UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Aplicación móvil y web para la gestión de una biblioteca virtual (Bibliotecapp)

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Esteve Carbonell, Albert

Tutor/a: Buendía García, Félix

CURSO ACADÉMICO: 2022/2023

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Desarrollo de una aplicación híbrida para la gestión de una biblioteca de forma virtual (Bibliotecapp)

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Esteve Carbonell, Albert

Tutor: Buendía García, Félix

2022/2023

Agradecimientos

A mi familia, por haberme dado la oportunidad de poder haber estudiado en Valencia. Por todo su cariño y apoyo en los momentos más difíciles. Por haberme enseñado a valerme por mí mismo y por haberme mostrado el camino correcto. Es la mejor familia que se puede tener.

A mis compañeros de universidad, convertidos en verdaderos amigos. Gracias por darme una increíble experiencia en esta nueva ciudad, por haberme ayudado siempre que lo he necesitado y por acompañarme en esta inolvidable etapa de mi vida. Gracias Jaime, Grego, Xavi, Joan Gómez y Joan Matarredona.

A mis profesores, por su dedicación en la formación y por haberme mostrado este maravilloso mundo del desarrollo de software.

A toda la demás gente que me ha estado apoyando durante estos cuatro años.

Gracias.

Resumen

En un mundo cada vez más automatizado y digitalizado, la eficiencia y comodidad son objetivos clave en todos los ámbitos de nuestra vida. En este contexto, surge la necesidad de mejorar la automatización en sectores específicos para brindar un mejor servicio. En el caso del sector bibliotecario, se busca no solo facilitar el acceso al patrimonio cultural y científico, sino también optimizar los procesos relacionados con los préstamos de libros y otros recursos.

Este Trabajo de Fin de Grado presenta el diseño, desarrollo e implementación de una aplicación multiplataforma utilizando Angular e Ionic como pilares fundamentales. La aplicación tiene como objetivo agilizar los procesos de acceso y préstamo de libros, así como informatizar los trámites necesarios para operar en una biblioteca, como la gestión de reservas de los recursos.

Al utilizar tecnologías de vanguardia en desarrollo web y móvil, se logra una gestión eficiente de la biblioteca de forma virtual. La aplicación permite a los lectores realizar búsquedas rápidas y sencillas de libros, reservarlos y recogerlos posteriormente en la biblioteca.

Los resultados de este trabajo demuestran el impacto positivo de la automatización y digitalización en el sector bibliotecario, al mejorar la eficiencia operativa y ofrecer un servicio más ágil y cómodo para los usuarios. Estas conclusiones refuerzan la importancia de la tecnología en la transformación digital de las bibliotecas y su papel en la promoción del acceso a la cultura y al conocimiento.

Palabras clave: multiplataforma, híbrida, desarrollo, biblioteca.

Resum

En un món cada vegada més automatitzat i digitalitzat, l'eficiència i comoditat són objectius clau en tots els àmbits de la nostra vida. En aquest context, sorgeix la necessitat de millorar l'automatització en sectors específics per a brindar un millor servei. En el cas del sector bibliotecari, es busca no sols facilitar l'accés al patrimoni cultural i científic, sinó també optimitzar els processos relacionats amb els préstecs de llibres i altres recursos.

Aquest Treball de Fi de Grau presenta el disseny, desenvolupament i implementació d'una aplicació multiplataforma utilitzant Angular i Ionic com a pilars fonamentals. L'aplicació té com a objectiu agilitzar els processos d'accés i préstec de llibres, així com informatitzar els tràmits necessaris per a operar en una biblioteca, com la gestió de reserves dels recursos.

Com s'utilitzen tecnologies d'avantguarda en desenvolupament web i mòbil, s'aconsegueix una gestió eficient de la biblioteca de manera virtual. L'aplicació permet als lectors realitzar cerques ràpides i senzilles de llibres, reservar-los i recollir-los posteriorment a la biblioteca.

Els resultats d'aquest treball demostren l'impacte positiu de l'automatització i digitalització en el sector bibliotecari, en millorar l'eficiència operativa i oferir un servei més àgil i còmode per als usuaris. Aquestes conclusions reforcen la importància de la tecnologia en la transformació digital de les biblioteques i el seu paper en la promoció de l'accés a la cultura i al coneixement.

Paraules clau: multiplataforma, híbrida, desenvolupament, biblioteca.

Abstract

In an increasingly automated and digitized world, efficiency and convenience are key objectives in all areas of our lives. In this context, the need to improve automation in specific sectors arises to provide better service. In the case of the library sector, the goal is not only to facilitate access to cultural and scientific heritage but also to optimize processes related to book loans and other resources.

This Thesis presents the design, development, and implementation of a cross-platform application using Angular and Ionic as fundamental pillars. The application aims to streamline the processes of accessing and borrowing books, as well as digitize the necessary procedures for operating in a library, such as resource reservation management.

By utilizing cutting-edge web and mobile development technologies, efficient virtual library management is achieved. The application enables readers to perform quick and easy book searches, make reservations, and subsequently collect the books at the library.

The results of this work demonstrate the positive impact of automation and digitization in the library sector, improving operational efficiency and offering a more agile and convenient service for users. These conclusions reinforce the importance of technology in the digital transformation of libraries and their role in promoting access to culture and knowledge.

Keywords: multiplataform, front-end, development, library.

Tabla de contenidos

I. Índice de ilustraciones

II. Índice de tablas

1. Introducción

1.1 Motivación

En la era digital actual, la mayoría de los servicios privados se han adaptado a la digitalización y automatización para mantener su competitividad. Sin embargo, las bibliotecas, siendo servicios no lucrativos, a menudo carecen de incentivos para actualizar y optimizar sus servicios. Por lo tanto, este proyecto tiene como objetivo mejorar la accesibilidad a estos servicios tan importantes mediante la creación de una biblioteca virtual donde los usuarios puedan solicitar, reservar y verificar la disponibilidad de libros y otros recursos en la biblioteca física.

Asimismo, los tiempos del Covid-19 han acelerado esta tendencia natural hacia la informatización. Particularmente, muchas personas presentan reticencia a interactuar con otras personas por este hecho y buscan minimizar las interacciones sociales, por tanto, buscan soluciones digitales. Una plataforma de estas características podría ayudar a informatizar todos los procesos que no urgen de ser presenciales, además de mejorar la disponibilidad de los servicios pudiéndose presentar las 24h del día, ya que no tendría por qué ajustarse a los horarios de una biblioteca tradicional. También se lograría mayor eficiencia puesto que se agilizarían muchos procesos respecto a la manera tradicional de hacer las cosas. Cabe destacar que el objetivo no es reemplazar a los trabajadores existentes, sino aliviar su carga de trabajo al presentar otras formas de realizar sus funciones. Además, la comodidad se vería favorecida al ser una aplicación multiplataforma que se adapte a los dispositivos preferidos por los usuarios.

Con todas estas razones, nace la idea de Bibliotecapp, una aplicación para la gestión virtual de una biblioteca.

1.2 Objetivos

Los objetivos principales consistirían en la creación de este trabajo de fin de grado son la simplificación de procesos administrativos relacionados con el funcionamiento de la biblioteca como son: gestión de libros y sus reservas (incluyendo solicitudes, devoluciones, transporte de libros desde otras bibliotecas y consulta de disponibilidad de los libros), y la gestión de clientes/lectores (alta y baja de clientes, y visualización de sus acciones en la plataforma).

Se persiguen también otros objetivos que incluyen simplificar los procedimientos administrativos asociados al funcionamiento de la plataforma Bibliotecapp. Esto implica agilizar la gestión de altas y bajas de eventos propios de la biblioteca, así como recibir notificaciones de errores o sugerencias relacionadas con la plataforma.

Adicionalmente, se plantean otros objetivos relevantes en el proyecto, entre los cuales se encuentran:

- Acceso multiplataforma (web, web optimizada para dispositivos móviles y aplicación móvil (Android)).
- Accesibilidad para todos los usuarios independientemente de sus condiciones físicas.
- Interfaz simple, clara, rápida y dinámica, con un diseño intuitivo que permita a los usuarios encontrar y solicitar los recursos que necesiten de manera eficiente.
- Código de calidad, siguiendo las buenas prácticas de Angular, SQL, NodeJS y todos los lenguajes usados en el proyecto, para garantizar la seguridad y el buen funcionamiento de la aplicación.
- Posibilidad de traducir la plataforma al valenciano, inglés o castellano para atender a una amplia variedad de usuarios.
- Autenticación de usuarios segura, con protección de datos y medidas de seguridad para garantizar la privacidad y la confidencialidad de los usuarios.
- Personalización de usuarios.
- Promoción de la lectura y la cultura, fomentando el acceso a los recursos bibliográficos y la organización de eventos culturales para la comunidad.

1.3 Estructura de la Memoria

La estructura de la memoria está organizada de la siguiente manera:

- Capítulo 1: se presenta la motivación detrás del trabajo y se establecen los objetivos que se pretenden alcanzar. Además, se proporciona una visión general de la estructura de la memoria, delineando la organización de los capítulos subsiguientes.
- Capítulo 2: aborda el estado del arte, donde se analizan diversas alternativas existentes en el campo de estudio. También se plantea la propuesta del trabajo.
- Capítulo 3: se realiza un análisis detallado del problema. Se incluye un diagrama de casos de uso, identificando los actores involucrados y los casos específicos a considerar. Se especifican los casos de uso en detalle y se desarrolla un modelado conceptual, así como un diagrama de clases.
- Capítulo 4: se dedica al diseño de la solución. Se describe la estructura propuesta y se presentan los diseños preliminares, incluyendo una paleta de colores y un diagrama de flujo de reservas.
- Capítulo 5: se centra en el desarrollo de la solución propuesta. Se detallan las tecnologías utilizadas y se explica cómo se integran estas tecnologías en la estructura del proyecto. Asimismo, se mencionan otras herramientas utilizadas. Además, se nombran los patrones de diseño empleados. Finalmente, se describe la estructura de carpetas utilizada en el proyecto.

- Capítulo 6: se aborda la implantación de la solución. Se discuten las dificultades encontradas durante el desarrollo. Se mencionan las interconexiones, la configuración de la base de datos y se destaca la versión móvil de la solución. Además, se abordan aspectos de seguridad y rendimiento.
- Capítulo 7: se dedica a las pruebas realizadas, incluyendo pruebas de código y pruebas de campo.
- Capítulo 8: se aborda el mantenimiento y la gestión de versiones, explicando las acciones de mantenimiento llevadas a cabo y la gestión de versiones del proyecto.
- Capítulo 9: se presentan las conclusiones obtenidas a partir del trabajo realizado.
- Capítulo 10: se reserva para posibles trabajos futuros que puedan surgir a partir de este trabajo.
- Capítulo 11: se refiere a las referencias utilizadas en la memoria.

Además de los capítulos principales, se informa al lector sobre la existencia de anexos adicionales que complementan el contenido del TFG. Estos anexos son:

- Anexo I: Objetivos de desarrollo sostenible (ODS)
- Anexo II: API de Bibliotecapp
- Anexo III: Encuesta de Bibliotecapp

2. Estado del arte

En el ámbito de la gestión de bibliotecas, existen diversas aplicaciones que permiten llevar a cabo funcionalidades similares a las que se propone desarrollar en este TFG. A continuación, se describen algunas de las alternativas más relevantes:

2.1 **Alternativas**

2.1.1 Polibuscador

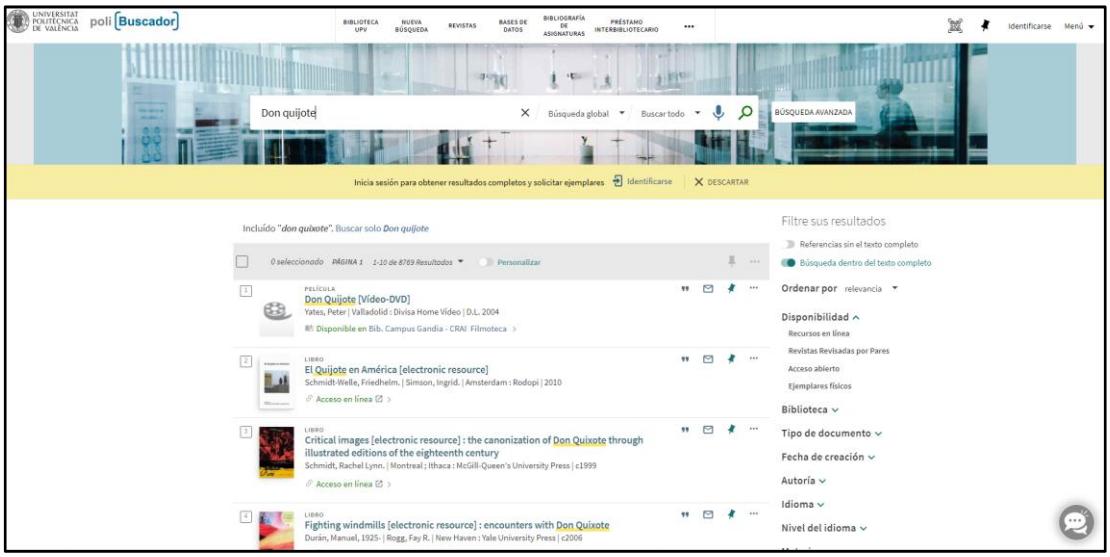

Ilustración 1 - Alternativa a Bibliotecapp: Polibuscador

Polibuscador se trata de plataforma web de la Universidad Politécnica de Valencia (UPV) cuya función es buscar bibliografía de la propia institución. Además de presentar una interfaz de los servicios de la biblioteca. Es una alternativa interesante para analizar debido a su amplia experiencia en el sector [1]. En la ilustración 1 se muestra una imagen de la página principal de esta plataforma.

2.1.2 Aplicación de la BNE

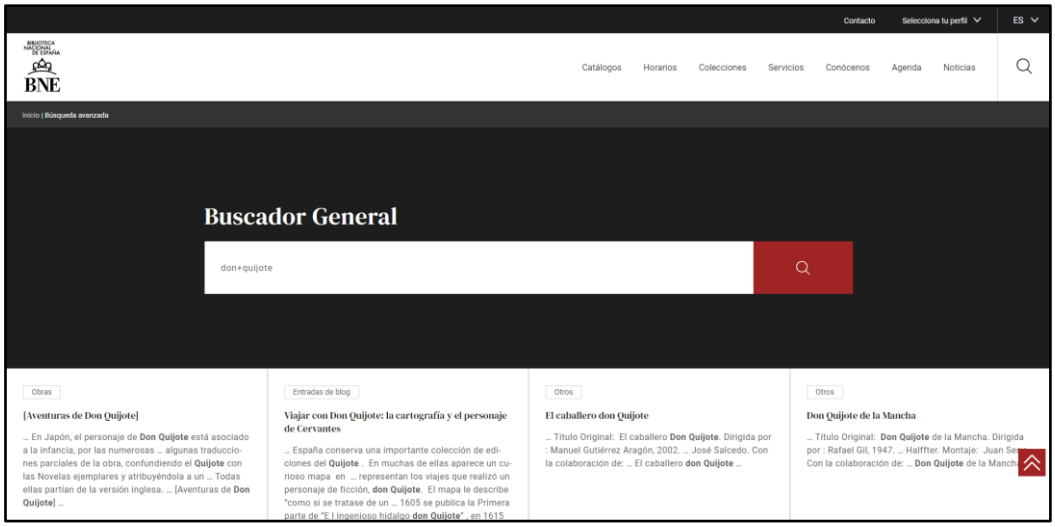

Ilustración 2 - Alternativa a Bibliotecapp: Aplicación de la BNE

La aplicación de la Biblioteca Nacional de España es otra plataforma web encargada de dar servicios de información y consulta bibliográfica especializada y de préstamos interbibliotecarios. Uno de sus componentes más destacados es la Biblioteca Digital Hispánica, que es proporciona acceso libre a una amplia variedad de documentos, libros, manuscritos, mapas y grabaciones sonoras en un formato digitalizado). Esta plataforma podría resultar de interés para entender cómo se ha llevado a cabo la gestión de documentos en una institución académica [2]. En la ilustración 2 se muestra una imagen de la página principal de esta aplicación.

2.1.3 Repositorio Documental “Gredos”

Ilustración 3 - Alternativa a Bibliotecapp: Repositorio Documental "Gredos"

El repositorio documental “Gredos” de la Universidad de Salamanca es un sitio web para la consulta de libros, documentos, manuscritos, repositorios docentes o

científicos digitales de la Universidad de Salamanca [3]. En la ilustración 3 se muestra una imagen de la página principal de esta aplicación.

2.1.4 Casa del libro

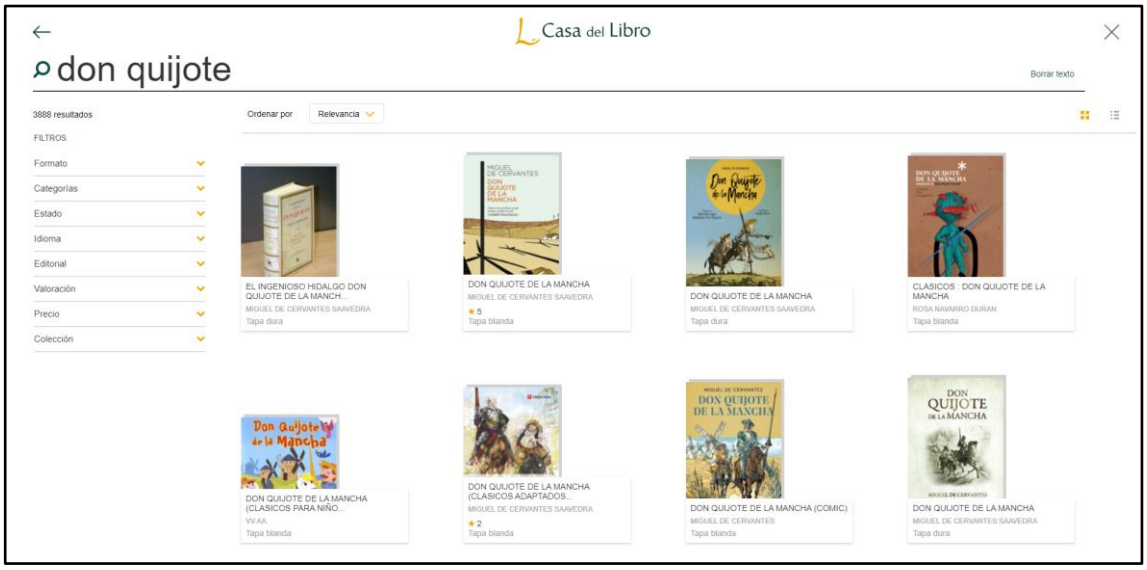

Ilustración 4 - Alternativa a Bibliotecapp: Casadellibro.com

La casadellibro.com se trata de una plataforma principalmente enfocada en la venta de libros. Sin embargo, podría ser interesante examinar cómo se presenta la información de los libros en línea y cómo se realizan las ventas en una plataforma en línea además de cómo se promocionan sus eventos [4]. En la ilustración 4 se muestra una imagen de la página principal de esta plataforma.

2.1.5 Nextory

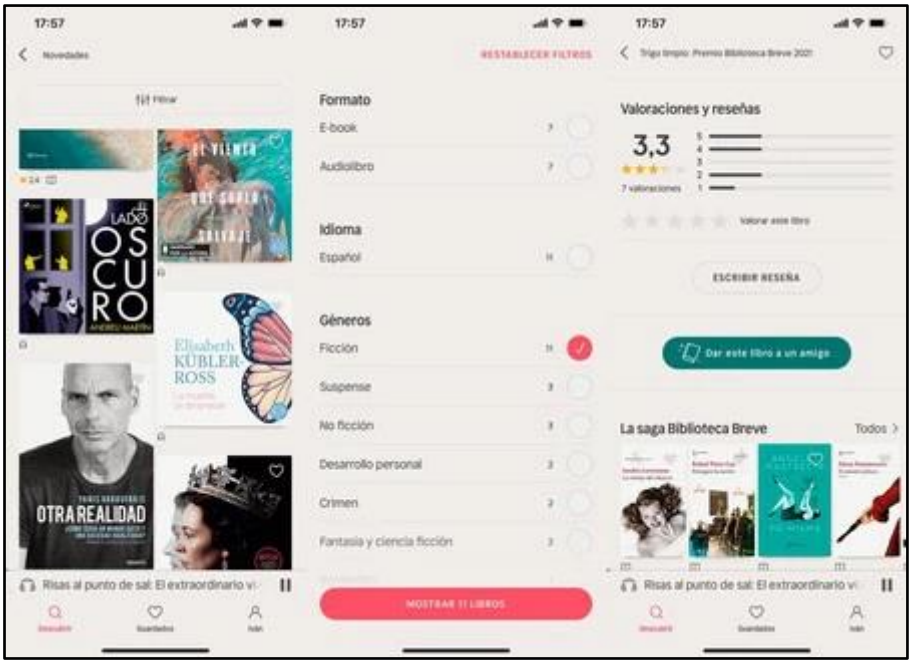

Ilustración 5 - Alternativa a Bibliotecapp: Nextory

Nextory es un servicio de alquiler de libros y audiolibros en formato digital sueco en el que participa Movistar. Este modelo de negocio podría resultar interesante para analizar cómo se gestionan los alquileres de libros en una plataforma en línea [5] [6]. En la ilustración 5 se muestra una imagen de la página principal de este servicio.

2.2 Tabla comparativa

Tras describir las diferentes alternativas en el ámbito de la gestión de bibliotecas, se introducirá una tabla comparativa para evaluar las características y funcionalidades de cada plataforma:

	Bibliotecapp	**Polibuscador**	**Biblioteca Nacional de España**	**Repositorio Documental "Gredos"**	**Casa del libro**	**Nextory**
Accesibilidad	Cualquier persona, gratuita	Sólo miembros, gratuita	Cualquier persona, gratuita	Cualquier persona, gratuita	Cualquier persona, de pago (1*)	Cualquier persona, de pago
Filtro de recursos	Sí	Sí	Sí	Sí	Sí	Sí
Idiomas disponibles (aplicación)	Español, Valenciano, Inglés	Español, Valenciano, Inglés	6 idiomas	5 idiomas	Español	10 idiomas
Idiomas disponibles (libros)	15 idiomas (iniciales)	13 idiomas	Más de 10 idiomas	Más de 12 idiomas	40 Idiomas	Más de 10 idiomas
Instalaciones físicas	Sí	Sí	Sí	No	Sí	No
Interfaz amigable	Sí	Sí	Sí	Sí	Sí	Sí
Organización de eventos y promoción de eventos	Sí	Sí*	Sí	No	Sí	No
Plataforma	Web + Móvil	Web	Web + Móvil	Web	Web + Móvil	Móvil
Préstamo de libros físicos	Sí	Sí	Sí	No	No	No
Préstamos interbibliotecarios	Sí	Sí	Sí	No	No	No
Recomendaciones de libros	No	No	No	No	Sí	Sí
Reserva recursos bibliotecarios	Sí	Sí	Sí	No	No	No
Tipo de bibliografía	Cualquiera	Recursos de la organización	Bibliografía española y latinoamericana	Recursos de la organización	Enfoque en libros comerciales	Enfoque en libros comerciales

Tabla 1 - Comparación de características contra aplicaciones rivales

(1*) El acceso al sitio web e instalaciones son gratuitas, pero hay que pagar por los servicios ofrecidos como la compra de libros.

(2*) Dentro del Polibuscador (UPV) no se promocionan eventos, pero en las bibliotecas sí se hacen y promocionan.

2.3 Crítica al estado del arte

En esta sección, se llevará a cabo un análisis de los trabajos previos relacionados con bibliotecas digitales y sus funciones básicas, con el objetivo de identificar fallos, ineficacias, lagunas o aspectos no abordados que justifiquen el desarrollo del TFG. De igual forma se analizarán las alternativas mencionadas en la sección anterior.

2.3.1 Análisis de las alternativas

Primeramente, se comentará la competencia de la plataforma de manera directa. Como se puede observar en la tabla 1, todas las aplicaciones relacionadas cuentan con una interfaz amigable y un método de filtrado de recursos. Es por tanto indispensable en una aplicación que maneje muchos recursos tener esta opción. Sin embargo, debido a la evolución individual de cada aplicación durante su desarrollo, las demás características no siempre se presentan de manera uniforme o con igual enfoque.

Además del filtrado de recursos y una interfaz amigable, hay otras características que varían entre las diferentes aplicaciones. Poniendo el foco en cuan son accesibles las alternativas se puede observar que estas varían según su ámbito. Por ejemplo, modelos de negocio como lacasadellibro.com permite a los usuarios hacer uso de las instalaciones propias, pero no acceder a sus recursos a no ser que haya un evento en específico. En este sentido, no se brinda un acceso abierto y gratuito a los recursos disponibles en la plataforma. Otras plataformas como el Polibuscador de la UPV permite el acceso abierto y gratuito a todos los recursos, pero únicamente a los miembros de esta universidad. Bibliotecapp al ser una plataforma bibliotecaria sin ánimo de lucro ha optado por el acceso abierto y gratuito a todos los usuarios de la biblioteca.

Otro punto que es importante a considerar en términos de accesibilidad es la cantidad de idiomas en los que está disponible la plataforma. Como se puede observar en la tabla 1 casi todas las aplicaciones rivales cuentan con diversos idiomas en los que está disponible la plataforma. Este apartado es muy importante puesto que al poder presentar la plataforma en distintos idiomas se consigue una mayor inclusión a personas que no dominan la lengua española. Asimismo, desde la plataforma se aboga por incluir más recursos en distintos idiomas para alcanzar un mayor público. De la forma en que está creada la plataforma bibliotecaria, simplemente con un traductor se podría incluir cualquier idioma que se desease ya que las traducciones están en archivos independientes de la implementación.

Para finalizar con la accesibilidad cabe resaltar los dispositivos en los que están disponibles las alternativas. Como se puede observar en la tabla 1, el 60% de las plataformas analizadas únicamente presentan una interfaz o web o móvil. Esto es debido a que las plataformas web normalmente intentan presentar una interfaz adaptable a

dispositivos móviles o tabletas. Un ejemplo de esto es el Polibuscador o la aplicación del repositorio documental de “Gredos”:

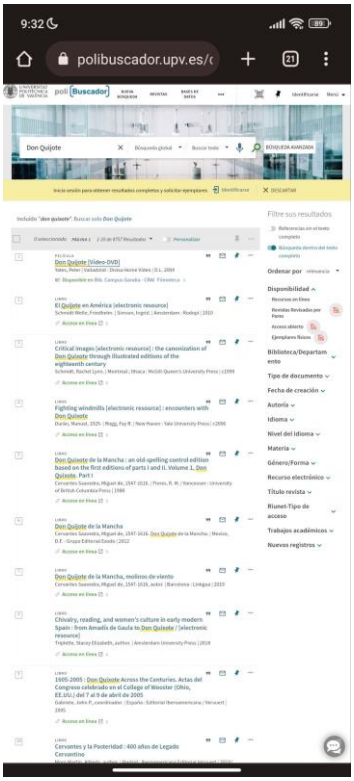

Ilustración 6 - Alternativa a Bibliotecapp: Polibuscador (versión web en dispositivo móvil)

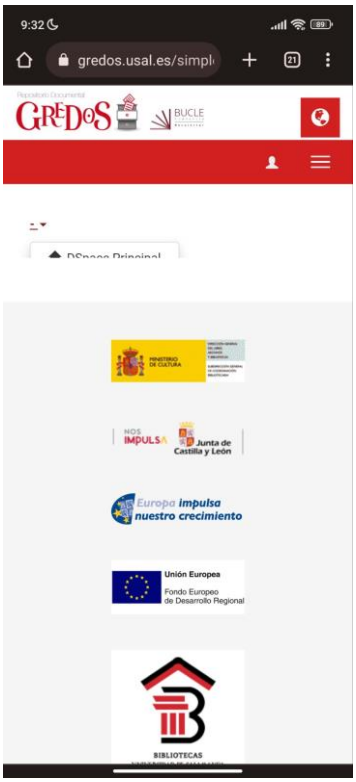

Ilustración 7 - Alternativa a Bibliotecapp: Gredos (versión web en dispositivo móvil)

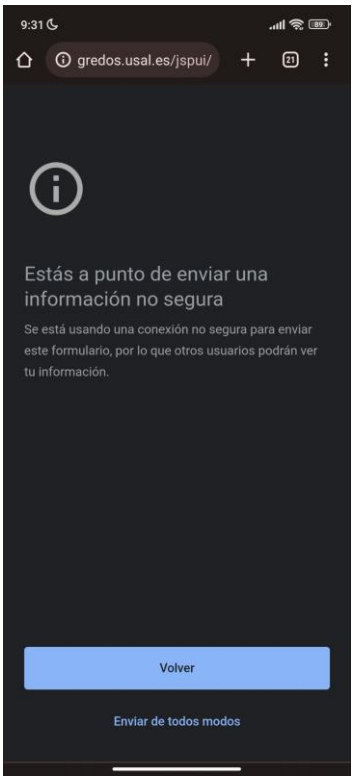

Ilustración 8 - Alternativa a Bibliotecapp: Gredos (versión web en dispositivo móvil - error)

Como se puede ver en las ilustraciones 6 y 7, se intenta adaptar el desarrollo web a móvil. En casos como el Polibuscador, se consigue tener un acceso completo a todas las funcionalidades de la aplicación. Sin embargo, en la prueba con "Gredos" hay opciones que en dispositivos móviles fallan o están mal optimizadas. Un ejemplo de esto sería como en la ilustración 7 se puede observar que existe un selector que está partido y no se puede acceder a su funcionalidad. Asimismo, en la ilustración 8 se puede ver que aparecen errores al cambiar de ventanas en la versión móvil. Es posible apreciar con estos dos ejemplos contrapuestos que al crear una versión web se deben tener en cuenta todos los tamaños y resoluciones de la mayoría de los dispositivos.

En el desarrollo de Bibliotecapp se han tenido en cuenta estos aspectos. Es por esto que presenta una interfaz adaptativa a dispositivos móviles o tabletas y a su vez a grandes dispositivos como smartsTV. De igual modo y para mejorar la experiencia en dispositivos móviles se han creado dos aplicaciones para que usuarios de Android o iOS puedan operar con la plataforma de una forma más optimizada para sus dispositivos.

Dejando a un lado la parte de accesibilidad y poniendo el foco en las alternativas que ofrecen un préstamo de recursos como son Polibuscador o la aplicación de la BNE, se puede observar que al ser instituciones académicas su bibliografía se centra en recursos de la propia institución normalmente de ámbito académico. Es por esto que Bibliotcapp potencialmente tiene más capacidad de cubrir una demanda más grande de recursos puesto que puede poseer una bibliografía académica como más enfocada a libros comerciales como la casa del libro. Si se dirige la atención al proceso para tomar o reservar recursos se observa una parecida forma de actuar en todas las alternativas. Es por

esto que no se desarrollará esta parte. Sin embargo, es importante recalcar que esta es la funcionalidad principal de la plataforma. Y por tanto ha sido necesario comprobar distintos métodos de actual para verificar que los procesos se hayan hecho de la manera más familiar y habitual posible para cualquier usuario de la plataforma.

Finalmente, tras analizar todas las alternativas se han constatado el buen hacer de algunas prácticas o características y se han descubierto nuevas que pueden ser incluidas en el futuro. Por ejemplo una de ellas está presente en el repositorio documental de “Gredos”. Esta plataforma se centra en recursos digitales y permite la visualización de estos. Esta acción de momento no está disponible en la plataforma Bibliotecapp, pero si es deseada. De igual forma se desea incluir en un futuro recomendaciones de recursos basadas en los recursos ya pedido.

2.3.2 Trabajos académicos

En la búsqueda realizada en la colección de trabajos académicos de la ETSINF, se encontró un único resultado de un proyecto similar a Bibliotecapp. El TFG titulado "Desarrollo de una biblioteca digital" [7]. Aunque el proyecto objeto de este documento se enfoca en la creación de una biblioteca física con operación digital, comparte ciertas similitudes con el trabajo previo analizado.

En el proyecto de Durà García se desarrolló una biblioteca digital con funciones básicas de préstamo y devolución de libros, lo que se asemeja a la funcionalidad que esta biblioteca busca implementar. Además, abordó la interconexión entre la biblioteca digital y la biblioteca física. Sin embargo, es importante mencionar que este proyecto fue presentado hace más de 13 años, y desde entonces han surgido nuevas tecnologías y herramientas que mejoran la eficiencia y el rendimiento de los procesos de la plataforma. Además, otra diferencia relevante es que el proyecto objeto de este TFG busca ofrecer una interfaz más atractiva y amigable para el usuario, lo que puede contribuir a una mejor experiencia de navegación y a la fidelización de los usuarios de la biblioteca digital. Finalmente, esta solución también se enfoca en mejorar la accesibilidad de los usuarios, ya que se contará con una solución híbrida compatible con dispositivos móviles, lo que permitirá a los usuarios realizar pedidos, devoluciones y reservas desde cualquier lugar y en cualquier momento.

2.4 Propuesta

El propósito principal de este trabajo es el desarrollo de una aplicación para una biblioteca que permita a los usuarios hacer pedidos, devoluciones y otras funciones básicas desde cualquier dispositivo móvil, tablet, ordenador o cualquier dispositivo inteligente con acceso a internet. A través del análisis de alternativas existentes, se ha observado que muchas de ellas son de pago o se centran únicamente en bibliotecas de archivos digitales sin posibilidad de obtener productos en formato físico. Además, algunas se especializan en recursos de un tema concreto y otras no incluyen todos los formatos disponibles en las bibliotecas físicas, como CDs, DVDs, e-books o cómics.

Esta propuesta intenta combinar las mejores soluciones de todos estos competidores junto a un diseño robusto, seguro, atractivo, actual y accesible y que promueva la lectura y cultura.

3. Análisis del problema

Una vez realizado el estudio sobre el estado del arte en la gestión de bibliotecas digitales, es importante llevar a cabo un análisis del problema para identificar oportunidades de innovación y negocio en este Trabajo de Fin de Grado. Para ello, se usarán las siguientes técnicas y métodos: Diagrama de casos de uso, Diagrama de clases y Modelado conceptual.

3.1 Introducción

Esta aplicación o plataforma consta de tres partes: la primera es accesible por todo el mundo desde ella se pueden hacer funciones de consulta de los recursos de la biblioteca y sus disponibilidades, además de poder consultar los eventos. La segunda y tercera parte únicamente son accesibles para el usuario si se registra con sus credenciales y obtiene un carnet de la biblioteca. Un usuario autenticado puede acceder a todas las funciones básicas de la aplicación de autenticación y de reservas de los recursos disponibles. Finalmente, la tercera parte es únicamente accesible para personal autorizado. Esta parte se accede con una verificación de que el usuario es un usuario administrador como bien podría ser el operario de la biblioteca. Este operario puede realizar operaciones de gestión de usuarios, libros y otros recursos, reservas, eventos y recomendaciones, errores y comentarios.

Los usuarios que deseen acceder a la segunda parte podrán acceder mediante un inicio de sesión. Hay dos maneras de iniciar sesión: hacerlo con una cuenta de Google o una de Facebook. No hay diferencia entre acceder por una o por otra. Una vez autenticados es imprescindible para poder reservar un recurso de la biblioteca, registrar el número de teléfono propio. Este número de teléfono únicamente será usado para alertar al usuario del fin de plazo de una de sus reservas o cualquier problema relacionado con su cuenta o sus reservas. Además de esto, será necesario registrar su biblioteca favorita. En esta biblioteca se enviarán todos los libros/recursos en caso de un traslado de una biblioteca a otra, y es donde estarán disponibles los recursos una vez pedidos.

Para acceder a la tercera parte se necesitará una autorización. Esta autorización será dada a los empleados de la biblioteca. Una vez autorizados, podrán acceder a una parte de la plataforma especializada desde la que podrán gestionar los recursos de la biblioteca, los clientes, los eventos, las reservas y los comentarios. Además, serán usuarios de la biblioteca por lo que podrán hacer reservas, pedir recursos y todas las funcionalidades disponibles para los usuarios normales de la plataforma.

Los usuarios sin registrar no podrán realizar acciones de reserva, pedida o petición de traslado de libros. Asimismo, no podrán consultar las reservas que tengan puesto que no tendrán. Ni podrán acceder a la modificación de su perfil.

3.1.1 Aclaración

Ilustración 9 - Aclaración de los recursos

Como se puede apreciar en la ilustración 9, en los siguientes casos de uso y de ahora en adelante en toda la aplicación cuando se hable de recursos se referirá a los diferentes recursos que se pueden reservar o pedir en la biblioteca. Puesto que no únicamente ofrece libros se ha optado por nombrar "recursos". Estos recursos pueden ser: libros, revistas, eBooks, DVDs, CDs, comics… o cualquier recurso que la biblioteca ponga a disposición de los usuarios en su catálogo.

3.2 Diagrama de casos de uso

Los casos de uso son una técnica esencial para identificar las necesidades y funcionalidades de un sistema de información. Permiten a los desarrolladores entender las necesidades de los usuarios y establecer objetivos para el sistema.

3.2.1 Actores

Los actores en los casos de uso son los usuarios que interactúan con el sistema en cuestión (Bibliotecapp). Son importantes para comprender los requisitos y funcionalidades necesarias del sistema.

En la plataforma o aplicación existen tres roles de actores bien diferenciados ilustrados en la ilustración 10:

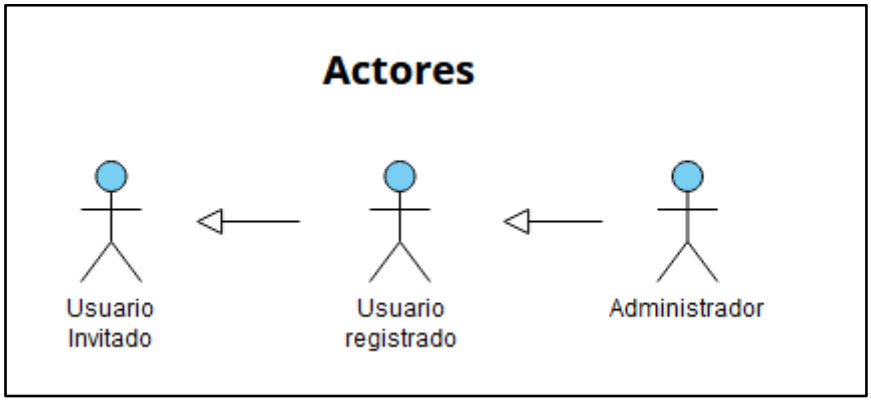

Ilustración 10 - Actores

- Usuario Invitado: corresponde a la primera parte de la plataforma explicada arriba. No se ha registrado, no tiene un perfil y por tanto no puede acceder a las funciones básicas de la plataforma. Sin embargo,

puede hacer operaciones de consulta de eventos y recursos. También podría registrarse, lo que desencadenaría que ascendiera a “Usuario Registrado”.

- Usuario Registrado: corresponde a la segunda parte de la plataforma explicada arriba. Un usuario registrado es aquel que ha iniciado sesión en la plataforma y ha recibido su carnet de la biblioteca. Puede hacer las acciones del usuario invitado y además, puede hacer operaciones de gestión de reserva de recursos como pudieran ser: pedir, reservar o pedir traslado de recursos. Asimismo, pueden escribir comentarios o sugerencias y editar su perfil.
- Administrador: corresponde a la tercera parte de la plataforma explicada arriba. Un usuario administrador es creado para cualquier trabajador de la biblioteca que se encargue de la gestión de reservas. Debe ser dado de alta en la base de datos de administradores por otro administrador. Puede hacer todas las funciones de los anteriores actores. Igualmente puede hacer funciones de gestión de inventario de recursos, así como gestión de reservas, gestión de comentarios (lectura y borrado), gestión de cola de reservas, gestión de eventos y gestión de usuarios.

3.2.2 Casos

En este proyecto, se han identificado diferentes casos de uso que se han agrupado en categorías según su naturaleza. Para identificar cada caso de uso, se utilizará una nomenclatura estandarizada que sigue el patrón "CU-xx" (donde xx representa el número del caso de uso).

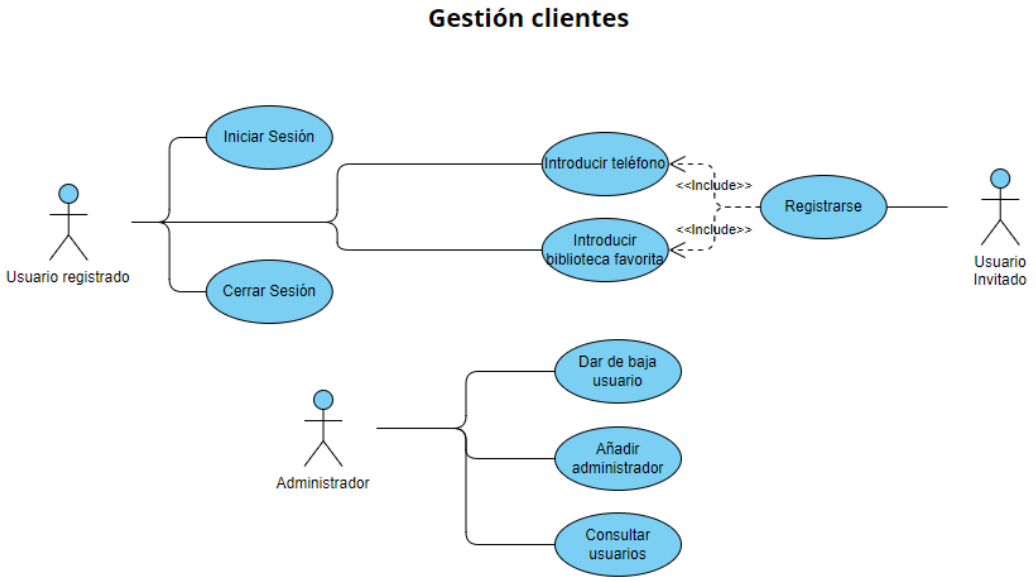

Ilustración 11 - Diagrama de casos de uso: Gestión de clientes

- Gestión de clientes: son las acciones relacionadas con la gestión de usuarios de la plataforma. Es importante para interactuar y gestionar eficazmente a los usuarios,

lo que puede mejorar la retención y satisfacción del usuario, y proporcionar información valiosa para la toma de decisiones de negocio de la plataforma.

- Iniciar sesión (CU-01)
- Registrarse (CU-02)
- Cerrar sesión (CU-03)
- Dar de baja usuario (CU-04)
- Añadir administrador (CU-05)
- Consultar usuarios (CU-06)
- Introducir teléfono (CU-07)
- Introducir biblioteca favorita (CU-08)

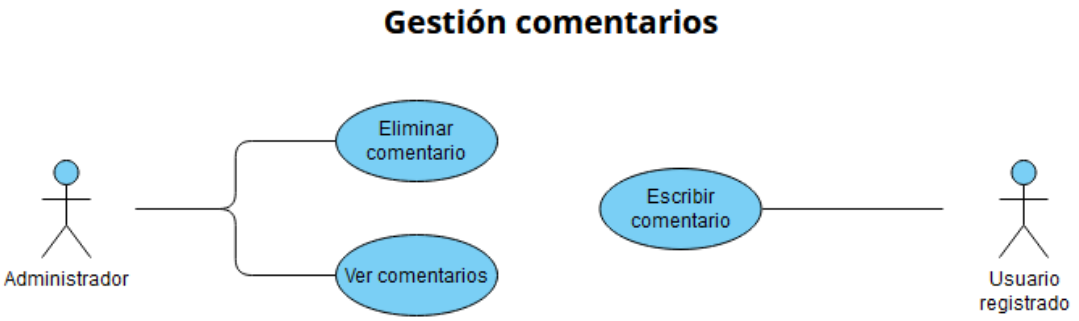

Ilustración 12 - Diagrama de casos de uso: Gestión de comentarios

- Gestión de comentarios: son las acciones relacionadas con la gestión de comentarios y/o sugerencias de la plataforma. Es importante puesto que es una forma directa de recibir retroalimentación para posibles cambios o mejoras de la plataforma en un futuro.
  - Eliminar comentario (CU-09)
  - Ver comentarios (CU-10)
  - Escribir comentario (CU-11)

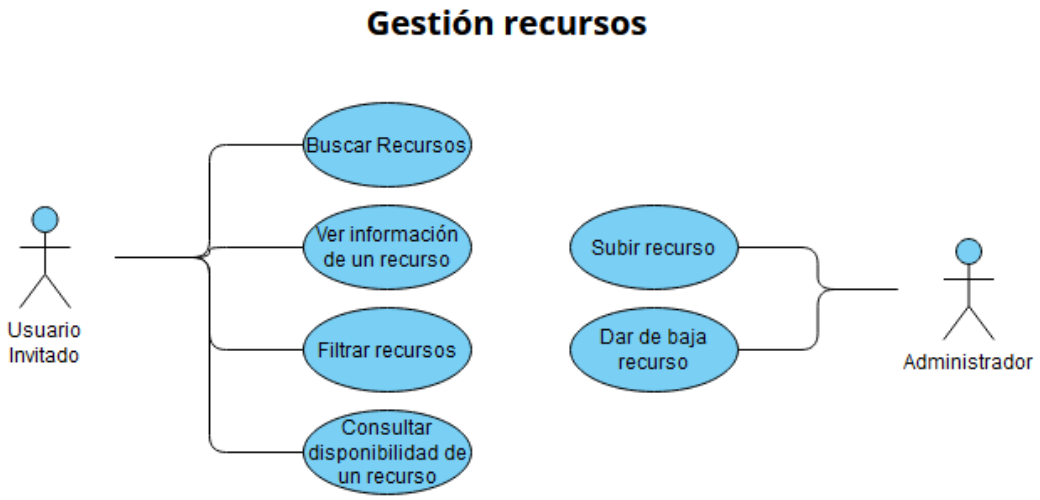

Ilustración 13 - Diagrama de casos de uso: Gestión de recursos

- Gestión de recursos: son las acciones relacionadas con la gestión de libros, comics, e-books, DVDs, CDs… de la plataforma, sin contar con la gestión de reservas. Es de las funcionalidades más cruciales.
    - Buscar recursos (CU-12)
    - Ver información de un recurso (CU-13)
    - Filtrar recursos (CU-14)
    - Consultar disponibilidad de un recurso (CU-15)
    - Subir recurso (CU-16)
    - Dar de baja recurso (CU-17)

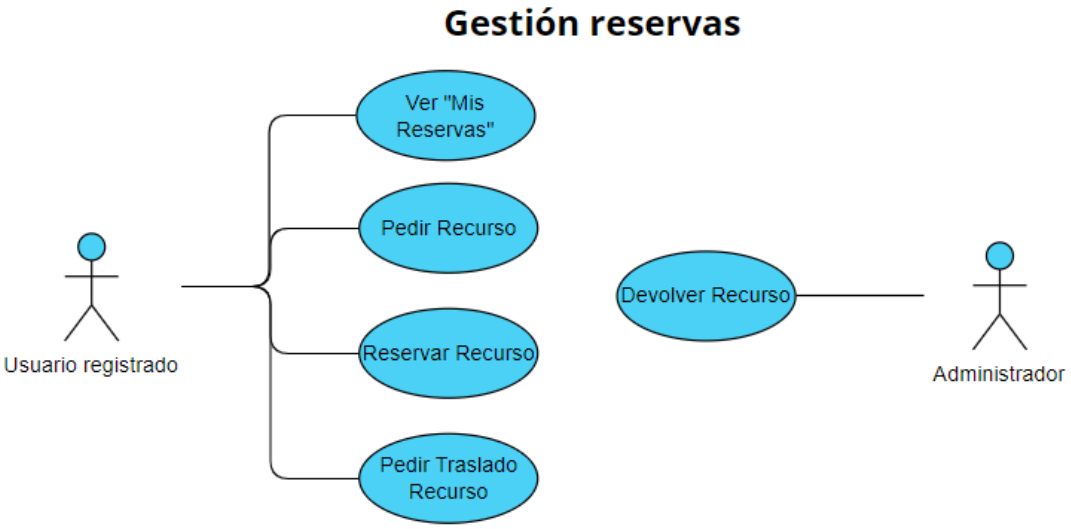

Ilustración 14 - Diagrama de casos de uso: Gestión de reservas

- Gestión de reservas: se refiere a las acciones relacionadas con la gestión de reserva de recursos de la plataforma. Es de las funcionalidades más críticas.
    - Ver “Mis Reservas” (CU-18)
    - Pedir recurso (CU-19)
    - Reservar recurso (CU-20)
    - Pedir traslado de recurso (CU-21)
    - Devolver recurso (CU-22)

Ilustración 15 - Diagrama de casos de uso: Gestión de eventos

- Gestión de eventos: se define como todos los procesos que involucran el uso de eventos de la plataforma. Es importante ya que permite a los usuarios acceder a información actualizada y precisa sobre los eventos disponibles en la plataforma, así como participar en ellos de manera eficiente y efectiva.
    - Subir evento (CU-23)
    - Dar de baja evento (CU-24)
    - Ver eventos disponibles (CU-25)

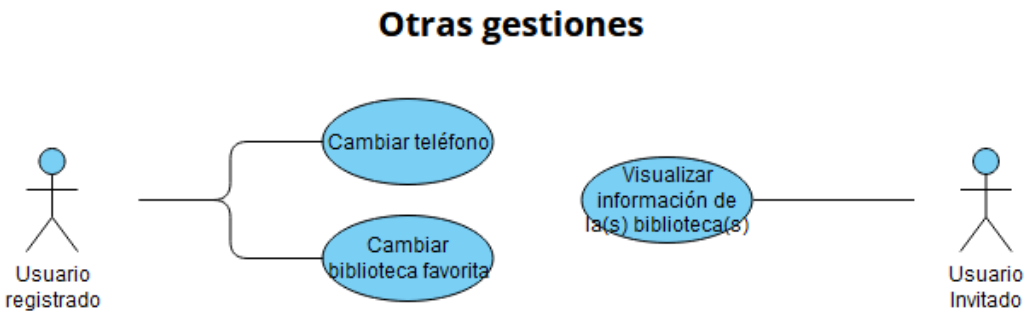

Ilustración 16 - Diagrama de casos de uso: Otras gestiones

- Otras gestiones: en esta sección aparecen todos los casos de uso que no pueden ser (por su naturaleza) incluidos en los demás grupos.
    - Cambiar teléfono (CU-26)
    - Cambiar biblioteca favorita (CU-27)
    - Visualizar información de la(s) biblioteca(s) (CU-28)

3.3 Especificación de los casos de uso

Una vez definidos, se detallarán:

Caso de Uso (CU-01)	Iniciar sesión
Descripción	Un usuario puede acceder al sistema en línea proporcionando sus credenciales de acceso. Para verificar la identidad y autorización se usará un servicio externo de Google Firebase [9].
Actores	Usuario registrado.
Precondiciones	Registrarse (CU-02).
Flujo normal	1. Al iniciar la aplicación hacer clic en “Iniciar sesión”. 2. Elegir una cuenta de las vinculadas.
Postcondiciones	El usuario queda autenticado y pasará a ser un usuario registrado, por tanto, podrá hacer todas las acciones posibles de su rol.
Excepciones	Ninguna.

Tabla 2 - Caso de Uso (CU-01): Iniciar Sesión

Caso de Uso (CU-02)	Registrarse
Descripción	Un usuario invitado puede darse de alta en el sistema.
Actores	Usuario invitado.
Precondiciones	Tener una cuenta de Google o Facebook.
Flujo normal	1. Al iniciar la aplicación hacer clic en “Registrarse”. 2. Elegir “Registrarse con Facebook” o “Registrarse con Google” según prefiera el usuario. 3. (Opcional) Escribir correo y contraseña de Google/Facebook. 4. Pulsar sobre la cuenta que desee para autenticarse.
Flujo alternativo	1. Desde cualquier pantalla hacer clic en el perfil como usuario. 2. Se redirigirá a la pantalla de registro y desde ahí se podrá acceder al flujo normal.
Postcondiciones	La cuenta del usuario ahora está registrada en base de datos y en el navegador. A partir de ahora iniciará sesión automáticamente dándole a “Iniciar sesión” ya que su sesión ha sido guardada.
Excepciones	Ninguna.

Tabla 3 - Caso de Uso (CU-02): Registrarse

Caso de Uso (CU-03)	Cerrar sesión
Descripción	Un usuario registrado puede dar de baja su sesión.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión (CU-01).
Flujo normal	1. En la pantalla principal hacer click en el perfil. 2. Pulsar “Cerrar sesión”.
Postcondiciones	Se borrará la sesión del usuario.
Excepciones	Ninguna.

Tabla 4 - Caso de Uso (CU-03): Cerrar Sesión

Caso de Uso (CU-04)	Dar de baja usuario
Descripción	El administrador podrá eliminar la cuenta de un usuario si considera que ha infringido las normas de la biblioteca o de la plataforma.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. El administrador se da cuenta del uso indebido de un usuario. 2. Accede a la plataforma web de Firebase con las credenciales de la biblioteca [8]. 3. Pulsa en el apartado de “Authentication”. 4. Busca el usuario deseado y hace clic sobre el icono “ ⋮ ”. 5. Debe darle a la opción de “Borrar cuenta”. 6. Aparecerá un diálogo desplegable “popup”, hay que darle al botón de “Borrar”.
Postcondiciones	Se elimina de la base de datos toda la información de ese usuario y nunca más podrá volver a registrarse o a iniciar sesión.
Excepciones	Ninguna.

Tabla 5 - Caso de Uso (CU-04): Dar de baja usuario

Caso de Uso (CU-05)	Añadir administrador
Descripción	Un administrador podrá darle los permisos de administrador a un usuario registrado no administrador.
Actores	Administrador.

Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. El administrador debe acceder a Firebase con sus credenciales de la biblioteca. 2. Pulsa en “Realtime Database”. 3. Pulsa en “/admins”. 4. Introduce el email y un userID para el nuevo administrador.
Postcondiciones	El nuevo administrador tendrá los permisos de un administrador normal, podrá acceder a la parte de administradores y realizar acciones propias del rol.
Excepciones	Ninguna.

Tabla 6 - Caso de Uso (CU-05): Añadir administrador

Caso de Uso (CU-06)	Consultar usuarios
Descripción	Un administrador podrá ver los usuarios que se han registrado en la plataforma, su último acceso a la misma, igual que su fecha de primer registro.
Actores	Administrador.
Precondiciones	Ninguna.
Flujo normal	1. El administrador se da cuenta del uso indebido de un usuario. 2. Accede a la plataforma web de Firebase con las credenciales de la biblioteca. 3. Pulsa en el apartado de “Authentication”.
Postcondiciones	Ninguna.
Excepciones	Ninguna.

Tabla 7 - Caso de Uso (CU-06): Consultar usuarios

Caso de Uso (CU-07)	Introducir teléfono
Descripción	El usuario registrado deberá introducir el teléfono para que se le avise a ese número si ha habido un problema en la biblioteca o en la plataforma.
Actores	Usuario registrado.
Precondiciones	Registrarse (CU-02).
Flujo normal	1. Acceder a perfil. 2. Pulsar en “Editar Perfil”. 3. Añadir un número de teléfono válido. 4. Hacer “clic” en el botón de “Aplicar Cambios”.

Postcondiciones	Se guardará el número del usuario. La biblioteca podrá llamar al usuario si lo siente necesario. El usuario podrá cambiar el número más adelante.
Excepciones	Ninguna.

Tabla 8 - Caso de Uso (CU-07): Introducir teléfono

Caso de Uso (CU-08)	Introducir biblioteca favorita
Descripción	El usuario registrado deberá una biblioteca favorita donde le llegarán los recursos que pida o donde irán destinados los traslados de recursos que igualmente pida.
Actores	Usuario registrado.
Precondiciones	Registrarse (CU-02).
Flujo normal	1. Acceder a perfil. 2. Pulsar en “Editar Perfil”. 3. Añadir una biblioteca favorita. 4. (Opcional) añadir una segunda biblioteca favorita. 5. Hacer “clic” en el botón de “Aplicar Cambios”.
Postcondiciones	Se guardará la biblioteca favorita del usuario. El usuario podrá cambiar estos valores más adelante.
Excepciones	Ninguna.

Tabla 9 - Caso de Uso (CU-08): Introducir biblioteca favorita

Caso de Uso (CU-09)	Eliminar comentario
Descripción	Un usuario administrador podrá eliminar los comentarios que pongan los usuarios registrados si los considera ofensivos o considera que no deberían estar en la plataforma.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal acceder a la parte de administradores pulsando en el botón “Acceder a Administración” que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un ojo con un aviso pequeño amarillo. 3. Se desplegará una lista con todos los comentarios, sugerencias y errores escritos.

	4. Seleccionar el error o errores que se deseen eliminar. 5. Pulsar el botón de "Borrar (comentario)".
Postcondiciones	El/Los comentario(s) serán borrados de la base de datos y nunca más volverán a aparecer.
Excepciones	Ninguna.

Tabla 10 - Caso de Uso (CU-09): Eliminar comentario

Caso de Uso (CU-10)	Ver comentarios
Descripción	Un usuario administrador podrá ver todos los comentarios que pongan los usuarios registrados en la plataforma.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal acceder a la parte de administradores pulsando en el botón "Acceder a Administración" que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un ojo con un aviso pequeño amarillo.
Postcondiciones	Se desplegará una lista con todos los comentarios, sugerencias y errores escritos.
Excepciones	Ninguna.

Tabla 11 - Caso de Uso (CU-10): Ver comentarios

Caso de Uso (CU-11)	Escribir comentario
Descripción	Un usuario registrado podrá escribir un comentario, sugerencia o error. No hay distinción entre estos. Siempre respetando a los otros usuarios de la plataforma.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página inicial, en el encabezado (header) se debe pulsar en "Escribir comentario". 2. En la nueva pantalla que aparecerá, se deberá escribir el texto que se quiera enviar. 3. Finalmente se debe presionar en "Enviar" para mandar el comentario.

Postcondiciones	El comentario quedará registrado en base de datos y podrá ser revisado por los usuarios administradores.
Excepciones	Ninguna.

Tabla 12 - Caso de Uso (CU-11): Escribir comentario

Caso de Uso (CU-12)	Buscar recursos
Descripción	Cualquier usuario al acceder a la plataforma puede buscar el libro que desee entre todos los recursos disponibles.
Actores	Usuario invitado.
Precondiciones	Ninguna.
Flujo normal	1. Cuando el usuario entre, le aparecerá un listado de todos los recursos disponibles en la plataforma.
Postcondiciones	Ninguna.
Excepciones	Ninguna.

Tabla 13 - Caso de Uso (CU-12): Buscar recursos

Caso de Uso (CU-13)	Ver información de un recurso
Descripción	Cualquier usuario puede ver los detalles de un recurso.
Actores	Usuario invitado.
Precondiciones	Ninguna.
Flujo normal	1. En la pantalla principal, se debe pulsar sobre el recurso que se desee seleccionar. Preferiblemente sobre la portada del recurso.
Postcondiciones	Aparecerá una lista detallada sobre el recurso con su portada si tuviera.
Excepciones	Ninguna.

Tabla 14 - Caso de Uso (CU-13): Ver información de un recurso

Caso de Uso (CU-14)	Filtrar recursos
Descripción	Los recursos se podrán filtrar según categoría, nombre y/o orden (ascendente o descendente).
Actores	Usuario invitado.
Precondiciones	Ninguna.

Flujo normal (Filtro por nombre de recurso)	1. Desde la pantalla de inicio hay un buscador en el que se puede introducir el nombre del recurso deseado.
Flujo normal (Filtro por categoría y orden)	1. Desde la pantalla de inicio se debe pulsar el botón que pone "Filtrar". 2. Se abrirá un desplegable (pop-up) desde el cual se puede seleccionar la categoría y/o el orden en el que le saldrán los recursos filtrados. (Por defecto el orden es ascendente).
Postcondiciones	Se mostrarán los recursos filtrados.
Excepciones	Si no hay recursos que se adapten a los parámetros marcados, no se mostrarán resultados.

Tabla 15 - Caso de Uso (CU-14): Filtrar recursos

Caso de Uso (CU-15)	Consultar disponibilidad de un recurso
Descripción	Cualquier usuario podrá comprobar directamente la disponibilidad de cualquier recurso que desee.
Actores	Usuario invitado.
Precondiciones	Ninguna.
Flujo normal	1. Desde la pantalla principal o desde la pantalla de detalle de un recurso aparecerá un texto de disponibilidad y/o botón debajo de cada uno. Si un recurso no está disponible para pedir, aparecerá una alerta de "Este recurso no está disponible". Asimismo, el botón indica la disponibilidad de cada uno y las acciones que se pueden realizar con ese recurso.
Flujo de administrador	1. Desde la página principal acceder a la parte de administradores pulsando en el botón "Acceder a Administración" que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un libro abierto con bombilla encendida dentro. 3. Se deberá introducir el número de carnet o la ID del usuario para verificar que existe y pulsar en "Buscar". Deberá aparecer un mensaje de que es un usuario real junto a su email. 4. Se exigirá introducir la id del recurso que se quiera buscar (el isbn en el caso de que fuera un libro) y se presionará en el segundo botón de "Buscar". 5. Aparecerá la disponibilidad del libro y su

	ubicación actual.
Postcondiciones	Ninguna.
Excepciones	(Normal) Si el recurso no se encuentra sale una pantalla indicando este fallo. (Administrador) Si la id del usuario, el carnet del mismo o la id del recurso no existen saldrá un mensaje de error.

Tabla 16 - Caso de Uso (CU-15): Consultar disponibilidad de un recurso

Caso de Uso (CU-16)	Subir recurso
Descripción	Un usuario administrador podrá añadir un recurso al catálogo de la biblioteca siempre que la biblioteca disponga de él y no esté disponible en la plataforma.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal se debe acceder a la parte de administradores pulsando en el botón "Acceder a Administración" que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un libro con una pequeña flecha verde apuntando hacia arriba. 3. Aparecerá una pantalla nueva donde se deberán incluir todos los campos obligatorios que se pidan. Los campos obligatorios a la hora de añadir un recurso como un libro son: título, nombre del autor, isbn, categoría, tipo de recurso e idioma. Y los campos no obligatorios que se podrían rellenar si se quisiese son: editorial y número de páginas. 4. (Opcional) Es posible subir, asimismo, una foto del recurso para que se muestre cuando el recurso esté subido. 5. Los campos introducidos serán validados y si todo está en un formato correcto se podrá subir el recurso pulsando el botón de "Subir recurso".
Postcondiciones	El recurso quedará subido a la base de datos y permanecerá allí de forma
Excepciones	Si el recurso no se ha podido subir aparecerá un error indicando que ha habido un fallo en la subida.

Tabla 17 - Caso de Uso (CU-16): Subir recurso

Caso de Uso (CU-17)	Dar de baja recurso
Descripción	Un usuario administrador podrá retirar un recurso del catálogo de la biblioteca siempre y cuando tenga un motivo justificado.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal acceder a la parte de administradores pulsando en el botón "Acceder a Administración" que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un libro con una pequeña flecha verde apuntando hacia arriba. 3. Se mostrará un listado con todos los recursos disponibles en la biblioteca. 4. (Opcional) Se puede buscar en la barra de búsqueda un recurso en concreto. 5. Seleccionar el recurso que se desee eliminar pulsando sobre él. 6. Presionar "Borrar (título del recurso)".
Postcondiciones	El recurso será borrado de la base de datos para siempre y nunca más volverá a aparecer en el catálogo.
Excepciones	Si el recurso no se ha podido borrar aparecerá un error indicando que ha habido un fallo en la eliminación.

Tabla 18 - Caso de Uso (CU-17): Dar de baja recurso

Caso de Uso (CU-18)	Ver "Mis Reservas"
Descripción	Un usuario registrado podrá ver los recursos que ha pedido, es decir los recursos que tiene en su posesión y no están en la biblioteca. Si un recurso se debía devolver y no se ha devuelto aparecerá en rojo, indicando que se debe devolver con la mayor brevedad posible.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal de la plataforma acceder en el encabezado (header) a "Mis reservas".
Postcondiciones	Se mostrarán todos los recursos que ha pedido el usuario. Y en rojo si tiene que devolver alguno.

Excepciones	Si el usuario no tiene recursos pedidos, no se mostrará ningún resultado.

Tabla 19 - Caso de Uso (CU-18): Ver "Mis Reservas"

Caso de Uso (CU-19)	Pedir recurso
Descripción	Un usuario registrado podrá pedir un recurso y recogerlo en la biblioteca de su preferencia siempre y cuando el recurso esté disponible y en esta misma biblioteca.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la pantalla principal o desde la pantalla de detalles de un recurso, se podrá presionar sobre el botón "Pedir" siempre y cuando el recurso lo permita. 2. Aparecerá un diálogo (pop-up) avisando de las condiciones de la pedida de un recurso. Se deberá presionar en "Pedir".
Flujo de administrador	1. Replicar el Caso de uso CU-15 en el flujo de administrador. 2. Presionar el botón "Pedir". 3. Aparecerá un diálogo (pop-up) avisando de las condiciones de la pedida de un recurso. Se deberá presionar en "Pedir".
Postcondiciones	El libro quedará pedido y se podrá recoger en la biblioteca de preferencia en un periodo de 3 días. Si no ha sido recogido en ese tiempo, se cancelará la pedida. Si es recogido, se creará un periodo de 15 días en los cuales se podrá disponer del recurso en nuestra posesión. Una vez acabado ese periodo tocará devolverlo. Asimismo, en "Mis Reservas" aparecerá el recurso pedido y la fecha de devolución del mismo. Finalmente, el nuevo recurso aparecerá "No disponible" para otros usuarios de la plataforma.
Excepciones	Si la id del usuario, el carné del mismo o la id del recurso no existen saldrá un mensaje de error.

Tabla 20 - Caso de Uso (CU-19): Pedir recurso

Caso de Uso (CU-20)	Reservar recurso
Descripción	Un usuario registrado podrá reservar un recurso si este ya ha sido previamente pedido por otro usuario.
Actores	Usuario registrado.

Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la pantalla principal o desde la pantalla de detalles de un recurso, se podrá presionar sobre el botón “Reservar” siempre y cuando el recurso lo permita. Aparecerá como “No disponible” si se puede reservar. 2. Aparecerá un diálogo (pop-up) avisando de las condiciones de la reserva de un recurso. Se deberá presionar en “Reservar”.
Flujo de administrador	1. Replicar el Caso de uso CU-15 en el flujo de administrador. 2. Presionar el botón “Reservar”. 3. Aparecerá un diálogo (pop-up) avisando de las condiciones de la reserva de un recurso. Se debe presionar en “Reservar”.
Postcondiciones	Una vez el usuario que tuviera el recurso a su disposición lo devuelva a la biblioteca, aparecerá disponible para este usuario y tendrá 3 días para recogerlo en la biblioteca de su preferencia. Si pasados esos 3 días no lo recoge, se le anulará el derecho a pedirlo y volverá a estar disponible para todos los usuarios. Si en vez de eso lo recoge, tendrá 15 días para tenerlo en su posesión.
Excepciones	Un mismo usuario no puede pedir un libro y luego reservarlo. Deberá volver a pedirlo una vez acaben los 15 días si ningún otro usuario lo ha reservado previamente.

Tabla 21 - Caso de Uso (CU-20): Reservar recurso

Caso de Uso (CU-21)	Pedir traslado de recurso
Descripción	Un usuario registrado podrá pedir que un recurso sea transportado a su biblioteca de preferencia. Para que pueda pedirlo.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la pantalla principal o desde la pantalla de detalles de un recurso, se podrá presionar sobre el botón “Pedir Traslado” siempre y cuando el recurso lo permita. Aparecerá como “No disponible” si se puede trasladar. 2. Aparecerá un diálogo (pop-up) avisando de las condiciones de las pedidas de traslado de un

	recurso a otra biblioteca. Se debe presionar en "Pedir Traslado".
Flujo de administrador	1. Replicar el Caso de uso CU-15 en el flujo de administrador. 2. Presionar el botón "Pedir Traslado". 3. Aparecerá un diálogo (pop-up) avisando de las condiciones de la pedida de traslado de un recurso a otra biblioteca. Se debe presionar en "Pedir Traslado".
Postcondiciones	Una vez realizado el procedimiento se debe proceder a empacar cuidadosamente el recurso para evitar daños durante el transporte. Posteriormente, un transportista llevará el libro a la biblioteca de destino, donde estará disponible para el usuario que lo haya solicitado. Se debe establecer un plazo para que el usuario recoja el libro, el cual puede ser de tres días hábiles. De esta manera, se garantiza que el libro llegue a su destino y se encuentre a disposición del usuario que lo necesita.
Excepciones	Ninguna.

Tabla 22 - Caso de Uso (CU-21): Pedir traslado de recurso

Caso de Uso (CU-22)	Devolver recurso
Descripción	Un usuario administrador puede devolver el recurso que un usuario registrado pueda haber pedido o reservado. Para ello el usuario registrado debe devolverlo a la biblioteca de manera física.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04), Consultar disponibilidad de un recurso (CU-15)
Flujo normal	1. Replicar el Caso de uso CU-15 en el flujo de administrador. 2. Presionar el botón "Se ha devuelto".
Postcondiciones	El recurso será devuelto a la biblioteca y añadido al catálogo. Si el recurso tenía cola de espera pasará al siguiente en la cola. Se avisará a este usuario de que ya puede pasar a recoger su libro por teléfono. Si no tenía cola de espera ni nadie más lo ha reservado, volverá a estar disponible para cualquier usuario. El usuario que lo ha devuelto no verá el recurso en "Mis reservas".
Excepciones	Ninguna.

Tabla 23 - Caso de Uso (CU-22): Devolver recurso

Caso de Uso (CU-23)	Subir evento
Descripción	Un usuario administrador podrá añadir un evento. Este evento se mostrará en la pantalla principal de la aplicación y también en la sección de eventos para promocionar actividades relevantes que se realicen en la biblioteca o en la plataforma de la misma. De esta manera, se le da la oportunidad al usuario de estar al tanto de las actividades que se desarrollan en la biblioteca y fomentar su participación.
Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal acceder a la parte de administradores pulsando en el botón "Acceder a Administración" posicionado debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un calendario con una "v" dentro y con una pequeña flecha verde apuntando hacia arriba. 3. Aparecerá una pantalla nueva donde se deberán incluir todos los campos obligatorios que se pidan. Los campos obligatorios a la hora de añadir un evento son: título y descripción del evento. 4. (Opcional) Se podrá subir, asimismo, una foto relacionada con el evento para que se muestre en la pantalla principal cuando el evento esté subido. 5. Los campos introducidos serán validados y si todo está en un formato correcto se podrá subir el recurso pulsando el botón de "Subir evento".
Postcondiciones	Una vez que el evento es añadido, este se almacenará en la base de datos de la aplicación y se mostrará en la pantalla principal de la misma para que todos los usuarios puedan verlo. De esta forma, se garantiza que la información sobre el evento esté disponible y sea accesible para todos los usuarios interesados.
Excepciones	Si el recurso no se ha podido subir aparecerá un error indicando que ha habido un fallo en la subida.

Tabla 24 - Caso de Uso (CU-23): Subir evento

Caso de Uso (CU-24)	Dar de baja evento
Descripción	Un usuario administrador podrá retirar un evento de la biblioteca siempre y cuando tenga un motivo justificado.

Actores	Administrador.
Precondiciones	Iniciar sesión (CU-04).
Flujo normal	1. Desde la página principal acceder a la parte de administradores pulsando en el botón "Acceder a Administración" que está debajo del encabezado (header). 2. Pulsar en el icono con la imagen de un calendario con una "v" dentro y con una pequeña flecha roja apuntando hacia abajo. 3. Se mostrará un listado con todos los eventos disponibles en la biblioteca. 4. Seleccionar el evento que se desee eliminar pulsando sobre él. 5. Se deberá presionar sobre el botón cuyo contenido es "Borrar (título del evento)".
Postcondiciones	El evento será borrado de la base de datos para siempre y nunca más volverá a aparecer en la plataforma.
Excepciones	Si el evento no se ha podido borrar aparecerá un error indicando que ha habido un fallo en la eliminación.

Tabla 25 - Caso de Uso (CU-24): Dar de baja evento

Caso de Uso (CU-25)	Ver eventos disponibles
Descripción	Al ingresar a la plataforma de la biblioteca, cualquier usuario tendrá acceso a ver los eventos de la plataforma. De esta manera, se asegura que toda la comunidad de usuarios tenga acceso a la información de los eventos y pueda participar en los mismos si lo desea.
Actores	Usuario invitado.
Precondiciones	Ninguna.
Flujo normal	1. Al final de la página principal o pulsando en el texto "Eventos" del encabezado (header) se podrá ver los eventos disponibles con su descripción e imágenes o vídeos si dispusieran de los mismos.
Postcondiciones	Ninguna.
Excepciones	Ninguna.

Tabla 26 - Caso de Uso (CU-25): Ver eventos disponibles

Caso de Uso (CU-26)	Cambiar teléfono
Descripción	Un usuario registrado en cualquier momento puede cambiar el teléfono con el que se registró en la plataforma.
Actores	Usuario registrado.
Precondiciones	Registrarse (CU-02), Introducir teléfono (CU–07)
Flujo normal	1. Acceder a perfil. 2. Pulsar en “Editar Perfil”. 3. Añadir un nuevo número de teléfono válido. 4. Hacer “clic” en el botón de “Aplicar Cambios”.
Postcondiciones	El nuevo número será registrado y vinculado con el usuario.
Excepciones	Ninguna.

Tabla 27 - Caso de Uso (CU-26): Cambiar teléfono

Caso de Uso (CU-27)	Cambiar biblioteca favorita
Descripción	Un usuario registrado en cualquier momento puede cambiar la biblioteca de preferencia con la que se registró en la plataforma.
Actores	Usuario registrado.
Precondiciones	Registrarse (CU-02), Introducir biblioteca favorita (CU–08)
Flujo normal	1. Acceder a perfil. 2. Pulsar en “Editar Perfil”. 3. Añadir una nueva biblioteca favorita. 4. (Opcional) añadir nueva una segunda biblioteca favorita. 5. Hacer “clic” en el botón de “Aplicar Cambios”.
Postcondiciones	Se guardará la nueva biblioteca favorita del usuario.
Excepciones	Ninguna.

Tabla 28 - Caso de Uso (CU-27): Cambiar biblioteca favorita

Caso de Uso (CU-28)	Visualizar información de la(s) biblioteca(s)
Descripción	Al ingresar a la plataforma de la biblioteca, cualquier usuario tendrá acceso a ver la información de la(s) biblioteca(s) asociada(s) a la plataforma. Está información será el nombre de la(s) biblioteca(s), el horario, el número de teléfono y la ubicación de la(s)

	biblioteca(s).
Actores	Usuario invitado.
Precondiciones	Ninguna.
Flujo normal	Al final de la página principal o pulsando en el texto “Conócenos” del encabezado (header) se podrá ver esta información.
Postcondiciones	Ninguna.
Excepciones	Ninguna.

Tabla 29 - Caso de Uso (CU-28): Visualizar información de la(s) biblioteca(s)

3.4 Modelado conceptual

El modelado conceptual es importante para diseñar sistemas efectivos. En el caso de una biblioteca, permite definir las entidades y relaciones necesarias para gestionar los recursos y las solicitudes de los usuarios. Esto es esencial para establecer una base sólida y coherente que permita implementar un sistema eficaz de gestión de la biblioteca. A diferencia del diagrama de clases, este enfoca en la estructura y los elementos clave del sistema, mientras que el segundo se centra en las interacciones y funcionalidades del sistema desde la perspectiva de los actores. A continuación, en la ilustración 17, se mostrará una propuesta del modelado conceptual creado para la aplicación de Bibliotecapp:

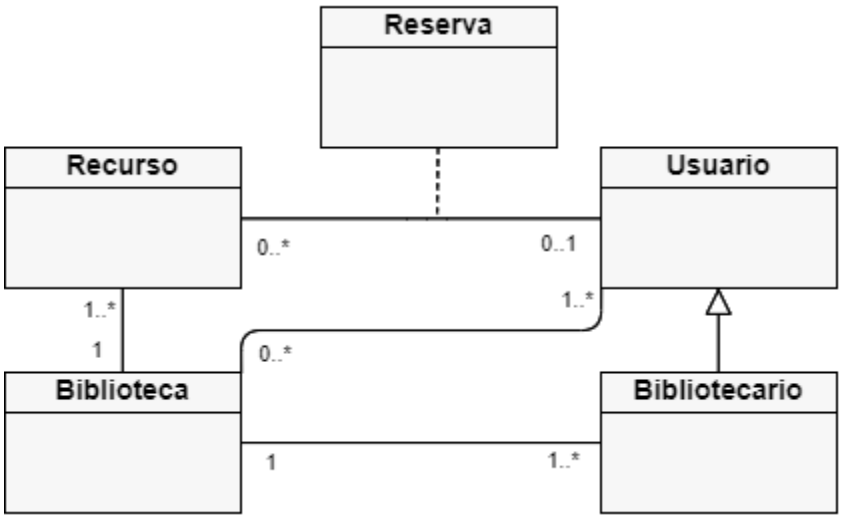

Ilustración 17 - Diagrama de modelo conceptual

3.5 Diagrama de clases

Un diagrama de clases es una herramienta visual que permite representar de manera gráfica las clases, atributos y relaciones de un sistema. Cada clase representa una entidad de la plataforma, como, por ejemplo, un usuario, un recurso o una reserva. Los atributos de cada clase se representan como variables y las relaciones entre las clases se representan mediante líneas que unen las distintas entidades y que indican el tipo de relación entre ellas.

En este sentido, el diagrama de clases permite visualizar de manera clara la estructura y las interconexiones entre las distintas entidades del sistema de la biblioteca. Esto es esencial para diseñar e implementar un sistema de información que permita gestionar de manera eficiente los recursos, los usuarios y sus peticiones. A continuación, en la ilustración 18, se presenta el diagrama de clases para el sistema de gestión de una biblioteca:

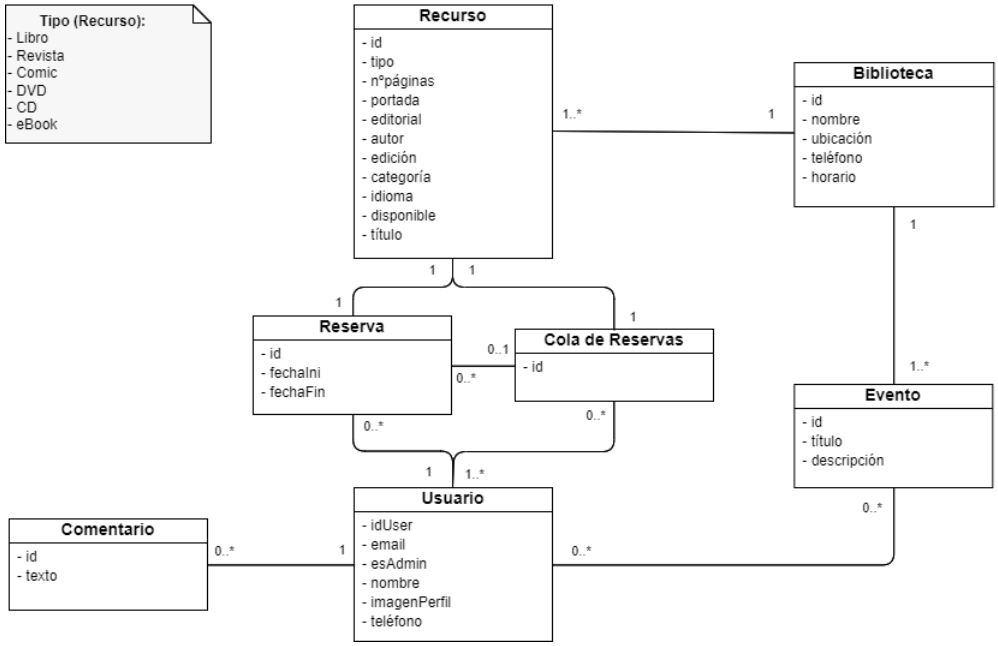

Ilustración 18 - Diagrama de clases

4. Diseño de la solución

4.1 Estructura

La estructura (o patrón de diseño) elegida para desarrollar esta plataforma ha sido una estructura Modelo-Vista-Controlador (MVC) [10]. Se ha considerado que este patrón de diseño sería una opción sólida para implementar la aplicación. Hay diversas razones por las que se ha planteado seguir esta estructura, una de ellas y la principal es la separación de responsabilidades. Es decir, cada parte o componente que más adelante se describirá desempeña una función distinta e independiente a la de los demás componentes. Esta división clara mejora la estructura del proyecto y facilita el desarrollo del proyecto. Otro factor a relucir de esta estructura es que es altamente escalable. El patrón MVC proporciona una gran flexibilidad ya que, al separar los componentes, estos se pueden modificar sin afectar a los componentes existentes. Esta modularidad facilita la evolución y el crecimiento del proyecto a medida que se agregan características adicionales o se realizan mejoras en el sistema. Esto produce que a largo plazo sea más mantenible. Finalmente, este patrón facilita las pruebas unitarias. Al estar separados los componentes pueden ser probados de manera aislada. Esto mejora la calidad del software al permitir identificar y corregir errores de forma más precisa.

Las responsabilidades de una aplicación en las que esta estructura arquitectónica se separara se dividen en tres componentes principales:

- Modelo (Model): representa la capa de datos y la lógica de negocio de la aplicación. Aquí se definen las estructuras de datos, la interacción con la base de datos y las reglas de negocio. El modelo se encarga de interactuar con la base de datos, almacenar y recuperar información, así como ejecutar la lógica de negocio relacionada con el procesamiento y transformación de los datos.
- Vista (View): es la capa de presentación de la aplicación, la interfaz de usuario con la que los usuarios interactúan. La vista se encarga de mostrar los datos (proporcionados por el modelo) al usuario. Además, la vista también recoge cualquier interacción o entrada del usuario, como clics en botones o formularios, y las envía al controlador para su procesamiento.
- Controlador (Controller): se encuentra la lógica de negocio y la interacción con el modelo y la vista. Es el componente que gestiona las acciones del usuario y coordina la comunicación entre la vista y el modelo. El controlador recibe las solicitudes del usuario desde la vista, interpreta y procesa esos datos, realiza las operaciones correspondientes en el modelo y finalmente actualiza la vista para mostrar los resultados.

4.2 Diseños preliminares

En esta sección, se presentan y describen los bocetos del diseño de las interfaces de usuario desarrolladas para este proyecto. Los bocetos se han creado con el objetivo de

visualizar y comunicar de manera efectiva las ideas y conceptos clave del sistema. Esto con el fin de poder transformar estos bocetos en vistas reales.

El primer boceto se enfoca en la página de inicio, donde se ha priorizado la simplicidad y la facilidad de uso. Se presenta una interfaz limpia y minimalista, con un diseño de navegación intuitivo que permite a los usuarios acceder rápidamente a las funciones principales del sistema. La interfaz consiste en una lista de libros que se pueden buscar y/o filtrar para obtener los resultados esperados. Asimismo, se pueden pulsar en cada libro para abrir el segundo boceto. También se dispondría la opción de filtrar por favoritos. A continuación, se presentará el primer boceto en la ilustración 19:

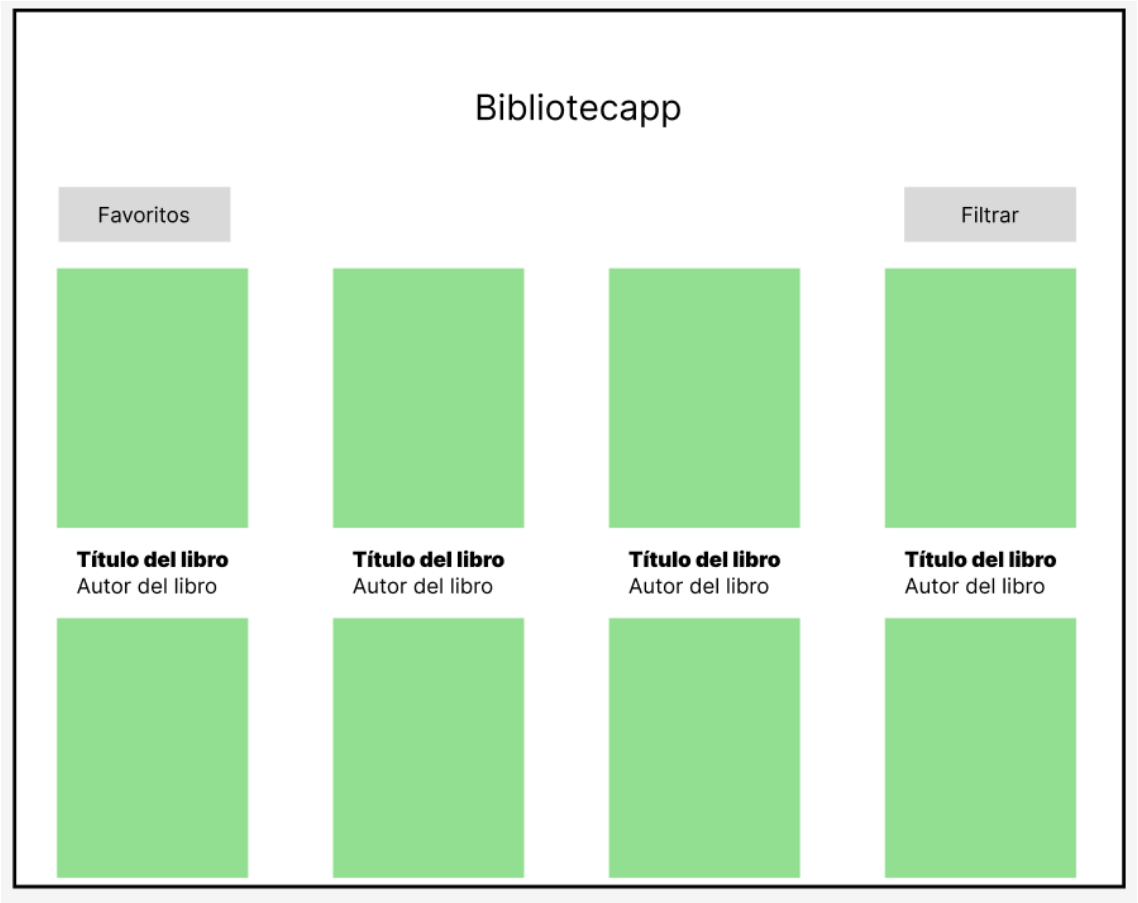

Ilustración 19 - Diseño preliminar 1: Página principal

El segundo boceto, presente en la ilustración 20, se centra en la página de información sobre un recurso, en este caso sobre un libro. Se ha diseñado una interfaz que muestra la información relevante del libro, como su nombre, autor y foto de portada. Se han incluido también botones adicionales para modificar la disponibilidad de un libro pidiéndolo o reservándolo por el usuario registrado. Sobre este boceto luego se añadió más información que presentan los libros de nuestra plataforma y además un modelo unificado para recursos con menos información o distinta como puedan ser CDs.

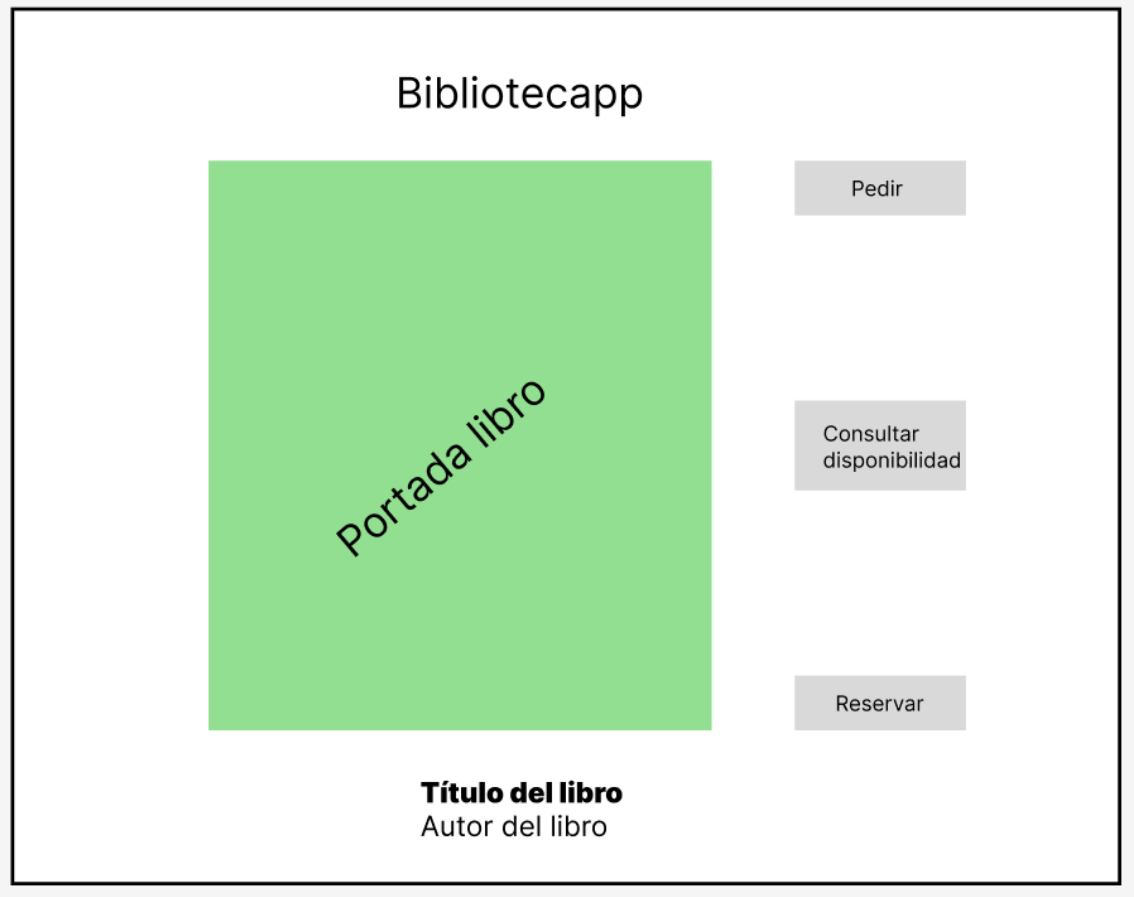

Ilustración 20 - Diseño preliminar 2: Información de un recurso (libro)

El último boceto creado muestra una imagen que representa los diferentes tipos de inicio de sesión y registro disponibles en la plataforma. La imagen presenta una interfaz clara y organizada, con un diseño intuitivo que permite a los usuarios elegir la opción que mejor se adapte a sus necesidades.

En la imagen, se encuentran dos encabezados que destacan las dos opciones principales: "Iniciar sesión" y "Registrarse". Justo debajo de cada opción, se muestran tres botones grandes que representan visualmente cada tipo de acción de las opciones principales.

Se han optado por utilizar servicios de autenticación externos para simplificar el proceso de registro y para aumentar el nivel de seguridad y confianza ya que estos servicios de autenticación externos, como Google, Facebook o Apple, cuentan con sistemas de verificación de identidad robustos. Al aprovechar la autenticación social, los usuarios pueden registrarse y acceder a la aplicación utilizando sus cuentas existentes en estas plataformas, evitando la necesidad de crear y recordar una nueva contraseña.

Además, al utilizar servicios de autenticación externos, la aplicación puede acceder a cierta información del perfil del usuario, como su nombre, dirección de correo electrónico y foto de perfil. Esta información puede ser utilizada para personalizar la experiencia del usuario, facilitar el llenado de formularios o incluso para conectar a los usuarios con sus contactos y amigos en la plataforma. El tercer boceto se muestra en la ilustración 21:

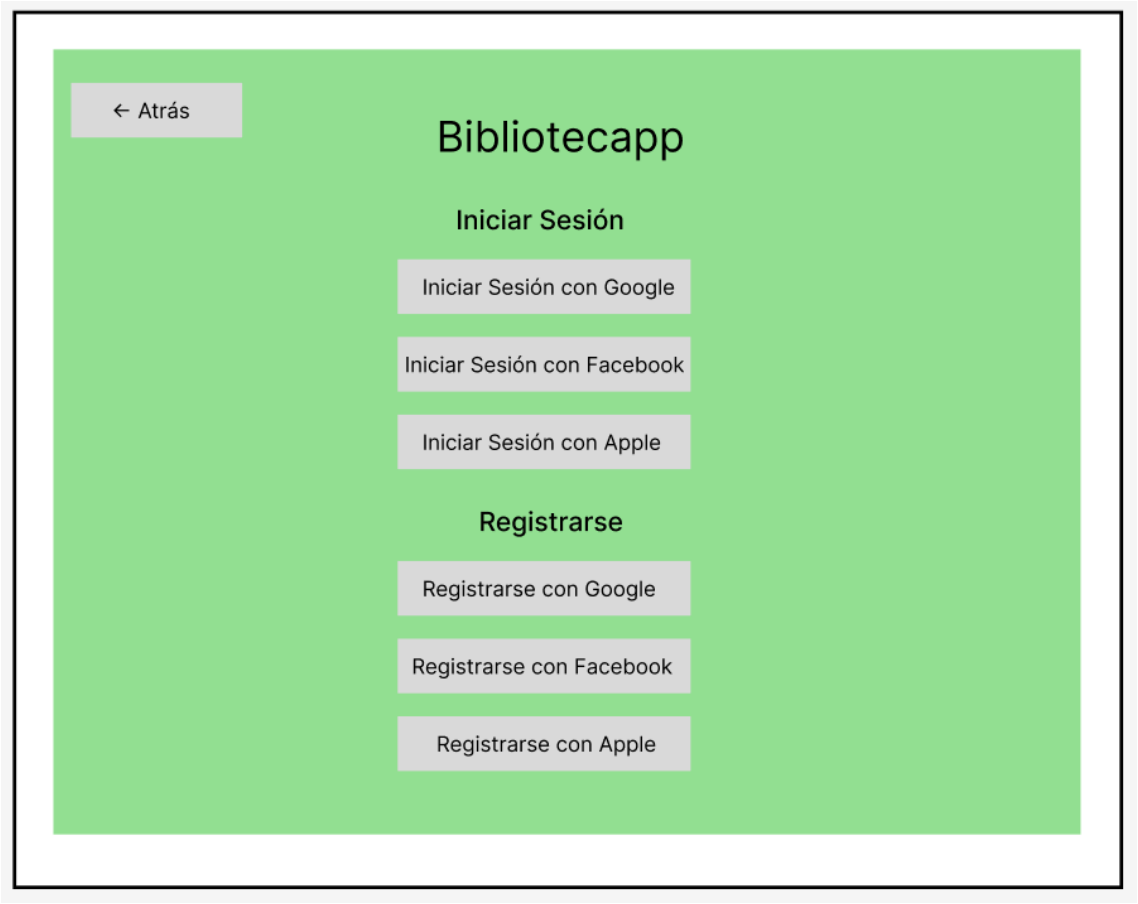

Ilustración 21 - Diseño preliminar 3: Métodos de autentificación

4.2.1 Paleta de colores

Para brindar a los usuarios una experiencia visual coherente en toda la aplicación, se ha adoptado una paleta de colores cuidadosamente definida. Aunque se han utilizado algunos colores adicionales para distinguir componentes específicos dentro de la interfaz, se ha priorizado el uso consistente de los siguientes colores principales. En la ilustración 22 se muestran los colores usados junto a su código de color en formato hexadecimal:

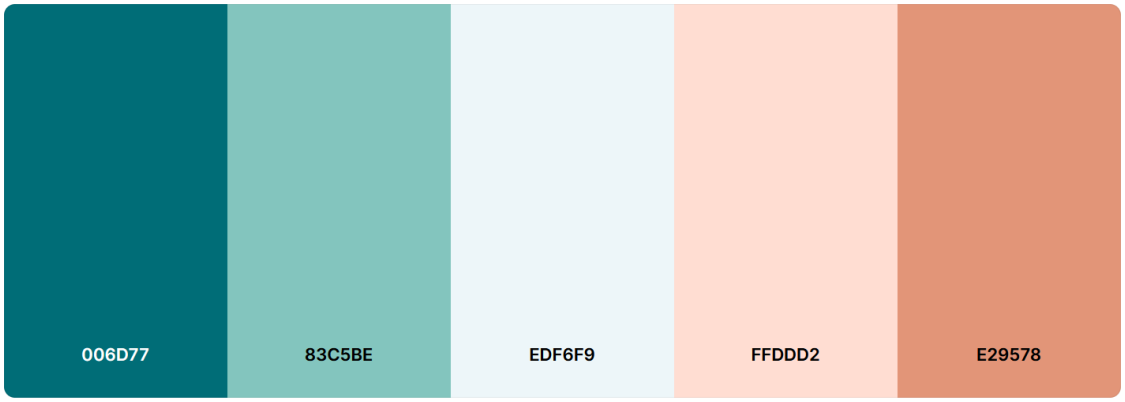

Ilustración 22 - Paleta de colores empleada

Esta paleta usada en la plataforma ha sido obtenida de la página web coolors.co [12].

4.3 Diagrama de flujo de reservas

Aunque ya se ha mencionado en la especificación de casos de uso, es importante destacar el proceso de reserva de un recurso. Para ello, se describirá de manera sencilla y se acompañará con un diagrama de flujo.

Primeramente, un recurso como ya se ha visto puede ser tanto un libro como una revista, como un DVD… etc. Estos recursos están disponibles en la biblioteca que se tenga como favorita o en otras que utilicen la plataforma. Si el recurso se encuentra en otra biblioteca aparecerá como no disponible.

Se pueden hacer tres acciones principalmente con los recursos en el ámbito de las reservas. Primeramente, si un recurso se encuentra en la biblioteca y no está pedido por nadie, se podrá pedir y una vez hecho esto se dispondrá de 3 días para ir a recogerlo y 15 días para disponer de él. Una vez pasado este tiempo se deberá devolver. Si el recurso ha sido pedido por otro usuario, se podrá reservar y una vez esté disponible pasará a dominio si nadie más lo ha reservado antes. Y al igual que en el caso anterior se dispondrá de un plazo de 3 días para ir a recogerlo y 15 días de posesión. Si en el caso contrario, alguien lo hubiera reservado ya, el usuario pasaría a estar en una cola de espera donde una vez llegue su turno podría disponer de él.

Para aclarar este procedimiento se ha creado un diagrama de flujo exhibido en la ilustración 23:

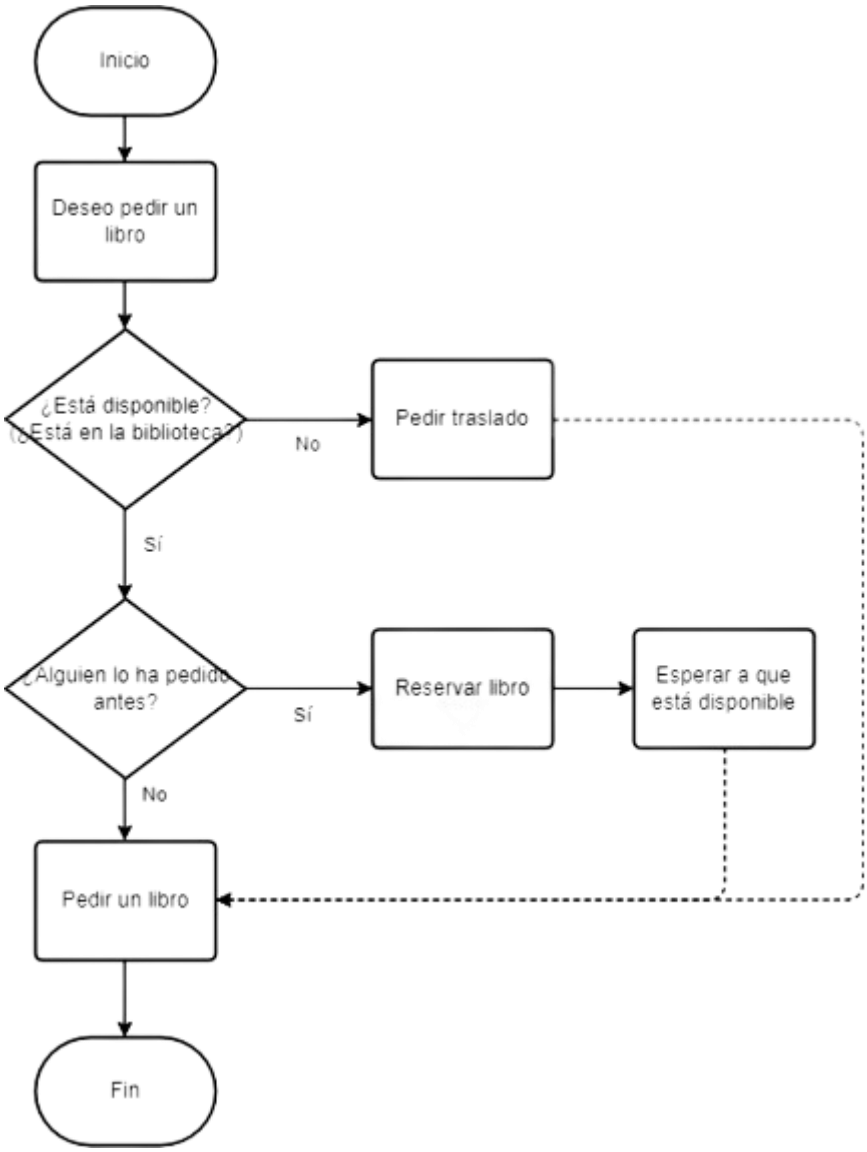

Ilustración 23 - Diagrama de flujo de reservas

5. Desarrollo de la solución propuesta

5.1 Tecnologías usadas

Se han utilizado varias tecnologías para desarrollar este proyecto de manera efectiva y eficiente. Estas tecnologías incluyen:

5.1.1 Angular

Es un marco o "framework" de desarrollo de aplicaciones web basado en JavaScript. Angular proporciona una estructura sólida y modular para crear interfaces de usuario interactivas y dinámicas. Es el lenguaje pilar de Bibliotecapp, se ha usado tanto en la capa de presentación de la aplicación como en la parte lógica de muchos componentes [13].

5.1.1.1 TypeScript

Es un lenguaje de programación basado en JavaScript que permite un desarrollo más escalable y seguro. Se utiliza en conjunto con Angular. A diferencia de JavaScript, TypeScript agrega características de tipado estático, lo que significa que se pueden declarar tipos de datos para variables, parámetros de función y valores de retorno. Estos tipos de datos ayudan a detectar errores en el código durante el desarrollo y mejoran la calidad y mantenibilidad del código. En él está escrito toda la lógica de Bibliotecapp hecha en Angular como los servicios de la plataforma [14].

5.1.1.2 HTML

HyperText Markup Language (HTML) es el lenguaje de marcado estándar utilizado para crear la estructura y contenido de las páginas web. Se trata de un lenguaje de etiquetas que define la semántica de los elementos en una página web, indicando cómo deben ser presentados y cómo interactúan entre sí. En Angular se utiliza para definir la estructura y el contenido de las aplicaciones web desarrolladas con este framework. Angular extiende HTML con características y directivas adicionales para agregar funcionalidad y dinamismo a las páginas web [15].

5.1.1.3 SCSS y CSS

Cascading Style Sheets(CSS) es un lenguaje de estilo utilizado para definir la apariencia y el diseño visual de una página web. Permite controlar aspectos como colores, fuentes, márgenes, espaciado, alineación, animaciones y mucho más. Los estilos CSS se aplican a los elementos HTML mediante selectores y reglas de estilo [16].

SCSS (Sassy CSS) es una extensión del lenguaje CSS que agrega características adicionales y mejoras. SCSS utiliza una sintaxis similar a CSS, pero con la capacidad de

utilizar variables, anidamiento de selectores, mixins, herencia y más. Esto hace que la escritura y el mantenimiento de estilos sean más eficientes y flexibles [17].

En Angular se pueden usar ambos, de hecho, en Bibliotecapp se utilizan ambos lenguajes. Esto es debido a que CSS es un lenguaje estándar y funciona en más navegadores, y para funciones específicas como el cambio de una variable dinámicamente funciona mejor CSS que SCSS. El motivo es que SCSS se compila a CSS y por tanto no hay modo de cambiar una variable durante la ejecución. Pese a esto, normalmente se opta por usar SCSS en la plataforma puesto que está más estructurado y por tanto es más fácil de leer y más escalable.

5.1.1.4 Angular Materials

Angular Materials es una biblioteca de componentes de interfaz de usuario predefinidos diseñados específicamente para aplicaciones web desarrolladas con Angular. Proporciona una amplia gama de componentes listos para usar, como botones, barras de navegación, formularios, diálogos, tablas y muchos más. También proporciona funcionalidades adicionales, como animaciones, gestión de eventos, control de estado y adaptabilidad a diferentes tamaños de pantalla [18].

5.1.2 Ionic Framework

Es un framework de desarrollo de aplicaciones móviles híbridas queº utiliza tecnologías web. Está construido sobre Angular. Permite crear aplicaciones multiplataforma que se ejecutan tanto en dispositivos iOS como Android utilizando un código base común [19].

5.1.3 Capacitor

Es una herramienta que se utiliza junto con Ionic para desarrollar aplicaciones móviles híbridas. Es un entorno de desarrollo que permite acceder a funcionalidades nativas de los dispositivos móviles, como la cámara, el GPS y las notificaciones, utilizando tecnologías web como Angular. Esta tecnología junto a Ionic han permitido crear la versión móvil de la aplicación [20].

5.1.4 NodeJS

Consiste en un entorno de ejecución de JavaScript en el servidor. Se ha usado NodeJS para crear la lógica del servidor y para realizar operaciones con la base de datos MySQL. Asimismo, se conecta a la capa frontal o “frontend” y lo conecta con la capa trasera o “backend” estableciendo una comunicación mediante solicitudes HTTP (HTTP Request) [21] [22].

5.1.5 Express

Express es un framework de aplicaciones web para Node.js. Proporciona una capa adicional de abstracción sobre el servidor HTTP de Node.js, simplificando el proceso de desarrollo de aplicaciones web y permitiendo crear servidores web de manera más rápida y sencilla. En Bibliotecapp se ha usado para la creación del servidor [23].

5.1.6 MySQL

MySQL es un sistema de gestión de bases de datos relacional. Proporciona una forma eficiente de almacenar, organizar y administrar grandes cantidades de datos de manera estructurada. En Bibliotecapp se utiliza como base de datos para almacenar y gestionar los datos de la aplicación [24].

5.1.7 Firebase

Firebase es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. En la aplicación pese a en un principio usar el servicio de la base de datos no relacional Realtime Database, finalmente se descartó y únicamente se usan los servicios de autentificación de Firebase Authentication y la base de datos de Firebase Storage.

Firebase Authentication proporciona una solución completa de autenticación y gestión de usuarios. En Bibliotecapp se utiliza tanto para el registro e inicio de sesión con Google o Facebook como para la gestión de los usuarios en la plataforma y el control de sus reservas [25].

Firebase Storage permite almacenar y servir archivos estáticos, como imágenes, videos, documentos y cualquier otro tipo de archivo, de manera segura y escalable. En esta aplicación o plataforma se usa para cargar y descargar archivos desde la aplicación móvil o web como son las portadas de los recursos o de los eventos. Se suben al servicio mediante la aplicación y se guarda la ruta relativa del archivo subido en la propia base de datos. Una vez se desee cargar las imágenes o vídeos, se descargan directamente de este servicio web [26].

5.1.8 GitHub

GitHub es una plataforma en línea que se utiliza para alojar y controlar versiones de proyectos de desarrollo de software utilizando el sistema de control de versiones Git. Proporciona un entorno colaborativo donde los desarrolladores pueden almacenar y compartir su código fuente, colaborar con otros miembros del equipo, realizar un seguimiento de los cambios realizados en el código y gestionar las diferentes versiones de un proyecto. En la plataforma se ha usado esta herramienta para hacer el control de versiones [27].

5.1.9 Postman

Postman es una herramienta en línea que se utiliza para realizar pruebas, documentar y trabajar con APIs (Interfaces de Programación de Aplicaciones). Proporciona un entorno completo para enviar solicitudes HTTP a diferentes endpoints de una API y recibir respuestas en tiempo real.

En primer lugar, se ha utilizado Postman para realizar pruebas en la API de Bibliotecapp. Esto implica enviar solicitudes HTTP, como GET, POST, PUT o DELETE, a los

diferentes endpoints de la API para probar su funcionalidad. Postman permite configurar los parámetros de la solicitud, como los encabezados, los datos de entrada o los parámetros de consulta, y proporciona una interfaz intuitiva para inspeccionar las respuestas recibidas. De esta manera, puede asegurarse que la API responda correctamente y se comporte como se espera.

Además de las pruebas, Postman también ha permitido documentar la API. Se han podido crear colecciones de solicitudes que representan las diferentes rutas y funcionalidades de la API. Dentro de cada solicitud, se han agregado descripciones detalladas que explican cómo utilizar correctamente la API. Esta documentación resulta útil para otros desarrolladores que trabajan con la API, ya que proporciona una referencia completa y clara de cómo interactuar con ella Postman es una herramienta en línea que se utiliza para realizar pruebas, documentar y trabajar con APIs (Interfaces de Programación de Aplicaciones). Proporciona un entorno completo para enviar solicitudes HTTP a diferentes endpoints de una API y recibir respuestas en tiempo real [28].

5.2 Implementación de la estructura

Tras haber explicado la estructura que se iba a utilizar, conocida como patrón Modelo-Vista-Controlador (MVC), se procederá a describir en detalle la implementación cada uno de sus componentes. A continuación, se detallan los componentes fundamentales del MVC y su implementación dentro de la aplicación de Bibliotecapp:

- Modelo (Model): responsable de la gestión de los datos y la lógica de negocio de nuestra aplicación. En Bibliotecapp esta capa la compone principalmente la base de datos MySQL puesto que es la fuente de datos principal. Asimismo, las tablas y esquemas MySQL representan entidades y relaciones dentro de la aplicación.

  Las entidades de la aplicación se definen mediante interfaces de TypeScript y son las que interactúan con la base de datos. A continuación, en la ilustración 24 se mostrará un ejemplo de una interfaz de la entidad Evento de la aplicación:

```
export interface Evento {
  id: string;
  nombre: string;
  descripcion: string;
  // Puede que la portada no tenga foto ni fecha
  portadaImgPath?: string;
  fecha?: Date;
}
```

Ilustración 24 - Ejemplo en código de la interfaz de Evento

En el ejemplo mostrado por la ilustración de arriba, la interfaz "Evento" define los campos y tipos de datos que conforman un evento en la aplicación. Por ejemplo, se tiene un campo "id" de tipo cadena de texto para identificar de manera única cada evento, un campo "nombre" del mismo tipo para almacenar el nombre del

evento, y un campo "descripción" de tipo cadena de texto para proporcionar información adicional sobre el evento. Además de estos campos, existen otros campos opcionales relacionados con el evento como son un campo "fecha" de tipo fecha para indicar la fecha en la que se realizará el evento y un campo "portadaImgPath" que haría referencia a la imagen o GIF que se mostrará junto al evento si hay una imagen distinta a la predefinida [11].

Estas interfaces permiten establecer una estructura clara y coherente para las entidades en la aplicación, lo que facilita su uso y mantenimiento a lo largo del desarrollo del proyecto.

- Vista (View): consistente en la representación visual de los datos y la interacción con el usuario. En la plataforma objeto de este proyecto se compone de componentes y plantillas HTML. Los componentes se encargan de recibir y presentar los datos del modelo a los usuarios. Y las plantillas HTML se utilizan para definir la estructura y el diseño de la interfaz de usuario. A continuación, un ejemplo gráfico:

```
<div id="main">
  <div id="fooder">
    <div id="primeraCol">
      <p class="tituloHorario">Horario:</p>
      <p><span> Lunes a Viernes </span> 9:00 a 21:00</p>
      <p><span>Sábado </span> 9:00 a 14:00</p>
      <p><span>Domingo </span> Cerrado</p>

      <div id="divCalendario">
        <mat-card class="demo-inline-calendar-card">
          <mat-calendar></mat-calendar>
        </mat-card>
      </div>      Esteve Carbonell external, 7 months ago • Pre prettier
    </div>

    <div id="segundaCol">
      <p><span>Biblioteca 33</span></p>
      <p><span>C/</span>Carrer la Palmera, 7, 03320 Torre del Pla, Alicante</p>
      <p><span>Teléfono: </span>666 666 666</p>
      <img
        class="imgBiblio"
        alt="Imagen Biblioteca"
        src="../assets/images/principales/biblio.jpg"
        width="50%"
      />
    </div>
  </div>

  <div id="postfooder">
```

Ilustración 25 - Ejemplo en código de la plantilla de "conocenos"

Esta sería la plantilla HTML que creará la vista mostrada en la siguiente imagen. Este código es una parte de la plantilla llamada "conocenos.component.html" que

está dentro del componente “conocenos”. Este es un componente sin mucha carga lógica y que muestra en este caso la información de una biblioteca de la aplicación, la Biblioteca 33.

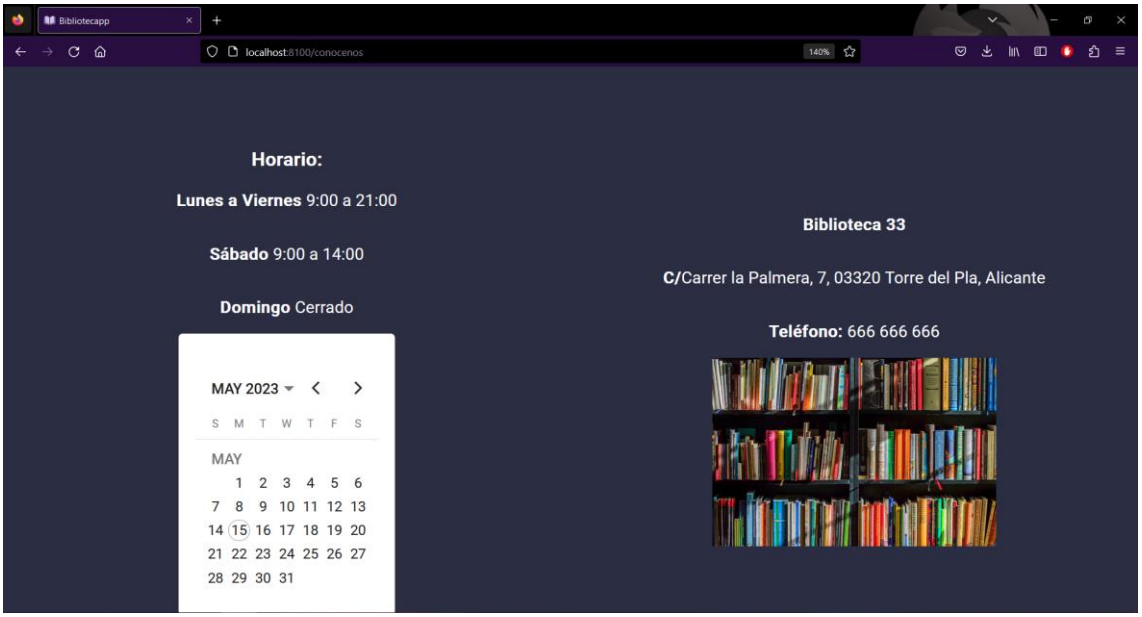

Ilustración 26 - Vista de la plantilla "conocenos"

La ilustración 26 presenta la vista que se muestra una vez se ha abierto en un navegador el código de la ilustración 25.

- Controlador (Controller): intermediario entre el modelo y la vista. En este proyecto, los servicios de Angular actuarían como el controlador. Los servicios se encargan de obtener los datos del modelo desde MySQL y proporcionarlos a los componentes para su visualización. También realizan operaciones de escritura y actualización en MySQL. Además, los servicios manejan la autenticación y autorización utilizando Firebase Authentication.

  A continuación, en la ilustración 27, se presenta un ejemplo de servicio que llama a la base de datos MySQL para obtener los libros disponibles o dar de alta uno. (El servicio es más amplio y tiene más funcionalidad, pero se ha recortado para la ejemplificación).

Ilustración 27 - Ejemplo en código del servicio de "libro.service"

Además de los tres componentes principales cabe destacar la capacidad de enlace de datos (Data Binding) que proporciona Angular. Esta, proporciona enlaces de datos bidireccionales que permiten sincronizar automáticamente los datos entre el modelo y la vista. Esto facilita la actualización y visualización de los datos en tiempo real.

Asimismo, cabe señalar que Angular (la tecnología pilar de este proyecto) no sigue estrictamente el patrón MVC de forma pura, sino que proporciona una estructura que se inspira en estos patrones y se adapta a las necesidades de cada desarrollo. La estructura y distribución de las tecnologías varía según el enfoque y las necesidades de cada proyecto. Pese a esto, en Bibliotecapp se ha intentado seguir lo máximo posible este patrón de diseño implementado por Angular siguiendo sus directrices definidas.

5.3 Representación de las tecnologías en la estructura

Una vez que presentadas las tecnologías utilizadas, es importante comprender cómo se integran dentro del patrón Modelo-Vista-Controlador (MVC) y el papel que desempeñan en cada una de las capas.

Como se ha mencionado previamente la capa de modelo, la componen tanto las interfaces como la base de datos MySQL. Las interfaces definidas en TypeScript representan las entidades de la aplicación y definen los campos y tipos de datos que conforman cada una. Por otro lado, la base de datos MySQL actúa como la fuente de datos principal en la capa de modelo. Aquí es donde se almacenan y gestionan los datos de la aplicación, incluyendo las entidades y las relaciones entre ellas.

La capa de vista la compondría principalmente Angular porque es necesario recordar que es una tecnología enfocada en la parte frontal de la aplicación junto a estilos proporcionados por librerías externas. En el caso de Bibliotecapp esas librerías externas que mejoran la apariencia visual y la experiencia del usuario las proporcionaría Angular Material.

Finalmente, la capa de controlador es responsable de manejar la lógica de negocio de la aplicación, procesar las solicitudes del cliente y coordinar las interacciones entre la capa de modelo y la capa de vista. Esta capa la formarían tanto NodeJS y Express, como Firebase Authentication y otros servicios definidos en Angular. NodeJS y Express son fundamentales para gestionar las rutas y los controladores que manejan las solicitudes del cliente. NodeJS proporciona un entorno de tiempo de ejecución eficiente y Express simplifica la definición de rutas y el manejo de las solicitudes y respuestas HTTP.

Por otro lado, Firebase Authentication y los servicios web definidos en Angular ofrecen funcionalidades adicionales relacionadas con la autenticación de usuarios y la implementación de la lógica de negocio específica de la aplicación.

A continuación, la ilustración 28 muestra la representación de estas en la estructura:

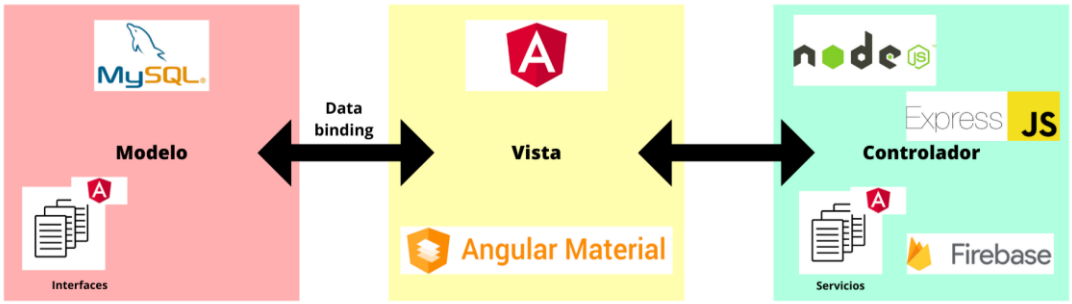

Ilustración 28 - Representación de las tecnologías empleadas en la estructura MVC

Como se puede observar en la ilustración no se han incluido ni Ionic Framework ni Capacitor puesto que su ubicación no está tan clara. Principalmente se sitúan en la capa de vista, ya que se utilizan para construir la interfaz de usuario y gestionar la presentación de datos. Además, parte de la lógica de control puede residir en los controladores de Ionic. Sin embargo, la gestión y manipulación de los datos en la capa de modelo, así como la comunicación con el backend, estarían fuera del alcance directo de Ionic y Capacitor.

5.4 Otras herramientas utilizadas

5.4.1 Visual Paradigm Online

Todos los diagramas utilizados en la creación de este documento han sido generados utilizando Visual Paradigm Online. Visual Paradigm Online es una

herramienta en línea que proporciona una amplia variedad de herramientas de diagramación para crear diagramas profesionales de manera eficiente [29].

5.4.2 Canvas

Para la creación de la aplicación, se ha utilizado la herramienta de diseño gráfico en línea, Canva. Esta herramienta ha permitido crear el logo de la aplicación, así como los iconos de los botones y todas las imágenes utilizadas en la plataforma. Canva es una plataforma que ofrece una amplia gama de plantillas, elementos gráficos y herramientas de edición para crear diseños atractivos y profesionales [30].

5.4.3 Visual Studio Code

Todo el código de la aplicación se ha desarrollado utilizando Visual Studio Code. Visual Studio Code es un IDE ligero y potente que ofrece características y extensiones para facilitar el desarrollo de software [31].

5.4.4 Figma

Con Figma, se han creado bocetos detallados de la aplicación, facilitando la parte de desarrollo del diseño. Figma es una herramienta web que ofrece una amplia gama de características para diseñar y prototipar aplicaciones [32].

5.4.5 Android Studio

Una vez generada la aplicación con Ionic, se requirió realizar modificaciones para asegurar que la funcionalidad de la aplicación fuera consistente tanto en la versión web como en la versión móvil. Para lograr esto, fue necesario realizar cambios en los archivos generados por el entorno de Android Studio. [33].

5.5 Tablas de la Base de Datos

Para el backend de la plataforma se decidió crear una base de datos, puesto que es importante tener un almacenamiento estructurado de datos, en un lugar seguro y de acceso concurrente además de que se pensó que sería la solución más eficiente para el manejo de datos.

A continuación, una representación de las tablas de la base de datos con sus relaciones. Esta representación se muestra en la ilustración 29 y ha sido creada mediante la herramienta de ingeniería inversa de MySQL Workbench:

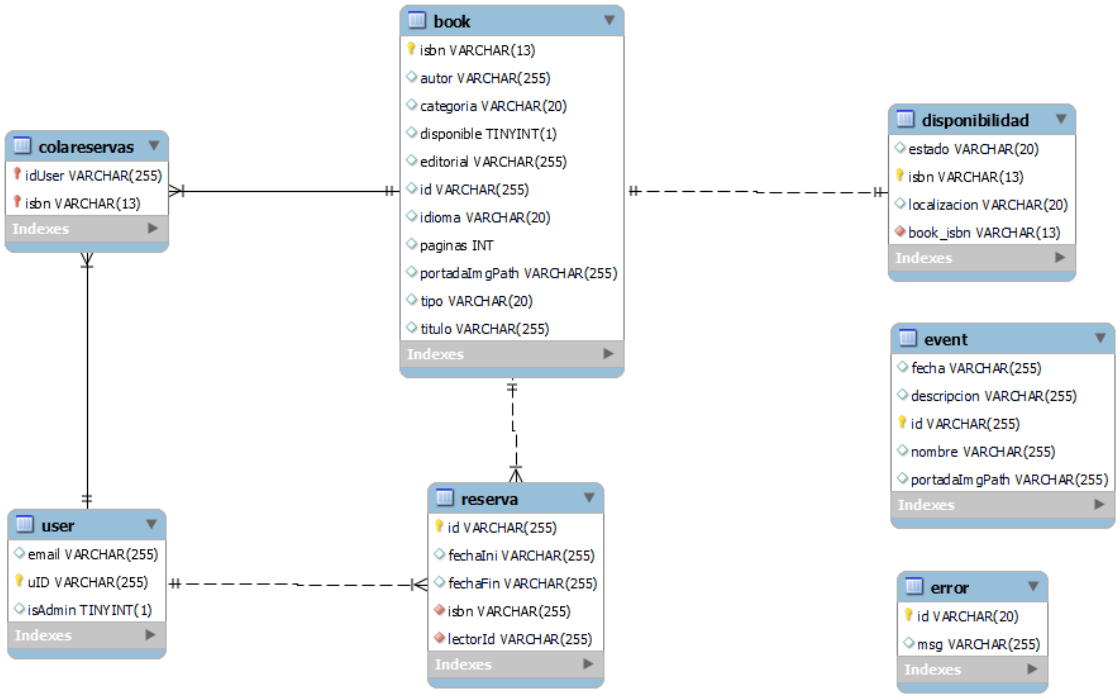

Ilustración 29 - Representación de las tablas de la base de datos

La base de datos del proyecto se compone de 7 tablas. Cada tabla representa una entidad de la plataforma que serán usadas por las interfaces del frontend.

La tabla “book” o mejor llamada “recurso” es la más importante puesto que representa todos los recursos disponibles en la plataforma. Cada fila de esta tabla corresponde a un recurso específico y almacena información relevante sobre ese recurso, como su ISBN (o id si no es un libro), autor, categoría, disponibilidad, editorial (opcional), idioma, número de páginas (opcional), tipo, imagen de portada (opcional) y título. Esta tabla permite realizar consultas y realizar operaciones relacionadas con la gestión de recursos, como agregar nuevos recursos, actualizar la información existente, buscar recursos por autor, categoría o título, y verificar la disponibilidad de un recurso en particular (por el atributo “disponible”). Además, contiene el enlace al servidor donde está almacenada su imagen de portada si es que tiene.

La tabla “event” representa los eventos que pueden ocurrir en la biblioteca como podría ser un club de lectura los jueves o el día de Sant Jordi. Cada fila en la imagen representa los atributos de esta tabla como son la fecha en la que se celebrará este evento, la descripción del mismo, un id para identificarlo, un título que aparecerá para resaltar el evento y finalmente la ruta de la portada (si es que tiene) en la base de datos de Firebase Storage.

La tabla "error" o mejor llamada “comentario” almacena información sobre comentarios en la base de datos. Cada fila representa un comentario específico y contiene un identificador único y un mensaje descriptivo del mismo. Esta tabla permite registrar y gestionar comentarios ocurridos en el sistema para su posterior análisis y resolución si

fuera el caso o simplemente para tener una retroalimentación por parte de los usuarios de la plataforma a los administradores.

La tabla "user" almacena información de usuarios en la base de datos. Cada fila representa un usuario y contiene su dirección de correo electrónico, un identificador único y un indicador de si el usuario es un administrador. Esta tabla permite gestionar y autenticar usuarios en la plataforma Bibliotecapp. También se puede comprobar y operar sobre los usuarios desde el servicio externo de Firebase Authentication. Para algunas acciones como eliminar usuarios antiguos la única manera será accediendo a esta página y operando en ella.

La tabla "reserva" almacena información sobre las reservas realizadas en el sistema. Cada fila representa una reserva específica y contiene un identificador único, fecha de inicio y fin de la reserva, el ISBN (o id) del recurso reservado y el identificador del usuario (número de carné de la biblioteca) que realiza la reserva. Esta tabla permite rastrear y gestionar las reservas de recursos en la base de datos. Esta tabla únicamente guarda las reservas activas. Una reserva deja de estar activa cuando se el recurso devuelve a la biblioteca y no cuando se vence el tiempo de fin de reserva. Si el tiempo de fin de reserva se venciera (unos 15 días después de recoger el recurso por primera vez) aparecería en rojo en el apartado de “Mis Reservas”. Y si esto se mantuviese la biblioteca debería tomar las medidas apropiadas.

La tabla "disponibilidad" almacena información sobre la disponibilidad de los recursos en la base de datos. Cada fila representa un recurso específico y contiene detalles como el estado de disponibilidad (por ejemplo, "disponible" o "no disponible"), el ISBN del libro (o id del recurso) y la ubicación física del libro. Esta tabla permite verificar y gestionar la disponibilidad de los libros en la base de datos. Está relacionada con el atributo disponible de la tabla “book”. Esta tabla es útil si hay diversas bibliotecas que la plataforma gestiona puesto que se puede obtener de cada recurso su ubicación dentro de la red de bibliotecas y su disponibilidad y hacer operaciones como el traslado de recursos entre bibliotecas.

Finalmente, la tabla “cola de reservas” como su propio nombre indica, almacena información sobre las colas de reservas de cada recurso en la base de datos. Cada fila representa una reserva pendiente. Esta reserva se efectuará en orden de llegada cuando el recurso deje de estar en posesión de otro usuario. Esta tabla contiene el identificador del usuario que realiza la reserva y el ISBN del libro (o la id del recurso) reservado. Esta tabla permite gestionar el orden de las reservas y controlar qué usuarios están esperando para obtener un recurso específico.

Tras la definición de las tablas, cabe destacar las relaciones entre las mismas. Tras la definición de las tablas, cabe destacar las relaciones entre las mismas. La base de datos presenta las siguientes relaciones entre las tablas:

- Relación entre "Book" y "Reserva": La tabla "Reserva" tiene una clave externa llamada "isbn" que se refiere a la clave primaria "isbn" en la tabla "Book". Esta relación establece que una reserva está asociada a un recurso específico. Permite rastrear qué recursos han sido reservados y a quién pertenecen esas reservas.
- Relación entre "User" y "Reserva": La tabla "Reserva" tiene una clave externa llamada "lectorId" que se refiere a la clave primaria "uID" en la tabla "User". Esta relación indica qué usuario ha realizado una reserva en particular. Permite asociar cada reserva con el usuario correspondiente y facilita la gestión de las reservas realizadas por cada usuario.
- Relación entre "Book" y "Disponibilidad": La tabla "Disponibilidad" tiene una clave primaria llamada "isbn" que se refiere a la clave primaria "isbn" en la tabla "Book". Esta relación establece la conexión entre la disponibilidad de un recurso y el recurso mismo. Permite verificar la disponibilidad de un recurso consultando su registro correspondiente en la tabla "Disponibilidad".
- Relación entre "User" y "ColaReservas": La tabla "ColaReservas" tiene una clave externa llamada "idUser" que se refiere a la clave primaria "uID" en la tabla "User". Esta relación indica qué usuario se encuentra en la cola de reservas para un recurso determinado. Permite mantener un seguimiento del orden de las reservas y determinar qué usuarios están esperando para obtener un recurso específico.

Estas relaciones establecen vínculos significativos entre las tablas y facilitan consultas y operaciones que involucran datos relacionados. Por ejemplo, se pueden obtener todas las reservas realizadas por un usuario, verificar la disponibilidad de un recurso en particular o determinar el próximo usuario en la cola de reservas para un recurso específico. Estas relaciones son esenciales para mantener la integridad de los datos y facilitar la gestión eficiente de la base de datos.

Cabe aclarar que la variable “book_isbn” no existe realmente, si no que se relaciona el “isbn” de la tabla “disponibilidad” con el “isbn” de la tabla “book”. Asimismo, pese a la tabla llamarse “book” realmente representa cualquier recurso de la plataforma como comics, ebooks o DVDs. De igual forma la tabla “error” representa los comentarios escritos en la plataforma, pudiendo ser tanto errores reportados como recomendaciones o simplemente comentarios normales de usuarios. Finalmente, los parámetros booleanos han sido definidos como “TINYINT” puesto que así se define en las convenciones de nomenclatura establecidas en el estándar SQL y en MySQL. Un “TINYINT” ocupa 1 byte de almacenamiento y es el tamaño más pequeño disponible para almacenar valores enteros.

5.6 Peticiones HTTP y API

Las peticiones HTTP son solicitudes que un cliente envía a un servidor web para interactuar con él y obtener o enviar información. Estas solicitudes se basan en el

protocolo de transferencia de hipertexto (HTTP) y se utilizan ampliamente en el desarrollo web para obtener recursos, enviar datos y realizar diferentes acciones en un servidor.

Como ya se ha mencionado previamente, en el caso del proyecto, se ha utilizado Express y Node.js para crear y gestionar estas peticiones HTTP en el servidor. Las peticiones HTTP que se han creado para la plataforma han sido las siguientes:

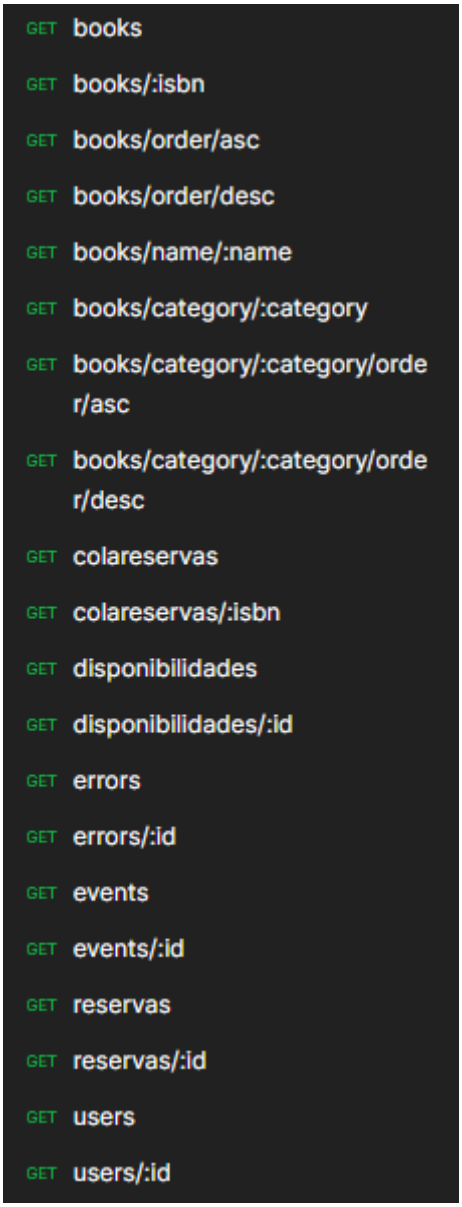

Ilustración 30 - Lista de peticiones HTTP de tipo GET existentes en la plataforma

Ilustración 31 - Lista de peticiones HTTP de tipo DELETE existentes en la plataforma

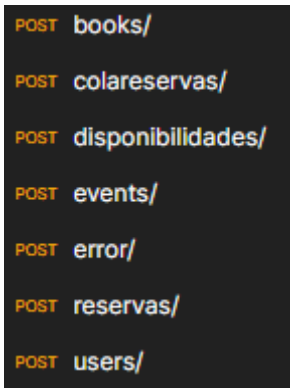

Ilustración 32 - Lista de peticiones HTTP de tipo POST existentes en la plataforma

Ilustración 33 - Lista de peticiones HTTP de tipo PUT existentes en la plataforma

Estas cuatro ilustraciones (Ilustración 30, 31, 32, 33) forman el listado de todas las peticiones que se pueden hacer desde Bibliotecapp. Puesto que son demasiado numerosas para hablar sobre que hace cada petición en este TFG, se ha creado una documentación externa a este documento de la API disponible en el anexo II o asimismo, en el siguiente enlace: https://documenter.getpostman.com/view/23029514/2s93mAVLWA [34].

Esta documentación ha sido creada con la ayuda de Postman. Esta documentación estará activa para que cualquier usuario o desarrollador de la plataforma pueda documentar, tratar o incluso ampliar. Es recomendable visualizar las llamadas en formato HTTP, ya que es el formato en el que se han creado. Pese a esto, se podrían llegar a llamar desde una aplicación externa con cualquier otro lenguaje.

5.7 Patrones de diseño utilizados

En este apartado se enumerarán algunos de los patrones de diseños empleados en el desarrollo de Bibliotecapp:

5.7.1 Patrón de arquitectura MVC

Patrón de arquitectura Modelo-Vista-Controlador: Como ya se ha mencionado previamente, Angular sigue este patrón para estructurar las aplicaciones web. El Modelo representa los datos y la lógica, la Vista se encarga de los aspectos visuales, y el Controlador actúa como intermediario que maneja las interacciones entre el Modelo y la Vista.

5.7.2 Singleton

El patrón Singleton es un patrón de diseño que garantiza que solo exista una única instancia de una clase en toda la aplicación, lo cual es especialmente útil en Angular para crear servicios.

En Angular, los servicios son clases que proporcionan funcionalidades compartidas en la aplicación. Al utilizar el patrón Singleton para los servicios, Angular asegura que solo haya una instancia de cada servicio y que esta instancia sea compartida entre todos los componentes que lo requieran.

Para lograr el patrón Singleton en Angular, se utiliza la directiva @Injectable. Al decorar una clase con @Injectable, Angular se encarga de crear una única instancia del servicio y la mantiene disponible para su inyección en otros componentes que lo necesiten. Esta directiva marca la clase como un servicio y habilita la inyección de dependencias. Angular se encarga de crear y administrar automáticamente la instancia del servicio, asegurando que solo exista una instancia globalmente accesible.

Por ejemplo, la ilustración 34 representaría un ejemplo de la cabecera de un servicio siguiendo este patrón. En este caso, es la definición del servicio de recursos que hace todas las operaciones de la parte lógica relacionada con los recursos y su gestión.

```
@Injectable({
  providedIn: 'root',
})
export class LibrosService {

  constructor(private httpClient: HttpClient) {}
```

Ilustración 34 - Ejemplo en código de la cabecera de un servicio usando el patrón Singleton

Si se desease usar este servicio en otra parte del código simplemente invocándolo como se muestra en la ilustración 35, se crearía una instancia compartida para toda la aplicación:

```
export class InfoLibroComponent implements OnInit {
  libroSeleccionado: Libro;
  currentUser?: User;

  constructor(
    private libroService: LibrosService,
    private tituloService: TitulosService,
    private reservasService: ReservasService,
    private storageService: StorageAndroidService,
    private colaService: ColaReservasService,
    public dialog: MatDialog
  ) {}
```

Ilustración 35 - Ejemplo en código de la instancia de un servicio usando el patrón Singleton

El uso del patrón Singleton en Angular aporta beneficios en términos de consistencia y coherencia en los datos y funcionalidades compartidas. Además, tiene ventajas en cuanto a eficiencia y rendimiento, ya que se evita la duplicación innecesaria de recursos y se optimiza el consumo de memoria al tener una única instancia del servicio.

5.7.3 Observador

El patrón Observador permite establecer una comunicación eficiente entre objetos, donde un objeto, llamado publicador, notifica y actualiza automáticamente a otros objetos, llamados observadores, sobre cualquier cambio de estado. La comunicación entre el publicador y los observadores se realiza mediante la emisión y la suscripción de eventos. La librería utilizada para gestionar estas emisiones y suscripciones de eventos es RxJS [35]. Los publicadores en RxJS se implementan mediante Subjects que actúan como emisor y receptor al mismo tiempo. Asimismo, los observadores se llaman Observables.

En Bibliotecapp se utiliza en varios escenarios, como la comunicación entre componentes o entre servicios. Por ejemplo, cuando se quieren obtener los recursos de la base de datos, primeramente, se hace una petición HTTP. Una vez obtenida la respuesta, un Subject la manda a todos los componentes que la necesitan (pantalla principal, detalle del recurso, pantalla de administración de recursos...). Cada uno de estos componentes previamente

estaba esperando con un Observable una respuesta y una vez la obtienen ya trabajan con ella. En el caso de la pantalla principal, muestra todo el catálogo de recursos que dispone la biblioteca. Tanto los Subject como los Observables se quedan esperando puesto que podría cambiar el estado de los libros. Si por ejemplo en tiempo de ejecución se borrara un recurso de la base de datos automáticamente y gracias a la implementación de este patrón quedaría reflejado en cada componente que está escuchando estos cambios automáticamente.

5.8 Estructura de carpetas

La estructura de carpetas en Angular desempeña un papel fundamental en el desarrollo de aplicaciones web, ya que ayuda a mejorar la legibilidad del código, la escalabilidad del proyecto y la colaboración en equipos de desarrollo.

En Angular, la estructura de carpetas no está estrictamente definida ni en el marco de trabajo ni por las mejores prácticas de los desarrolladores de la tecnología. Es por esto que cada proyecto tiene cierta flexibilidad a la hora de organizar sus archivos. Sin embargo, existen convenciones comunes en la estructura de carpetas creadas por Google y la comunidad de Angular.

La compañía creó la guía de estilos de código ("Angular coding style guide") para ayudar a los desarrolladores a seguir ciertas convenciones [36]. En este proyecto, se ha aplicado la convención LIFT para mejorar la estructura de carpetas. Esta convención tiene como objetivo principal facilitar la ubicación de archivos, identificar rápidamente su propósito, evitar niveles de anidamiento excesivos y promover la reutilización de código. Las siglas significan:

- "Locate": la estructura de la aplicación debe permitir localizar código rápidamente.

- "Identify": los nombres de los archivos deben ser descriptivos y precisos. Es adecuado saber inmediatamente que contiene y que representa.

- "Flat": un fichero no debe tener muchos archivos. Se debe considerar crear subficheros si una carpeta llega a siete o más archivos. Esto es debido a que buscar en una larga lista de archivos cansa a los desarrolladores.

- "Try to be DRY": DRY se entiende como: "Don't Repeat Yourself" (No te repitas). Esta regla se entiende como que los nombres de los archivos no deberían incluir aspectos redundantes. Por ejemplo, un archivo llamado "vista-eventos.html" es redundante puesto que la extensión .html indica que es un archivo de vista. Por tanto, debería renombrarse a "eventos.html". De todos modos, como dice la regla: "Try" (intenta), es decir, a veces es conveniente crear nombres de ficheros o archivos redundantes puesto que quizá para algunos usuarios sean más fáciles de comprender.

En Bibliotecapp se ha seguido la estructura de carpetas que se muestra en la ilustración 36:

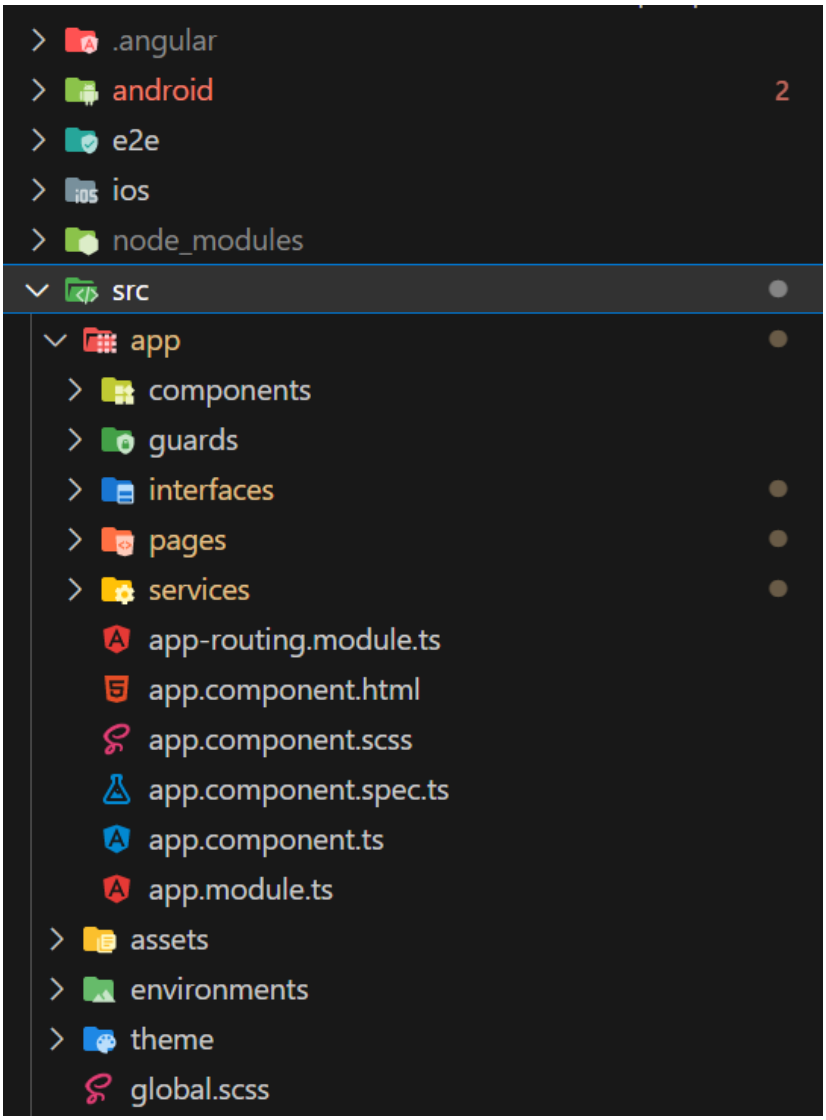

Ilustración 36 - Estructura de carpetas

- .angular: guarda caché de las librerías usadas.
- android/ ios: guarda el código del proyecto para cada versión móvil. En Android el lenguaje del código generado es Kotlin y en iOS es Swift.
- e2e: contiene archivos de las pruebas "end-to-end". Son pruebas automatizadas que imitan todas las acciones y comportamientos que un usuario podría realizar en la aplicación.
- node_modules: está compuesto de librerías y paquetes externos al proyecto Angular, pero que este utiliza. Por ejemplo, en esta carpeta estaría la librería de Firebase.
- src: es la carpeta raíz del proyecto.
    - app: es la carpeta que almacena el código de la plataforma.
        - components: es la carpeta donde se guardan los componentes (o elementos) reutilizables de la aplicación.
        - guards: se guardan archivos relacionados con la seguridad en el acceso a ciertas rutas de la plataforma.

        - interfaces: es una colección de las interfaces de la aplicación.
        - pages: es una carpeta donde se guardan las diferentes pantallas de la aplicación. Estas pantallas se crean usando los componentes.
        - services: es una carpeta para almacenar los servicios utilizados para lógica de negocio.
    - assets: directorio para almacenar los archivos estáticos como imágenes, fuentes, gifs, iconos y videos.
    - environments: es un directorio utilizado para almacenar los archivos de configuración de entorno.
    - theme: contiene archivos relacionados con la apariencia visual y el estilo de la aplicación. Por ejemplo, el estilo de un botón o de una tabla están declarados aquí para que siempre que un componente quiera hacer uso del mismo pueda hacerlo fácilmente.

Normalmente todos los proyectos Angular tienen una carpeta llamada “shared” que almacena los componentes o las directivas compartidas de la aplicación. En Bibliotecapp no se utiliza la carpeta “shared” puesto que los recursos compartidos como componentes ya se guardan en esta misma carpeta.

6. Implantación

6.1 Dificultades en el desarrollo

En esta sección del TFG, se desea abordar y analizar algunas de las principales dificultades que se ham enfrentado durante el desarrollo de la plataforma bibliotecaria. A través de una exploración detallada de estos desafíos, se pretende identificar las soluciones adoptadas, los aprendizajes obtenidos y cómo se han abordado estas dificultades en el contexto específico del proyecto.

6.1.1 Implementación de Ionic

Al comienzo del proyecto, se decidió desarrollar la aplicación utilizando Angular únicamente. Se implementó una estructura completa conectada con base de datos Firebase y servicios como Firebase Authentication y Storage. Durante el desarrollo se identificó la necesidad de crear una versión móvil de la aplicación para alcanzar a un público más amplio. Además, se requería un rendimiento óptimo y una experiencia de usuario fluida en dispositivos móviles. Pese a intentar suplir este nuevo requisito creando una aplicación responsive que luego se convertiría en una PWA, Se llegó a la conclusión de que crear un proyecto Ionic sería la mejor opción.

La decisión de cambiar a Ionic implicó una reconstrucción significativa del proyecto. Se tuvieron que adaptar las funcionalidades existentes y todas las vistas se tuvieron que cambiar para usar componentes y estilos Ionic que eran más propicios para trabajar sobre aplicaciones híbridas. Pese a esto, el cambio a Ionic trajo consigo varios beneficios. Permitió desarrollar una aplicación funcionaba tanto en dispositivos iOS como Android. Y optimizó el rendimiento en dispositivos móviles, ya que, usando Capacitor para acceder a las características nativas del dispositivo resultaba en un rendimiento más rápido y una mejor experiencia de usuario en comparación con una aplicación web tradicional. Aunque el cambio de Angular a Ionic implicó un esfuerzo significativo y una reestructuración completa del proyecto, los resultados justificaron el esfuerzo invertido.

6.1.2 Firebase Realtime Database

Inicialmente, se utilizó Realtime Database de Firebase junto con la librería de Angular: AngularFireDatabase (descontinuada, actualmente se llama AngularFire2) [37]. Realtime Database es una base de datos en tiempo real en la que almacenaba los datos de los recursos, usuarios, reservas... en tiempo real. Esto permitía una actualización instantánea de la información en todos los dispositivos conectados. Además, facilitaba el acceso a los datos desde todos los dispositivos debido a que estaban en un servidor externo. Pese a esto, el desarrollo de Bibliotecapp, surgieron varios problemas y limitaciones que afectaron la plataforma y a su flexibilidad. La complejidad del esquema de datos, la falta de consultas avanzadas y la escalabilidad limitada fueron algunos de los problemas identificados. Teniendo en cuenta los problemas mencionados anteriormente,

se tomó la decisión de cambiar de Realtime Database a MySQL como base de datos principal. Esta elección se basó en la estructura de datos relacional de MySQL, las consultas avanzadas y funcionalidades que ofrece, así como su escalabilidad y rendimiento.

Para migrar la base de datos original creada con Firebase Realtime Database y codificada en formato JSON, a MySQL, se realizó un proceso de conversión:

6.1.2.1 Migración de datos

Inicialmente, fue necesario importar el formato JSON en el que estaba la base de datos original, a la base de datos local en MySQL. Para lograr esto, se usaron herramientas en línea. En primer lugar, se empleó la página web ConvertCSV [38] para convertir el JSON a formato XLSX, que es el formato utilizado por programas como Excel. Esto permitió trabajar con los datos de manera más eficiente y realizar las transformaciones necesarias para adaptarlos a la estructura de tablas de MySQL. Debido al tamaño y la complejidad de la base de datos, se optó por realizar esta conversión en partes, es decir, por secciones de datos. Esto permitió abordar el proceso de manera más efectiva y evitar posibles errores en la migración. Una vez obtenidos los datos en formato XLSX, se usó la herramienta en línea de conversión de ASPOSE [39] para convertir XLSX en SQL. Estas herramientas junto a conocimiento de bases de datos relacionales facilitó la creación de tablas SQL que luego se importarían a MySQL.

Este proceso de migración fue cuidadosamente ejecutado para garantizar la integridad y la consistencia de los datos durante la transición de Firebase Realtime Database a MySQL puesto que se manejaba gran cantidad de datos de prueba ya creados.

6.2 Arquitectura cliente-servidor

En el siguiente apartado se comentará la arquitectura cliente-servidor empleada en el funcionamiento de la plataforma de Bibliotecapp. La arquitectura cliente-servidor que implementada en la aplicación es un modelo de distribución de software en el que se establece una clara separación de responsabilidades entre dos componentes principales: el cliente y el servidor.

El cliente es la parte de la aplicación que interactúa directamente con la aplicación final creada usando Angular principalmente. La parte cliente pueden ser por tanto usuarios de la biblioteca o administradores de la biblioteca. La parte cliente puede operar y hacer todas las funciones recogidas en los casos de uso. Esta parte se ejecuta en los dispositivos de los usuarios, como ordenadores, tabletas o teléfonos, y se comunica con el servidor para realizar sus operaciones. La parte cliente puede tiene 3 variantes. La parte cliente puede ser ejecutada en versión web, en versión de aplicación nativa de Android o en versión de aplicación nativa de iOS. Como se mostrará en la próxima ilustración, la parte cliente se representaría a la derecha del dibujo en versión web o versión móvil.

El servidor es el componente central de la arquitectura y se encarga de gestionar y almacenar la información de la biblioteca, así como de procesar las solicitudes de los clientes. Está diseñado para ser escalable y robusto, capaz de manejar múltiples solicitudes simultáneas de diferentes clientes. Asimismo, debe estar disponible el mayor tiempo posible sin fallar para no comprometer el funcionamiento de la aplicación. El servidor almacena la base de datos de los recursos, los eventos, la información de los usuarios, los historiales de reservas y todas las tablas descritas en apartados anteriores. Además, implementa la lógica de negocio necesaria para asegurar el funcionamiento de la aplicación y garantizar la integridad de los datos manejados. Contiene el servidor por tanto una versión de Bibliotecapp diferente a la parte cliente. Esta parte que contiene el servidor para manejar toda la lógica de la aplicación relacionada con la base de datos está separada. La parte de la plataforma de Bibliotecapp que tiene el servidor es la que está desarrollada usando NodeJS, Express y MySQL y nada de Angular. Por consiguiente, se podría considerar que la aplicación está dividida en dos. En la ilustración 37, el servidor quedaría representado por el dibujo de la parte izquierda de la imagen:

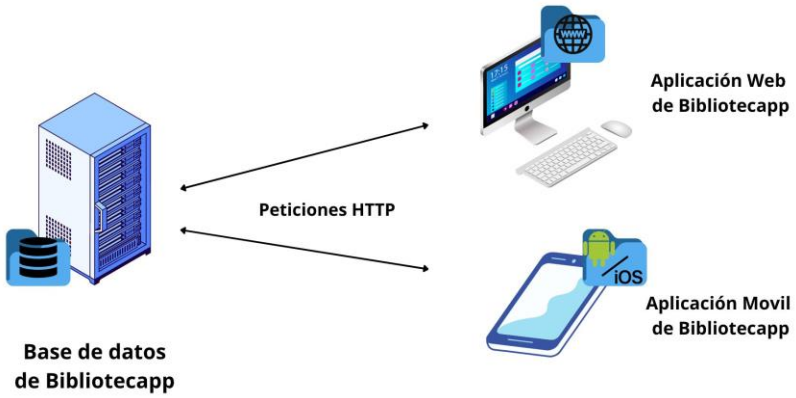

Ilustración 37 - Representación de las interconexiones de la plataforma

La comunicación entre el cliente y el servidor se lleva a cabo a través de la red utilizando el protocolo HTTP como se representa en la ilustración previa mediante el uso de flechas bidireccionales. El cliente envía solicitudes al servidor, como buscar un libro, y el servidor responde a esas solicitudes proporcionando los datos solicitados o un error si no se ha encontrado este recurso. Esta comunicación se realiza de manera transparente para el usuario. Pese a esto se pueden replicar las llamadas HTTP que hace la parte servidor para ver los resultados que se envían en estos mensajes. En la ilustración 38 se muestra una llamada HTTP GET para obtener todos los recursos de la plataforma y sus detalles.

Ilustración 38 - Captura de pantalla del resultado de una petición HTTP de tipo GET al conectarse al backend desde un dispositivo móvil

Como se puede observar el servidor envía un archivo en formato JSON [40] a la parte cliente mediante el puerto 3000. Y esta parte la interpretará y mostrará los resultados esperados en la forma deseada.

6.3 Versión de aplicación móvil

Una vez creada una arquitectura estable e independiente de la plataforma, se desarrolló la parte de la aplicación móvil. Al estar creada en Ionic el paso fue relativamente sencillo.

Con la tecnología de Capacitor se generó el código de la parte móvil tanto en Kotlin para la versión de Android, como en Swift para la versión de iOS. Una vez hecho esto, se deberieron asignar los permisos necesarios en el código de las aplicaciones para que se pudieran conectar a internet, que se pudiera leer y escribir en su almacenamiento interno y se pudiese acceder a la cámara. El acceso a la cámara es opcional. Es útil para los usuarios administradores que deseen subir un recurso o un evento y hacer una foto desde su dispositivo. Estas funciones por defecto están deshabilitadas por motivos de seguridad.

Una vez hecho esto y al tener los servicios de la parte servidora de la plataforma independientes, únicamente faltaba tratar los servicios de autenticación. Al ser una aplicación no web, no se podían usar los servicios antiguos, por tanto, se usó la librería proporcionada por Google Plus y Cordova llamada: '@ionic-native/google-plus/ngx' para la autenticación desde la aplicación móvil [41].

Finalmente, se realizaron cambios para que se guardaran los datos de inicio en el almacenamiento del dispositivo móvil y que no se tuviera que volver a iniciar sesión o registrarse una vez hecho.

En las siguientes ilustraciones (ilustraciones 39, 40, 41) se muestran capturas de pantalla del funcionamiento de la aplicación en un dispositivo móvil Android:

Ilustración 39 - Captura de pantalla de la pantalla inicial de la aplicación en móvil

Ilustración 40 - Captura de pantalla de la pantalla de registro de la aplicación en móvil

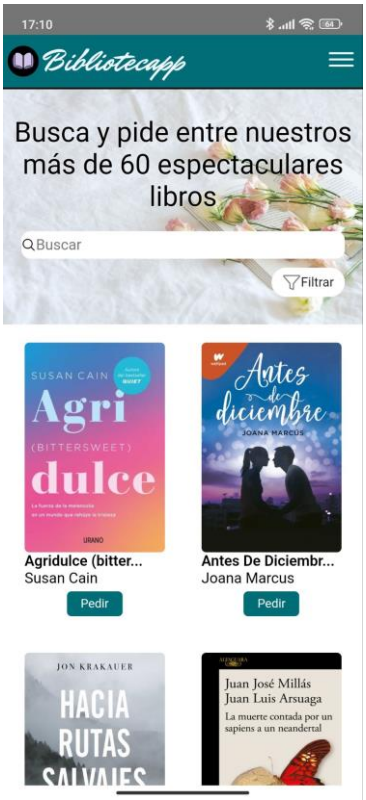

Ilustración 41 - Captura de pantalla de la pantalla principal de la aplicación en móvil

6.4 Seguridad

Un aspecto importante que merece mención en este proyecto de fin de grado es la seguridad de la aplicación. Durante el desarrollo se han intentado seguir e implementar las mejores prácticas posibles para salvaguardar la información de los usuarios y de la biblioteca y su plataforma.

6.4.1 Guards

En Angular, los guards son una característica que permite controlar el acceso a determinadas rutas dentro de una aplicación. Los guards se utilizan para proteger las rutas y decidir si un usuario puede acceder a ellas o no, según ciertas condiciones o reglas definidas.

Como ya se ha visto en este proyecto hay tres tipos de actores que tienen 3 niveles de permisos distintos. Siendo el administrador el único que puede acceder al 100% de las funcionalidades de la aplicación. Y siendo el usuario invitado el que menos acciones pudiera realizar.

En Bibliotecapp existen únicamente dos guards. El primero revisa si se tiene el permiso de usuario registrado. El segundo revisa si se tiene el permiso de administrador y el de usuario registrado, asimismo. Un ejemplo de la implementación de esta característica en la aplicación se muestra en la ilustración 42 relativa al primer guard:

```
export class GuardAuthGuard implements CanActivate {
  constructor(private authService: AutenticacionService, private router: Router, private storage:StorageAndroidService){};
  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
    if(isPlatform('mobileweb')){
      if (!this.authService.getLocalUser()) {
        alert('No estás logueado');
        this.router.navigate(['/']);
        return false;
      }
    }
    else if(isPlatform('android') || isPlatform('ios')){
      if (!this.storage.getUser()) {
        alert('No estás logueado');
        this.router.navigate(['/']);
        return false;
      }
    }else{
      // Web
      if (!this.authService.getLocalUser()) {
        alert('No estás logueado');
        this.router.navigate(['/']);
        return false;
      }
    }

    return true;
  }
```

Ilustración 42 - Ejemplo en código del guard de usuario registrado

En este guard se observa que se han tenido en cuenta los 4 tipos de dispositivos que pueden acceder: web, web desde el móvil o aplicación móvil tanto Android como iOS. En los cuatro casos, un usuario invitado tendrá vetado el acceso a la ruta que se quiera proteger. Por ejemplo, en la plataforma de Bibliotecapp un usuario invitado no puede

acceder a la pantalla de “Mis Reservas” puesto que no tiene ninguna reserva. Si este usuario invitado se las ingeniase para saltarse el proceso de registro y accediera a esta parte de la aplicación sin estar registrado le aparecería el mensaje de la ilustración 43:

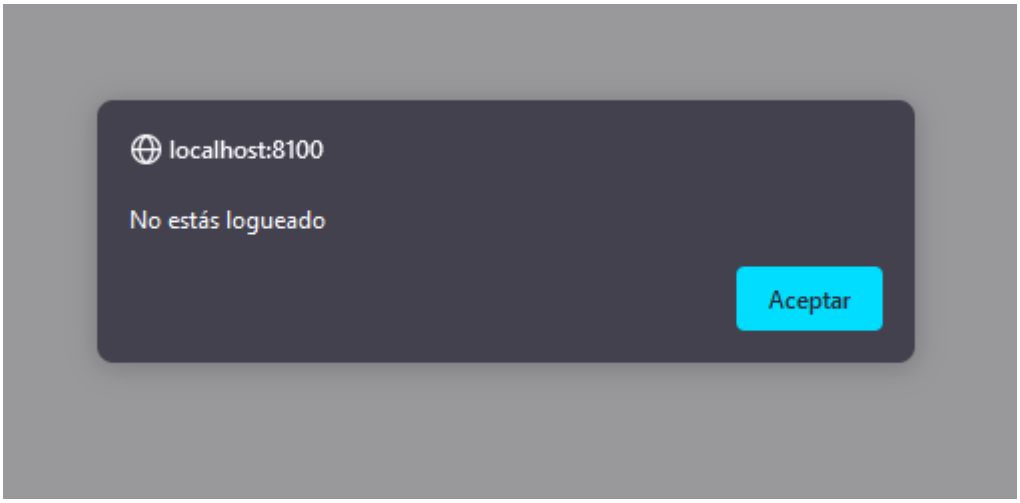

Ilustración 43 - Mensaje de error al acceder a una ruta sin los permisos requeridos

La página a la cual se quisiera acceder nunca se cargaría hasta que se obtuvieran los permisos necesarios. En este caso el permiso se conseguiría al iniciar sesión. Tras pulsar en el botón con el texto de “Aceptar” de la ilustración se volvería a la página de inicio de sesión para que el usuario se identificase y obtuviera el permiso para acceder a esta parte. Al igual que en este ejemplo, el segundo guard funciona de similar manera. Únicamente cambian los permisos necesarios para acceder a las rutas de administración de la plataforma que en este caso se obtendrían si el usuario actual está dado de alta en la base de datos de usuario como administrador.

Con esta simple medida de seguridad, se consigue proteger las rutas de la aplicación, asegurando que únicamente los usuarios autorizados puedan acceder a ellas.

6.4.2 Inyecciones SQL

La inyección SQL es una técnica de ataque utilizada en aplicaciones web para explotar vulnerabilidades en las capas de acceso a bases de datos SQL. Consiste en "inyectar" código SQL malicioso en las consultas enviadas a la base de datos a través de la aplicación. En la plataforma de Bibliotecapp se ha tenido en cuenta este tipo de ataques al crear la parte del servidor y por tanto se han implementado medidas para mitigar estos ataques. Cada petición HTTP efectuada a la API de Bibliotecapp ha sido creada siguiendo buenas prácticas para evitar estos ataques. Se han implementado técnicas de parametrización y saneamiento de los datos en las consultas a la base de datos, con el fin de impedir la incorrecta interpretación de los valores introducidos y con ello minimizar la exposición a este tipo de ataques.

```
// GET books by id
router.get("/books/:id", (req, res) => {
  const { id } = req.params;
  const isbnPattern = '^(?=(?:[^0-9]*[0-9]){10}(?:(?:[^0-9]*[0-9]){3})?$)[\\d-]+$';
  if(!id.match(isbnPattern)){
    res.status(400).json({err: "El formato del isbn no es válido"});
  }
  let sql = "select * from Book where isbn = ?";
  connexion.query(sql, [id], (err, rows) => {
    if (err) throw err;
    else {
      res.status(200).json(rows);
    }
  });
});
```

Ilustración 44 - Ejemplo de código para ilustrar la protección contra inyecciones SQL en las peticiones HTTP

Como se muestra en la ilustración 44, existen varios factores utilizados para evitar inyecciones SQL que puedan comprometer la integridad de la plataforma de Bibliotecapp. En primer lugar, se utiliza el signo de interrogación "?" al pasar un parámetro, como en el caso del ISBN de un libro. Esto ayuda a garantizar que, si se introduce un código incorrecto en lugar de un ISBN, la sentencia fallará y se lanzará un error. Por otro lado, el servidor está configurado por defecto para rechazar múltiples sentencias en una sola llamada, lo cual previene la introducción de código malicioso. Además, se ha implementado un patrón o "pattern" en cada llamada de la API para validar el parámetro requerido. Por ejemplo, en la ilustración anterior, se utiliza el patrón "isbnPattern" para aceptar exclusivamente ISBN de diez o trece dígitos, con o sin guiones, y que no contengan letras, espacios u otros caracteres especiales. Aunque este patrón también se utiliza en el validador de la interfaz al introducir el ISBN, se ha agregado una doble comprobación para mayor seguridad. Todas las consultas HTTP han sido creadas siguiendo este procedimiento.

6.4.3 Ataques XSS

Un ataque XSS (Cross-Site Scripting) es una vulnerabilidad de aplicaciones web que permite a un atacante inyectar y ejecutar código malicioso en el navegador web. El objetivo de este ataque es permitir la ejecución de scripts o código en el navegador sin el conocimiento de la aplicación web.

En la plataforma de Bibiotecapp, al igual que en el caso de prevenir las inyecciones SQL en los datos que interactúan directamente con la base de datos, se han implementado medidas para mitigar este tipo de ataques. Se han aplicado técnicas de saneamiento en las entradas de datos para evitar la ejecución de código malicioso en el navegador. En la ilustración 45 se mostrará un ejemplo de cómo los validadores de las entradas de datos o "inputs" evitan la introducción de caracteres que puedan alterar el correcto funcionamiento de la plataforma.

Ilustración 45 - Ejemplo para ilustrar la protección contra ataques XSS en la entrada de datos

6.4.4 Evolución de la seguridad

Aunque se hayan tomado medidas para mitigar los ataques malintencionados a la plataforma, cabe resaltar que ninguna medida de seguridad ofrece protección total y que siempre existe un riesgo de posibles ataques. Nuevas técnicas o variantes de ataques pueden surgir en cualquier momento, lo que resalta la importancia de mantener una actitud vigilante incluso después de finalizar el desarrollo de la plataforma.

6.5 **Rendimiento**

El rendimiento es la medición objetiva y la experiencia percibida por el usuario del tiempo de carga y el tiempo de ejecución. El rendimiento en las aplicaciones es de vital importancia para una mejor experiencia del usuario ya que un tiempo de carga lento puede frustrar al mismo. Asimismo, es importante tener un alto rendimiento puesto que esto reduce el uso de recursos y, por tanto, reduce los costes operativos.

En Bibliotecapp se ha intentado reducir este tiempo de carga de las siguientes formas:

6.5.1 Lazy Loading

En Angular por defecto, todos los componentes están agrupados en un módulo principal. Este módulo cuando se inicia la aplicación carga todos los componentes del mismo a la vez. Esto produce que el inicio de la aplicación se vea retardado conforme al número de componentes que haya en la aplicación. Para reducir esta carga inicial, en la aplicación se ha utilizado la técnica de Lazy Loading. Esta técnica utiliza un script para que los componentes de un módulo se carguen solo cuando sean requeridos. Esto produce una carga mucho más veloz al inicio y en la ejecución de la aplicación puesto que va cargando los módulos según se vaya requiriendo.

Para ejemplificar la mejoría en el rendimiento tras el uso de esta técnica se comparará el tiempo total de carga con y sin el uso de esta. A continuación, se mostrarán dos ilustraciones (ilustración 46 y 47) comparando sus tiempos:

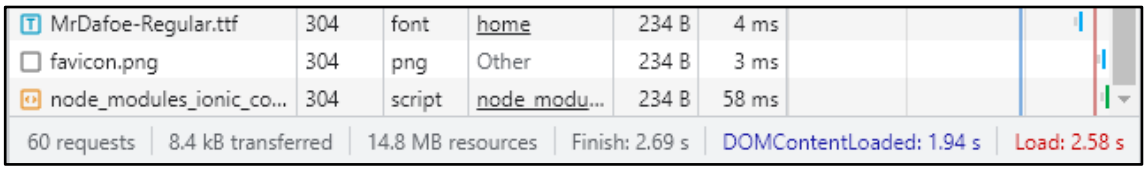

Ilustración 46 - Captura de pantalla del tiempo de carga de la aplicación sin Lazy Loading

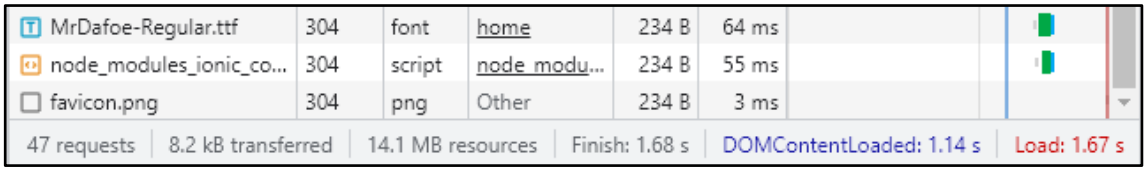

Ilustración 47 - Captura de pantalla del tiempo de carga de la aplicación con Lazy Loading

La primera ilustración anterior mostrada representa la carga de toda la aplicación sin el uso de ningún Lazy Loading. La segunda ilustración mostrada en cambio representa el tiempo real de carga de un usuario invitado sin registrar. Como se puede apreciar en la primera imagen se cargan más peticiones o "requests", el peso de los archivos descargados es mayor y se tarda hasta 1s más en cargar. Con esta comparativa se demuestra que esta técnica permite minimizar el tiempo de carga y con ello mejorar la experiencia del usuario.

Todos los usuarios tengan el permiso que tengan únicamente cargan la pantalla principal y la de fallo. Una vez estos usuarios han iniciado sesión o se han registrado, se van cargando progresivamente los demás módulos. No tiene sentido que un usuario invitado pueda cargar un componente como pudiera ser "Mis Reservas" cuando nunca tendrá acceso a él. De igual manera el módulo con los componentes de administración de la plataforma no serán cargados hasta que el usuario tenga permisos de administrador e intente acceder a este apartado.

6.5.2 Otras prácticas

Además de cargar únicamente los componentes que se necesitan se han implementado otras prácticas más simples para mejorar el rendimiento de la aplicación. A continuación, se enumerará un listado de buenas prácticas que se han seguido:

- Usar formatos ligeros para imágenes: se ha intentado usar formatos como webp y avif en vez de png. Para los iconos se ha usado el formato svg.
- Utilizar la etiqueta nativa de lazy en imágenes para cargar imágenes únicamente cuando se vayan a mostrar.
- Emplear CSS en vez de JavaScript siempre que se pueda.

6.6 Interfaz adaptativa

Bibliotecapp es una plataforma híbrida. Esto quiere decir que se ha desarrollado para que pueda ejecutarse en múltiples plataformas. Esto implica la necesidad de adaptarse a diferentes resoluciones y tamaños de pantalla. Asimismo, se han debido de adaptar funcionalidades para hacerlas más accesibles en dispositivos con pantallas más pequeñas. Para lograr esto, se ha utilizado un enfoque basado en el uso del módulo de CSS de media queries [42].

Este módulo de CSS permite adaptar la representación del contenido a características del dispositivo como la resolución de pantalla o incluso el uso del braille. Y es un principio básico de la tecnología de diseño web adaptativo.

La aplicación permite la adaptación de componentes o elementos a las distintas resoluciones en las que se puede presentar la misma. Durante el desarrollo se ha usado este módulo juntamente con las directivas de Angular *ngClass y *ngIf para cambiar el aspecto de los componentes, redimensionarlos, reorganizarlos o incluso cambiarlos por otros de manera automática y pudiéndose cambiar en tiempo de ejecución [43] [44].

En la ilustración 48 se mostrará un ejemplo en código en CSS del uso de una media query:

```
@media (min-width: 650px) {
    .titulo{
        padding-bottom: 10px;
        font-size: 25px;
    }
```

Ilustración 48 - Ejemplo en código del uso de una media query

Como se puede observar en la ilustración anterior, se ha usado una media query para aplicar estilos específicos a un elemento con la clase "título" cuando se cumple una condición relacionada con el ancho de la ventana mínimo del navegador.

Además, con media query se ha llevado a cabo una estrategia de globalización de componentes del CSS basada en el tamaño de los dispositivos. Se han utilizado tamaños de fuente, márgenes, tamaños de imagen y otros atributos visuales que se ajustan proporcionalmente al tamaño de la pantalla desde un archivo de estilos global a toda la aplicación. Permitiendo así evitar duplicación de código y permitir cambios más fáciles en estos aspectos visuales relacionados con el tamaño de la pantalla.

7. Pruebas

En el proceso de desarrollo de cualquier aplicación, es fundamental efectuar pruebas exhaustivas para garantizar su calidad, funcionalidad y usabilidad. Estas pruebas permiten identificar posibles fallos en la aplicación. En este apartado se detallarán en dos tipos de pruebas primordiales: las pruebas de software y las pruebas con usuarios.

7.1 Pruebas con usuarios

Las pruebas con usuarios se centran en evaluar la experiencia del usuario final al interactuar con la aplicación. Estas pruebas permiten obtener información valiosa sobre la usabilidad, la accesibilidad y la satisfacción del usuario con la aplicación, entre otros aspectos. Al involucrar a los usuarios en el proceso de prueba, se obtienen experiencias reales, una perspectiva más amplia, y se pueden identificar mejoras y ajustes necesarios para lograr una mejor experiencia de usuario. Además de que se pueden encontrar fallos no contemplados durante el desarrollo.

Para las pruebas con usuarios se desplegó la plataforma Bibliotecapp en el servidor de Web Avanzado de la Universidad Politécnica de Valencia. Asimismo, se generaron dos aplicaciones para que se pudiese acceder vía aplicación móvil. Conjuntamente con el acceso a la plataforma, se proporcionó una encuesta para que los usuarios pudiesen dar una retroalimentación del software que estaban probando. Esta encuesta está disponible en el Anexo 3.

A continuación, se presentará un breve informe de los resultados de la encuesta realizada. En total, se han recibido 13 respuestas, las cuales proporcionan información valiosa para comprender la experiencia de los usuarios y mejorar el servicio. Este informe, se centrará en resaltar los resultados más relevantes obtenidos a partir de las respuestas de los usuarios. Si bien todas las respuestas son valiosas, se pondrá el foco en los aspectos que presentan un impacto significativo en la satisfacción general de los usuarios. Esto con el fin de identificar las áreas clave que requieren atención y/o mejoras, para tomar medidas al respecto y así, garantizar una mejor experiencia para todos usuarios.

7.1.1 Análisis de la encuesta

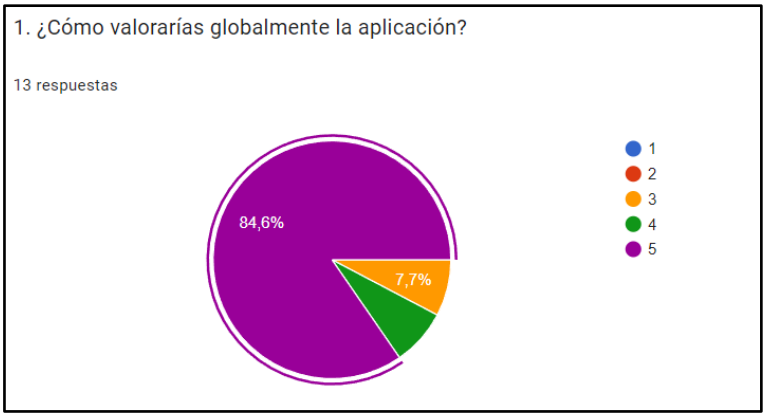

Ilustración 49 - Resultados de pregunta 1 de la encuesta

(1 = Muy negativamente, 5 = Muy positivamente)

En la primera pregunta, que evaluaba globalmente la aplicación, se observa que el 84,6% de los encuestados otorgó la máxima puntuación. Este resultado refleja una alta satisfacción general de los usuarios con la plataforma, lo cual es muy positivo y sugiere que la mayoría de ellos están contentos con la aplicación. Este alto nivel de valoración es un indicador alentador de la calidad y funcionalidad de la plataforma.

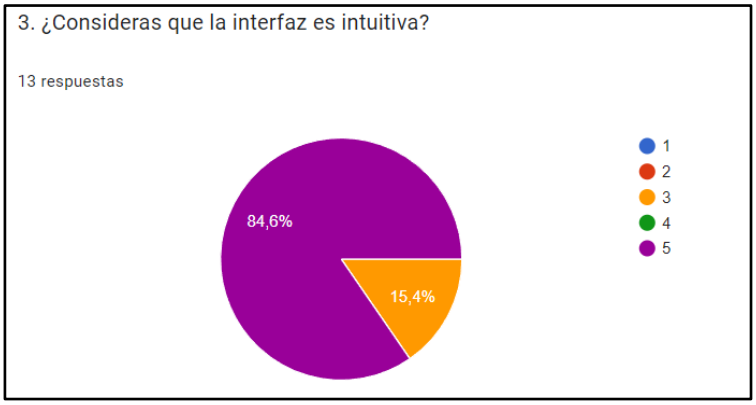

Ilustración 50 - Resultados de pregunta 3 de la encuesta

En la tercera pregunta, que cuestionaba la intuitividad de la interfaz de la aplicación, se destaca que el 84,6% de los participantes otorgó la puntuación máxima. Pese a esto se muestra que el restante 15,4% se mostraron indiferentes con la intuitividad. Este resultado indica que la gran mayoría de los usuarios percibe la interfaz como intuitiva, lo cual es altamente positivo para la usabilidad y la experiencia general de la plataforma. Esto es un factor clave para retener y atraer a más usuarios en el futuro, además de que se mejora la satisfacción del usuario.

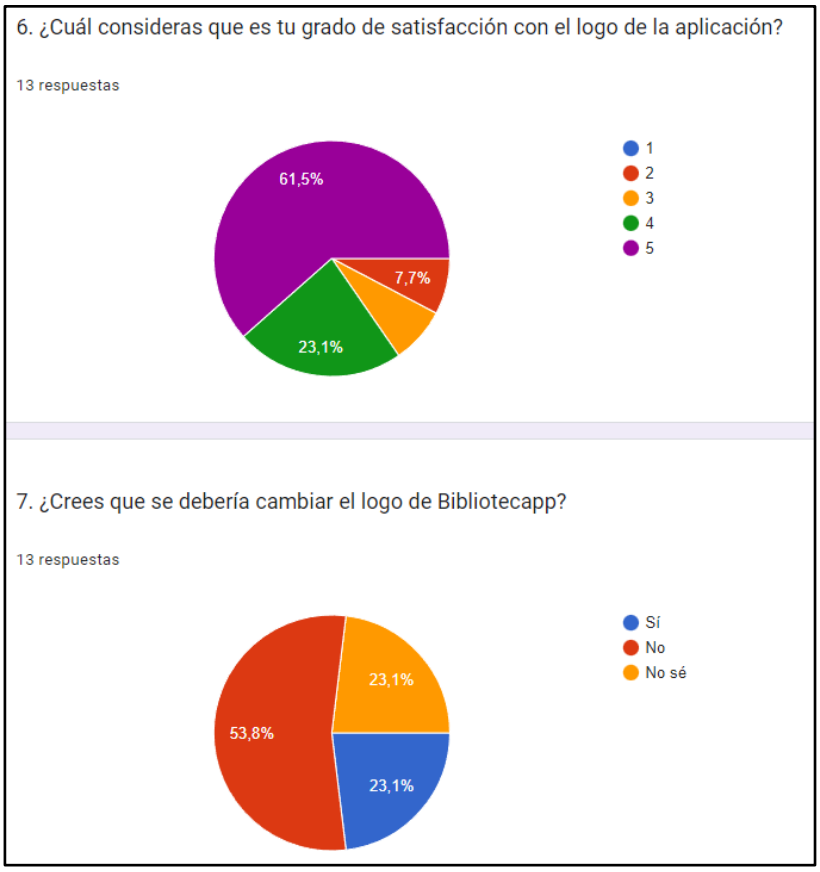

Ilustración 51 - Resultados de las preguntas 6 y 7 de la encuesta

A continuación, como se puede observar en la ilustración 51, se analizará el logo de la aplicación. Para ello en el formulario se les mostró la imagen del logo para evitar confusiones. Como se puede ver los usuarios tienen opiniones distintas sobre el mismo. En la pregunta 6 se puede observar que un 15,4% de los encuestados ha puntuado por debajo del 4 al logo. Asimismo, en la siguiente pregunta, un 23,1% de los encuestados ha afirmado que cambiaría el logo de Biblioteca. Pese a no ser valores muy grandes se debe tener en cuenta para plantear un posible rediseño del mismo en el futuro.

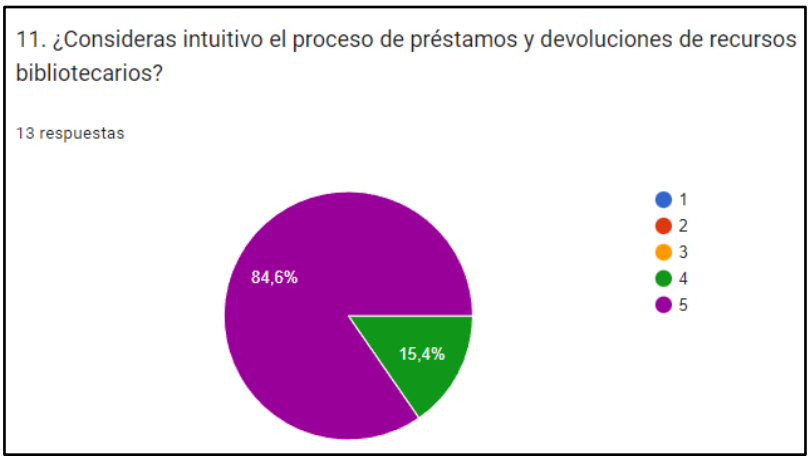

Ilustración 52 - Resultados de la pregunta 11 de la encuesta

En la undécima pregunta, que evaluaba si los encuestados consideraban intuitivo el proceso de préstamos y devoluciones de recursos bibliotecarios con Bibliotecapp, se puede apreciar que 11 de 13 participantes consideran muy positiva su experiencia con esta función. Esto indica que la mayoría de los usuarios perciben este proceso como fácil e intuitivo. Este alto porcentaje es realmente bueno puesto que esta es la función principal de la plataforma o aplicación y es imprescindible para el buen funcionamiento de la biblioteca.

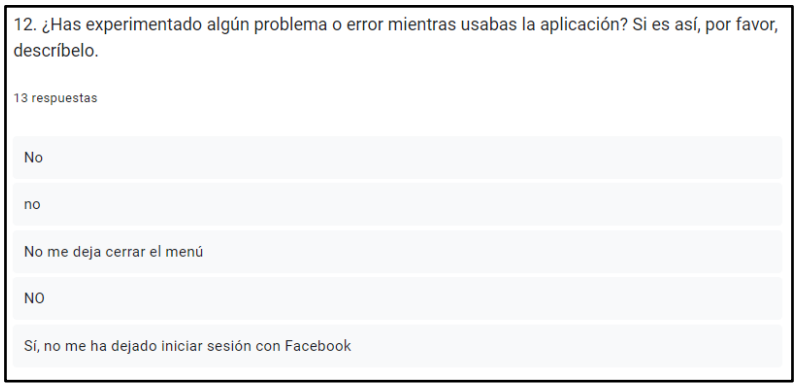

Ilustración 53 - Resultados de la pregunta 12 de la encuesta

Finalmente, se ha puesto el foco sobre la duodécima pregunta. Como se puede observar en las respuestas, entre los usuarios se han detectado dos fallos. Tras recoger los resultados se hicieron pruebas y se determinó que eran ciertas. Como se puede ver en la ilustración previa, un usuario de Android no podía cerrar el menú lateral. Tras varias pruebas se determinó que tras abrirlo varias veces sin seleccionar una opción, este se bloqueaba, no permitiendo realizar ninguna otra acción. Por otra parte, el fallo en la autenticación se pudo demostrar que sucedía puesto que se habían caducado los certificados de Facebook. Gracias a la retroalimentación que proporcionaron los usuarios respondiendo a esta encuesta, se pudieron arreglar estos fallos que de otro modo no se hubiesen descubierto. Mejorando así la usabilidad de la aplicación.

Asimismo, se ha comprobado que se han usado todas las plataformas posibles para probar la aplicación.

Por último, pese a haber más resultados interesantes no se han decidido incluir en el informe puesto que no eran tan importantes.

7.2 Pruebas de software

"*Las pruebas de un programa pueden ser usadas para demostrar la presencia de fallos, pero nunca para demostrar su ausencia*" - Edsger Wybe Dijkstra

Las pruebas de una aplicación software son imprescindibles en el desarrollo, ya que desempeñan un papel crucial en la detección de errores y problemas antes de que el software se implemente en un entorno en producción. Pese a esto no son suficientes por sí mismas para garantizar la calidad. Otras prácticas, como la revisión de código o el uso

de estándares de desarrollo sólidos, también son necesarias para minimizar la presencia de errores.

Las pruebas de software son fundamentales para evaluar y verificar que una aplicación cumpla con sus funcionalidades previstas. Las pruebas ofrecen una serie de notables beneficios, como la detección temprana y prevención de errores, la optimización de los costos de desarrollo y la mejora del rendimiento del software.

Hay distintos tipos de pruebas de software, cada una de estas persigue objetivos específicos. En este proyecto se han usado pruebas unitarias y pruebas de integración.

7.2.1 Pruebas unitarias

Las pruebas unitarias o test unitario son formas de verificar el comportamiento o la funcionalidad de unidades individuales de código de forma aislada. Al probar cada unidad de forma aislada se verifica que funcione de forma correcta sin depender de otros. Normalmente tras las pruebas unitarias se suele hacer pruebas de integración para verificar el funcionamiento de la aplicación al completo.

En Angular, por defecto, los test unitarios de componentes son autogenerados y se utiliza la tecnología Jasmine [45]. Cada componente y servicio se crea con un archivo de testing con el formato “. spec.ts”. Dento de este archivo se pueden crear nuevas pruebas a las ya existentes. De forma automática estos archivos se crean con funciones para probar que se inicializan bien los componentes. Las pruebas con Jasmine se definen siguiendo una estructura de resultado esperado y resultado real. Para probar cualquier función se debe implementar una de prueba donde se describen los pasos a seguir para obtener el resultado esperado y una vez se tenga, que se compare con el resultado real. Una vez creadas todas las pruebas, si se ejecutase el comando "ng test", se escanearán todos los archivos de test existentes en la aplicación y se mostrarían los resultados en una ventana que se crearía.

Tras hacer la ejecución del comando estos son los resultados:

Ilustración 54 - Resultados de pruebas unitarias con Jasmine

Como se puede ver en la ilustración 54, se han ejecutado 34 test, de los cuales 0 han fallado. Por tanto, se podría asegurar que en todos los test de las funciones a las que se ha puesto el foco han dado un resultado correcto. Estos resultados son orientativos puesto que se puede introducir más test para demostrar mayor fiabilidad. Un buen indicador de la fiabilidad de estos test es la cobertura.

La cobertura de un conjunto de prueba es un parámetro que indica cuánto porcentaje del código se ha probado. En Bibliotecapp al hacer el comando anterior se genera automáticamente un informe de la cobertura de las pruebas. El resultado obtenido de cobertura en las pruebas realizados con Jasmine es del 84%. Normalmente se considera

que un código con un 80% de cobertura es aceptable, por tanto, se podría considerar que las pruebas cumplen su función.

7.2.2 Pruebas de integración

Las pruebas de integración o test e2e (end to end) es mecanismo de testeo de software que analiza si los elementos unitarios funcionan correctamente cuando se integran en grupo. Estas pruebas comprueban la correcta interacción de los componentes.

En Angular por defecto como en las pruebas unitarias existe el comando “ng e2e” para generar automáticamente pruebas de integración. Pese a esto, se ha decidido hacer las pruebas de integración con la tecnología Cypress [46]. Esto debido a que Cypress ofrece una interfaz de usuario intuitiva lo que facilita la escritura y ejecución de pruebas. Asimismo, esta herramienta ejecuta las pruebas en el mismo hilo que el navegador, lo que le permite realizar cambios en tiempo real. Esto hace que las pruebas sean más rápidas y permite una depuración más fácil al ver las acciones en tiempo real.

Cypress es una herramienta basada en Javascript para ejecutar pruebas end-to-end. Permite crear pruebas que se ejecuten automáticamente en la web devolviendo un análisis de las pruebas ejecutadas. En la ilustración 55 se muestra un fragmento de código escrito de una prueba de integración escrita en Cypress:

```
describe('prueba-busqueda-libros', () => {
  it('passes', () => {
    cy.visit('http://localhost:8100/home');
    cy.wait(400);
    cy.get('#input')
      .click()
      .type("Agridulce")
      .type('{enter}');
  })
})
```

Ilustración 55 - Resultados de pruebas unitarias con Jasmine

Como se puede ver en ilustración previa, este código entra a la página principal y hace una búsqueda en el buscador de Bibliotecapp para comprobar que funciona correctamente. Tras realizar este código, la herramienta mostrará un informe de este y de todos los resultados de las pruebas que se hayan realizado.

Durante el desarrollo del proyecto se han creado pruebas como estas para comprobar el correcto funcionamiento de todas las funcionalidades de la aplicación cuando se añade una nueva funcionalidad al resto ya existente.

8. Mantenimiento y gestión de versiones

8.1 Gestión de versiones

Como previamente se ha mencionado, la herramienta usada para el control de versiones escogida ha sido GitHub. Esta herramienta permite llevar un registro de los cambios que se realizan en el proyecto. Dentro de Github existen "ramas" que son conjuntos de registros organizados bajo un mismo nombre. Estas ramas se pueden crear, eliminar y, sobre todo, se pueden fusionar. A la hora de realizar un proyecto de alta envergadura como Bibliotecapp a veces aparecen fallos o tareas más prioritarias que otras y por esto se debe cambiar de rama. Para tener un control de todas las ramas creadas, su contenido y su funcionalidad se ha decidido implementar el sistema de ramas o "branching" de GitFlow.

Gitflow se basa en la distinción de ramas. Normalmente se suelen crear dos ramas principales, una rama de desarrollo llamada "develop" y otra rama principal llamada "master" o "main". En la primera de estas ramas, se suelen añadir las funcionalidades nuevas de la aplicación. Es altamente recomendable añadir estas funcionalidades sin que afecten a la lógica de la aplicación. Una vez se suman suficientes funcionalidades y se comprueba que funcionan todas ellas, se suele fusionar con una nueva rama llamada por el nombre de "release" juntamente con un número de versión, normalmente superior a la que había. Una vez se prueba esta nueva rama funciona como se espera, se sube a la rama "master". Esta última rama debería contener siempre una versión estable de la aplicación puesto que tras la fusión con la rama "release" se suele subir a producción. Asimismo, existen ramas de funcionalidad llamadas "feature" donde se va añadiendo nuevo código con el fin de terminar esta funcionalidad y poder fusionarla con la rama "develop". Finalmente, la rama "hotfix" o rama corrección de errores, se utiliza para solventar errores críticos en la versión actual de la aplicación en producción. Se crean a partir de la rama "master", se soluciona el fallo y se vuelve a fusionar con esa rama sin pasar por "develop".

En la ilustración 56 se muestra una representación gráfica simplificada de este método de flujo de trabajo:

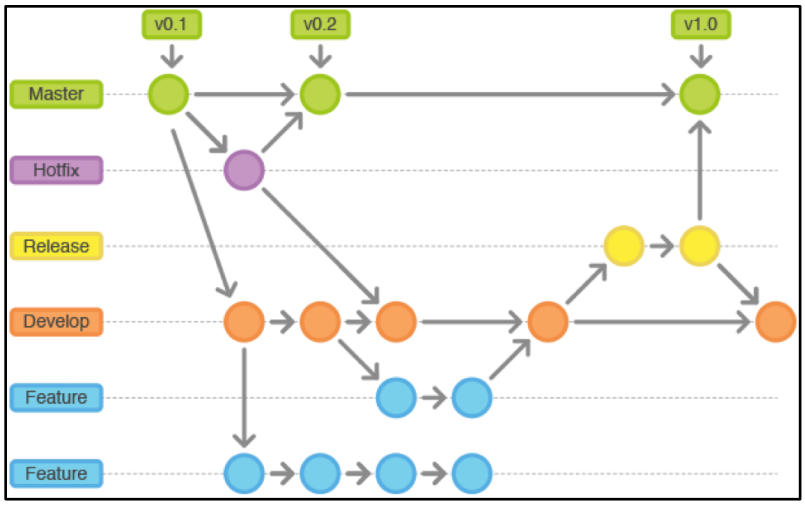

Ilustración 56 - Representación de Git Flow

8.2 Mantenimiento

El mantenimiento de una plataforma digital como Bibliotecapp es fundamental. Como se ha visto en anteriores puntos, pese a que la aplicación cuente con protecciones contra acciones malintencionadas, estas evolucionan constantemente. Es imprescindible mantener el foco en la seguridad de la aplicación para proteger su integridad y los datos de los usuarios. Asimismo, se debe seguir un monitoreo constante para regular posibles vulnerabilidades en el sistema. De igual forma, es importante tener un control constante de la aplicación para presentar siempre un funcionamiento óptimo. La revisión de la infraestructura tecnológica, la detección y solución de errores, y la eliminación de posibles fallos que puedan afectar la experiencia del usuario son esenciales para mantener este nivel óptimo. Además, deberán existir operarios que trabajen con la aplicación actualizando los eventos, recursos, comentarios, clientes y reservas, puesto que estos variarán con el transcurso del uso de la aplicación. Periódicamente, se deberán actualizar las licencias de los servicios de autenticación y los demás servicios externos puesto que tienen fecha de expiración. Finalmente, pueden surgir mejoras, o nuevas funcionalidades necesarias. Al realizar un mantenimiento sobre estos aspectos, se garantiza que los usuarios tengan una experiencia gratificante lo que mejorará su experiencia y los mantendrá comprometidos con la plataforma.

9. Conclusiones

En este trabajo de fin de grado, se han establecido diversos objetivos con el fin de mejorar la accesibilidad y optimizar los servicios de las bibliotecas a través de la creación de una biblioteca virtual. Al finalizar el desarrollo y análisis de este proyecto, se pueden extraer las siguientes conclusiones:

Se ha logrado mejorar la accesibilidad a los servicios bibliotecarios. Gracias a la creación de esta aplicación se ha facilitado el acceso a los recursos y servicios de la biblioteca física. Los usuarios ahora pueden solicitar, reservar y verificar la disponibilidad de libros y otros recursos de manera más eficiente y cómoda. Asimismo, se ha añadido durante el desarrollo la opción de pedir el traslado de libros desde otras bibliotecas. Además, se ha favorecido la comodidad de los usuarios al presentarse en formato híbrido, haciendo que la aplicación se adapte a las necesidades o preferencias de cada individuo. De igual forma, se ha cumplido el objetivo de presentar la plataforma en varios idiomas.

Se considera que este proyecto ha respondido a la necesidad de informatizar los servicios bibliotecarios. La plataforma implementada ha permitido la automatización de procesos no presenciales, lo que contribuirá a minimizar las interacciones sociales. Asimismo, ha incrementado la disponibilidad de estos servicios a las veinticuatro horas del día, sin tener que ajustarse al horario predefinido de la biblioteca física.

Se ha favorecido la comodidad y adaptabilidad para los usuarios: La aplicación desarrollada es multiplataforma, lo que permite a los usuarios acceder a los servicios de la biblioteca virtual desde sus dispositivos preferidos. Esto ha aumentado la comodidad y la satisfacción de los usuarios al adaptarse a sus necesidades y preferencias individuales.

Tras la exposición de las siguientes conclusiones se considera que se han cumplido todos los objetivos propuestos.

10. Trabajos futuros

Pese a haberse realizado un trabajo completo de una aplicación totalmente funcional, siempre se pueden añadir nuevas mejorar o cambios con el fin de mejorar la experiencia del usuario.

Primeramente, cabe destacar que los libros presentados al presentar la aplicación final no son los libros reales de ninguna biblioteca, por eso se deberían introducir a mano los recursos disponibles en la biblioteca final con la ayuda de las funcionalidades de Bibliotecapp. Igualmente se deberían modificar los eventos disponibles.

Otro punto donde puede mejorar la aplicación sería la posibilidad de abrir documentos como libros digitales o e-books para poder leerlos desde la misma plataforma.

Como se ha visto en la tabla “Tabla 1 - Comparación de características contra aplicaciones rivales”, muchas aplicaciones competidoras usan herramientas para la recomendación de libros. Esto podría ser interesante puesto que permitiría ofrecer sugerencias de libros que se ajusten a los gustos de cada persona, lo que mejoraría la experiencia de búsqueda y descubrimiento de nuevos títulos.

Además, se podría incorporar una función de reseñas y valoraciones de libros por parte de los usuarios. Esto permitiría a los usuarios compartir sus opiniones y experiencias sobre los libros que han leído, brindando a otros usuarios información del recurso al momento de elegir qué libro leer.

Otro aspecto donde podría mejorar la aplicación sería el cambio de los servicios de autenticación de Google o Facebook. Para tener una mejor privacidad y control de los datos de los usuarios, se podrían buscar alternativas a estos servicios usados de autenticación.

Finalmente, como se ha podido observar, pese a en un principio estar diseñada la aplicación para soportar múltiples bibliotecas a la vez, no se han llegado a implementar del todo. Sería altamente beneficioso crear una red de bibliotecas que funcionasen con la aplicación, ya que Bibliotecapp cuenta con las capacidades necesarias para ello.

11. Referencias

El objetivo de este apartado es proporcionar al lector una lista de fuentes de información que se han utilizado en el trabajo y permitirle comprobar la veracidad de las afirmaciones y el conocimiento previo que se ha tenido en cuenta.

A continuación, se presenta la lista solicitada:

[1] Polibuscador UPV, En línea, Último acceso 29/06/2023 https://polibuscador.upv.es/discovery/search?vid=34UPV_INST:bibupv

[2] BNE, En línea, Último acceso 29/06/2023 https://www.bne.es/es

[3] Biblioteca Digital Gredos, En línea, Último acceso 29/06/2023 https://gredos.usal.es/handle/10366/1

[4] Casadellibtro.com, En línea, Último acceso 29/06/2023 https://www.casadellibro.com/

[5] Nextory, En línea, Último acceso 29/06/2023 https://www.nextory.es/

[6] Aplicación Android Nextory, En línea, Último acceso 29/06/2023 https://play.google.com/store/apps/details?id=com.gtl.nextory&hl=es_419&gl=US

[7] TFG de Encarnació Maria Durà Garcia, tutorizado por Félix Buendía García, En línea, Último acceso 29/06/2023 https://riunet.upv.es/handle/10251/8731

[8] Firebase, En línea, Último acceso 29/06/2023 https://console.firebase.google.com

[9] Firebase Authentication, En línea, Último acceso 29/06/2023 https://firebase.google.com/docs/auth?hl=es-419

[10] Patrón Modelo-Vista-Controlador de freecodecamp.org, En línea, Último acceso 29/06/2023 https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/

[11] Definición de GIF, En línea, Último acceso 29/06/2023 https://es.wikipedia.org/wiki/Graphics_Interchange_Format

[12] Paleta de Bibliotecapp ofrecida por coolors.co, En línea, Último acceso 29/06/2023 https://coolors.co/palette/006d77-83c5be-edf6f9-ffddd2-e29578

[13] Documentación de Angular, En línea, Último acceso 29/06/2023 https://angular.io/

[14] Documentación de Typescript, En línea, Último acceso 29/06/2023 https://www.typescriptlang.org/

[15] Documentación de HTML, En línea, Último acceso 29/06/2023 https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure

[16] Documentación de CSS, En línea, Último acceso 29/06/2023 https://developer.mozilla.org/es/docs/Web/CSS

[17] SCSS, En línea, Último acceso 29/06/2023 https://codepen.io/Lucasob/pen/QVmBwQ

[18] Documentación de Angular Material, En línea, Último acceso 29/06/2023 https://material.angular.io/

[19] Documentación de Ionic Framework, En línea, Último acceso 29/06/2023 https://ionicframework.com/

[20] Documentación de CapacitorJS, En línea, Último acceso 29/06/2023 https://capacitorjs.com/

[21] Documentación de NodeJs, En línea, Último acceso 29/06/2023 https://nodejs.org/es/docs

[22] Peticiones HTTP, En línea, Último acceso 29/06/2023 https://developer.mozilla.org/es/docs/Web/HTTP/Methods

[23] Documentación de ExpressJs, En línea, Último acceso 29/06/2023 https://expressjs.com/es/guide/routing.html

[24] Documentación de MySQL, En línea, Último acceso 29/06/2023 https://dev.mysql.com/doc/

[25] Documentación de Firebase Authentication, En línea, Último acceso 29/06/2023 https://firebase.google.com/docs/auth?hl=es-419

[26] Documentación de Firebase Cloud Storage, En línea, Último acceso 29/06/2023 https://firebase.google.com/docs/storage?hl=es-419

[27] Documentación de GitHub, En línea, Último acceso 29/06/2023 https://docs.github.com/es

[28] Documentación de Postman, En línea, Último acceso 29/06/2023 https://learning.postman.com/docs/introduction/overview/

[29] Visual Paradigm Online, En línea, Último acceso 29/06/2023 https://online.visual-paradigm.com/es/

[30] Canva, En línea, Último acceso 29/06/2023 https://www.canva.com/es_es/

[31] Documentación de Visual Studio Code, En línea, Último acceso 29/06/2023 https://code.visualstudio.com/docs

[32] Figma, En línea, Último acceso 29/06/2023 https://www.figma.com/

[33] Android Studio, En línea, Último acceso 29/06/2023 https://developer.android.com/docs?hl=es-419

[34] API de Bibliotecapp en Postman, En línea, Último acceso 29/06/2023 https://documenter.getpostman.com/view/23029514/2s93mAVLWA

[35] Documentación de RxJs, En línea, Último acceso 29/06/2023 https://rxjs.dev/guide/overview

[36] Documentación de NgModules de Angular, En línea, Último acceso 29/06/2023 https://angular.io/guide/styleguide#application-structure-and-ngmodules

[37] Documentación de AgularFire2, En línea, Último acceso https://firebaseopensource.com/projects/angular/angularfire2/

[38] ConvertCSV, En línea, Último acceso https://www.convertcsv.com/json-to-csv.htm

[39] Convert XSLX to SQL, En línea, Último acceso https://products.aspose.app/cells/conversion/xlsx-to-sql

[40] Documentación de JSON, En línea, Último acceso 29/06/2023 https://www.json.org/json-en.html

[41] https://ionicframework.com/docs/v5/native/google-plus

[42] Documentación de Media Queries en CSS, En línea, Último acceso 29/06/2023 https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries

[43] Documentación de NgClass en Angular, En línea, Último acceso 29/06/2023 https://angular.io/api/common/NgClass

[44] Documentación de NgIf en Angular, En línea, Último acceso 29/06/2023 https://angular.io/api/common/NgIf

[45] Documentación de Jasmine, En línea, Último acceso 29/06/2023 https://jasmine.github.io/

[46] Documentación de Cypress, En línea, Último acceso 29/06/2023 https://www.cypress.io/

ANEXO I

OBJETIVOS DE DESARROLLO SOSTENIBLE

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**

Bibliotecapp está relacionado con diversos objetivos de desarrollo sostenible. Destacando su impacto en áreas como la igualdad de acceso, la educación de calidad y salud y bienestar, entre otros.

Primeramente, el objetivo principal de la aplicación es promover la cultura y la lectura. Este objetivo está claramente relacionado con el ODS 4 de promover la educación de calidad. Esto es debido a que Bibliotecapp facilita el acceso a libros y otros recursos, además de que promociona eventos para promover la lectura y que más usuarios utilicen la biblioteca.

Asimismo, desde el primer momento del desarrollo se ha abogado por presentar el mayor grado de accesibilidad posible en la aplicación. Tanto los múltiples dispositivos desde los que se puede acceder a la aplicación hasta los varios idiomas en los que está presente la aplicación, conforman una gran accesibilidad para todos los usuarios. Esta característica de Bibiotecapp se alinea perfectamente con la ODS 10 (Reducción de las desigualdades), ya que promueven la igualdad de oportunidades y la reducción de las desigualdades al asegurar que todos los individuos puedan acceder a los recursos de la biblioteca de manera equitativa independientemente de sus condiciones físicas.

Finalmente, la biblioteca, como espacio público, desempeña un papel crucial en la promoción de la igualdad de género y la salud y el bienestar. A través de su colección de libros y otros recursos, la biblioteca puede ofrecer obras literarias y no literarias que aborden temas relacionados con la igualdad de género y/o la salud y el bienestar, tales como el empoderamiento de las mujeres, la lucha contra la discriminación, la promoción del bienestar mental, la nutrición, el cuidado personal, etc.

Bibliotecapp

Introduction

This is Bibliotecapp's API

Overview

The Bibliotecapp API enables developers to create, read, update, and delete various entities within a library, including books, borrowers, transactions, and more. This API empowers library administrators, librarians, and other authorized personnel to efficiently handle the day-to-day tasks associated with managing a library.

Authentication

Only allows HTTP Requests

Error Codes

No errors

Rate limit

No number of requests per user

GET books

books

Obtiene todos los recursos de la biblioteca y sus detalles

GET books/:isbn

books/:isbn

Obtiene un recurso de la plataforma filtrado por su isbn

PATH VARIABLES

isbn

GET books/order/asc

books/order/asc

Obtiene todos los recursos de la biblioteca y sus detalles ordenados por título de manera ascendente

GET books/order/desc

books/order/desc

Obtiene todos los recursos de la biblioteca y sus detalles ordenados por título de manera descendente

GET books/name/:name

books/name/:name

Obtiene un recurso de la plataforma filtrado por su título

PATH VARIABLES

name

GET books/category/:category

books/category/:category

Obtiene todos los recursos de la biblioteca y sus detalles ordenados según su categoría

PATH VARIABLES

category

GET books/category/:category/order/asc

books/category/:category/order/asc

Obtiene todos los recursos de la biblioteca y sus detalles ordenados según su categoría y en orden ascendente según su título

PATH VARIABLES

category

GET books/category/:category/order/desc

books/category/:category/order/desc

Obtiene todos los recursos de la biblioteca y sus detalles ordenados según su categoría y en orden descendente según su título

PATH VARIABLES

category

GET colareservas

colareservas

Obtiene todos las colas de reservas de todos los recursos de la plataforma con sus detalles

---

GET colareservas/:isbn

colareservas/:isbn

Obtiene la cola de reservas de todos un recurso de la plataforma con sus detalles según su isbn

PATH VARIABLES

isbn

GET disponibilidades

disponibilidades

Obtiene las disponibilidades de los recursos de la plataforma y sus detalles

GET disponibilidades/:id

disponibilidades/:id

Obtiene la disponibilidad de un recurso de la plataforma y sus detalles

PATH VARIABLES

id

GET errors

errors

Obtiene todos los comentarios de la plataforma y sus detalles

GET errors/:id

errors/:id

Obtiene todos un comentario de la plataformay sus detalles filtrado por su id

PATH VARIABLES

id

GET events

events

Obtiene todos los eventos de la biblioteca y sus detalles

GET events/:id

events/:id

Obtiene un evento de la plataforma filtrado por su id

PATH VARIABLES

id

GET reservas

reservas

Obtiene todas las reservas activas de recursos efectuadas por usuarios de la biblioteca y sus detalles

---

GET reservas/:id

reservas/:id

Obtiene una reserva activa de un recurso efectuada por un usuario de la biblioteca filtrando por la id de la reserva y sus detalles

PATH VARIABLES

id

GET users

users

Obtiene todos los usuarios de la plataforma con sus atributos

---

GET users/:id

users/:id

Obtiene un usuario de la plataforma filtrado por su id (canet de la biblioteca)

PATH VARIABLES

id

DELETE books/:isbn

books/:isbn

Borra un libro seleccionado según su isbn

PATH VARIABLES

isbn

DELETE colareservas/:isbn

colareservas/:isbn

Borra la cola de reservas de un recurso seleccionado según su isbn

PATH VARIABLES

isbn

DELETE disponibilidades/:id

disponibilidades/:id

Borra la disponibilidad de un recurso seleccionado según su id

PATH VARIABLES

id

DELETE errors/:id

errors/:id

Borra un comentario seleccionado según su id

PATH VARIABLES

id

DELETE events/:id

events/:id

Borra un evento seleccionado según su id

PATH VARIABLES

id

DELETE reservas/:isbn

reservas/:isbn

Borra la reserva de un recurso seleccionado según su isbn

PATH VARIABLES

isbn

DELETE user/:id

user/:id

Borra un usuario seleccionado según su id (carnet de biblioteca)

PATH VARIABLES

id

POST books/

books/

Añade un nuevo recurso al catálogo de la biblioteca

POST colareservas/

colareservas/

Añade una nueva cola reserva de un recurso seleccionado a la base de datos de la plataforma

POST disponibilidades/

disponibilidades/

Añade una nueva disponibilidad de un recurso a la base de datos de la plataforma

POST events/

events/

Añade un nuevo evento a la base de datos de la plataforma

POST error/

error/

Añade un nuevo comentario a la base de datos de la plataforma

POST reservas/

reservas/

Añade una nueva reserva de un recurso seleccionado a la base de datos de la plataforma

---

POST users/

users/

Añade un nuevo usuario a la base de datos de la plataforma

PUT books/:isbn

books/:isbn

Modifica un libro seleccionado según su isbn

PATH VARIABLES

isbn

PUT colareservas/:isbn

colareservas/:isbn

Modifica la cola de reservas de un recurso seleccionado según su isbn

PATH VARIABLES

isbn

PUT disponibilidades/:id

disponibilidades/:id

Modifica la disponibilidad de un recurso seleccionado según su id

PATH VARIABLES

id

PUT error/:id

error/:id

Modifica un comentario seleccionado según su id

PATH VARIABLES

id

PUT events/:id

events/:id

Modifica un evento seleccionado según su id

PATH VARIABLES

id

PUT reservas/:isbn

reservas/:isbn

Modifica la reserva de un recurso seleccionado según su isbn

PATH VARIABLES

isbn

PUT users/:id

users/:id

Modifica un usuario seleccionado según su id (carnet de biblioteca)

PATH VARIABLES

id

ANEXO III

ENCUESTA DE BIBLIOTECAPP

1. ¿Cómo valorarías globalmente la aplicación?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
2. ¿Cómo describirías tu experiencia usando la aplicación?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
3. ¿Consideras que la interfaz es intuitiva?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
4. ¿Crees que es fácil de navegar con la interfaz?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
5. ¿Cuál consideras que es tu grado de satisfacción con la interfaz de la aplicación?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
6. ¿Cuál consideras que es tu grado de satisfacción con el logo de la aplicación?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○ 

7. ¿Crees que se debería cambiar el logo de Bibliotecapp?
   Sí. ○ No. ○ No sé. ○
8. ¿Cuál es tu nivel de satisfacción con el rendimiento y la velocidad de la aplicación?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
9. ¿Cuál es tu nivel de satisfacción con la forma de autenticación de la aplicación (Google o Facebook)?
   1. ○ 2. ○ 3. ○ 4. ○ 5. ○
10. ¿Piensas que Bibliotecapp facilita el proceso de préstamos y devoluciones de recursos bibliotecarios?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
11. ¿Consideras intuitivo el proceso de préstamos y devoluciones de recursos bibliotecarios?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
12. ¿Has experimentado algún problema o error mientras usabas la aplicación? Si es así, por favor, descríbelo.

13. ¿Has tenido alguna dificultad para realizar un préstamo o reserva de libros o recursos en la plataforma? Si es así, por favor, descríbela.

14. ¿Consideras sencilla e intuitiva la búsqueda de recursos en la plataforma?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
15. ¿Consideras completo e intuitivo el filtrado de recursos en la plataforma?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○

16. ¿Bibliotecapp te ha proporcionado información suficiente sobre los libros o recursos como autor, categoría, como isbn, etc.?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
17. (Administrador) ¿Te parece intuitivo el proceso de dar de alta un recurso o un evento de la biblioteca?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
18. (Administrador) ¿Consideras intuitivo el proceso de eliminar un recurso o un evento de la biblioteca?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
19. (Administrador) ¿Has tenido alguna dificultad para realizar una operación de eliminado o alta de algún recurso o evento de la plataforma? Si es así, por favor, descríbelo.

20. (Administrador) ¿Piensas que es intuitivo el proceso de visualización y eliminado de los comentarios de la plataforma Bibliotecapp?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
21. (Administrador) ¿Consideras intuitivo el proceso de gestión de los préstamos, devoluciones y reservas de la plataforma Bibliotecapp?
    1. ○ 2. ○ 3. ○ 4. ○ 5. ○
22. (Administrador) ¿Has tenido alguna dificultad para realizar una operación de gestión de reservas de la plataforma? Si es así, por favor, descríbela.

23. (Administrador) ¿Has tenido alguna dificultad para realizar alguna otra operación de administración diferente a las antes preguntadascomo gestión de clientes? Si es así, por favor, descríbela.

24. ¿Hay alguna característica o funcionalidad que te gustaría que se agregara a la aplicación?

25. ¿Recomendarías esta aplicación a otras personas? ¿Por qué sí o por qué no?

26. ¿En qué dispositivo(s) has probado la aplicación? (Ejemplo: teléfono, tablet, ordenador).

27. ¿Tienes alguna sugerencia adicional o comentario que te gustaría compartir?