2010*).

En el proyecto actual, la capacitación se está organizando de manera paralela a otras fases de la implementación, lo cual permite que los usuarios vayan aprendiendo conforme el sistema se desarrolla. Esta estrategia favorece una mejor comprensión, ya que los conocimientos adquiridos pueden aplicarse inmediatamente en un entorno práctico.

Además, se ha considerado el apoyo de la dirección como un elemento clave, ya que su involucramiento y liderazgo facilitan la adopción del sistema, validan los recursos asignados y refuerzan la importancia del proceso formativo dentro de la cultura de la organización (*Laudon & Laudon, 2016*).

- Desarrollo de contenidos y materiales

En esta etapa se están elaborando los materiales de apoyo que servirán como base para la capacitación. Estos incluyen manuales de usuario, guías prácticas, documentación técnica y recursos digitales.

Dado que en muchos casos la documentación proporcionada por los desarrolladores suele ser limitada o poco accesible, se está poniendo especial énfasis en generar materiales claros, estructurados y adaptados al contexto de la biblioteca. Esto permite que los usuarios puedan consultar información de manera autónoma cuando lo necesiten, promoviendo el autoaprendizaje (*Sommerville, 2011*).

Asimismo, se están incorporando recursos complementarios como tutoriales y ejemplos prácticos, los cuales facilitan la comprensión del sistema y refuerzan el aprendizaje.

- Ejecución de la capacitación (proceso formativo)

La capacitación se está llevando a cabo mediante diferentes metodologías, combinando sesiones presenciales con el uso de herramientas digitales. Este enfoque mixto (blended learning) permite adaptarse a las necesidades de los usuarios y optimizar el proceso de aprendizaje (*Graham, 2006*).

Durante esta fase, los usuarios están aprendiendo a utilizar el sistema en función de sus tareas específicas, tales como registrar libros, gestionar préstamos, realizar consultas y generar reportes. La capacitación tiene un enfoque práctico, lo que permite a los participantes interactuar directamente con el sistema y resolver dudas en tiempo real.

Además, el número de sesiones, su duración y los recursos utilizados se han definido en función del nivel de complejidad del sistema y de la disponibilidad del personal, garantizando así un proceso formativo adecuado e integrado en la jornada laboral.

- Transferencia de conocimiento dentro del equipo de implementación

La capacitación no solo se dirige a los usuarios finales, sino también al equipo de implementación en su conjunto. Esto incluye tanto a los usuarios como a los consultores o responsables técnicos.

En este sentido, los usuarios adquieren conocimientos sobre el funcionamiento del sistema aplicado a sus tareas, mientras que los consultores profundizan en los procesos del negocio y en las necesidades específicas de la biblioteca. Esta doble transferencia de conocimiento permite una mejor alineación entre la tecnología y los procesos organizacionales (*Henderson & Venkatraman, 1993*).

Sin embargo, durante este proceso pueden surgir deficiencias en la transferencia de conocimientos, las cuales pueden afectar directamente los tiempos de implementación y generar costos adicionales. Por ello, se está prestando especial atención a la calidad del proceso formativo.

- Evaluación del aprendizaje

Una vez ejecutadas las actividades de capacitación, se está llevando a cabo una evaluación del aprendizaje de los usuarios. Esta evaluación permite medir el nivel de comprensión del sistema, identificar posibles dificultades y determinar si los usuarios están preparados para utilizar el sistema de manera autónoma (*Kirkpatrick & Kirkpatrick, 2006*).

En el proyecto de la biblioteca, esta evaluación se realiza mediante ejercicios prácticos, observación del desempeño y retroalimentación directa de los participantes. Esto permite ajustar el

proceso de capacitación en caso de ser necesario y reforzar aquellos aspectos que presenten mayores dificultades.

- Factores que afectan la capacitación

Durante la ejecución del proyecto, se han identificado diversos factores que pueden influir en la efectividad de la capacitación. Entre los más relevantes se encuentran:

- Limitaciones en los tiempos de implementación, que pueden reducir el tiempo disponible para la formación.
- Diferencias en el nivel de conocimientos del personal.
- Falta de definición clara de políticas y procedimientos dentro de la organización.
- Escaso involucramiento de la dirección en el proceso.

Estos factores de riesgo sociotécnicos son críticos, ya que pueden generar retrasos, errores en el uso del sistema y una marcada resistencia al cambio estructural (Cobarsí-Morales, 2011). Por ello, se están gestionando de manera activa para minimizar su impacto y asegurar el éxito del proyecto.

- Soporte continuo y refuerzo de la capacitación

La capacitación no finaliza con las sesiones iniciales, sino que continúa durante la fase de operación del sistema. En este proyecto, se está proporcionando soporte continuo a los usuarios, permitiendo resolver dudas, corregir errores y reforzar conocimientos mediante la asistencia técnica posterior a la puesta en marcha (*Pressman, 2010*)

Además, se están utilizando materiales de apoyo como manuales y recursos digitales que permiten a los usuarios consultar información en cualquier momento. Este enfoque contribuye a consolidar el aprendizaje y a mejorar el uso del sistema a largo plazo.

- Importancia de la capacitación en el éxito del sistema

Finalmente, es importante destacar que la capacitación es un elemento determinante para el éxito de la implementación. Un sistema puede estar correctamente desarrollado desde el punto de vista técnico, pero si los usuarios no perciben su utilidad o les resulta difícil de operar, el índice de adopción real colapsará (Davis, 1989).

En el caso del proyecto de la base de datos de la biblioteca, la capacitación está permitiendo que los usuarios se apropien del sistema, integrándolo en sus actividades diarias y aprovechando al

máximo sus funcionalidades. Esto se traduce en una mejora en la gestión de la información, mayor eficiencia en los procesos y una mejor toma de decisiones dentro de la organización.

### Conclusiones de la Implementación

El llevar a cabo la implementación del sistema de información para la base de datos de la biblioteca permite evidenciar que tanto la implementación técnica como la capacitación de usuarios no son procesos aislados, sino componentes profundamente interdependientes que determinan el éxito o fracaso del sistema en su conjunto. A lo largo de la ejecución del proyecto, se ha podido constatar que la implementación no se limita únicamente a la construcción del sistema, sino que implica una integración completa de elementos tecnológicos, organizacionales y humanos que deben funcionar de manera coordinada, respondiendo a un enfoque sistémico (Laudon & Laudon, 2016).

Por un lado, la implementación se ha consolidado como una fase compleja y estructurada que abarca desde la codificación hasta el soporte posterior a la puesta en marcha. Cada una de sus etapas —codificación, pruebas, instalación, documentación, puesta en marcha y soporte— ha sido desarrollada de manera progresiva, permitiendo materializar los requerimientos definidos en las fases previas (Pressman, 2010). Este proceso ha requerido no solo conocimientos técnicos, sino también una adecuada planificación, selección de herramientas y control de calidad constante, asegurando que el sistema cumpla con los estándares necesarios para operar de forma confiable.

Asimismo, se ha evidenciado que la implementación representa un cambio organizacional significativo. La introducción de la base de datos en la biblioteca no solo transforma la forma en que se gestiona la información, sino también la manera en que los usuarios interactúan con ella. En este sentido, factores como el apoyo de la dirección, la comunicación efectiva y la gestión del cambio cultural han sido determinantes para facilitar la transición desde los métodos tradicionales hacia el uso del nuevo sistema (Kendall & Kendall, 2014).

Por otro lado, la capacitación de usuarios ha demostrado ser un elemento clave para garantizar la adopción del sistema. A través de un proceso estructurado que incluye la identificación de usuarios, la planificación, el desarrollo de materiales, la ejecución de la formación y la evaluación del aprendizaje, se ha logrado preparar al personal para utilizar el sistema de manera adecuada. Este proceso ha permitido reducir errores, mejorar la eficiencia operativa y fomentar la confianza de los usuarios en la nueva herramienta.

Además, la capacitación ha puesto en evidencia la importancia de considerar factores humanos dentro del proyecto, tales como el nivel de conocimientos previos, la resistencia al cambio

y la necesidad de adaptación a nuevas dinámicas de trabajo. La atención a estos aspectos ha permitido ajustar las estrategias formativas y fortalecer la transferencia de conocimientos, evitando retrasos y costos adicionales (*Davis, 1989*).

Un aspecto relevante que surge de la integración de ambos puntos es que un sistema de información no puede considerarse exitoso únicamente por su correcto funcionamiento técnico. Su verdadero valor radica en el uso efectivo que los usuarios hacen de él. En este sentido, la implementación proporciona la estructura y las herramientas, mientras que la capacitación garantiza que dichas herramientas sean comprendidas y aprovechadas al máximo.

En el caso específico del proyecto de la biblioteca, la combinación de una implementación bien estructurada y un proceso de capacitación adecuado está permitiendo alcanzar una operación estable del sistema, en la cual los usuarios pueden gestionar la información de manera eficiente, acceder a datos confiables y mejorar la calidad del servicio ofrecido. Esto se traduce en beneficios concretos, como una mejor organización del material bibliográfico, mayor control de los préstamos y una toma de decisiones más informada.

## USABILIDAD Y ACCESIBILIDAD

Las Pruebas de Aceptación del Usuario (UAT - User Acceptance Testing) representan la validación final y el criterio de estimación crítico dentro del ciclo de vida del desarrollo de software (SDLC). En el contexto metodológico de un sistema de gestión bibliotecaria implementado sobre la plataforma Microsoft Access, esta fase no constituye un elemento opcional, sino un requerimiento imperativo para el despliegue en producción. Mientras que las pruebas unitarias y estructurales de caja blanca y caja negra se orientan a asegurar la estabilidad del código y la ausencia de excepciones de desbordamiento en las macros de VBA, las pruebas UAT garantizan rigurosamente que la lógica de negocio subyacente —el flujo transaccional de componentes, la trazabilidad de usuarios y el algoritmo de sanciones— sea coherente con la realidad operativa y administrativa de la institución (*Pressman & Maxim, 2020*).

De acuerdo con la ingeniería de requerimientos de *Sommerville (2016)*, la validación se define como el proceso matemático y operativo de asegurar que el sistema de información cumpla plenamente con las expectativas y necesidades del cliente final (respondiendo a la premisa: ¿estamos construyendo el producto correcto?). En el entorno práctico de una biblioteca, esto significa que la arquitectura del software debe poseer la resiliencia necesaria para gestionar excepciones humanas complejas y contingencias del día a día —tales como el extravío de credenciales institucionales, la gestión de prórrogas extraordinarias o el retorno de materiales bibliográficos con daños físicos—, trascendiendo la simple captura estructurada de datos.

Por lo tanto, la fase UAT actúa como el filtro de calidad definitivo donde los futuros operadores (bibliotecarios y alumnos) validan la usabilidad de las interfaces y la precisión de los reportes generados. Este proceso de retroalimentación cíclica no solo mitiga la resistencia al cambio tecnológico dentro de la organización, sino que establece las bases de la mejora continua, asegurando que cualquier optimización posterior de las tablas o los formularios se ejecute bajo estándares de calidad, eficiencia y sostenibilidad técnica a largo plazo (*Elmasri & Navathe, 2016*).

#### Profundización en los protocolos de aceptación

Para ejecutar una auditoría exhaustiva sobre el comportamiento de la plataforma, los protocolos de validación final se desglosan en dimensiones analíticas específicas. Esta categorización permite evaluar tanto el rigor técnico del back-end como la ergonomía de la interfaz gráfica (*Sommerville, 2016*):

#### **A. Dimensión Funcional y Reglas de Negocio**

Esta dimensión dictamina si el motor relacional de Microsoft Access responde con precisión matemática y consistencia lógica a las reglas de negocio del entorno bibliotecario:

- **Gestión Avanzada de Colecciones:** Valida que el esquema de la base de datos admita la clasificación catalográfica normalizada (como el Sistema Dewey o LC) y controle correctamente la cardinalidad de ejemplares complejos. Esto asegura que un único registro maestro de título mapee simétricamente hacia múltiples tomos físicos independientes en la tabla de existencias, manteniendo un inventario real y transparente (*Silberschatz et al., 2019*).

**Seguridad Lógica y Privilegios de Rol:** Verifica la efectividad del Control de Acceso Basado en Roles (RBAC). El protocolo audita que la lógica de los formularios impida físicamente a los usuarios con perfil básico la ejecución de sentencias de borrado o modificaciones no autorizadas en las tablas de historial transaccional, blindando la integridad del sistema.

#### **B. Dimensión de Usabilidad y Experiencia de Usuario (UX)**

Orientada bajo las heurísticas de *diseño de interfaces de Nielsen (1994)*, esta dimensión se enfoca en mitigar la carga cognitiva del operador y asegurar una navegación libre de frustraciones operativas:

- **Mensajería del Sistema y Manejo de Excepciones:** Las rutinas de interrupción en VBA (On Error GoTo) se configuran para interceptar códigos nativos del motor de datos y traducirlos en cuadros de mensaje (MsgBox) semánticamente claros. En lugar de

desplegar una alerta genérica e incomprensible como "Error de tiempo de ejecución 3022", la interfaz notifica de forma asertiva al bibliotecario: "El código de barras o ISBN ingresado ya se encuentra indexado en la base de datos. Verifique el registro."

- **Arquitectura de Navegación Intuitiva:** Se evalúa la implementación de formularios modales dirigidos y menús de cinta (Ribbons) personalizados. Esta disposición visual restringe el acceso a funciones secundarias cuando se ejecuta una tarea crítica, guiando al administrador de forma secuencial paso a paso a través de las operaciones y garantizando un entorno ergonómico, fluido y eficiente (*Microsoft, 2023*).

## AJUSTE Y REFACTORIZACIÓN

Tras la culminación de las Pruebas de Aceptación del Usuario (UAT), el ciclo de vida del software entra en una fase de optimización iterativa. En esta etapa, el desarrollador ejecuta modificaciones arquitectónicas y cosméticas basadas directamente en la bitácora de incidencias y la retroalimentación de los operadores. En el entorno de desarrollo de Microsoft Access, este proceso de refactorización se concentra en tres ejes técnicos fundamentales para elevar el rendimiento y la usabilidad de la plataforma (*Pressman & Maxim, 2020*):

Optimización de Consultas de Acción e Indexación: Cuando los usuarios reportan retardos en la respuesta del sistema al procesar volúmenes densos de datos (como la emisión de reportes ejecutivos de sanciones o el cálculo de morosidades), se ejecuta una refactorización en el back-end. Esto consiste en la indexación selectiva de campos con alta concurrencia en las cláusulas WHERE y JOIN, tales como Fec\_Devolucion e ID\_Usuario. Esta técnica reduce drásticamente el tiempo de ejecución del motor de búsqueda al evitar el escaneo secuencial de las tablas maestras (*Silberschatz et al., 2019*).

Automatización de Tareas Repetitivas y Ergonomía: Para mitigar la carga de trabajo del personal administrativo y acelerar el flujo transaccional en ventanilla, se eliminan los procesos manuales propensos a errores. Un ejemplo de esta optimización es la configuración de propiedades nativas en los controles del formulario, asignando funciones síncronas como Fecha() o Date() en el valor predeterminado de los campos de captura, logrando que el sistema estampe el tiempo real de manera automatizada.

Validación de Datos y Cuadros de Combinación en Cascada: Como estrategia avanzada para la prevención de errores en la captura, se implementa una lógica de filtrado jerárquico en la

interfaz de usuario. Al configurar cuadros de lista o cuadros combinados dependientes en cascada, la selección de una entidad padre (por ejemplo, la categoría Ciencias) altera dinámicamente la propiedad RowSource del control hijo mediante un evento AfterUpdate en VBA, restringiendo las opciones visibles únicamente a subcategorías válidas (tales como Física o Química). Este mecanismo blindo la integridad de los datos clasificados y optimiza la experiencia del operador (*Microsoft, 2023*).

## IMPACTO SOCIO-TÉCNICO

Desde una perspectiva integral, la fase de Pruebas de Aceptación del Usuario (UAT) trasciende la mera verificación algorítmica para consolidarse como un catalizador psicológico fundamental en la gestión del cambio organizacional. El despliegue de una infraestructura digital en una biblioteca que operaba tradicionalmente bajo metodologías manuales o sistemas de registro obsoletos suele inducir fricciones operativas y resistencia pasiva entre los actores involucrados. La mitigación de estos factores de riesgo se logra mediante la inclusión activa del personal en los ciclos iterativos de evaluación del software.

De acuerdo con el Modelo de Aceptación Tecnológica (TAM) *desarrollado por Davis (1989)*, la adopción exitosa de un sistema de información está determinada por la percepción de utilidad y la facilidad de uso experimentadas por el operador final. Al facultar a los bibliotecarios y administradores para participar activamente en la auditoría, diseño y refinamiento de su propio entorno de trabajo digital, se reduce drásticamente la ansiedad tecnológica (tecnofobia) y se acelera la asimilación del software.

Esta estrategia participativa de la ingeniería de usabilidad transforma al usuario de un espectador pasivo a un coautor de la solución informática. Como consecuencia directa, se optimiza la curva de aprendizaje institucional y se incrementa la eficiencia operativa desde la primera jornada de implementación definitiva del sistema en producción, alineando simétricamente el rendimiento técnico del back-end con la estabilidad del clima laboral (*Sommerville, 2016*).

## PLAN DE MITIGACIÓN DE RIESGOS

El Plan de Mitigación de Riesgos Operativos constituye una estrategia metodológica y de control crítico indispensable para garantizar la estabilidad, la alta disponibilidad y la continuidad del servicio del sistema de gestión bibliotecaria durante su fase de despliegue inicial en producción. En arquitecturas desarrolladas sobre el motor de bases de datos de Microsoft Access, la planeación de contingencias adquiere un papel preponderante debido a las limitaciones intrínsecas de la

plataforma frente a variables de concurrencia simultánea, el tamaño del archivo binario y la estabilidad de la infraestructura de red local (LAN) de la institución (*Microsoft, 2023*).

Durante las primeras jornadas de uso intensivo y transaccional por parte de los usuarios, la plataforma se expone a escenarios de estrés operativo y excepciones físicas que, de no ser anticipadas mediante controles de ingeniería de software, podrían derivar en la corrupción de las tablas relacionales, la pérdida de consistencia en la información o la interrupción del servicio administrativo. Por consiguiente, este plan estratégico se orienta a tipificar de manera proactiva las amenazas tecnológicas, delimitar protocolos de contención inmediata y establecer las salvaguardas de seguridad lógica necesarias para proteger el acervo informativo del CECyT 5 y asegurar una transición digital transparente y confiable (*Pressman & Maxim, 2020*).

## CONTROL DE CONCURRENCIA Y BLOQUEO TRANSACCIONAL

Uno de los vectores de riesgo técnico más críticos en el despliegue de sistemas multiusuario sobre entornos de red local es la concurrencia. Este fenómeno ocurre cuando dos o más estaciones de trabajo intentan acceder, modificar o actualizar simultáneamente un mismo registro físico en el back-end (por ejemplo, al procesar de forma paralela la asignación de un préstamo sobre el mismo ejemplar de libro), lo que puede inducir a la colisión de datos, condiciones de carrera o inconsistencias en los balances de inventario (*Silberschatz et al., 2019*).

Para mitigar estas anomalías operacionales y blindar la integridad del repositorio, la lógica del sistema en Microsoft Access implementa un protocolo robusto de control de concurrencia optimista y pesimista a través de las siguientes salvaguardas de ingeniería:

- Bloqueo Homogéneo de Registros (Record Locking): Configuración de la propiedad del motor de datos de Access para aislar de forma dinámica el registro específico (*Form.RecordLocks = 2 / Edited Record*). Esto garantiza que, en el instante en que un bibliotecario inicia la edición de una transacción, el registro queda bloqueado para cualquier otra terminal, impidiendo modificaciones cruzadas hasta que la transacción actual ejecute un comando Commit o guardado definitivo.
- Manejo de Excepciones y Mensajería de Advertencia: Intercepción activa de los códigos de error de concurrencia nativos (como el Error 3197 del motor Jet/ACE). En lugar de interrumpir el flujo del software, el sistema despliega un cuadro de diálogo descriptivo

(MsgBox) que notifica de forma asertiva al operador: "El registro que intenta modificar está siendo actualizado en este momento por otra estación de trabajo. Por favor, reintente la operación."

- Validación de Versión en Guardados Controlados: Programación de rutinas de verificación síncronas antes de la confirmación de cambios. El sistema valida si los valores originales del registro han cambiado en la tabla base durante el tiempo de edición del formulario; si se detecta un cambio perimetral, la interfaz bloquea la sobrescritura automática y solicita la confirmación o cancelación de la operación por parte del usuario (*Coronel & Morris, 2019*).

## PROTOCOLO DE CAIDA DE RED

Dado que la arquitectura física del sistema implementa un modelo distribuido en red local (LAN) —donde el archivo de interfaz de usuario (front-end) se vincula dinámicamente al repositorio central de tablas (back-end)—, la disponibilidad de la plataforma es directamente dependiente de la estabilidad de la conectividad interna. En entornos institucionales del sector público, donde la infraestructura de telecomunicaciones puede presentar limitaciones temporales o saturación de ancho de banda, la pérdida del enlace de datos representa una amenaza crítica de interrupción del servicio (Microsoft, 2023).

Para anular el impacto de estas fallas tecnológicas y garantizar una alta tolerancia a fallos, se estructura un protocolo de continuidad operativa y recuperación ante desastres descompuesto en cuatro fases de ingeniería de procesos:

Modo de Operación Degradada (Registro Manual Síncrono): Al interrumpirse la conexión con las tablas vinculadas (evento interceptado por el front-end a través de excepciones de conectividad ODBC/LAN), el personal de ventanilla suspende la interacción con los formularios y activa de inmediato el uso de bitácoras físicas de contingencia. Estas hojas de control impresas replican los campos obligatorios de la tabla TBL\_Prestamos (Identificador del alumno, código de barras del libro y marca de tiempo), impidiendo la paralización de los servicios de circulación de la biblioteca.

Salvaguarda y Respaldo Local de Datos de Excepción: Procedimiento metodológico que norma el almacenamiento seguro de los formatos físicos generados durante el periodo de desconexión. Estos documentos se resguardan cronológicamente bajo la supervisión del administrador del sistema, fungiendo como el repositorio de auditoría primario hasta el restablecimiento de los servicios de red.

Protocolo de Reintegración y Consolidación Síncrona (Data Reconciliation): Una vez que el canal de comunicación LAN recupera su estabilidad operativa, se ejecuta una fase de captura dirigida en la interfaz de Access. El sistema procesa secuencialmente los registros de la bitácora manual en orden cronológico inverso (FIFO / First In, First Out), forzando al motor de datos a validar nuevamente las restricciones de integridad y las claves foráneas antes de consolidar permanentemente los movimientos en el back-end, erradicando la aparición de inconsistencias (*Silberschatz et al., 2019*).

Plan de Comunicación Interna y Mitigación de Pánico: Protocolo que define los flujos de notificación inmediata. Al detectarse la caída del enlace, el administrador del sistema emite una alerta técnica al personal operativo delimitando las acciones de contingencia a seguir, regulando la carga de trabajo en el mostrador de atención y manteniendo los estándares de calidad en el servicio ofrecido a la comunidad estudiantil.

La adopción sistemática de este plan de contingencia faculta a la biblioteca del CECyT 5 para mantener su operatividad de forma ininterrumpida frente a imprevistos de hardware, protegiendo la persistencia de los registros y optimizando la resiliencia del entorno informático corporativo (*Sommerville, 2016*).

## IMPORTANCIA DEL BLINDAJE DE IMPLEMENTACIÓN

La gestión proactiva y sistemática de los riesgos durante la fase de despliegue no se limita a la salvaguarda de los componentes algorítmicos y estructurales del software; constituye un factor de influencia crítica en la construcción de la percepción analítica y actitudinal de los usuarios. El surgimiento de fallas técnicas recurrentes, interrupciones en el servicio o excepciones mal administradas durante las primeras fases de operación induce un ecosistema de desconfianza institucional, catalizando la resistencia pasiva al cambio y demeritando el valor percibido de la solución tecnológica frente a los métodos manuales tradicionales (*Davis, 1989*).

Por consiguiente, articular un entorno de blindaje operativo a través de un plan de mitigación rigurosamente estructurado provee las siguientes ventajas estratégicas en la ingeniería del proyecto:

- **Contención Eficiente del Impacto de Infancia**: Amortigua de forma controlada las anomalías y desviaciones técnicas imprevistas inherentes a la puesta en marcha de cualquier sistema de información, impidiendo que escalen hacia fallas catastróficas en el back-end (*Pressman & Maxim, 2020*).
- **Alineación de Procesos en la Transición Digital**: Faculta una migración fluida, ordenada y modular desde los flujos de trabajo análogos hacia los procesos automatizados en Microsoft Access, reduciendo los tiempos muertos en las ventanillas de atención del CECyT 5.

- Maximización del Índice de Aceptación Institucional: Al proveer interfaces estables, protocolos de contingencia claros y un entorno de alta disponibilidad, se disipa la fricción operativa, promoviendo un clima de apropiación tecnológica favorable entre el personal administrativo y la comunidad estudiantil.
- Aseguramiento Extrínseco de la Continuidad Operativa: Blindar las transacciones garantiza que, independientemente de los fallos de hardware o las degradaciones de la infraestructura de red local, los servicios de préstamo, devolución y control del acervo bibliográfico permanezcan activos bajo estándares de alta tolerancia a fallos (*Sommerville, 2016*).

En conclusión, la fase de implementación de la plataforma no debe ser interpretada de manera aislada como un hito estrictamente técnico o de codificación. Representa, en su sentido más amplio, un proceso de reingeniería y cambio organizacional profundo que demanda una planificación simétrica, mecanismos de control estocástico y una capacidad de adaptación continua para armonizar con éxito el comportamiento del software con las dinámicas humanas e institucionales de la organización.

## PRUEBAS

### FUNDAMENTOS DE LAS PRUEBAS

Las pruebas en sistemas de información constituyen un proceso de control crítico y sistemático dentro de la ingeniería de software, cuyo propósito medular es verificar, validar y evaluar la calidad, funcionalidad, seguridad y confiabilidad de una plataforma tecnológica antes de su despliegue definitivo en un entorno de producción real. Estas actividades permiten identificar desviaciones técnicas, mitigar riesgos operativos, auditar el cumplimiento de los requerimientos de diseño y garantizar que la arquitectura del software responda fielmente a los objetivos institucionales para los cuales fue proyectada (*Pressman & Maxim, 2020*).

En el ámbito de las ciencias de la computación, las pruebas han trascendido la noción tradicional de ser una simple revisión de la capa de presentación. Hoy en día, se definen como una disciplina analítica orientada a auditar el comportamiento del sistema bajo escenarios de estrés,

conurrencia masiva y condiciones límite. Este enfoque evalúa no solo la respuesta funcional síncrona del software, sino también sus atributos no funcionales, tales como el rendimiento del motor de base de datos, los perímetros de seguridad lógica y su capacidad de adaptación ante fallas de infraestructura.

## Evolución metodológica y cambio de paradigma

Históricamente, los modelos de desarrollo lineales (como la metodología tradicional en Cascada) relegaban la fase de pruebas al último eslabón del ciclo de vida del software. Este enfoque secuencial resultaba altamente ineficiente y costoso, debido a que los defectos de lógica o de requerimientos detectados en etapas tardías exigían reestructuraciones profundas en el código base, impactando negativamente en los costos y cronogramas del proyecto. Como consecuencia, las soluciones finales corrían el riesgo de quedar obsoletas o desalineadas respecto a las necesidades reales del entorno operativo (*Sommerville, 2016*).

Con el advenimiento de los marcos de trabajo ágiles (Agile) y la cultura DevOps, las pruebas se han transformado en un proceso continuo, transversal e integrado de forma nativa a lo largo de todo el ciclo de desarrollo. Esta integración temprana (enfoque Shift-Left) faculta la detección proactiva de anomalías en las primeras etapas de abstracción, promoviendo una retroalimentación iterativa constante, minimizando el costo de corrección de errores y asegurando una adaptabilidad óptima ante la evolución de las necesidades del sistema (*CircleCI, 2024*).

## El Ciclo de Vida de las Pruebas de Software (STLC)

Para estructurar operativamente esta disciplina, la ingeniería de calidad se apoya en el Ciclo de Vida de las Pruebas de Software (STLC - Software Testing Life Cycle), el cual organiza las actividades de validación a través de seis fases secuenciales y sistemáticas:

- Análisis de Requisitos de Calidad: Los ingenieros de pruebas examinan las especificaciones funcionales y técnicas del sistema. El objetivo no se limita a comprender el alcance del software, sino a dictaminar la viabilidad de su validación automatizada e identificar de forma temprana ambigüedades o riesgos latentes en las reglas de negocio.
- Planificación Estratégica de Pruebas: Fase de gobernanza donde se delimita el alcance del plan de pruebas (Test Plan). En esta etapa se establecen los objetivos de calidad, los recursos de hardware y software requeridos, los entornos de ejecución, los cronogramas operativos y los criterios de aceptación o éxito transaccional.

- Diseño y Modelado de Casos de Prueba: Consiste en el desarrollo analítico de escenarios de prueba (Test Cases). Cada caso describe minuciosamente las precondiciones, los datos de entrada específicos, la secuencia de pasos operativos y los resultados esperados en la base de datos, garantizando una cobertura total sobre las funcionalidades del sistema.
- Ejecución y Monitoreo de Pruebas: Fase medular en la que se aplican los escenarios diseñados sobre la plataforma. Los evaluadores registran de forma síncrona las respuestas del software, documentan los fallos del sistema y alimentan la bitácora de incidencias para su posterior corrección.
- Repruebas y Pruebas de Regresión: Una vez que el equipo de desarrollo solventa los defectos reportados, se ejecutan pruebas dirigidas para validar la efectividad del parche. Simultáneamente, se aplican pruebas de regresión para asegurar que las modificaciones de código o macros de VBA no hayan introducido de forma colateral nuevas anomalías en módulos estables preexistentes (*Coronel & Morris, 2019*).
- Cierre de Pruebas e Informe Ejecutivo: Fase de consolidación donde se suspende la ejecución, se analizan las métricas de cobertura de código y densidad de defectos, y se generan los reportes técnicos finales. Este hito permite evaluar la madurez del software y formalizar las lecciones aprendidas para futuros proyectos.

Esta estructura metodológica se acopla con prácticas modernas de Integración Continua (CI), donde cada actualización en el repositorio de código dispara automáticamente rutinas de prueba instrumentadas, asegurando la entrega inmediata de software funcional libre de regresiones (*CircleCI, 2024*).

Finalmente, en el ecosistema de los sistemas de información corporativos, estas actividades convergen en la fase de Pruebas de Sistema, la cual adopta un enfoque holístico de "extremo a extremo" (End-to-End). Generalmente estructuradas bajo la metodología de Caja Negra, estas pruebas prescinden del análisis de la lógica algorítmica interna de las macros o la sintaxis de las consultas SQL, concentrándose estrictamente en auditar que las salidas y comportamientos del software satisfagan con precisión la experiencia de usuario y las demandas operativas del entorno de ventanilla (*IBM, 2023*).

## IMPORTANCIA DEL PROCESO DE PRUEBAS

La trascendencia de las pruebas en los sistemas de información radica en su impacto directo sobre los niveles de calidad, la seguridad lógica y la viabilidad económica del software. En el panorama tecnológico contemporáneo, donde el software actúa como el núcleo operativo de

sectores críticos como la industria automatizada, la salud, los sistemas de transporte, las finanzas globales y las redes de comunicación, la aparición de una desviación técnica o fallo algorítmico puede desencadenar consecuencias catastróficas, tanto operacionales como financieras (*Pressman & Maxim, 2020*).

### DetECCIÓN TEMPRANA Y LA LEY DE AMPLIFICACIÓN DE COSTOS

Uno de los pilares fundamentales que justifican el despliegue de una estrategia de pruebas es la detección temprana de defectos (enfoque de prevención). Cuando un error de lógica, una inconsistencia de requerimientos o una anomalía de normalización se identifica en las fases iniciales del ciclo de vida del desarrollo, su mitigación y corrección es significativamente más sencilla, limpia y económica.

Por el contrario, de acuerdo con la curva de amplificación de costos del defecto, si una anomalía logra evadir los filtros de calidad y se manifiesta de forma tardía en el entorno de producción, los costos asociados a su reparación (parches de código, recubrimiento de datos dañados, tiempo de inactividad del sistema y despliegues de emergencia) se multiplican de manera exponencial (*Sommerville, 2016*).

Bajo esta premisa, la literatura especializada estima que las actividades de verificación y validación pueden absorber entre el 30% y el 50% del presupuesto total asignado al desarrollo de software (*Myers et al., 2011*). Aunque esta asignación de recursos representa una inversión sustancial dentro del proyecto, se encuentra plenamente justificada desde la perspectiva del retorno de inversión (ROI) y la gestión de riesgos, toda vez que el impacto económico, legal y reputacional de un fallo en operación supera con creces el costo del blindaje de calidad pre-producción.

### CASOS DE ESTUDIO HISTÓRICOS: EL COSTO DEL FALLO DE SOFTWARE

La historia de la ingeniería de software registra múltiples precedentes que evidencian la gravedad de omitir o subestimar los protocolos de pruebas en sistemas complejos. Entre los incidentes más representativos que demuestran la criticidad del software y su impacto directo en la sociedad y la vida humana se incluyen (*Myers et al., 2011*):

- Colapsos en Infraestructuras Aeroportuarias: Fallas de sincronización en los sistemas automatizados de logística y equipaje que derivaron en la paralización total de terminales aéreas, la cancelación masiva de vuelos y pérdidas millonarias para las aerolíneas.

- Inconsistencias en el Diseño Industrial y CAD/CAM: Desviaciones en las herramientas de software de modelado que provocaron desajustes físicos durante el ensamblaje de aeronaves comerciales, retrasando la cadena de producción global por meses y forzando costosas reingenierías de componentes.
- Defectos Críticos en Dispositivos Médicos (Sistemas de Radioterapia): Errores de desbordamiento de variables y condiciones de carrera en el código de control de aceleradores lineales que ocasionaron la administración accidental de sobredosis masivas de radiación, resultando en consecuencias fatales para múltiples pacientes.

### Beneficios estratégicos del aseguramiento de calidad (SQA)

Más allá de actuar como un mecanismo de contención de errores, la implementación sistemática de un plan de pruebas estructurado provee un conjunto de ventajas competitivas y técnicas para el sistema de información (*Sommerville, 2016*):

- Aseguramiento Multidimensional de la Calidad: Garantiza de forma simétrica que la solución tecnológica satisfaga con precisión tanto los requerimientos funcionales (operaciones explícitas del software) como los atributos no funcionales (seguridad, escalabilidad, portabilidad e integridad de los datos), entregando un producto final robusto.
- Maximización de la Experiencia y Satisfacción del Usuario (UX): El despliegue de una plataforma libre de interrupciones o comportamientos erráticos edifica un entorno digital confiable, incrementando los índices de adopción por parte de los operadores y mitigando la resistencia institucional ante la transición tecnológica.
- Optimización Dinámica del Rendimiento: A través de las pruebas de carga, estrés y rendimiento, es posible mapear el comportamiento del software ante flujos densos de información, identificando cuellos de botella en el procesador de consultas y optimizando la velocidad de respuesta del sistema en ventanilla.
- Mitigación de Riesgos y Continuidad del Negocio: Auditar el software de extremo a extremo antes de su liberación definitiva reduce drásticamente la probabilidad de incidentes críticos en entornos reales, asegurando la alta disponibilidad del servicio administrativo.
- Retroalimentación Orientada a la Mejora Continua del Proceso: Los datos recopilados en la bitácora de incidencias no solo evalúan el estado del software actual; actúan como métricas analíticas que permiten diagnosticar el proceso de desarrollo completo, identificando fallas en la captura de requerimientos o malas prácticas de codificación para prevenir su repetición en futuros proyectos.

## TIPOS DE PRUEBAS

Las actividades de verificación y validación dentro de la ingeniería de software se ramifican en diversas categorías taxonómicas en función de su alcance, propósito operativo, nivel de abstracción y enfoque metodológico. Esta diversificación de pruebas permite auditar la plataforma desde múltiples vectores lógicos, garantizando que tanto la microestructura del código como la macroestructura del sistema integrado operen de manera armónica, segura y libre de defectos (*Pressman & Maxim, 2020*).

### Pruebas según el nivel de desarrollo (Niveles de Prueba)

Esta clasificación responde a la evolución del software a lo largo de sus etapas de construcción e integración, estructurándose de forma jerárquica:

- Pruebas Unitarias (Unit Testing): Constituyen el nivel de validación más elemental y atómico. Se enfocan estrictamente en verificar el comportamiento correcto de componentes individuales y aislados del código (tales como funciones específicas, procedimientos o subrutinas de cálculo en las macros de VBA). Su objetivo principal es asegurar que la lógica de cada módulo procese las entradas y devuelva las salidas de forma exacta, abstrayéndose del resto de la arquitectura del sistema.
- Pruebas de Integración (Integration Testing): Se ejecutan inmediatamente después de las pruebas unitarias y están diseñadas para evaluar la interacción, conectividad y flujos de datos entre múltiples módulos o subsistemas preestablecidos. Este enfoque es crítico para detectar anomalías en la comunicación perimetral, problemas de acoplamiento, colisiones de variables y pérdida de integridad en la transferencia de parámetros lógicos (*Sommerville, 2016*).
- Pruebas de Sistema (System Testing): Implican la evaluación holística y de extremo a extremo (End-to-End) de la plataforma de software completamente integrada. Su propósito es auditar el comportamiento global del sistema simulando escenarios reales de operación intensiva. Este nivel valida simultáneamente los requerimientos funcionales y los atributos no funcionales, tales como la estabilidad del motor de datos, los tiempos de respuesta y los mecanismos de seguridad perimetral (*IBM, 2023*).

### Pruebas Funcionales

Las pruebas funcionales se orientan a verificar de manera síncrona que el comportamiento del sistema se alinee con las especificaciones y reglas de negocio documentadas. Este enfoque prescinde de la estructura algorítmica interna del back-end (metodología de Caja Negra) para

concentrarse exclusivamente en responder a la premisa operativa: ¿el sistema ejecuta con precisión las acciones para las cuales fue programado? (Coronel & Morris, 2019).

Dentro de este espectro funcional, las estrategias de validación más críticas comprenden:

- **Pruebas de Aceptación (User Acceptance Testing - UAT)**: Evaluaciones finales coordinadas con los usuarios operativos y administradores del sistema. Su objetivo es certificar que la interfaz, los reportes y los flujos transaccionales satisfacen las necesidades del día a día de la institución y cumplen con los criterios de éxito para autorizar el despliegue en producción.
- **Pruebas de Humo (Smoke Testing)**: Auditorías rápidas y preliminares que se ejecutan inmediatamente después de compilar una nueva versión o aplicar un parche en el sistema. Consisten en verificar que las funciones más básicas y críticas de la plataforma operen sin colapsar (por ejemplo, asegurar que la aplicación abra y permita conectarse a las tablas de Access), validando la estabilidad inicial del entorno antes de proceder con pruebas más exhaustivas.
- **Pruebas de Integración Funcional**: Análisis dirigidos que evalúan cómo se comportan e interactúan los distintos módulos del software en términos de procesos lógicos de negocio encadenados (por ejemplo, comprobar que la acción de registrar una devolución active correctamente la actualización del inventario y, a su vez, habilite de forma automática el formulario de encuestas de valoración), garantizando la consistencia del flujo de trabajo continuo.

## Pruebas no funcionales

Mientras que las pruebas funcionales validan qué acciones ejecuta la plataforma, las pruebas no funcionales se concentran en evaluar cómo las ejecuta. Este enfoque analiza las propiedades estructurales, el comportamiento operativo y las restricciones de diseño del sistema, dictaminando si la experiencia del usuario, la eficiencia del motor de datos y los perímetros de seguridad satisfacen los estándares de calidad exigidos para un entorno de producción estable (Pressman & Maxim, 2020).

Dentro del espectro de validación no funcional, se implementan las siguientes auditorías instrumentadas:

- **Pruebas de Rendimiento (Performance Testing)**: Evalúan la velocidad de respuesta, el consumo de recursos de hardware (CPU, memoria RAM) y la eficiencia general del

software bajo condiciones normales de operación. Su objetivo es detectar latencias en la ejecución de macros de VBA o cuellos de botella en el procesamiento de sentencias SQL.

- **Pruebas de Carga (Load Testing)**: Analizan el comportamiento dinámico del sistema cuando se somete a un volumen de transacciones síncronas equivalente a la demanda máxima esperada en la realidad institucional (por ejemplo, simular que múltiples terminales de biblioteca realizan préstamos y consultas de forma simultánea), verificando que los tiempos de respuesta se mantengan dentro de los umbrales tolerables (*Sommerville, 2016*).
- **Pruebas de Estrés (Stress Testing)**: Examinan el comportamiento del software más allá de sus límites teóricos de diseño para identificar sus puntos de ruptura y evaluar su capacidad de recuperación ante fallas catastróficas. En entornos de Microsoft Access, estas pruebas determinan cómo reacciona el archivo del back-end cuando se aproxima al límite físico de almacenamiento de 2 GB o ante picos inusuales de concurrencia.
- **Pruebas de Seguridad (Security Testing)**: Auditorías orientadas a identificar vulnerabilidades lógicas, vectores de inyección y fallas en los controles de acceso. Garantizan la efectividad del cifrado de la base de datos, protegen la integridad de las tablas relacionales contra alteraciones no autorizadas y validan el aislamiento de privilegios entre los roles de administrador y alumno (*Silberschatz et al., 2019*).
- **Pruebas de Usabilidad (Usability Testing)**: Evalúan el grado de ergonomía, la facilidad de aprendizaje y la experiencia de usuario (UX) que ofrece la interfaz gráfica. Su propósito es certificar que la navegación a través de los formularios sea intuitiva, fluida y libre de ambigüedades cognitivas.
- **Pruebas de Compatibilidad**: Verifican la consistencia operativa y visual del sistema al interactuar con diferentes arquitecturas de software, versiones del motor relacional de Office (como variantes de 32 y 64 bits) y configuraciones específicas del sistema operativo anfitrión.
- **Pruebas de Accesibilidad**: Aseguran que el diseño de las interfaces e instrumentos de captura cumpla con las directrices de inclusión y usabilidad universal, permitiendo que la plataforma sea operable de manera eficiente por usuarios con capacidades diferenciadas.
- **Pruebas de Localización (Localization Testing)**: Validan la correcta adaptación del software frente a contextos culturales, formatos de fecha y hora del sistema operativo, y parámetros regionales lingüísticos, garantizando la consistencia en el despliegue de máscaras de entrada y reportes ejecutivos (*IBM, 2023*).

# CASOS DE PRUEBA

## PRUEBAS DE INTERFAZ Y USABILIDAD

Access no es "responsivo" (no se adapta a celulares), pero sí debe ser funcional en diferentes resoluciones de monitor.

- **Resolución de Pantalla:** Abrir el formulario principal en una laptop con pantalla pequeña y en un monitor grande.
  - Resultado esperado: Los formularios deben tener barras de desplazamiento si no caben, o usar "Anclaje" para estirarse sin deformar los botones (*Microsoft, 2023*).
- **Navegación de Formularios (Tabulación):** Presionar la tecla Tab en un formulario de registro.
  - Resultado esperado: El cursor debe moverse de arriba hacia abajo y de izquierda a derecha. (En Access, esto se configura en la propiedad "Orden de tabulación") (*IBM, 2023*)
- **Botones de Navegación:** Usar las flechas de la parte inferior del formulario.
  - Resultado esperado: Debe permitir pasar al siguiente registro o crear uno nuevo sin errores de validación.

## PRUEBAS LOGICAS Y DE SEGURIDAD

En Access, la comunicación no es por API, sino por vinculación de tablas o consultas SQL internas.

- **Seguridad a nivel de Interfaz:** Intentar abrir el formulario de "Configuración" o "Reportes de Multas" con una cuenta de usuario normal.
  - Resultado esperado: Si implementaron un sistema de login manual, el botón debe aparecer bloqueado (Enabled = False) o invisible (*Microsoft, 2023*).
- **Integridad de Datos en Formularios:** Intentar cerrar un formulario de préstamo a medias (sin guardar).
  - Resultado esperado: Access suele guardar automáticamente; la prueba es verificar si el sistema lanza un aviso (BeforeUpdate) preguntando si desea guardar los cambios (*IBM, 2023*).

## PRUEBAS DE FUNCIONABILIDAD

- **Cuadros Combinados (Combobox) Dependientes:** Seleccionar un "Autor" en un desplegable.
  - Resultado esperado: El segundo desplegable ("Libros") debe actualizarse automáticamente para mostrar solo los libros de ese autor (esto se hace con el evento AfterUpdate y un .Requery) (Microsoft, 2023).
- **Generación de Reportes:** Hacer clic en el botón "Imprimir Reporte".
  - Resultado esperado: Se debe abrir la "Vista Previa de Informe" de Access, formateada correctamente para una hoja tamaño carta (IBM, 2023).

## CAPA DE DATOS

La capa de datos es responsable del almacenamiento, recuperación y gestión de la información del sistema. Para este propósito, se ha implementado un sistema gestor de bases de datos relacional (RDBMS) mediante el motor Microsoft Jet/ACE, el cual permite manejar la información de manera estructurada y eficiente; el uso de estructuras relacionales es fundamental para garantizar que los datos sean recuperables y seguros ante grandes volúmenes de registros (*Silberschatz et al., 2019*).

La base de datos está diseñada siguiendo principios de normalización, lo que evita redundancias y asegura la integridad de los datos en entornos multiusuario (Coronel & Morris, 2018). Entre las principales tablas del sistema se encuentran:

- **Usuarios:** Datos generales de los miembros.
- **Libros:** Información bibliográfica general.
- **Ejemplares:** Registro físico de cada unidad (permite distinguir si hay 5 copias de un mismo libro).

Cada tabla está relacionada mediante claves primarias y foráneas, permitiendo establecer vínculos lógicos. Por ejemplo, un préstamo está asociado mediante integridad referencial a un usuario y a un ejemplar específico. Además, se han establecido protocolos de respaldo manual y compacto de la base de datos, garantizando la disponibilidad de la información ante posibles corrupciones de archivo.

## ARQUITECTURA BASADA EN SERVICIOS (SOA)

El sistema incorpora principios de Arquitectura Orientada a Servicios (SOA), lo que permite que sus funcionalidades estén expuestas como servicios independientes. La orientación a servicios es un paradigma de diseño que garantiza que los componentes del software sean autónomos y reutilizables, facilitando la interoperabilidad entre diferentes plataformas (Erl, 2016).

Esta estructura facilita la integración con otros sistemas externos, como plataformas educativas o aplicaciones móviles. Una arquitectura SOA permite que cada servicio sea escalable, asegurando que el sistema de gestión de la biblioteca pueda evolucionar o recibir actualizaciones en módulos específicos sin afectar el funcionamiento global de la aplicación (Sommerville, 2016).

## SISTEMA DE SEGURIDAD

La arquitectura del sistema incluye múltiples mecanismos de seguridad diseñados para proteger la integridad de la información y garantizar un acceso controlado. La seguridad informática debe cubrir tanto la confidencialidad como la disponibilidad de los datos (Stallings, 2017). Entre los mecanismos implementados se encuentran:

- **Autenticación y Autorización:** Se utiliza un control de acceso basado en roles (RBAC). Esto asegura que solo el personal autorizado (administrador o bibliotecario) acceda a funciones críticas como la modificación de registros de multas o la eliminación de usuarios (Pressman & Maxim, 2020).
- **Encriptación y Protección de Datos:** Se aplican técnicas de cifrado para proteger información sensible. En el contexto de bases de datos, es de suma importancia prevenir la Inyección SQL, asegurando que las consultas al motor de búsqueda estén parametrizadas y no permitan la ejecución de código malicioso (Silberschatz et al., 2019).
- **Protocolos de Transmisión:** Para las interfaces que operan en red, se asegura el uso de protocolos cifrados que protegen la información en tránsito.
- **Auditoría y Trazabilidad:** El sistema registra las actividades relevantes en bitácoras de eventos (logs). Esto permite realizar auditorías para detectar anomalías o errores humanos en el manejo de la biblioteca.

## ESCALABILIDAD Y MANTENIMIENTO

La arquitectura del sistema ha sido diseñada para soportar un crecimiento constante tanto en el volumen de datos (libros y registros) como en el número de usuarios concurrentes. La escalabilidad es la capacidad de un sistema para entregar servicios de manera efectiva mientras aumenta la demanda de procesamiento (Sommerville, 2016). Para asegurar este rendimiento, se han implementado las siguientes estrategias:

- **Optimización de Consultas SQL**: Se han diseñado índices en las tablas de mayor tráfico (como Libros y Préstamos). La indexación adecuada reduce drásticamente el tiempo de respuesta del RDBMS al evitar el escaneo completo de las tablas durante las búsquedas (Silberschatz et al., 2019).
- **Uso de Caché y Memoria**: Se han implementado mecanismos para el almacenamiento temporal de consultas frecuentes, lo que minimiza el acceso repetitivo al disco duro y agiliza la visualización en la capa de presentación.
- **Balanceo y Despliegue**: Aunque el sistema opera actualmente en un entorno controlado, la arquitectura SOA y el diseño de la base de datos permiten un despliegue en la nube (como en Azure o AWS). Esta flexibilidad asegura que el sistema pueda escalar vertical u horizontalmente mediante balanceadores de carga en caso de una expansión institucional (Pressman & Maxim, 2020).

## DESCRIPCIÓN FUNCIONAL DEL SISTEMA

El sistema de gestión de bibliotecas privadas se basa en la interconexión de procesos operativos que garantizan la integridad de la información y la eficiencia en el servicio. Los requerimientos funcionales definen los servicios que el sistema debe proporcionar, reaccionando a entradas específicas y comportándose de manera predecible ante diversas situaciones (Sommerville, 2016).

## GESTIÓN DE USUARIOS

El proceso de registro permite dar de alta a nuevos miembros tras una validación de datos personales. Este módulo se complementa con el proceso de autenticación, que garantiza que solo usuarios autorizados accedan según su rol (administrador, bibliotecario o usuario), aplicando principios de seguridad de acceso controlado (Pressman & Maxim, 2020).

## GESTIÓN DE CATALOGO Y MOTOR DE BUSQUEDA

El catálogo constituye el núcleo del sistema. Permite a los bibliotecarios gestionar libros y ejemplares, clasificándolos por categorías, autores y editoriales. Para la recuperación de esta información, el sistema incorpora un motor de búsqueda avanzado que utiliza filtros por ISBN o título, optimizando la experiencia del usuario (*Silberschatz et al., 2019*).

## RESULTADOS

La implementación del sistema de gestión para bibliotecas privadas culminó con una plataforma funcional que integra de manera cohesiva la administración de usuarios, ejemplares y procesos financieros. Según *Pressman y Maxim (2020)*, los resultados de un proyecto de software deben evaluarse en función de su capacidad para satisfacer los requisitos del usuario y su fiabilidad operativa.

### EFICIENCIA EN PROCESO DE DATOS

Tras la ejecución de las pruebas, se observó una mejora significativa en la velocidad de recuperación de información. Gracias a la optimización de consultas y la normalización de la base de datos en Access, el motor de búsqueda entrega resultados en tiempos de respuesta menores a un segundo, incluso con una carga simulada de registros extensos. Un diseño relacional sólido es el factor determinante para mantener el rendimiento cuando el volumen de datos aumenta (*Silberschatz et al., 2019*).

### INTERFAZ Y USABILIDAD

La capa de presentación ha demostrado ser intuitiva para los dos roles principales (Bibliotecario y Usuario). Mediante el uso de formularios dinámicos y menús de navegación simplificados, se redujo el tiempo de capacitación necesario para operar el sistema. Una interfaz exitosa es aquella que permite al usuario concentrarse en su tarea (gestionar libros) en lugar de en la complejidad del software (*Sommerville, 2016*).

### SEGURIDAD Y CONTROL DE ACCESO

Los resultados de las pruebas de seguridad confirmaron que el sistema restringe adecuadamente las funciones administrativas. Se validó que los usuarios con rol de "Lector" no pueden acceder a las tablas de configuración o reportes de inventario, cumpliendo con el principio de privilegios mínimos para la protección de datos sensibles (*Stallings, 2017*).

# ANÁLISIS DE ERRORES Y ESTRATEGIAS DE MEJORA CONTINUA.

La implementación de un sistema de gestión bibliográfico no es un proceso estático. Durante el desarrollo y puesta en marcha del proyecto, se identificaron áreas de oportunidad que, de no atenderse, podrían comprometer la viabilidad del software. La evolución del software es una fase inevitable donde el sistema debe adaptarse a las realidades operativas y organizacionales (Sommerville, 2016).

## GESTIÓN ESTRATEGICA Y DEFINICIÓN DE OBJETIVOS

Un hallazgo crítico en la gestión de sistemas es la ambigüedad en las metas iniciales. La falta de objetivos claros impide medir el éxito del sistema.

Mejora: Se propone la adopción de la metodología SMART (Específicos, Medibles, Alcanzables, Relevantes y Temporales) para futuras actualizaciones. Como señala Pressman y Maxim (2020), el éxito de la ingeniería de software depende de una visión compartida entre los desarrolladores y las partes interesadas.

## FACTOR HUMANO

Un sistema técnico es solo una herramienta; su efectividad depende del personal que lo opera. Se detectó que la falta de capacitación puede derivar en una subutilización de las funciones de Access (como la generación de reportes).

Mejora: Establecer un programa de capacitación integral y continua que fomente una cultura de calidad. La capacitación reduce la resistencia al cambio y asegura que los procedimientos se sigan según lo documentado (Pressman & Maxim, 2020). Incluso para disminuir los posibles problemas que se le puedan presentar a los usuarios se llevara a cabo un manual para estos, con la finalidad de que se sientan familiarizados con el programa incluso antes de comenzar a utilizarlo.

## DOCUMENTACIÓN Y CUMPLIMIENTO NARRATIVO

La ausencia de documentación técnica y de usuario genera inconsistencias y errores operativos. Además, en el manejo de bibliotecas, ignorar el cumplimiento normativo (como la Ley de Protección de Datos Personales en Posesión de Particulares) puede traer sanciones legales.

Mejora: Implementar un sistema de gestión documental donde cada proceso esté estandarizado. La documentación detallada del esquema de la base de datos es esencial para el mantenimiento preventivo y correctivo (*Silberschatz et al., 2019*).

## MANTENIMIENTO

### CONCEPTO DE MANTENIMIENTO DE SOFTWARE

El mantenimiento de software es el conjunto de actividades que se realizan después de que un sistema ha sido implementado y puesto en funcionamiento dentro de una organización. Su propósito principal es asegurar que el sistema continúe operando correctamente, adaptándose a los cambios del entorno y cumpliendo con las necesidades de los usuarios.

En los sistemas de información, el mantenimiento no es una fase opcional, sino una parte esencial del ciclo de vida del software. Esto se debe a que, con el tiempo, los sistemas pueden presentar errores, volverse obsoletos o dejar de ser compatibles con nuevas tecnologías. Por ello, el mantenimiento permite prolongar la vida útil del sistema y mantener su eficiencia operativa (*Sommerville, 2016*).

Además, el mantenimiento no solo implica corregir fallas, sino también realizar mejoras continuas, optimizar procesos y garantizar que el sistema siga siendo útil y competitivo dentro del entorno tecnológico actual.

## IMPORTANCIA DEL MANTENIMIENTO

El mantenimiento es fundamental porque garantiza que los sistemas de información sigan funcionando de manera eficiente, segura y confiable. Un sistema que no recibe mantenimiento puede volverse lento, inseguro o incluso dejar de funcionar, afectando directamente las operaciones de una empresa.

Entre sus principales beneficios se encuentra la mejora del rendimiento del sistema, ya que permite optimizar procesos y reducir errores. También contribuye a cerrar vulnerabilidades de seguridad, lo cual es especialmente importante ante el aumento constante de ciberataques. Asimismo, ayuda a adaptar el software a nuevas tecnologías, lo que permite que las organizaciones se mantengan actualizadas y competitivas en el mercado.

Por otro lado, el mantenimiento influye directamente en la satisfacción del usuario, ya que un sistema funcional, rápido y actualizado facilita el trabajo diario y mejora la experiencia de uso (*Dinnova, 2024*).

## TIPOS DE MANTENIMIENTO DE SOFTWARE

### MANTENIMIENTO CORRECTIVO

El mantenimiento correctivo se realiza cuando el sistema presenta errores o fallas que afectan su funcionamiento. Este tipo de mantenimiento es reactivo, ya que se aplica después de detectar un problema.

Este tipo de mantenimiento es uno de los más frecuentes, especialmente después de la implementación del software, cuando comienzan a detectarse errores que no fueron identificados durante las pruebas iniciales (*OK Hosting, 2024*).

### MANTENIMIENTO ADAPTATIVO

El mantenimiento adaptativo consiste en realizar modificaciones al sistema para que pueda funcionar correctamente en nuevos entornos tecnológicos. Esto incluye cambios en sistemas operativos, hardware, dispositivos móviles o plataformas digitales.

Su objetivo es asegurar que el sistema siga siendo funcional y compatible, evitando que quede obsoleto frente a nuevos avances tecnológicos (*Pressman & Maxim, 2020*).

## MANTENIMIENTO PERFECTIVO

El mantenimiento perfectivo se enfoca en mejorar el rendimiento, la eficiencia y la funcionalidad del sistema. A diferencia del correctivo, no se realiza por errores, sino para optimizar el software.

Este mantenimiento puede incluir la mejora de la interfaz de usuario, la reducción de tiempos de carga, la eliminación de código innecesario o la incorporación de nuevas funciones que faciliten el uso del sistema.

Gracias a este tipo de mantenimiento, los sistemas pueden evolucionar y adaptarse a las necesidades cambiantes de los usuarios, mejorando su experiencia y aumentando la productividad (*Sommerville, 2016*).

## MANTENIMIENTO PREVENTIVO

El mantenimiento preventivo tiene como objetivo anticiparse a posibles fallos antes de que ocurran. Se basa en revisiones periódicas, actualizaciones y análisis del sistema para detectar vulnerabilidades o problemas potenciales.

Además, permite mantener el sistema estable y confiable a lo largo del tiempo (*Stallings, 2017*).

## PROCESO DE MANTENIMIENTO

El mantenimiento de software debe seguir un proceso estructurado para garantizar su efectividad. Este proceso incluye varias etapas que permiten identificar, analizar y solucionar problemas de manera organizada.

Primero se realiza la identificación de los componentes que requieren mantenimiento. Luego se lleva a cabo un análisis para evaluar los cambios necesarios y su impacto en el sistema. Posteriormente, se diseña la solución, se implementa y se realizan pruebas para verificar su correcto funcionamiento.

Finalmente, se actualiza el sistema y se documentan los cambios realizados. Este proceso asegura que el mantenimiento se realice de manera eficiente y reduce el riesgo de errores adicionales (*Pressman & Maxim, 2020*).

## DOCUMENTACIÓN Y RETROALIMENTACIÓN

La documentación es un elemento clave en el mantenimiento de software, ya que permite registrar los cambios realizados, los errores detectados y las soluciones aplicadas. Esto facilita el trabajo de los desarrolladores y permite que nuevos integrantes del equipo comprendan el funcionamiento del sistema.

Por otro lado, la retroalimentación de los usuarios es fundamental, ya que proporciona información sobre cómo se utiliza el sistema en la práctica y qué aspectos pueden mejorarse. Escuchar a los usuarios permite realizar mejoras más efectivas y alineadas con sus necesidades (*Sommerville, 2016*).

## RETOS Y SOLUCIONES EN EL MANTENIMIENTO

El mantenimiento de software puede enfrentar diversos desafíos, como la falta de recursos, problemas técnicos inesperados o una documentación insuficiente. Estos factores pueden dificultar la implementación de mejoras y la resolución de errores.

Para superar estos retos, es importante adoptar un enfoque proactivo, capacitar constantemente al personal, utilizar herramientas de monitoreo y análisis, y mantener una documentación clara y actualizada. De esta manera, se pueden prevenir problemas y mejorar la eficiencia del sistema (*Pressman & Maxim, 2020*).

## CORRECCIÓN DE ERRORES

Dentro del ciclo de vida de un sistema de información, la etapa de mantenimiento es fundamental para garantizar su correcto funcionamiento a lo largo del tiempo. En el caso de un sistema de administración de bibliotecas, esta etapa adquiere una gran relevancia debido a la cantidad de información que se maneja y a la necesidad de mantener los procesos operativos sin interrupciones. Uno de los aspectos más importantes dentro del mantenimiento es la corrección de errores, también conocida como mantenimiento correctivo.

La corrección de errores consiste en identificar, analizar y solucionar fallas que se presentan durante el uso del sistema. Estas fallas pueden originarse por diversas causas, como errores en la programación, problemas en la base de datos, fallas en la interfaz de usuario o incluso errores derivados del uso incorrecto del sistema por parte de los usuarios. En un sistema de gestión bibliotecaria, estos errores pueden afectar directamente procesos críticos como el registro de libros, la gestión de préstamos y devoluciones, o el almacenamiento de encuestas realizadas a los usuarios.

Uno de los tipos más comunes de errores en este tipo de sistemas son los errores de captura de datos. Estos ocurren cuando la información ingresada por los administradores o usuarios no es correcta o está incompleta. Por ejemplo, puede suceder que un libro sea registrado sin autor, con un título incorrecto o con datos duplicados. Para evitar este tipo de problemas, el sistema debe contar con mecanismos de validación que permitan verificar la información antes de almacenarla en la base de datos.

Otro tipo de errores frecuentes son los errores lógicos o de programación, los cuales se presentan cuando el sistema no realiza correctamente las operaciones para las que fue diseñado. Por ejemplo, el sistema podría permitir el préstamo de un libro que ya no está disponible, o podría no actualizar correctamente la información después de una devolución. Estos errores suelen detectarse durante el uso del sistema y requieren la intervención del equipo de desarrollo para ser corregidos mediante la modificación del código fuente.

También pueden presentarse errores en la base de datos, los cuales afectan la integridad y consistencia de la información almacenada. Estos errores pueden incluir la pérdida de registros, la duplicación de datos o inconsistencias entre diferentes tablas. En un sistema de biblioteca, este tipo de errores puede provocar problemas graves, como la pérdida de información sobre los libros o los usuarios, lo que afecta directamente el funcionamiento de la institución.

Además, existen errores de interfaz de usuario, los cuales afectan la forma en que los usuarios interactúan con el sistema. Por ejemplo, una interfaz confusa, botones que no funcionan correctamente o mensajes de error poco claros pueden dificultar el uso del sistema y generar errores adicionales. Por esta razón, es importante realizar pruebas de usabilidad que permitan identificar y corregir estos problemas.

El proceso de corrección de errores comienza con la detección de la falla. Esto puede ocurrir a través de reportes de los usuarios, monitoreo del sistema o pruebas realizadas por el equipo técnico. Una vez identificado el error, se procede a su análisis para determinar la causa del problema. Este análisis puede incluir la revisión del código, la verificación de la base de datos y la reproducción del error en un entorno de prueba.

Después de identificar la causa del error, se realiza la corrección del problema, lo cual puede implicar la modificación del código del sistema, la actualización de la base de datos o la mejora de la interfaz de usuario. Una vez realizada la corrección, es necesario realizar pruebas para verificar que el problema ha sido resuelto y que no se han generado nuevos errores.

Finalmente, la corrección de errores debe ir acompañada de un proceso de documentación, en el cual se registren las fallas detectadas, las soluciones implementadas y las mejoras realizadas.

Esta documentación es importante para facilitar el mantenimiento futuro del sistema y evitar la repetición de errores similares (*Pressman & Maxim, 2020*).

## ACTUALIZACIÓN DEL SISTEMA

Además de la corrección de errores, otro aspecto fundamental dentro de la etapa de mantenimiento de un sistema de información es la actualización del sistema, también conocida como mantenimiento evolutivo o adaptativo. Este proceso consiste en realizar modificaciones y mejoras al sistema con el objetivo de adaptarlo a nuevas necesidades, incorporar nuevas funcionalidades y mejorar su rendimiento.

En el contexto de la gestión de bibliotecas, la actualización del sistema es especialmente importante debido a los cambios constantes en las necesidades de los usuarios, el crecimiento de la colección bibliográfica y el avance de la tecnología. Un sistema que no se actualiza con el tiempo puede volverse obsoleto, dificultando la administración de la biblioteca y reduciendo la calidad del servicio ofrecido.

Las actualizaciones del sistema pueden clasificarse en diferentes tipos. Uno de ellos es el mantenimiento adaptativo, el cual se enfoca en adaptar el sistema a cambios en su entorno. Por ejemplo, si la biblioteca decide implementar un nuevo sistema de clasificación de libros o integrar el acceso en línea al catálogo, será necesario realizar modificaciones en el sistema para adaptarse a estos cambios.

Otro tipo es el mantenimiento evolutivo, que consiste en agregar nuevas funcionalidades al sistema para mejorar su desempeño. En el caso de un sistema de biblioteca, esto puede incluir la incorporación de módulos de recomendación de libros, mejoras en el sistema de encuestas, integración con aplicaciones móviles o la implementación de notificaciones automáticas para recordar a los usuarios la fecha de devolución de los libros.

También existe el mantenimiento preventivo, el cual tiene como objetivo mejorar la calidad del sistema y prevenir posibles fallas en el futuro. Este tipo de mantenimiento puede incluir la optimización del código, la mejora de la estructura de la base de datos y la actualización de las herramientas tecnológicas utilizadas.

El proceso de actualización del sistema comienza con la identificación de necesidades de mejora, las cuales pueden surgir a partir de la retroalimentación de los usuarios, el análisis del desempeño del sistema o los cambios en los objetivos de la biblioteca. Una vez identificadas estas necesidades, se procede a planificar las modificaciones que se realizarán en el sistema.

Posteriormente, se lleva a cabo el diseño de las actualizaciones, en el cual se definen los cambios que se implementarán y la forma en que estos se integrarán al sistema existente. Este diseño debe considerar aspectos como la compatibilidad con la base de datos, la facilidad de uso y la seguridad de la información.

Una vez definido el diseño, se realiza la implementación de las mejoras, lo cual implica la modificación del código del sistema, la actualización de la base de datos y la incorporación de nuevas funciones. Después de la implementación, es necesario realizar pruebas para verificar que las actualizaciones funcionan correctamente y que no afectan el funcionamiento de otras partes del sistema.

Finalmente, las actualizaciones deben ir acompañadas de un proceso de capacitación y documentación, para que los usuarios puedan comprender y utilizar las nuevas funciones del sistema. Esto es especialmente importante en el caso de los administradores de la biblioteca, quienes deben adaptarse a los cambios para aprovechar al máximo las nuevas herramientas.

La actualización del sistema también permite mejorar la experiencia de los usuarios, ya que facilita el acceso a la información, mejora la velocidad de respuesta del sistema y ofrece nuevas funcionalidades que hacen más eficiente el uso de la biblioteca. Además, contribuye a mantener el sistema seguro y protegido frente a posibles amenazas o vulnerabilidades (*Sommerville, 2016*).

## SOPORTE AL USUARIO

El soporte al usuario es el conjunto de servicios que se brindan para ayudar a los usuarios a utilizar correctamente los sistemas de información y resolver problemas técnicos que puedan surgir durante su uso. Este soporte no solo se enfoca en solucionar fallas, sino también en prevenirlas, capacitar a los usuarios y garantizar que los sistemas funcionen de manera adecuada. En este sentido, el soporte es esencial para mantener la continuidad de las operaciones dentro de una organización.

Por lo tanto, este servicio resulta fundamental porque permite que los sistemas de información se utilicen de manera eficiente. Sin un soporte adecuado, los usuarios pueden enfrentar dificultades que afectan su productividad y generan retrasos en las actividades. Además, el soporte contribuye a la seguridad de la información, ya que ayuda a prevenir errores humanos y promueve buenas prácticas en el uso de la tecnología. También facilita la adaptación a nuevas herramientas y sistemas, lo que permite a las organizaciones mantenerse actualizadas (*SOSMATIC, 2024*).

## SOPORTE INTERNO

El soporte interno es aquel que se brinda dentro de la organización, generalmente a través de una mesa de ayuda. Este es el primer nivel de atención y se encarga de resolver problemas comunes y brindar asistencia directa a los usuarios.

Puede realizarse de forma presencial, telefónica o remota. La atención remota es especialmente útil, ya que permite resolver problemas sin necesidad de desplazarse, lo que reduce costos y tiempo (*ACE Project, s.f.*).

## SOPORTE EXTERNO

El soporte externo es proporcionado por empresas o especialistas externos cuando los problemas son más complejos y no pueden ser resueltos internamente.

Este tipo de soporte suele ser más costoso, pero es necesario para realizar reparaciones especializadas o atender situaciones críticas. Para mejorar su eficiencia, muchas organizaciones establecen acuerdos de servicio que garantizan tiempos de respuesta y costos definidos (*Sommerville, 2016*).

## BENEFICIOS DEL SOPORTE AL USUARIO

El soporte al usuario aporta múltiples beneficios, como el aumento de la productividad, la reducción de costos por fallas, la mejora en la seguridad de la información y una mejor experiencia para el usuario.

Además, permite que las organizaciones se adapten a los cambios tecnológicos y mantengan un funcionamiento continuo y eficiente (*Pressman & Maxim, 2020*).

## MEJORAS EN EL SISTEMA

La mejora continua de un sistema de gestión bibliotecaria debe concebirse como un proceso estratégico orientado no solo a la corrección de fallas, sino a la optimización permanente de su desempeño, escalabilidad y adaptabilidad a nuevos requerimientos. En entornos donde se emplea Microsoft Access, estas mejoras adquieren especial relevancia, ya que esta plataforma permite evolucionar progresivamente desde soluciones básicas hasta sistemas más complejos mediante el uso de herramientas avanzadas.

En este contexto, la incorporación de automatización, el fortalecimiento de la estructura de datos, la mejora de la experiencia del usuario y la optimización de la arquitectura del sistema

constituyen pilares fundamentales para garantizar un funcionamiento eficiente y sostenible en el tiempo (*Pressman & Maxim, 2020*).

## FORTALECIMIENTO DE LA INTEGRIDAD REFERENCIAL

La integridad referencial es un componente crítico en cualquier base de datos relacional. En un sistema bibliotecario, garantiza que las relaciones entre entidades (libros, autores, usuarios, préstamos) se mantengan coherentes y libres de inconsistencias. Para ello, es necesario implementar:

- Restricciones de integridad, que impidan la eliminación o modificación de registros relacionados.
- Relaciones uno a muchos correctamente definidas, por ejemplo, entre libros y préstamos.
- Reglas de actualización y eliminación en cascada, cuando sea pertinente.

Un ejemplo claro es impedir que un libro sea eliminado si se encuentra asociado a un préstamo activo, evitando así la pérdida de información crítica. Este tipo de control no solo mejora la confiabilidad del sistema, sino que también facilita auditorías y procesos de seguimiento dentro de la biblioteca (*Silberschatz et al., 2019*).

## NORMALIZACIÓN Y ESTABILIDAD DE LA BASE DE DATOS

La normalización de datos es un proceso esencial para estructurar la base de datos de manera eficiente, evitando redundancias y mejorando la organización de la información. Aplicar correctamente las formas normales permite:

- Minimizar la duplicidad de datos.
- Reducir inconsistencias.
- Facilitar actualizaciones y mantenimiento.
- Optimizar el rendimiento de consultas.

En un sistema bibliotecario, esto se traduce en una mejor gestión de entidades como autores, libros, categorías y usuarios, permitiendo que el sistema escale sin perder eficiencia.

Gracias a una adecuada normalización, el sistema puede evolucionar desde un pequeño catálogo hasta una base de datos de gran volumen, manteniendo tiempos de respuesta adecuados y una estructura lógica consistente (*Silberschatz et al., 2019*).

## OPTIMIZACIÓN DE LA EXPERIENCIA AL USUARIO

La experiencia del usuario (UX) es un factor determinante en la aceptación y uso continuo del sistema. Un sistema funcional pero difícil de usar puede generar rechazo, mientras que uno intuitivo facilita la adopción y mejora la productividad. Entre las mejoras más relevantes se encuentran:

- Búsquedas dinámicas en tiempo real, que permiten filtrar información mientras el usuario escribe.
- Formularios simplificados, que reducen la carga cognitiva del usuario.
- Diseño ergonómico, con distribución lógica de elementos.
- Retroalimentación visual inmediata, como mensajes de confirmación o advertencia.

Asimismo, el uso de colores adecuados y tipografías legibles contribuye a reducir la fatiga visual, especialmente en jornadas prolongadas de trabajo.

Estas mejoras no solo incrementan la eficiencia, sino que también fortalecen la percepción profesional del sistema (*Pressman & Maxim, 2020*).

## ARQUITECTURA MEJORADA

Una de las mejoras más importantes a nivel estructural es la implementación de la arquitectura de base de datos dividida (Split Database) en Microsoft Access. Este modelo separa el sistema en dos componentes:

- Back-end: Contiene las tablas y los datos.
- Front-end: Incluye formularios, consultas, reportes y lógica del sistema.

Esta separación ofrece múltiples ventajas:

- Mayor seguridad, al proteger los datos frente a modificaciones accidentales.
- Facilidad de mantenimiento, permitiendo actualizar la interfaz sin afectar la base de datos.
- Mejor rendimiento, especialmente en entornos multiusuario.
- Reducción del riesgo de corrupción de datos, un problema común en Access cuando múltiples usuarios acceden simultáneamente.

Además, esta arquitectura permite distribuir el front-end a diferentes equipos, manteniendo un único repositorio central de datos (*Silberschatz et al., 2019*).

## PROYECCIÓN A FUTURO Y MEJORA CONTINUA

El proceso de mejora no debe detenerse una vez implementado el sistema. Es fundamental establecer una cultura de evaluación constante que permita identificar nuevas necesidades y oportunidades de optimización.

- Entre las mejoras futuras que pueden considerarse se encuentran:
- Integración con sistemas externos o plataformas digitales.
- Implementación de respaldos automáticos.
- Migración a sistemas más robustos si el volumen de datos crece significativamente.
- Incorporación de indicadores de desempeño (KPIs) para la gestión bibliotecaria.

De esta manera, el sistema no solo responde a las necesidades actuales, sino que se adapta a los cambios del entorno, garantizando su vigencia a largo plazo (*Sommerville, 2016*).

## RESPALDO DE INFORMACIÓN

En el contexto de sistemas desarrollados en Microsoft Access, el respaldo de la información debe entenderse como una estrategia integral orientada a garantizar la continuidad operativa y la protección de los datos ante posibles fallos. A diferencia de otros gestores de bases de datos más robustos, Access trabaja con archivos locales, lo que lo hace más vulnerable a problemas como interrupciones eléctricas, fallas en la red o cierres inesperados del sistema. Por esta razón, el respaldo no puede limitarse a la simple copia del archivo, sino que debe formar parte de un proceso estructurado de prevención y recuperación.

Dentro de este enfoque, el mantenimiento preventivo adquiere un papel fundamental. Antes de realizar cualquier respaldo, es recomendable aplicar procesos como la compactación y reparación de la base de datos, ya que estos permiten optimizar el archivo, eliminar fragmentaciones internas y corregir posibles inconsistencias. De esta manera, se asegura que la copia generada sea íntegra, funcional y confiable, evitando trasladar errores a los respaldos. Esta

práctica no solo mejora el rendimiento del sistema, sino que también incrementa la seguridad de la información almacenada.

En cuanto a la implementación de una estrategia de recuperación, es importante considerar modelos ampliamente aceptados como la regla 3-2-1, la cual establece que deben mantenerse al menos tres copias de la información, almacenadas en dos medios distintos y con una de ellas fuera del sitio físico. En un sistema desarrollado en Microsoft Access, esta estrategia se complementa eficazmente con la división de la base de datos en front-end y back-end. Esta separación permite concentrar los respaldos frecuentes en el archivo que contiene los datos, mientras que la interfaz y la lógica del sistema solo requieren respaldo cuando se realizan modificaciones, optimizando así el uso de recursos y facilitando la administración del sistema.

Por otro lado, la automatización del proceso de respaldo constituye un elemento clave para garantizar su correcta ejecución. Depender exclusivamente de acciones manuales incrementa el riesgo de omisiones, por lo que resulta recomendable implementar mecanismos automatizados, ya sea mediante herramientas del sistema operativo o a través de programación en VBA dentro del propio Access. Estos mecanismos pueden encargarse de generar copias en intervalos definidos, verificar la fecha del último respaldo o alertar al administrador cuando no se haya cumplido con la política establecida. De esta manera, se reduce la dependencia del factor humano y se asegura la consistencia en la protección de los datos.

No obstante, la existencia de respaldos no garantiza por sí sola la recuperación de la información. Es indispensable realizar pruebas periódicas de restauración que permitan comprobar que los archivos respaldados funcionan correctamente. Un respaldo que no ha sido validado puede resultar inútil en una situación crítica, por lo que la verificación constante se convierte en un componente esencial del plan de seguridad. Estas pruebas también permiten identificar posibles fallas en los procedimientos y mejorar la capacidad de respuesta ante incidentes.

Asimismo, el respaldo de la información debe contemplar medidas de seguridad que protejan los datos frente a accesos no autorizados. Esto implica almacenar las copias en ubicaciones seguras, restringir su acceso al personal autorizado e incluso aplicar mecanismos de cifrado cuando sea necesario. En un entorno bibliotecario, donde se manejan datos de usuarios y registros históricos, esta protección adquiere especial relevancia, ya que contribuye a preservar la confidencialidad y la integridad de la información.

Desde una perspectiva más amplia, la gestión adecuada de respaldos representa un componente esencial de la calidad del sistema. Como señalan diversos autores en el campo de la ingeniería de software, la confiabilidad de un sistema está estrechamente relacionada con su

capacidad de recuperación ante fallos. En este sentido, un sistema de gestión bibliotecaria que cuenta con un plan de respaldo sólido no solo protege su información, sino que también fortalece la confianza de sus usuarios y asegura la continuidad del servicio (*Silberschatz et al., 2019*).

# BIBLIOGRAFIA

- I. **Alonso-Arévalo, J.** (2019). *Alfabetización informacional en la biblioteca universitaria del siglo XXI*. Ediciones Universidad de Salamanca.
- II. **American Library Association.** (2021). *Standards for libraries in higher education*. American Library Association.
- III. **Arjonilla, S. J., & Medina, J. A.** (2018). *La gestión de los sistemas de información en la empresa*. Pirámide.
- IV. **Baeza-Yates, R., & Ribeiro-Neto, B.** (2011). *Modern information retrieval: The concepts and technology behind search* (2a ed.). ACM Press.
- V. **Calva González, J. J.** (2016). *Las necesidades de información y el comportamiento informativo de los usuarios*. UNAM, Instituto de Investigaciones Bibliotecológicas y de la Información.
- VI. **Cano, V.** (2020). Evaluación de software libre para la gestión de bibliotecas escolares. *Revista Española de Documentación Científica*, 43(2), Artículo e265. <https://doi.org/10.3989/redc.2020.2.1732>
- VII. **CircleCI.** (2024). *Continuous integration and data validation workflows*. CircleCI Docs. <https://circleci.com/docs/>
- VIII. **Codd, E. F.** (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377–387. <https://doi.org/10.1145/362384.362685>
- IX. **Coronel, C., & Morris, S.** (2019). *Sistemas de bases de datos: Diseño, implementación y gestión* (12a ed.). Cengage Learning.
- X. **Date, C. J.** (2004). *Introducción a los sistemas de bases de datos* (8a ed.). Pearson Educación. (Nota: Se unificó con la mención "Date, 2003" de tu lista suelta).
- XI. **Davis, F. D.** (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340. <https://doi.org/10.2307/249008>
- XII. **Elmasri, R., & Navathe, S. B.** (2016). *Fundamentos de sistemas de bases de datos* (7a ed.). Pearson Educación.
- XIII. **Ernesto, G., & Bolia, L.** (2007). *Modelos de gestión de costos para entidades tecnológicas*. Editorial Universitaria.
- XIV. **García López, F. J.** (2017). *Estrategias de desarrollo de sistemas de información basados en el usuario*. Editorial Síntesis.
- XV. **GeeksforGeeks.** (2025). *Data structures and relational database management systems (RDBMS)*. GeeksforGeeks. <https://www.geeksforgeeks.org/>
- XVI. **Gómez Fuentes, H.** (2021). *Automatización de unidades de información: Guía práctica para bibliotecas*. Trillas.
- XVII. **Graham, P.** (2006). *Hackers & painters: Big ideas from the computer age*. O'Reilly Media.
- XVIII. **Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P.** (2018). *Metodología de la investigación* (6a ed.). McGraw-Hill.

- XIX. **IBM.** (2023). *Pruebas de integridad de datos en entornos de desarrollo acelerado*. IBM Knowledge Center.
- XX. **IBM.** (2025a). *Arquitectura de capas y servicios web en aplicaciones de escritorio*. IBM Developer Docs.
- XXI. **IBM.** (2025b). *Principios de la capa lógica o de negocio en sistemas transaccionales*. IBM Systems Journal.
- XXII. **International Federation of Library Associations and Institutions.** (2018). *Directrices para bibliotecas escolares* (2a ed.). IFLA. (Nota: Se unificó con la mención "IFLA, s.f." de tu lista suelta).
- XXIII. **International Organization for Standardization.** (2019). *Information and documentation — Systems for libraries and information centers — Vocabulary* (Norma ISO 5127:2019).
- XXIV. **International Organization for Standardization.** (2020). *Information and documentation — Relational library database structures* (Norma ISO 21127:2020).
- XXV. **Kendall, K. E., & Kendall, J. E.** (2014). *Análisis y diseño de sistemas* (8a ed.). Pearson Educación.
- XXVI. **Koha Community.** (s. f.). *Estándares de sistemas integrados de gestión bibliotecaria y protocolos de interoperabilidad*. <https://koha-community.org/documentation>
- XXVII. **Kontacto.** (s. f.). *Estrategias de comunicación y recopilación de requerimientos de software*. Kontacto Docs.
- XXVIII. **Lancaster, F. W.** (2008). *Evaluación de los servicios de biblioteca*. Universidad de Antioquia.
- XXIX. **Laudon, K. C., & Laudon, J. P.** (2016). *Sistemas de información gerencial* (14a ed.). Pearson Educación.
- XXX. **Lucas, H. C.** (1993). *La gestión del cambio en la implementación de sistemas de información*. McGraw-Hill.
- XXXI. **Lucidea.** (2025). *Inteligencia de datos y tendencias en la gestión bibliotecaria moderna*. Lucidea White Papers.
- XXXII. **Mampel, L.** (2025). Técnicas de investigación para la recolección de requerimientos de usuarios finales. *Journal of Software Engineering & Design*, 14(1), 45–58.
- XXXIII. **Martínez de Sousa, J.** (2014). *Manual de edición y autoedición*. Ediciones Pirámide.
- XXXIV. **McClure, C. R.** (2015). *Library electronic metrics, measures and statistics*. Information Today.
- XXXV. **MDN Web Docs.** (2025). *Web technologies and structured data guidelines*. Mozilla Developer Network. <https://developer.mozilla.org/>
- XXXVI. **Microsoft.** (2023a). *Documentación oficial de Microsoft Access: Formulación de controles y objetos de interfaz*. Microsoft Learn.
- XXXVII. **Microsoft.** (2023b). *Guía de programación en Visual Basic para Aplicaciones (VBA) y manejo de macros*. Microsoft Learn.
- XXXVIII. **Microsoft.** (2024). *Optimización de consultas SQL en el motor de base de datos Access JET/ACE*. Microsoft Developer Network. (Nota: Los registros 36, 37 y 38 cubren y expanden las menciones "Microsoft 2023, 2025" de la lista superior).

- XXXIX. **Monfasani, R. E., & Curzel, M. F.** (2016). *Usuarios de la información: Formación y desafíos*. Alfagrama.
- XL. **Nielsen, J.** (1994). *Ingeniería de usabilidad*. Academic Press.
- XLI. **Norman, D. A.** (2013). *The design of everyday things* (Ed. revisada). Basic Books.
- XLII. **Orera Orera, L.** (2017). *La biblioteca universitaria ante los nuevos retos del entorno digital*. Editorial Síntesis.
- XLIII. **Pérez López, C.** (2018). *Análisis de datos y diseño relacional con Access*. Alfaomega.
- XLIV. **Pressman, R. S., & Maxim, B. R.** (2020a). *Ingeniería de software: Un enfoque práctico* (9a ed., Vol. 1). McGraw-Hill.
- XLV. **Pressman, R. S., & Maxim, B. R.** (2020b). *Diseño arquitectónico y de interfaces en la ingeniería del software moderno* (9a ed., Vol. 2). McGraw-Hill.
- XLVI. **Ranganathan, S. R.** (1931). *The five laws of library science*. Madras Library Association.
- XLVII. **Rowley, J.** (2016). *The electronic library* (4a ed.). Facet Publishing.
- XLVIII. **Sánchez Bañares, J.** (2022). *Desarrollo de aplicaciones de escritorio orientadas a bases de datos*. Marcombo.
- XLIX. **Shneiderman, B., Pleasant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N.** (2016). *Diseño de interfaces de usuario: Estrategias para la interacción humano-computadora efectiva* (6a ed.). Pearson.
- L. **Silberschatz, A., Korth, H. F., & Sudarshan, S.** (2019a). *Fundamentos de bases de datos: Modelado relacional y normalización* (7a ed., Vol. 1). McGraw-Hill.
- LI. **Silberschatz, A., Korth, H. F., & Sudarshan, S.** (2019b). *Fundamentos de bases de datos: Lenguaje de consultas e integridad transaccional* (7a ed., Vol. 2). McGraw-Hill.
- LII. **Sommerville, I.** (2016). *Ingeniería de software* (10a ed.). Pearson Educación.
- LIII. **Sweller, J.** (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. [https://doi.org/10.1207/s15516709cog1202\\_4](https://doi.org/10.1207/s15516709cog1202_4)
- LIV. **Torres Ramírez, I.** (2015). *Las fuentes de información: Metodología del diseño y estudio bibliográfico*. Síntesis.
- LV. **TutorialsPoint.** (2025). *Database management systems and advanced SQL tutorials*. TutorialsPoint Library. <https://www.tutorialspoint.com/>
- LVI. **Tramullas, J.** (2020). Sistemas de gestión bibliotecaria de código abierto frente a soluciones locales en la educación media. *Journal of Information Systems*, 18(3), 112–125.
- LVII. **W3C.** (2024). *Web content accessibility guidelines (WCAG) 2.2*. World Wide Web Consortium. <https://www.w3.org/TR/WCAG22/>
- LVIII. **Whitten, J. L., & Bentley, L. D.** (2017). *Análisis y diseño de sistemas de información* (7a ed.). McGraw-Hill.
- LIX. **Yourdon, E.** (2015). *Análisis estructural clásico y moderno*. Prentice Hall.

## TABLA DE IMAGENES

Tabla 1_PREGUNTA1_ENTREVISTA .....	8
Tabla 2_PREGUNTA2_ENTREVISTA .....	8
Tabla 3_PREGUNTA3_ENTREVISTA .....	9
Tabla 4_PREGUNTA3_ENTRESVISTA .....	9
Tabla 5_PREGUNTA4_ENTREVISTA .....	9
Tabla 6_PREGUNTA5_ENTREVISTA .....	9
Tabla 7_PREGUNTA6_ENTREVISTA .....	9
Tabla 8_DIAGRAMA-DE-FLUJO_ .....	29
Tabla 9_DIAGRAMA-DE-FLUJO .....	44