



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño e implementación de una aplicación web para la
gestión de libros en bibliotecas

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Santos Martínez, Natalia

Tutor/a: Sapena Vercher, Oscar

CURSO ACADÉMICO: 2022/2023

Resumen

El propósito de llevar a cabo este proyecto tiene dos enfoques fundamentales. Por un lado, busca la aplicación práctica de gran parte de los conocimientos obtenidos a lo largo de la formación universitaria. Por otro lado, busca abordar una necesidad existente en una parte de la sociedad, específicamente en lo relacionado con la obtención de libros en las bibliotecas públicas.

BiblioConnect es una aplicación web diseñada para mejorar el acceso y la gestión de libros en las bibliotecas públicas de la Comunidad Valenciana. La iniciativa surge con la intención de fomentar la lectura en su formato físico, al mismo tiempo que se agiliza el proceso de acceso a los recursos de lectura.

La aplicación se compone de dos sesiones: una para lectores y otra para administradores de bibliotecas. Los administradores tienen la capacidad de gestionar eventos y libros, mientras que los lectores pueden conocer su disponibilidad y solicitar sus recursos. Las funcionalidades se complementan con notificaciones y opciones de filtrado, mejorando la experiencia del usuario.

El sistema busca simplificar el acceso a los diferentes recursos educativos y culturales. La centralización de información en una única aplicación y la eliminación de barreras para acceder a recursos benefician tanto a los administradores de bibliotecas como a los lectores.

Las tecnologías usadas en el desarrollo de este sistema han sido ASP.NET y C# para manejar la lógica del servidor, HTML para la definición de la estructura y el contenido de las páginas web, SQL Server para el almacenamiento de datos y JavaScript para gestionar la interacción del usuario con la interfaz web. Como entorno de desarrollo para la implementación y diseño de la aplicación se ha utilizado Visual Studio.

Palabras clave: aplicación web, API REST, portal de reservas, gestión de bibliotecas, lector, geolocalización.

Abstract

The purpose of carrying out this project has two fundamental approaches. On the one hand, it seeks the practical application of much of the knowledge obtained throughout university education. On the other hand, it seeks to address an existing need in part of society, specifically in relation to obtaining books in public libraries.

BiblioConnect is a web application designed to improve access to and management of books in the public libraries of the Valencian Community. The initiative arises with the intention of promoting reading in its physical format, while streamlining the process of access to reading resources.

The application consists of two sessions: one for readers and one for library administrators. Administrators have the ability to manage events and books, while readers can know their availability and request their resources. The functionalities are complemented by notifications and filtering options, enhancing the user experience.

The system seeks to simplify access to different educational and cultural resources. The centralization of information in a single application and the elimination of barriers to access resources benefit both library administrators and readers.

The technologies used in the development of this system have been ASP.NET and C# to manage the server logic, HTML for the definition of the structure and content of the web pages, SQL Server for data storage and JavaScript to manage the user interaction with the web interface. Visual Studio was used as development environment for the implementation and design of the application.

Keywords: web application, REST API, reservation portal, library management, reader, geolocation.

Índice de contenido

1. Introducción.....	9
1.1 Motivación	9
1.2 Objetivos	9
1.3 Estructura	10
2. Estado del arte	10
2.1 eBiblio.....	10
2.2 Biblio Digital.....	11
2.3 Conclusiones	13
3. Análisis del problema.....	14
3.1 Análisis del problema.....	14
3.2 Solución actual	14
3.3 Estudio de la viabilidad.....	14
3.3.1 Viabilidad temporal.....	15
3.3.2 Diagrama de Gantt.....	16
3.3.3 Viabilidad operativa	16
3.3.4 Viabilidad empresarial.....	17
4. Especificación de requisitos.....	17
4.1 Requisitos Funcionales.....	17
4.2 Requisitos No Funcionales.....	18
4.3 Diagramas UML.....	18
4.3.1 Diagrama de clases	18
4.3.2 Diagrama de contexto.....	23
4.3.3 Diagrama de casos de uso.....	23
4.4 Descripción de casos de uso y Diagramas de Flujo	24
5. Diseño	33
5.1 Modelo de la base de datos.....	33
5.2 Arquitectura Software	34
5.2.1 Patrón arquitectónico MVC.....	34
5.2.2 Patrón DAO	35
5.3 Diseño con prototipado	36
5.3.1 Login	36
5.3.2 Registro lector	36
5.3.3 Listado libros y eventos.....	37
5.3.4 Modales	39
5.3.5 Préstamos.....	40
5.3.6 Menús	41
5.4 Estilo	41
5.4.1 Logo	42
5.4.2 Tipografía	42
5.4.3 Colores.....	42
6. Tecnologías utilizadas	43
6.1 Frontend	43



6.1.1	HTML.....	43
6.1.2	CSS.....	44
6.1.3	Bootstrap	44
6.1.4	Leaflet.....	45
6.1.5	FullCalendar	45
6.2	BackEnd	45
6.2.1	ASP.NET	45
6.2.2	C#	46
6.2.3	SQL Server	46
6.2.4	Firebase	46
6.2.5	JavaScript	47
6.2.6	REST	47
6.3	Otras herramientas.....	47
6.3.1	Visual Studio 2019	47
6.3.2	GitHub.....	48
6.3.3	Figma.....	48
6.3.4	Trello	49
6.3.5	TeamGantt	49
6.3.6	Postman	49
6.4	Conclusiones	50
7.	Implementación	50
7.1	API REST de la aplicación.....	50
7.2	Estructura del proyecto.....	53
7.3	Modelo Vista Controlador.....	55
7.3.1	Modelo.....	55
7.3.2	Vista	56
7.3.3	Controlador.....	57
7.4	Patrón DAO.....	58
8.	Pruebas.....	60
8.1	Encuesta inicial de usabilidad	60
8.1.1	Primera prueba.....	60
9.	Conclusiones	61
9.1	Relación del proyecto con los estudios cursados	62
9.2	Trabajo futuro.....	63
10.	Bibliografía	64
11.	Apéndice	66
11.1	Manual de usuario	66
11.1.1	Pantalla inicio	66
11.1.2	Menús	69
11.1.3	Sesión administrador biblioteca.....	70
11.1.4	Sesión lector	82
11.1.5	Cerrar sesión.....	93
12.	Anexo ODS.....	94

Índice de figuras

Figura 1:	Catálogo de libros en eBiblio	11
Figura 2:	Préstamo y reserva en eBiblio	11
Figura 3:	Catálogo libros en Biblio Digital.....	12
Figura 4:	Listado eventos en Biblio Digital.....	12
Figura 5:	Directorio de búsqueda de bibliotecas en Biblio Digital.....	13
Figura 6:	Diagrama de Gantt.....	16
Figura 7:	Diagrama de clases.....	19
Figura 8:	Diagrama de contexto.....	23
Figura 9:	Diagrama de casos de uso	23
Figura 10:	Diagrama de flujo Login	24
Figura 11:	Diagrama de flujo CRUD libros.....	28
Figura 12:	Diagrama de flujo CRUD eventos.....	29
Figura 13:	Diagrama de fujo asignación libro	31
Figura 14:	Diagrama modelo de base de datos	34
Figura 15:	Patrón Modelo-Vista-Controlador.....	35
Figura 16:	Prototipo Login	36
Figura 17:	Prototipo Registro lector	36
Figura 18:	Prototipo Listado libros desde administrador.....	37
Figura 19:	Prototipo Listado eventos desde administrador.....	37
Figura 20:	Prototipo Catálogo libros.....	38
Figura 21:	Prototipo Catálogo eventos	38
Figura 22:	Prototipo Modal “registro libro”	39
Figura 23:	Prototipo Modal “registro evento”	39
Figura 24:	Prototipo Listado préstamos desde administrador.....	40
Figura 25:	Prototipo Listado Préstamos desde lector.....	41
Figura 26:	Prototipos menú.....	41
Figura 27:	Logo BiblioConnect	42
Figura 28:	Tipografía aplicación.....	42
Figura 29:	Paleta de colores.....	43
Figura 30:	Logo HTML	43
Figura 31:	Logo CSS	44
Figura 32:	Logo Bootstrap.....	44
Figura 33:	Logo Leaflet	45
Figura 34:	Logo FullCalendar.....	45
Figura 35:	Logo ASP.NET	45



Figura 36:	Logo C#.....	46
Figura 37:	Logo SQL Server	46
Figura 38:	Logo Firebase.....	46
Figura 39:	Logo JavaScript.....	47
Figura 40:	Logo servicios REST.....	47
Figura 41:	Logo Visual Studio.....	47
Figura 42:	Logo GitHub	48
Figura 43:	Logo Figma	48
Figura 44:	Logo Trello.....	49
Figura 45:	Logo teamgantt.....	49
Figura 46:	Logo Postman.....	49
Figura 47:	Estructura del proyecto.....	53
Figura 48:	Código Modelo Libro.....	55
Figura 49:	Código Herencia de clases	56
Figura 50:	Código Vista.....	56
Figura 51:	Código Controlador biblioteca	57
Figura 52:	Código Controlador lector.....	57
Figura 53:	Diagrama patrón DAO	58
Figura 54:	Código funciones DAL	59
Figura 55:	Vista pantalla de inicio	66
Figura 56:	Vista credenciales incorrectas Login.....	67
Figura 57:	Vista Modal registro lector.....	67
Figura 58:	Vista Modal registro biblioteca	68
Figura 59:	Vista campos incorrectos.....	69
Figura 60:	Vista Menú Biblioteca.....	69
Figura 61:	Vista Menú Lector.....	70
Figura 62:	Vista Dashboard	70
Figura 63:	Vista Perfil biblioteca.....	71
Figura 64:	Vista seleccionar “Editar perfil”.....	71
Figura 65:	Vista Modal editar perfil biblioteca.....	71
Figura 66:	Vista Listado de libros desde biblioteca.....	72
Figura 67:	Vista búsqueda de libros desde biblioteca.....	72
Figura 68:	Vista Modal registrar libro	73
Figura 69:	Vista seleccionar icono “modificar libro”	73
Figura 70:	Vista Modal modificar libro	74
Figura 71:	Vista seleccionar icono “eliminar libro”	74
Figura 72:	Vista cuadro de diálogo confirmar eliminación libro.....	75

Figura 73:	Vista Listado de eventos desde biblioteca.....	75
Figura 74:	Vista Calendario de eventos formato “Mes”	76
Figura 75:	Vista Calendario de eventos formato “Semana”	76
Figura 76:	Vista Calendario de eventos formato “Día”	77
Figura 77:	Vista Calendario de eventos formato “Listado”	77
Figura 78:	Vista Modal información evento en calendario.....	78
Figura 79:	Vista Listado de todos los préstamos desde biblioteca	78
Figura 80:	Vista búsqueda de préstamos	79
Figura 81:	Vista listado de préstamos “En espera”	79
Figura 82:	Vista listado de préstamos “Pendiente recogida”	80
Figura 83:	Vista Modal recogida del libro	80
Figura 84:	Vista listado de préstamos “Pendiente devolución”	81
Figura 85:	Vista Modal devolución del libro	81
Figura 86:	Vista listado de préstamos “Devueltos”	82
Figura 87:	Vista Modal estado del libro devuelto.....	82
Figura 88:	Vista pestaña Inicio desde lector	83
Figura 89:	Vista Catálogo de libros desde lector	84
Figura 90:	Vista Menú filtrado de libros.....	84
Figura 91:	Vista Filtrado de libros	85
Figura 92:	Vista Búsqueda de libros.....	85
Figura 93:	Vista información detalle de un libro	86
Figura 94:	Vista “Mostrar más”	87
Figura 95:	Vista “Mostrar menos”	87
Figura 96:	Vista Tarjeta Préstamo/Reserva de un libro	88
Figura 97:	Vista seleccionar enlace perfil biblioteca	88
Figura 98:	Vista Reseñas de un libro	89
Figura 99:	Vista Modal escritura de una reseña.....	89
Figura 100:	Vista Listado de Eventos desde lector	90
Figura 101:	Vista información sobre un evento	90
Figura 102:	Vista historial de los préstamos de un lector	91
Figura 103:	Vista Notificaciones de un lector.....	91
Figura 104:	Vista perfil de un lector	92
Figura 105:	Vista Modal editar perfil lector	92
Figura 106:	Vista cuadro diálogo confirmación edición perfil	93
Figura 107:	Vista Cierre de sesión.....	93

Índice de tablas

Tabla 1: Comparativa Estado del arte.....	13
Tabla 2: Fases del desarrollo del proyecto	15
Tabla 3: Estimación total de horas	16
Tabla 4: Resultados de la encuesta	61

1. Introducción

1.1 Motivación

La inspiración detrás de este Trabajo de Fin de Grado surge directamente de mis propias experiencias personales mientras cursaba mis estudios en el instituto. Durante ese tiempo, me enfrenté repetidamente a la tarea de conseguir en bibliotecas públicas, libros de texto y otros recursos educativos que eran requeridos para mis asignaturas. Cada vez que necesitaba obtener un libro, ya fuera de literatura, inglés o valenciano, tenía que realizar diferentes acciones para conseguirlos.

El principal inconveniente que percibía era la necesidad de contactar individualmente con cada biblioteca cercana para verificar si el libro que necesitaba estaba disponible. Esto a menudo implicaba realizar numerosas llamadas telefónicas, muchas de las cuales quedaban sin respuesta. Además, en algunos casos tenía que visitar físicamente las bibliotecas para obtener la información necesaria.

Tras reflexionar sobre estas dificultades, me di cuenta de que la tecnología actual tenía el potencial suficiente para resolver este problema y mejorar la experiencia educativa para otros estudiantes que se encontraran en situaciones similares. Por tanto, tomé la decisión de diseñar e implementar un sistema que centralizara la información sobre la disponibilidad de libros en las bibliotecas públicas más cercanas, eliminando así la necesidad de realizar múltiples llamadas o visitas que consumieran una cantidad significativa del tiempo del usuario.

1.2 Objetivos

El objetivo principal del proyecto es crear una aplicación web que permita visualizar los libros disponibles y realizar préstamos en las distintas bibliotecas.

Por otro lado, los subobjetivos u objetivos secundarios que han permitido adquirir nuevas competencias son los siguientes:

- Realizar un análisis comparativo de las tecnologías y soluciones similares existentes.
- Estudiar la documentación de diversas APIs públicas para su consumo e integración de datos.
- Aplicar las técnicas de planificación y estimación de Gestión de Proyectos.
- Realizar el diseño e implementación completa de una aplicación funcional.
- Crear una interfaz de usuario amigable con soporte a la lectura, inserción, modificación y borrado de datos.
- Aplicar técnicas de validación de formularios.

1.3 Estructura

A continuación, se proporcionará un resumen de los temas tratados en cada capítulo de la memoria, con el propósito de facilitar la búsqueda de información específica.

- En el primer capítulo, se explica de manera resumida cuáles han sido los motivos que han llevado a realizar este proyecto y los objetivos de este.
- En el segundo, se ha realizado un análisis de las características que contienen las aplicaciones similares a la aplicación que se desarrollará en este proyecto.
- En el tercer capítulo, se hace un análisis del problema y se plantea una solución al mismo, al mismo tiempo que se realiza un estudio de la viabilidad del proyecto.
- En el cuarto capítulo, se especifican los requisitos funcionales y no funcionales del sistema, además de presentar diagramas UML con sus correspondientes descripciones.
- En el quinto capítulo, se incluye el modelo de la base de datos, una descripción de la arquitectura de la aplicación, el diseño del prototipado y la definición de los estilos visuales.
- En el sexto, se detallan las tecnologías utilizadas tanto para el Frontend como para el Backend, junto con otras herramientas empleadas para el desarrollo.
- En el séptimo capítulo, se presenta la implementación del sistema mediante una API REST, se describe la estructura de carpetas del proyecto y se exploran los patrones arquitectónicos MVC y DAO.
- El octavo capítulo consiste en las pruebas realizadas para comprobar el nivel de usabilidad de la aplicación.
- El noveno capítulo contiene las conclusiones del trabajo, que incluyen un resumen del trabajo hecho y los objetivos alcanzados, la relación del proyecto con los estudios cursados y se discuten las posibles mejoras en la aplicación.
- El décimo, contiene los enlaces bibliográficos de la redacción de la aplicación siguiendo la norma internacional ISO 690-2010.
- En el penúltimo capítulo, se incluye el apéndice del proyecto que contiene el manual de la aplicación acompañado de capturas de pantalla de la versión final.
- En este último capítulo, se identifican y describen los Objetivos de Desarrollo Sostenible (ODS) que la aplicación cumple.

2. Estado del arte

Existen multitud de aplicaciones en el mercado para la gestión de bibliotecas, con sus diferentes funcionalidades y características.

2.1 eBiblio

Las bibliotecas públicas españolas tienen disponible el servicio gratuito en línea llamado eBiblio [21] para el préstamo de audiolibros y libros electrónicos junto a revistas y periódicos. Es un servicio de la red electrónica de lectura pública valenciana que hace posible la lectura de libros electrónicos a través de internet. Esta aplicación está subvencionada por la Generalitat Valenciana, junto con el Ministerio de Cultura y Deporte.

La única condición para comenzar a leer sin costo alguno es poseer un carné válido y estar registrado en alguna de las bibliotecas públicas participantes del sistema. Esta plataforma se encuentra disponible durante todo el día a través de Internet y permite tanto la lectura en línea como la lectura sin conexión a través de la descarga. La colección de eBiblio dispone de obras de ficción y no ficción para todo tipo de públicos, clásicos y novedades. La estructura del catálogo de libros de la aplicación eBiblio se muestra en la Figura 1.

Figura 1: Catálogo de libros en eBiblio

El funcionamiento de los préstamos es sencillo ya que, básicamente, es idéntico al de una biblioteca tradicional. El catálogo de eBiblio dispone de una gran cantidad de copias digitales que van circulando por los lectores en forma de préstamo. En caso de que el libro esté disponible aparecerá la opción de “Prestar”, en ese momento se podrá comenzar a leer el recurso. Por otra parte, si el libro está en uso por otros lectores, aparecerá la opción de “Reservar” con la fecha en la que estará disponible.



Figura 2: Préstamo y reserva en eBiblio

2.2 Biblio Digital

Biblio Digital [22] es una Red de Bibliotecas Municipales que está formada por más de 228 bibliotecas, que colaboran de forma coordinada para que todos los habitantes de la provincia de Barcelona tengan un acceso igualitario a la cultura

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

El proyecto está promovido en colaboración con el Departamento de Cultura de la Generalitat, la Diputación de Barcelona y el Ayuntamiento de Barcelona. Con este servicio se garantiza la descarga, acceso y consulta legal de contenidos digitales, que se ofrece a través de una plataforma sencilla y abierta a los usuarios de las bibliotecas públicas de Cataluña.

A partir de la aplicación es posible acceder al catálogo centralizado que contiene más de 8 millones de ejemplares (libros, DVD, CD, diarios y revistas) de todas las bibliotecas municipales de la provincia de Barcelona. En la Figura 3 se puede observar el catálogo de libros de la aplicación Biblio Digital.

The screenshot shows the Biblio Digital website's search interface. At the top, there are links for 'INICIO' and 'CLUB DE LECTURA', and a search bar with a magnifying glass icon. Below the search bar are logos for 'Biblioteques Pùbliques de Catalunya', 'Generalitat de Catalunya Departament de Cultura', 'Diputació de Barcelona Xarxa de Biblioteques Municipals', and 'Biblioteques de Barcelona'. On the right, there is a logo for 'Biblio Digital'. The main content area has a blue header 'Materia' with dropdown menus for 'Artes', 'Ciencia y tecnología', 'Ciencias de la tierra y medio ambiente', 'Ciencias sociales', 'Desarrollo personal', 'Ficción', and 'VER TODOS'. Below this is a section titled 'Wittgenstein, Ludwig: genio o visionario?' featuring several book covers by Ludwig Wittgenstein, such as 'Ludwig Wittgenstein' by RAY Monk, 'LUDWIG WITTGENSTEIN' by JUAN GENS, and 'Wittgenstein /' by WITTGENSTEIN. There is also a 'VER MÁS' button.

Figura 3: Catálogo libros en Biblio Digital

Hay una sección de actividades que incluye clubs del libro, que son un ciclo de encuentros con escritores en propias bibliotecas, promovido por la Diputación de Barcelona y con la colaboración de varias editoriales.

The screenshot shows the Biblio Digital website's event listing. At the top, there are links for 'BibliotecaVirtual', 'Diputació de Barcelona Xarxa de Biblioteques Municipals', and a search bar. Below this is a red navigation bar with links for 'Colección', 'Bibliotecas', 'Actividades', 'Recomendaciones', 'Catálogo Aladí', 'Préstamo digital', 'Gènus', and 'Trencadís'. The main content area shows a section for 'Club del libro' with a red banner for 'EL CLUB DEL LLIBRE'. It includes a bio for Gemma Ruiz and a photo of her. Another section for Espartac Peran is also shown with his bio and a photo.

Figura 4: Listado eventos en Biblio Digital

La propia aplicación de BibliotecaVirtual ofrece un directorio de búsqueda de las bibliotecas municipales de la provincia de Barcelona.

The screenshot shows the header with navigation links: Colección, Bibliotecas, Actividades, Recomendaciones, Catálogo Aladi, Préstamo digital, Génius, and Trencadís. Below the header is a breadcrumb trail: Biblioteca Virtual > Bibliotecas > Busca una biblioteca. There are three buttons: Busca por municipio, Bibliobuses, and Mapa de bibliotecas. The main title is "Directorio de la Red de Bibliotecas Municipales de la provincia de Barcelona". Below it is a redacted section labeled "A" followed by a table with columns: MUNICIPI, BIBLIOTECA, TELÉFON, and ADREÇA ELECTRÒNICA. The table lists municipalities like Abrera, Aguilar de Segarra, Aiguafreda, Alella, Alpens, Ametlia del Valles, Arenys de Mar, Arenys de Munt, and Argençola, along with their respective library names, phone numbers, and email addresses.

MUNICIPI	BIBLIOTECA	TELÉFON	ADREÇA ELECTRÒNICA
Abrera	Josep Roca i Bros	937 700 881	b.abrera.jrb@diba.cat
Aguilar de Segarra	Bibliobús Serra de Rubió	666 541 136	b.serraderubio@diba.cat
Aiguafreda	Lluís Millet i Pagès	935 141 207	b.aiguafreda.lmp@diba.cat
Alella	Ferrer i Guardia	935 559 055	b.alella.fg@diba.cat
Alpens	Bibliobús Guilleries	616 978 825	b.guilleries@diba.cat
Ametlia del Valles, l'	Josep Badia i Moret	938 431 966	b.ametllav.jbm@diba.cat
Arenys de Mar	Pare Fidel Fita	937 923 253	b.arenysma.pff@diba.cat
Arenys de Munt	Antònia Torrent i Martori	937 938 448	b.arenysmu.sm@diba.cat
Argençola	Bibliobús Serra de Rubió	666 541 136	b.serraderubio@diba.cat

Figura 5: Directorio de búsqueda de bibliotecas en Biblio Digital

2.3 Conclusiones

En la Tabla 1 se ha realizado un análisis de las características que presenta cada una de las plataformas estudiadas en relación con BiblioConnect.

	eBiblio	Biblio Digital	BiblioConnect
Gestión de eventos literarios	No	Sí	Sí
Calendario de eventos	No	No	Sí
Geolocalización lectores	No	No	Sí
Geolocalización Bibliotecas	No	Sí	Sí

Tabla 1: Comparativa Estado del arte

Las aplicaciones de eBiblio y Biblio Digital son dos aplicaciones oficiales de la red española de bibliotecas. Dependiendo de la comunidad autónoma del usuario es necesario el uso de una u otra; aunque ambas dan acceso a un catálogo de libros muy similar. La restricción de uso es la misma: disponer de un carnet válido de una biblioteca. Las comunidades para las que son compatibles:

- **Biblio Digital:** Cataluña
- **eBiblio:** Andalucía, Aragón, Asturias, Canarias, Cantabria, Castilla y León, Castilla-La Mancha, Ceuta, Comunidad Valenciana, Extremadura, Galicia, Islas Baleares, La Rioja, Madrid, Melilla, Murcia y Navarra.

A pesar de tener diferentes nombres y creadores, las dos aplicaciones funcionan e interactúan de manera muy similar. Siempre que el libro esté disponible, puede obtener un

préstamo digital para leer el libro en un dispositivo electrónico durante el período de préstamo de tres semanas.

3. Análisis del problema

3.1 Análisis del problema

Los principales inconvenientes de los sistemas de bibliotecas actuales:

- Algunas personas prefieren la experiencia de lectura en papel, debido a aspectos táctiles, como sentir el peso y la textura del libro, pasar páginas físicamente o subrayar páginas de manera directa. La imposibilidad de realizar reservas para la lectura en papel puede resultar negativo para gran parte de los usuarios.
- Existen múltiples ejemplares que no están disponibles en formato digital y solo se pueden encontrar en bibliotecas públicas. Todavía hay obras que no han sido digitalizadas o que no están incluidas en las colecciones digitales disponibles. Por ejemplo: *Crónica de una muerte anunciada* (Gabriel García Márquez) o *El principito* (Antoine de Saint-Exupéry), son ejemplares que no han sido digitalizados en eBiblio pero si se encuentran disponibles en algunas bibliotecas públicas de la Comunidad Valenciana.
- Actualmente los usuarios que deseen un libro en papel tienen que llamar o acudir presencialmente a las bibliotecas más cercanas para verificar su disponibilidad. Esto conlleva a una pérdida de tiempo, ya que los usuarios es posible que tengan que comunicarse con varias bibliotecas para encontrar el libro que buscan.
- Muchas de las bibliotecas de la Comunidad Valenciana tienen de manera única e independiente su propia página web. Esto conlleva a que cada usuario que quiera acceder a la información de alguna de las bibliotecas tenga que realizar diversas búsquedas sobre páginas que contiene que presentan estructuras diferentes. Además, cabe la posibilidad de que muchas de ellas no dispongan de una página web y por tanto, la información sobre ellas sea demasiado escueta.

3.2 Solución actual

Con el sistema actual se pretende lidiar con los problemas citados anteriormente, facilitando la obtención de libros en papel a aquellos usuarios que tengan dichas preferencias.

El objetivo del sistema será centralizar todas las bibliotecas públicas de la Comunidad Valenciana en una sola aplicación. Se permitirá a los usuarios ver la disponibilidad de los libros físicos en todas las bibliotecas de forma rápida y conveniente, evitando así la necesidad de llamar o visitar personalmente cada biblioteca. Los usuarios a partir de búsquedas en la aplicación podrán conocer las bibliotecas que contienen el ejemplar deseado. Además, se utilizarán técnicas de geolocalización para indicar cuales de dichas bibliotecas son más próximas al usuario. Esto optimizará la experiencia del usuario y fomentará un acceso más eficiente a los libros físicos deseados.

3.3 Estudio de la viabilidad

Se evaluará en esta sección si los recursos y el tiempo disponible son adecuados para llevar a cabo el proyecto.

3.3.1 Viabilidad temporal

En esta sección se llevará a cabo un análisis de la viabilidad temporal del proyecto. Como se puede apreciar en la Tabla 2, se realizará una estimación de todas las etapas que involucra el desarrollo del sistema.

- **Desarrollo propuesta técnica:** En esta etapa, se realiza un estudio del contexto actual y se identifican posibles mejoras. Se definen las posibles tareas que contendrá la aplicación, y se diseña el diagrama de Gantt.
- **Análisis y especificación de requisitos:** En esta fase se identifican y definen los requisitos funcionales y no funcionales de la aplicación web, determinando su funcionalidad, características y restricciones.
- **Diseño:** En esta etapa, se realiza el diseño detallado de la aplicación web, incluyendo la arquitectura, el diseño de la interfaz de usuario, la estructura de la base de datos y otros componentes técnicos. Se definen los diagramas y las especificaciones necesarias para guiar el desarrollo.
- **Desarrollo:** En esta fase, se lleva a cabo la implementación de la aplicación web según el diseño establecido. Se escribe y se prueba el código, se crean las funcionalidades, se integran los componentes y se desarrolla la lógica de negocio de la aplicación.
- **Pruebas:** En esta etapa, se prueba todo lo desarrollado en la etapa anterior, buscando posibles fallos o problemas. Se verifica que la aplicación web cumple con los requisitos establecidos y con las expectativas del usuario. Se realiza una revisión final de la funcionalidad, el rendimiento y la usabilidad para asegurar que la aplicación está lista para el usuario final.
- **Documentación:** En esta etapa, se genera la documentación del TFG, que incluye el informe técnico, manuales de usuario, diagramas, descripciones de diseño y cualquier otra documentación relevante para el proyecto. Se documenta el proceso de desarrollo y se explica el funcionamiento de la aplicación.

En la Tabla 2 se puede visualizar cada una de las fases previamente definidas.

Fases del desarrollo	Fecha de inicio	Fecha de finalización	Duración (Horas)
Desarrollo propuesta técnica	21/05/2023	28/05/2023	40
Ánalisis y especificación de requisitos	29/06/2023	07/06/2023	50
Diseño	08/06/2023	18/06/2023	55
Desarrollo	19/06/2023	21/07/2023	165
Pruebas	22/07/2023	26/07/2023	25
Documentación	21/05/2023	30/08/2023	65
Total			400

Tabla 2: Fases del desarrollo del proyecto

Como se indica en la tabla anterior, se ha estimado que el desarrollo del proyecto requerirá un total de 400 horas de trabajo, calculadas considerando una jornada de 5 horas al día. Esta estimación supone que el desarrollador no enfrentará contratiempos imprevistos, como enfermedades inesperadas o retrasos significativos en alguna fase del proyecto. Para mitigar posibles imprevistos, se ha decidido agregar un 10% adicional a las horas estimadas. De esta manera, si el proyecto avanza según lo previsto, debería completarse en su totalidad para el día "30/07/2023", lo que corresponde a 8 días antes de la fecha de entrega programada.

Según lo expuesto previamente, en la Tabla 3 se presenta la estimación de horas totales requeridas para el desarrollo del proyecto.

Estimación del proyecto software	Imprevistos (10% sobre el total de horas)
400	40
Total (Horas) proyecto	440

Tabla 3: Estimación total de horas

3.3.2 Diagrama de Gantt

Un diagrama de Gantt [26] es una herramienta visual utilizada en la gestión de proyectos para planificar, programar y controlar las tareas a lo largo del tiempo. En la Figura 6 se puede ver el diagrama de Gantt seguido por el sistema, cuyo objetivo ha sido mostrar la cronología de las actividades del proyecto, así como su duración y sus posibles solapamientos.

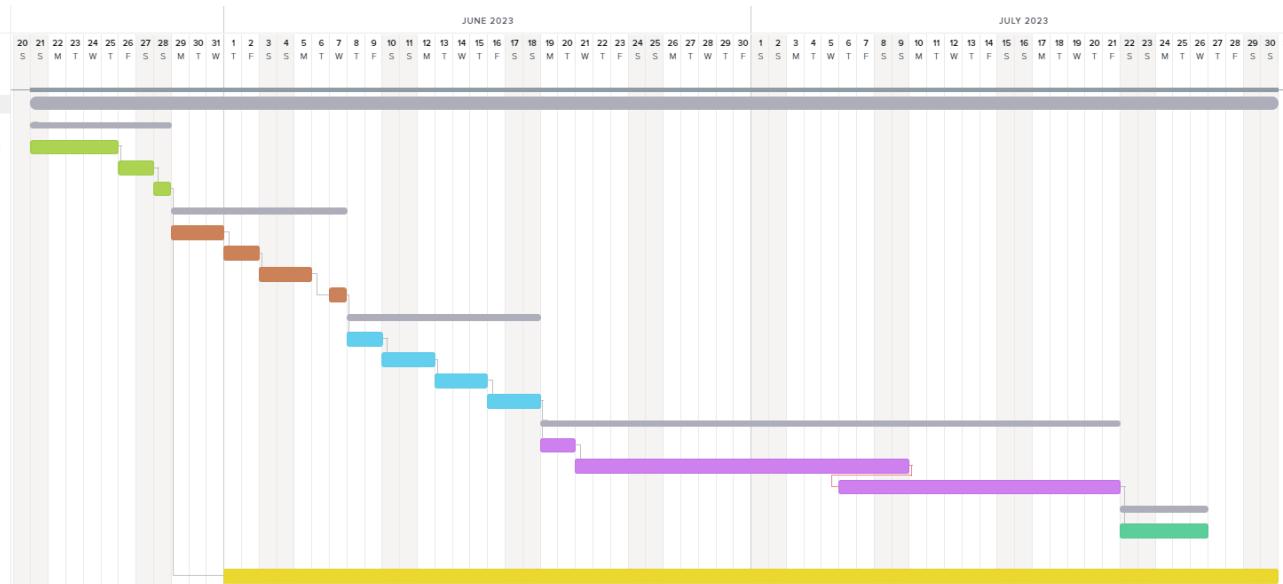


Figura 6: Diagrama de Gantt

3.3.3 Viabilidad operativa

En el contexto del desarrollo de la aplicación web, resultó fundamental contar con un nivel sólido de conocimientos en servicios web y habilidades en el uso de tecnologías como AJAX. Tras esta necesidad, se reconoció la importancia de adquirir la formación adecuada en servicios web.

Para cumplir con este requisito, se llevó a cabo una fase de formación específica en servicios web, donde el desarrollador se dedicó a adquirir los conocimientos necesarios. Durante este proceso, se emplearon diversos recursos en línea, tutoriales interactivos y cursos especializados para obtener información detallada sobre los fundamentos de los servicios web, los protocolos, los formatos de intercambio de datos y las mejores prácticas para su implementación.

Durante la etapa de formación, el desarrollador adquirió una comprensión profunda de los conceptos clave relacionados con los servicios web, incluyendo protocolos como REST. Además, el desarrollador se enfocó en aprender a interactuar con diversas APIs, comprendiendo sus características, protocolos de comunicación y estructuras de datos. Se exploraron las API

RESTful, las cuales ofrecen una forma estándar y eficiente de acceder y manipular recursos a través de protocolos HTTP.

3.3.4 Viabilidad empresarial

En cuanto al modelo de negocio, la aplicación no tiene fines de lucro por lo que será subvencionada por la Generalitat Valenciana. Se espera que la financiación necesaria para desarrollar, mantener y mejorar la aplicación provenga de fondos públicos. Estos fondos pueden asignarse a través de programas gubernamentales que respalden el desarrollo de proyectos tecnológicos o promuevan el acceso a la cultura y la educación. La finalidad de no buscar un beneficio económico directo es garantizar que la aplicación sea accesible y esté disponible para todos los usuarios sin importar su capacidad económica.

4. Especificación de requisitos

4.1 Requisitos Funcionales

En esta sección, se establecerán los requisitos del software con el fin de formalizar las capacidades de la aplicación y crear un esquema de su funcionamiento. A continuación, se presentarán los Requisitos Funcionales del sistema.

- Biblioteca: el administrador de una biblioteca podrá crear, leer y actualizar el perfil de la biblioteca. El lector podrá leer la información de una biblioteca.
- Lector: un lector podrá crear, leer y actualizar el perfil del lector. El administrador de biblioteca podrá leer la información de un lector.
- Libro: el administrador de una biblioteca podrá crear, leer, actualizar y eliminar los libros que están asociados a dicha biblioteca. El lector podrá leer los detalles de todos los libros de cualquier biblioteca.
- Préstamo: el lector podrá solicitar el préstamo de un libro y el administrador de la biblioteca que contiene dicho libro actualizará la información del préstamo cuando el lector recoja y devuelva el libro que ha sido prestado.
- Reserva: en caso de que no esté disponible en ese momento un libro, el lector podrá realizar una reserva en la biblioteca que el lector haya seleccionado. Cuando el libro vuelva a estar disponible, el lector que había realizado la reserva recibirá una notificación para recoger el mismo.
- Notificación: cuando el lector realiza un préstamo o una realiza un préstamo o una reserva de un libro que ya está disponible, el lector podrá leer un aviso en su cuenta de que el libro está pendiente de recogida.
- Reseña: los lectores podrán leer y escribir reseñas de cualquier libro.
- Evento: el administrador de una biblioteca podrá crear, leer, actualizar y eliminar los eventos que están asociados a dicha biblioteca. El lector podrá leer todos los eventos de cualquier biblioteca.

- Calendario: el administrador de una biblioteca podrá leer los eventos registrados por la biblioteca.
- Filtrado de libros y eventos: el lector podrá filtrar el tipo de libro o evento que quiere visualizar en la aplicación a partir de la selección de una serie de parámetros.
- Login: el administrador de una biblioteca y el lector para acceder al sistema tendrán que introducir el correo y la contraseña con los que se hayan registrado.
- Logout: el administrador de una biblioteca y el lector podrán cerrar la sesión y salir del sistema.

4.2 Requisitos No Funcionales

En esta sección, se establecerán las propiedades que debe poseer la aplicación. El propósito es determinar los requisitos específicos que describan el comportamiento necesario del software. A continuación, se detallarán los Requisitos No Funcionales del sistema.

- Usabilidad: el sistema debe ser fácil de usar e intuitivo para el usuario. El usuario solo con la visualización del sistema deberá saber cómo usarlo. Los mensajes de error y éxito deben ser claros y comprensibles para los usuarios, proporcionando retroalimentación adecuada.
- Compatibilidad e Interoperabilidad: la aplicación deberá ser compatible con cualquier tipo de navegador o dispositivo electrónico (PC, móvil, tablet) que utilice el usuario.
- Adaptabilidad: el sistema debe ser adaptable y permitir la incorporación de nuevas funcionalidades o la modificación de las existentes de manera fácil y rápida.
- Mantenibilidad: el código fuente del sistema debe estar bien estructurado, siguiendo buenas prácticas de programación y permitiendo una fácil comprensión y mantenimiento.

4.3 Diagramas UML

El **Lenguaje de Modelado Unificado** (UML) [27] es una herramienta para representar visualmente el comportamiento y la estructura de un sistema o proceso. Su uso es crucial para el desarrollo de un software, ya que permite que los errores relativos a la estructura, comportamiento u otros procesos de la aplicación software sean más fáciles de detectar en fases tempranas del diseño de un proyecto software.

4.3.1 Diagrama de clases

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que representan de manera clara la configuración de un sistema al describir sus clases, atributos, operaciones y las conexiones entre objetos de manera visual y precisa.

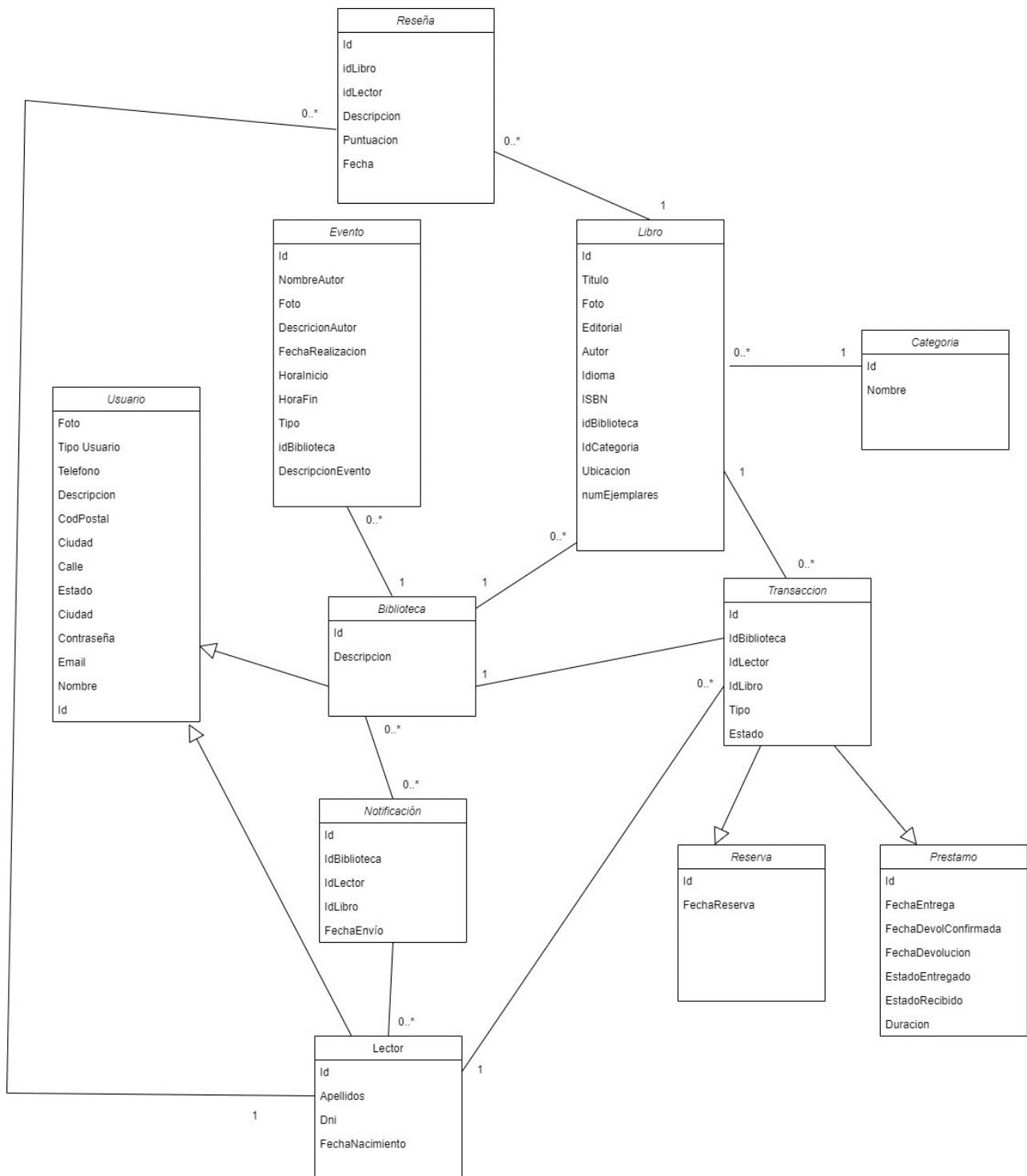


Figura 7: Diagrama de clases

A continuación, se va a realizar un análisis de cada uno de los atributos de cada tabla:

- **Libro:** En esta tabla se almacenan los libros que han sido registrados por una biblioteca.
 - Id: identificador del libro
 - Titulo: título del libro
 - Foto: cadena de texto que contiene una URL con el enlace a una imagen del libro
 - Editorial: editorial del libro
 - Autor: autor del libro
 - Idioma: idioma del libro (castellano, inglés o catalán)
 - ISBN: isbn del libro
 - idBiblioteca: identificador de la biblioteca a la que pertenece el libro
 - idCategoria: identificador a una categoría
 - Ubicación: ubicación donde se encuentra almacenado el libro
 - numEjemplares: número de ejemplares que existen del libro en la biblioteca
- **Categoría:** En esta tabla se almacenan las posibles categorías que puede tener un libro.
 - Id: identificador de la categoría
 - Nombre: nombre de la categoría (Humanidades, Ciencias sociales, Arte, Ficción, Romance, etc.).
- **Evento:** En esta tabla se almacenan los eventos que han sido registrados por una biblioteca.
 - Id: identificador del evento
 - NombreAutor: nombre completo del autor de un evento
 - Foto: cadena de texto que contiene una URL con el enlace a una imagen del evento.
 - DescripcionAutor: descripción del autor que va a dirigir un evento.
 - DescripcionEvento: descripción completa del evento
 - FechaRealizacion: día en que se va a realizar el evento
 - HoraInicio: hora en la que empezará el evento
 - HoraFin: hora en la que acabará el evento
 - Tipo: tipo de evento (Firma de libro, Taller escritura, Conferencia, Club de lectura, etc.).
 - idBiblioteca: identificador de la biblioteca que registra el evento
- **Transacción:** En esta tabla se almacena información que es heredada por la tabla préstamo o reserva.
 - Id: identificador de la transacción
 - idBiblioteca: identificador de la biblioteca que contiene el libro solicitado
 - idLector: identificador del lector que solicita el libro
 - idLibro: identificador del libro que es solicitado
 - Tipo: el tipo de transacción (reserva o préstamo)
- **Prestamo:** En esta tabla se almacenan todos los préstamos que se han realizado en el sistema.
 - Id: identificador del préstamo

- FechaEntrega: fecha en la que el libro fue entregado a un lector.
 - FechaDevolucion: fecha máxima en la que el lector debe devolver el libro.
 - FechaDevolConfirmada: fecha real en la que el lector devuelve el libro.
 - EstadoRecibido: estado del libro cuando fue entregado al lector.
 - EstadoEntregado: estado del libro cuando el lector lo devuelve a la biblioteca.
 - Duración: número de días que se permitirá el préstamo desde la FechaEntrega. Por defecto la duración se ha asignado a 21 días.
- **Reserva:** En esta tabla se almacenan todas las reservas de libros que están activas.
 - Id: identificador de la reserva
 - FechaReserva: fecha en la que un lector solicitó un libro que no estaba disponible, y por tanto se le asignó en la lista de espera con esta fecha.
- **Reseña:** En esta tabla se almacenan las reseñas que escriben los lectores sobre los distintos libros.
 - Id: identificador de una reseña
 - idLibro: identificador del libro del que se escribe la reseña
 - idLector: identificador del lector que ha escrito la reseña
 - Descripcion: descripción de la reseña
 - Puntuación: puntuación que un lector pone a un libro
 - Fecha: fecha en que se escribe la reseña
- **Notificacion:** Esta tabla almacena todas las notificaciones activas en el sistema.
 - Id: identificador de la notificación
 - idBiblioteca: identificador de la biblioteca que contiene el libro que debe ser recogido
 - idLector: identificador del lector que debe recoger un libro.
 - FechaEnvío: fecha desde la que el libro está disponible para su recogida.
- **Usuario:** Esta tabla contiene información que es heredada por la tabla lector o biblioteca.
 - Id: identificador del usuario
 - Nombre: nombre del usuario
 - Email: email del usuario
 - Foto: cadena de texto que contiene una URL con el enlace a una imagen del usuario
 - TipoUsuario: el tipo del usuario (lector o biblioteca)
 - Telefono: el teléfono del usuario
 - CodPostal: el código postal del usuario
 - Ciudad: la ciudad del usuario
 - Calle: la calle del usuario
 - Estado: el estado al que pertenece el usuario
 - Contraseña: la contraseña del usuario
- **Lector:** En esta tabla, se almacenan todos los lectores que han sido registrados en el sistema. Se establece que los lectores comparten aspectos comunes con los usuarios generales al compartir un identificador en la tabla Usuario. Aunque comparten información básica, los lectores también pueden tener atributos específicos que son los siguientes:

- Id: identificador del lector
 - Apellidos: apellidos del lector
 - Dni: dni del lector
 - FechaNacimiento: la fecha de nacimiento del lector
- **Biblioteca:** En esta tabla, se almacenan todas las bibliotecas que han sido registradas en el sistema. Una Biblioteca tendrá un campo identificador que deberá coincidir con uno de los identificadores de la tabla Usuario. Esto significa que las bibliotecas registradas comparten atributos comunes con los usuarios generales, pero también pueden tener atributos específicos relacionados con su función como bibliotecas.
 - Id: identificador de la biblioteca
 - Descripcion: contiene una pequeña descripción de la biblioteca.

Una vez presentada la descripción de las tablas y los atributos del sistema, se procederá a explicar cómo estas tablas están interconectadas.

- La tabla Usuario actúa como una superclase o clase padre de la cual se derivan las subclases Lector y Biblioteca. Esta relación de herencia implica que tanto los lectores como las bibliotecas comparten ciertos atributos y comportamientos definidos en la clase Usuario, pero también pueden tener atributos y comportamientos específicos que los diferencian. Esto implica que tanto un lector como una biblioteca son usuarios del sistema.
- La tabla Transacción es una superclase, mientras que la clase Reserva y Préstamo son subclases de Transacción. Esto implica que tanto una reserva como un préstamo son transacciones del sistema.
- La tabla Biblioteca mantiene una relación con las tablas Evento y Libro mediante una cardinalidad de Muchos a Muchos (N a N). Esto implica que una biblioteca puede estar asociada con numerosos eventos y libros. A su vez, tanto los libros como los eventos están exclusivamente vinculados a una sola biblioteca. Es importante destacar que, para cada biblioteca, no se permiten repeticiones de libros y eventos, garantizando así la singularidad de las asociaciones.
- La tabla Libro está relacionada con las tablas Reseña y Transacción con una cardinalidad de N a N, ya que con un libro se pueden hacer diversas transacciones (de préstamos o reservas) y también se permite la escritura de múltiples reseñas diversos lectores acerca de un libro. Un libro también está relacionado con la tabla Categoría con una cardinalidad de 1:1, lo que significa que cualquier libro deberá tener asociado una categoría.
- La tabla Lector está relacionada con las tablas Transacción y Notificacion con una cardinalidad de N a N, ya que un lector puede haber recibido múltiples notificaciones de diversas bibliotecas y puede solicitar diversas transacciones de libros al mismo tiempo.

4.3.2 Diagrama de contexto

El diagrama de contexto sirve para identificar cual es el entorno del sistema, es decir, los límites del sistema software a desarrollar.

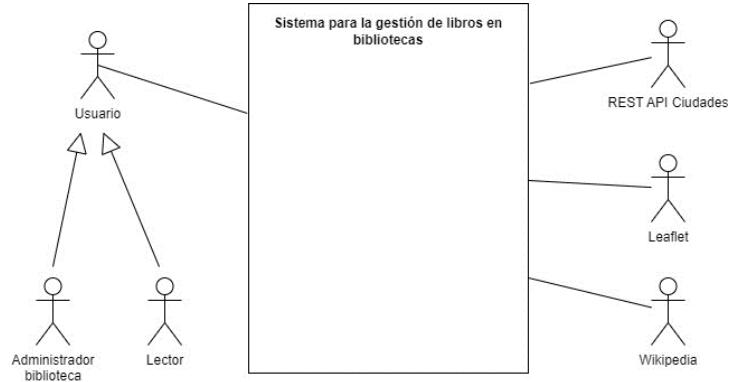


Figura 8: Diagrama de contexto

4.3.3 Diagrama de casos de uso

El diagrama de casos de uso es una herramienta gráfica que se utiliza en la ingeniería de software para representar las interacciones entre los usuarios y el sistema software.

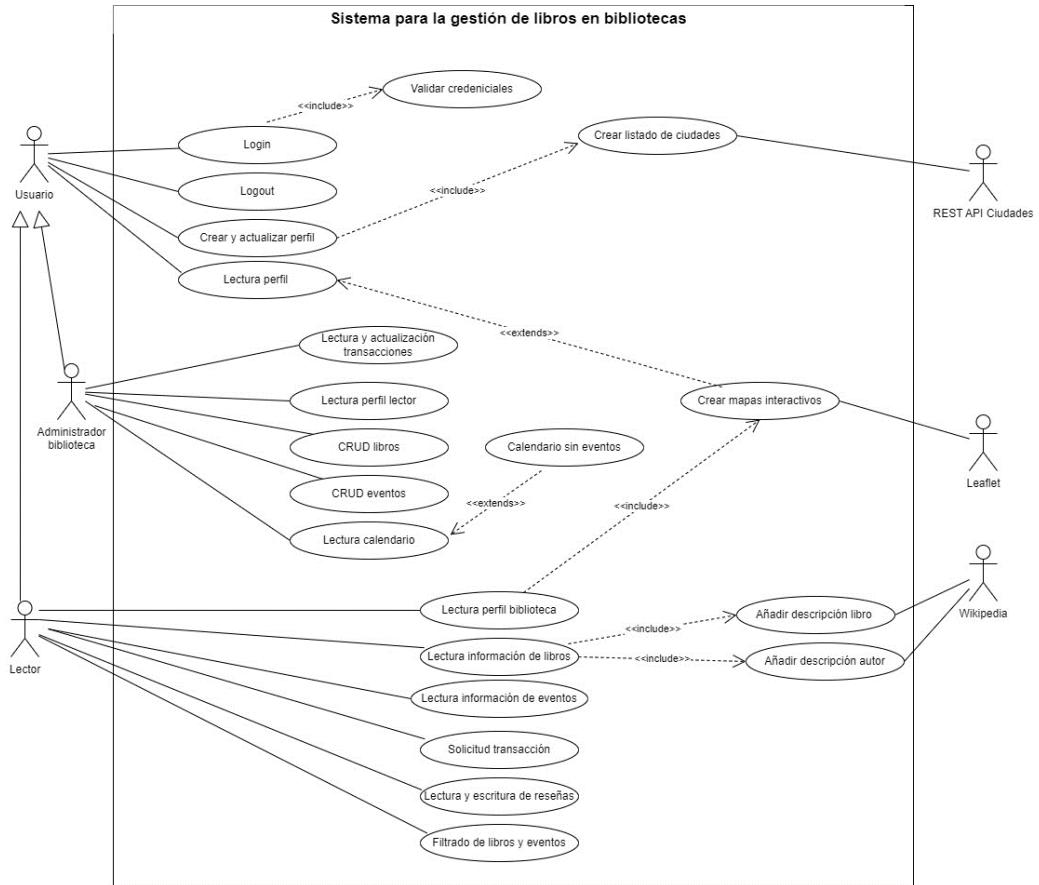


Figura 9: Diagrama de casos de uso

4.4 Descripción de casos de uso y Diagramas de Flujo

En este apartado se van a mostrar con más detalles los casos de uso de la Figura anterior. Además, para algunos casos de uso se incluirán diagramas de flujo para un mayor entendimiento del sistema.

En las siguientes 2 tablas se explicará los casos de uso asociados al “Login” y “Logout” del sistema. Se incluirá el diagrama de flujo del Login.

Caso de uso	Login
Actores	Usuario
Propósito	Autenticarse para poder acceder a la aplicación.
Resumen	Un usuario (administrador o lector) de la aplicación con su email y contraseña, puede identificarse para acceder al sistema
Precondiciones	Las credenciales introducidas son válidas
Postcondiciones	El usuario queda autenticado

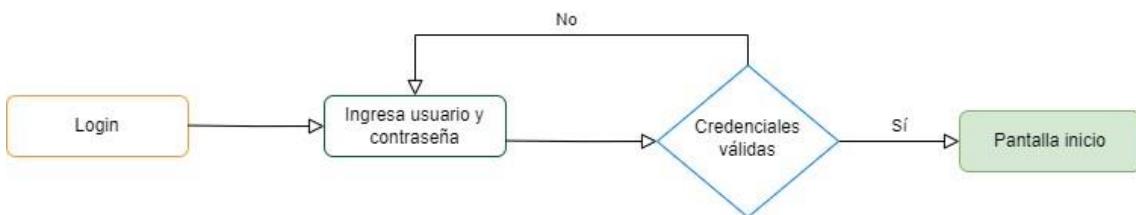


Figura 10: Diagrama de flujo Login

Caso de uso	Logout
Actores	Usuario
Propósito	Cerrar la sesión del usuario
Resumen	El usuario puede cerrar su sesión del sistema pulsando un ícono de la aplicación.
Precondiciones	El usuario ha iniciado sesión. El usuario pulsa “sí” al mensaje de confirmación que aparece.
Postcondiciones	El usuario sale del sistema

En las siguientes 3 tablas se explicarán los casos de uso asociados al Perfil del Usuario.

Caso de uso	Crear perfil
Actores	Usuario
Propósito	Registro en la aplicación para poder tener acceso a su contenido
Resumen	El usuario completa los campos para su registro
Precondiciones	Todos los campos rellenados por el usuario son válidos
Postcondiciones	El usuario ha sido registrado
Incluye	Una REST API de ciudades proporciona un listado de las ciudades que contiene el Estado seleccionado por el usuario

Caso de uso	Modificar perfil
Actores	Usuario
Propósito	Modificar la información del perfil
Resumen	El usuario completa los campos que quiere modificar
Precondiciones	El usuario ha iniciado sesión. Todos los campos modificados por el usuario son válidos
Postcondiciones	La información del perfil del usuario ha sido modificada
Incluye	Una REST API de ciudades proporciona un listado de las ciudades que contiene el Estado seleccionado por el usuario

Caso de uso	Lectura perfil
Actores	Usuario
Propósito	Ver información del perfil
Resumen	El usuario puede acceder a aquella información que los demás usuarios ven sobre él
Precondiciones	El usuario ha iniciado sesión.
Postcondiciones	Información del perfil mostrada
Extiende	En caso de que el usuario sea el administrador de una biblioteca, Leaflet mostrará con mapas interactivos la ubicación de la biblioteca.

En las siguientes 2 tablas se explicarán los casos de uso asociados a la gestión de transacciones (reservas y préstamos) de una biblioteca.

Caso de uso	Lectura transacciones
Actores	Administrador biblioteca
Propósito	Ver el listado de transacciones (reservas/préstamos)
Resumen	El administrador de una biblioteca puede ver las transacciones (préstamos y reservas) que se han realizado con sus libros
Precondiciones	El administrador ha iniciado sesión.
Postcondiciones	Transacciones listadas

Caso de uso	Actualización transacciones
Actores	Administrador biblioteca
Propósito	Actualizar el estado de una transacción
Resumen	El administrador de una biblioteca puede actualizar el estado de las transacciones pasando por 4 estados: “en espera”, “pendiente de recogida”, “pendiente devolución” o “devuelto”
Precondiciones	El administrador ha iniciado sesión. Deben existir transacciones en algunos de los estados: “en espera”, “pendiente de recogida” o “pendiente devolución”
Postcondiciones	Estado de la transacción actualizado

En las próximas 4 tablas se explicarán los casos de uso asociados al CRUD (Create, Read, Update and Delete) y se mostrara el diagrama de flujo de “CRUD libro”.

Caso de uso	Crear libro
Actores	Administrador biblioteca
Propósito	Añadir nuevo libro a la biblioteca
Resumen	El administrador de una biblioteca puede crear un nuevo libro rellenando un formulario
Precondiciones	El administrador ha iniciado sesión. Todos los campos del nuevo libro son válidos.
Postcondiciones	El libro ha sido creado

Caso de uso	Lectura libro
Actores	Administrador biblioteca
Propósito	Listados de libros de la biblioteca
Resumen	El administrador de una biblioteca puede visualizar la información de uno de sus libros
Precondiciones	El administrador ha iniciado sesión. Existe algún libro en la biblioteca.
Postcondiciones	Libros listados

Caso de uso	Modificar libro
Actores	Administrador biblioteca
Propósito	Actualizar un libro de una biblioteca
Resumen	El administrador de una biblioteca puede actualizar el contenido de un libro rellenando un formulario
Precondiciones	El administrador ha iniciado sesión. Existe algún libro en la biblioteca. Todos los campos del nuevo libro son válidos.
Postcondiciones	El libro ha sido actualizado

Caso de uso	Eliminar libro
Actores	Administrador biblioteca
Propósito	Eliminar un libro de una biblioteca
Resumen	El administrador de una biblioteca puede eliminar un libro pulsando un botón.
Precondiciones	El administrador ha iniciado sesión. Existe algún libro en la biblioteca. El administrador pulsa “sí” en el mensaje de confirmación.
Postcondiciones	El libro ha sido eliminado

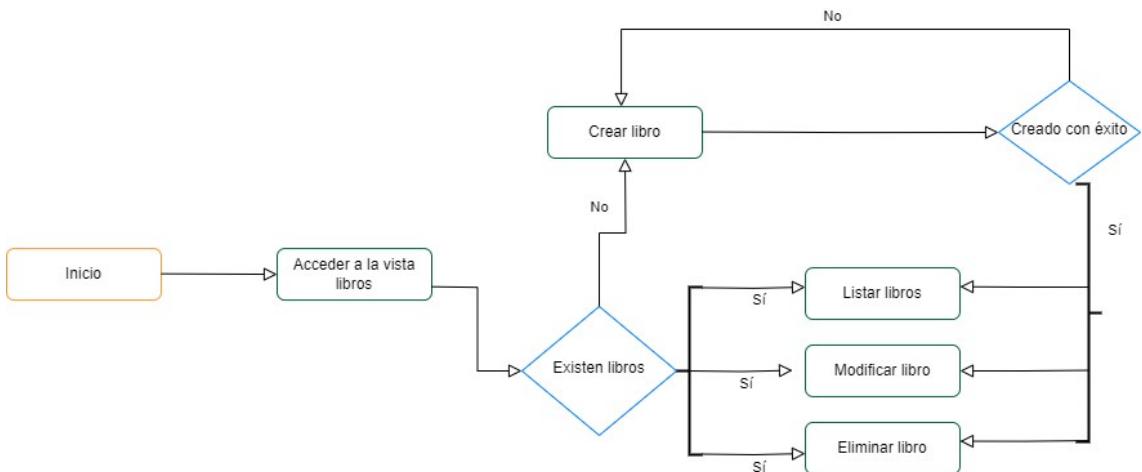


Figura 11: Diagrama de flujo CRUD libros

En las próximas cuatro tablas se detallarán los casos de uso del CRUD para eventos y se mostrará su diagrama de flujo.

Caso de uso	Crear evento
Actores	Administrador biblioteca
Propósito	Añadir nuevo evento a la biblioteca
Resumen	El administrador de una biblioteca puede crear un nuevo evento rellenando un formulario
Precondiciones	El administrador ha iniciado sesión. Todos los campos del nuevo evento son válidos.
Postcondiciones	El evento ha sido creado

Caso de uso	Lectura evento
Actores	Administrador biblioteca
Propósito	Ver el listado de eventos de la biblioteca
Resumen	El administrador de una biblioteca puede visualizar la información de uno de sus eventos.
Precondiciones	El administrador ha iniciado sesión. Existe algún evento en la biblioteca.
Postcondiciones	Eventos listados

Caso de uso	Modificar evento
Actores	Administrador biblioteca
Propósito	Actualizar un evento de una biblioteca
Resumen	El administrador de una biblioteca puede actualizar el contenido de un evento rellenando un formulario
Precondiciones	El administrador ha iniciado sesión. Existe algún evento en la biblioteca. Todos los campos del nuevo evento son válidos.
Postcondiciones	El evento ha sido actualizado

Caso de uso	Eliminar evento
Actores	Administrador biblioteca
Propósito	Eliminar un evento de una biblioteca
Resumen	El administrador de una biblioteca puede eliminar un evento pulsando un botón.
Precondiciones	El administrador ha iniciado sesión. Existe algún evento en la biblioteca. El administrador pulsa "sí" en el mensaje de confirmación.
Postcondiciones	El evento ha sido eliminado

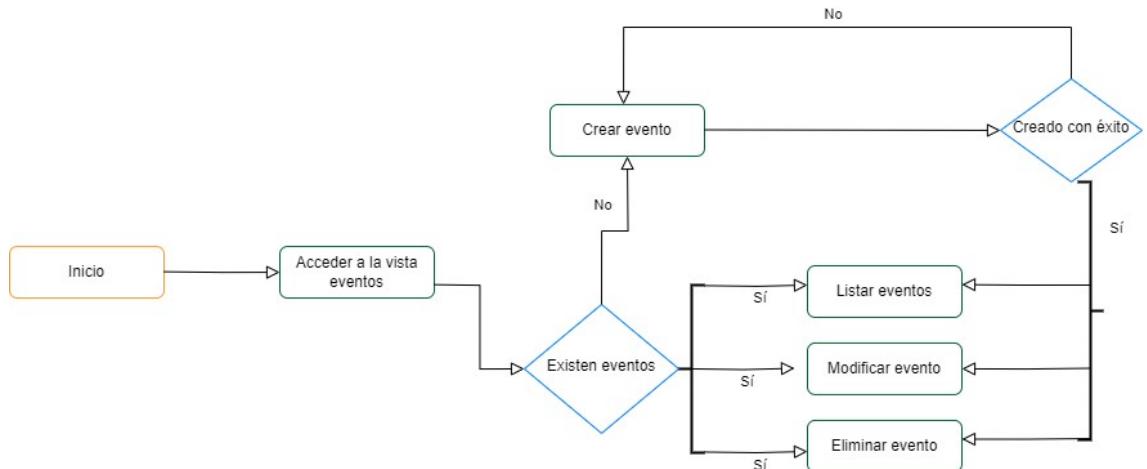


Figura 12: Diagrama de flujo CRUD eventos

En la siguiente tabla se muestra el caso de uso para la lectura del calendario desde la sesión de un administrador de biblioteca.

Caso de uso	Lectura calendario
Actores	Administrador biblioteca
Propósito	Ver la información de un calendario
Resumen	El administrador de una biblioteca puede visualizar la información de su calendario. El calendario incluirá los eventos literarios en las fechas que han sido programados.
Precondiciones	El administrador ha iniciado sesión.
Postcondiciones	Información del calendario mostrada.
Extiende	Puede ocurrir que la biblioteca no haya registrado eventos y por tanto, el calendario no muestre información relevante.

En las próximas 2 tablas se definirán los casos de uso asociados a la lectura de los perfiles del otro tipo de usuario, es decir, si el usuario es una “biblioteca” podrá ver el perfil del tipo de usuario “lector” y a la inversa.

Caso de uso	Lectura perfil biblioteca
Actores	Lector
Propósito	Ver el perfil de una biblioteca
Resumen	Un lector puede ver la información de cualquiera de las bibliotecas.
Precondiciones	El lector ha iniciado sesión. Existe alguna biblioteca registrada en el sistema.
Postcondiciones	Información de la biblioteca mostrada
Incluye	Al visualizar la información de la biblioteca se incluye un mapa interactivo creado a partir del sistema Leaflet. Este mapa se incluye para identificar la ubicación exacta de la biblioteca.

Caso de uso	Lectura perfil lector
Actores	Administrador biblioteca
Propósito	Ver el perfil de un lector
Resumen	El administrador puede ver la información de contacto de los lectores que han realizado transacciones con sus libros.
Precondiciones	El administrador ha iniciado sesión. El lector ha realizado una transacción con alguno de los libros de la biblioteca.
Postcondiciones	Información del lector mostrada

En las próximas 2 tablas se definirán los casos de uso asociados a la lectura de los detalles del libro y su solicitud. Para el caso de la solicitud del libro, se mostrará su diagrama de flujo.

Caso de uso	Lectura información libro
Actores	Lector
Propósito	Ver información de un libro
Resumen	Un lector puede ver la información de cualquiera de los libros.
Precondiciones	El lector ha iniciado sesión. Existe algún libro creado en el sistema.
Postcondiciones	Información del libro mostrada
Incluye	Al visualizar la información del libro se incluyen dos secciones que se obtienen tras el uso de la API de Wikipedia, estas son la descripción del libro y la descripción del autor.

Caso de uso	Solicitud de un libro
Actores	Lector
Propósito	Solicitar un libro
Resumen	El lector se encuentra en la página de detalles del libro y presiona un botón para reservar o realizar un préstamo del libro. En caso de que el libro no esté disponible, solo se podrá “reservar”, en caso contrario se podrá solicitar el “préstamo”.
Precondiciones	El lector ha iniciado sesión. Existe algún libro creado en el sistema.
Postcondiciones	Se ha registrado la reserva o el préstamo

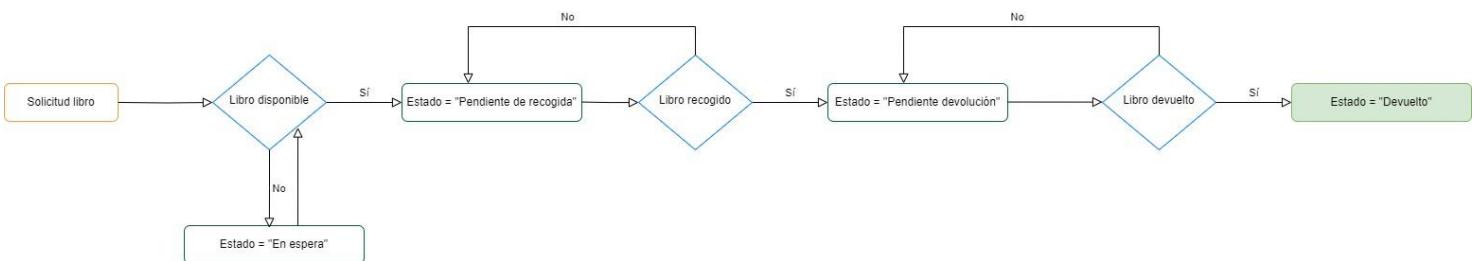


Figura 13: Diagrama de flujo asignación libro

En las próximas 2 tablas se definirán los casos de uso asociados a la lectura y creación de las Reseñas.

Caso de uso	Lectura reseñas
Actores	Lector
Propósito	Ver el listado de reseñas
Resumen	El lector al acceder a la información de un libro podrá ver las reseñas que el resto de los lectores ha escrito acerca del libro.
Precondiciones	El lector ha iniciado sesión. Existe alguna reseña para ese libro.
Postcondiciones	Reseñas listadas

Caso de uso	Crear reseña
Actores	Lector
Propósito	Escribir una nueva reseña
Resumen	El lector al acceder a la información de un libro podrá añadir una nueva reseña completando un formulario.
Precondiciones	El lector ha iniciado sesión. Campos del formulario válidos.
Postcondiciones	Reseña creada.

En las siguientes 2 tablas se definirán los casos de uso asociados a los filtros de la tabla Libro y Evento.

Caso de uso	Filtrado de libros
Actores	Lector
Propósito	Filtrar libros según características
Resumen	El lector al acceder al catálogo de los libros podrá filtrar los que quiere visualizar a partir de la selección de una serie de parámetros.
Precondiciones	El lector ha iniciado sesión. Existen libros en el sistema.
Postcondiciones	Libros filtrados

Caso de uso	Filtrado de eventos
Actores	Lector
Propósito	Filtrar eventos según características
Resumen	El lector al acceder al catálogo de los eventos podrá filtrar los que quiere visualizar a partir de la selección de una serie de parámetros.
Precondiciones	El lector ha iniciado sesión. Existen eventos en el sistema.
Postcondiciones	Eventos filtrados

5. Diseño

En el proceso de desarrollo de software, el diseño se destaca como una fase crítica. Durante esta etapa, se busca encontrar una solución que cumpla con los requisitos funcionales y no funcionales previamente definidos en la fase de análisis.

El éxito del software depende en gran medida de la calidad del diseño, ya que un error en esta etapa puede tener un impacto significativo en todo el proyecto, llevando a revisiones continuas y cambios constantes. Esta fase comprende varios componentes:

- **Diseño de datos:** En esta subetapa, se establece la estructura de la base de datos y las relaciones entre las tablas que la componen. Para este proyecto, se emplea SQL Server como herramienta para modelar la base de datos.
- **Diseño arquitectónico:** Aquí se describe la comunicación entre los componentes del software, así como con otros sistemas y usuarios. En el proyecto, se ha adoptado la arquitectura Modelo-Vista-Controlador (MVC), junto con el patrón arquitectónico DAO.
- **Diseño de la interfaz:** En esta subetapa, se crean prototipos que representan la interfaz visual de la aplicación, proporcionando una vista previa de su aspecto final. Para este proyecto, se utiliza la herramienta Figma para definir los prototipos de las vistas de la aplicación.

5.1 Modelo de la base de datos

A continuación, se mostrará un diagrama que representará el modelo de datos de la aplicación. Este diagrama se genera automáticamente desde la herramienta de SQL Server una vez que todas las clases han sido creadas en la base de datos, con sus correspondientes referencias entre las tablas.

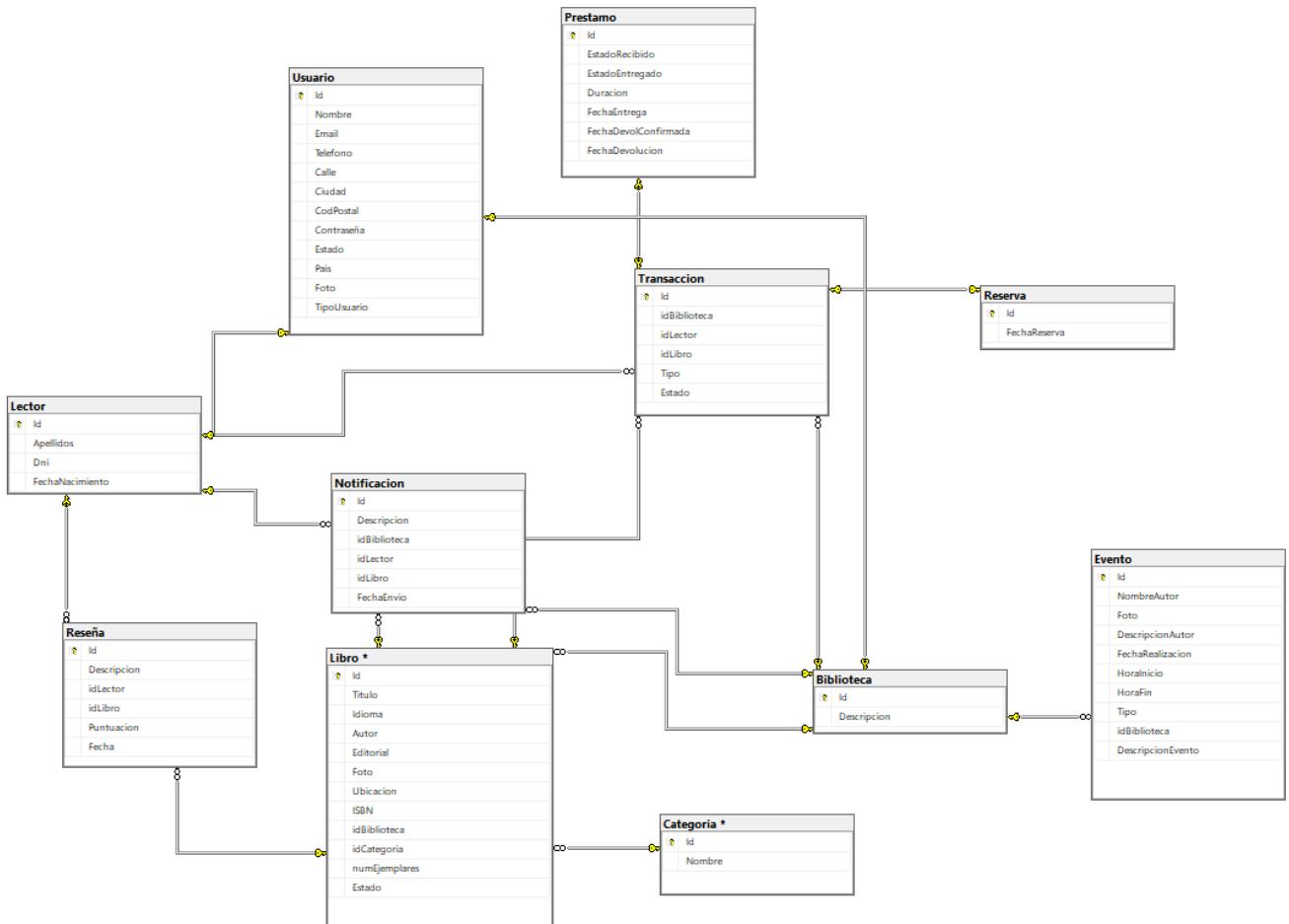


Figura 14: Diagrama modelo de base de datos

5.2 Arquitectura Software

La arquitectura de software proporciona una visión global de cómo funcionan de manera conjunta los diferentes componentes de un sistema. Posteriormente, se van a explicar los dos patrones arquitectónicos que se han utilizado para estructurar cada una de las partes de la aplicación.

5.2.1 Patrón arquitectónico MVC

El sistema se ha estructurado siguiendo la arquitectura del Modelo Vista y Controlador (MVC). El patrón MVC [24] es una arquitectura de diseño software que se utiliza en el desarrollo de aplicaciones software para separar la lógica de negocio con la visualización de los datos.

Para lograr este patrón arquitectónico el código de la aplicación se ha tenido que estructurar en 3 partes:

- **Modelo**: es la capa donde se trabaja directamente con los datos, ya que contiene mecanismos para el acceso y manipulación de la información. Los datos están alojados en una base de datos y en cada modelo se alojan las funciones para acceder a cada una de las tablas que contiene la base de datos.

- **Vista:** permite la visualización de las interfaces de usuario, y por tanto renderizar los estados de la aplicación a código HTML. Una vista es la representación visual de los datos, así como un formulario, un grid, una caja de texto, etc.
- **Controlador:** responde a las peticiones del usuario, así como, visualizar un elemento, realizar un registro de un libro, buscar un libro, etc. El controlador se encarga de traducir lo que le llega del usuario, al mismo tiempo que se lo comunica al Modelo y a la Vista. Por una parte, a los Modelos les solicita datos o les realiza una petición para la actualización de los datos. Por otra parte, a las Vistas les solicita la salida después de haberse realizado las correspondientes operaciones. Por lo que se puede concluir que el controlador es el intermediario entre el Modelo y la Vista.

A continuación, se muestra un diagrama con la interacción entre las distintas capas del patrón arquitectónico previamente explicado.

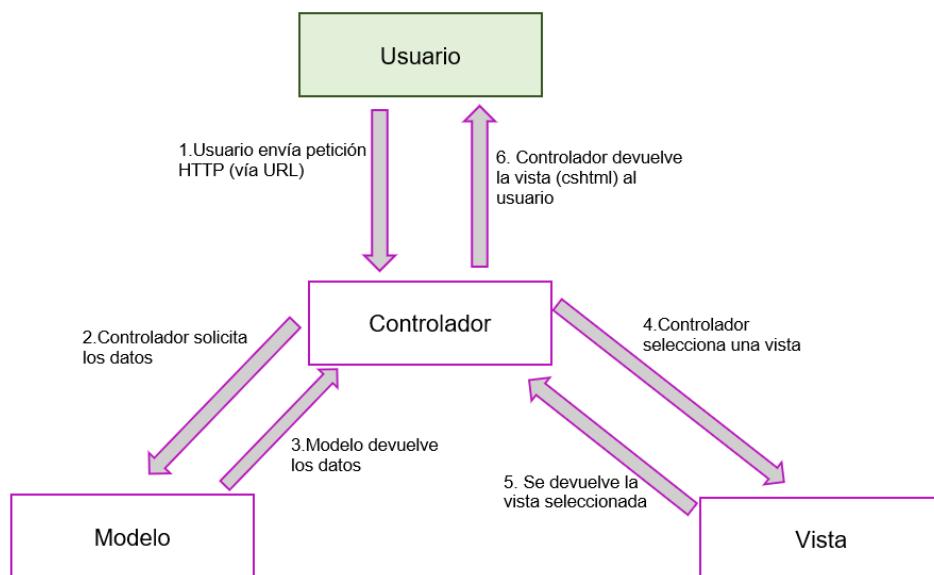


Figura 15: Patrón Modelo-Vista-Controlador

5.2.2 Patrón DAO

El patrón DAO (Data Acces Object) [23] pretende mantener una separación entre la lógica de negocio y la lógica de acceso a datos. El DAO proporciona los métodos necesarios para insertar, actualizar, borrar y consultar la información. Por otro lado, la capa de negocio (Modelo) se centra únicamente en la definición de reglas que rigen el funcionamiento de la aplicación y utiliza el DAO para interactuar con la fuente de datos.

Este patrón ofrece varios beneficios:

- **Reutilización de código:** Dado que el DAO proporciona una interfaz clara y coherente para acceder a los datos, este componente puede reutilizarse en diferentes partes de la aplicación, lo que ahorra tiempo en el desarrollo.
- **Mantenimiento:** La separación de la lógica de negocio y la lógica de acceso a datos mejora la legibilidad y la organización del código, lo que hace que el sistema sea más fácil de entender y mantener.

5.3 Diseño con prototipado

Esta sección contiene una gran variedad de los mockups diseñados para representar la aplicación. El diseño del prototipado sirve para crear una versión temprana de la interfaz y garantizar así que se cumple con las necesidades iniciales.

5.3.1 Login

La Figura 16 representa la página de inicio se sesión de un usuario en la aplicación.

El prototipo de la página de login muestra el siguiente diseño:

- Logo APP:** Un cuadro vacío para el logo de la aplicación.
- Bienvenido a BiblioConnect:** Texto de bienvenida centralizado.
- Campos de ingreso:** Dos cuadros para "Email" y "Contraseña".
- Botones:** Dos botones redondeados: "Ingresar" y "Regístrate".
- Mensaje de ayuda:** Una línea de texto que dice "¿Eres una Biblioteca yquieres darte de alta? [Dar de alta Biblioteca](#)".

Figura 16: Prototipo Login

5.3.2 Registro lector

A través del uso de prototipos se hizo un primer diseño del formulario de registro de un lector con campos como (dni, teléfono, email, fecha de nacimiento, etc.).

El prototipo del formulario de registro de lector muestra el siguiente diseño:

- Bienvenido a BiblioConnect:** Texto de bienvenida centralizado.
- Campos de información:** Sección con campos para Nombre, Apellidos, Dni, Email, Teléfono, Fecha nacimiento, Estado, Ciudad, Calle, Código postal, Contraseña (con placeholder "Introduzca tu contraseña") y Repita contraseña (con placeholder "Introduzca tu contraseña").
- Botones y campos adicionales:** Botón "Insertar Foto" y botón "Seleccionar" para cargar imágenes.
- Botón final:** Botón "Regístrate" en el pie de la página.

Figura 17: Prototipo Registro lector

5.3.3 Listado libros y eventos

La aplicación al contar con dos sesiones de usuario, una para el administrador y otra para el lector, ha tenido que adaptarse a diferentes interfaces de vistas. Para el caso del administrador de una biblioteca, solo podrá visualizar y editar los libros que el mismo haya subido. Por otro lado, el lector podrá ver todos los libros y eventos registrados por cualquier tipo de biblioteca.

En la Figura 18 se muestra cómo se vería el listado de libros desde la sesión de administrador de una biblioteca.



El prototipo muestra una lista de libros con los siguientes encabezados: Imagen, Título, Autor, ISBN, Editorial, Categoría, Idioma, Ejemplares, Ubicación y Acciones. La sección de Acciones incluye iconos para editar y borrar. Hay dos filas de datos en la tabla.

Imagen	Título	Autor	ISBN	Editorial	Categoría	Idioma	Ejemplares	Ubicación	Acciones
34x32									34x32 34x32
34x32									34x32 34x32

Figura 18: Prototipo Listado libros desde administrador

Para el caso de eventos sería la misma estructura que la anterior, pero cambiando alguno de los nombres de columna.



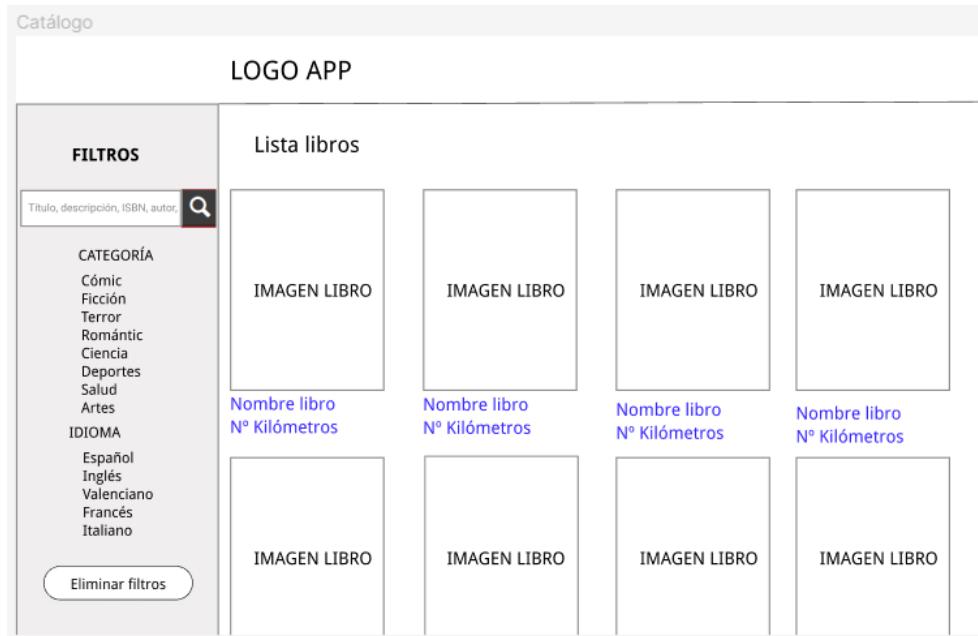
El prototipo muestra una lista de eventos con los siguientes encabezados: Imagen, Autor, Tipo, Fecha realización, Hora inicio, Hora fin y Acciones. La sección de Acciones incluye iconos para editar y borrar. Hay dos filas de datos en la tabla.

Imagen	Autor	Tipo	Fecha realización	Hora inicio	Hora fin	Acciones
34x32						34x32 34x32
34x32						34x32 34x32

Figura 19: Prototipo Listado eventos desde administrador

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

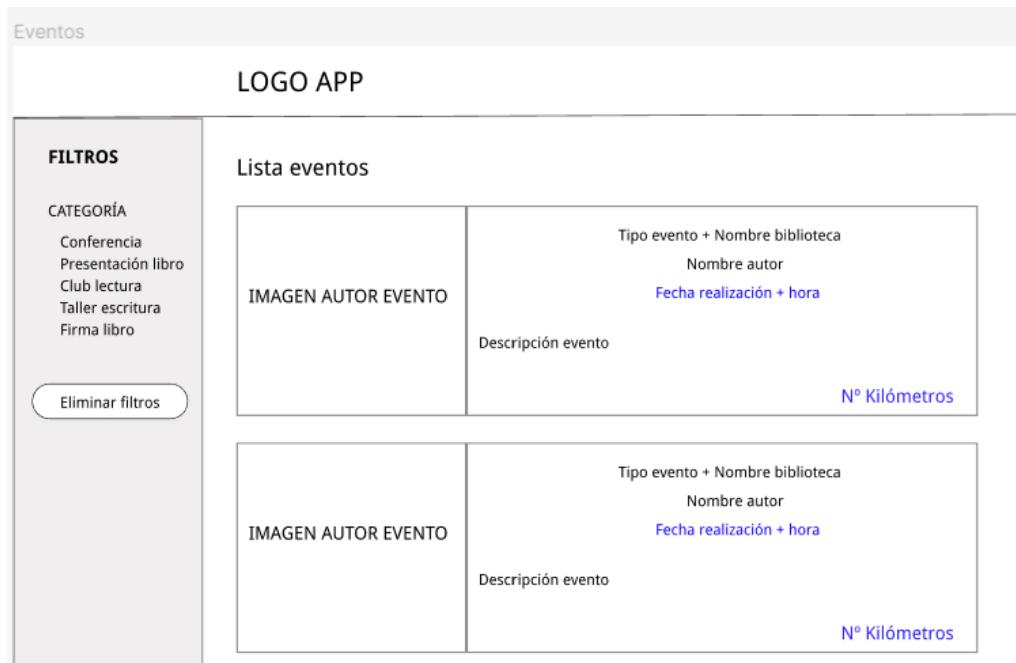
A continuación, se visualizará el listado de libros y eventos desde la sesión de un lector. La siguiente Figura representa el listado de todos los libros que han sido registrados por alguna de las bibliotecas.



Este prototipo muestra la interfaz de usuario para el catálogo de libros. En la parte superior izquierda, se encuentra un menú lateral titulado "Catálogo" con el logo de la aplicación "LOGO APP". A la derecha, se titula "Lista libros". El panel de filtros incluye un campo de búsqueda "Título, descripción, ISBN, autor," con un icono de lupa, y sección para "CATEGORÍA" (Cómics, Ficción, Terror, Romántic, Ciencia, Deportes, Salud, Artes) y "IDIOMA" (Español, Inglés, Valenciano, Francés, Italiano). Un botón "Eliminar filtros" se encuentra en la parte inferior del panel de filtros. La lista principal muestra ocho tarjetas de libro, cada una con un cuadro vacío para "IMAGEN LIBRO" y datos en azul: "Nombre libro" y "Nº Kilómetros".

Figura 20: Prototipo Catálogo libros

La Figura 21, muestra el listado de eventos que han sido programados por las distintas bibliotecas.



Este prototipo muestra la interfaz de usuario para el catálogo de eventos. Se titula "Eventos" en la parte superior izquierda con el logo "LOGO APP". A la derecha, se titula "Lista eventos". El panel de filtros incluye una sección para "CATEGORÍA" (Conferencia, Presentación libro, Club lectura, Taller escritura, Firma libro) y un botón "Eliminar filtros". La lista principal muestra dos tarjetas de evento, cada una con un cuadro vacío para "IMAGEN AUTOR EVENTO" y datos en azul: "Tipo evento + Nombre biblioteca", "Nombre autor", "Fecha realización + hora", "Descripción evento" y "Nº Kilómetros".

Figura 21: Prototipo Catálogo eventos

5.3.4 Modales

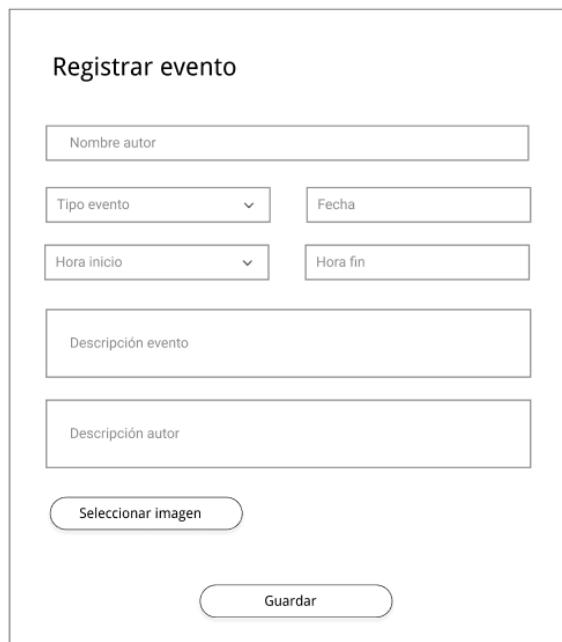
La aplicación cuenta con una gran variedad de ventanas modales, pero para simplificar, solo se mostrarán algunas de ellas. Cuando se hace clic en un botón dentro del sistema, se abre un formulario que permite realizar diversas acciones correspondientes a la funcionalidad del sistema.

En la Figura 22 y 23, se puede apreciar que estas ventanas modales constan de campos de entrada de texto, calendarios para seleccionar fechas, relojes para elegir la hora y menús desplegables para seleccionar una o varias opciones. Además, se incluye un botón principal para enviar el formulario.



Este es un prototipo modal para registrar un libro. La interfaz tiene un encabezado "Registrar libro". Los campos de entrada están dispuestos en dos filas: la primera fila contiene "Título" y "Nombre autor"; la segunda fila contiene "ISBN" y "Editorial". Ambos campos tienen un cuadro desplegable para seleccionar. La tercera fila contiene "Categoría" y "Idioma", cada uno con su respectivo cuadro desplegable. La cuarta fila contiene "Número ejemplares" y "Ubicación". Abajo de los campos hay un botón "Seleccionar imagen" y un botón "Guardar" en el centro.

Figura 22: Prototipo Modal “registro libro”



Este es un prototipo modal para registrar un evento. La interfaz tiene un encabezado "Registrar evento". Los campos de entrada están dispuestos en tres filas: la primera fila contiene "Nombre autor" y "Fecha"; la segunda fila contiene "Tipo evento" (cuadro desplegable) y "Hora inicio" (cuadro desplegable); la tercera fila contiene "Hora fin". Abajo de estos campos hay un cuadro grande para "Descripción evento". La cuarta fila contiene un cuadro para "Descripción autor". Abajo de los campos hay un botón "Seleccionar imagen" y un botón "Guardar" en el centro.

Figura 23: Prototipo Modal “registro evento”

5.3.5 Préstamos

En el sistema actual hay una sección de préstamos según el tipo de usuario que ha iniciado sesión se visualizará una vista u otra. Para identificar en qué situación está un lector con un libro se ha asignado el concepto de “estado”. El estado de un préstamo tiene 4 posibilidades:

1. **En espera:** el lector ha reservado un libro que no está disponible, pero en el momento que el lector que dispone del libro lo devuelva el estado del libro del lector en espera pasará a “pendiente de recogida” de manera automática, al mismo tiempo en que el lector será notificado para su recogida.
2. **Pendiente de recogida:** a este estado se puede llegar de dos formas, por un lado, si un libro está disponible el lector que hace click sobre el botón de “préstamo” será directamente asignado con este estado para su posterior recogida. Por otro lado, puede ser que el libro no estuviera disponible por lo que en el momento en que el lector que dispone del libro lo devuelva el estado del libro del lector en espera pasará automáticamente a este estado.
3. **Pendiente devolución:** este estado espera a que el lector recoja físicamente el libro en la biblioteca, en ese momento el administrador de la biblioteca deberá confirmar un formulario con la información del lector acreditando que el libro ha sido recogido por el lector.
4. **Devuelto:** este último estado se asigna cuando el lector que había obtenido un libro devuelve el libro a dicha biblioteca, en ese momento el administrador de la biblioteca deberá confirmar un formulario con la información del lector que había realizado el préstamo del libro acreditando que el libro ha sido devuelto por el lector.

Para el caso de un administrador, se mostrarán los préstamos o reservas que los lectores han realizado con los libros de la propia biblioteca. El administrador deberá actualizar el estado de un libro cuando se realice una acción concreta, así como devolución o recogida de un libro por un lector.

Estado	Título	Nombre	Apellidos	Dni	Fecha devolución	Fecha confirmada	Acciones

Figura 24: Prototipo Listado préstamos desde administrador

Por otro lado, un lector podrá ver el estado de todos sus préstamos.

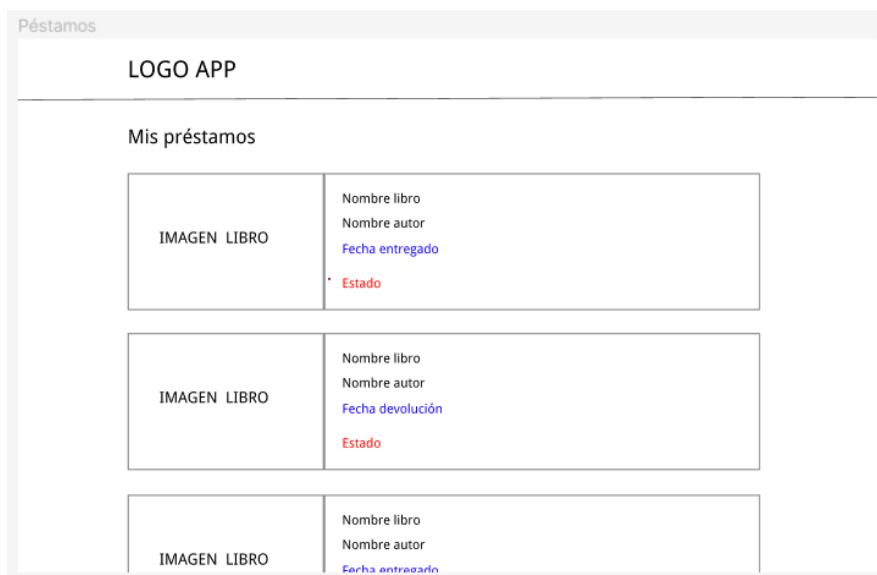


Figura 25: Prototipo Listado Préstamos desde lector

5.3.6 Menús

En las siguientes figuras se muestran los menús del sistema. El menú izquierdo es accesible desde una sesión de administrador de biblioteca, mientras que el derecho es solo para los lectores.

El menú de biblioteca cuenta con iconos para cada opción de vista, a demás de un nombre descriptivo. El menú lector cuenta con nombres descriptivos, a excepción de tres de los enlaces “Notificaciones”, “Perfil” y “Logout” que solo contienen icono.

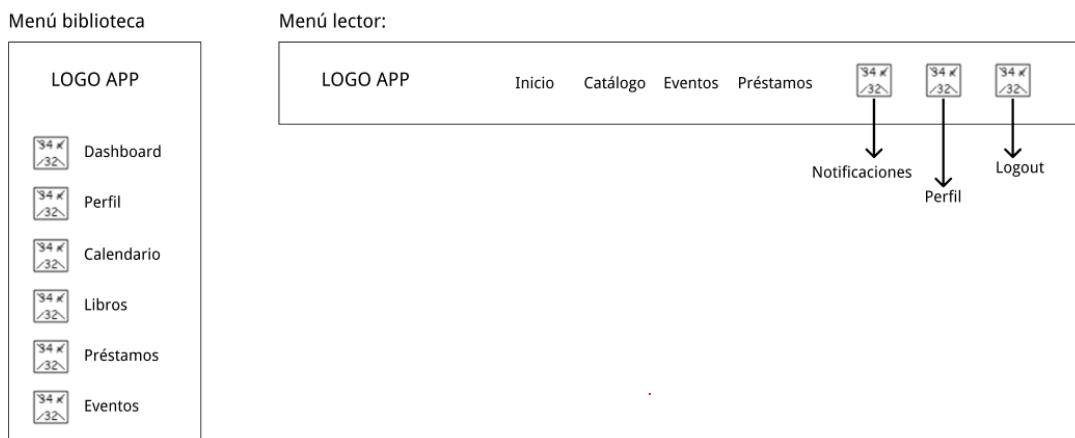


Figura 26: Prototipos menú

5.4 Estilo

Este apartado establece las pautas de diseño que se han aplicado en lo que respecta a la estética visual de la aplicación.

5.4.1 Logo

El logo del sistema está formado por un libro abierto con el icono de la ubicación. El libro representa claramente la biblioteca, mientras que el icono de ubicación simboliza la geolocalización y la capacidad del sistema por indicar los libros más cercanos al lector.

El nombre de la aplicación “BiblioConnect” refleja una de las funciones principales del sistema, que es conectar y centralizar las bibliotecas de la Comunidad Valenciana en una única aplicación.



BiblioConnect

Figura 27: Logo BiblioConnect

5.4.2 Tipografía

Para toda la aplicación se ha utilizado “Open Sans”, y para el nombre de la aplicación en el menú “EB Garamond”.

Open Sans

Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk,
Ll, Mm, Nn, Ññ, Oo, Pp, Qq, Rr, Ss, Tt,
Uu, Vv, Ww, Xx, Yy, Zz

EB Garamond

Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk,
Ll, Mm, Nn, Ññ, Oo, Pp, Qq, Rr, Ss, Tt,
Uu, Vv, Ww, Xx, Yy, Zz

Figura 28: Tipografía aplicación

5.4.3 Colores

Los colores principales de la aplicación son el azul y el morado oscuro. Ambos colores se han utilizado para diferenciar los dos tipos de sesiones de los usuarios; para el caso del administrador de una biblioteca se ha utilizado el azul y para los lectores el morado.

El naranja se ha utilizado para tratar toda la funcionalidad relacionada con las reseñas, así como el modal y los botones. El color el blanco que se ha utilizado para el texto o iconos internos a los botones y el negro para el texto de toda la aplicación.

Por un lado, cuando un libro está a la espera o pendiente (recogida, devolución) aparece una etiqueta o texto de color rojo. En caso de que el libro se haya devuelto la etiqueta aparecerá de color verde. El verde también se ha utilizado para el enlace directo al perfil de una biblioteca.



Figura 29: Paleta de colores

6. Tecnologías utilizadas

En esta sección, se explicarán las tecnologías empleadas en el desarrollo del proyecto. Para un mayor entendimiento del ámbito de cada una de ellas, se ha dividido en dos áreas el frontend (desarrollo del lado del cliente) y el backend (desarrollo del lado del servidor).

6.1 Frontend

Frontend se refiere a la parte software con la que los usuarios interactúan directamente, desde su estructura hasta sus estilos. Como pueden ser la definición de colores, tipografías, posición de los elementos, formas de los componentes, etc.

6.1.1 HTML



Figura 30: Logo HTML

HTML [1] o Lenguaje de Marcado de Hipertexto es el lenguaje que sirve para definir y estructurar el contenido de las páginas web. Este lenguaje consta de un conjunto de etiquetas para la definición de imágenes, listas, tablas, y que el propio navegador deberá de interpretar.

Este tipo de lenguaje tiene una estructura muy concreta, ya que los elementos que lo forman son únicos. Por ejemplo, para una imagen se utilizará la etiqueta ``. Para la distinción de un elemento se utilizan los signos “`<`” y “`>`”.

6.1.2 CSS



Figura 31: Logo CSS

Aunque HTML es el lenguaje fundamental para la creación de páginas web, no es suficiente por sí solo para producir sitios web que sean estéticamente agradables y útiles. En esta situación entra en juego el lenguaje conocido como CSS [2], que permite controlar la apariencia y el estilo de las páginas web.

CSS maneja la presentación, mientras que HTML maneja la estructura y el contenido. CSS es un lenguaje de hojas de estilo, por tanto, no es un lenguaje de programación en sí, permite ajustar cosas como el color, la tipografía, el espaciado, el tamaño y cómo se organizan los elementos en la página.

En el proyecto, los ficheros CSS son independientes a los ficheros HTML por lo que han sido aplicados al principio del código HTML introduciendo una línea como la siguiente:

```
<link rel="stylesheet" type="text/css" href("~/CSS/Catalogo.css") />
```

6.1.3 Bootstrap



Figura 32: Logo Bootstrap

Bootstrap [3] es un framework de desarrollo con HTML, CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Este framework permite la inclusión de una gran cantidad de componentes que pueden ser usados para una mejora en la interacción y comunicación con el usuario. Los principales componentes son: el uso de etiquetas flotantes, ventanas modales, spinners, tarjetas, paginación, etc.

El diseño responsive es una de las principales características de Boostrap, ya que permite la adaptación de la página según el tipo de dispositivo utilizado. Para el uso de Boostrap en el proyecto, se ha añadido un archivo CSS al inicio y otro JavaScript al final del documento HTML:

```
<link href "~/Content/bootstrap.min.css" rel="stylesheet">
<script src "~/Scripts/bootstrap.js"></script>
```

6.1.4 Leaflet



Figura 33: Logo Leaflet

Leaflet [4] es una biblioteca JavaScript de código abierto utilizada para crear aplicaciones de mapas web. Es compatible con la mayoría de las plataformas móviles y de escritorio. Esta biblioteca está diseñada teniendo en cuenta la simplicidad, el rendimiento y la facilidad de uso.

6.1.5 FullCalendar



Figura 34: Logo FullCalendar

FullCalendar [5] es uno de los plugins más populares y de tamaño completo para crear un calendario con JavaScript. Este complemento permite crear calendarios que pueden ser obtenidos por Ajax. Una de sus características destacadas es la capacidad de arrastrar y soltar eventos, e incluso sincronizarlos con Google Calendar.

Una ventaja adicional es su amplia personalización visual, lo que brinda la posibilidad de adaptar el calendario a diferentes estilos y preferencias.

6.2 BackEnd

El backend se encarga de la lógica del servidor y del procesamiento de las solicitudes del cliente.

6.2.1 ASP.NET



Figura 35: Logo ASP.NET

ASP.NET [6] es un marco de desarrollo web de código abierto creado por Microsoft. Su objetivo principal radica en facilitar la creación de aplicaciones y servicios web modernos utilizando la plataforma .NET. Este marco de desarrollo es multiplataforma y se puede ejecutar en todos los tipos de sistemas operativos actuales (Windows, Linux, macOS y Docker).

6.2.2 C#



Figura 36: Logo C#

C# [8] es un lenguaje de programación desarrollado por Microsoft que se utiliza ampliamente en la plataforma .NET. En el contexto de ASP.NET, C# es uno de los lenguajes de programación que se puede usar para escribir el código de las aplicaciones web. Juntos, ASP.NET y C# permiten desarrollar aplicaciones web robustas y dinámicas utilizando la plataforma .NET.

6.2.3 SQL Server



Figura 37: Logo SQL Server

SQL Server [9] es un sistema de gestión de base de datos relacional que permite modelar y organizar la información en tablas con relaciones definidas. SQL Server ofrece un potente lenguaje de consulta “SQL” que permite realizar consultas complejas y realizar análisis de datos.

Además, permite la implementación y el diseño de procedimientos almacenados [10] que son un conjunto de instrucciones SQL precompilados y almacenados en la base de datos, lo que da lugar a un procesamiento más rápido de las consultas.

6.2.4 Firebase



Figura 38: Logo Firebase

Firebase [11] es una plataforma digital diseñada para el desarrollo de aplicaciones web y móviles proporcionada por Google. Esta plataforma ofrece un amplio número de herramientas y servicios para la creación de aplicaciones de manera rápida y segura. Los distintos módulos que contiene Firebase se dividen en el almacenamiento en la nube, autenticación de usuarios, base de datos en tiempo real, hosting web, etc.

6.2.5 JavaScript



Figura 39: Logo JavaScript

JavaScript [12] es un lenguaje de programación avanzado, con interpretación automática y orientado al paradigma de objetos. Es ampliamente empleado en el campo del desarrollo web, con el propósito de añadir capacidad de interacción y dinamismo a los sitios web.

Este lenguaje de programación es compatible con cualquier tipo de navegador y se ejecuta del lado del cliente. Además, proporciona una gran cantidad de funcionalidades, así como la interacción de elementos del sitio web, la validación de formularios o la comunicación con el servidor a partir de Ajax.

Ajax [25] es una técnica de programación en JavaScript que permite realizar solicitudes al servidor de forma asíncrona, sin tener que recargar toda la página web. Esto significa que los datos pueden enviarse y recibir desde el servidor sin afectar la experiencia del usuario en la página actual.

6.2.6 REST



Figura 40: Logo servicios REST

Un Web Service [13], también conocido como servicio web, es un método de comunicación entre dos dispositivos electrónicos bajo una misma red. Además, es una colección de protocolos y estándares para el intercambio de datos entre sistemas. Existen diferentes tipos de servicios web, pero para el proyecto actual se ha optado por el servicio REST.

REST [14] es un estilo arquitectónico para la construcción de servicios web basados en el estándar HTTP. Se enfoca en acceder a recursos nombrados a través de una única interfaz consistente. Este estilo arquitectónico, permite muchos formatos de datos diferentes (JSON, Text, HTML) mientras que otros tipos de servicios como SOAP, solo admite el estándar XML.

6.3 Otras herramientas

6.3.1 Visual Studio 2019



Figura 41: Logo Visual Studio

Visual Studio [15] es una herramienta que centraliza todo el ciclo de desarrollo en un único lugar. Se trata de un entorno de desarrollo integrado que permite realizar diversas tareas, desde escribir y editar el código hasta depurarlo y compilarlo, y finalmente implementar la aplicación.

Además de las funciones básicas, Visual Studio incorpora una amplia variedad de características adicionales, como compiladores, control de versiones, extensiones y muchas otras utilidades que optimizan cada etapa del proceso de desarrollo de software.

6.3.2 GitHub



Figura 42: Logo GitHub

El proyecto se ha desarrollado utilizando Git [16] como sistema de control de versiones y GitHub como la plataforma para almacenar el repositorio del código. El uso de un sistema de control de versiones presenta ventajas en los procesos de desarrollo, ya que permite el diseño de diferentes versiones con sus respectivas funcionalidades en paralelo.

Además, se mantiene un control del histórico de cambios, lo que permite realizar un seguimiento del progreso del proyecto, identificar y solucionar problemas. En concreto, Git se basa en un sistema de ramas para el que se permite crear una rama principal estable y ramas de desarrollo para nuevas características.

6.3.3 Figma



Figura 43: Logo Figma

Figma [17] es una plataforma líder en diseño de interfaces, que destaca por su naturaleza online y colaborativa. Esta herramienta permite realizar una amplia gama de tareas en diseño gráfico, así como crear páginas web o interfaces gráficas para aplicaciones.

En el sistema desarrollado se ha utilizado únicamente para el diseño del prototipado. De esta forma, el desarrollador puede adquirir una visión más clara del flujo de la aplicación y las interacciones previstas. Esto permite identificar y solucionar posibles problemas antes de iniciar la fase de desarrollo.

6.3.4 Trello



Figura 44: Logo Trello

En el ámbito de la gestión de proyectos, se ha utilizado la herramienta Trello [18] para realizar un seguimiento del proyecto de una manera visual y sencilla. Trello es un tipo de flujo de trabajo Kanban para ayudar a realizar un seguimiento de las distintas actividades de un proyecto de una manera más visual y corporativa. Esta herramienta permite crear tableros de proyectos virtuales, además de listas y tarjetas para representar las diversas fases del trabajo.

Para ordenar el flujo del trabajo, se ha utilizado un tablero Kanban en el que se han incorporado 4 estados relativos a las tareas del proyecto: “Pendientes”, “En desarrollo”, “Testing” y “Publicadas”. En el siguiente ejemplo, se puede visualizar el uso de las tarjetas en Trello para el seguimiento de las distintas actividades.

6.3.5 TeamGantt



Figura 45: Logo teamgantt

TeamGantt [19] es una herramienta altamente especializada en la creación de diagramas de Gantt, lo que permite a los usuarios optimizar la organización de sus proyectos desde una fase inicial. Tras el uso de este método de gestión, los usuarios pueden visualizar de manera clara y detallada las tareas, plazos y dependencias de sus proyectos en un solo lugar. Esto facilita la identificación de posibles cuellos de botella y la asignación eficiente de recursos.

6.3.6 Postman



Figura 46: Logo Postman

Postman [20] es una herramienta eficiente y efectiva para facilitar el desarrollo y pruebas de APIs. Su funcionalidad principal radica en configurar rutas de endpoints del API y ejecutarlas para probar el backend de los sistemas.

Además, Postman ofrece comodidad al trabajar con todos los métodos del protocolo HTTP. De lo contrario, sin una herramienta como Postman, los desarrolladores tendrían que generar código de cliente, usualmente con JavaScript y Ajax, para realizar las solicitudes al servidor y probar todos esos métodos.

6.4 Conclusiones

En el contexto de la aplicación de gestión de bibliotecas donde se tiene relaciones entre distintas entidades como bibliotecas, lectores, libros, eventos, etc, y donde la consistencia de datos es crucial, tanto SQL Server como MySQL podrían ser opciones apropiadas. Ambos sistemas de gestión de bases de datos relacionales ofrecen la opción de definir estructuras de datos y gestionar relaciones complejas entre tablas.

En el proyecto actual se han utilizado los servicios de almacenamiento en la nube (Firebase Cloud Storage). Este servicio permite el almacenamiento de imágenes en la nube a partir de la generación de URL prefirmadas sobre las fotos almacenadas. Estas URLs son seguras y pueden ser utilizadas directamente para acceder a las fotos sin necesidad de configurar un sistema complejo para el almacenamiento de imágenes. Para ello, se ha creado una columna en cada una de las tablas SQL para el almacenamiento de imágenes. Las tablas son 4: Biblioteca, Usuario, Evento y Libro. Estas tablas tendrán una columna llamada “Foto” que almacenarán la URL prefirmada generada por Firebase Storage.

Para el desarrollo de la aplicación web se ha optado por ASP.NET y C#, las cuales se complementan para proporcionar una plataforma sólida para la implementación de aplicaciones web complejas. ASP.NET ha facilitado la arquitectura de tres capas MVC, el manejo de sesiones y la validación de formularios en la aplicación.

El flujo de trabajo o workflow que se ha seguido consiste en un control preciso de las diferentes versiones de la aplicación web a través del uso de 2 ramas en Git, “master” y “development”. La primera de ellas es la rama principal, que contiene las funcionalidades que han sido probadas a través de la técnica de testing. Por otro lado, se dispone de otra rama development, en la que se implementan las nuevas necesidades de la aplicación, que una vez han sido probadas pasarán a formar parte de la rama master.

7. Implementación

7.1 API REST de la aplicación

Con el sistema actual se va a crear una API REST que consumiremos desde la aplicación. Para el consumo de dicha API, simplemente bastará con hacer peticiones HTTP al servidor a través de diferentes métodos (GET, POST, DELETE, PUT).

Login y registro usuario

Acción	Método	Ruta
Acceder al sistema (obtener cookie de sesión)	POST	Acceso/Login
Cerrar sesión (eliminar cookie de sesión)	GET	Acceso/Logout
Registrar una nueva biblioteca	POST	Acceso/RegistrarBiblioteca
Registrar un nuevo lector	POST	Acceso/RegistrarLector

Biblioteca

Acción	Método	Ruta
Modificar el perfil de una biblioteca	PUT	Biblioteca/ModificarBiblioteca?id={id}

Lector

Acción	Método	Ruta
Modificar el perfil de un lector	PUT	Lector/ModificarLector?id={id}

Libros

Acción	Método	Ruta
Crear un nuevo libro	POST	Biblioteca/GuardarLibro
Obtener un listado de todos los libros	GET	Lector>ListarAllLibros
Obtener un listado de los libros según un determinado idioma y/o categoría (filtros)	GET	Lector>ListarLibrosPorIdiomaCategoria?idioma={idioma}&categoria={categoria}
Obtener un libro	GET	Lector/ObtenerLibro?id={id}
Modificar un libro	PUT	Biblioteca/ModificarLibro?id={id}
Eliminar un libro	DELETE	Biblioteca/EliminarLibro?id={id}

Eventos

Acción	Método	Ruta
Crear un nuevo evento	POST	Biblioteca/GuardarEvento
Obtener un listado de todos los eventos	GET	Lector>ListarAllEventos
Obtener un listado de todos los eventos de una biblioteca	GET	Biblioteca>ListarEventoPorBiblio?id={id}
Obtener un listado de los eventos según el tipo de evento seleccionado (filtros)	GET	Lector>ListarEventosPorTipo?tipo={tipo}

Obtener un evento	GET	Lector/ObtenerEvento?{id}
Modificar un evento	PUT	Biblioteca/ModificaEvento?id={id}
Eliminar un evento	DELETE	Biblioteca/EliminarEvento?id={id}

Reseñas

Acción	Método	Ruta
Crear una nueva reseña	POST	Lector/RegistrarReseña
Obtener un listado de todas las reseñas por libro	GET	Lector/ListarReseñasPorLibro?id={id}

Reservas

Acción	Método	Ruta
Registrar nueva reserva	POST	Lector/RegistrarReserva
Obtener un listado de todas las reservas de una biblioteca	GET	Lector>ListarReservasBiblio?id={id}

Préstamos

Acción	Método	Ruta
Registrar nuevo préstamo	POST	Lector/RegistrarPrestamo
Obtener un listado de todos los préstamos de una biblioteca	GET	Lector>ListarPrestamosBiblio?id={id}
Obtener un préstamo	GET	Lector/ObtenerPrestamo?{id}
Devolver un préstamo de una biblioteca	POST	Biblioteca/DevolverPrestamo

Notificaciones

Acción	Método	Ruta
Enviar una notificación a un lector	POST	Biblioteca/EnviarNotificacion
Listar notificaciones por lector	GET	Lector/ListarNotificacionPorLector?id={id}
Eliminar notificación	DELETE	Lector/EliminarNotificacion?id={id}

7.2 Estructura del proyecto

En esta sección, se explicará la estructura de la aplicación internamente, mostrando de forma gráfica y detallada la estructura de las carpetas del proyecto.

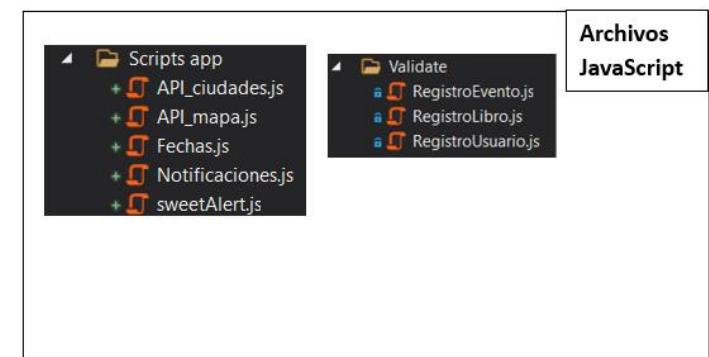
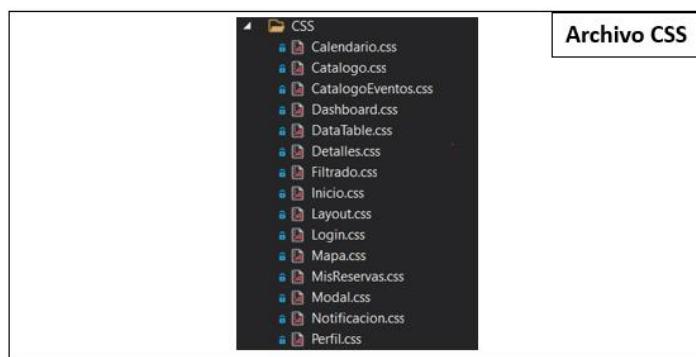
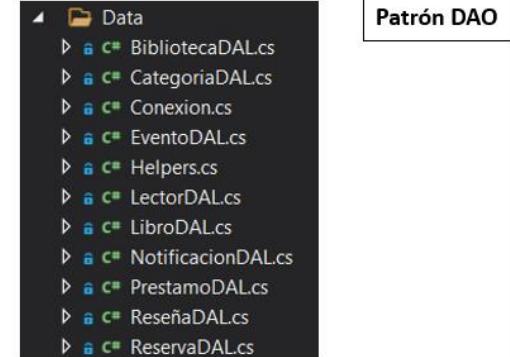
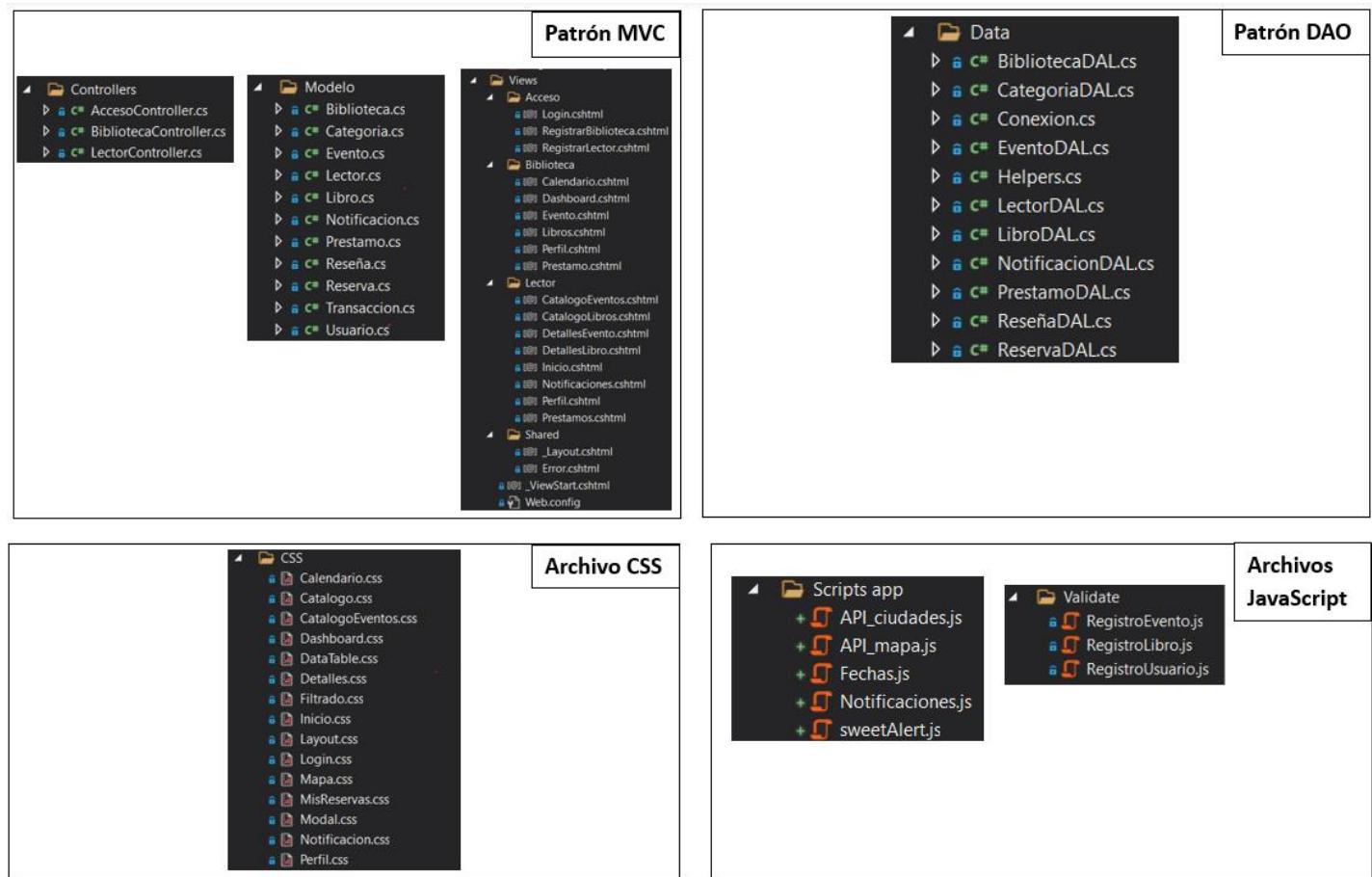


Figura 47: Estructura del proyecto

1. Directorios siguiendo el patrón MVC:

- **Modelo (Model):** En esta sección, se tienen las clases que representan las entidades o datos de la aplicación.
- **Vista (View):** Aquí se encontrarán las vistas HTML que se muestran a los usuarios. Estas vistas utilizan los estilos CSS y el código JavaScript para crear la interfaz de usuario.
- **Controlador (Controller):** En este directorio, se encuentran los controladores que manejan las solicitudes del cliente. Los controladores procesan la lógica, interactúan con el modelo y devuelven la vista adecuada al cliente.

- 2. Directorio para el DAO (Data Access Object):** En esta carpeta, se encuentran los archivos relacionados con el acceso a la base de datos y la interacción con los datos. En sus archivos se alojan las consultas SQL, operaciones CRUD (Crear, Leer, Actualizar, Eliminar), y la comunicación con la base de datos.
- 3. Directorio para CSS:** En esta sección, se alojan los archivos de hojas de estilo en cascada (CSS) que controlan la apariencia visual y el diseño de las vistas HTML. Estos archivos definen cómo se verá el contenido en el navegador.

4. Directorio para JavaScript (js):

- **Validate:** En este directorio, se almacenan archivos JavaScript que contienen la lógica de validación de formularios al registrar o modificar eventos, libros o usuarios.
- **Scripts system:** Esta carpeta contiene
 - API_ciudades.js: contiene código para interactuar con la API de “Universal Tutorial” para obtener los nombres de las ciudades en función de los estados seleccionados por el usuario en la interfaz HTML. Además, también maneja el almacenamiento y recuperación de la ciudad seleccionada.
 - API_mapa.js: es un archivo JavaScript que se encarga de obtener las coordenadas de una determinada ubicación utilizando la API de “OpenStreetMap”, posteriormente utiliza Leaflet para mostrar el mapa interactivo. Además, este archivo utiliza la fórmula de Haversine para conocer la distancia en kilómetros entre el lector y una determinada biblioteca.
 - Fechas.js: contiene funciones relacionadas con el manejo y formateo de fechas en JavaScript. Estas funciones facilitan el trabajo con fechas al ofrecer diversos formatos de salida y permitir la manipulación de las mismas.
 - Notificaciones.js: es un archivo JavaScript que contiene funciones para obtener, procesar y mostrar las notificaciones en la interfaz del sistema.
 - sweetAlert.js: contiene fragmentos de código JavaScript que usan promesas con la biblioteca SweetAlert (Swal) para crear ventanas emergentes con mensajes de éxito y confirmación.

7.3 Modelo Vista Controlador

Como se ha explicado anteriormente el sistema sigue una arquitectura Modelo Vista Controlador. Las próximas secciones contendrán fragmentos de código de la aplicación que pondrán en práctica el patrón arquitectónico.

Las siguientes ilustraciones corresponderán a la tabla “Libro” de la base de datos. Se ha optado por esta tabla por considerarse el elemento principal de la aplicación.

7.3.1 Modelo

El Modelo se refiere a la representación de los datos y la lógica de negocio de la aplicación. En la siguiente Figura se muestra la clase “Libro” como parte del modelo de datos. La clase libro define las propiedades que representan los atributos de un libro en el sistema.

```
public class Libro
{
    public int Id { get; set; }
    public string Titulo { get; set; }
    public string Idioma { get; set; }
    public string Foto { get; set; }
    public string Ubicacion { get; set; }
    public string Autor { get; set; }
    public string Editorial { get; set; }
    public int idCategoria { get; set; }
    public int idBiblioteca { get; set; }
    public int numEjemplares { get; set; }
    public string ISBN { get; set; }
    public bool Estado { get; set; }
}
```

Figura 48: Código Modelo Libro

En la implementación del sistema se ha querido estructurar parte del código de una manera más eficiente, por lo que se ha utilizado la herencia entre clases.

La herencia de clases [7] es una característica de los lenguajes de programación orientados a objetos que permite la definición de una clase base, que contiene funcionalidad específica y las clases derivadas heredan esa funcionalidad.

En el sistema existe una clase “Usuario” que contiene atributos comunes que los dos tipos de usuario (lector y biblioteca) deben tener. Por tanto, se dice que las subclases “Biblioteca” y “Lector” heredan las propiedades de la superclase “Usuario”. Además, tanto Biblioteca como Lector pueden tener sus propias propiedades adicionales que son específicas de cada uno. Por ejemplo, la clase Lector tiene propiedades como *Apellidos*, *Dni* y *FechaNacimiento*, que son relevantes solo para los lectores.

Superclase	Subclases
<pre>public class Usuario { public int Id { get; set; } public string Nombre { get; set; } public string Email { get; set; } public string Telefono { get; set; } public string Contraseña { get; set; } public string Ciudad { get; set; } public string Calle { get; set; } public string País { get; set; } public string Estado { get; set; } public string CodPostal { get; set; } public string Foto { get; set; } public string TipoUsuario { get; set; } public string ConfirmarContraseña { get; set; } }</pre>	<pre>public class Lector : Usuario { 3 referencias public string Apellidos { get; set; } 3 referencias public string Dni { get; set; } 3 referencias public DateTime FechaNacimiento { get; set; } } public class Biblioteca : Usuario { 3 referencias public string Descripcion { get; set; } }</pre>

Figura 49: Código Herencia de clases

7.3.2 Vista

En el sistema arquitectónico de la aplicación, las vistas desempeñan un papel esencial al presentar datos y ejecutar código JavaScript en el navegador del usuario. Este código desempeña un rol crítico en la realización de acciones básicas, como efectuar solicitudes HTTP mediante Ajax, así como en la recepción y procesamiento de respuestas entregadas por los controladores del servidor. Estas respuestas, se fusionan con las vistas y se exponen en el navegador en formato HTML, complementadas con estilos CSS.

En la Figura 50, se mostrarán varias capturas de pantalla que ejemplificarán las vistas. En estas capturas, se presentarán ejemplos de llamadas Ajax utilizadas para llevar a cabo peticiones HTTP a la API REST del sistema.

```
function EnviarReseña() {
    var oReseña = {
        Descripción: $("#descripcionReseña").val(),
        Puntuacion: $("input[name='puntuacion']:checked").val(),
        idLibro: @Model,
        idLector: @((Usuario)Session["Usuario"]).Id,
    }
    $.ajax({
        url: "/Lector/RegistrarReseña",
        type: "POST",
        data: oReseña,
        success: function (data) {
            alertSuccess('¡Enhorabuena!', 'Acabas de escribir una reseña.').then(() => { window.location.reload(); })
        },
        error: function (error) {
            console.log("Error al obtener los datos del préstamo:", error);
            reject(error);
        }
    });
}

function getReseñas() {
    var idLibro = @Model;
    var numReseñas = 0;
    var sumaPuntuacion = 0;
    $.ajax({
        url: "/Lector/ListarReseñasPorLibro",
        type: "GET",
        data: { id: idLibro },
        success: function (data) {
            $("#reseñas-list").empty();
            $.each(data.data, function (index, reseña) {
                mostrarInfoReseña(reseña);
                numReseñas++;
                sumaPuntuacion += reseña.Puntuacion;
            });
            mostrarInfoTotalReseñas(numReseñas, sumaPuntuacion);
        },
        error: function (error) {
            console.log("Error al obtener las reseñas:", error);
        }
    });
}
```

Figura 50: Código Vista

7.3.3 Controlador

El controlador actúa como intermediario entre la vista y el modelo. Se han desarrollado 3 controladores en el sistema “AccesoController”, “LectorController” y “BibliotecaController”. Cada controlador representa una parte importante del sistema y se enfoca en acciones específicas permitidas con la aplicación.

- Controlador Acceso: se encarga de la autenticación y el inicio de sesión de los usuarios.
- Controlador Biblioteca: este controlador gestiona las acciones que el administrador de una biblioteca puede realizar con el sistema.

```
[HttpGet]
0 referencias
public JsonResult ListarLibrosPorBiblioteca(int id)
{
    List<Libro> oLista = new List<Libro>();
    oLista = LibroDAL.Instancia.ListarPorBiblioteca(id);
    return Json(new { data = oLista }, JsonRequestBehavior.AllowGet);
}

[HttpPost]
0 referencias
public JsonResult EliminarLibro(int id)
{
    bool respuesta = false;
    respuesta = LibroDAL.Instancia.Eliminar(id);
    return Json(new { resultado = respuesta }, JsonRequestBehavior.AllowGet);
}

[HttpPost]
0 referencias
public async Task<JsonResult> GuardarLibro(string objeto, HttpPostedFileBase imagenArchivo)
{
    Stream image = imagenArchivo.InputStream;
    string fileName = Path.GetFileName(imagenArchivo.FileName);
    string urlImagen = await new Helpers().SetImageToFirebase(image, fileName, "Fotos_Libros");

    Libro oLibro = new Libro();
    oLibro = JsonConvert.DeserializeObject<Libro>(objeto);
    oLibro.Foto = urlImagen;

    bool respuesta = false;
    respuesta = LibroDAL.Instancia.Registrar(oLibro);
    return Json(new { resultado = respuesta }, JsonRequestBehavior.AllowGet);
}
```

Figura 51: Código Controlador biblioteca

- Controlador Lector: gestiona las operaciones que el usuario lector es capaz de ejecutar en la aplicación.

```
[HttpGet]
0 referencias
public JsonResult ObtenerLibro(int id)
{
    Libro oLibro = LibroDAL.Instancia.Obtener(id);
    return Json(new { data = oLibro }, JsonRequestBehavior.AllowGet);
}

[HttpGet]
0 referencias
public JsonResult ListarLibroPorIdiomaCategoria(string idioma, string categoria)
{
    List<Libro> oLista = new List<Libro>();
    oLista = LibroDAL.Instancia.ListarLibroPorIdiomaCategoria(idioma, categoria);
    return Json(new { data = oLista }, JsonRequestBehavior.AllowGet);
}

[HttpGet]
0 referencias
public JsonResult ListarLibroPorBusqueda(string nombre)
{
    List<int> oLista = new List<int>();
    oLista = LibroDAL.Instancia.ListarId(nombre);
    return Json(new { data = oLista }, JsonRequestBehavior.AllowGet);
}

[HttpGet]
0 referencias
public JsonResult ListarAllLibros()
{
    List<Libro> oLista = new List<Libro>();
    oLista = LibroDAL.Instancia.Listar();
    return Json(new { data = oLista }, JsonRequestBehavior.AllowGet);
}
```

Figura 52: Código Controlador lector

7.4 Patrón DAO

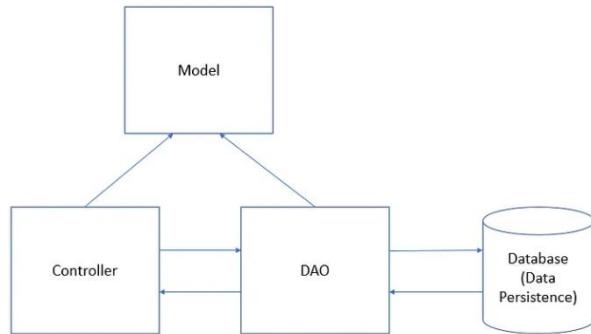


Figura 53: Diagrama patrón DAO

El patrón DAO (Data Access Object) es uno de los patrones de diseño utilizados para implementar el DAL (Data Access Layer) o Capa de Acceso a Datos. La Capa de Acceso a Datos es una capa interna de la aplicación que se encarga de manejar las operaciones relacionadas con la persistencia de datos, así como la actualización, inserción, consulta y borrado de datos.

El patrón DAO es un patrón arquitectónico que pretende implementar esta capa, ya que busca la separación de la lógica de negocio con la lógica para acceder a los datos. El objetivo principal de este patrón es lograr una mejor estructura y mantenimiento de la aplicación, al aislar la lógica de acceso a datos en una capa separada. Con esto se permite la modificación de la fuente de datos sin afectar la lógica de la aplicación.

En el proyecto actual se ha creado un archivo “DAL” por cada una de las tablas de la base de datos que requiere sus correspondientes operaciones CRUD. Si se acceder al archivo “LibroDAL” se tienen las siguientes operaciones CRUD:

```

public class LIBRODAL
{
    private static LIBRODAL instancia = null;
    private LIBRODAL()
    {
        public static LIBRODAL Instancia
        {
            get
            {
                if (instancia == null)
                {
                    instancia = new LIBRODAL();
                }
                return instancia;
            }
        }
    }
    public Libro obtener(int id)
    {
        Libro libro = new Libro();
        string consultaSql = "SELECT Titulo, Foto, idBiblioteca, Autor, idCategoria, Editorial, numEjemplares, Idioma "
                            + "FROM LIBRO "
                            + "WHERE Id = @libroId ";
        using (SqlConnection connection = new SqlConnection(Conexion.CN))
        {
            SqlCommand command = new SqlCommand(consultaSql, connection);
            command.Parameters.AddWithValue("@libroId", id);
            connection.Open();

            SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                libro.Titulo = reader["Titulo"].ToString();
                libro.Idioma = reader["Idioma"].ToString();
                libro.Foto = reader["Foto"].ToString();
                libro.numEjemplares = reader.GetInt32(reader.GetOrdinal("numEjemplares"));
                libro.idBiblioteca = reader.GetInt32(reader.GetOrdinal("idBiblioteca"));
                libro.Autor = reader["Autor"].ToString();
                libro.idCategoria = reader.GetInt32(reader.GetOrdinal("idCategoria"));
                libro.Editorial = reader["Editorial"].ToString();
            }
            connection.Close();
            reader.Close();
        }
        return libro;
    }

    public List<Libro> Listar()
    {
        List<Libro> listaLibros = new List<Libro>();
        using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
        {
            string query = @""
                            + "SELECT l.Id, l.Titulo, l.Foto, l.idBiblioteca, l.Autor, l.idCategoria, l.Editorial, "
                            + "l.Ubicacion, l.numEjemplares, l.Estado "
                            + "FROM LIBRO l "
                            + "INNER JOIN biblioteca b ON b.Id = l.idBiblioteca "
                            + "INNER JOIN CATEGORIA c ON c.Id = l.idCategoria";

            SqlCommand cmd = new SqlCommand(query, oConexion);
            cmd.CommandType = CommandType.Text;
            try
            {
                oConexion.Open();
                SqlDataReader dr = cmd.ExecuteReader();

                while (dr.Read())
                {
                    listaLibros.Add(new Libro()
                    {
                        Id = Convert.ToInt32(dr["Id"].ToString()),
                        Titulo = dr["Titulo"].ToString(),
                        Foto = dr["Foto"].ToString(),
                        idBiblioteca = Convert.ToInt32(dr["idBiblioteca"].ToString()),
                        Autor = dr["Autor"].ToString(),
                        idCategoria = Convert.ToInt32(dr["idCategoria"].ToString()),
                        Editorial = dr["Editorial"].ToString(),
                        Ubicacion = dr["Ubicacion"].ToString(),
                        numEjemplares = Convert.ToInt32(dr["numEjemplares"].ToString()),
                    });
                }
                dr.Close();
            }
            return listaLibros;
        }
        catch (Exception ex)
        {
            listaLibros = null;
            return listaLibros;
        }
    }

    public bool Registrar(Libro objeto)
    {
        bool respuesta = true;
        using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
        {
            try
            {
                SqlCommand cmd = new SqlCommand("sp_RegistrarLibro", oConexion);
                cmd.Parameters.AddWithValue("Titulo", objeto.Titulo);
                cmd.Parameters.AddWithValue("Foto", objeto.Foto);
                cmd.Parameters.AddWithValue("Idioma", objeto.Idioma);
                cmd.Parameters.AddWithValue("Estado", objeto.Estado);
                cmd.Parameters.AddWithValue("Autor", objeto.Autor);
                cmd.Parameters.AddWithValue("idCategoria", objeto.idCategoria);
                cmd.Parameters.AddWithValue("Editorial", objeto.Editorial);
                cmd.Parameters.AddWithValue("idBiblioteca", objeto.idBiblioteca);
                cmd.Parameters.AddWithValue("Ubicacion", objeto.Ubicacion);
                cmd.Parameters.AddWithValue("numEjemplares", objeto.numEjemplares);
                cmd.Parameters.AddWithValue("ISBN", objeto.ISBN);
                cmd.Parameters.Add("Resultado", SqlDbType.Int).Direction = ParameterDirection.Output;
                cmd.CommandType = CommandType.StoredProcedure;

                oConexion.Open();

                cmd.ExecuteNonQuery();

                respuesta = Convert.ToBoolean(cmd.Parameters["Resultado"].Value);
            }
            catch (Exception ex)
            {
                respuesta = false;
            }
        }
        return respuesta;
    }

    public bool Eliminar(int id)
    {
        bool respuesta = true;
        using (SqlConnection oConexion = new SqlConnection(Conexion.CN))
        {
            try
            {
                SqlCommand cmd = new SqlCommand("delete from LIBRO where Id = @id", oConexion);
                cmd.Parameters.AddWithValue("@id", id);
                cmd.CommandType = CommandType.Text;

                oConexion.Open();

                cmd.ExecuteNonQuery();

                respuesta = true;
            }
            catch (Exception ex)
            {
                respuesta = false;
            }
        }
        return respuesta;
    }
}

```

Figura 54: Código funciones DAL



8. Pruebas

8.1 Encuesta inicial de usabilidad

Una vez la aplicación web estaba completamente desarrollada, se llevó a cabo una encuesta de usabilidad con un grupo diverso de personas. En la encuesta, se pedía a los participantes que realizaran una tarea específica sin recibir ningún tipo de ayuda. El objetivo principal era que esta prueba fuera adecuada tanto para personas con experiencia tecnológica como sin experiencia previa. Mediante esta encuesta se buscaba mejorar la interfaz, ajustando los elementos que resultaban más confusos o que requerían más tiempo.

Al inicio de la actividad, se les leía a los usuarios la siguiente información:

Actualmente eres el administrador de una biblioteca, lo que significa que tienes el control completo sobre sus libros y eventos, por tanto, podrás visualizarlos, editarlos, eliminarlos y registrar nuevos recursos en cualquier momento. Además, dispondrás de una sección de “préstamos” fraccionada en varios estados. Los estados siguen el siguiente orden.

- **En espera:** este estado es meramente informativo, ya que cuando un libro se encuentra en espera el administrador no tiene que realizar ninguna acción. Cuando el libro esté disponible el estado del lector que estaba en espera pasará a “pendiente de recogida”.
- **Pendiente recogida:** cuando un lector recoge un libro físicamente el administrador registrará su recogida.
- **Pendiente devolución:** cuando un lector devuelve un libro el administrador registrará su devolución.
- **Devuelto:** un administrador puede consultar las condiciones en que el libro fue devuelto.

Una vez dicho esto, se comenzaba inmediatamente a describir la prueba, la idea es que el usuario fuera realizando la tarea a medida que se le iba leyendo el enunciado.

8.1.1 Primera prueba

Una vez se ha leído el punto 8.1 se comienza inmediatamente a describir la prueba que deberá de ejecutar el participante. A continuación, se describirá la prueba.

- *Paso 1: Un administrador de una biblioteca se ha dado cuenta de que el libro con ISBN 9788498387209 contiene un campo incorrecto. Para ello, realiza una búsqueda por ISBN y posteriormente edite la categoría, que debería de ser “Medicina y Salud”. A continuación, vuelva al listado de libros inicial.*
- *Paso 2: Un administrador de una biblioteca quiere cambiar su información de perfil, modifica su foto de biblioteca y cambia su número de teléfono por uno nuevo. Comprueba que los cambios se han guardado de manera correcta.*
- *Paso 3: La lectora “Lucía Santos González” ha acudido a la biblioteca para recoger el libro “Bajo la misma estrella” ejecute las acciones correspondientes para que se registre la recogida del libro.*
- *Paso 4: El lector “Carlos Poveda Martínez” ha acudido a la biblioteca para devolver el libro “Classical Art” ejecute las acciones correspondientes para que se registre la devolución del libro. El estado de la devolución debe contener un comentario positivo.*

- *Paso 5: Al administrador de una biblioteca le ha llegado la información de que el libro “El principito” está en pésimas condiciones, por lo que se quiere conocer al lector que fue el responsable de su deterioro. Para ello, revise los estados en que fueron devueltos los libros e identifique el negativo.*
- *Paso 6: El administrador de una biblioteca quiere cerrar la sesión.*

Número de usuario	1	2	3
Problemas surgidos	La redacción del texto del Paso 1 ponía “Actualice la categoría” y el participante no entendía lo que tenía que hacer, tras decirle “Edite la Categoría” encontró el icono con facilidad	Ninguno	Ninguno
Sugerencias de mejora	Ninguna	En el Modal de registro de una biblioteca cambiar la palabra Estado por Provincia	Cambiar el botón para modificar un libro de “Registrar” a “Guardar cambios”
Tiempo tardado	3min 42segundos	3min 24segundos	3min 03segundos

Tabla 4: Resultados de la encuesta

Como se puede observar en la Tabla 4, tanto los problemas como las sugerencias de mejora tienen poco impacto en el desarrollo. Esto puede ocurrir porque la interfaz destinada a la tarea es bastante intuitiva, lo que disminuirá posibles fallos o confusiones en el usuario final.

9. Conclusiones

Tras finalizar el Trabajo de Fin de Carrera, se puede concluir que se han alcanzado los objetivos planteados en su inicio. En primer lugar, se realizó un análisis de las tecnologías existentes en la actualidad, con lo que se vio la necesidad de crear un sistema de bibliotecas que permitiese la reserva de libros físicos en línea.

Posteriormente, se pasó a la gestión del proyecto en la que se realizaron las estimaciones de tiempo de cada una de las fases del proyecto. En cuanto a las estimaciones, cabe destacar que el proyecto ha cumplido los tiempos estimados tras el inicio, en suma, ha permitido la implementación de nuevas funcionalidades en el desarrollo, así como la escritura de reseñas por los lectores y la presencia de calendarios en la parte del administrador de una biblioteca.

Otra fase importante del proyecto fue el estudio de la documentación de las distintas APIs públicas. En esta etapa del proyecto aprendí a realizar peticiones asíncronas a un servidor desde el lado del cliente con el uso de AJAX. Cabe destacar que el concepto de servicio web lo había estudiado en la carrera, pero no lo había puesto en práctica en un proyecto funcional. El uso de las APIs fue necesario para la obtención de datos geoespaciales de los mapas interactivos y para obtener los datos de las descripciones de los libros a través de Wikipedia.



En cuanto al desarrollo de la aplicación, se ha conseguido implementar una aplicación funcional para la reserva de libros enfocado a las distintas bibliotecas de la Comunidad Valenciana. Esta aplicación permite dos tipos de usuarios el “administrador de una biblioteca” y el “lector”. En primer tipo de usuario, se encarga de gestionar todos sus libros y eventos en cualquier momento. Por otro lado, los lectores pueden solicitar cualquier tipo de libro, y dependiendo de su disponibilidad se permitirá su préstamo o reserva. Además, los lectores podrán visualizar los próximos eventos que han sido programados por cualquier biblioteca.

Se han realizado pruebas de usabilidad con usuarios reales para obtener comentarios de posibles mejoras en la interfaz de la aplicación. Tras estas pruebas se ha podido concluir que la interfaz era amigable y intuitiva para el usuario final, además se ha tenido en cuenta las sugerencias de mejora para lograr una mayor eficiencia con la interacción con el sistema.

El problema por el que ha pasado la aplicación ha sido el elevado tiempo de carga en una de las ventanas del sistema. Esta ventana incluía la distancia en kilómetros del lector por cada uno de los libros, por lo que si había muchos libros cargados en el sistema la página no cargaba hasta que todas las distancias habían sido calculadas. Con este problema, se optó por que apareciera simplemente la ciudad en la que se encontraba el libro. Una vez que el lector accedía a la información del libro, podía conocer la distancia en kilómetros del lector a la biblioteca donde se encontraba almacenado el libro. De esta forma este cálculo costoso solo se realizaba una vez y no numerosas veces en la misma página.

En resumen, el desarrollo de este proyecto va mucho más allá de simplemente programar una interfaz y documentar su funcionamiento. Ha proporcionado una valiosa oportunidad para comprender, el panorama laboral que aguarda a los ingenieros informáticos.

9.1 Relación del proyecto con los estudios cursados

Durante mi trayectoria académica en la Universidad Politécnica de Valencia en el estudio del grado de Ingeniería Informática, he tenido la oportunidad de adquirir conocimientos con diversas herramientas y tecnologías fundamentales para el desarrollo de un software de calidad y para la gestión del proyecto.

Durante mis estudios de grado, cursé la asignatura de GPR (Gestión de proyectos) en la que me familiaricé en realizar estimaciones temporales de un proyecto a partir del uso de diversos tipos de diagramas, así como el diagrama de Gantt. Además, tuve la oportunidad de utilizar una herramienta de gestión de proyectos basada en tableros denominada “Trello”. Esta plataforma me proporcionó una base sólida para planificar y administrar las etapas de desarrollo de mi TFG de manera efectiva.

Por otro lado, cursar la asignatura de ISW (Ingeniería del software) incluyó un enfoque en el control de versiones y la colaboración en proyectos de software. Utilicé GitHub para gestionar el código fuente de mi proyecto, lo que me ha permitido un seguimiento preciso de las distintas modificaciones en el código. Además, ISW me ha proporcionado los conocimientos para el diseño del sistema utilizando los diferentes tipos de diagramas UML (diagrama de clase, diagrama de casos de uso, diagrama de contexto, etc.). Estos diagramas me han permitido modelar y visualizar tanto la estructura como las interacciones de la aplicación.

En el grado he cursado la asignatura de BDA (Bases de datos y sistemas de información) en la que he adquirido conocimientos en la gestión de bases de datos relacionales, así como el diseño de esquemas de bases de datos y la creación de consultas SQL. Este aprendizaje me ha ayudado a utilizar el sistema SQL Server con mayor confianza y fluidez.

La asignatura de DDS (Diseño del software) me ha enseñado a aplicar buenas prácticas de desarrollo y a utilizar patrones de diseño como MVC (Modelo Vista Controlador) y el patrón DAO. También me ha ayudado a aplicar técnicas de refactorización en el código, para una mejora en la legibilidad, la eficiencia y la escalabilidad del código.

Una de las asignaturas clave en mi formación ha sido PIN (Proyecto de ingeniería de software) en la que aprendí la importancia de comprender y documentar los requisitos del sistema de manera detallada. Además, puse en práctica la creación de prototipos lo que me permitió visualizar el diseño software de la interfaz del usuario antes de su implantación. Una parte fundamental de PIN fue la implementación completa de la aplicación. Desde la elección de las tecnologías hasta las pruebas, esta asignatura me hizo experimentar en primera persona todo el proceso de desarrollo de un proyecto real.

En conclusión, mi formación en el Grado de Informática me ha dotado de un conjunto de habilidades para abordar una variedad de desafíos en las diversas áreas de la informática.

9.2 Trabajo futuro

El sistema actualmente desarrollado ha cumplido con el desarrollo y diseño de todos los requisitos que habían sido especificados al inicio del proyecto.

Con el objetivo de mejorar la experiencia de los usuarios se ha pensado una posible mejora en la aplicación. Actualmente con la aplicación se quiere centralizar el sistema de libros de todas las bibliotecas de la Comunidad Valenciana en un único sistema, en el caso de que cada una de las bibliotecas tenga su propia base de datos con los libros registrados, el paso al nuevo sistema puede resultar algo costoso. Se propone como posible mejora la capacidad de importar datos de libros directamente en la aplicación desde ficheros Excel (XLSX). De esta forma un administrador de una biblioteca deberá pasar todo el contenido de su base de datos a un Excel y al cargar dicho fichero en el nuevo sistema de manera automática leerá el contenido del archivo, a través de “mappings” deberá registrar los libros que figuraban en la biblioteca.

Por otro lado, el software se ha diseñado para uso de ordenadores. Sin embargo, es relevante señalar que en el futuro es una posibilidad considerar la adaptación del diseño para dispositivos móviles y tablets.

10. Bibliografía

- [1] HTML: Qué es y para qué sirve – VADAVO. Blog de VADAVO [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>
- [2] Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes. (2023, 25 de julio). Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web. <https://blog.hubspot.es/website/que-es-css>
- [3] Bootstrap: ¿qué es, para qué sirve y cómo instalarlo? Rock Content - ES [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://rockcontent.com/es/blog/bootstrap/>
- [4] Leaflet — an open-source JavaScript library for interactive maps. Leaflet - a JavaScript library for interactive maps [en línea]. [consultado el 12 de julio de 2023] Disponible en: <https://leafletjs.com/>
- [5] FullCalendar: cómo insertar un calendario con este plugin jQuery. También sincronizable con Google Calendar. Dreaming Bytes [en línea]. [consultado el 14 de agosto de 2023] Disponible en: <https://www.dreamingbytes.com/fullcalendar-como-insertar-un-calendario-con-este-plugin-jquery-tambien-sincronizable- con-google-calendar/>
- [6] ¿Qué es ASP.NET? | .NET. Microsoft [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet>
- [7] Herencia en C#. Microsoft Learn: Build skills that open doors in your career [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/inheritance>
- [8] Un paseo por C#: información general. Microsoft Learn: Build skills that open doors in your career. [en línea]. [consultado el 04 de agosto de 2023] <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- [9] ¿Qué es Microsoft SQL Server y para qué sirve?. Intelequia. [en línea]. [consultado el 04 de agosto de 2023] <https://intelequia.com/blog/post/que-es-microsoft-sql-server-y-para-que-sirve>
- [10] Glosario informático - Definición de términos informáticos. Glosario informático - Definición de términos informáticos [en línea]. [consultado el 13 de julio de 2023] Disponible en: https://www.glosarioit.com/Procedimiento_almacenado
- [11] ¿Qué es Firebase? ¿Qué ventajas ofrece en 2023 a nuestras apps? | SEIDOR. SEIDOR | Consultoría Tecnológica [en línea]. [consultado el 13 de julio de 2023]. Disponible en: <https://www.seidor.com/blog/firebase-que-es>
- [12] Qué es JavaScript, para qué sirve y cómo funciona. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web [en línea]. 10 de julio de 2023. Disponible en: <https://blog.hubspot.es/website/que-es-javascript>
- [13] Web Services: qué son y qué tecnología usar con ellos | Arsys. Blog de arsys.es. [en línea]. [consultado el 04 de agosto de 2023] <https://www.arsys.es/blog/web-services-desarrollo>
- [14] Diferencias entre REST y SOAP. Red Hat - We make open source technologies for the enterprise [en línea]. [consultado el 01 de julio de 2023] Disponible en: <https://www.redhat.com/es/topics/integration/whats-the-difference-between-soap-rest>

- [15] ¿Qué es Visual Studio? Microsoft Learn: Build skills that open doors in your career. [en línea]. [consultado el 24 de agosto de 2023] <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- [16] Acerca de Git - GitHub Enterprise Cloud Docs. GitHub Docs. [en línea]. [consultado el 24 de agosto de 2023] <https://docs.github.com/es/enterprise-cloud@latest/get-started/using-git/about-git>
- [17] Figma: qué es y cómo funciona | El Blog de Pixartprinting [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://www.pixartprinting.es/blog/figma-que-es/>
- [18] MESH, Janet. Metodología Kanban: revoluciona tu manera de trabajar más ágil. Trello Blog | Helping teams work better, together. [en línea]. 29 de marzo de 2020. Disponible en: <https://blog.trello.com/es/metodologia-kanban>
- [19] TeamGantt: The Easiest FREE Gantt Chart Maker Online. TeamGantt: The Easiest FREE Gantt Chart Maker Online [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://www.teamgantt.com/>
- [20] Cómo usar Postman para probar nuestras APIs. DesarrolloWeb.com [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://desarrolloweb.com/articulos/como-usar-postman-probar-api>
- [21] Inici - eBiblio Comunitat Valenciana (eBiblio). Inici - eBiblio Comunitat Valenciana (eBiblio) [en línea]. [consultado el 12 de julio de 2023] Disponible en: <https://comunitatvalenciana.ebiblio.es/>
- [22] Quiénes somos - Biblioteca Virtual. Inici - Biblioteca Virtual [en línea]. [consultado el 20 de julio de 2023] Disponible en: <https://bibliotecavirtual.diba.cat/es/qui-som>
- [23] Data Access Object (DAO) Pattern - Oscar Blancarte - Software Architecture. Oscar Blancarte - Software Architecture [en línea]. [consultado el 02 de agosto de 2023] Disponible en: <https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/>
- [24] Qué es MVC. DesarrolloWeb.com [en línea]. [consultado el 04 de agosto de 2023] Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- [25] ¿Qué es AJAX y cómo funciona? [en línea]. [consultado el 24 de agosto de 2023] Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-ajax>
- [26] ¿Qué es un diagrama de Gantt y para qué sirve? [en línea]. [consultado el 14 de agosto de 2023] OBS Business School. <https://www.obsbusiness.school/blog/que-es-un-diagrama-de-gantt-y-para-que-sirve#:~:text=El%20diagrama%20de%20Gantt%20es,las%20etapas%20de%20un%20proyecto.>
- [27] Qué es el lenguaje unificado de modelado (UML). [en línea]. [consultado el 14 de agosto de 2023] Lucidchart. <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

11. Apéndice

11.1 Manual de usuario

Este apartado contiene instrucciones detalladas sobre cómo utilizar la aplicación web BiblioConnect. Se incluyen imágenes de la interfaz para ilustrar los pasos y proporcionar una referencia visual para los usuarios finales.

11.1.1 Pantalla inicio

La aplicación comienza con la pantalla de inicio de sesión.

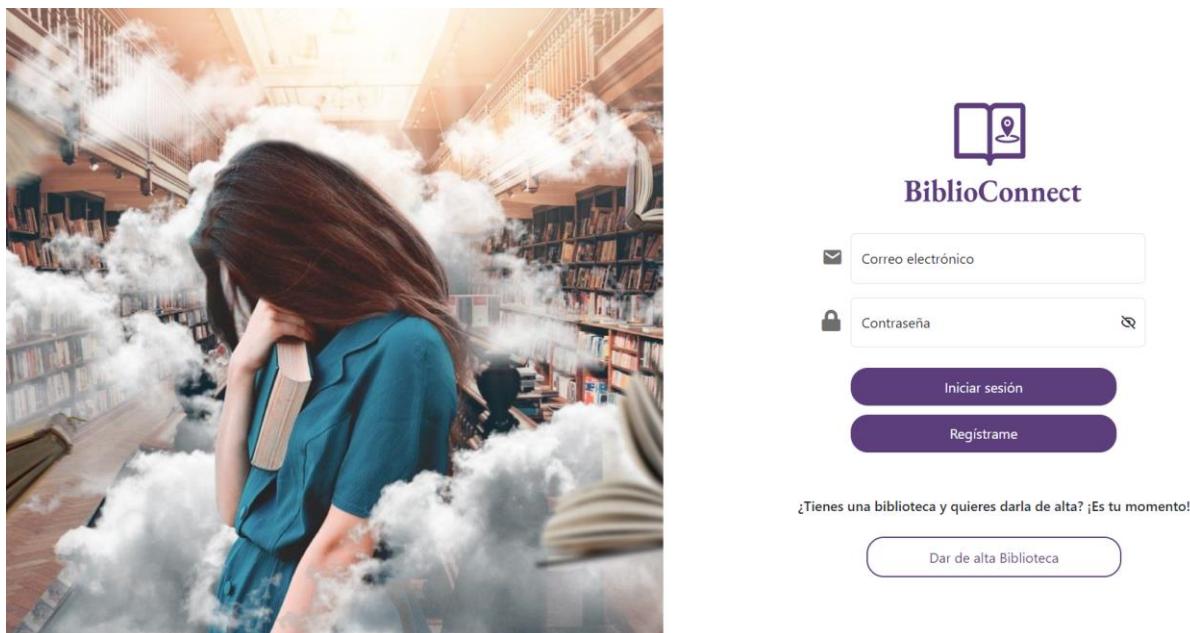


Figura 55: Vista pantalla de inicio

En la Figura 56, se muestra la ventana de inicio de sesión de la aplicación. El acceso se hace mediante una comprobación de credenciales en la base de datos. El usuario debe tener un correo electrónico y una contraseña válida para conectar con la aplicación. Si el usuario no se encuentra en la base de datos, no se podrá acceder al sistema.

En caso de que las credenciales no sean válidas aparecerá una alerta en la parte superior de los Inputs.



Figura 56: Vista credenciales incorrectas Login

Además del inicio de sesión, también se podrá realizar un registro de un nuevo lector o dar de alta una biblioteca. Para el registro de un lector habrá que presionar sobre el botón “Regístrate”. A continuación, se abrirá un modal con todos los campos necesarios para el registro.

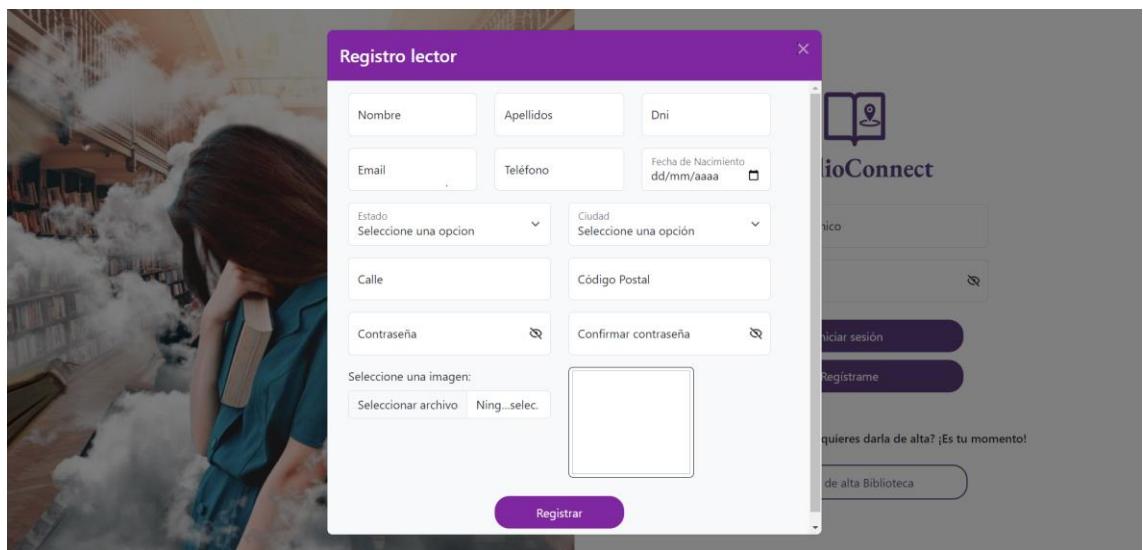


Figura 57: Vista Modal registro lector

Validación de los campos:

- Nombre: cadena de texto
- Apellidos: cadena de texto
- Dni: 8 dígitos numéricos seguidos de una letra
- Email: El sistema sólo aceptará correos electrónicos cumplen con la estructura [nombre de usuario]@[nombre de dominio].[TLD].
- Teléfono: 8 dígitos numéricos

- Fecha nacimiento: debe adaptarse a la estructura *dd/mm/aaaa* con un valor anterior al actual
- Estado y Ciudad: selección de una opción del desplegable
- Calle: cadena de texto
- Código postal: 5 dígitos numéricos
- Contraseña y Confirmar contraseña: ambos Inputs deben coincidir en los datos
- Imagen: selección de una imagen del sistema de archivos

Para dar de alta una nueva biblioteca en el sistema, habrá que presionar sobre el botón “Dar de alta Biblioteca”. A continuación, se abrirá un modal con todos los campos necesarios para el alta.

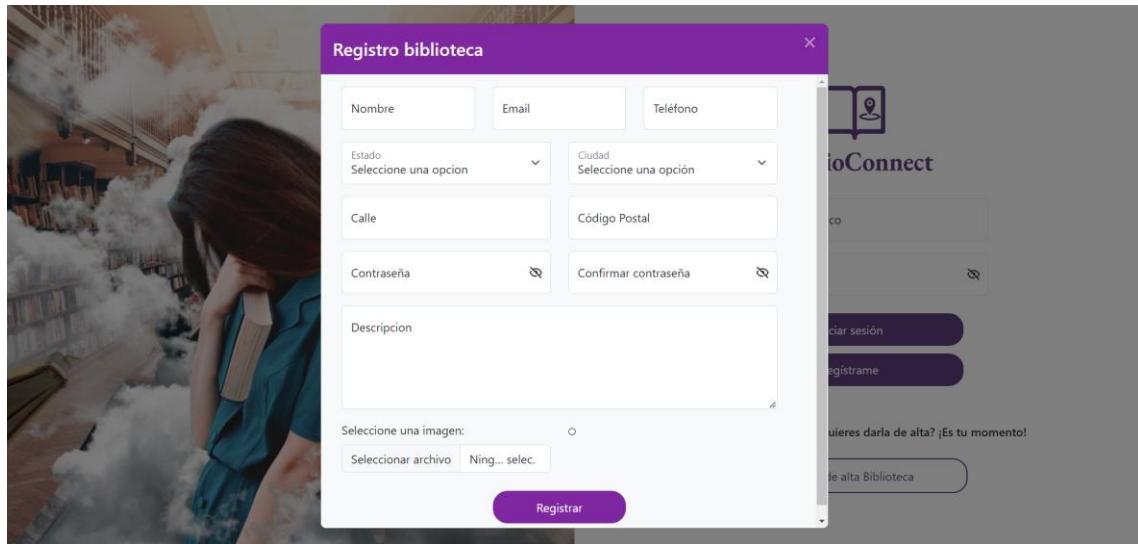


Figura 58: Vista Modal registro biblioteca

Validación de los campos:

- Nombre: cadena de texto
- Email: El sistema sólo aceptará correos electrónicos cumplen con la estructura *[nombre de usuario]@[nombre de dominio].[TLD]*.
- Teléfono: 8 dígitos numéricos
- Estado y Ciudad: selección de una opción del desplegable
- Calle: cadena de texto
- Código postal: 5 dígitos numéricos
- Contraseña y Confirmar contraseña: ambos Inputs deben coincidir en los datos
- Descripción: cadena de texto que no debe superar los 500 caracteres
- Imagen: selección de una imagen del sistema de archivos

Para el registro de usuario se han aplicado técnicas de validación personalizadas, en caso de que alguno de los campos no cumpla sus requisitos de estructura, el campo incorrecto aparecerá enmarcado de color rojo y se mostrará un mensaje de error debajo del campo.

Registro lector

- Nombre: Marta
- Apellidos: López Aguilera
- Dni: 26566761E
- Email: martita@g (Error: Tu email debe contener el simbolo @ y un dominio.)
- Teléfono: 11181 (Error: Introduce un teléfono de 9 dígitos)
- Fecha de Nacimiento: dd/mm/aaaa (Error: Seleccione una fecha)
- Estado: Valencia
- Ciudad: Paiporta
- Calle: Calle Séquia de Favara 4
- Código Postal: 46016
- Contraseña: ****
- Confirmar contraseña: ****
- Seleccionar imagen: maria.jpg

Registro biblioteca

- Nombre: Biblioteca Petxina
- Email: bibliopetxin@gmail.es
- Teléfono: 676534162
- Estado: Selecione una opción (Error: Introduce un estado)
- Ciudad: Selecione una opción (Error: Introduce una ciudad)
- Calle: Passeig de la Petxina, 41
- Código Postal: 46001
- Contraseña: ****
- Confirmar contraseña: ****
- Descripción:
- Seleccionar imagen: Petxina.jpg

Figura 59: Vista campos incorrectos

11.1.2 Menús

El sistema contiene dos tipos de menús según la sesión del usuario (administrador o lector). La sesión del administrador se ha diseñado siguiendo un enfoque elegante y profesional, a diferencia de la del lector que se ha buscado un diseño más llamativo y atractivo.

El siguiente menú, es visible por el administrador de una biblioteca.

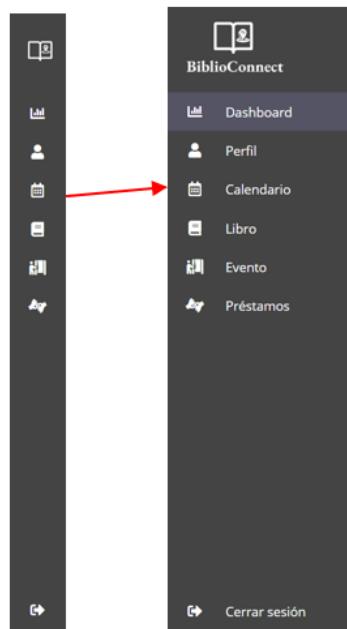


Figura 60: Vista Menú Biblioteca

El siguiente menú, es visible por cualquiera de los lectores.



Figura 61: Vista Menú Lector

11.1.3 Sesión administrador biblioteca

Cuando un administrador ha iniciado sesión, lo primero que visualiza es un dashboard que contabiliza una serie de métricas relevantes sobre la biblioteca. Este panel se actualiza de manera automática cada vez que una biblioteca realiza nuevos registros, o bien los lectores solicitan libros de la biblioteca.

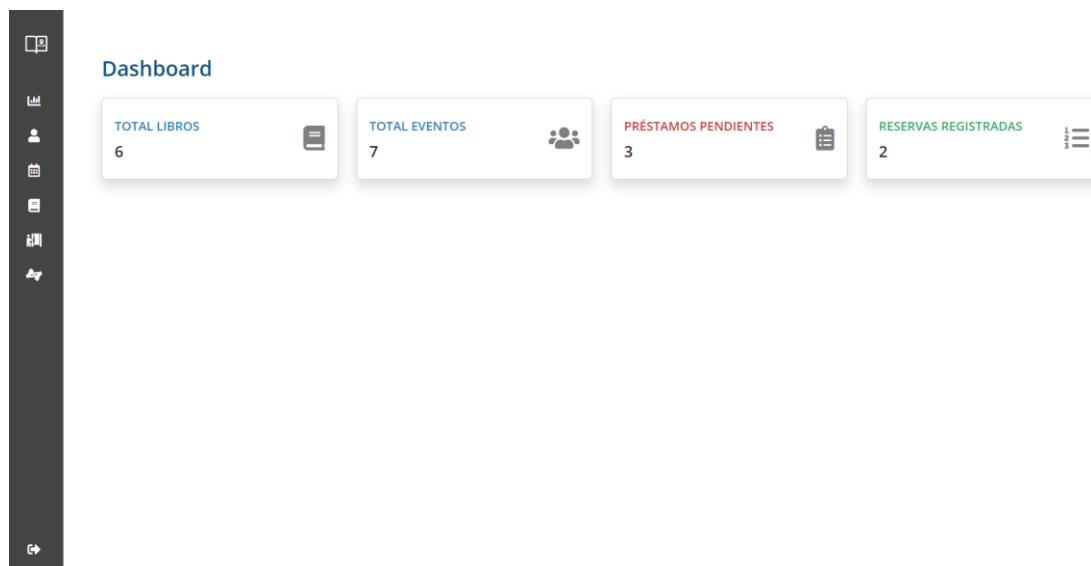


Figura 62: Vista Dashboard

Si en el menú izquierdo se selecciona la opción “Perfil” se abrirá una ventana con toda la información de contacto de la biblioteca y que será visible para cualquier lector

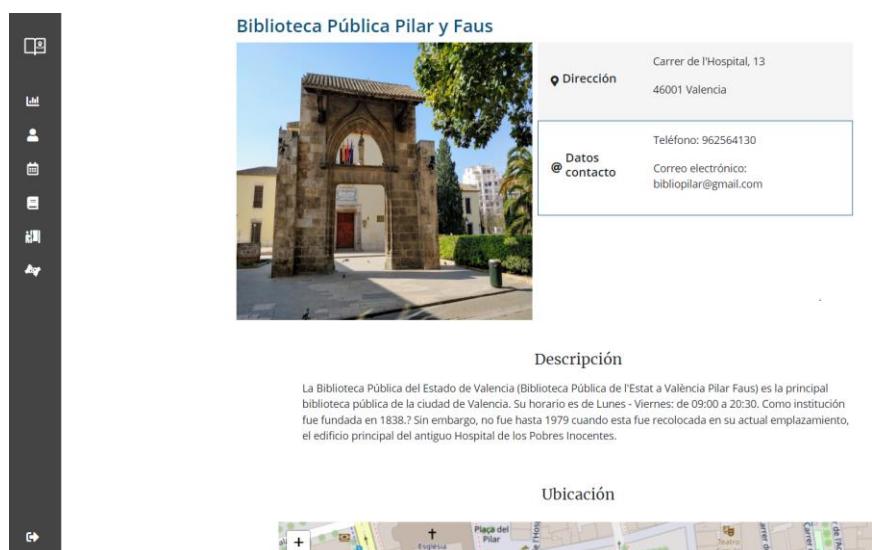


Figura 63: Vista Perfil biblioteca

Esta ventana contiene una barra de desplazamiento (scrollbar) al bajar sobre ella se mostrará un mapa interactivo con la ubicación exacta de la biblioteca. Además, de un botón que permitirá la edición de su información.



Figura 64: Vista seleccionar “Editar perfil”

Si se pulsa sobre el botón de “Editar perfil” se abrirá en esa misma ventana un modal con la información de contacto de la biblioteca. Para la modificación de la información se aplicarán técnicas de validación de campos, por lo que si al pulsar sobre el botón “Guardar” y alguno de los campos es inválido el usuario visualizará el campo marcado en rojo con un mensaje de error.

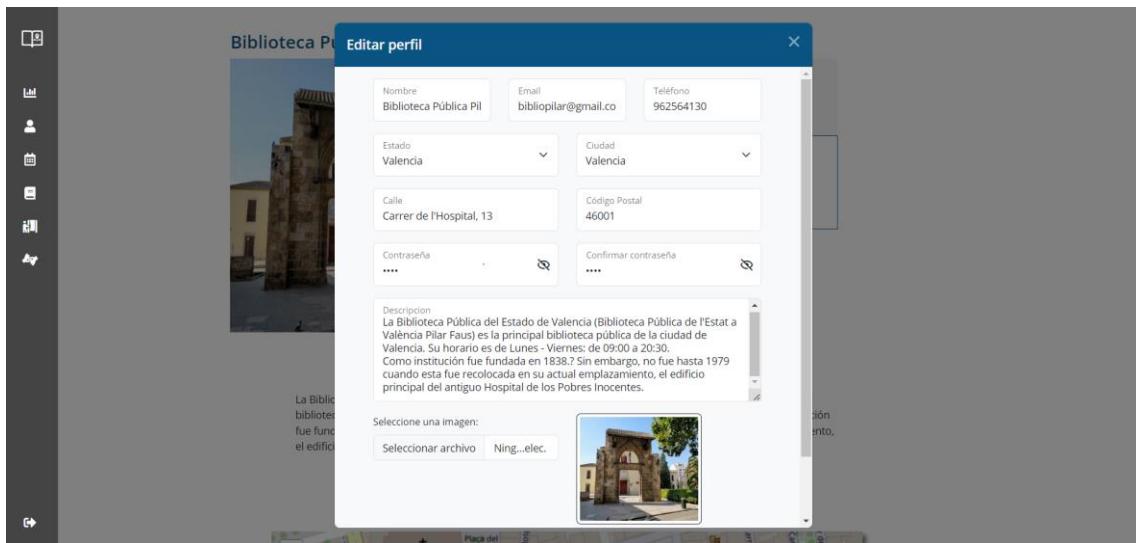


Figura 65: Vista Modal editar perfil biblioteca

Si en el menú izquierdo se selecciona la opción “Libro” se abrirá una ventana con un listado de los libros registrados por la biblioteca.

Imagen	Título	Autor	ISBN	Editorial	Categoría	Idioma	Ejemplares	Ubicación	Acciones
	Bajo la misma estrella	John Green	9780142424179	Penguin Books	Romance	Castellano	2	Planta 2, Estante 19B	
	El principio	Antoine de Saint-Exupéry	9783140464079	Emecé Editores	Infantil	Castellano	1	Planta 1, Estante 3A	
	Classical Art	Mary Beard	9780192842374	Oxford University Press	Arte	Inglés	3	Planta 2, Estante 3A	
	El hombre en busca de sentido	Viktor Frankl	9788425432026	HERDER	Crecimiento personal	Castellano	1	Planta 1, Estante 4T	

Figura 66: Vista Listado de libros desde biblioteca

Sobre este listado se permitirá la búsqueda de cualquier tipo de información de un libro (título, editorial, ISBN, Categoría, etc.). Si hay algún tipo de campo que coinciden sus caracteres con algún elemento del listado directamente aparece en tiempo real su información.

Imagen	Título	Autor	ISBN	Editorial	Categoría	Idioma	Ejemplares	Ubicación	Acciones
	Antes todo no hagas daño	Henry Marsh	9788498387209	SALAMANDRA	Medicina y salud	Castellano	2	Planta 2, Estante 10A	

Figura 67: Vista búsqueda de libros desde biblioteca

Sobre el listado de libros se pueden realizar más acciones, así como el registro de un nuevo libro. Para ello habrá que pulsar sobre el botón superior “Añadir libro”.

A continuación, se abrirá un modal con una serie de campos vacíos para el registro del nuevo libro. Para el registro se aplicarán técnicas de validación de campos, por lo que si al pulsar sobre el botón “Registrar” alguno de los campos es inválido el usuario visualizará el campo marcado en rojo con un mensaje de error. En caso de que el usuario introduzca los campos correctos y pulse registrar, se recargará de manera automática el listado y aparecerá el nuevo registro sobre él.

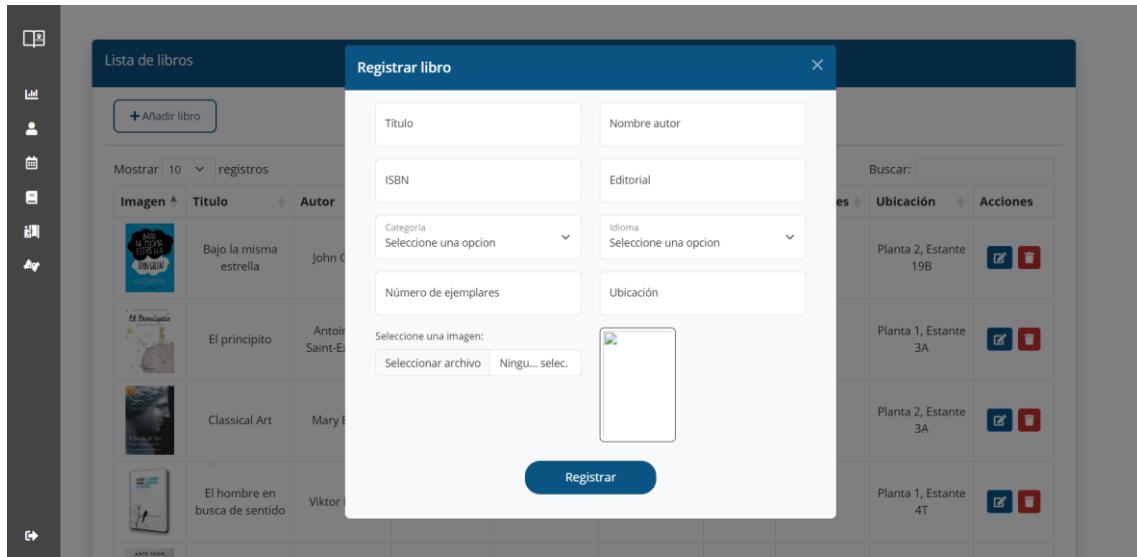


Figura 68: Vista Modal registrar libro

Otra de las acciones disponibles, es la modificación de cualquier registro, para ello habrá que seleccionar el ícono de modificar sobre el libro deseado.

Imagen	Título	Autor	ISBN	Editorial	Categoría	Idioma	Ejemplares	Ubicación	Acciones
	Bajo la misma estrella	John Green	9780142424179	Penguin Books	Romance	Castellano	2	Planta 2, Estante 19B	
	El principito	Antoine de Saint-Exupéry	9783140464079	Emecé Editores	Infantil	Castellano	1	Planta 1, Estante 3A	
	Classical Art	Mary Beard	9780192842374	Oxford University Press	Arte	Inglés	3	Planta 2, Estante 3A	
	El hombre en busca de sentido	Viktor Frankl	9788425432026	HERDER	Crecimiento personal	Castellano	1	Planta 1, Estante 4T	

Figura 69: Vista seleccionar ícono “modificar libro”

Una vez seleccionado, aparecerá un modal con toda la información del libro. Para editarlo simplemente habrá que escribir sobre el campo que se quiere modificar, posteriormente habrá que pulsar el botón “Guardar”. Para la modificación de la información se aplicarán técnicas de validación de campos, por lo que si al pulsar sobre el botón “Guardar” y alguno de los campos es inválido el usuario visualizará el campo marcado en rojo con un mensaje de error. En caso de que el usuario introduzca los campos correctos y pulse guardar, se recargará de manera automática el listado y aparecerá el registro con la información modificada.

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

Imagen	Título	Autor
	Bajo la misma estrella	John Green
	El principito	Antoine de Saint-Exupéry
	Classical Art	Mary Beard
	El hombre en busca de sentido	Viktor Frankl

Figura 70: Vista Modal modificar libro

Sobre el listado también se permite la eliminación de cualquier registro, para ello habrá que seleccionar el ícono de eliminar sobre el libro deseado.

Imagen	Título	Autor	ISBN	Editorial	Categoría	Idioma	Ejemplares	Ubicación	Acciones
	Bajo la misma estrella	John Green	9780142424179	Penguin Books	Romance	Castellano	2	Planta 2, Estante 19B	
	El principito	Antoine de Saint-Exupéry	9783140464079	Emecé Editores	Infantil	Castellano	1	Planta 1, Estante 3A	
	Classical Art	Mary Beard	9780192842374	Oxford University Press	Arte	Inglés	3	Planta 2, Estante 3A	
	El hombre en busca de sentido	Viktor Frankl	9788425432026	HERDER	Crecimiento personal	Castellano	1	Planta 1, Estante 4T	

Figura 71: Vista seleccionar ícono “eliminar libro”

En ese momento se abrirá un cuadro de diálogo que advertirá al usuario de la posterior eliminación del registro. Si el usuario pulsa sobre “Si” se realizará la eliminación con éxito, en caso contrario no se realizará ninguna acción.

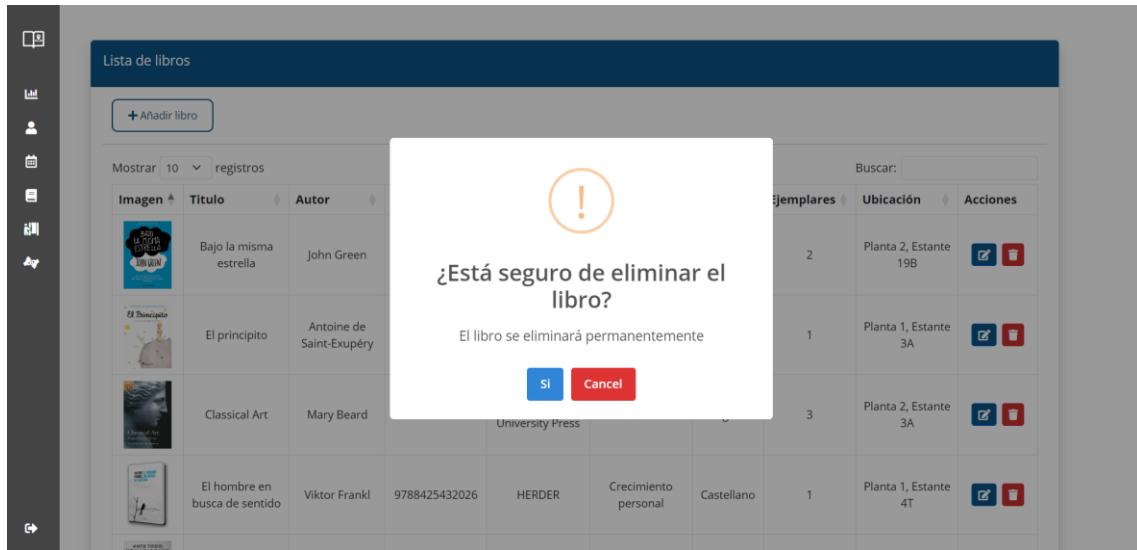


Figura 72: Vista cuadro de diálogo confirmar eliminación libro

Sobre el menú izquierdo aparece otra funcionalidad de listado similar a la del libro, que es para el caso del “Evento”. Si pulsamos sobre ella aparecerá una ventana con el listado de todos los eventos programados por la biblioteca.

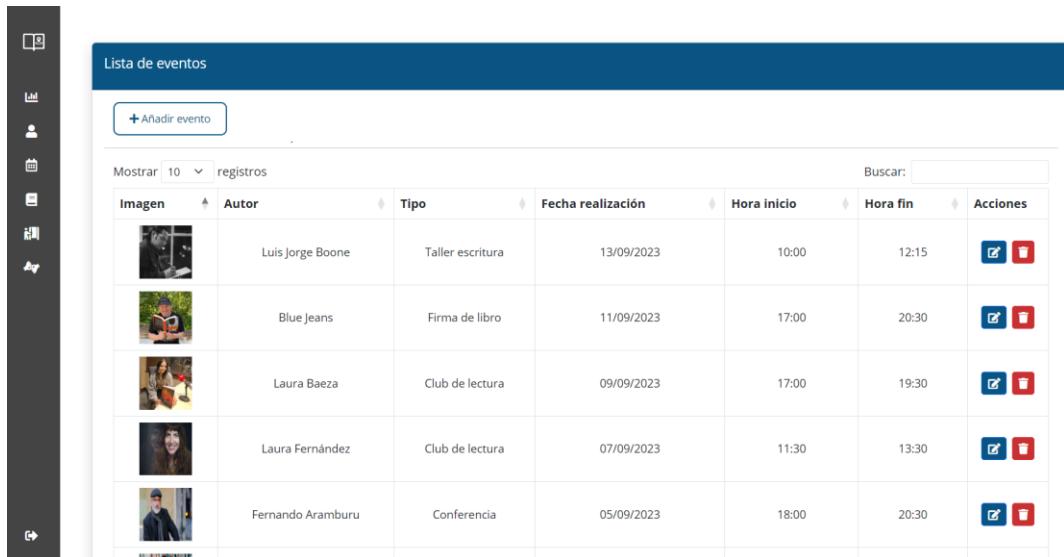


Figura 73: Vista Listado de eventos desde biblioteca

La metodología del sistema es prácticamente idéntica a la anterior, la única diferencia es que aparecen diferentes campos a la hora de registrar o modificar un evento (hora inicio, hora fin, autor, tipo, etc.).

Sobre el menú izquierdo hay una opción que es “Calendario”. Este calendario interactivo recoge automáticamente la información de cada uno de los eventos que han sido registrados (según una fecha y una hora). De esta forma, una biblioteca puede visualizar de una manera rápida y sencilla cuales son los próximos eventos que hay programados.

El calendario ofrece diferentes tipos de vista, así como el mes, semana, día o un listado completo de todos los eventos. La vista de mes muestra un calendario completo de un mes determinado, los eventos se representan como bloques en las fechas correspondientes.

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

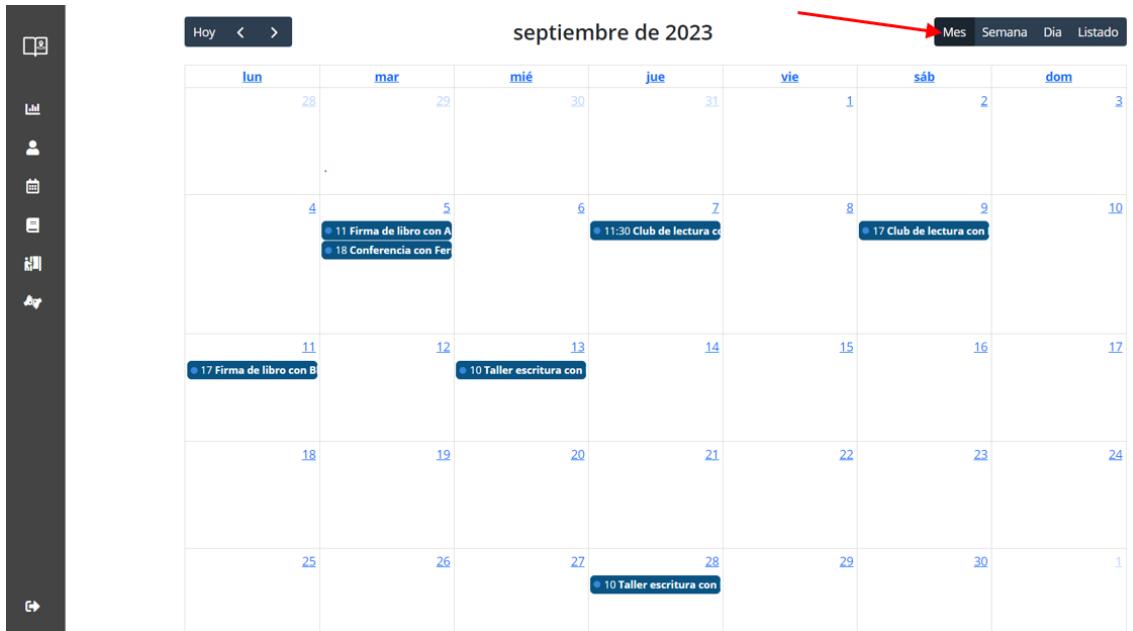


Figura 74: Vista Calendario de eventos formato “Mes”

La vista de semana muestra una semana completa, comenzando desde el lunes. Permite una vista detallada de los eventos y citas durante una semana específica.

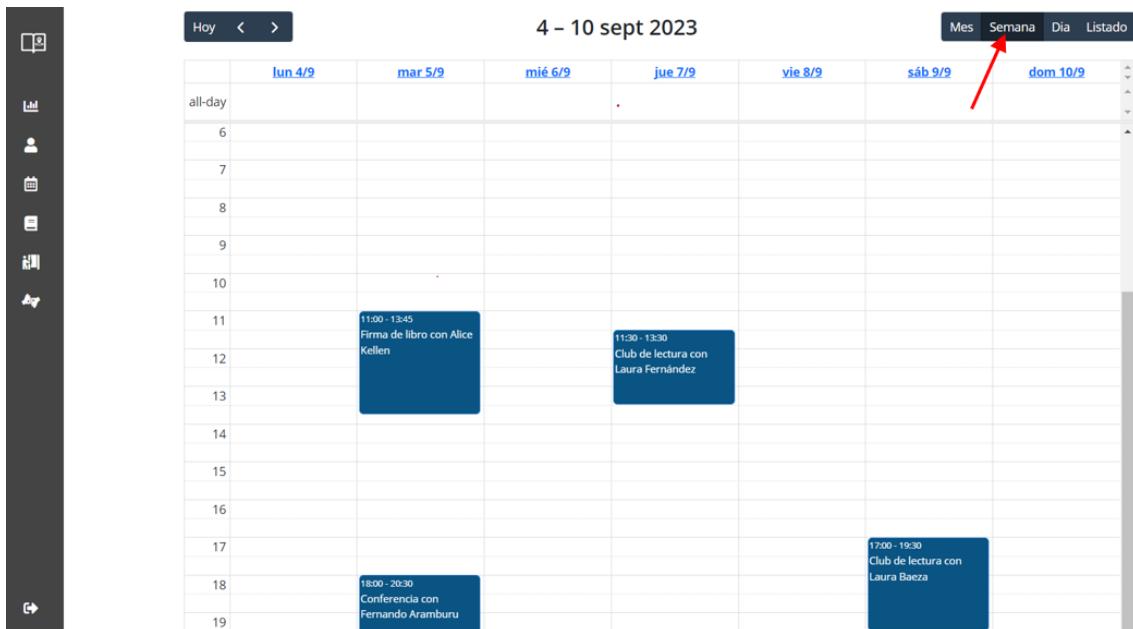


Figura 75: Vista Calendario de eventos formato “Semana”

La vista de día muestra un día específico en detalle. Aquí se puede ver todos los eventos que ocurren en ese día, lo que es útil para obtener una visión detallada de las actividades de un día en particular.

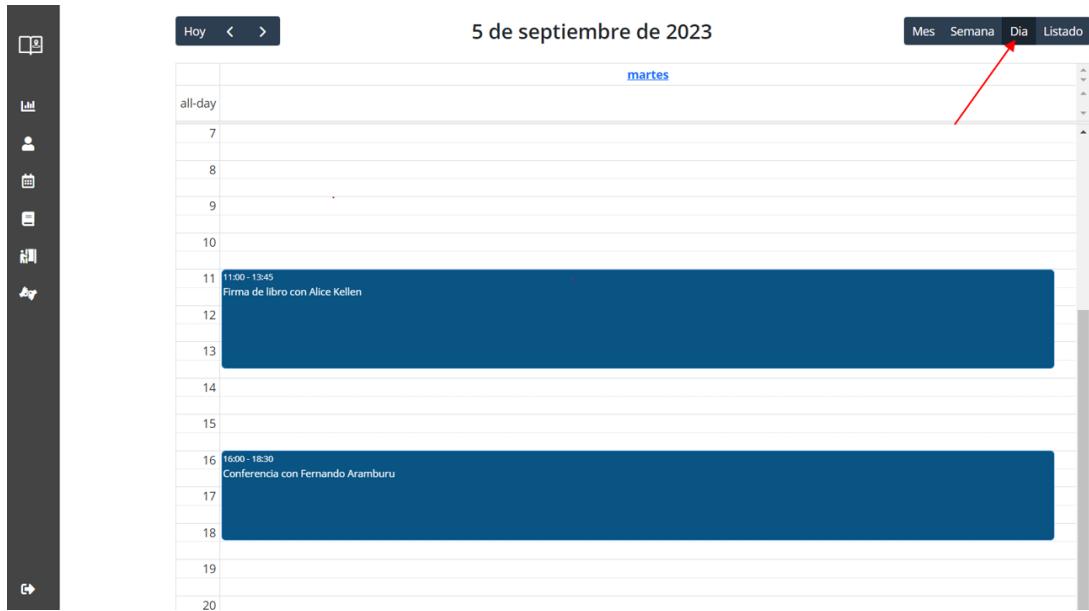


Figura 76: Vista Calendario de eventos formato “Día”

La vista de listado es una vista de lista de eventos en un formato más compacto. En esta vista, los eventos se muestran en forma de lista, lo que es útil cuando se necesita una visión general de los eventos en orden cronológico sin necesidad de un formato de calendario.



Figura 77: Vista Calendario de eventos formato “Listado”

Si se pulsa sobre cualquier evento en el calendario desde cualquier tipo de vista (día, semana, mes o listado) se abrirá un modal con la información del evento.

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

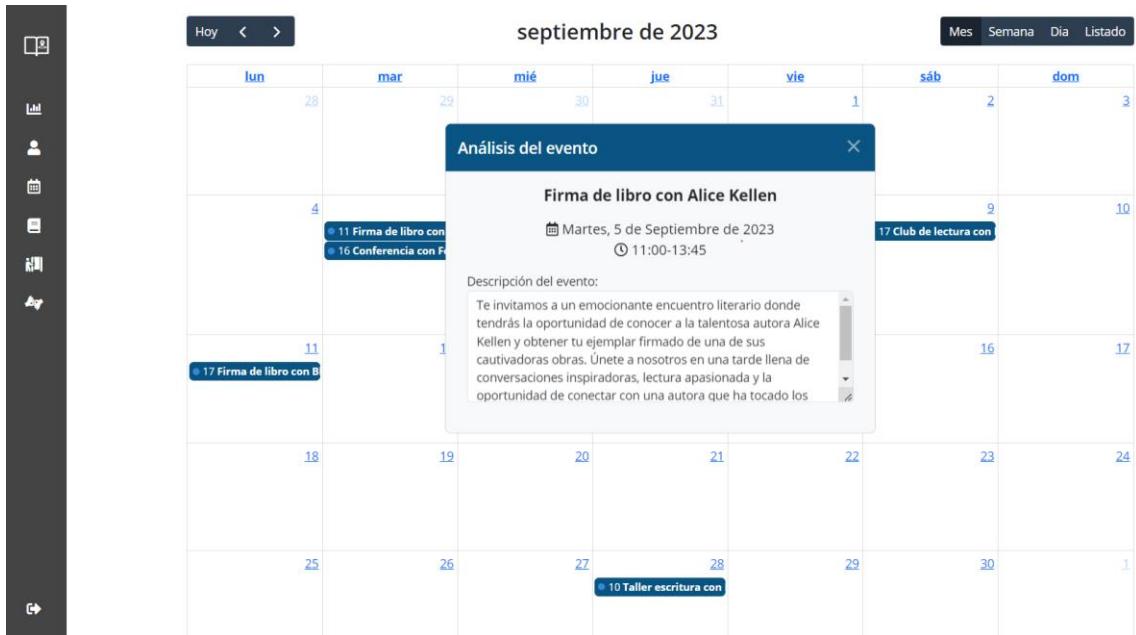


Figura 78: Vista Modal información evento en calendario

Si en el menú izquierdo se selecciona la opción “Préstamos” se abrirá una ventana con un listado de los préstamos de libros que los lectores han solicitado de la biblioteca. En el sistema actual hay una sección de préstamos según el tipo de usuario que ha iniciado sesión se visualizará una vista u otra. Para identificar en qué situación está un lector con un libro se ha asignado el concepto de “estado”. El estado de un préstamo tiene 4 posibilidades: en espera, pendiente de recogida, pendiente devolución y devuelto.

En el grupo de botones existe una opción que es “Todos”, esta selección recoge todos los préstamos que se han realizado para la biblioteca según cualquier tipo de estado.

Lista de préstamos								
Estado		Título	Nombre	Apellidos	Dni	Fecha devolución	Fecha confirmada	Acciones
Devuelto		Bajo la misma estrella	Lucía	Santos González	58977613A	09/05/2023	04/05/2023	
Devuelto		El principito	Marta	Tejeda Fernández	24355143T	09/06/2023	03/06/2023	
Devuelto		El hombre en busca de sentido	Juan	Pérez Martín	56544123E	09/05/2023	04/05/2023	
Devuelto		La historia de los tres cerditos	Carlos	Poveda Martínez	47544221R	09/07/2023	04/07/2023	
Devuelto		Bajo la misma estrella	Juan	Pérez Martín	56544123E	09/07/2023	08/07/2023	
En espera		Bajo la misma estrella	Carlos	Poveda Martínez	47544221R	-	-	
Pendiente devolución		Classical Art	Carlos	Poveda Martínez	47544221R	11/09/2023	-	
Pendiente recogida		El hombre en busca de sentido	Lucía	Santos González	58977613A	-	-	
Pendiente recogida		Bajo la misma estrella	Marta	Tejeda Fernández	24355143T	-	-	

Figura 79: Vista Listado de todos los préstamos desde biblioteca

Desde el listado se permite la búsqueda de cualquiera de los campos que aparece en la tabla (nombre, apellidos, dni, fecha devolución, etc.). En caso de que haya datos que correspondan a la búsqueda aparecerán de manera automática en el listado.

The screenshot shows a search interface for loans. At the top, there's a navigation bar with icons for home, user, calendar, and search. Below it is a header 'Lista de préstamos' with tabs: 'Todos', 'En espera', 'Pendiente recogida', 'Pendiente devolución', and 'Devuelto'. A red arrow points from the left towards the search bar. The search bar contains the text 'Buscar: 47544221R' and has a red circle around it. Below the search bar is a table with columns: Estado, Título, Nombre, Apellidos, Dni, Fecha devolución, Fecha confirmada, and Acciones. The table shows three rows of data. The first row has 'Devuelto' in the Estado column. The second row has 'Pendiente devolución'. The third row has 'Pendiente recogida'. All three rows have '47544221R' in the Dni column, which is also circled in red. At the bottom of the table, it says 'Mostrando registros del 1 al 3 de un total de 3 registros (filtrado de un total de 9 registros)'. On the right side, there are buttons for 'Anterior' (page 1), 'Siguiente', and a small number '1'.

Figura 80: Vista búsqueda de préstamos

Si se selecciona el botón “En espera” aparecerá un listado con los lectores que han reservado un libro que no está disponible, pero en el momento en que él sea devuelto, el estado del lector en espera pasará a “pendiente de recogida” de manera automática, al mismo tiempo en que el lector será **notificado** para su recogida.

This screenshot shows the same interface as Figure 80, but with the 'En espera' tab selected. The table now displays only one row of data. The first column, 'Estado', shows 'En espera'. The other columns show the loan details: 'Bajo la misma estrella', 'Carlos', 'Poveda Martínez', '47544221R', and '-' for both 'Fecha devolución' and 'Fecha confirmada'. The bottom message indicates 'Mostrando registros del 1 al 1 de un total de 1 registros'.

Figura 81: Vista listado de préstamos “En espera”

Al seleccionar el botón “Pendiente recogida” se obtiene un listado con los libros que ya está disponible su recogida por el lector.

Figura 82: Vista listado de préstamos “Pendiente recogida”

En caso de que el lector haya recogido el libro, el administrador de la biblioteca deberá seleccionar el ícono que aparece en el campo “Acciones”.

Posteriormente, se abrirá un modal informativo con los detalles del lector y el libro que va a ser recogido. En caso de que la información sea correcta se seleccionará el botón “Asignar” y el libro pasará al siguiente estado “pendiente devolución”. Por otro lado, si todavía no se ha recogido el libro se podrá seleccionar el botón que representa una cruz y no se realizará ninguna acción.

Figura 83: Vista Modal recogida del libro

Al seleccionar el botón “Pendiente devolución” se obtiene un listado con los libros que han sido prestados a un lector.

- Pendiente devolución:** este estado espera a que el lector recoja físicamente el libro en la biblioteca, en ese momento el administrador de la biblioteca deberá confirmar un

formulario con la información del lector acreditando que el libro ha sido recogido por el lector.

Figura 84: Vista listado de préstamos “Pendiente devolución”

En caso de que el lector haya devuelto el libro, el administrador de la biblioteca deberá seleccionar el icono que aparece en el campo “Acciones”.

Aparecerá un modal con dos comentarios, uno estará deshabilitado y representará el estado en el que el libro fue entregado. El otro deberá ser completado por la biblioteca y será el estado en que el lector ha devuelto el libro.

Figura 85: Vista Modal devolución del libro

Al seleccionar el botón “Devuelto” se obtiene un listado con los libros que han sido devueltos por los lectores.

Estado	Título	Nombre	Apellidos	Dni	Fecha devolución	Fecha confirmada	Acciones
Devuelto	Bajo la misma estrella	Lucía	Santos González	58977613A	09/05/2023	04/05/2023	
Devuelto	El principio	Marta	Tejeda Fernández	24355143T	09/06/2023	03/06/2023	
Devuelto	El hombre en busca de sentido	Juan	Pérez Martín	56544123E	09/05/2023	04/05/2023	
Devuelto	La historia de los tres cerditos	Carlos	Poveda Martínez	47544221R	09/07/2023	04/07/2023	
Devuelto	Bajo la misma estrella	Juan	Pérez Martín	56544123E	09/07/2023	08/07/2023	
Devuelto	Bajo la misma estrella	Marta	Tejeda Fernández	24355143T	11/09/2023	21/08/2023	

Mostrando registros del 1 al 6 de un total de 6 registros

Anterior 1 Siguiente

Figura 86: Vista listado de préstamos “Devueltos”

En caso de que el administrador quiera visualizar el estado de devolución del libro por un lector, deberá seleccionar el ícono que aparece en el campo “Acciones”. A continuación, se abrirá un modal con la información de devolución del libro.

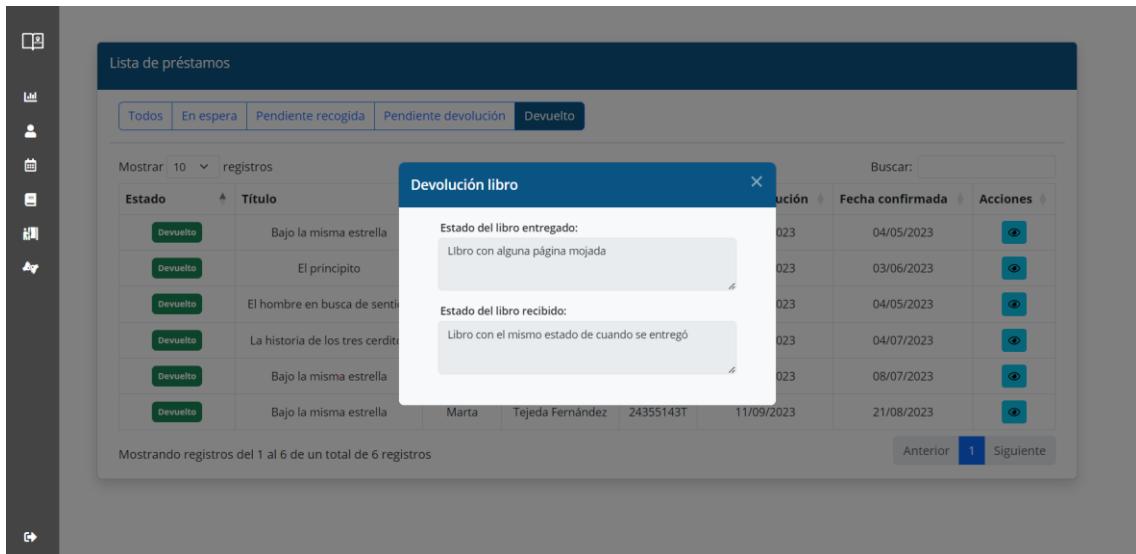


Figura 87: Vista Modal estado del libro devuelto

11.1.4 Sesión lector

Cuando un lector inicia sesión, la primera página que se abrirá será la que corresponde a la pestaña de “Inicio”. Esta sección contiene una bienvenida a la aplicación, además de un breve resumen de la funcionalidad que se ofrece.

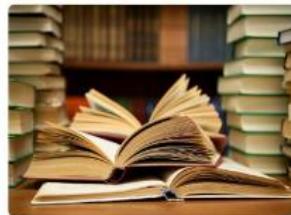
BiblioConnect

Bienvenido a BiblioConnect, la aplicación que te conectará con las bibliotecas de la Comunidad Valenciana directamente desde tu dispositivo. ¡Descubre libros, eventos y mucho más, todo en un mismo lugar!



Geolocalización para encontrar libros cercanos

BiblioConnect utiliza la geolocalización para mostrarte los libros en papel disponibles más cercanos a ti.



Reserva libros no disponibles y recibe notificaciones

Si el libro que deseas no está disponible, resérvalo y recibe notificaciones cuando esté listo para recoger. ¡No pierdas la oportunidad de leer ese libro que tanto quieras!



Filtrado avanzado

Encuentra fácilmente libros por género, autor o título utilizando nuestro avanzado sistema de filtrado.



Asiste a eventos literarios organizados por las bibliotecas

No te pierdas ni un solo evento literario en tu zona. Consulta fechas y ubicaciones de eventos gratuitos organizados por bibliotecas de la Comunidad Valenciana. ¡Únete a actividades gratuitas como clubes de lectura, firmas de libros, presentaciones de autores y mucho más!



Reseñas de libros

Las reseñas en BiblioConnect te permiten leer y escribir opiniones sobre los libros que te interesan. Conoce lo que otros piensan antes de tomar la decisión.

Figura 88: Vista pestaña Inicio desde lector

En el menú que aparece en la parte superior se puede seleccionar la opción de “Catálogo” que contiene un listado de todos los libros que han registrado las bibliotecas en el sistema. Cada uno de estos libros contiene la distancia en kilómetros del lector actual al libro.

The screenshot shows the BiblioConnect library catalog interface. At the top, there's a purple header bar with the title "BiblioConnect" and navigation links for "Inicio", "Catálogo" (which is highlighted), "Eventos", and "Préstamos". There are also icons for notifications (with 1 notification), user profile, and a search function.

Filtros

BÚSQUEDA

Título, Autor, ISBN, Edito

CATEGORÍAS

- Arte
- Ciencia y tecnología
- Ciencias sociales
- Crecimiento personal
- Ficción
- Romance
- Medicina y salud
- Infantil
- Humanidades

Libros

1-15 de 15 resultados

A grid of 15 book covers is displayed, each with a small location pin icon indicating its availability at the Valencia branch. The books include titles like "BAJO LA MISMA ESTRELLA" by John Green, "El Principito", "Classical Art", "El hombre en busca de sentido", "HENRY MARSH", "La historia de los tres cerditos", "Harry Potter y la piedra filosofal", "El viaje de la Humanidad", "BAJO LA MISMA ESTRELLA" (repeated), "EL DIOS MÁS PODEROSO", "Vida de Walt Whitman", "CARL ZIMMER", "She has her mother's laugh", and "LA NOIA DEL TREBALL" by Paula Hawkins.

Figura 89: Vista Catálogo de libros desde lector

En la parte izquierda aparece una sección con los posibles filtros. En el aparece un botón de “Eliminar filtros” que eliminará en cualquier momento los filtros que hayan sido aplicados y se volverá al listado de libros inicial.

This is a modal window titled "Filtros" (Filters). It contains sections for "BÚSQUEDA" (Search) and "CATEGORÍAS" (Categories). The categories listed are Arte, Ciencia y tecnología, Ciencias sociales, Crecimiento personal, Ficción, Romance, Medicina y salud, Infantil, and Humanidades. Below these is a section for "IDIOMA" (Language) with options for Castellano, Inglés, and Catalán. At the bottom is a large blue button labeled "Eliminar Filtros" (Remove Filters).

Figura 90: Vista Menú filtrado de libros

El filtrado se puede realizar seleccionando una categoría y un idioma de manera conjunta, o bien por separado. Al seleccionar alguno de los botones de los filtros de manera automática aparecerán en el listado tan solo los libros que se correspondan con las preferencias marcadas.

The screenshot shows the BiblioConnect interface. On the left, there is a sidebar titled 'Filtros' (Filters) containing several categories and languages. Under 'CATEGORÍAS', the following options are listed: Arte, Ciencia y tecnología, Ciencias sociales, Crecimiento personal, Ficción, Romance, Medicina y salud, Infantil, and Humanidades. Under 'IDIOMA', the options are Castellano, Inglés, and Catalán. A red arrow points from the text 'Figura 91: Vista Filtrado de libros' to the search results area. The main content area is titled 'Libros' and shows '1-2 de 2 resultados'. It displays two book covers: 'EL DIOS MÁS PODEROSO' by 'VIDA DE WALT WHITMAN' and 'VENENO' by 'PAUL THIRION'. Below each cover, the title and author are listed, along with the location 'Vila-real' or 'Alacant'.

Figura 91: Vista Filtrado de libros

También se permite una búsqueda por ISBN, título del libro, autor o editorial.

The screenshot shows the BiblioConnect interface with a search bar in the top left. The search term 'Matilda' is highlighted with a red circle and a red arrow points from the text 'Figura 92: Vista Búsqueda de libros' to this area. The main content area is titled 'Libros' and shows '1-15 de 15 resultados'. It displays one book cover for 'ROALD DAHL MATILDA'. Below the cover, the title and author are listed, along with the location 'Alacant'.

Figura 92: Vista Búsqueda de libros

En el momento en que se presione alguno de los libros, aparecerá una página con toda la información relevante al libro.

BiblioConnect
Inicio
Catálogo
Eventos
Préstamos



Bajo la misma estrella

John Green

Bajo la misma estrella es una novela escrita por John Green, la sexta en su carrera como autor literario, publicada el 10 de enero de 2012. La historia es narrada por una adolescente de 16 años de edad llamada Hazel Grace Lancaster, quien padece cáncer de tiroides que acaba haciendo metástasis y trasladándose a los pulmones. Por ese motivo, tiene que usar siempre un tanque de oxígeno, al que ella llama Philip. Sus padres la obligan a acudir a un grupo de apoyo para jóvenes afectados por la enfermedad situado en el sótano de una iglesia, llamada El corazón literal...

[Mostrar más](#)

El ejemplar no está disponible, résérvalo y te avisaremos cuando lo esté.

[Reserva](#)

Editorial

Penguin Books

Idioma

Castellano

Categoría

Romance

ISBN

90642832379

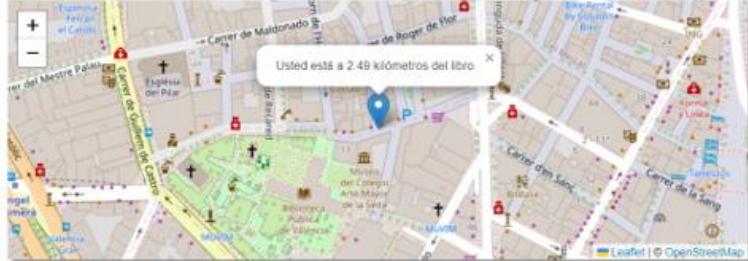


Información sobre el autor

John Michael Green es un escritor estadounidense de literatura juvenil, blogger en YouTube y productor ejecutivo. Es reconocido mayormente por su libro Bajo la misma estrella y Ciudades de papel. Sus padres, quienes lo apoyaron con su decisión, fueron Mike Green y Sydney Green. Su padre era director de un...

[Mostrar más](#)

Ubicación biblioteca



Biblioteca Pública Pilar y Faus

Usted está a 2.49 kilómetros del libro

Figura 93: Vista información detalle de un libro

La descripción del libro y del autor se obtienen a partir del uso de la API de Wikipedia, por lo que puede ocurrir que su contenido sea demasiado largo. Por eso se ha desarrollado la funcionalidad de “Mostrar más” y “Mostrar menos”. Los lectores pueden expandir o reducir el contenido según su preferencia. Esto permite acceder a información detallada sin abrumar al usuario con contenido demasiado extenso de manera predeterminada.

86



Bajo la misma estrella

John Green

Bajo la misma estrella es una novela escrita por John Green, la sexta en su carrera como autor literario, publicada el 10 de enero de 2012. La historia es narrada por una adolescente de 16 años de edad llamada Hazel Grace Lancaster, quien padece cáncer de tiroides que acaba haciendo metástasis y trasladándose a los pulmones. Por ese motivo, tiene que usar siempre un tanque de oxígeno, al que ella llama Philip. Sus padres la obligan a acudir a un grupo de apoyo para jóvenes afectados por la enfermedad situado en el sótano de una iglesia, llamada El corazón literal...

[Mostrar más](#)

Ficha Técnica

Editorial	Idioma	Categoría	ISBN
Penguin Books	Castellano	Romance	90642832379

Figura 94: Vista “Mostrar más”

Al pulsar sobre “Mostrar más” el contenido se expande.



Bajo la misma estrella

John Green

Bajo la misma estrella es una novela escrita por John Green, la sexta en su carrera como autor literario, publicada el 10 de enero de 2012. La historia es narrada por una adolescente de 16 años de edad llamada Hazel Grace Lancaster, quien padece cáncer de tiroides que acaba haciendo metástasis y trasladándose a los pulmones. Por ese motivo, tiene que usar siempre un tanque de oxígeno, al que ella llama Philip. Sus padres la obligan a acudir a un grupo de apoyo para jóvenes afectados por la enfermedad situado en el sótano de una iglesia, llamada El corazón literal de Jesús, en la cual conoce y se enamora a un joven de 17 años llamado Augustus Waters, exjugador de baloncesto que tiene amputada una pierna a causa del osteosarcoma. Green expresa que el título del libro se inspiró en la famosa frase de la obra «Julio César», de William Shakespeare, cuando Casio, en el Acto 1/Escena 2 de la obra, le dice a Brutus: «[La culpa, querido Brutus, no es de nuestras estrellas, sino de nosotros mismos, que consentimos en ser inferiores]. La adaptación cinematográfica de la novela, de nombre homónimo, se estrenó el 6 de junio del 2014, en los Estados Unidos, protagonizada por Shailene Woodley y Ansel Elgort. El estreno en España fue el 15 de julio, mientras que en varios países de Latinoamérica su estreno tuvo lugar a fines de junio. Desde la opinión popular se han generado críticas generalmente positivas hacia la obra, considerada en la actualidad una gran novela del actualmente muy popular género young adult. Se le ha dado gran importancia a la enseñanza y los valores morales que el libro intenta transmitir al lector. También suele alabarse la construcción de los personajes, aunque aun así hay críticas varias que difieren respecto a la solidez de la misma. Tras su gran éxito y la escasa cantidad de críticas negativas, esta novela ha comenzado a ganar una imagen de obra sobrevalorada dentro de cierto sector en la opinión popular. Si se toma como válida esta sobrevaloración, podría afirmarse que la moda, la valoración popular, la propagación de la novela dentro de las redes sociales y el éxito de la película contribuyeron en gran medida. La fuente de inspiración del autor fue Esther Grace Earl, quien padecía cáncer de tiroides y murió el 25 de agosto del 2010, a los 16 años. Entabló con ella una gran amistad, y se conocieron en un hospital infantil. Al respecto, Green pidió que no confundieran a Hazel y Esther, y que ambas son completamente distintas, solo que esta última ayudó al autor a desarrollar el personaje.

[Mostrar menos](#)

Figura 95: Vista “Mostrar menos”

A la derecha de la descripción del libro aparece una tarjeta que permite el préstamo o reserva del libro. En caso de que el libro esté disponible, el botón contendrá la palabra “Préstamo”, en caso contrario contendrá la palabra “Reserva”. En la Figura 96, se muestra el transcurso de cuadros de diálogo según cada opción.

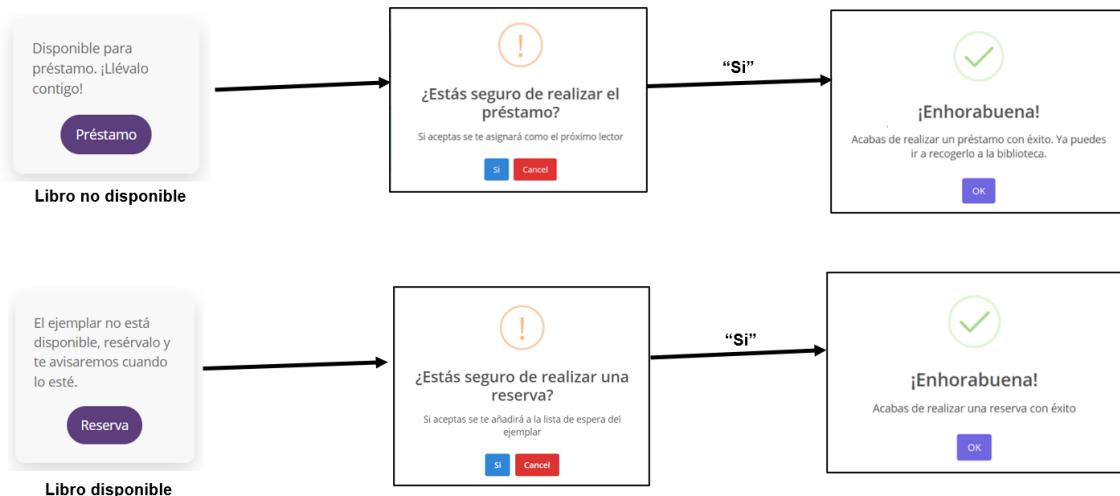


Figura 96: Vista Tarjeta Préstamo/Reserva de un libro

Esta página contiene un scrollbar vertical que permite su desplazamiento hacia abajo. El contenido que le sigue es la descripción del autor y un mapa web interactivo con la ubicación exacta a la biblioteca que contiene el libro. Además, contiene un botón que representa un enlace al perfil de la biblioteca con toda su información de contacto.



Figura 97: Vista seleccionar enlace perfil biblioteca

Si se continúa el desplazamiento hacia abajo en la misma página aparece una sección de reseñas del libro.

Reseñas de los lectores

4/5

★★★★★

[Escribir reseña](#)

(4 opiniones de lectores)

 Juan Este libro no tiene criterio, toca temas muy densos como el cáncer y la muerte a la ligera. Muestra la vida de los jóvenes con cáncer como si fuera fácil, color de rosa. Es surrealista, la protagonista a la hora de conocer a Augustus.
★★★★★
19/08/2023

 Lucía ¡Este libro es totalmente perfecto! Me hizo entender un poco como es que se sienten las personas que están enfermas en general y que saben que en algún momento van a partir de nuestro lado. Además, me ayudó a comprender que todos nos afortunados de alguna manera y que debemos de vivir a vida al instante, disfrutarla al máximo y no esperar para simplemente ser felices.
★★★★★
19/05/2023

 Carlos Es una historia que te engancha desde que inicias a leer a Hazel, adoras la manera en que se conduce y logras entender porque actúa de cierto modo. Conoces la felicidad con un toque de amargura porque nunca sabes cuando será la última vez que tenga la oportunidad de realmente disfrutar del momento, de algo, de alguien...
★★★★★
19/08/2023

 Marta Nunca había llorado por libro, pero este me hizo hasta hacer pucheros, tiene demasiados sentimientos y guau que hermoso libro, me hizo ver la vida de otra forma y no debemos a esperar a estar enfermos para recién cumplir nuestros sueños.
★★★★★
19/08/2023

Figura 98: Vista Reseñas de un libro

Si se selecciona el botón “Escribir reseña” se permite la escritura de una nueva. Como autor de la reseña se obtiene el nombre y la fotografía correspondiente al lector que ha iniciado sesión.

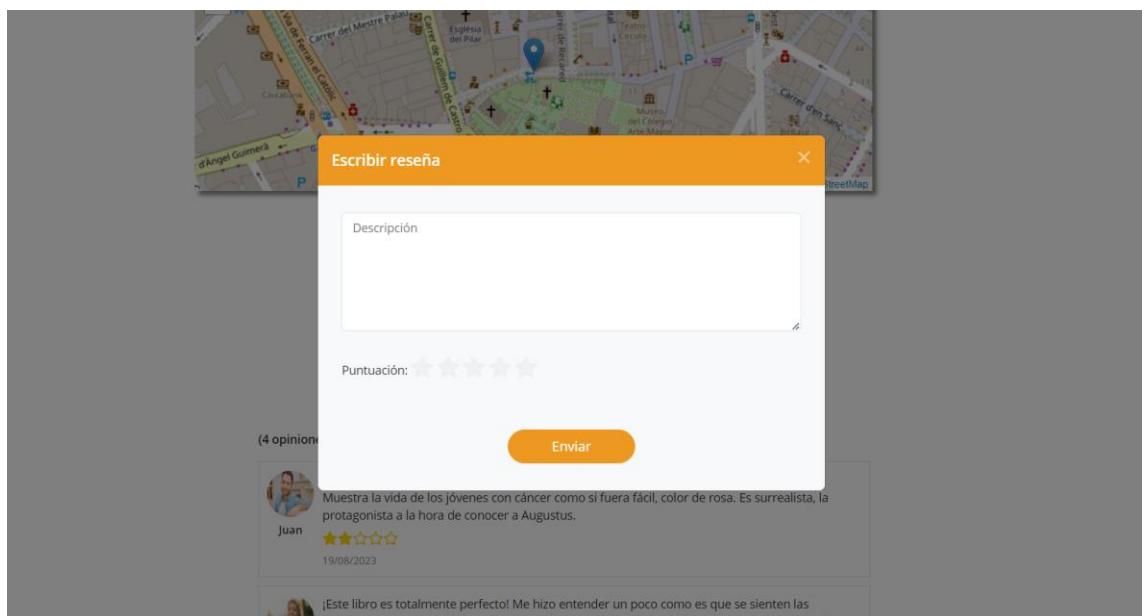


Figura 99: Vista Modal escritura de una reseña

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

Otra de las opciones disponibles en la sesión de un lector, es la sección de “Eventos” al pulsar sobre ella se mostrará una página con el listado de todos los eventos que se han programado por las bibliotecas. El catálogo de eventos permite la selección de filtros con un enfoque similar al del catálogo de libros.

Filtros

TIPO

Conferencia

Presentación libro

Club de lectura

Taller escritura

Firma de libro

Eliminar Filtros

BiblioConnect

Inicio Catálogo Eventos Préstamos

Eventos

1-10 de 10 resultados

Taller escritura en la Biblioteca Pública Pilar y Faus
Luis Jorge Boone
Miércoles, 13 de Septiembre de 2023
10:00-12:15

En este taller único y enriquecedor, el escritor y maestro Luis Jorge Boone te guiará a través de un emocionante viaje hacia la creación literaria. Aprenderás las técnicas esenciales para desarrollar historias impactantes, personajes memorables ...

Valencia

Club de lectura en la Biblioteca Virgen Del Remedio
Chimamanda Ngozi
Sábado, 16 de Septiembre de 2023
11:00-12:30

Chimamanda Ngozi Adichie nació en 1977 en Nigeria. A día de hoy Literatura Random House ha publicado sus tres novelas: La flor púrpura, ganadora del Commonwealth Writers' Prize y el Hurston / Wright Legacy Award; Medio sol amarillo, galardonada ...

Alacant

Figura 100: Vista Listado de Eventos desde lector

Al seleccionar alguno de los eventos del listado se abrirá una página con la información detallada del evento.

BiblioConnect

Inicio Catálogo Eventos Préstamos

Información sobre el evento

Firma de libro

Lunes, 11 de Septiembre de 2023 17:00-20:30

¡Amantes de la literatura y seguidores de Blue Jeans, prepárense para una experiencia única e inolvidable! Te invitamos a un emocionante evento de firma de libros en el que tendrás la oportunidad de conocer al reconocido autor Blue Jeans en persona y sumergirte en el mundo de sus cautivadoras historias.

Información sobre el autor

Blue Jeans
Blue Jeans, seudónimo de Francisco de Paula Fernández, nació en Sevilla y vivió hasta los dieciocho años en Carmona. Licenciado en Periodismo, es autor de las trilogías Canciones para Paula (Canciones para Paula, ¿Sabes que te quiero? y Cállame con un beso), El Club de los Incomprendidos, La chica invisible, etc.

Ubicación biblioteca

Biblioteca Pública Pilar y Faus

Usted está a 2.48 kilómetros del evento

Leaflet | OpenStreetMap

Figura 101: Vista información sobre un evento

Un lector puede visualizar el historial de libros que ha solicitado su préstamo pulsando la pestaña “Préstamos”. En esta página aparecerá el listado con todos los libros que el lector ha solicitado, además aparecerá el estado en el que se encuentran (devuelto, pendiente recogida, en espera, pendiente devolución). En caso de que el libro esté pendiente su devolución, se mostrará la máxima fecha permitida en que el libro debe ser devuelto.

Mis préstamos

	Bajo la misma estrella John Green Entregado: Miércoles, 19 de Abril de 2023 Devuelto
	El hombre en busca de sentido Viktor Frankl Pendiente recogida *El libro ya está disponible para ser recogido en la biblioteca.
	Veneno Pablo Trincia Entregado: Lunes, 21 de Agosto de 2023 Pendiente devolución (Lunes, 11 de Septiembre de 2023)

Figura 102: Vista historial de los préstamos de un lector

El sistema contiene una sección de notificaciones para cada uno de los lectores. Se puede acceder a este listado pulsando sobre la pestaña que contiene el ícono de la notificación. Un lector será notificado cuando un determinado libro que ha solicitado está disponible para su recogida. La única forma en la que dicha notificación desaparezca es recogiendo el libro físicamente en la biblioteca, ya que el bibliotecario en ese momento registrará su recogida y el sistema de manera automática eliminará la notificación en la sesión del lector.

Notificaciones

	Biblioteca Pública Pilar y Faus El libro "El hombre en busca de sentido" ya está disponible puede ir a recogerlo hoy mismo. 19/08/2023	
--	--	--

Figura 103: Vista Notificaciones de un lector

El lector puede acceder en cualquier momento a la información de su perfil, pulsando sobre el ícono de usuario del menú superior.

Diseño e implementación de una aplicación web para la gestión de libros en bibliotecas

The screenshot shows the BiblioConnect application interface. At the top, there's a purple header bar with the logo 'BiblioConnect' and navigation links: 'Inicio', 'Catálogo', 'Eventos', 'Préstamos', a notification bell icon with '1' notification, and a user profile icon.

The main content area displays a profile for 'Lucia Santos González'. On the left is a circular profile picture of a smiling woman. To the right are three sections: 'Dirección' (Address) with 'Vicente Blasco Ibáñez, 124' and '03182 Torrevieja'; 'Datos contacto' (Contact Data) with 'Teléfono: 675132091' and 'Correo electrónico: luciasantos@gmail.com'; and 'Datos personales' (Personal Data) with 'Dni: 58977613A' and 'Fecha de nacimiento: 22/06/1997'.

Below this is a map titled 'Mi ubicación' (My Location) showing a street view with a blue marker indicating the user's position. A purple button at the bottom of the map says 'Editar perfil' (Edit profile).

Figura 104: Vista perfil de un lector

El lector puede modificar la información de su perfil pulsando sobre el botón “Editar perfil”. A continuación, se abrirá un modal con todos los campos del perfil del lector rellenados. Para editar alguno de ellos, simplemente habrá que modificar su información y pulsar el botón “Guardar”. El sistema verificará que los campos modificados tienen una estructura válida.

This screenshot shows the 'Editar perfil' (Edit profile) modal window. It contains fields for personal information: Nombre (Name), Apellidos (Last Name), Dni, Email, Teléfono (Phone), Fecha de Nacimiento (Birth Date), Estado (State), Ciudad (City), Calle (Street), Código Postal (Postal Code), Contraseña (Password), and Confirmar contraseña (Confirm Password). There's also a section for 'Seleccionar una imagen' (Select an image) with a preview of the profile picture and buttons to 'Seleccionar archivo' (Select file) or 'Ning...elec.' (None). A large purple 'Guardar' (Save) button is at the bottom.

Figura 105: Vista Modal editar perfil lector

El sistema verificará que los campos modificados tienen una estructura válida. En caso de tenerla, se mostrará un cuadro de diálogo de éxito informando de la correcta modificación del

perfil. En caso contrario, se marcará el campo o campos incorrectos en rojo con su correspondiente mensaje de error.

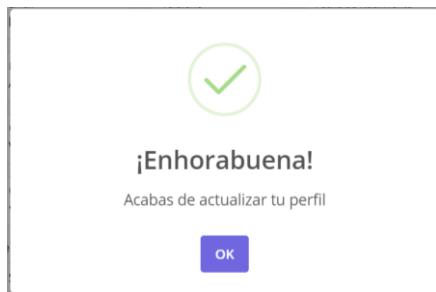


Figura 106: Vista cuadro diálogo confirmación edición perfil

11.1.5 Cerrar sesión

Los administradores de bibliotecas y los lectores pueden cerrar su sesión en cualquier momento pulsando el correspondiente botón. Una vez pulsado, aparecerá un cuadro de diálogo de confirmación que le preguntará al usuario si desea cerrar su sesión, en caso de que el usuario acepte se volverá a la ventana de inicio de sesión. En caso contrario, no se realizará ningún tipo de acción en el sistema.

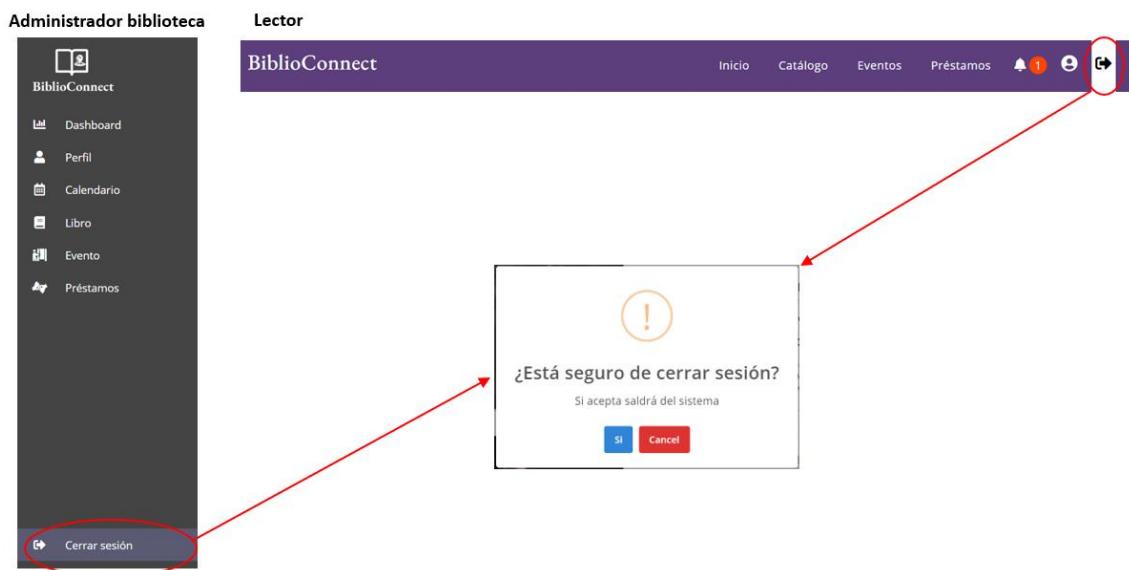


Figura 107: Vista Cierre de sesión

12. Anexo ODS

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.			X	
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.	X			
ODS 5. Igualdad de género.			X	
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.			X	
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.	X			
ODS 17. Alianzas para lograr objetivos.	X			

A continuación, se va a realizar una reflexión sobre la relación del TFG con los ODS relacionados.

- **ODS 4. Educación de calidad:**

La aplicación contribuye a este objetivo al permitir el acceso a libros y eventos literarios en las distintas bibliotecas de la Comunidad Valenciana. La educación de calidad es esencial para el desarrollo sostenible, y el sistema ayuda a los usuarios a mejorar su conocimiento.

Por un lado, se permite el registro de bibliotecas, de esta forma las instituciones educativas formarán parte de la red de recursos disponibles para los lectores. Esto amplía el acceso a la variedad de recursos literarios y fomenta la educación a través de la difusión de la información.

Además, al permitir a las bibliotecas programar eventos literarios como las presentaciones de libros o los talleres de escritura, promueve el aprendizaje y el intercambio de ideas.

- **ODS 9. Industria, innovación e infraestructuras:**

La utilización de servicios web de geolocalización y los mapas interactivos para mostrar las ubicaciones exactas de cada biblioteca reflejan la innovación tecnológica.

Como muchas otras aplicaciones similares, cada lector podrá visualizar un listado con todos los libros que han sido registrados en el sistema. El principal avance tecnológico ha sido la inclusión de servicios web de geolocalización, el lector visualizará la distancia exacta a la que se encuentra del libro deseado. Con esto se facilitará la elección correcta del libro deseado, ya que podrán existir diferentes ejemplares de libros (según las distintas bibliotecas) pero será recomendable escoger el libro más próximo al lector para su posterior recogida física.

Además, los perfiles de bibliotecas tienen la funcionalidad de mapas interactivos que marcan la ubicación exacta de una biblioteca del sistema. Este tipo de mapas permite a los lectores visualizar ubicaciones geográficas de bibliotecas de una manera precisa e intuitiva.

Otra de las funcionalidades que ha adoptado una mejora en la infraestructura, es el enfoque centralizado de todas las bibliotecas de la Comunidad Valenciana. La centralización y coordinación de todas las bibliotecas en una sola plataforma pueden mejorar la eficiencia al simplificar procesos y reducir la duplicación de esfuerzos.

- **ODS 11. Ciudades y comunidades sostenibles:**

La unión de bibliotecas y lectores en una plataforma centralizada, se fomenta la colaboración entre miembros de la comunidad lo que contribuye a fortalecer los lazos locales y promover la cohesión social. Del mismo modo, que ayuda a preservar y promover la identidad cultural y literaria única de la Comunidad Valenciana.

- **ODS 12. Producción y consumo responsable:**

La aplicación ofrece la funcionalidad del préstamo de libros en bibliotecas. Con esta funcionalidad se permite que los usuarios tomen prestados recursos de manera temporal. Así pues, la aplicación reduce el consumo innecesario y el desperdicio de recursos incitando a los lectores al préstamo de libros, en lugar de comprarlos individualmente. El préstamo de libros refleja la idea de compartir y reutilizar recursos, lo cual es un principio fundamental de la producción y el consumo responsable.

- **ODS 16. Paz, justicia e instituciones sólidas:**

El préstamo de libros garantiza que los lectores tengan acceso equitativo a cualquier tipo de información y conocimiento, lo que es fundamental para conseguir una sociedad informada y justa. Independientemente de la capacidad financiera de los usuarios, podrán acudir físicamente a cualquier tipo de evento literario programado por las bibliotecas del sistema.

Al ofrecer servicios gratuitos que fomentan la educación y la participación cultural, la aplicación contribuye a la promoción de valores pacíficos al brindar a las personas herramientas para aprender, dialogar y enriquecerse.

- **ODS 17. Alianzas para lograr objetivos:**

La aplicación crea una red de colaboración entre bibliotecas y lectores. Esta colaboración refleja la importancia de las alianzas para lograr metas más amplias. Al unir fuerzas con bibliotecas, la aplicación puede maximizar su impacto en la promoción de la educación y el acceso a la cultura.

La colaboración entre bibliotecas permite una amplia gama de contenidos y eventos, desde una variedad de géneros literarios hasta eventos culturales que reflejen la diversidad de intereses de la comunidad.

