



GESTIÓN DE BIBLIOTECA

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Web

Instituto de Fomento Montecastelo

Saúl Jorge López

saul.jorge.lopez@gmail.com

ÍNDICE

1.	Introducción	2
1.1.	Descripción del proyecto	2
1.2.	Objetivos	2
1.2.1.	Objetivos generales	2
1.2.2.	Objetivos específicos	2
1.3.	Módulos	3
2.	Desarrollo	4
2.1.	Análisis	4
2.2.	Diseño	5
2.2.1.	Especificación del funcionamiento del sistema	5
2.2.2.	Estructuras de datos	6
2.2.3.	Módulos del programa	6
2.3.	Implementación	7
2.3.1.	Modelos	7
2.3.2.	Vistas	8
2.3.3.	Controladores	8
2.3.4.	Autenticación y seguridad	9
2.3.5.	Front-End	9
2.3.6.	Despliegue	9
2.4.	Pruebas	9
2.4.1.	Verificación del código	9
2.4.2.	Validación del código	10
3.	Conclusiones	10
4.	Líneas futuras	11
5.	Bibliografía	13

1. INTRODUCCIÓN

1.1. DESCRIPCION DEL PROYECTO

Este Proyecto de Fin de Ciclo consiste en una aplicación web que simula el funcionamiento de una biblioteca. Será una aplicación web integral destinada a la gestión y manejo de los servicios básicos de una biblioteca. La solución permitirá interactuar a los usuarios con una interfaz clara y amigable para realizar actividades tales como buscar libros y ver si están disponibles para ser prestados o si ya se han prestado a algún otro usuario, gestionar sus préstamos, acceder a información detallada de cada publicación, crear, editar o eliminar publicaciones y acceder a una vista general de su propio perfil de usuario donde puedan tener información de sus últimas acciones realizadas en la aplicación y una lista de sus libros prestados.

El sistema también brindará funcionalidades administrativas para el registro y mantenimiento de usuarios, funcionalidades reservadas a los usuarios que sean administradores exclusivamente.

1.2. OBJETIVOS

Los objetivos que se pretenden alcanzar con el desarrollo de este aplicativo se pueden clasificar según su especificidad en dos grupos.

1.2.1. OBJETIVOS GENERALES

- Crear una plataforma web interactiva que facilite la gestión eficiente de una biblioteca, ofreciendo una buena experiencia a los usuarios y administradores.
- Implementar un sistema robusto que permita la administración y gestión de libros y usuarios asegurando la integridad y seguridad de los datos.
- Integrar un sistema de seguimiento de actividades que proporcione una visión clara del movimiento de los libros y las acciones de los usuarios.

1.2.2. OBJETIVOS ESPECÍFICOS

- Desarrollar un módulo de autenticación de usuarios que diferencie entre permisos de administrador y permisos de usuario estándar.
- Implementar un sistema de CRUD (Create, Read, Update y Delete) para el manejo de libros y usuarios en la base de datos.
- Incorporar un sistema de préstamo que gestione las fechas de préstamo y devolución de los libros, así como la disponibilidad en tiempo real de los ejemplares.
- Crear un registro de actividades (logs) que almacene y muestre las acciones realizadas por los usuarios, proporcionando un historial de eventos para auditoría y seguimiento.

- Asegurar que la interfaz de la aplicación sea responsiva y accesible desde distintos dispositivos y tamaños de pantalla.
- Ofrecer una opción de personalización de la interfaz mediante un “modo oscuro” mejorando así la accesibilidad y comodidad visual para los usuarios.

1.3. MÓDULOS

Para la realización de este proyecto he desempeñado habilidades adquiridas a lo largo de todo el ciclo de Desarrollo de Aplicaciones Web, en concreto las siguientes divididas por módulos de aprendizaje:

- Para el desarrollo de las vistas e interfaces con las que el usuario interactúa en mi aplicación, me he servido de los conocimientos adquiridos en la materia **Lenguaje de Marcas**, ya que he empleado código HTML inserto en el lenguaje Razor propio de Visual Studio.
- Para el diseño de las mencionadas vistas me ha servido todo lo impartido en la asignatura de **Diseño de Interfaces Web**.
- Para el diseño de la base de datos que contiene las tablas con las inserciones de datos pertinentes para el funcionamiento de la Biblioteca, me he nutrido de lo aprendido en la materia **Bases de Datos**, en la que aprendí SQL en profundidad.
- Para el esquema general de mi aplicación a nivel del funcionamiento lógico, las relaciones entre tablas, la conexión entre las distintas páginas de mi web, etc., he usado los conocimientos adquiridos en la asignatura de **Entornos de Desarrollo** (UML, manejo de entornos de desarrollo, control de versiones, patrones de diseño...).
- Como es evidente, la materia de **Programación** del primer curso del ciclo es transversal y está presente en la capacidad de desarrollar el código en un lenguaje en este caso C# que de hecho es similar a Java, lenguaje aprendido el primer año y que sirve de base para todo desarrollo de software.
- Las asignaturas de **Desarrollo Web en Entorno Cliente y en Entorno Servidor** han sido claves para la comprensión, por un lado, de la programación de la interactividad y dinamismo de una web (entorno cliente con JavaScript) y, por otro, del desarrollo de un backend robusto, eficiente y seguro (entorno servidor con C# y comprensión del Modelo-Vista-Controlador).
- La materia de **Despliegue de Aplicaciones Web** ha sido de ayuda para comprender cómo se lanzan las aplicaciones en los diversos entornos y para tener una visión clara del funcionamiento conjunto de todas las partes y aplicativos que eran necesarios para que mi Biblioteca funcione en cualquier entorno (en definitiva, para poder migrar mi aplicación desde mi PC en local y que pueda funcionar con los mismos datos y funcionalidades en otros ordenadores y dispositivos).

2. DESARROLLO

En esta sección hablaré en profundidad del desarrollo en sí de mi aplicación, desde su propuesta inicial hasta las pruebas realizadas al final, pasando por el diseño de las entidades, vistas y funcionalidades en base a los requisitos iniciales tenidos en consideración.

2.1. ANÁLISIS

El cliente en cuestión gestiona una biblioteca y necesita un sistema que le permita digitalizar y simplificar la gestión de sus operaciones diarias. El cliente necesita una aplicación que facilite el seguimiento y la gestión de los libros, los préstamos y las devoluciones, así como el mantenimiento de registros de usuarios y sus actividades dentro de la biblioteca. La aplicación debe ser intuitiva y fácil de usar para garantizar una rápida adaptación por parte del personal que gestionará los libros y usuarios.

Lo requisitos del software son los siguientes:

- **Gestión de libros.**
 - Capacidad para agregar, editar y eliminar libros en la base de datos.
 - Registro de detalles del libro como título, autor y año de publicación.
 - Estado del libro (disponible o prestado).
- **Gestión de préstamos.**
 - Funcionalidad para prestar libros a los usuarios.
 - Registro de la fecha del préstamo.
 - Devolución de libros prestados y actualizaciones de su estado.
- **Gestión de usuarios.**
 - Creación y manejo de cuentas de usuario con información de autenticación.
 - Asignación de roles para diferenciar entre administradores y usuarios comunes.
- **Registro de actividades.**
 - Captura de todas las operaciones realizadas, como creación, modificación y eliminación de registros.
 - Visualización de actividades recientes en la cuenta de usuario.
- **Interfaz de usuario.**
 - Diseño amigable y accesible para una navegación sencilla.
 - Visualización clara de las listas de libros y usuarios.
 - Facilidades para la búsqueda y filtrado de libros.

- **Seguridad.**

- Protección de la información de los usuarios y de la base de datos de la biblioteca.
- Sistema de login para el acceso a funcionalidades administrativas.

2.2. DISEÑO

La aplicación de Biblioteca se ha diseñado para ser robusta, segura y fácil de usar, asegurando que cumple con las necesidades del cliente y proporciona una experiencia de usuario satisfactoria.

2.2.1. ESPECIFICACIÓN DEL FUNCIONAMIENTO DEL SISTEMA

La aplicación Biblioteca está diseñada para ofrecer una solución integral a la gestión de una biblioteca, brindando a los administradores y a los usuarios una plataforma para la administración y uso de recursos bibliográficos. El núcleo del sistema reside en su capacidad para gestionar libros, usuarios, préstamos y registros de actividad de manera eficiente y segura.

Cuando un administrador o un usuario con los permisos necesarios accede al sistema, se encuentra con su página de cuenta de usuario y en una barra de navegación superior con un panel de control intuitivo desde el cual puede realizar una variedad de tareas. Puede añadir nuevos libros al inventario de la biblioteca, detallando información esencial como el título, el autor y el año de publicación, y también puede marcar si el libro está actualmente prestado o no. Esta última es una funcionalidad crucial para mantener un registro actualizado del catálogo de la biblioteca.

Además de gestionar los libros, el sistema permite a los usuarios registrados pedir prestados los libros que estén disponibles. Cada préstamo registra la fecha y se asocia con el usuario correspondiente, lo cual facilita el seguimiento y la gestión de los préstamos. Los usuarios pueden ver estos registros y, cuando un libro es devuelto automáticamente tras el periodo de tiempo pertinente, se actualiza su estado a disponible de nuevo, lo que permite que otros usuarios lo pidan prestado.

El manejo de los usuarios es otro aspecto crítico de la aplicación. La plataforma permite la creación de nuevos perfiles de usuario, incluyendo información relevante y credenciales de acceso. Los administradores pueden asignar roles a cada usuario, diferenciando entre administradores y usuarios estándar, lo que permite un control sobre quién puede realizar ciertas acciones dentro del sistema.

El sistema también mantiene un registro detallado de las actividades, lo cual asegura que cualquier acción realizada, ya sea la creación, edición o eliminación de un registro de libros o usuarios, así como la solicitud de

préstamos, quede documentada. Esto no sólo ayuda a mantener la integridad de la base de datos, sino que también proporciona una pista de auditoría que puede ser valiosa para la resolución de problemas o la detección de un uso indebido del sistema.

La experiencia de usuario en la aplicación es fluida y sin interrupciones, con una interfaz limpia y clara que presenta la información de manera coherente y accesible. Los usuarios pueden buscar libros utilizando la función de búsqueda integrada, que permite filtrar resultados por título o nombre del autor. Esta capacidad de búsqueda y filtrado permite encontrar un libro específico de forma rápida y con menos esfuerzo que buscarlo manualmente. Se incluye también una opción de cambio entre modo oscuro y modo claro para hacer la experiencia del usuario más agradable a nivel visual y le otorga mayor capacidad de personalización de las interfaces con las que interactúa.

La aplicación se construye sobre una arquitectura segura y robusta, utilizando buenas prácticas y el patrón de diseño Modelo-Vista-Controlador para garantizar que los datos de los usuarios y la integridad del sistema no se vean comprometidos. En resumen, la aplicación Biblioteca está meticulosamente diseñada para ser una herramienta confiable en la gestión de una biblioteca a través de una aplicación web.

2.2.2. ESTRUCTURAS DE DATOS

El diseño de la base de datos sigue el modelo relacional, con tablas para libros, préstamos, logs y usuarios. Las relaciones entre estas tablas permiten la integridad referencial y la eficiencia en las consultas. Se emplea Entity Framework para mapear las tablas a clases dentro de la aplicación, facilitando así la gestión de datos.

2.2.3. MÓDULOS DEL PROGRAMA

El sistema se divide en varios módulos principales que interactúan entre sí para cumplir con los requisitos establecidos en el análisis:

- A. Módulo de autenticación.
 - i. Gestiona el acceso de usuarios, asegurando que solamente los usuarios autorizados puedan acceder a ciertas funcionalidades.
 - ii. Utiliza sesiones para mantener el estado de la autenticación.
- B. Módulo de gestión de libros.
 - i. Interfaz para la entrada de datos de nuevos libros y modificación de los existentes.
 - ii. Implementa un sistema CRUD (crear, leer, actualizar y eliminar) para la administración de la base de datos de libros.

- C. Módulo de gestión de préstamos.
 - i. Permite a los usuarios (o administradores) realizar préstamos de libros.
 - ii. Registra la fecha del préstamo y gestiona automáticamente la devolución transcurridos ciertos días desde el préstamo, actualizando el estado del libro correspondiente.
- D. Módulo de gestión de usuarios.
 - i. Permite la creación de nuevos usuarios y la asignación de roles.
 - ii. Ofrece la posibilidad de editar y eliminar cuentas de usuario.
- E. Módulo de registros de actividades.
 - i. Captura y almacena en una tabla de logs todas las actividades realizadas por los usuarios.
 - ii. Muestra un resumen de la actividad reciente en la interfaz del usuario.
- F. Interfaz de usuario.
 - i. Se construye utilizando vistas que se adaptan al dispositivo desde el que se accede.
 - ii. Utiliza plantillas para mantener la coherencia visual en toda la aplicación.
- G. Módulo de búsqueda y filtrado.
 - i. Ofrece una barra de búsqueda que actúa como filtro para facilitar la localización de libros específicos.
 - ii. Permite filtrar libros por el criterio de su título o autor.
- H. Módulo de seguridad.
 - i. Implementa medidas de seguridad para proteger contra el acceso no autorizado y la inyección SQL.
 - ii. Maneja la validación de entradas del usuario.

2.3. IMPLEMENTACIÓN

La implementación de la aplicación Biblioteca se lleva a cabo a través de la traducción del diseño del sistema y sus especificaciones en código fuente funcional, estructurado de acuerdo con el patrón de diseño Modelo-Vista-Controlador (MVC). Dicho patrón de diseño facilita la separación de la lógica de la aplicación, la interfaz de usuario y la interacción con la base de datos, permitiendo así un desarrollo y un mantenimiento más claros y manejables.

2.3.1. MODELOS

Los modelos en la aplicación son la representación de las entidades de la base de datos y actúan como la capa de abstracción de los datos. Por ejemplo, el modelo *Libro.cs* encapsula las propiedades que definen un libro, tales como su ID, título, autor, año de publicación y si está o no prestado (mediante un booleano). De forma similar, *Usuario.cs* define los atributos de los usuarios, incluyendo su ID, nombre, contraseña y rol (también mediante un booleano, si

es *true* es administrador y si es *false* es usuario común). Estos modelos son utilizados por el Entity Framework Core para generar las tablas correspondientes en la base de datos SQL y para manejar las operaciones CRUD de lectura, creación, actualización y eliminación.

2.3.2. VISTAS

Las vistas son archivos con extensión *.cshtml* que contienen la sintaxis HTML y Razor permitiendo mezclar lenguaje de marcas con código C#. Las vistas se comunican con los modelos para presentar la información de forma dinámica. Por ejemplo, *ListaLibros.cshtml* muestra todos los libros en una tabla HTML, y cada libro tiene acciones asociadas que permiten editar, ver detalles o eliminar el libro.

Así como con los libros, todas las entidades son manejadas en las vistas y mostradas al usuario de la forma más apropiada y visualmente coherente que corresponda. Se implementan también formularios con validaciones para garantizar la integridad de los datos al crear o editar registros.

Para una mayor apariencia de calidad estética se ha empleado el preprocesador de CSS Bootstrap en todas las vistas, logrando así una profesionalidad y sobre todo una coherencia estética en todas las páginas y secciones de la aplicación web. Para el manejo del cambio de modo oscuro a claro y viceversa se ha definido un archivo CSS propio, con la correspondiente función en JavaScript inserta en el código de la vista *_Layout.cshtml*.

2.3.3. CONTROLADORES

Los controladores manejan la lógica de negocio y las solicitudes del usuario, comunicándose con los modelos y seleccionando las vistas apropiadas para responder a dichas solicitudes. *LibroController.cs*, por ejemplo, contiene métodos que responden a las rutas definidas, como *ListaLibros()*, *DetalleLibro(int Id)* o *CrearLibro()*. Cada método utiliza el mánager *BibliotecaManager.cs*, un servicio que encapsula la lógica de interacción con la base de datos, aprovechando las funcionalidades del Entity Framework para las operaciones de los datos.

El mánager *BibliotecaManager.cs* es crucial en la capa de servicio y gestiona la lógica de acceso a datos. Provee métodos como *GetAllLibros()*, *CrearLibro(Libro libro)* y *PedirPrestado(int libroId, int userId)*. Este último, por ejemplo, implica verificar si el libro está o no está disponible y, de ser así, actualizar su estado a prestado y crear un nuevo registro en la tabla de la base de datos *LibrosPrestados*.

2.3.4. AUTENTICACIÓN Y SEGURIDAD

La autenticación de los usuarios se maneja a través de sesiones, utilizando *HttpContext.Session* para almacenar y recuperar información del usuario actual. Se implementa un filtro personalizado llamado *AuthorizeUser* que se aplica a los controladores o acciones para proteger las rutas que requieren de un usuario autenticado.

2.3.5. FRONT-END

Como se mencionó anteriormente, la interfaz de usuario se enriquece con Bootstrap para un diseño responsivo y atractivo visualmente. Se incluye un mecanismo de búsqueda y filtrado, permitiendo a los usuarios buscar libros y filtrar por título y autor. Además, se emplea JavaScript para mejorar la interactividad, como en la funcionalidad de alternancia entre el modo claro y oscuro anteriormente tratado.

2.3.6. DEPSLIEGUE

Para el despliegue, la aplicación se configura para ejecutarse en servidores compatibles con ASP.NET Core, y se preparan scripts de base de datos para una puesta en marcha rápida en diferentes entornos.

En resumen, la implementación efectiva de la aplicación Biblioteca se basa en una traducción metódica y detallada del diseño del sistema en un código fuente, en este caso en lenguaje C# combinado con Razor, CSS y JavaScript. Dicho código está bien organizado y es escalable, asegurando que el producto final sea robusto, seguro y sencillo de usar.

2.4. PRUEBAS

El proceso de pruebas en el desarrollo de software es una fase crítica que garantiza la calidad, funcionalidad y fiabilidad de la aplicación antes de su lanzamiento al mercado. Es una práctica que implica dos aspectos fundamentales: la verificación y la validación del código. A través de estas prácticas se busca asegurar que la aplicación no sólo esté construida correctamente según su diseño técnico, sino que también cumpla con las necesidades y expectativas del usuario final.

2.4.1. VERIFICACIÓN DEL CÓDIGO

La verificación es una evaluación interna del código y la arquitectura de la aplicación para asegurar que se adhiere a los estándares de calidad y cumple con todos los requisitos técnicos especificados en la fase de diseño. Es un proceso detallado que se lleva a cabo a lo largo del desarrollo y se enfoca en evitar errores, buscando consistencia, completitud y la ausencia de defectos técnicos.

En la aplicación "Biblioteca", la verificación comienza con la revisión de código, donde se evalúa la sintaxis, la lógica y las buenas prácticas de programación. Se verifica la correcta implementación del patrón MVC, donde los modelos representan con precisión las entidades de la base de datos, las vistas muestran la información de manera clara y accesible, y los controladores gestionan la lógica de negocio de forma eficiente.

Se emplean pruebas unitarias, especialmente en el *BibliotecaManager.cs*, para verificar cada método y función. Estas pruebas unitarias evalúan la lógica de negocio, como la creación de un nuevo libro, la edición, el préstamo y la devolución, así como las validaciones de los datos de entrada.

Las revisiones de código son también parte de este proceso, donde otros desarrolladores pueden examinar el código para detectar errores que el autor original podría haber pasado por alto. Esto incluye verificar la seguridad del código, como la protección contra inyecciones SQL y otros ataques comunes.

2.4.2. VALIDACIÓN DEL CÓDIGO

La validación, por otro lado, es un procedimiento que involucra pruebas funcionales y de usuario para confirmar que la aplicación cumple con los requerimientos y necesidades del cliente. Se centra en la utilidad y la experiencia del usuario final con el software, más que en los detalles técnicos internos.

Para la aplicación "Biblioteca", la validación incluye pruebas de funcionalidad donde se simulan escenarios de uso real para asegurar que todas las características trabajan como se espera. Por ejemplo, se verifica que los usuarios puedan buscar libros, realizar préstamos y devoluciones, y que los administradores puedan gestionar los registros de usuarios y libros adecuadamente.

En conjunto, la verificación y la validación forman el proceso de pruebas de la aplicación "Biblioteca". Este proceso no solo es esencial para detectar y corregir problemas antes de que el software sea accesible al público, sino que también ayuda a asegurar que la aplicación cumple con su propósito de facilitar la gestión de una biblioteca y proporcionar una experiencia agradable y eficiente a los usuarios.

3. CONCLUSIONES

Al finalizar el proyecto de desarrollo de la aplicación Biblioteca, se pueden extraer varias conclusiones significativas que reflejan tanto el proceso de desarrollo como el resultado final del software. Este proyecto se inició con la intención de crear una aplicación web intuitiva y eficiente para la gestión de una biblioteca, permitiendo la administración de libros, préstamos y usuarios.

Los objetivos generales del proyecto se han cumplido satisfactoriamente. En primer lugar, se ha desarrollado una interfaz de usuario amigable y accesible, lo que permite a los usuarios, independientemente de su nivel de habilidad técnica, interactuar efectivamente con el sistema. Esto se logró a través de un diseño cuidadoso y la implementación de principios de usabilidad y accesibilidad en la aplicación web.

En segundo lugar, la aplicación proporciona una gestión robusta y fiable de datos. La utilización de Entity Framework Core para la interacción con la base de datos ha asegurado que las operaciones de creación, lectura, actualización y eliminación (CRUD) de los registros de libros y usuarios se manejen de manera eficaz. Esto cumple con el objetivo general de facilitar las operaciones diarias de la biblioteca a través de una solución digital.

Respecto a los objetivos específicos, se ha logrado la implementación de un sistema de autenticación que asegura que solo los usuarios autorizados puedan acceder a ciertas funcionalidades del sistema. Esto garantiza la seguridad de los datos y la integridad del sistema. Además, se ha incorporado un sistema de registro y seguimiento de préstamos de libros, que permite un control detallado sobre los libros prestados y disponibles, así como un historial de la actividad de préstamo para cada usuario.

Otro objetivo específico cumplido es la adaptabilidad de la aplicación a diferentes dispositivos, gracias al uso de Bootstrap para el diseño responsivo. Esto asegura que los usuarios puedan acceder a la aplicación desde cualquier dispositivo con conexión a internet, brindando flexibilidad y movilidad en su uso.

En resumen, la aplicación Biblioteca desarrollada en este Proyecto de Fin de Ciclo del Grado Superior de Desarrollo de Aplicaciones Web cumple con los requisitos planteados al inicio de esta memoria, y ha seguido las líneas de desarrollo marcadas en el diseño de la estructura de datos y de las necesidades de los usuarios.

4. LÍNEAS FUTURAS

En primer lugar, la primera propuesta de mejora sería la de ampliar las pruebas y test a los que se somete la aplicación. En un entorno de producción real, deberían realizarse además de las hechas en este proyecto, pruebas de integración y pruebas de aceptación.

Las pruebas de aceptación del usuario son otro componente crucial de la validación. Aquí, usuarios reales o representantes del cliente utilizan la aplicación en un entorno controlado para proporcionar retroalimentación. Se presta especial atención a la usabilidad, accesibilidad y satisfacción general del usuario.

En cuanto a las pruebas de integración, éstas verifican que todos los componentes de la aplicación funcionen juntos de manera armoniosa. Esto es especialmente importante cuando se integran servicios externos o APIs, como los sistemas de autenticación o servicios de base de datos.

Además de esa propuesta en cuanto a las pruebas realizables sobre mi aplicación, y a pesar de que la solución que he desarrollado ha demostrado ser robusta y funcional para la gestión y el mantenimiento de una biblioteca a través de una aplicación web, el mundo de la tecnología es dinámico y siempre hay espacio para la innovación y la mejora. A continuación, se presentan algunas propuestas de mejora y líneas futuras para el desarrollo de la aplicación:

- A. Integración con sistemas de identificación digital: Ampliar las funcionalidades de registro y login para incorporar sistemas de autenticación más avanzados como OAuth, permitiendo a los usuarios acceder mediante redes sociales o cuentas de Google.
- B. Recomendaciones personalizadas: Implementar algoritmos de aprendizaje automático que analicen los libros pedidos como préstamo por los usuarios para ofrecer recomendaciones personalizadas de libros.
- C. Interfaz de usuario adaptable y accesible: Rediseñar la interfaz de usuario para que sea completamente responsiva y accesible, asegurando el fácil acceso para personas con discapacidades.
- D. Expansión de la funcionalidad de búsqueda: Enriquecer las opciones de búsqueda para incluir filtros avanzados por año de publicación o estado de préstamo, y la posibilidad de buscar dentro del contenido de los libros si se dispone de las versiones digitales.
- E. Internacionalización y localización: Preparar la aplicación para soportar múltiples idiomas, lo que permitiría su uso en diversas regiones y aumentaría su alcance de mercado.
- F. Sistema de notificaciones: Desarrollar un sistema de notificaciones para alertar a los usuarios sobre fechas de devolución de libros, nuevos lanzamientos o eventos de la biblioteca.
- G. Integración de un módulo de análisis de datos: Incorporar herramientas de inteligencia de negocio para que los administradores puedan obtener estadísticas sobre el uso de la biblioteca y tomar decisiones basadas en datos.
- H. Ampliación de la base de datos y servicios de API: Facilitar la integración con otras bibliotecas y servicios en línea mediante APIs para intercambiar datos de libros y disponibilidad en tiempo real.

Estas mejoras y expansiones son propuestas que tienen como objetivo no sólo mejorar la experiencia del usuario, sino también aumentar la funcionalidad y el valor comercial de la aplicación Biblioteca, asegurando su relevancia y utilidad en el futuro.

5. BIBLIOGRAFÍA

- Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- Microsoft. (n.d.). Entity Framework Core. Documentación oficial de Microsoft. Recuperado de <https://docs.microsoft.com/en-us/ef/core/>
- Mozilla. (n.d.). MDN Web Docs. Recuperado de <https://developer.mozilla.org/>
- Twitter. (n.d.). Bootstrap. Documentación oficial de Bootstrap. Recuperado de <https://getbootstrap.com/docs/>
- Stack Exchange. (n.d.). Stack Overflow. Recuperado de <https://stackoverflow.com>