



Sistema de Gestión de Biblioteca

Mariana Garbanzo Céspedes

Universidad Internacional San Isidro Labrador

(ISB-32) Programación Avanzada

Estefanía Boza Villalobos

Agosto del 2025

Nota del autor

El presente trabajo es requisito para optar por la aprobación del curso (ISB-32) PROGRAMACIÓN AVANZADA de la carrera Ingeniería de Sistemas de la Universidad Internacional San Isidro Labrador. Para contactar al autor escribir al correo garbanzomariana21@gmail.com

Contenido

Índice de Tablas	4
Índice de Ilustraciones	5
Introducción.....	6
Objetivo General	7
Objetivos Específicos.....	7
Justificación.....	8
Alcances del Sistema	9
Requerimientos del Proyecto	9
Tecnologías por Utilizar	10
Tecnologías por Utilizar	10
ASP.NET Web Forms con C# (Backend)	10
SQL Server (Base de Datos)	10
Entity Framework (ORM)	11
HTML5, CSS3 y Bootstrap (Frontend)	11
JavaScript y jQuery (Interactividad)	11
Librerías utilizadas	11
Diseño del sistema.....	13
Descripción de cada vista	13
HTML:	14
CSS:.....	14
Diseño de la Base de Datos y Backend	15
Diagrama entidad-relación del sistema eLibrary	15
Código de conexión a la base de datos explicado	16

Uso de la cadena de conexión en el código C#	17
Explicación del flujo básico para usar la conexión	18
Ventajas de esta implementación	19
Requerimientos del Proyecto	20
Conclusiones Parciales	32
Tareas Pendientes	34
Enlace del funcionamiento del sistema y código	35

índice de Tablas

Tabla 1: Requerimiento 1.....	20
Tabla 2: Requerimiento 2.....	20
Tabla 3: Requerimiento 3.....	22
Tabla 4: Requerimiento 4.....	22
Tabla 5: Requerimiento 5.....	23
Tabla 6: Requerimiento 6.....	23
Tabla 7: Requerimiento 7.....	24
Tabla 8: Requerimiento 8.....	24
Tabla 9: Requerimiento 9.....	25
Tabla 10: Requerimiento 10.....	25
Tabla 11: Requerimiento 11.....	27
Tabla 12: Requerimiento 12.....	27
Tabla 13: Requerimiento 13.....	28
Tabla 14: Requerimiento 14.....	28
Tabla 15: Requerimiento 15.....	29
Tabla 16: Requerimiento 16.....	29
Tabla 17: Requerimiento 17.....	30
Tabla 18: Requerimiento 18.....	30
Tabla 19: Requerimiento 19.....	31
Tabla 20: Requerimiento 20.....	31

índice de Ilustraciones

Ilustración 1: Diagrama Entidad Relación - SQLSERVER.....	16
---	----

Introducción

En la actualidad, las bibliotecas se enfrentan al reto de adaptarse a las demandas tecnológicas y organizativas de un entorno digital en constante cambio, debido a que, muchas instituciones educativas aún utilizan métodos manuales o herramientas informáticas básicas para gestionar sus bibliotecas se provocan ineficiencias, pérdidas de información y dificultades para ofrecer un servicio ágil y moderno a los usuarios, por lo que, esta situación limita el acceso oportuno a los recursos bibliográficos, dificulta el control de préstamos y devoluciones, y representa una carga administrativa para el personal encargado.

Ante esto se hace evidente la necesidad de desarrollar un sistema web especializado que permita automatizar, centralizar y optimizar los procesos esenciales de una biblioteca. El presente proyecto tiene como objetivo la creación de un Sistema de Gestión de Biblioteca, que integra funcionalidades clave como el registro y control de libros, la administración de préstamos y devoluciones, la gestión de usuarios con diferentes niveles de acceso, esto se implementa a través de una plataforma web segura, accesible desde cualquier dispositivo con conexión a internet y diseñada con una interfaz clara e intuitiva.

La solución propuesta no solo busca mejorar la eficiencia operativa de las bibliotecas, sino también elevar la calidad del servicio que se brinda a estudiantes, docentes y demás usuarios. Además, se prioriza la implementación de buenas prácticas de desarrollo, utilizando tecnologías modernas como ASP.NET MVC, SQL Server, Entity Framework y herramientas de diseño responsivo, que garantizan la escalabilidad, mantenibilidad y rendimiento del sistema.

Este proyecto ayuda a representar un avance hacia la transformación digital de las bibliotecas ya sean escolares o en general, contribuyendo a una gestión más ordenada, transparente y eficaz de los recursos informativos, y promoviendo así una experiencia de usuario más satisfactoria y productiva.

Objetivo General

Crear y desarrollar un sistema web para la gestión de bibliotecas que posibilite el registro de libros, la supervisión del préstamo y la devolución de ejemplares, la administración de usuarios y la facilitación del control bibliotecario en instituciones educativas de manera segura, ágil y eficiente.

Objetivos Específicos

Establecer un módulo de autenticación de usuarios que contemple distintos niveles de acceso (administrador y usuario).

Desarrollar características para el registro, búsqueda, modificación y eliminación de libros.

Implementar un proceso automatizado para préstamos y devoluciones, incluyendo validaciones de disponibilidad.

Crear la administración de autores, usuarios, libros, editores y préstamos desde el panel de control del administrador.

Diseñar una interfaz intuitiva y funcional que mejore la experiencia de uso para los encargados de la biblioteca.

Justificación

Las bibliotecas desempeñan un papel fundamental en el proceso de formación personal, profesional y cultural, ya que fomentan la lectura, el desarrollo del pensamiento crítico y el acceso a una amplia variedad de recursos informativos. Sin embargo, muchas instituciones, tanto educativas como comunitarias o especializadas, todavía administran sus bibliotecas de manera manual, utilizando libros de registro físicos o sistemas informáticos básicos no centralizados. Este tipo de gestión tradicional presenta múltiples limitaciones, tales como la pérdida de libros, la dificultad para realizar un seguimiento eficiente de los préstamos, errores humanos en el registro, acumulación de tareas administrativas y, en muchos casos, la falta de reportes analíticos que permitan tomar decisiones estratégicas sobre el uso y administración de los recursos bibliográficos.

El proyecto Sistema de Gestión de Biblioteca se presenta como una solución tecnológica destinada a mejorar la eficiencia, transparencia y organización en la gestión de bibliotecas en general. A través de una plataforma web moderna e intuitiva, se busca automatizar los procesos de registro y control de libros, la administración de préstamos, devoluciones y con la incorporación de funcionalidades como el control de disponibilidad, autenticación de usuarios y categorización de libros por autor, editorial o tema, responde a las necesidades operativas de una biblioteca moderna, permitiendo optimizar el servicio que se brinda a los usuarios.

El sistema está diseñado con distintos niveles de acceso, asegurando que solo los usuarios autorizados puedan realizar tareas administrativas como la edición de registros, el control de inventario o la gestión de detalles de los libros. Esto refuerza la seguridad de la información, facilita la gestión del personal y mejora significativamente la experiencia de los usuarios, quienes podrán acceder a los servicios bibliotecarios de manera ordenada, rápida y desde cualquier dispositivo con acceso a internet.

Alcances del Sistema

- Registro y gestión de libros, que abarca título, autor, editorial, año, idioma, género, cantidad y carga de imagen.
- Registro y autenticación de usuarios a través de formularios seguros y roles diferenciados (administrador y miembro).
- Préstamo y devolución de libros con control automático de inventario y registro de historial.
- Visualización de libros no devueltos mediante una lista de morosos.
- Gestión de autores, editoriales, categorías e idiomas relacionados con los libros.
- Generación de facturas por la venta de libros con cálculo automático de totales e impuestos.
- Exportación de reportes y estadísticas gráficas en formatos PDF y Excel.
- Interfaz adaptable y compatible con dispositivos móviles gracias al uso de Bootstrap 5.
- Validaciones, control de acceso y gestión de credenciales (incluye la opción de cambio de contraseña).
- Canal de contacto para soporte a través de un formulario.

Requerimientos del Proyecto

Funcionales:

- Autenticación de usuarios con roles diferenciados.
- CRUD completo para libros y usuarios.
- Registro y control de préstamos y devoluciones.
- Alertas visuales sobre la disponibilidad y el estado del libro.

No funcionales:

- Interfaz web amigable, accesible y responsiva.
- Base de datos relacional (SQL Server).
- Seguridad en la gestión de contraseñas.
- Eficiencia en consultas y procesamiento de datos.

Tecnologías por Utilizar

Para el desarrollo del sistema web Sistema de Gestión de Biblioteca, se optó por tecnologías modernas, robustas y ampliamente reconocidas en la industria del desarrollo de software. Estas herramientas permiten la creación de un sistema seguro, eficiente, escalable y con una interfaz amigable para el usuario final. A continuación, se detallan:

Tecnologías por Utilizar

Para el desarrollo del sistema web Sistema de Gestión de Biblioteca, se optó por tecnologías modernas, robustas y ampliamente reconocidas en la industria del desarrollo de software. Estas herramientas permiten la creación de un sistema seguro, eficiente, escalable y con una interfaz amigable para el usuario final. A continuación, se detallan:

ASP.NET Web Forms con C# (Backend)

ASP.NET Web Forms es un modelo de desarrollo web que Microsoft ofrece, permitiendo la creación de aplicaciones web dinámicas a través de una estructura basada en formularios. A diferencia de ASP.NET MVC, Web Forms no sigue el patrón Modelo-Vista-Controlador, sino que organiza el código en archivos .aspx (interfaz) y .aspx.cs (lógica del servidor), lo que facilita un desarrollo ágil mediante controles visuales y eventos del lado del servidor.

C# es el lenguaje empleado para programar la lógica del servidor, establecer conexiones a la base de datos y gestionar eventos. Se trata de un lenguaje robusto, seguro y orientado a objetos, perfecto para el desarrollo de aplicaciones empresariales. Su integración con Web Forms permite el uso de herramientas como SqlConnection, SqlCommand y controles como GridView o DropDownList.

SQL Server (Base de Datos)

SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Se utilizará para almacenar toda la información del sistema: libros, usuarios, préstamos, devoluciones, reportes, entre otros. Este motor proporciona

alto rendimiento, seguridad en el manejo de datos, integridad referencial y soporte para consultas complejas.

Entity Framework (ORM)

Entity Framework es un ORM (Object-Relational Mapper) para .NET que permite interactuar con la base de datos utilizando objetos del lenguaje C#, sin necesidad de escribir sentencias SQL manualmente. Esto facilita el desarrollo, mejora la legibilidad del código y reduce la posibilidad de errores.

HTML5, CSS3 y Bootstrap (Frontend)

Estas tecnologías constituyen la base del diseño visual del sistema:

HTML5 define la estructura del contenido de las páginas web.

CSS3 se encarga del estilo visual: colores, tipografías, márgenes, etc.

Bootstrap es un marco de diseño que permite crear interfaces responsivas y modernas, adaptables a dispositivos móviles, tablets y computadoras.

JavaScript y jQuery (Interactividad)

JavaScript es el lenguaje de programación utilizado en los navegadores, que facilita la incorporación de dinamismo e interactividad al sistema.

jQuery es una biblioteca liviana que simplifica la utilización de JavaScript, especialmente en la gestión de eventos, animaciones y solicitudes AJAX sin necesidad de recargar la página.

Librerías utilizadas

En el desarrollo del sistema eLibrary se utilizaron diversas bibliotecas que complementan y mejoran las funcionalidades del sistema, lo que permite una experiencia de usuario más fluida, una presentación visual profesional y una gestión eficiente de la información.

Bootstrap 5 fue la herramienta principal para la creación de una interfaz responsiva, moderna y adaptable a cualquier dispositivo. Gracias a su estructura basada en

rejillas y clases predefinidas para componentes como formularios, tarjetas, botones y menús, se logró un desarrollo rápido y coherente del diseño visual. Se utilizó para asegurar la compatibilidad móvil, mejorar la accesibilidad y conseguir una apariencia profesional en todas las vistas del sistema, incluyendo SignUp.aspx, Reports.aspx y los formularios CRUD.

Font Awesome enriqueció el diseño con una colección de íconos vectoriales escalables que mejoran la navegación visual del usuario. Se integró en Link-botones facilitando la comprensión de las funciones sin necesidad de etiquetas extensas. Su facilidad de uso a través de clases CSS permite una personalización rápida en tamaño, color y estilo.

Para la gestión interactiva de tablas, se implementó DataTables.js, una extensión de jQuery que convierte tablas HTML en componentes dinámicos. Esta biblioteca añadió búsqueda instantánea, ordenamiento por columnas, paginación automática y adaptabilidad visual. Se aplicó en módulos como la gestión de libros, usuarios, facturas y préstamos, permitiendo al usuario consultar y filtrar información de manera eficiente, incluso con grandes volúmenes de datos.

En el módulo de estadísticas y visualización del comportamiento del sistema, se incorporó Chart.js, una biblioteca de JavaScript ligera y flexible para generar gráficos animados y personalizables. Se utilizó principalmente en el Reports.aspx para mostrar, por ejemplo, los libros más leídos y los géneros más solicitados. Esta herramienta permite que el administrador tenga una vista rápida y clara de los indicadores clave mediante gráficos de torta, barras y líneas.

Finalmente, se integró iTextSharp, una biblioteca .NET para la creación y exportación de archivos PDF. Gracias a ella, el sistema permite generar documentos con información administrativa o estadística. Esta funcionalidad aporta valor al sistema al ofrecer salidas impresas profesionales y portables, útiles para archivos físicos, respaldo o presentación institucional.

Cronograma de Actividades		
Semana	Actividad	Resultado
1-4	Propuesta del proyecto	Documentación
2-3	Diseño de la base de datos	Diagrama ER, tablas
4-5	Desarrollo de las vistas	Interfaces
6-7	Desarrollo de la autenticación y roles	Login funcional
8-9	CRUD de usuarios y libros	Gestión funcional
10	Módulo de préstamos y devoluciones	Gestión funcional
11-12	Pruebas finales	Sistema funcional completo
13	Documento final y exposición	Entrega del proyecto completo

Diseño del sistema

El sistema consiste en una aplicación web creada con tecnologías frontend (HTML5 y CSS3), organizada con múltiples vistas que reflejan las funciones fundamentales de un blog personal. Su diseño tiene como objetivo proporcionar una navegación clara y accesible, con un estilo visual atractivo y moderno.

Las principales funcionalidades incluyen:

- Visualización de publicaciones recientes.
- Acceso a un panel administrativo con un listado de entradas.
- Creación de nuevas entradas a través de un formulario.
- Formulario de contacto simulado.

Este sistema está destinado a personas interesadas en publicar artículos personales, experiencias, conocimientos o reflexiones sin requerir conocimientos técnicos complejos. Está diseñado para ser ligero, responsivo y adaptable a dispositivos móviles.

Descripción de cada vista

1. **index.html – Página de Inicio:** Esta página incluye el encabezado principal, un menú de navegación que enlaza a otras secciones, una imagen

introductoria y un área dedicada a publicaciones destacadas. Se emplea un diseño en tarjetas para mostrar las entradas más recientes. Todo el diseño es responsivo y se ajusta a pantallas pequeñas.

2. **login.html – Formulario de Acceso:** Esta vista presenta un formulario visualmente centrado, donde el administrador puede introducir su nombre de usuario y contraseña. Los campos cuentan con validación HTML y efectos de enfoque. Incluye un botón estilizado y un acceso rápido simulado al panel de control.
3. **dashboard.html – Panel del Sistema:** Muestra un mensaje de bienvenida, un botón para crear nuevas entradas y una tabla con publicaciones existentes. La tabla es completamente responsiva y en pantallas pequeñas se transforma en una estructura vertical.
4. **registro.html – Crear Nueva Entrada:** Formulario destinado a ingresar una nueva publicación. Se requiere el título y el contenido en un campo de texto expandido. Los campos están diseñados con bordes suaves y un botón que presenta una animación al pasar el cursor.:
5. **contacto.html – Formulario de Contacto:** Formulario sencillo que permite al usuario simular el envío de un mensaje al administrador. Aunque carece de un backend funcional, simula una redirección al panel de control. La imagen superior se escala de manera proporcional.

Explicación sobre el uso de HTML y CSS

HTML:

Se utilizó HTML5 semántico para proporcionar una estructura lógica al contenido, empleando etiquetas como <header>, <main>, <section>, <article>, entre otras. Se usaron formularios con las etiquetas adecuadas (<label>, <input>, <textarea>) y se aplicaron atributos como required y type.

CSS:

La personalización visual se logró mediante la aplicación de:

- Selectores por etiqueta, clase e ID

- Modelo de caja: padding, margin, border-radius, box-shadow
- Diseño responsivo con @media queries
- Efectos visuales utilizando transition, hover y object-fit

Diseño de la Base de Datos y Backend

Diagrama entidad-relación del sistema eLibrary

El diagrama entidad-relación que se presenta a continuación ilustra la estructura lógica de la base de datos elibraryDB, la cual respalda todas las funcionalidades que se han implementado en el sistema eLibrary. Este modelo está compuesto por tablas interconectadas a través de claves primarias y relaciones lógicas implícitas, lo que facilita la gestión eficiente de libros, usuarios, préstamos, editoriales y autores.

El diseño abarca las siguientes tablas principales:

- **admin_login_tbl:** se encarga de almacenar las credenciales y datos de los administradores.
- **author_master_tbl:** incluye el catálogo de autores.
- **publisher_master_tbl:** conserva las editoriales registradas.
- **book_master_tbl:** administra el inventario de libros junto con sus datos principales, que abarcan autor, editorial, género, idioma, edición, stock e imagen.
- **member_master_tbl:** contiene la información personal, credenciales y estado de los miembros del sistema.
- **book_issue_tbl:** documenta los préstamos de libros, incluyendo fechas de emisión, devolución y estado.

Las claves primarias están correctamente establecidas en todas las tablas, tales como **book_id**, **member_id**, **author_id**, etc. En el caso de **book_issue_tbl**, se establece una clave primaria compuesta (**member_id**, **book_id**) para garantizar que un usuario no registre múltiples préstamos del mismo libro al mismo tiempo.

Aunque las claves foráneas no están declaradas de manera explícita en el script SQL, el sistema mantiene la integridad referencial mediante validaciones en la capa de aplicación. Por ejemplo, al realizar un préstamo, se comprueba que **book_id** y **member_id** existan en sus respectivas tablas.

El tipo de relaciones es principalmente uno a muchos, como, por ejemplo:

- Un autor puede tener muchos libros → **author_master_tbl** → **book_master_tbl**.
- Una editorial publica muchos libros → **publisher_master_tbl** → **book_master_tbl**.
- Un miembro puede emitir múltiples préstamos → **member_master_tbl** → **book_issue_tbl**. Este diseño permite una gestión eficiente de los datos, evita redundancias y asegura la escalabilidad del sistema para futuras funcionalidades.



Ilustración 1: Diagrama Entidad Relación - SQLSERVER

Código de conexión a la base de datos explicado

La conexión a la base de datos es un paso esencial para cualquier aplicación que requiera interactuar con datos almacenados. En el contexto de una aplicación

ASP.NET utilizando **C#**, esta conexión se establece generalmente mediante una cadena de conexión que se define en el archivo de configuración **Web.config** y luego se emplea en el código para abrir un canal de comunicación con la base de datos. En el archivo **Web.config**, dentro del nodo **<connectionStrings>**, se especifica una cadena de conexión que contiene toda la información necesaria para que la aplicación pueda localizar y autenticar la base de datos. Por ejemplo:

```
<connectionStrings>

  <add name="con"
        connectionString="Data Source=localhost;Initial Catalog=elibraryDB;Integrated
Security=True"
        providerName="System.Data.SqlClient"/>

</connectionStrings>
```

- **name="con"**: Este es el identificador único que se utilizará para referirse a esta cadena en el código. Puede ser cualquier nombre, pero es fundamental que sea claro y coherente.
- **Data Source=localhost**: Señala el servidor donde se aloja la base de datos. En este caso, localhost indica que el servidor está en la misma máquina que ejecuta la aplicación.
- **Initial Catalog=elibraryDB**: Define el nombre de la base de datos a la que se conectará, en este caso **elibraryDB**.
- **Integrated Security=True**: Indica que se empleará la autenticación integrada de Windows para conectarse a la base de datos, evitando así la necesidad de proporcionar un usuario y una contraseña de forma explícita.
- **providerName="System.Data.SqlClient"**: Establece el proveedor de datos que se utilizará para crear la conexión, que en este caso es el proveedor para SQL Server.

Uso de la cadena de conexión en el código C#

En cualquier parte del código donde se requiera conectar con la base de datos (por ejemplo, en páginas ASP.NET o clases de acceso a datos), se puede acceder a la

cadena de conexión mediante la clase ConfigurationManager que lee el archivo Web.config. Luego, se crea un objeto SqlConnection para manejar la conexión:

```
using System.Configuration;
using System.Data.SqlClient;

// Crear la conexión a partir de la cadena almacenada en Web.config
SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["con"].ConnectionString);
// Ejemplo de integración usado en eLibrary
string strcon = ConfigurationManager.ConnectionStrings["con"].ConnectionString;
```

- **ConfigurationManager.ConnectionStrings["con"]** busca en el archivo de configuración la cadena con el nombre "con".
- **.ConnectionString** accede a la cadena que contiene los parámetros de conexión.
- **new SqlConnection(...)** crea una nueva instancia que representa la conexión con la base de datos.

Explicación del flujo básico para usar la conexión

- **Crear el objeto SqlConnection:** Se instancia usando la cadena de conexión leída del Web.config.
- **Abrir la conexión:** Con con.Open() se establece la conexión física con el servidor de base de datos.
- **Ejecutar comandos o consultas:** A través de objetos como SqlCommand se pueden ejecutar instrucciones SQL.
- **Cerrar la conexión:** Es importante cerrar la conexión con con.Close() para liberar recursos.

Ventajas de esta implementación

Separación de configuración y código: La cadena de conexión no está **"hardcodeada"** en el código, lo que facilita cambiar la base de datos sin recompilar la aplicación.

Seguridad: Al usar `Integrated Security=True` se evita poner usuario y contraseña en el archivo, aumentando la seguridad.

Reusabilidad: Cualquier página o clase puede reutilizar la cadena de conexión sin repetir datos.

Facilidad de mantenimiento: Cambiar el servidor o base de datos solo requiere modificar el archivo de configuración.

Requerimientos del Proyecto

Tabla 1: Requerimiento 1

Requerimiento	REQ-F001	Estado	Aprobado
Nombre del Módulo	Registro de miembros (SignUp.aspx)		
Descripción			
Habilitar un formulario para que los nuevos usuarios creen su cuenta en la biblioteca.			
Especificaciones			
Ingreso de datos: nombre completo, cédula, correo electrónico, teléfono, etc. Validaciones de campos obligatorios y formato de correo. Verificación para evitar registros duplicados. Mensajes de confirmación y redirección al login.			
Importancia	Primario	Dependencia	n/a
Justificación			
Facilita el acceso de nuevos miembros al sistema sin intervención del administrador, agilizando el uso de la plataforma.			

Tabla 2: Requerimiento 2

Requerimiento	REQ-F002	Estado	Aprobado
Nombre del Módulo	Inicio de Sesión de Usuario (login.aspx)		
Descripción			
Permite a los usuarios registrados acceder al sistema usando su usuario y contraseña.			
Especificaciones			
Formulario con campos de identificación y contraseña. Verificación de credenciales en la tabla member_master_tbl. Almacenamiento de información en Session. Redirección al perfil del usuario (userProfile.aspx) tras login exitoso. Mensaje de error si las credenciales son incorrectas.			
Importancia	Primario	Dependencia	REQ-F001

Justificación
Permite a los usuarios autenticarse para consultar sus préstamos, actualizar datos y acceder a sus funcionalidades personalizadas.

Tabla 3: Requerimiento 3

Requerimiento	REQ-F003	Estado	Aprobado
Nombre del Módulo	Inicio de Sesión de Administrador (adminLogin)		
Descripción			
Permite a los administradores autenticarse mediante usuario y contraseña para acceder al sistema.			
Especificaciones			
Formulario con campos de nombre de usuario y contraseña. Validación contra la tabla admin_login_tbl. Almacenamiento de datos en Session (rol, usuario, nombre completo). Redirección a homepage.aspx si el login es correcto poniendo visibles las funcionalidades del administrador.			
Importancia	Primario	Dependencia	REQ-F002
Justificación			
Es necesario para restringir el acceso al sistema y permitir la navegación por los módulos solo a administradores autorizados.			

Tabla 4: Requerimiento 4

Requerimiento	REQ-F004	Estado	Aprobado
Nombre del Módulo	Validaciones y Seguridad		
Descripción			
Implementa validaciones en formularios y restricciones por rol.			
Especificaciones			
Validaciones de campos requeridos. Restricciones de acceso mediante variables de sesión. Mensajes de error personalizados con JavaScript y C#.			
Importancia	Primario	Dependencia	REQ-F001, REQ-F002, REQ-F003.
Justificación			
Evita errores, asegura integridad de datos y protege módulos sensibles.			

Tabla 5: Requerimiento 5

Requerimiento	REQ-F005	Estado	Aprobado
Nombre del Módulo	Búsqueda de libros (viewBooks.aspx)		
Descripción			
Permitir a los usuarios realizar búsquedas avanzadas de libros utilizando distintos filtros combinables.			
Especificaciones			
Filtrar por título, autor, editorial, idioma, género, disponibilidad y año de publicación. Mostrar los resultados dinámicamente en una tabla o lista. Botón de “Buscar” para restablecer la búsqueda.			
Importancia	Primario	Dependencia	REQ-F002
Justificación			
Optimiza la experiencia de navegación del usuario y facilita el acceso rápido a la información deseada, sobre todo cuando el catálogo es extenso.			

Tabla 6: Requerimiento 6

Requerimiento	REQ-F006	Estado	Aprobado
Nombre del Módulo	Perfil de Usuario (userProfile.aspx)		
Descripción			
Muestra al usuario sus datos personales y sus libros prestados.			
Especificaciones			
Consulta de información desde la base. Visualización de datos personales y préstamos activos. Estructura separada en columnas para datos y libros.			
Importancia	Primario	Dependencia	REQ-F002
Justificación			

Mejora la experiencia del usuario al darle acceso a su información y estado de préstamos.

Tabla 7: Requerimiento 7

Requerimiento	REQ-F007	Estado	Aprobado
Nombre del Módulo	Perfil de Usuario (userProfile.aspx)		
Descripción			
Permitir al usuario cambiar su contraseña desde su perfil.			
Especificaciones			
Validación de contraseña actual antes de permitir el cambio. Validación de longitud y complejidad de la nueva contraseña. Confirmación de nueva contraseña antes de guardar. Mensaje de éxito o error según resultado.			
Importancia	Primario	Dependencia	REQ-F006
Justificación			
Refuerza la seguridad del sistema y da autonomía al usuario para gestionar sus credenciales.			

Tabla 8: Requerimiento 8

Requerimiento	REQ-F008	Estado	Aprobado
Nombre del Módulo	Gestión de Autores (adminAuthorManagement)		
Descripción			
Permite registrar, actualizar, eliminar y consultar información de autores.			
Especificaciones			
Formulario con campos author_id y author_name. Operaciones CRUD contra la tabla author_master_tbl. Validación de existencia de ID antes de operaciones. Grid con DataTable para visualizar los registros.			

Importancia	Primario	Dependencia	REQ-F003
Justificación			
Se requiere para mantener un catálogo actualizado de autores que alimenta otros módulos como la gestión de libros.			

Tabla 9: Requerimiento 9

Requerimiento	REQ-F009	Estado	Aprobado
Nombre del Módulo	Gestión de Editoriales (adminPublisherManagement)		
Descripción			
Permite registrar, actualizar, eliminar y consultar editoriales.			
Especificaciones			
Manejo de publisher_id y publisher_name. Conexión a la tabla publisher_master_tbl. GridView con DataSource y operaciones CRUD.			
Importancia	Primario	Dependencia	REQ-F003
Justificación			
Es esencial para mantener actualizado el catálogo de editoriales que serán asociadas a los libros registrados.			

Tabla 10: Requerimiento 10

Requerimiento	REQ-F0010	Estado	Aprobado
Nombre del Módulo	Inventario de Libros (bookInventory.aspx)		
Descripción			
Permite gestionar el inventario de libros, incluyendo detalles, stock, autores, editoriales, géneros y descripción.			
Especificaciones			

Formulario completo para ingresar toda la información de un libro. Carga de imagen. Gestión de géneros múltiples (ListBox), idiomas, autores, editores. Control del stock actual, disponible y prestado. CRUD contra book_master_tbl.			
Importancia	Primario	Dependencia	REQ-F008, REQ-F009
Justificación			
Base esencial del sistema: gestiona los libros disponibles para préstamo o compra.			

Tabla 11: Requerimiento 11

Requerimiento	REQ-F0011	Estado	Aprobado
Nombre del Módulo	Gestión de Categorías		
Descripción			
Permite administrar categorías o géneros de libros.			
Especificaciones			
Formulario para agregar, editar y eliminar categorías. Integración con formulario de libros (ListBox múltiple).			
Importancia	Primario	Dependencia	REQ-F0010
Justificación			
Las categorías permiten clasificar los libros para facilitar su búsqueda.			

Tabla 12: Requerimiento 12

Requerimiento	REQ-F0012	Estado	Aprobado
Nombre del Módulo	Gestión de Idiomas		
Descripción			
Permite seleccionar el idioma de los libros registrados.			
Especificaciones			
Uso de DropDownList con idiomas predefinidos. Se guarda el valor en la tabla book_master_tbl.			
Importancia	Secundario	Dependencia	REQ-F0010
Justificación			
Facilita la organización de los libros para usuarios con preferencias lingüísticas específicas.			

Tabla 13: Requerimiento 13

Requerimiento	REQ-F0013	Estado	Aprobado
Nombre del Módulo	Gestión de Miembros (Usuarios)		
Descripción			
Permite editar el estado de cuenta, eliminar y consultar usuarios del sistema.			
Especificaciones			
Formulario con datos personales: nombre, email, contacto. Consulta contra la tabla member_master_tbl. GridView para visualizar los datos.			
Importancia	Primario	Dependencia	REQ-F003
Justificación			
Los usuarios son los clientes de la biblioteca, su gestión es esencial para las operaciones de préstamos.			

Tabla 14: Requerimiento 14

Requerimiento	REQ-F0014	Estado	Aprobado
Nombre del Módulo	Emisión de Libros (bookIssued.aspx)		
Descripción			
Permite registrar el préstamo de libros a usuarios registrados.			
Especificaciones			
Formulario con ID de usuario y libro. Consulta de datos antes de registrar el préstamo. Actualización automática del stock disponible. Registro en la tabla book_issue_tbl.			
Importancia	Primario	Dependencia	REQ-F0010, REQ-F0013
Justificación			
Fundamental para el funcionamiento del sistema como biblioteca, controlando quién tiene qué libro.			

Tabla 15: Requerimiento 15

Requerimiento	REQ-F0015	Estado	Aprobado
Nombre del Módulo	Devolución de Libros		
Descripción			
Permite a los administradores registrar la devolución de libros prestados.			
Especificaciones			
Formulario para buscar y devolver libros. Actualiza stock disponible. Marca el registro como devuelto en la base de datos.			
Importancia	Primario	Dependencia	REQ-F0014
Justificación			
Mantiene el control del inventario y asegura la disponibilidad de libros devueltos.			

Tabla 16: Requerimiento 16

Requerimiento	REQ-F0016	Estado	Aprobado
Nombre del Módulo	Control de morosos		
Descripción			
Mostrar en la lista de distinto color los usuarios que no han devuelto libros a tiempo.			
Especificaciones			
Mostrar nombre del miembro, ID del libro, título, fecha de emisión, fecha límite y días de atraso. Utilizar estilos visuales (color de tabla) para identificar casos urgentes.			
Importancia	Primario	Dependencia	REQ-F0014
Justificación			
Permite a los administradores controlar y reducir el incumplimiento de devoluciones, manteniendo el orden en el inventario y reforzando la disciplina del préstamo.			

Tabla 17: Requerimiento 17

Requerimiento	REQ-F0017	Estado	Aprobado
Nombre del Módulo	Facturación / Compra de Libros (Bills.aspx)		
Descripción			
Permite registrar compras de libros con usuarios y clientes, calculando el total con IVA y generando facturas.			
Especificaciones			
Formulario para seleccionar libro, cantidad, usuario, cliente y método de pago. GridView dinámico con los libros agregados a la factura. Cálculo automático de subtotal, IVA y total. Inserta en book_invoice_tbl y book_invoice_detail_tbl. Descuenta stock automáticamente.			
Importancia	Primario	Dependencia	REQ-F003, REQ-F0010
Justificación			
Módulo clave para realizar ventas y mantener control del inventario y facturación del sistema.			

Tabla 18: Requerimiento 18

Requerimiento	REQ-F0018	Estado	Aprobado
Nombre del Módulo	Dashboard de Estadísticas		
Descripción			
Muestra gráficos estadísticos sobre libros más leídos, géneros populares, etc.			
Especificaciones			
Uso de Chart.js para representar los datos gráficamente. Consulta de datos desde la base y agrupamiento por categorías. Visualización clara para toma de decisiones.			
Importancia	Secundario	Dependencia	REQ-F0010, REQ-F0014
Justificación			
Brinda una visión rápida del comportamiento de lectura y uso del sistema.			

Tabla 19: Requerimiento 19

Requerimiento	REQ-F0019	Estado	Aprobado
Nombre del Módulo	Generación de Reportes (PDF)		
Descripción			
Permite exportar información clave del sistema en formato PDF.			
Especificaciones			
Uso de la librería iTextSharp. Exportación de gráficos. Exportación desde GridView o consultas directas. Botones con funcionalidades específicas en los formularios.			
Importancia	Secundario	Dependencia	REQ-F0018
Justificación			
Facilita el análisis externo y la presentación de resultados administrativos.			

Tabla 20: Requerimiento 20

Requerimiento	REQ-F0020	Estado	Aprobado
Nombre del Módulo	Formulario de Contacto / Soporte (contacto.aspx)		
Descripción			
Permite a los usuarios enviar consultas o mensajes al administrador del sistema para recibir soporte o sugerencias.			
Especificaciones			
Formulario con nombre, email, asunto y mensaje. Validación de campos obligatorios. Envío de datos a una tabla o por correo (según implementación). Confirmación visual de mensaje enviado.			
Importancia	Secundario	Dependencia	n/a
Justificación			
Facilita la comunicación entre usuarios y administradores. Mejora la experiencia del usuario al ofrecer un canal de soporte.			

Requerimiento	REQ-F0021	Estado	Aprobado
Nombre del Módulo	Diseño Responsive y Accesible		
Descripción			
Asegura que el sistema se adapte correctamente a distintos dispositivos y sea visualmente accesible.			
Especificaciones			
Uso de Bootstrap 5 para grillas, tarjetas, formularios. Iconografía con Font Awesome. Compatibilidad móvil garantizada en todos los formularios.			
Importancia	Secundario	Dependencia	n/a
Justificación			
Mejora la experiencia del usuario en diferentes entornos y dispositivos.			

Conclusiones Parciales

El desarrollo del Sistema de Gestión de Biblioteca ha finalizado con éxito la implementación total de los 21 requerimientos establecidos inicialmente. El sistema incluye módulos sólidos y funcionales que satisfacen todas las necesidades de gestión administrativa y operativa. Se destacan la gestión integral de usuarios, que abarca autenticación y roles diferenciados; el manejo detallado de libros con registros completos y búsquedas avanzadas; y el módulo de facturación que realiza cálculos automáticos y actualiza el inventario en tiempo real.

Además, el sistema gestiona de manera eficiente los procesos de préstamo y devolución, controlando el stock y permitiendo el seguimiento de usuarios con morosidad. La generación de reportes en formato PDF, enriquecidos con gráficos estadísticos, proporciona información clave para la toma de decisiones. La interfaz, diseñada con tecnologías modernas, asegura una experiencia intuitiva y responsiva en múltiples dispositivos. La seguridad y las validaciones implementadas protegen la integridad de la información y limitan el acceso según los roles de usuario.

Estos avances posicionan al sistema como una solución completa y efectiva, contribuyendo a la modernización y digitalización de la gestión bibliotecaria, mejorando tanto los procesos administrativos como la satisfacción de estudiantes, docentes y personal.

Tareas Pendientes

- A pesar de los significativos avances, todavía hay algunas actividades clave que deben completarse para llevar a cabo el proyecto con éxito:
- Finalizar la documentación.
- Realizar pruebas de integración exhaustivas para garantizar que todos los módulos interactúan correctamente en condiciones reales de uso.
- Preparar la presentación final del proyecto, que debe incluir una demostración funcional y una explicación detallada de los beneficios y alcances del sistema.
- Con la finalización de estas tareas, se espera entregar un sistema completamente funcional cumpliendo así con los objetivos académicos y prácticos del proyecto.

Enlace del funcionamiento del sistema y código

- Explicación de la base de datos:
<https://drive.google.com/file/d/1HOliXOqWMoM5kvD2yEV9vLBGy5hc1L3/view?usp=sharing>
- Vídeo del sistema Final:
[Video Sistema Funcional - Hecho con Clipchamp 1754538307105.mp4 - Google Drive](#)
- Enlace al código del programa:
[PF SistemaGestionBiblioteca - Google Drive](#)