



UNIVERSIDAD DE MÁLAGA



Grado en Ingeniería Informática

Uniendo pasiones literarias: plataforma de lectura con funcionalidades de lector digital y red social para amantes de la lectura

Combining literary passions: reading platform with E-reader features and social network for book lovers

Realizado por
Lucía Cárdenas Palacios

Tutorizado por
José Francisco Chicano García

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, septiembre de 2024

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

**Uniando pasiones literarias: plataforma de lectura con
funcionalidades de lector digital y red social para amantes
de la lectura**

**Combining literary passions: reading platform with E-
reader features and social network for book lovers**

Realizado por
Lucía Cárdenas Palacios

Tutorizado por
José Francisco Chicano García

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2024

Fecha defensa: septiembre de 2024

Resumen

La experiencia de la lectura digital ha evolucionado significativamente en los últimos años, permitiendo a los usuarios disfrutar de libros electrónicos, interactuar con otros lectores, y calificar sus lecturas a través de una gran variedad de aplicaciones. Sin embargo, suele ser necesario el uso de distintas herramientas para garantizar una experiencia más completa.

Para superar estas limitaciones, se han introducido diversas mejoras en la aplicación Calibre-Web, ampliando sus capacidades. Además de las opciones originales, como la lectura de libros electrónicos y la posibilidad de calificarlos, los usuarios ahora pueden participar en debates y discusiones literarias a través de un foro. También se ha desarrollado un sistema de recomendaciones personalizadas, basado en un *test* que analiza las preferencias del usuario.

Dado el creciente interés por las aplicaciones de podcasts y audiolibros, se ha añadido la funcionalidad de lectura en formato de audio, mejorando además la accesibilidad de la plataforma para un público más amplio. Por último, se ha adaptado la web a una red social literaria, permitiendo que los usuarios puedan seguirse mutuamente, comunicarse, y explorar el perfil de otros usuarios, fomentando así una mayor interacción en la comunidad.

Para el desarrollo de estas nuevas características, se ha adoptado una metodología ágil SCRUM, que permite un enfoque incremental y adaptativo. El trabajo se ha organizado en *sprints* de dos semanas, permitiendo la planificación, desarrollo y validación de cada funcionalidad.

Palabras clave: lectura digital, foro, recomendaciones personalizadas, audiolibros, red social.

Abstract

Digital reading experiences have evolved significantly in recent years, allowing users to enjoy e-books, interact with other readers, and rate their readings through a wide variety of applications. However, different tools are often needed to ensure a more complete experience.

To address these limitations, various improvements have been introduced to the Calibre-Web application, expanding its capabilities. In addition to the original features, such as reading e-books and rating them, users can now participate in literary debates and discussions through a forum. A personalized recommendation system has also been developed, based on a test that analyzes the user's preferences.

Given the growing interest in podcasts and audiobooks, an audio reading feature has been added, enhancing the platform's accessibility for a wider audience. Finally, the website has been adapted into a literary social network, allowing users to follow each other, communicate, and explore other users' profiles, thereby fostering greater interaction within the community.

For the development of these new features, an agile SCRUM methodology has been adopted, which allows for an incremental and adaptive approach. The work has been organized into two-week *sprints*, enabling the planning, development, and validation of each functionality.

Keywords: digital reading, forum, personalized recommendations, audiobooks, social network.

Índice

Resumen	3
Abstract	4
Índice	5
1. Introducción	7
1.1 Motivación	7
1.2 Objetivos	8
1.3 Estructura de la memoria	10
1.4 Metodología de trabajo empleada	12
1.4.1 Estructura de los Sprints	12
1.4.2 Descripción de los Sprints	13
2. Análisis de Requisitos	15
2.1 Requisitos funcionales	15
2.2 Requisitos no funcionales	19
2.3 Casos de uso	20
3. Estudio de tecnologías utilizadas	37
3.1 Lenguajes	37
3.2 Frameworks y Librerías.....	38
3.3 Entornos de Desarrollo	39
3.4 Herramientas de Gestión de Proyectos.....	39
3.5 Herramientas de Diseño	40
3.6 Herramientas de Bases de Datos y Modelado.....	40
3.7 Herramientas de Pruebas	40
4. Diseño del Software	43
4.1 Diagrama de Clase	43
4.2 Diagrama de Entidad Relación.....	46
5. Implementación	49
5.1 Recomendador	50
5.2 Foro.....	55
5.3 Red social	59
5.4 Lectura en formato audio	64
6. Pruebas	71
6.1 Configuración para la Realización de Pruebas.....	71
6.2 Pruebas Realizadas	72
7. Conclusiones y Líneas Futuras	81
7.1 Conclusiones	81
7.2 Objetivos Cumplidos y Logros obtenidos	82
7.3 Problemas Encontrados.....	83

7.4 Líneas futuras.....	84
7.4.1 Lectura de Audio Personalizada	84
7.4.2 Base de Datos de Preferencias para las Recomendaciones	85
7.4.3 Notificaciones mejoradas.....	86
7.4.4 Perfiles Públicos y Privados.....	86
7.4.5 Cambio de Idioma en la Lectura de Voz.....	86
7.4.6 Acceso de Mensajería	87
Referencias	89
Manual de Despliegue	93
Requerimientos generales.....	93
Clonación e instalación de dependencias.....	94
Instalación de Paquetes para Pruebas.....	95
Manual de Usuario	97
Configuración Inicial	97
Configuración de la Biblioteca	97
Configuración de la Cuenta.....	97
Configuración del Correo Electrónico.....	98
Detalles de los Parámetros de Correo	99
Avisos	100

1

Introducción

1.1 Motivación

A pesar de la existencia de numerosas aplicaciones relacionadas con la lectura, ya sea para la valoración de libros, la interacción entre usuarios o leer en formato digital, ninguna combina todas estas funcionalidades. Esto obliga a los lectores a usar múltiples herramientas para tener una experiencia completa.

Mientras que algunas, como Calibre, destacan por su capacidad para gestionar bibliotecas digitales y ofrecer un lector integrado, otras, como Goodreads, se enfocan en la interacción social entre lectores.

Esta situación inspiró la creación de una aplicación que integrase todas estas funcionalidades en una única plataforma. Basada en la aplicación web de código abierto de Calibre, que ya proporciona un lector digital y opciones básicas como la calificación de libros, esta versión incorporaría características adicionales, como un sistema de recomendaciones de libros personalizado y una red social que permita a los usuarios interactuar y participar en discusiones sobre libros.

1.2 Objetivos

El objetivo principal de este proyecto consiste en transformar la aplicación web de Calibre existente, añadiendo una serie de funcionalidades innovadoras para convertirla en una plataforma que satisfaga las necesidades de los lectores.

Con estas mejoras, se busca ofrecer una experiencia de lectura más completa y variada que combine la gestión de bibliotecas, la interacción social y la lectura en distintos formatos en una única plataforma. Los objetivos específicos que se pretenden alcanzar con esta modificación incluyen:

1. Integrar un foro sobre libros

La implementación de un foro permitirá a los usuarios participar en debates y discusiones sobre una amplia gama de temas relacionados con libros y lectura. Este foro servirá como un espacio para compartir opiniones, recomendaciones y críticas, fomentando una comunidad activa de lectores. Los usuarios podrán crear hilos de discusión sobre libros específicos, géneros, autores y tendencias literarias e interactuar con otros miembros mediante comentarios y respuestas. Este componente social no solo enriquecerá la experiencia del usuario, sino que también promoverá una mayor interacción entre los miembros de la plataforma.

2. Incluir una red social

La integración de una red social dentro de la aplicación permitirá a los usuarios conectarse y comunicarse entre sí de manera más directa y personal. Los usuarios podrán seguirse mutuamente, intercambiar mensajes privados, y explorar los perfiles de otros lectores para ver sus actividades recientes y listas de libros leídos. Esta funcionalidad proporcionará un entorno en el que los lectores puedan construir y mantener conexiones basadas en intereses literarios comunes, compartir sus experiencias de lectura y fomentar una comunidad de apoyo.

3. Posibilidad de escuchar los libros en formato audiolibro

Dado el creciente interés en los audiolibros y la accesibilidad que estos ofrecen, se añadirá la opción de escuchar los libros en formato de audio. Esta funcionalidad permitirá a los usuarios disfrutar de una experiencia de lectura alternativa, especialmente útil para aquellos que prefieren escuchar libros mientras realizan otras actividades o tienen dificultades para leer texto impreso.

4. Añadir un recomendador de libros personalizado

Para mejorar la experiencia de descubrimiento de libros, se desarrollará un sistema de recomendación personalizado que sugiera títulos basados en los gustos individuales de cada usuario. Los usuarios completarán un *test* de preferencias literarias que evaluará sus intereses y hábitos de lectura. El sistema utilizará esta información para generar recomendaciones de libros que se alineen con las preferencias del usuario, facilitando la búsqueda de nuevas lecturas que sean de su agrado. Este componente no solo ayudará a los usuarios a encontrar libros que les interesen, sino que también potenciará la satisfacción y el compromiso con la plataforma al ofrecer recomendaciones más relevantes y ajustadas a sus gustos.

En resumen, la modificación de la aplicación web de Calibre tiene como objetivo crear una plataforma de lectura digital más completa y atractiva, integrando funcionalidades que faciliten la interacción social, la personalización de recomendaciones, y la accesibilidad a través de audiolibros. Estos desarrollos no solo mejorarán la experiencia del usuario, sino que también contribuirán a establecer una comunidad lectora más conectada y participativa.

1.3 Estructura de la memoria

A continuación, se procede a explicar de forma breve el contenido que tiene cada capítulo, facilitando la lectura y comprensión del documento.

Capítulo 1: Introducción

En este capítulo se introduce el proyecto y se expone la motivación detrás de la creación de la plataforma de lectura digital, así como los objetivos a alcanzar, teniendo como objetivo principal ofrecer una experiencia de lectura más completa y social en una única plataforma. Además, se detallan las metas específicas, como la integración de un foro de discusión, una red social, y un recomendador de libros, los cuales están diseñados para enriquecer la interacción entre los usuarios y mejorar la accesibilidad.

Por último, se describe la metodología de trabajo utilizada durante todo el proyecto, explicando brevemente su función y el por qué ha sido considerada la mejor opción.

Capítulo 2: Análisis de Requisitos

Este capítulo se centra en la recopilación de la información necesaria para definir y desarrollar las funcionalidades que el software debe cumplir para satisfacer las necesidades del usuario. Esto se realiza a través de las historias de usuario.

Los requisitos se clasifican en dos grupos: funcionales y no funcionales. Cada uno de ellos, es descrito de forma detallada, lo que permite una planificación más precisa y un desarrollo más efectivo.

También se presentan algunos de los casos de uso más relevantes, cubriendo cada una de las funcionalidades implementadas con, al menos, un caso de uso.

Capítulo 3: Diseño del software

En el capítulo 3 aborda el diseño del software, incluyendo la creación de distintos tipos de diagrama para visualizar la estructura del sistema. Se identificarán las mejores soluciones para implementar cada funcionalidad, teniendo en cuenta la escalabilidad, seguridad y mantenimiento del software.

Capítulo 4: Estudio de tecnologías utilizadas

En este capítulo, se mencionan todas las herramientas que han sido necesarias para el desarrollo del Trabajo de Fin de Grado. Se presentan tecnologías empleadas para la creación de los diagramas, las utilizadas durante el desarrollo del proyecto, así como las usadas para el testeado del código a través de pruebas de sistema.

Capítulo 5: Implementación

Durante este capítulo se explicará de forma detallada la implementación de las funcionalidades principales de la aplicación, incluyendo la implementación del foro, el recomendador, la red social y la lectura del libro en formato de audio.

Se detalla también una lista de las tecnologías y herramientas empleadas durante el proceso completo, abarcando desde la creación de los diagramas, hasta el desarrollo del código.

Capítulo 6: Pruebas

En este capítulo se presentarán pruebas realizadas para asegurar el correcto funcionamiento del código a través de pruebas de sistema. El fin de las mismas consiste en garantizar la interoperabilidad entre componentes, y validar la usabilidad.

Capítulo 7: Conclusiones y líneas futuras

En este último capítulo, se evaluarán los logros del proyecto, así como los problemas encontrados durante la realización del mismo. Se expondrán las conclusiones obtenidas tras la realización de este Trabajo de Fin de Grado, y se propondrán además posibles mejoras a implementar en futuras expansiones del código.

Apéndices

Con los apéndices, se pretende complementar y ampliar la información presentada, para garantizar una mejor comprensión del proyecto. Se incluirán para ello un manual de despliegue y un manual de usuario, proporcionando instrucciones detalladas para la correcta instalación, configuración, y uso de la aplicación. Estos manuales asegurarán que cualquier persona pueda desplegar y utilizar el sistema de manera efectiva.

1.4 Metodología de trabajo empleada

Para el desarrollo del proyecto, se ha adoptado una metodología ágil, específicamente Scrum, que se basa en un enfoque de desarrollo incremental dividido en bloques cortos denominados *sprints*. Esta metodología se ha elegido por su capacidad de adaptabilidad, lo que permite una planificación, desarrollo y validación continuas y flexibles de las funcionalidades, así como la rápida adaptación a cambios o nuevas necesidades que surjan durante el desarrollo.

1.4.1 Estructura de los *Sprints*

Cada *sprint* ha contado con una duración inicial aproximada de 2 semanas, con reuniones regulares con el tutor para revisar el progreso y realizar las adaptaciones necesarias. Sin embargo, algunas funcionalidades han requerido más tiempo de implementación, extendiendo la duración de ciertos *sprints* más allá de las 2 semanas. La estructura general de cada *sprint* ha sido la siguiente:

Planificación

Al inicio de cada *sprint*, se seleccionaron las historias de usuario a desarrollar, descomponiéndolas en tareas más manejables para simplificar el desarrollo y su seguimiento. Esta etapa de planificación era esencial para garantizar que cada tarea fuera comprendida y se centrara en los objetivos del proyecto.

Desarrollo

Durante la fase de desarrollo, se han implementado las funcionalidades planificadas siguiendo un enfoque incremental e iterativo. Cada funcionalidad se desarrollaba de forma modular, es decir, de forma independiente al resto, lo que permitía realizar entregas parciales al final de cada *sprint*, facilitando la integración progresiva con el resto del sistema.

Pruebas

De forma simultánea al desarrollo, se llevaron a cabo pruebas para asegurar la funcionalidad y calidad del código. Estas eran esenciales para verificar que cada componente

cumpliera con los requisitos establecidos, ayudando a detectar y corregir posibles errores de forma temprana, evitando problemas importantes en etapas posteriores.

Validación

Al concluir cada *sprint*, se realizaba una revisión exhaustiva de los avances con el tutor, con el fin de validar su funcionamiento. Este paso era fundamental para obtener retroalimentación, realizar ajustes necesarios, y asegurar que el proyecto avanzase en la dirección correcta antes de iniciar el siguiente *sprint*.

1.4.2 Descripción de los Sprints

Sprint 1

En el primer *sprint*, se marcó como primer objetivo el desarrollo e implementación del sistema de recomendación. Este se centró en construir su lógica, integrando los algoritmos necesarios para ofrecer sugerencias personalizadas a los usuarios.

Además de esta funcionalidad, dado que se debían construir las nuevas funciones sobre una aplicación ya existente, se realizó un estudio exhaustivo de las funcionalidades ya existentes y las conexiones internas en el código, lo que permitió una mayor comprensión del mismo para poder abordar las implementaciones de forma más acertada.

Aunque se planificó como un *sprint* de dos semanas, la complejidad encontrada al crear el algoritmo de recomendación junto a la necesidad de agregar una API para obtener la información de los libros, extendieron ligeramente la duración de este *sprint*.

Sprint 2

En este *sprint* se llevó a cabo el desarrollo del foro, incluyendo la capacidad de los usuarios para crear, comentar y gestionar discusiones. Se integraron también características de moderación de contenido, permitiendo a determinados usuarios realizar acciones como el borrado y la edición del contenido publicado.

Este *sprint* no supuso un gran problema, por lo que se ajustó a la duración inicialmente establecida, incluso llegando a ser desarrollada en menos tiempo del esperado.

Sprint 3

En el tercer *sprint* se realizó el desarrollo de la red social, incluyendo perfiles de usuario, seguimiento de actividades y conexiones entre usuarios. Se implementaron además características como las notificaciones, el buscador para encontrar nuevas conexiones con otros usuarios y la visualización de la interfaz del perfil.

La interacción entre usuarios y su integración con el resto del sistema en la base de datos supuso un reto al principio, extendiendo el *sprint* brevemente.

Sprint 4

En este *sprint* se implementó el sistema de audio, centrándose en la reproducción de libros y la gestión de capítulos. Además de la funcionalidad básica de reproducción, se desarrolló la capacidad de mostrar el tiempo total de lectura del capítulo, y se validó su actualización correcta al cambiar de capítulo.

Este *sprint* fue el más extenso de todos, llegando a superar las tres semanas de duración, puesto que se intentó desarrollar de diversas maneras sin éxito, hasta que se decidió incluir dentro del propio lector digital, mejorando la experiencia del usuario.

Sprints finales

Por último, se realizaron varios *sprints* finales donde se hizo una revisión completa del sistema, incluyendo pruebas de sistema, validación final de todas las funcionalidades y desarrollo de la memoria del proyecto. Estos *sprints* adicionales aseguraron que todas las funcionalidades trabajaran en conjunto de manera eficiente y que la documentación reflejara con precisión el trabajo realizado. La duración de estos *sprints* se ajustó a las dos semanas.

Siguiendo esta metodología, se ha logrado un desarrollo ordenado y flexible, permitiendo la corrección de errores y asegurando que el proyecto avanzase de manera eficiente. La estructura de los *sprints*, combinada con revisiones periódicas, ha facilitado un control continuo sobre el progreso, adaptándose a las necesidades y garantizando la entrega del producto final.

2

Análisis de Requisitos

En este capítulo se recopila información para desarrollar las historias de usuario que describan las funcionalidades esperadas del software. Para ello, se especificarán los requisitos funcionales y los no funcionales, garantizando la correcta planificación y ejecución del proyecto. Además, se presentarán los casos de uso más relevantes para mostrar el funcionamiento que debe tener cada funcionalidad.

2.1 Requisitos funcionales

Los requisitos funcionales son especificaciones que describen las funcionalidades y características que debe tener la aplicación para satisfacer las necesidades de los usuarios. Estos requisitos describen qué debe hacer el sistema, como la capacidad de crear foros, enviar mensajes directos o generar recomendaciones de libros.

Identificador	Requisito	Descripción
RF-01	Creación de foros	Los usuarios podrán crear nuevos foros sobre temas literarios de su interés.
RF-02	Creación de categorías	Los usuarios serán capaces de agregar nuevas categorías para emplearlas en la creación del foro.

RF-03	Creación de hilos	Los usuarios podrán crear nuevos hilos en el foro, así como añadir el primer comentario.
RF-04	Respuesta en hilos del foro	Los usuarios podrán responder a otros comentarios en los hilos de discusión existentes en el foro.
RF-05	Visualización del foro	Los usuarios podrán visualizar todos los foros existentes, así como acceder a sus hilos y ver los comentarios publicados.
RF-06	Moderación del foro	Los usuarios con rol de administrador podrán eliminar o editar foros, hilos, categorías y comentarios inapropiados en el foro.
RF-07	Seguimiento de usuarios	Los usuarios podrán seguir a otros usuarios para ver sus actividades en su perfil.
RF-08	Dejar de seguir usuarios	Los usuarios tendrán la opción de dejar de seguir a otros usuarios en cualquier momento.
RF-09	Perfil de usuario	Cada usuario tendrá un perfil donde se mostrarán sus libros descargados, sus estanterías, sus seguidores y seguidos y las publicaciones en el foro.
RF-10	Exploración de perfiles	Los usuarios podrán buscar perfiles de otros usuarios mediante su nombre de usuario.
RF-11	Mensajes directos	Los usuarios podrán enviar y recibir mensajes directos dentro de la aplicación, además de borrar sus mensajes.

RF-12	Notificaciones de interacción	Los usuarios recibirán notificaciones cuando, usuarios a los que sigan, creen o participen en publicaciones del foro. Las notificaciones podrán ser también eliminadas.
RF-13	Notificaciones de seguimiento	Los usuarios recibirán notificaciones cuando otro usuario comience a seguirlos.
RF-14	Notificaciones de mensajería	Los usuarios obtendrán notificaciones cuando otro usuario escriba un mensaje en el <i>chat</i> privado.
RF-15	Eliminación de notificación	Los usuarios serán capaces de borrar todas las notificaciones o algunas individualmente.
RF-16	Cuestionario de recomendación	La aplicación ofrecerá un cuestionario de preferencias literarias, que servirá al usuario para obtener recomendaciones de libros.
RF-17	Generación de recomendaciones	El sistema procesará las respuestas obtenidas en el cuestionario y generará recomendaciones personalizadas de libros para el usuario.
RF-18	Visualización de recomendaciones	El sistema mostrará al usuario una lista de libros recomendados, conteniendo datos como el título, autor, portada y descripción.
RF-19	Manejo de errores de recomendación	El sistema manejará los posibles errores obtenidos durante la generación de recomendaciones, proporcionando mensajes de estado en caso de no lograr obtener un libro o ciertos datos del mismo.

RF-20	Lectura de audiolibros	El sistema será capaz de leer el texto del libro con una voz preestablecida, de forma que los usuarios podrán escuchar los libros en formato de audio dentro de la aplicación.
RF-21	Control de reproducción de audio	Los usuarios podrán pausar y reanudar la reproducción de la lectura utilizando botones de control específicos.
RF-22	Visualización del progreso de audio	El sistema mostrará el progreso de la lectura dividida por capítulos a través de una barra de progreso, incluyendo el tiempo de lectura actual y el total.
RF-23	Control del progreso de audio	El usuario podrá avanzar y retroceder la lectura usando la barra de progreso, pudiendo leer desde el punto que desee.
RF-24	Actualización de tiempo total	El sistema calculará y actualizará el tiempo total de lectura para cada capítulo, basándose en la longitud del texto y la velocidad de lectura.

Cuadro 1. Tabla de Requisitos Funcionales

2.2 Requisitos no funcionales

Los requisitos no funcionales son características que especifican cómo debe actuar el sistema en términos de rendimiento, seguridad o usabilidad entre otros aspectos.

Identificador	Requisito	Descripción
RNF-01	Rendimiento	La aplicación debe ser capaz de manejar simultáneamente varios usuarios activos sin degradar su rendimiento.
RNF-02	Seguridad	La aplicación debe proteger los datos de usuario con encriptación SSL/TLS y almacenar contraseñas utilizando un algoritmo de hash seguro.
RNF-03	Escalabilidad	La arquitectura de la aplicación debe permitir su escalado para soportar un crecimiento de usuarios y datos sin necesidad de reescritura.
RNF-04	Compatibilidad multiplataforma	La aplicación debe ser compatible con los principales navegadores web y adaptable a dispositivos móviles y tablets.
RNF-05	Usabilidad	La interfaz de usuario debe ser intuitiva y fácil de navegar, con un tiempo de respuesta rápido por acción.
RNF-06	Mantenimiento	El sistema debe permitir actualizaciones y mantenimiento sin necesidad de detener el servicio o causar interrupciones al usuario final.

Cuadro 2. Tabla de Requisitos no Funcionales

Estos requisitos describen atributos del sistema que son cruciales para su funcionamiento y éxito, tales como la velocidad de respuesta, la capacidad para manejar una gran cantidad de usuarios, y la facilidad de uso de la aplicación. Cada uno de estos aspectos contribuye a garantizar que el sistema funcione de manera eficiente, sea capaz de soportar un alto volumen de tráfico y ofrezca una experiencia de usuario intuitiva y satisfactoria.

2.3 Casos de uso

Los casos de uso son descripciones de cómo los usuarios interactúan con un sistema para lograr un objetivo concreto. Cada caso de uso detalla un escenario en el que el usuario realiza una serie de acciones con el sistema para alcanzar un resultado deseado, como crear un foro, seguir a otro usuario o enviar un mensaje.

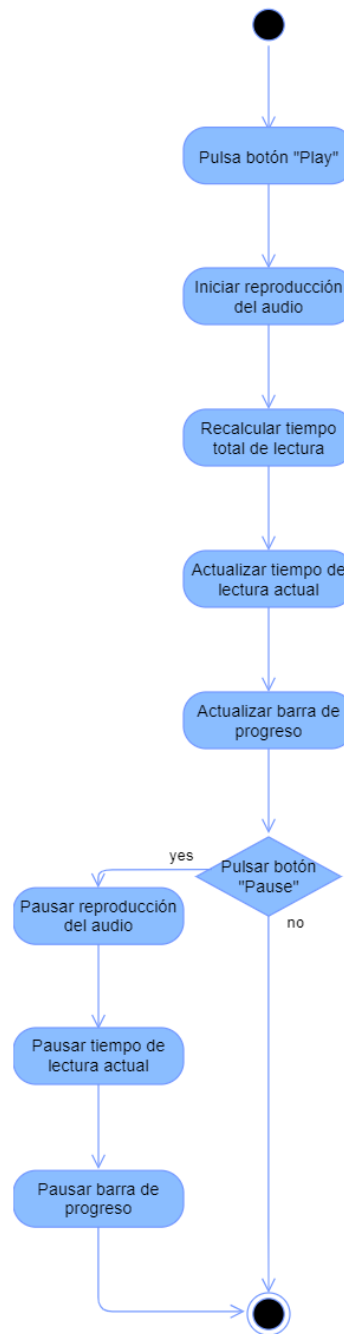
Estos casos ayudan a identificar y documentar los requisitos funcionales del sistema desde la perspectiva del usuario. A continuación, se muestran algunos de los casos de uso más relevantes, incluyendo al menos un caso de uso para cada funcionalidad añadida.

Identificador	CU-01
Título	Reproducir Audio
Descripción	El sistema debe reproducir el audio del libro cuando usuario pulse el botón <i>Play</i>
Requisitos	RF-20, RF-21, RF-22, RF-23, RF-24
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema y está en la página de lectura del libro ● El audio del libro está disponible y preparado para ser reproducido
Post-Condiciones	<ul style="list-style-type: none"> ● El audio del libro comienza a reproducirse o se reanuda si estaba pausado ● Se recalcula el tiempo total de lectura del libro

	<ul style="list-style-type: none"> ● Se actualiza el tiempo de lectura actual y la barra de progreso constantemente
Escenario principal	<ul style="list-style-type: none"> ● El usuario hace clic en el botón de <i>Play</i> en la interfaz del lector digital ● El sistema comienza a reproducir el audio del libro ● El sistema calcula el tiempo total de lectura del libro
Escenario alternativo	<ul style="list-style-type: none"> ● El usuario hace clic en el botón de <i>Pause</i> durante la reproducción ● El sistema pausa el audio y detiene la reproducción

Cuadro 3. CU-01: Reproducir Audio

Diagrama de actividad



Identificador	CU-02
Título	Recalcular Tiempo de Lectura
Descripción	El sistema recalcula el tiempo de lectura total y ajusta la barra de progreso cuando el usuario cambia de capítulo en el lector digital.

Requisitos	RF-20, RF-21, RF-22, RF-23, RF-24
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema y está en la página de lectura del libro ● El usuario está en el primer capítulo y el audio está en pausa
Post-Condiciones	<ul style="list-style-type: none"> ● El audio del libro comienza a reproducirse en el capítulo siguiente ● Se recalcula el tiempo total de lectura del libro mostrando el tiempo del nuevo capítulo ● Se actualiza el tiempo de lectura actual a 0:00 y la barra de progreso al inicio
Escenario principal	<ul style="list-style-type: none"> ● El usuario selecciona el botón para avanzar hasta llegar al siguiente capítulo ● El usuario pulsa el botón de Play ● El sistema comienza a reproducir el audio del libro ● El sistema recalcula el tiempo de lectura total para el nuevo capítulo ● El sistema restablece el tiempo de lectura actual a 0:00 ● El sistema actualiza la barra de progreso

Cuadro 4. CU-02: Recalcular Tiempo de Lectura

Diagrama de actividad

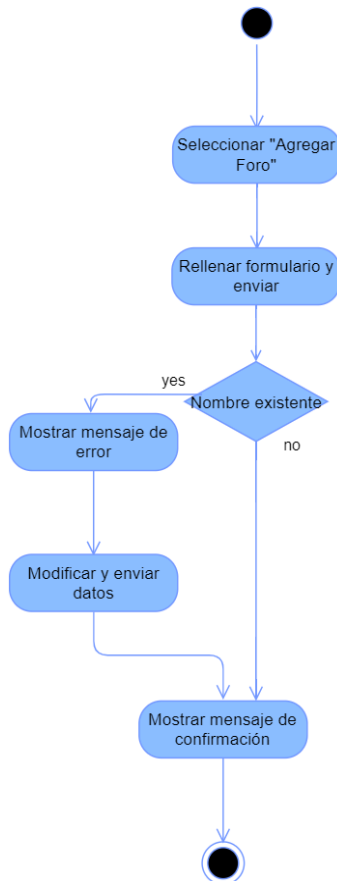


Identificador	CU-03
Título	Agregar Foro
Descripción	El usuario puede agregar un nuevo foro al sistema para iniciar nuevos temas de conversación.
Requisitos	RF-01, RF-05
Pre-Condiciones	<ul style="list-style-type: none">● El usuario ha iniciado sesión en el sistema y está en la página de foros
Post-Condiciones	<ul style="list-style-type: none">● El nuevo foro se agrega a la lista de foros disponibles● El sistema muestra un mensaje de confirmación indicando que el foro ha sido agregado con éxito
Escenario principal	<ul style="list-style-type: none">● El usuario selecciona el botón "Agregar Foro"

	<ul style="list-style-type: none"> ● El sistema muestra un formulario para ingresar detalles del nuevo foro ● El usuario completa el formulario con la información del foro y selecciona "Guardar" ● El sistema valida la información y agrega el nuevo foro a la lista ● El sistema muestra un mensaje de confirmación indicando que el foro ha sido agregado con éxito
Escenario alternativo	<ul style="list-style-type: none"> ● El nombre del foro ya existe y el sistema muestra un mensaje indicándolo ● El usuario modifica el nombre del foro y selecciona "Guardar"

Cuadro 5. CU-03: Agregar foro

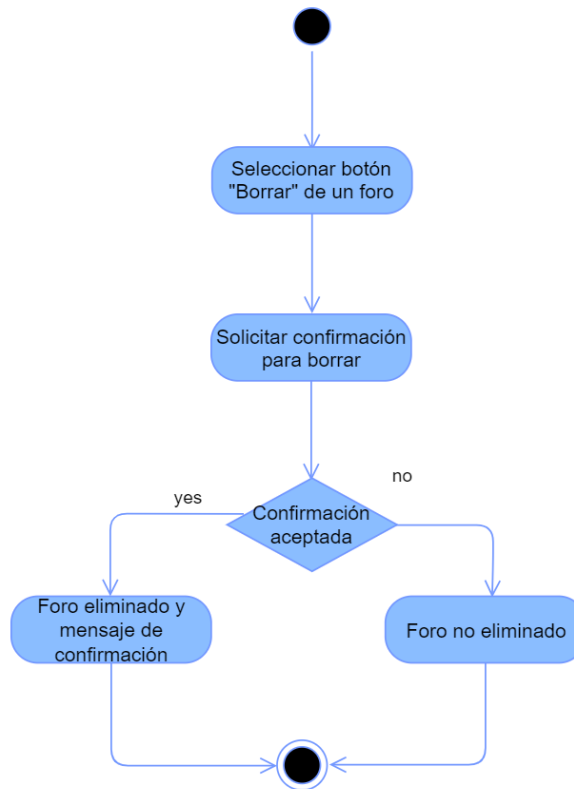
Diagrama de actividad



Identificador	CU-04
Título	Borrar foro
Descripción	Los administradores pueden borrar foros existentes para moderar la página
Requisitos	RF-05, RF-06, RNF-02
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema como administrador y está en la página de foros
Post-Condiciones	<ul style="list-style-type: none"> ● El foro seleccionado se elimina de la lista de foros ● El sistema muestra un mensaje de confirmación indicando que el foro ha sido borrado con éxito
Escenario principal	<ul style="list-style-type: none"> ● El usuario selecciona el botón "Borrar" junto al foro que desea eliminar ● El sistema solicita confirmación para borrar el foro ● El usuario confirma la acción ● El sistema elimina el foro y muestra un mensaje de confirmación indicando que el foro ha sido borrado con éxito

Cuadro 6. CU-04: Borrar Foro

Diagrama de actividad

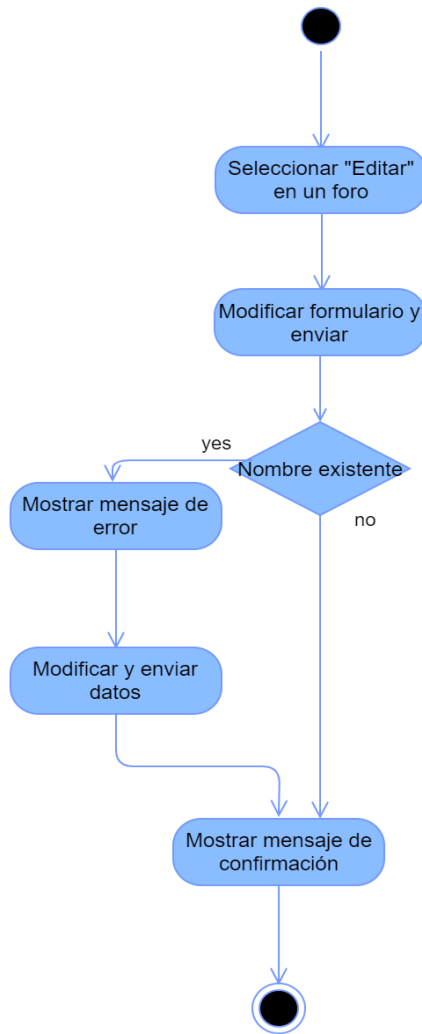


Identificador	CU-05
Título	Editar Foro
Descripción	El sistema permite a un usuario administrador editar la información de un foro existente
Requisitos	RF-05, RF-06, RNF-02
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema como administrador y está en la página de foros
Post-Condiciones	<ul style="list-style-type: none"> ● La información del foro se actualiza en el sistema ● El sistema muestra un mensaje de confirmación indicando que el foro ha sido editado con éxito
Escenario principal	<ul style="list-style-type: none"> ● El usuario selecciona el botón "Editar" del foro que desee modificar

	<ul style="list-style-type: none"> ● El sistema muestra un formulario con los detalles actuales del foro ● El usuario realiza las modificaciones necesarias y selecciona "Guardar" ● El sistema valida la información y actualiza los detalles del foro ● El sistema muestra un mensaje de confirmación indicando que el foro ha sido editado con éxito
Escenario alternativo	<ul style="list-style-type: none"> ● El nombre del foro ya existe y el sistema muestra un mensaje indicándolo ● El usuario modifica el nombre del foro y selecciona "Guardar"

Cuadro 7. CU-05: Editar Foro

Diagrama de actividad

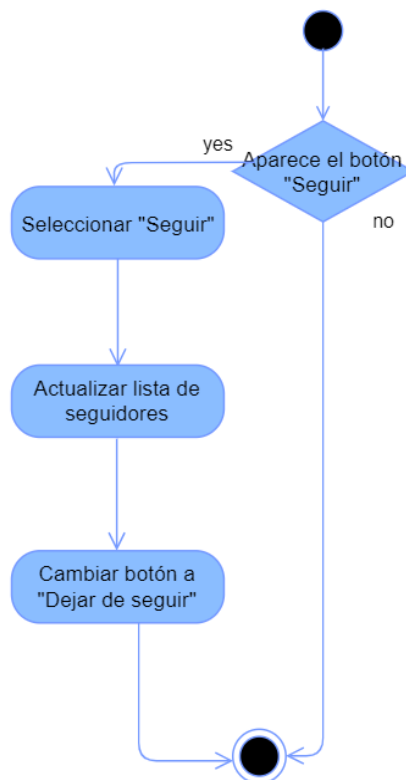


Identificador	CU-06
Título	Seguir a Otro Usuario
Descripción	El sistema permite a un usuario seguir a otro usuario para comunicarse con él y recibir actualizaciones de ciertas actividades
Requisitos	RF-07, RF-12, RF-13
Pre-Condiciones	<ul style="list-style-type: none"> • El usuario ha iniciado sesión en el sistema • El usuario está en la página del perfil del usuario que desea seguir

Post-Condiciones	<ul style="list-style-type: none"> • El usuario seleccionado es añadido a la lista de seguidores del usuario actual • El usuario actual comienza a recibir actualizaciones sobre el usuario seguido
Escenario principal	<ul style="list-style-type: none"> • El usuario selecciona el botón "Seguir" • El sistema procesa la solicitud y actualiza la lista de seguidores del usuario actual • El sistema cambia el botón del perfil para poder dejar de seguir al usuario
Escenario alternativo	<ul style="list-style-type: none"> • Si el usuario ya sigue al usuario objetivo, se muestra el botón "Dejar de seguir"

Cuadro 8. CU-06: Seguir a Otro Usuario

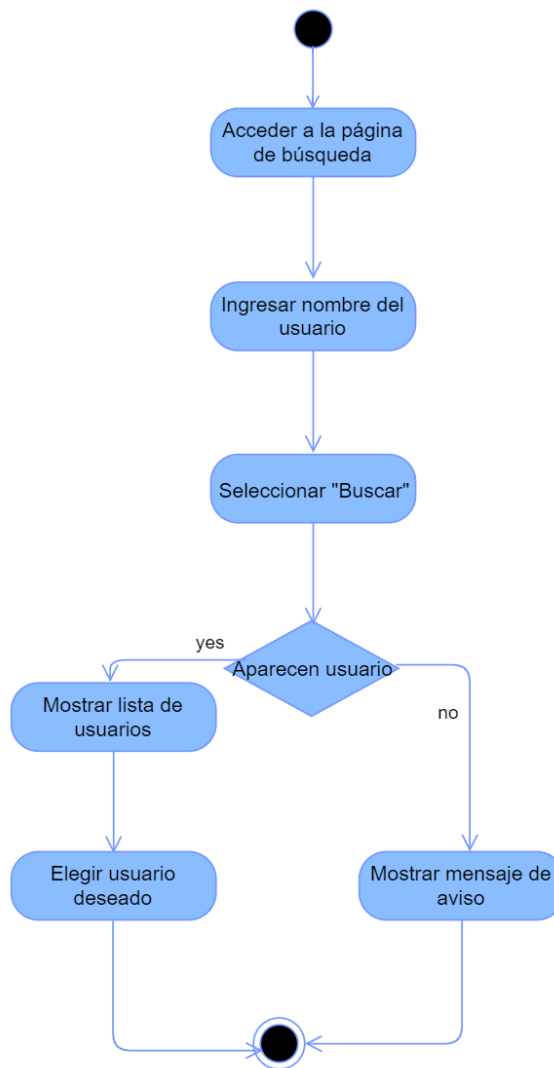
Diagrama de actividad



Identificador	CU-07
Título	Buscar Usuarios
Descripción	El sistema permite a un usuario buscar otros usuarios por su nombre para ver su perfil
Requisitos	RF-09, RF-10
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema ● El usuario está en la página de su perfil de usuario
Post-Condiciones	<ul style="list-style-type: none"> ● El sistema muestra una lista de usuarios que coinciden con los criterios de búsqueda
Escenario principal	<ul style="list-style-type: none"> ● El usuario accede a la página de búsqueda de usuarios ● El usuario ingresa el nombre del usuario que desea buscar en el campo de búsqueda ● El usuario selecciona el botón "Buscar" ● El sistema procesa la búsqueda y muestra una lista de usuarios que coinciden con los criterios de búsqueda ● El usuario elige el perfil requerido
Escenario alternativo	<ul style="list-style-type: none"> ● El sistema no encuentra usuarios con dicho nombre y muestra un mensaje indicando que no se encontraron usuarios

Cuadro 9. CU-07: Buscar Usuarios

Diagrama de actividad

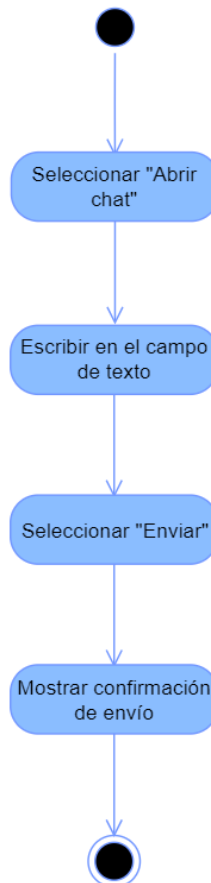


Identificador	CU-08
Título	Mandar Mensaje
Descripción	El sistema permite a un usuario enviar un mensaje directo a otro usuario
Requisitos	RF-11, RF-14
Pre-Condiciones	<ul style="list-style-type: none"> • El usuario ha iniciado sesión en el sistema • El usuario está en el perfil al que quiere mandar el mensaje

Post-Condiciones	<ul style="list-style-type: none"> ● El mensaje es enviado al usuario destinatario ● El usuario destinatario recibe una notificación del mensaje
Escenario principal	<ul style="list-style-type: none"> ● El usuario abre el sistema de mensajería seleccionando "Abrir chat" ● El usuario escribe el mensaje en el campo de texto ● El usuario selecciona el botón "Enviar" ● El sistema envía el mensaje y muestra una confirmación de que el mensaje ha sido enviado

Cuadro 10. CU-08: Mandar Mensaje

Diagrama de actividad

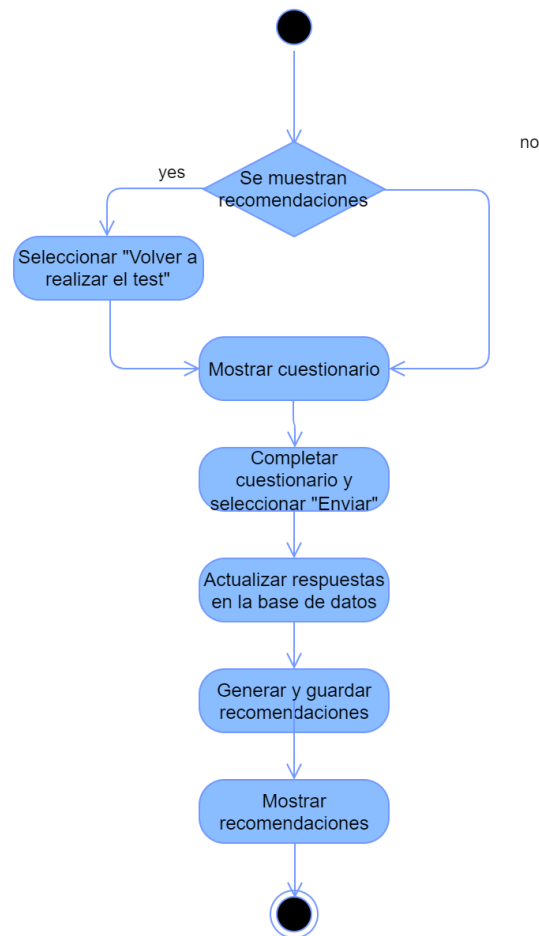


Identificador	CU-09
Título	Obtener Recomendaciones

Descripción	El sistema permite a un usuario realizar un <i>test</i> de recomendaciones de libros, calcular recomendaciones personalizadas y visualizarlas
Requisitos	RF-16, RF-17, RF-18, RF-19
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema ● El usuario está en la página de inicio del recomendador
Post-Condiciones	<ul style="list-style-type: none"> ● Las respuestas del usuario se guardan en la base de datos ● Se generan recomendaciones de libros basadas en las respuestas del test ● Las recomendaciones se guardan y se muestran al usuario
Escenario principal	<ul style="list-style-type: none"> ● El sistema muestra una serie de preguntas relacionadas con sus preferencias literarias ● El usuario selecciona sus respuestas y hace clic en "Enviar" ● Las respuestas se actualizan en la base de datos del sistema ● El sistema procesa las respuestas utilizando el algoritmo de recomendaciones y genera una lista de libros que podrían interesarle al usuario ● El sistema muestra una lista de recomendaciones de libros, con detalles como el título, autor y portada
Escenario alternativo	<ul style="list-style-type: none"> ● Ya existían recomendaciones generadas previamente, por lo que el usuario selecciona en "Volver a realizar el test"

Cuadro 11. CU-09: Obtener Recomendaciones

Diagrama de actividad

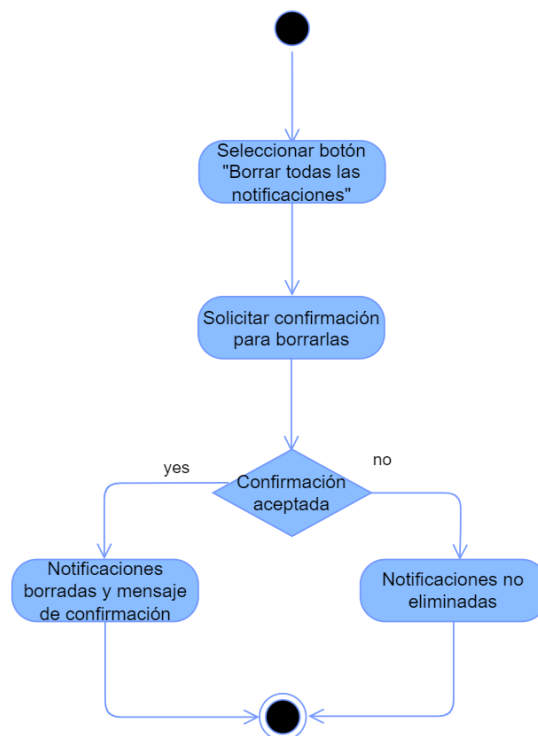


Identificador	CU-10
Título	Borrar Todas las Notificaciones
Descripción	El sistema permite a un usuario borrar todas sus notificaciones del sistema para tener un entorno más limpio
Requisitos	RF-16, RF-17, RF-18, RF-19
Pre-Condiciones	<ul style="list-style-type: none"> ● El usuario ha iniciado sesión en el sistema ● El usuario está en la página de notificaciones ● El usuario tiene al menos una notificación en su página
Post-Condiciones	<ul style="list-style-type: none"> ● Se eliminan todas las notificaciones y se muestra un mensaje indicando que no hay notificaciones

Escenario principal	<ul style="list-style-type: none"> ● El usuario selecciona el botón “Borrar todas las notificaciones” ● El sistema solicita confirmación para borrar las notificaciones ● El usuario confirma la acción ● El sistema elimina las notificaciones y muestra un mensaje de confirmación indicando que han sido eliminadas con éxito
Escenario alternativo	<ul style="list-style-type: none"> ● No se encuentran notificaciones, por lo que se muestra un mensaje avisando la ausencia de notificaciones

Cuadro 12. CU-010: Borrar Todas las Notificaciones

Diagrama de actividad



3

Estudio de tecnologías utilizadas

En esta sección se detallan las tecnologías utilizadas a lo largo del desarrollo del proyecto, abarcando desde la fase de implementación hasta los procesos relacionados anteriores o posteriores. Cada tecnología seleccionada ha sido clave para alcanzar los objetivos propuestos y optimizar la funcionalidad de la aplicación.

3.1 Lenguajes

Para el desarrollo del proyecto se utilizaron principalmente Python y HTML. Python se empleó por su versatilidad y extensa colección de librerías para el backend, mientras que HTML se utilizó para estructurar el contenido y diseñar las interfaces del frontend, proporcionando una base sólida para la presentación de la información y la interactividad del usuario.

- **Python.** Lenguaje de programación de alto nivel usado a nivel global por su simplicidad. Python ha sido utilizado para el desarrollo del servidor y la gestión de la interacción con la base de datos a través el framework Flask. Dada su versatilidad y la gran

cantidad de bibliotecas a las que tiene acceso, se trata de una herramienta idónea para el desarrollo de aplicaciones web.

- **HTML.** Es el estándar de marcado para crear contenido en la web. Se usa en todas las páginas web de la aplicación para definir su estructura, permitiendo añadir elementos como texto, botones o enlaces.
- **JavaScript.** Es un lenguaje de programación usado para el desarrollo web que permite implementar funcionalidades dinámicas en las páginas, lo que ayuda a mejorar la experiencia del usuario haciendo la web más interactiva.
- **CSS.** Es un lenguaje de hojas de estilo usado para definir el diseño de las páginas web. Gracias a esta herramienta, ha sido posible estilizar la aplicación para mostrarla de forma más atractiva y coherente.

3.2 Frameworks y Librerías

El proyecto hizo uso de frameworks como Flask para el desarrollo web, gracias a su flexibilidad, y librerías como Bootstrap para diseñar interfaces atractivas. Estas herramientas aceleraron el desarrollo y facilitaron la integración de funcionalidades complejas, mejorando tanto la apariencia como la usabilidad de la aplicación.

- **Flask.** Es un framework web flexible para Python que facilita el desarrollo de aplicaciones web, gestionando las rutas, manejando las solicitudes del servidor y permitiendo construir las funcionalidades del backend.
- **SQLAlchemy.** Utiliza la técnica de mapeo relacional de objetos u ORM, que facilita la interacción con bases de datos relacionales al mapear objetos Python a tablas de bases de datos. Se ha empleado en el código para la correcta integración de los datos en la base de datos.

- **Werkzeug.** Biblioteca de utilidades WSGI (Interfaz de Pasarela del Servidor Web) que proporciona herramientas para construir aplicaciones web. Tiene componentes como 'generate_password_hash' que permiten crear o verificar contraseñas hash para una mayor seguridad de estas.
- **numpy (Numerical Python).** Es una librería necesaria para el cálculo científico en Python. Da soporte para una amplia variedad de funciones matemáticas para operar con datos. En el código, se utiliza la operación estadística `numpy.mean` para el cálculo de la media de los elementos de un arreglo, con la intención de calcular, posteriormente, probabilidades.
- **Bootstrap.** Es un framework de diseño de código abierto que ofrece componentes predefinidos para la creación de interfaces web más atractivas. Bootstrap se ha utilizado para crear un diseño consistente en la aplicación, mejorando la experiencia del usuario.

3.3 Entornos de Desarrollo

Se eligió Visual Studio Code como entorno de desarrollo debido a su capacidad para manejar múltiples lenguajes y su amplia gama de extensiones útiles. Este entorno ofreció una buena integración con el resto de las herramientas y una interfaz amigable para la codificación.

- **Visual Studio Code.** Es un editor de código que ofrece soporte para múltiples lenguajes de programación, proporcionando un entorno de desarrollo eficiente y personalizable. Ha sido el entorno principal utilizado para escribir, editar y depurar el código de la aplicación.

3.4 Herramientas de Gestión de Proyectos

Para la gestión del proyecto se utilizó GitHub, que facilitó la planificación y seguimiento del progreso.

- **Github.** Es una plataforma de desarrollo que utiliza el sistema de control de versiones Git. Esta plataforma ha sido fundamental para el seguimiento de cambios en el código y la gestión de versiones del proyecto.

3.5 Herramientas de Diseño

El diseño del proyecto se realizó con Canva y Gitmind. Canva se empleó para la creación de elementos gráficos, mientras que Gitmind se utilizó para elaborar diagramas, facilitando la visualización de estructuras y flujos.

- **Canva.** Es una herramienta de diseño gráfico en línea que permite crear imágenes de forma sencilla. Canva ha sido de ayuda para diseñar algunos iconos y otros elementos visuales, mejorando la estética del proyecto.

- **Gitmind.** Es una herramienta en línea para crear el diseño de distintos tipos de diagramas o mapas mentales, que facilita una amplia gama de plantillas y opciones de personalización. Se ha hecho uso de Gitmind para generar los diagramas de clase.

3.6 Herramientas de Bases de Datos y Modelado

SQLite se utilizó como sistema de gestión de bases de datos por su facilidad de integración. Además, la herramienta Data Modeler ayudó a visualizar y diseñar la estructura de la base de datos, asegurando una organización eficiente de los datos.

- **Data Modeler.** Es una herramienta de modelado utilizada para crear el diseño y la estructura de bases de datos. Se ha empleado principalmente para crear diagramas que representan las entidades, relaciones y atributos en la base de datos, permitiendo visualizar cómo interactúan entre sí.

- **Sqlite3.** Es un sistema de gestión de bases de datos relacional, empleado para visualizar el contenido, facilitando una base de datos sencilla.

3.7 Herramientas de Pruebas

Para asegurar la calidad del código, se emplearon herramientas de pruebas como Selenium para realizar pruebas automatizadas en el navegador. Estas herramientas ayudaron a identificar errores y garantizar que las funcionalidades del proyecto se comportaran como se esperaba.

- **Pytest.** Pytest es un framework para Python que facilita la creación de pruebas simples y escalables. Es muy flexible y permite escribir pruebas con un código conciso. Pytest es capaz de soportar pruebas unitarias, pruebas funcionales y pruebas de integración.
- **Selenium.** Es una suite de herramientas para la automatización de navegadores web. Permite realizar pruebas automatizadas de aplicaciones web en diferentes navegadores, simulando la interacción del usuario con la interfaz de forma más rápida. Selenium es ideal para pruebas funcionales y de regresión de aplicaciones web, y es ampliamente utilizado para la automatización de pruebas en entornos de desarrollo ágil.

4

Diseño del Software

Para comprender y gestionar la estructura y comportamiento general de un sistema, es esencial usar herramientas de modelización que faciliten una visión más detallada del diseño. Los diagramas de clase y los diagramas entidad relación son dos herramientas fundamentales en este proceso. Ambos tipos facilitan la identificación y resolución de problemas del diseño, garantizando una solución más eficiente.

4.1 Diagrama de Clase

El diagrama de clase es una herramienta que representa la estructura estática del sistema. Muestra sus clases, atributos, métodos y las relaciones entre ellas.

Este diagrama permite visualizar cómo se organizan y relacionan los diferentes componentes del sistema, proporcionando una visión clara de la arquitectura del software y facilitando el diseño y la implementación de clases.

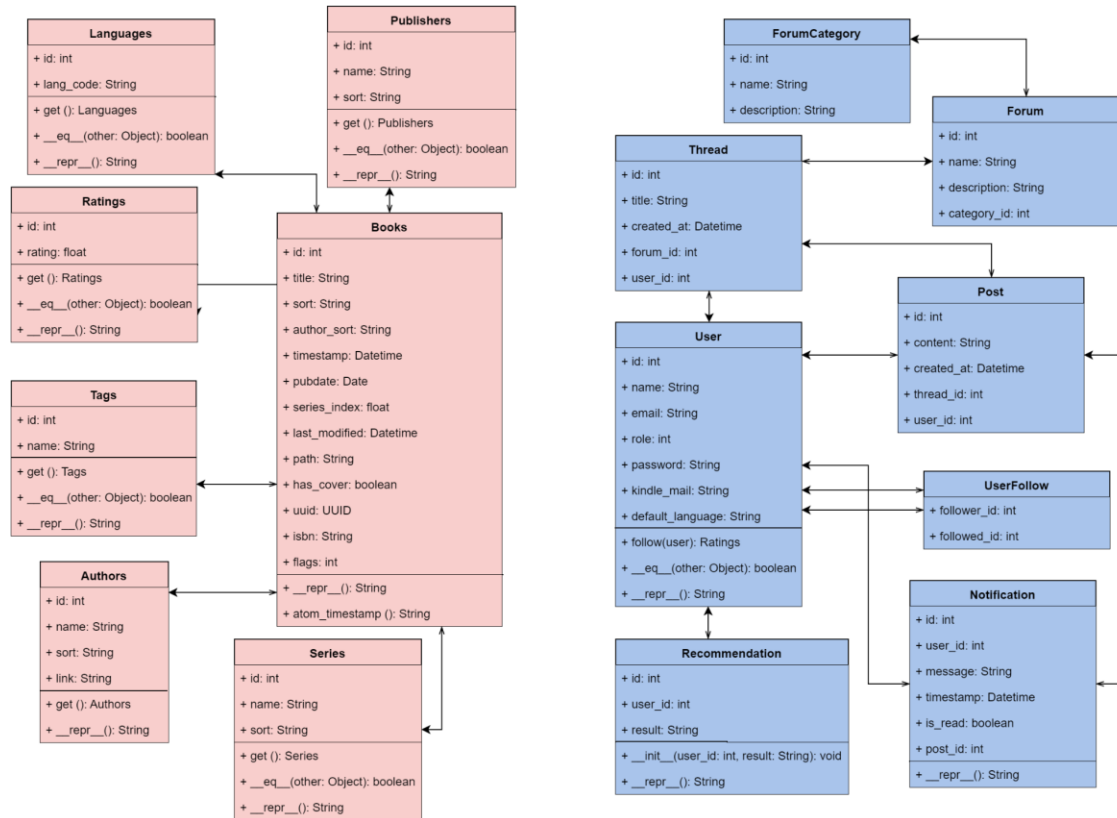


Figura 1. Diagrama de clases donde se especifican las clases con sus atributos, métodos y relaciones, mostrando la interacción dentro del sistema.

En la Figura 1 se muestra el diagrama de clases de la aplicación. Se ha hecho una distinción de colores para remarcar las clases ya existentes, las rosas, y las clases que ha sido necesario agregar o modificar para la creación de las funcionalidades, la azul.

A continuación, se aporta una breve descripción detallando la función de cada clase para tener un mayor contexto de la aplicación.

La clase ‘User’ representa a los individuos que interactúan con la aplicación. A pesar de existir previamente, ha sido necesaria su modificación para permitir a los usuarios participar en diversas actividades como crear foros e hilos de discusión, realizar publicaciones, seguir a otros usuarios, recibir recomendaciones y obtener notificaciones.

‘ForumCategory’ es una clase que clasifica los foros dentro de la aplicación al agruparlos bajo categorías temáticas comunes. Esto facilita a los usuarios la navegación y el acceso a las discusiones que más les interesan. Esta estructura es fundamental para manejar grandes volúmenes de contenido y asegurar que los usuarios puedan encontrar fácilmente foros que correspondan a sus intereses.

La clase 'Forum' representa un espacio donde los usuarios pueden crear y participar en discusiones. En los foros se agrupan hilos de conversación sobre temas específicos, permitiendo una organización clara del contenido generado por la comunidad. La relación con 'ForumCategory' permite a los foros estar clasificados bajo temas comunes, facilitando la navegación de los usuarios.

'Thread' define los hilos de discusión dentro de un foro. Cada hilo es una conversación centrada en un tema específico, iniciada por un usuario, y abierta para que otros usuarios contribuyan con sus publicaciones. Los hilos consisten en contenedores de publicaciones (definidos por la clase 'Post'), facilitando el seguimiento de conversaciones particulares.

'Post' estructura las publicaciones o mensajes individuales dentro de un hilo de discusión. Cada mensaje es publicado por usuario en un hilo como inicio del mismo o como respuesta a otro mensaje existente en el hilo, y es por tanto la unidad básica de interacción en los foros.

La clase 'UserFollow' controla la funcionalidad de seguimiento entre usuarios, permitiendo que un usuario siga algunas de las actividades que realicen otros. Al seguir a otros usuarios, los individuos pueden mantenerse al tanto de las discusiones en las que participen, fortaleciendo el apartado social de la aplicación.

La clase 'Recommendation' implementa un algoritmo basado en el teorema de Bayes que permite generar recomendaciones de libros personalizadas para los usuarios. Estas recomendaciones se basan en las respuestas proporcionadas por el usuario en un cuestionario. El algoritmo genera un conjunto de libros recomendados, mostrando detalles como el título, autor y fecha de publicación.

'Notification' se encarga de las notificaciones que los usuarios reciben, alertándolos sobre mensajes recibidos o actividades de otros usuarios. Las notificaciones incluyen avisos sobre nuevas publicaciones en hilos por parte de usuarios a los que sigue, alertas de nuevos seguidores y por mensajes recibidos. Esta clase es fundamental para mantener a los usuarios informados y comprometidos con la aplicación.

4.2 Diagrama de Entidad Relación

Los diagramas de entidad-relación (ER) son representaciones empleadas para visualizar la estructura y las relaciones entre entidades persistentes de la aplicación, es decir, aquellos elementos que se almacenan en la base de datos, mostrando cómo se organizan y conectan

internamente. Estos diagramas muestran las entidades (tablas), sus atributos y las relaciones entre ellas. Los diagramas ER ayudan a estructurar y organizar datos de manera que se facilite la comprensión y el diseño de la base de datos.

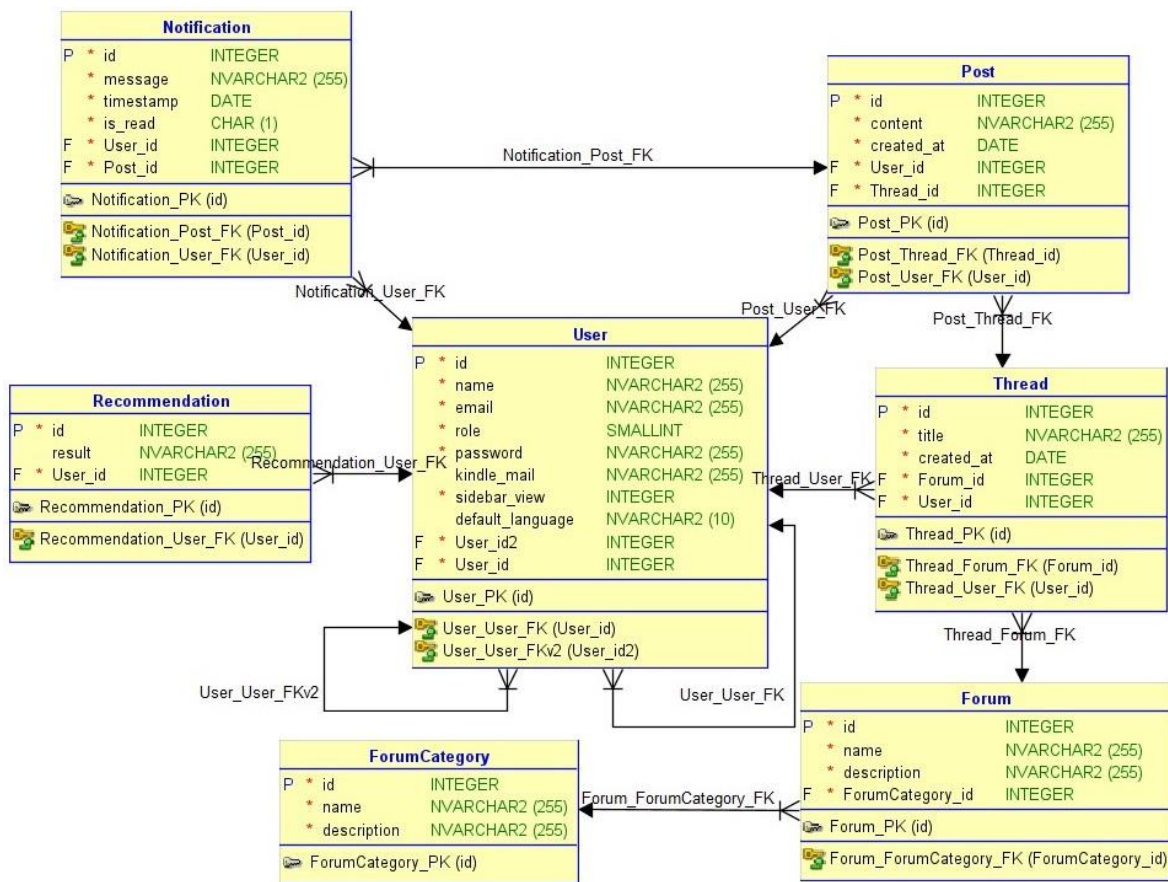


Figura 2. Diagrama ER donde se indican las clases de la base de datos relacionadas con los libros, mostrando sus atributos y relaciones.

En la Figura 2, se presenta el diagrama de entidad-relación con las entidades y relaciones previamente existentes que se han considerado más relevantes en la aplicación. Estas entidades constituyen la base del sistema, gestionando aspectos esenciales como la información de los libros, autores e idiomas, permitiendo una interacción básica con relación a los libros.

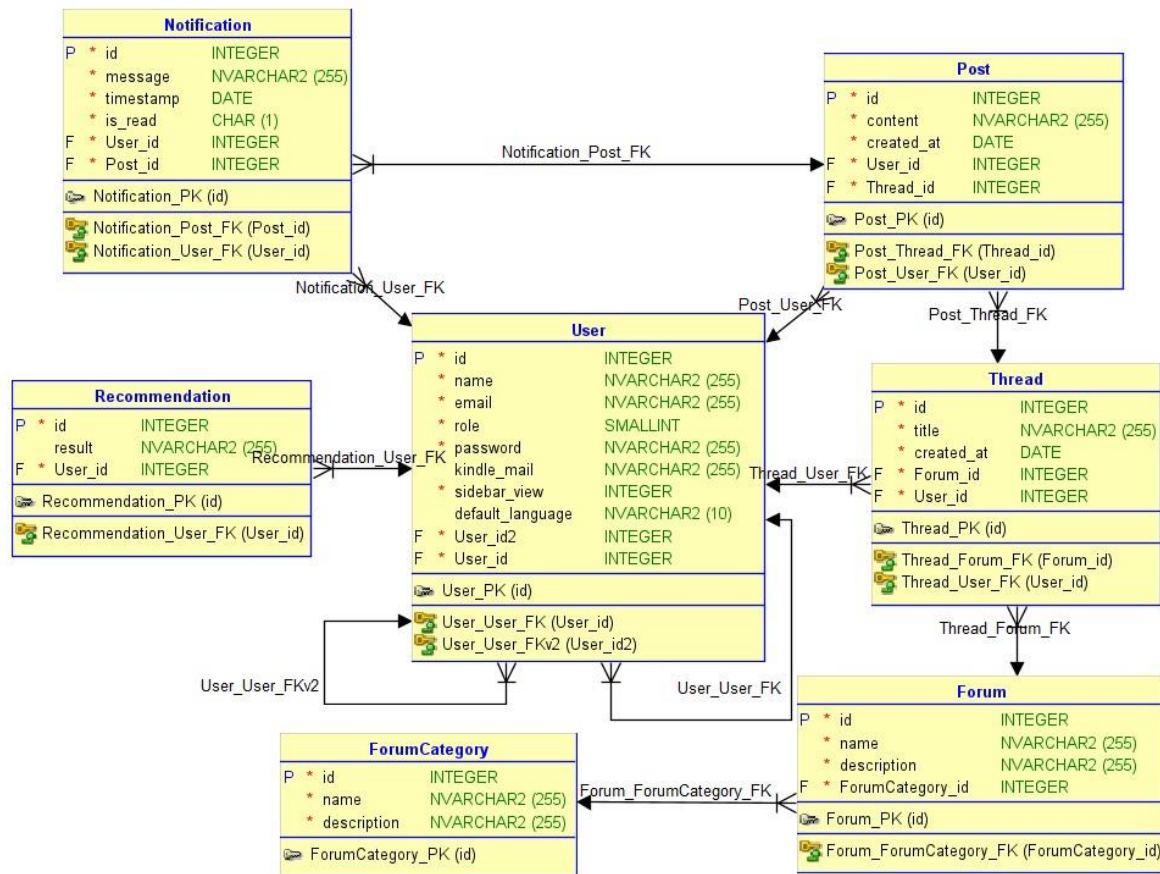


Figura 3. Diagrama ER donde se muestran las clases de la base de datos relacionadas con los usuarios, mostrando sus atributos y relaciones.

En la Figura 3, se muestra el diagrama ER actualizado, que incorpora nuevas entidades y relaciones desarrolladas específicamente para este proyecto. A la entidad ‘User’, que ya existía en la aplicación, se le han añadido nuevas relaciones y atributos para ampliar su funcionalidad. Además, se han creado entidades como ‘Recommendation’ o ‘Notification’, entre otras, para mejorar la interacción y personalización dentro de la plataforma.

La entidad ‘User’ es el núcleo central de la aplicación, representando a las personas que interactúan dentro de ella. Cada usuario cuenta con atributos como su nombre, correo electrónico, rol, y otros detalles personales que facilitan la personalización y seguridad dentro del sistema. Los usuarios son el eje de todas las actividades, por lo que están relacionados con muchas entidades, lo que les permite participar en discusiones o recibir notificaciones entre otras funciones. Además, la entidad cuenta con una autorrelación que permite manejar el seguimiento entre usuarios, favoreciendo la construcción de comunidades dentro de la aplicación.

‘ForumCategory’ es una entidad que organiza los foros en la aplicación bajo diferentes temas de interés. Esta categorización es esencial para mantener la estructura y facilitar la navegación, permitiendo a los usuarios encontrar fácilmente foros que coincidan con sus intereses.

‘Forum’ representa el espacio donde ocurren las discusiones dentro de la aplicación. Cada foro está vinculado con un ‘ForumCategory’, lo que asegura la agrupación de los hilos de discusión, permitiendo a los usuarios debatir sobre temas particulares de forma organizada.

La entidad ‘Thread’ es fundamental para la organización de las discusiones dentro de un foro. Un hilo es iniciado por un usuario y actúa como un tema específico dentro de un foro, donde otros usuarios pueden contribuir con sus opiniones a través de publicaciones.

‘Post’ representa la unidad básica de interacción dentro de un hilo de discusión. Cada publicación es el aporte de un usuario a una conversación específica dentro de un hilo, y puede incluir cualquier tipo de contenido relacionado con el tema discutido, indicando en todo momento su autoría.

La entidad ‘Notification’ gestiona las alertas que los usuarios reciben sobre diferentes actividades dentro de la aplicación. Estas notificaciones son esenciales para mantener a los usuarios informados del contenido en la plataforma, notificándolos sobre nuevas publicaciones en hilos de usuarios que siguen, mensajes recibidos, o nuevos seguidores.

‘Recommendation’ está diseñada para mejorar la experiencia del usuario dentro de la aplicación. Esta entidad almacena las recomendaciones de libros generadas por el algoritmo del recomendador, permitiendo que puedan ser recuperadas y mostradas nuevamente al usuario. Esta persistencia asegura que las recomendaciones sean accesibles incluso después de haber sido generadas inicialmente.

5

Implementación

La implementación de este proyecto ha sido desarrollada a partir de una aplicación web existente tomada como base, que cuenta individualmente con múltiples funcionalidades integradas. Esta aplicación permite a los usuarios la visualización de libros en varios formatos digitales (EPUB, MOBI, PDF...) y calificar o descargar los libros entre otras opciones, facilitando un acceso offline a su biblioteca personal.

Dado el considerable potencial de la aplicación original, se consideró la idea de expandir su capacidad. El proyecto no solo ha buscado preservar las características fundamentales con las que ya contaba, sino también expandirlas y unificarlas dentro de un entorno más completo. De este modo, se pretende ofrecer a los usuarios una experiencia más integrada y enriquecida. Se han añadido nuevas funcionalidades que complementan las existentes, haciendo de esta plataforma una herramienta aún más poderosa y versátil para los amantes de la lectura.

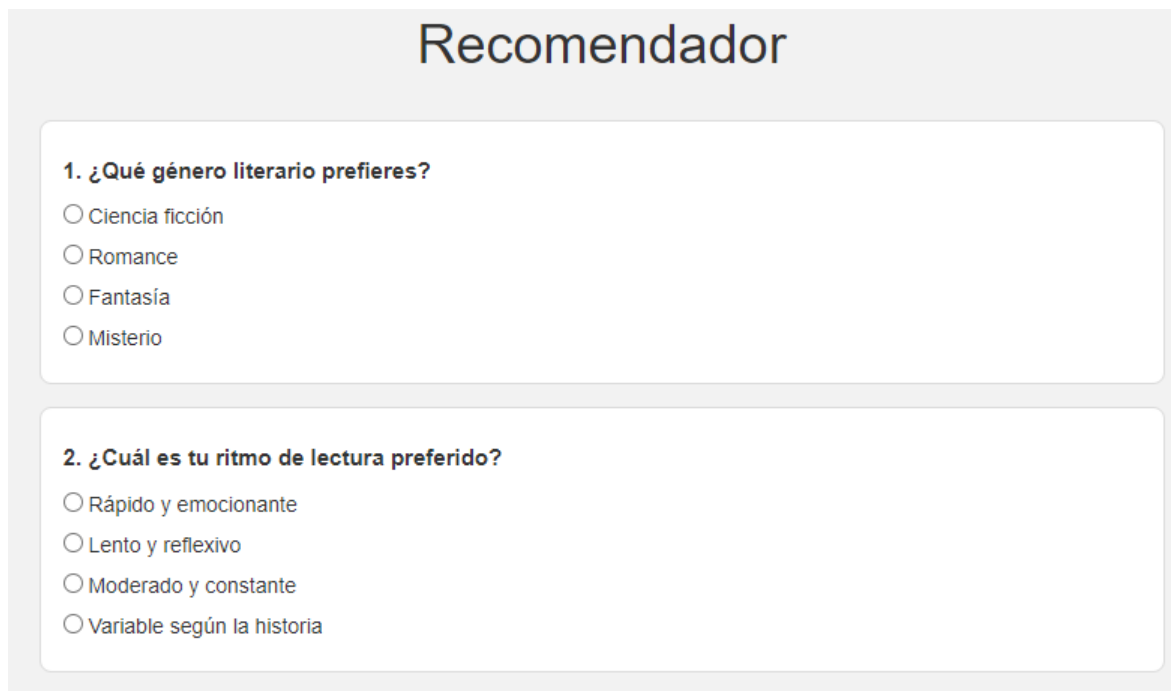
Para mejorar la experiencia del usuario, se han introducido cuatro funcionalidades esenciales que amplían el alcance de la aplicación. Estas complementan algunas de las ya existentes y abren nuevas posibilidades de interacción para los usuarios, contribuyendo a un entorno más dinámico y adaptado a las necesidades de todo tipo de lectores.

A continuación, se describen en detalle cada una de estas funcionalidades.

5.1 Recomendador

Cuestionario del Recomendador

El recomendador de libros cuenta con un cuestionario de preguntas con opción múltiple que recopila las preferencias del usuario. La interfaz está diseñada para ser intuitiva y fácil de usar. En la Figura 4 se presenta un fragmento del cuestionario para mostrar su forma.



Recomendador

1. ¿Qué género literario prefieres?

- Ciencia ficción
- Romance
- Fantasía
- Misterio

2. ¿Cuál es tu ritmo de lectura preferido?

- Rápido y emocionante
- Lento y reflexivo
- Moderado y constante
- Variable según la historia

Figura 4. Fragmento de la interfaz del recomendador, mostrando la estructura de las preguntas.

Generación de Recomendaciones

Una vez almacenadas las respuestas, el sistema las procesa para generar una lista de libros recomendados. Este procesamiento implica el uso de un algoritmo de recomendación basado en el Teorema de Bayes, que calcula la probabilidad de que un conjunto de libros se adecúe a las preferencias del usuario.

El Teorema de Bayes es un principio que nos permite calcular la probabilidad de un evento basándonos en condiciones previas relacionadas con dicho evento. Es especialmente útil en situaciones donde se dispone de información previa y se necesita actualizar las probabilidades al obtener nuevas evidencias (Chaves, 2024). El Teorema de Bayes se expresa mediante la siguiente fórmula:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Siendo:

- $P(A|B)$ es la probabilidad condicional, que mide la probabilidad de que ocurra el evento A si ya ha ocurrido el evento B.
- $P(B|A)$ es la probabilidad de que ocurra el evento B dado el evento A.
- $P(A)$ es la probabilidad de que ocurra el evento A sin estar condicionado por la ocurrencia de otro evento.
- $P(B)$ es la probabilidad de que ocurra el evento B independientemente de si ocurre A o no.

Este teorema permite actualizar la probabilidad de un evento A cuando se conoce una nueva evidencia B. Es particularmente útil en contextos como el aprendizaje automático, donde las probabilidades se ajustan continuamente a medida que se recibe nueva información. Esto se ve reflejado en el recomendador con la contestación de cada pregunta, de forma que se va añadiendo información nueva con cada una. De esta manera, se le permite tomar decisiones más precisas y basadas en la evidencia.

El algoritmo utiliza la función 'calculate_probabilities' para determinar cuáles son los libros más adecuados para el usuario en función de las respuestas obtenidas. Esta función calcula la probabilidad de que cada conjunto de libros de la lista sea la mejor opción dadas las respuestas del usuario, esto se realiza aplicando el Teorema de Bayes a través de la siguiente fórmula:

$$P(\text{book}|\text{answers}) = \frac{P(\text{answers}|\text{book}) \cdot P(\text{book})}{P(\text{answers})}$$

Al inicio, cada conjunto de libros tiene la misma probabilidad de ser seleccionado, por lo que la probabilidad individual 'P(book)' se reparte de forma uniforme como se ve en la siguiente línea de código.

```
P_book = 1 / len(books)
```

Posteriormente, se debe calcular la probabilidad condicional de cada opción, 'P(answers|book)'. Por cada pregunta respondida, la probabilidad de la respuesta dado el libro se actualiza multiplicando el valor actual por la diferencia entre la respuesta dada y la respuesta esperada para el libro, como se puede ver en la línea siguiente.

```
P_answers_given_book *= max(1 - abs(answer - book_answer(book, question)), 0.01)
```

La probabilidad de que la respuesta no corresponda al libro que estamos evaluando, se calcula con el promedio de la diferencia entre la respuesta dada y la esperada para otros libros. Esto se muestra en el siguiente fragmento de código.

```
P_answer_not_book = np.mean([1 - abs(answer -
book_answer(not_book, question))
```

Se debe calcular la probabilidad total de las respuestas, 'P_answers', combinando dos factores:

- La probabilidad de que las respuestas coincidan con las características del libro que se está observando.

Calculado como: $P(\text{answers}|\text{book}) \cdot P(\text{book})$

- La probabilidad de que las respuestas no coincidan con las características del libro que se está observando.

Calculado de la siguiente forma: $P(\text{answers}|\text{not book}) \cdot P(\text{not book})$

Este algoritmo fue modificado tomando como base uno ya existente basado en la estructura de Akinator, una popular aplicación que utiliza un proceso de preguntas y respuestas para adivinar en qué personaje está pensando el usuario. A través de preguntas sucesivas, Akinator reduce el conjunto de posibles personajes utilizando el teorema de Bayes, ajustando las probabilidades en función de las respuestas del usuario hasta llegar a una conclusión. Este proceso es el que sigue de forma aproximada nuestro recomendador (Clarke, 2020).

Esto se realiza en el código que se muestra a continuación.

```
P_answers = P_book * P_answers_given_book + (1 - P_book)
* P_answers_given_not_book
```

Finalmente, se usa la fórmula de Bayes anteriormente mencionada para calcular la probabilidad de que el libro que se está evaluando sea la recomendación correcta dadas las respuestas obtenidas. Se debe multiplicar la probabilidad de las respuestas dado el libro por la probabilidad del libro, y se divide por la probabilidad total de las respuestas.

```
P_book_given_answers = (P_answers_given_book * P_book) /
P_answers
```

De esta forma, se obtiene una recomendación ajustada a la probabilidad de las respuestas generadas por el usuario, aportando un conjunto de libros recomendados de forma adecuada. Esto se muestra al usuario en una interfaz donde cada libro aparece con su título, autor, portada y otros detalles relevantes.

A continuación, se muestra la Figura 5, donde se puede observar uno de los resultados obtenidos de las recomendaciones, para tener una visión más acertada de la información que podemos obtener de cada libro.



Figura 5. Ejemplo de una recomendación generada por el sistema, mostrando la portada del libro junto con información clave como el título, autor, e ISBN.

Google Books API

Para buscar los libros requeridos en el recomendador, se hace uso de la Google Books API. Se trata de una interfaz proporcionada por Google que permite acceder a información sobre libros almacenada en la base de datos de Google Books. Esta API facilita la búsqueda y recuperación de información detallada sobre los libros, obteniendo datos como el título, el autor o la portada (Google, 2024).

Dicha API utiliza una estructura RESTful para interactuar con el servicio, por lo que se pueden realizar solicitudes HTTP para buscar los libros deseados a través de cualquiera de sus datos, obteniendo como respuesta un JSON con la información de la consulta.

En el código, se da la posibilidad de buscar el libro por título o por ISBN, aunque nos centramos en la primera opción al realizar la búsqueda. Una vez especificada la información a buscar y la ruta donde se deben buscar los libros, se realiza una solicitud GET de la cual se obtiene una respuesta que se transforma en formato json.

```

base_url = 'https://www.googleapis.com/books/v1/volumes'
params = {}
if isbn:
    params['q'] = f'isbn:{isbn}'
elif title:
    params['q'] = f'intitle:{title}'
else:
    raise ValueError("La ISBN o el título deben ser proporcionados.")
params['key'] = 'ClaveAPIPrueba123ClaveSegura45DeEjemplo'

response = requests.get(base_url, params=params)
response.raise_for_status()
data = response.json()

```

Se obtiene el primer libro y se procede a extraer toda su información.

```

book = items[0]['volumeInfo']
title = book.get('title', 'Desconocido')
author = ', '.join(book.get('authors', ['Desconocido']))
#...
# Obtener ISBN
industry_identifiers = book.get('industryIdentifiers', [])
isbn = 'N/A'
for identifier in industry_identifiers:
    if identifier['type'] in ['ISBN_10', 'ISBN_13']:
        isbn = identifier['identifier']
        break

```

Por último, se compila toda la información en un diccionario y se devuelve en la función.

```

book_info = {
    'title': title,
    'author': author,
    'isbn': isbn,
    'publication_info': publication_info,
    'description': description,
    'page_count': page_count,
    'language': language,
    'thumbnail': thumbnail
}

```

Manejo de errores

El sistema está diseñado para manejar errores de forma eficiente, aportando mensajes de estado en caso de error.

Respecto a las portadas, en algunos casos no es posible recuperar la imagen de algún libro, por lo que se muestra en su lugar una imagen por defecto. Lo mismo sucede con el resto de las propiedades del libro como la descripción o el autor, si no se consigue obtener información al respecto, se indica con un mensaje.

5.2 Foro

Administración de Foros y Categorías

Los usuarios pueden acceder a una página que muestra todos los foros disponibles en el sistema. Cada foro incluye como detalles el nombre, una breve descripción y la categoría a la que pertenece.

Estos foros pueden ser agregados por los usuarios mediante un formulario donde se pedirán los datos anteriormente mencionados, además de poder crear sus propias categorías con las que clasificar los foros.

```
if request.method == 'POST':
    name = request.form.get('name')
    description = request.form.get('description', '')
    category_id = request.form.get('category_id')
    category_id = int(category_id)
    forum = ub.Forum(name=name, description=description,
category_id=category_id)
    ub.session.add(forum)
    ub.session.commit()
    flash('¡Foro agregado con éxito!', 'success')
```

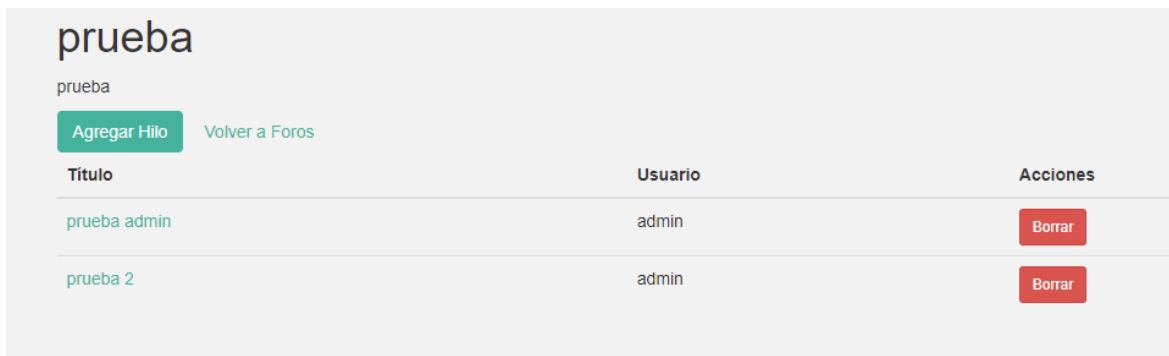
Permisos de Usuario

Dado que los usuarios tienen la posibilidad de crear foros, categorías, hilos y comentarios en cualquier momento, es necesaria la función del moderador. Para ello, existen ciertos usuarios que cuentan con el rol de administrador, el cual tiene la capacidad de editar o eliminar los foros y categorías en caso de ser necesario. Cualquier intento de acceso no autorizado a dichas operaciones, son redirigidos con un mensaje de error.

```
if not current_user.role_admin():
    flash('¡Acceso no autorizado!', 'danger')
    return redirect(url_for('manage_forums'))
```

Administración de Hilos y Comentarios

Como se puede observar en la Figura 6, al acceder a cada foro, los usuarios pueden ver una lista de todos los hilos pertenecientes a él. En cada uno, se puede apreciar el título y el usuario que comenzó dicho hilo.



Titulo	Usuario	Acciones
prueba admin	admin	Borrar
prueba 2	admin	Borrar

Figura 6. Interfaz de visualización de los hilos dentro de un foro. Las acciones únicamente son visibles para los administradores.

Todos los usuarios pueden agregar nuevos hilos, para lo que tendrán que aportar un título y escribir el contenido del primer comentario.

Cuando se accede a un hilo, se muestra un listado de comentarios donde se indica el autor, la fecha y hora de publicación y el contenido del mensaje. Cualquier usuario puede responder a los comentarios, participando así en el debate del hilo correspondiente. Esto se muestra con mayor detalle en la Figura 7.



Usuario	Fecha y Hora	Contenido	Acciones
admin	2024-08-05 19:29:20	hola	Borrar, Responder
lucia	2024-08-05 19:29:43	buenas	Borrar, Responder

Figura 7. Interfaz de visualización de los comentarios dentro de un hilo. Cada comentario se estructura en un contenedor individual para diferenciarse del resto.

Eliminación de Objetos

Es posible la eliminación, por parte de los moderadores, de categorías que estén siendo usadas en algún foro. Para ello, se actualizarán automáticamente todos los foros que usen dicha categoría, obteniendo en su lugar una categoría por defecto denominada “Sin categoría”. Esta se crea únicamente si alguna categoría es eliminada ocasionando que uno o más foros pierdan dicha calificación, esto es posible visualizarlo en la Figura 8. La categoría por defecto puede ser también borrada si no se le está dando ningún uso, pero, en caso de intentar borrarla mientras se está utilizando en algún foro, se volvería a crear de forma automática.

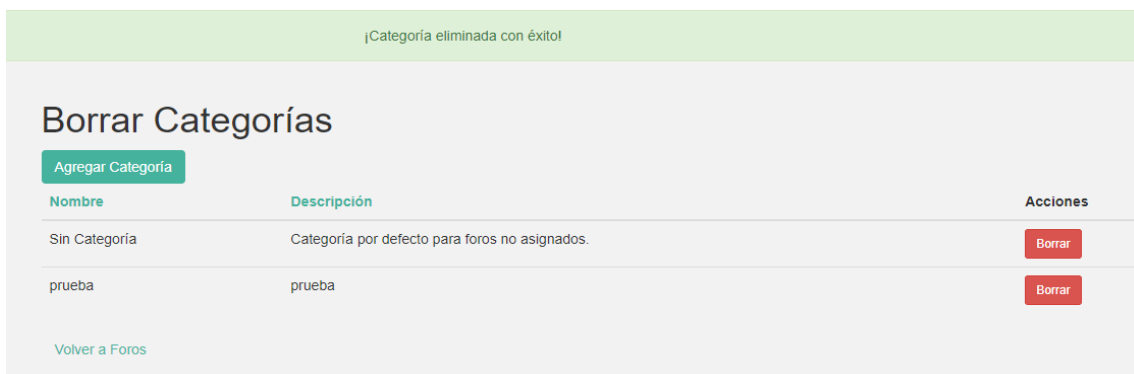


Figura 8. Interfaz de la aplicación para borrar categorías. En la misma, se puede acceder a la sección de Agregar Categorías. Se muestra la creación de la categoría por defecto tras borrar una categoría asignada a un foro.

Para el resto de los objetos, el administrador dispone de un botón que le permite borrar foros, hilos y comentarios. En el caso de los foros e hilos, puede visualizar un apartado “Acciones” invisible para los demás usuarios.

En caso de eliminar un foro o hilo con contenido en su interior, todos los hilos y/o comentarios asociados serían igualmente eliminados.

En la Figura 9, se presenta la visualización de los foros para los usuarios administradores, con el campo “Acciones” visible.

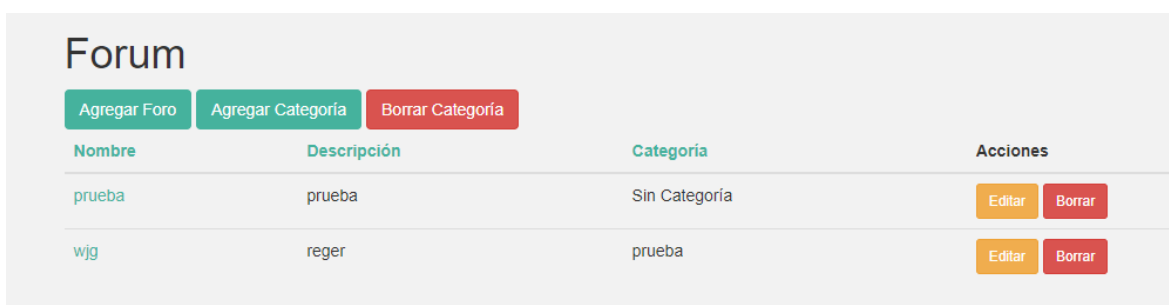


Figura 9. Interfaz para visualizar los foros. Se aprecian las acciones para agregar y eliminar foros o categorías, además de la opción para editar los foros.

Gestión de Campos no asignados en Formularios

En todos los formularios de creación o edición de objetos, existen campos que deben estar completos. En caso contrario, se le indicará al usuario la presencia de un campo no asignado y no podrá continuar hasta que lo complete, como se avisa en la Figura 10.

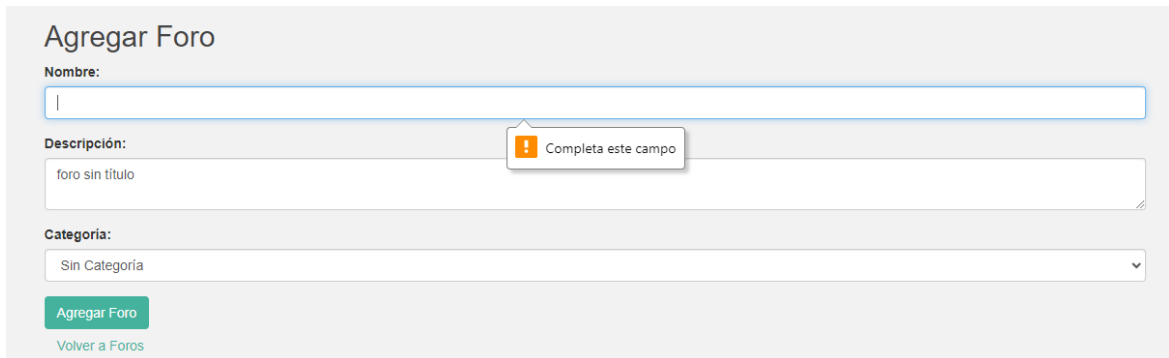
The image shows a web form titled "Agregar Foro". It has three main input fields: "Nombre:" (a text box), "Descripción:" (a text area containing "foro sin título"), and "Categoría:" (a dropdown menu with "Sin Categoría" selected). Below these fields are two buttons: "Agregar Foro" (green) and "Volver a Foros" (blue). A red error message box with a white exclamation mark icon is positioned over the "Descripción:" field, containing the text "Completa este campo".

Figura 10. Aviso de campo no completado al intentar agregar un foro sin título.

Todas estas funcionalidades permiten gestionar correctamente los elementos del foro y proporcionan una mejor experiencia para los usuarios.

5.3 Red social

Sistema de Seguimiento

El sistema de seguimiento permite a los usuarios seguir a otros, facilitando la interacción en la comunidad. Esta funcionalidad es fundamental para crear conexiones dentro de la plataforma.

Cuando un usuario sigue a otro, se registra en la base de datos dicha relación a través de una tabla de asociación. De forma similar, cuando el usuario deja de seguir a otros usuarios, se elimina dicha relación de seguimiento de la base de datos.

```
def follow(username):
    user = ub.session.query(ub.User).filter(ub.User.name
    == username).first()
    if user is None:
        flash('Usuario {} no
    encontrado.'.format(username))
        return redirect(url_for('web.index'))
    if user == current_user:
        flash(';No puedes seguirte a ti mismo!')
```

```

return redirect(url_for('user_profile',
username=username))

current_user.follow(user)
ub.session.commit()
# ...
flash(';Ahora sigues a {}'.format(username))

```

Perfil de Usuario

Cada usuario cuenta con un perfil visible para el resto donde se muestran datos como los libros descargados del usuario, las estanterías que tiene y las participaciones que ha tenido en los foros. En la Figura 11 se presenta la interfaz del perfil de usuario.

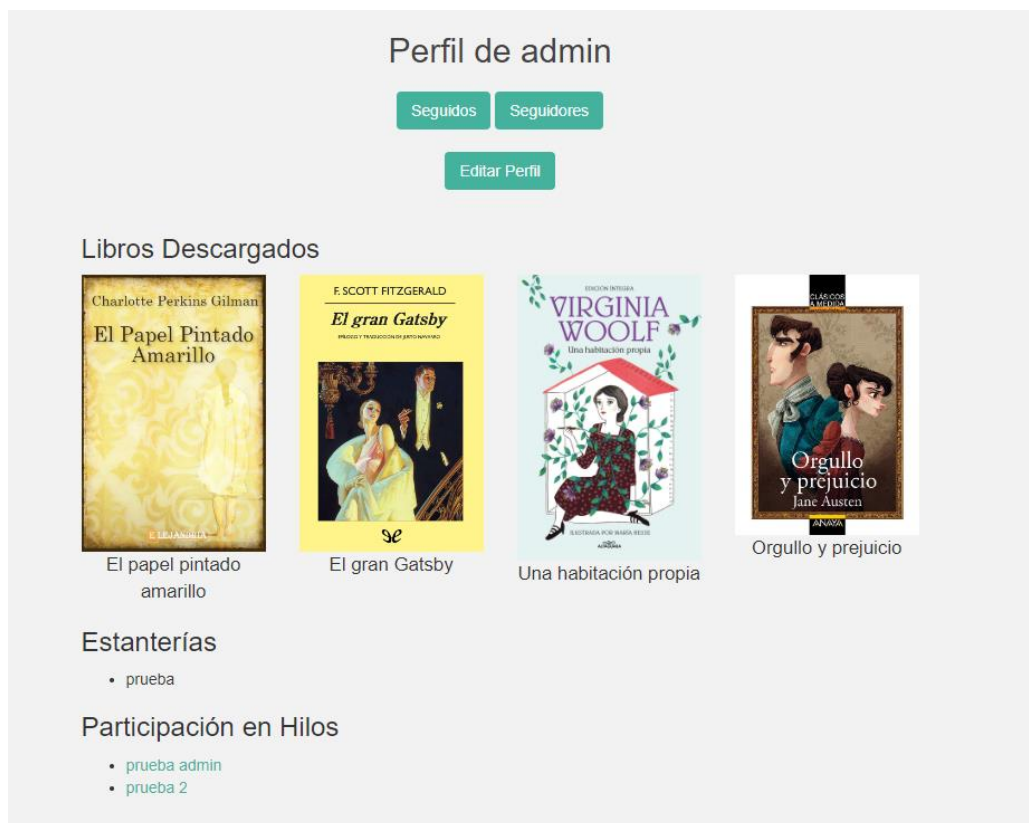


Figura 11. Perfil donde se muestran varios datos relevantes del usuario.

Quando se visita el perfil de otro usuario, es también posible ver sus seguidores y los usuarios que le siguen. Estos datos se muestran en listas donde aparece el nombre de cada usuario, pudiendo acceder a su perfil a través de su nombre, y está acompañado por un botón para seguir o dejar de seguir a dicho usuario, dependiendo de la opción que estuviera activa

para el usuario actual. Esta lista se observa en la Figura 12, mostrando los individuos que el usuario objetivo.



Figura 12. Listado de usuarios al entrar en la interfaz de usuarios que sigues (Seguidos).

Búsqueda de Usuarios

Se ha añadido en el perfil de usuario un botón que permite buscar otros usuarios en la plataforma. Esta opción facilita la navegación entre perfiles, ofreciendo la posibilidad de explorar nuevas conexiones.

```
def search():
    query = request.args.get('q', '')
    if query:
        users =
ub.session.query(ub.User).filter(ub.User.name.like(f'{quer
y}%')).all()
    else:
        users = []
        return render_title_template('searchFollow.html',
users=users, query=query, title="Search profiles",
page='search Profiles')
```

Permite buscar perfiles haciendo uso de su nombre de usuario como se muestra en la Figura 13. Al encontrar un perfil, es posible seguir a dicho usuario y acceder a su perfil haciendo clic en su nombre.



Figura 13. Barra de búsqueda de usuarios mostrando el resultado tras una búsqueda.

Mensajería

Se ha implementado un sistema de mensajería directa que permite a los usuarios enviar mensajes privados a otros desde su perfil. Esta característica facilita la comunicación dentro de la plataforma entre usuarios.

```
def chat(user_id):
    # Obtener el usuario con el que se quiere chatear
    other_user = ub.session.query(ub.User).get(user_id)
    if not other_user or other_user.id ==
current_user.id:
        flash('Usuario no encontrado o no puedes chatear
contigo mismo.', 'danger')
        return redirect(url_for('user_profile',
user_id=current_user.id))

    if request.method == 'POST':
        content = request.form.get('content')
        if content:
            message = ub.Message(sender_id=current_user.id,
receiver_id=other_user.id, content=content)
            ub.session.add(message)
            # ...
            ub.session.commit()
            flash('¡Mensaje enviado!', 'success')
            return redirect(url_for('chat',
user_id=user_id))

    # Obtener mensajes entre el usuario actual y el
usuario con el que se chatea
    messages = ub.session.query(ub.Message).filter(
```

```

        ((ub.Message.sender_id == current_user.id) &
 (ub.Message.receiver_id == other_user.id)) |
        ((ub.Message.sender_id == other_user.id) &
 (ub.Message.receiver_id == current_user.id))
        ).order_by(ub.Message.timestamp).all()

        return render_title_template('chat.html',
user=other_user, messages=messages, title="Chat",
page='chat')

```

Cuando un usuario envía un mensaje a otro, se crea un nuevo registro en la base de datos que contiene la información del remitente, el destinatario y el contenido del mensaje. Los mensajes enviados contienen el nombre de usuario del remitente y el contenido. Cada usuario tiene además la opción de borrar sus mensajes en caso de mensajes enviados por error o para mantener la privacidad. En la Figura 14, se puede observar la interfaz del *Chat* con un usuario.

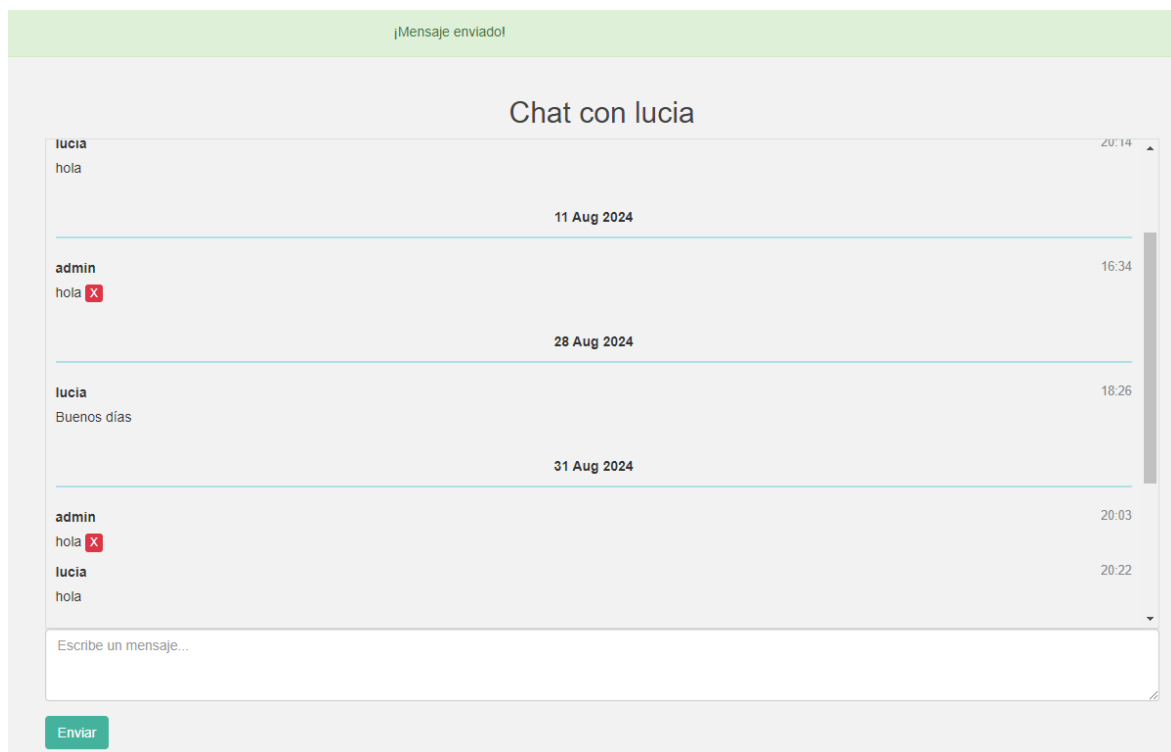


Figura 14. Sistema de mensajería con otro usuario. Se muestra la fecha y hora del mensaje así como su contenido y el nombre del usuario que escribió. Aparece también el botón “X” que permite a un usuario borrar sus mensajes.

Notificaciones

Las notificaciones se generan automáticamente en el sistema cuando ocurre un evento relevante de un usuario seguido. Las notificaciones existentes son de tres tipos: un usuario

empieza a seguir al usuario actual, se recibe un mensaje de otro usuario o uno de los usuarios seguidos publica un mensaje en los foros. Estas notificaciones son visibles a través de una interfaz donde se muestra una lista de eventos recientes relacionados.

```
# Notificaciones de seguimiento
notification = ub.Notification(user_id=user.id,
message=f"{current_user.name} ha comenzado a seguirte.",)
ub.session.add(notification)
ub.session.commit()

# Notificaciones de publicación en foro
followers =
ub.session.query(ub.User).join(ub.UserFollow, ub.User.id
==
ub.UserFollow.follower_id).filter(ub.UserFollow.followed_id
== user_id).all()
for follower in followers:
    notification = ub.Notification(user_id=follower.id,
message=f'El usuario {current_user.name} ha publicado en
el foro.', post_id=post.id)
    ub.session.add(notification)

# Notificaciones de mensaje recibido
notification_message = f"Has recibido un mensaje de
{current_user.name}"
notification = ub.Notification(user_id=other_user.id,
message=notification_message)
ub.session.add(notification)
```

Cada notificación dispone de un enlace que permite acceder al contenido sobre el cual el usuario ha recibido dicho aviso, facilitando la navegación en lugar de tener que buscarlo manualmente.

Con el fin de no tener notificaciones innecesarias, se ha agregado la posibilidad de eliminarlas de forma individual o borrar todas las existentes, como se muestra en la Figura 15.

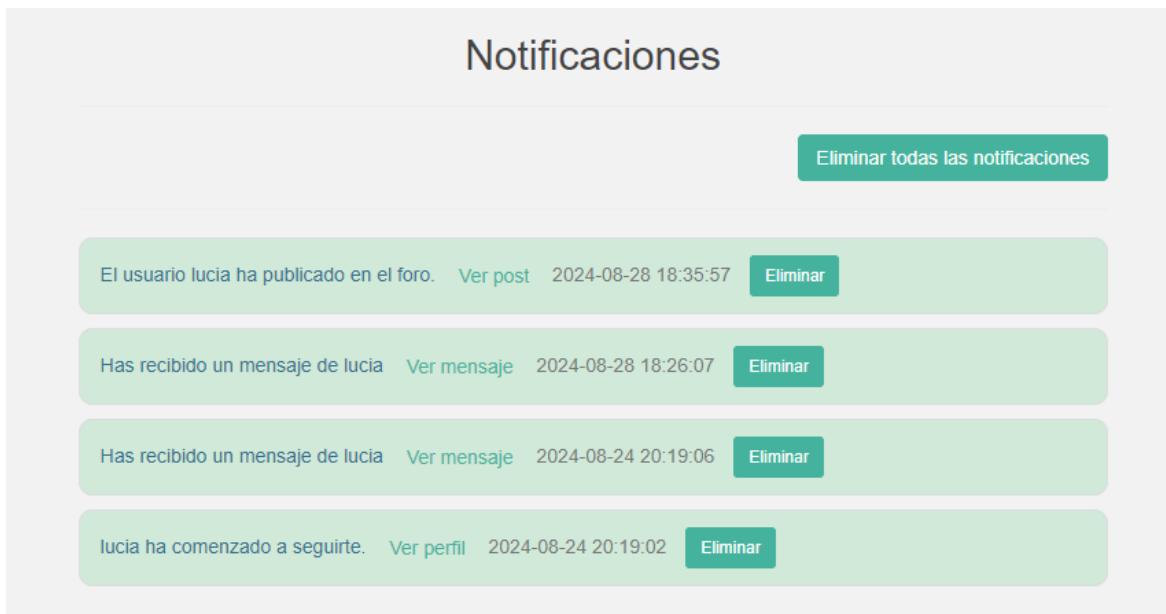


Figura 15. Visualización del sistema notificaciones, mostrando todos los tipos posibles.

5.4 Lectura en formato audio

Se ha implementado una funcionalidad de lectura de libros en formato de audio, aportando al usuario una experiencia más inmersiva. En la Figura 16, aparece la interfaz del lector digital, el cual ya estaba presente en la aplicación, pero se muestran además los elementos agregados para la función de audio en el lector. Es posible ver una barra de progreso en la parte inferior de la pantalla, así como el tiempo actual de lectura (que aparece a la izquierda en la barra) y el tiempo total del capítulo (mostrado a la derecha en la barra).

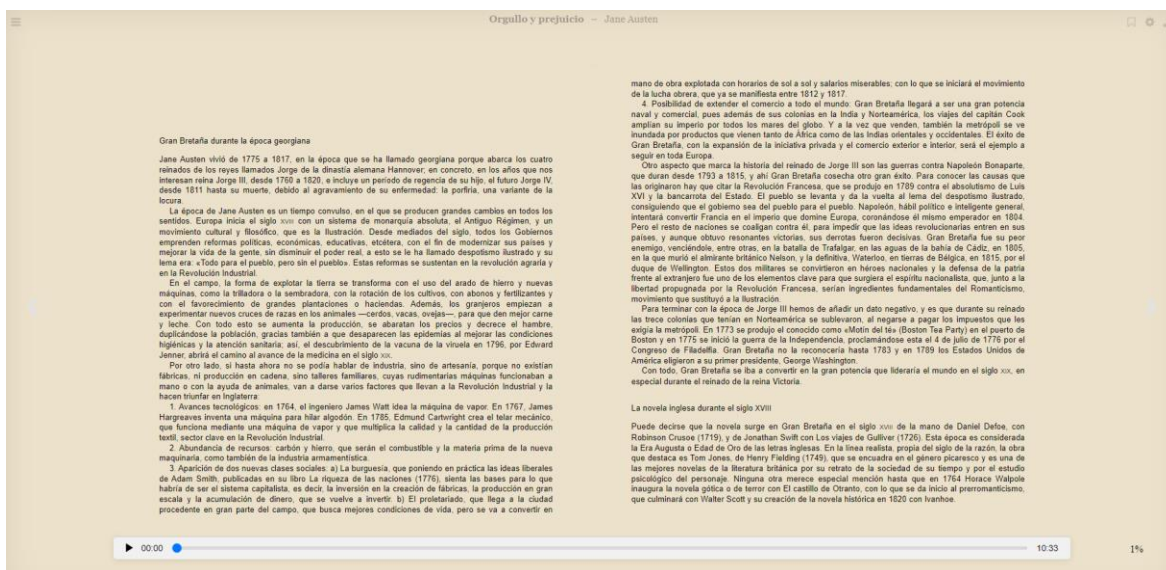


Figura 16. Interfaz del lector digital mostrando la barra de progreso del audio incorporada con indicaciones del tiempo actual y tiempo total de lectura.

Reproducción del texto

La funcionalidad principal del audio consiste en un sistema capaz de leer en voz alta el contenido disponible en la página. Esta lectura inicia al pulsar el botón de *Play*. El sistema utiliza la API 'SpeechSynthesis' para convertir el texto a voz, proporcionando una síntesis de voz directamente desde la web, sin necesidad de instalar software adicional. La API se encarga de tomar una cadena de texto, sintetizarla en voz y reproducirla a través del dispositivo de audio (Bidelman, 2020; Wang, 2023).

Uno de los objetos clave es 'SpeechSynthesis', que recoge el texto a transformar. Cuando se crea una instancia de este tipo, se puede definir el texto a leer y otros parámetros como el idioma, la velocidad o el volumen.

```
utterance = new
SpeechSynthesisUtterance(textToRead.substring(currentTextI
ndex));
utterance.lang = "es-ES";
utterance.rate = 1;
```

Cuando el objeto ha sido creado, se puede llamar al método 'speak' con dicho objeto para iniciar la lectura.

```
synth.speak(utterance);
```

Botones de Control

La lectura de voz cuenta con un botón para pausar y reanudar la lectura. Esto es posible cancelando la reproducción y almacenando el momento donde se pausó la lectura.

El método 'pauseReading' se encarga de cancelar la reproducción, esto lo hace alternando el estado entre 'isSpeaking' y 'isPaused'. Si la lectura está en curso y el usuario la pausa, se detiene y se guarda el momento actual.

```
function pauseReading() {
  if (isSpeaking && !isPaused) {
    synth.cancel();
    isPaused = true;
    isSpeaking = false;
    document.getElementById("play-button").style.display
= "block";
    document.getElementById("pause-button").style.display
= "none";
```

```

    pauseProgress();
  } else if (isPaused) {
    startReading();
  }
}

```

En caso contrario, si el usuario quiere reanudar la lectura, se llama a 'startReading', que continúa desde el punto guardado en la variable 'currentTextIndex'. Esta se actualiza con 'utterance.onboundary' cada vez que encuentra una nueva palabra, para poder guardar constantemente la posición actual de lectura. Cuando finaliza un capítulo, se recurre al evento 'onend', que reinicia las variables de estado y el índice de lectura (MDN, 2018).

```

function startReading() {
  //...
  utterance = new
SpeechSynthesisUtterance(textToRead.substring(currentTextI
ndex));
  utterance.lang = "es-ES";
  utterance.rate = 1;
  utterance.onboundary = function(event) {
    if (event.name === 'word') {
      currentTextIndex = event.charIndex;
    }
  };

  utterance.onend = function () {
    isSpeaking = false;
    isPaused = false;
    currentTextIndex = 0;
    document.getElementById("play-button").style.display
= "block";
    document.getElementById("pause-button").style.display
= "none";
  };
  synth.speak(utterance);
  startProgress();
}

```

Dado que el botón de *Play* y Pausa son el mismo, esta función también realiza el cambio de estilo para mostrar en cada momento el botón adecuado, teniendo en cuenta si el audio está reproduciéndose o si está en pausa.

Visualización del Progreso

Para complementar el proceso de la lectura, se ha añadido una barra de progreso y un temporizador que indica el avance del usuario. Esta función mejora la experiencia del usuario al permitirle ver el tiempo transcurrido y cuánto queda por leer en cada capítulo.

La función 'updateProgress' se encarga de actualizar el progreso de lectura. Esta función incrementa la variable 'currentTime' cada segundo y actualiza la barra de progreso y el temporizador en función del tiempo total y el tiempo transcurrido.

```
function updateProgress() {
    if (currentTime < totalTime) {
        currentTime += 1;
        document.getElementById('current-time').textContent
= formatTime(currentTime);
        document.getElementById('progress-bar').value
= (currentTime / totalTime) * 100;
    } else {
        clearInterval(interval);
    }
}
```

Otros métodos que sirven para controlar el progreso de la barra son 'startProgress' y 'pauseProgress', que inicia o detiene el progreso.

Control del Tiempo de Lectura

El usuario puede ajustar el tiempo de lectura desde la propia barra de progreso, pudiendo avanzar o retroceder a cualquier punto. Esta función actualiza también el punto de lectura actual, permitiendo que la lectura continúe desde el punto deseado.

Al interactuar con la barra, la función 'moveToTime' calcula el nuevo tiempo, actualizando el tiempo actual, el punto de lectura actual ('currentTextIndex'), y ajustando la barra a la nueva posición.

```
function moveToTime(newTime) {
    currentTime = newTime;
    document.getElementById('current-time').textContent
= formatTime(currentTime);
    document.getElementById('progress-bar').value
= (currentTime / totalTime) * 100;
}
```

```

    let charIndex = Math.floor((currentTime / totalTime)
* textToRead.length);
    currentTextIndex = charIndex;

    if (isSpeaking) {
        synth.cancel();
        startReading();
    }
}

```

Cálculo del Tiempo Total

El sistema calcula el tiempo total necesario para leer un capítulo completo en función de la velocidad de lectura y la cantidad de palabras contenidas en el texto. Esta funcionalidad da al usuario una estimación precisa del tiempo requerido para escuchar dicho capítulo.

De esto se encarga el método 'calculateTotalTime', que estima el tiempo total de lectura tomando como promedio una velocidad de 200 palabras por minuto. Este cálculo se realiza antes de comenzar la lectura, y se refleja en la interfaz mostrando el tiempo total de lectura del contenido.

```

function calculateTotalTime(text) {
    const wordsPerMinute = 200;
    const wordCount = text.split(/\s+/).length;
    return Math.ceil((wordCount / wordsPerMinute) * 60);
}

```

La implementación de estas funcionalidades mejora la accesibilidad para usuarios que quieran o necesiten escuchar el contenido en lugar de leerlo, ofreciendo una experiencia más interactiva, y ampliando el rango de usuarios.

6

Pruebas

En este capítulo se detalla el proceso de pruebas llevado a cabo para asegurar la funcionalidad y robustez de la aplicación desarrollada. Las pruebas son fundamentales para verificar que el software cumpla con los requisitos especificados y para identificar posibles errores o áreas donde mejorar.

6.1 Configuración para la Realización de Pruebas

Para el correcto desarrollo de las pruebas automatizadas, es necesario preparar el entorno. Se ha seguido para ello un proceso de configuración detallado.

Instalación del driver

Se ha instalado el driver ChromeDriver, una herramienta esencial para que Selenium pueda interactuar con el navegador Google Chrome. ChromeDriver actúa como un puente entre Selenium y el navegador, permitiendo la automatización de las acciones de prueba.

Configuración de Selenium en Google Chrome

Dado que se ha empleado el navegador Google Chrome para ejecutar las pruebas, este se debe configurar también. Se ha añadido la extensión de Selenium para facilitar la ejecución de las pruebas automatizadas. Esta extensión permite a Selenium controlar el navegador, realizando acciones como clics y desplazamientos, y pudiendo obtener información del contenido de las páginas web.

Instalación de Selenium en el entorno

Se ha instalado el paquete Selenium en el entorno de desarrollo, permitiendo la automatización de navegadores web mediante scripts. También se han configurado otros paquetes y dependencias necesarios para ejecutar las pruebas correctamente.

6.2 Pruebas Realizadas

Durante el proceso, se han llevado a cabo varias pruebas para evaluar diferentes aspectos del sistema. Para todos los *test* es necesario el inicio de sesión al principio, puesto que todas las características implementadas requieren la identificación del usuario para acceder a las mismas.

Las siguientes pruebas fueron realizadas de forma automatizada para verificar el correcto funcionamiento de distintas características en la aplicación. Cada prueba cubre un aspecto clave del sistema, desde la recomendación de libros hasta la interacción con el foro o la reproducción de audio. Estas pruebas se agrupan en cinco categorías y han sido diseñadas para asegurar que las funcionalidades operen correctamente bajo diversas condiciones. A continuación, se describen las pruebas realizadas para cada módulo nuevo añadido.

Pruebas del Recomendador

Se ha probado la capacidad del sistema para navegar correctamente a través de diferentes páginas y secciones, asegurando que los enlaces y botones funcionen como se espera. A continuación, se describen los *tests* dentro de esta clasificación.

Nombre	Ejecución normal	Ejecución alternativa
test_recomendador	Verifica la funcionalidad del sistema de recomendación de libros, asegurando que genere sugerencias basadas en las preferencias del usuario y recupere detalles de estos a través de la API de Google Books	En caso de que ya se estén mostrando recomendaciones obtenidas previamente, se permite volver a realizar el <i>test</i>

Cuadro 13. Pruebas del Recomendador

Pruebas del Foro

Esta sección reúne todas las pruebas que han sido generadas para el foro, agrupando la creación y eliminación de sus elementos, además de la edición para los foros, asegurando que las interacciones en los foros se realicen sin problemas.

Nombre	Ejecución normal	Ejecución alternativa
test_agregar_foro	Permite la creación de un nuevo foro por parte del usuario, validando que se añada correctamente	En caso de intentar agregar un foro con el nombre de otro ya existente, aparece un aviso y se añade una cadena de caracteres aleatoria al final del nombre
test_borrar_foro	Verifica que los usuarios administradores puedan borrar un foro	Si no existen filas suficientes, no se podrá borrar el foro, generando una excepción
test_editar_foro	Comprueba que la información del foro pueda ser editada, actualizando sus detalles correctamente	Al intentar cambiar el nombre, si se intenta poner uno ya existente, se avisa de ello y se añade una cadena de caracteres aleatoria al final del nombre
test_agregar_categoria	Testea la creación de nuevas categorías para organizar los foros	Si se intenta agregar una categoría con un nombre existente, se avisa de ello y se añade una cadena de caracteres aleatoria al final del nombre
test_borrar_categoria	Valida la correcta eliminación de categorías de foros	En caso de eliminar una categoría que esté en uso, se elimina y se le asigna a los foros que la estuvieran usando una categoría "Sin categoría"
test_agregar_hilo	Asegura que se pueden crear nuevos hilos dentro de un foro	
test_borrar_hilo	Valida la eliminación de hilos de discusión de manera correcta y sin dejar referencias a comentarios	Si no hay hilos suficientes, no es posible la eliminación, por lo que aparecería una excepción

	internos	
test_agregar_comentario	Prueba la adición de comentarios en hilos, asegurando que los usuarios puedan interactuar	Si un hilo no tiene comentarios, no es posible responder a otros usuarios, generando una excepción
test_borrar_comentario	Verifica que los comentarios puedan ser eliminados por el administrador del foro	Si no existen comentarios en el hilo, no se podrá eliminar, resultando en una excepción

Cuadro 14. Pruebas del Foro

A continuación, se muestra un *test* de ejemplo de las pruebas automatizadas creadas para verificar el correcto funcionamiento de cada función dentro de la aplicación web. En este caso, el *test* sigue la estructura AAA (*Arrange, Act, Assert*), que es un patrón común para la escritura de pruebas. La estructura AAA organiza el código de la siguiente manera: en la fase de *Arrange*, se configura el estado inicial y se preparan los datos necesarios; en la fase de *Act*, se ejecuta la acción que se desea probar; y en la fase de *Assert*, se verifica que los resultados sean los esperados.

En el código, se puede observar cómo, en la fase *Arrange*, se prepara al sistema para estar en la posición correcta para iniciar el *test*. Para ello, se inicia sesión y se genera el nombre del foro.

En la fase *Act*, se realizan todas las acciones de acceder al foro, agregar uno nuevo y rellenar el formulario con datos de prueba.

Por último, en la fase *Assert*, se espera un tiempo hasta que se muestra el mensaje de alerta confirmando la creación del foro. Luego, se verifica que el mensaje recibido contenga el texto esperado, y se confirma así que la acción fue realizada correctamente.

```
def test_agregarforo(self):
    # Arrange
    self.driver.get("http://localhost:8083/login?next=%2F")
    self.driver.set_window_size(974, 1040)
    self.driver.find_element(By.ID, "username").click()
```

```

        self.driver.find_element(By.ID,
"username").send_keys("admin")
        self.driver.find_element(By.ID, "password").click()
        self.driver.find_element(By.ID,
"password").send_keys("admin123")
        self.driver.find_element(By.NAME, "submit").click()

        unique_name = f"foro prueba
{self.generate_unique_suffix()}"

        # Act
        self.driver.get("http://localhost:8083/")
        self.driver.set_window_size(974, 1031)
        self.driver.find_element(By.LINK_TEXT, "Foro").click()
        self.driver.find_element(By.LINK_TEXT, "Agregar
Foro").click()
        self.driver.find_element(By.ID, "name").click()
        self.driver.find_element(By.ID,
"name").send_keys(unique_name)
        self.driver.find_element(By.ID,
"description").send_keys("Descripción del foro prueba")
        self.driver.find_element(By.ID, "category_id").click()
        dropdown = self.driver.find_element(By.ID,
"category_id")
        dropdown.find_element(By.XPATH, "//option[. =
'prueba']").click()
        self.driver.find_element(By.CSS_SELECTOR, ".btn-
primary").click()

        # Assert
        WebDriverWait(self.driver, 10).until(
            EC.presence_of_element_located((By.CSS_SELECTOR,
".alert-success")))
        )
        success_message =
self.driver.find_element(By.CSS_SELECTOR, ".alert-
success")
        assert ";Foro agregado con éxito!" in
success_message.text

```

Es posible que el nombre aportado ya exista en la lista de foros, por tanto, la prueba se encarga de añadir al nombre junto a una cadena de caracteres elegidos de forma aleatoria en el método 'generate_unique_suffix'.

```
def generate_unique_suffix(self):
    return ''.join(random.choices(string.ascii_letters +
string.digits, k=10))
```

Pruebas de Usuario

En esta categoría, se examinan las funcionalidades que permiten a los usuarios interactuar entre sí. Estas pruebas incluyen la búsqueda de otros usuarios, la capacidad de seguir o dejar de seguir a alguien, así como el envío y gestión de mensajes privados.

Nombre	Ejecución normal	Ejecución alternativa
test_buscar_usuario	Verifica que la búsqueda de usuarios funciona adecuadamente, devolviendo como resultado los usuarios obtenidos según los criterios de búsqueda	Si no se encuentran usuarios con los criterios de búsqueda indicados, se espera recibir un mensaje notificando la situación
test_dejar_de_seguir	Asegura que un usuario puede dejar de seguir a otro, eliminando correctamente la relación entre ambos	Si no se encuentra el usuario en el apartado "Seguidos" el <i>test</i> lanza una excepción
test_seguir	Valida que los usuarios pueden seguir a otros, actualizando la lista de seguidos de manera efectiva	Si encuentra el botón "Dejar de seguir",
test_ver_seguidores	Comprueba que el usuario pueda visualizar una lista de sus seguidores	Si el usuario actual no tiene seguidores, se muestra un mensaje indicándolo

test_ver_seguidos	Asegura que el usuario pueda ver a las personas que sigue, mostrándolas en la lista de seguidos	Si el usuario no sigue a nadie, esto se especifica en un mensaje en lugar de la lista de seguidos
test_mandar_mensaje	Valida la funcionalidad para enviar mensajes privados entre usuarios, asegurando una correcta comunicación	Si no se envió el mensaje correctamente, se lanza una excepción indicando este fallo
test_mandar_y_borrar_mensaje	Automatiza el proceso de enviar un mensaje a otro usuario, comprobando su éxito, para poder eliminarlo posteriormente, verificando también su correcta eliminación	En caso de no enviarse correctamente el mensaje, si no existen otros mensajes en el <i>chat</i> , aparecería una excepción indicando la imposibilidad de borrar un mensaje

Cuadro 15. Pruebas de Usuario

Pruebas de Notificación

En esta sección, se describirán algunos de las pruebas relacionadas con las notificaciones, mostrando su buen funcionamiento desde cualquiera de sus facetas.

Nombre	Ejecución normal	Ejecución alternativa
test_notificacion_ver_mensaje	Asegura que los usuarios puedan acceder al <i>chat</i> con otro usuario a través de las notificaciones de mensajes directos	
test_notificacion_ver_perfil	Comprueba que el <i>link</i> de	

	las notificaciones de nuevos seguidores permita visitar el perfil del seguidor	
test_notificacion_ver_post	Valida que el acceso al <i>post</i> publicado por un usuario seguido sea accesible a través de las notificaciones	En caso de que se borre el <i>post</i> mencionado en la notificación, el link quedaría innacesible, provocando una excepción
test_borrar_una_notificacion	Asegura que los usuarios puedan eliminar una notificación específica sin errores	Si no hay notificaciones disponibles, se genera una excepción
test_borrar_notificaciones	Verifica la funcionalidad de eliminar todas las notificaciones juntas	Si no hay ninguna notificación, verifica que el mensaje de error se muestre

Cuadro 16. Pruebas de Notificación

Pruebas de Reproducción de Audio

Se ha verificado que el sistema pueda reproducir audio correctamente, que la barra de progreso y el temporizador se actualicen en función del tiempo transcurrido, y que el sistema responda adecuadamente a las acciones del usuario, como avanzar o retroceder en el audio, además de pausar y reanudar la lectura. Se verifica así la interacción con los controles de reproducción, garantizando una experiencia fluida.

Nombre	Ejecución normal	Ejecución alternativa
test_escuchar_y_parar_libro	Verifica que los usuarios puedan iniciar y detener la reproducción	En caso de que los botones de control no respondan

	de audiolibros sin problemas	correctamente, se produce una excepción
test_avanzar_audio	Asegura que los usuarios puedan avanzar en la reproducción del audio asegurando que la funcionalidad responda correctamente	Si la barra de progreso no responde, se lanza una excepción indicando este mal funcionamiento
test_cambiar_capitulo	Valida la capacidad de recalcularse el tiempo total de lectura al cambiar de capítulo, comparando el tiempo inicial y el tiempo del nuevo capítulo	Si el tiempo total no se recalcula o alguna funcionalidad intermedia necesaria no se ejecuta con éxito, se genera una excepción

Cuadro 17. Pruebas de Reproducción de Audio

Dado que para la lectura del libro se debe acceder a una pestaña nueva en el navegador, se hace uso del método 'wait_for_window', método que espera que se abra una nueva ventana y luego devuelve su identificador.

```
def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
        return set(wh_now).difference(set(wh_then)).pop()
```

Los resultados obtenidos tras realizar las pruebas han confirmado el correcto funcionamiento de las características implementadas para cada componente, asegurando que el sistema cumpla con los requisitos establecidos y que se encuentra libre de errores críticos. La correcta ejecución de estas pruebas valida la estabilidad y eficacia del sistema en su conjunto.

7

Conclusiones y Líneas Futuras

En este capítulo se presentan las conclusiones derivadas del trabajo realizado, evaluando el cumplimiento de los objetivos propuestos y resaltando los logros obtenidos. Se analiza el impacto del proyecto en el contexto del problema abordado, considerando tanto sus fortalezas como sus debilidades. Asimismo, se proponen líneas futuras de desarrollo que podrían continuar y expandir el trabajo realizado, abordando aspectos que no se pudieron cubrir en el presente proyecto o sugiriendo mejoras y nuevas funcionalidades para optimizar el sistema o la investigación en futuras iteraciones.

7.1 Conclusiones

Desde el principio, el tema elegido para este Trabajo de Fin de Grado se basó en una combinación de intereses personales, particularmente mi pasión por la lectura y por el desarrollo web. Quería crear una aplicación que reflejara estas dos áreas, no solo como un desafío académico, sino también como una forma de contribuir a la comunidad lectora. A lo largo del proceso, he aprendido que, aunque siempre existen oportunidades para incorporar nuevas funcionalidades que podrían hacer la aplicación aún más atractiva, lo que he logrado

hasta ahora es una plataforma bastante completa, que en su estado actual ya ofrece muchas herramientas útiles y significativas para los lectores.

El desarrollo de este proyecto no estuvo exento de desafíos; hubo momentos de dificultad que pusieron a prueba mi perseverancia. Sin embargo, cada obstáculo superado ha valido la pena, ya que me ha permitido crecer no solo como desarrolladora, sino también en mi capacidad de gestionar el tiempo y cumplir con las entregas. Este proyecto ha sido una gran oportunidad para profundizar en mis conocimientos de Python y HTML, herramientas con las que ya estaba familiarizada gracias a asignaturas cursadas en años anteriores, pero que ahora manejo con mayor soltura.

Además, me ha brindado la oportunidad de adentrarme en el mundo de JavaScript. Aunque mi experiencia previa con este lenguaje era limitada, este proyecto me ha permitido empezar a comprender mejor sus capacidades y potencial, abriendo la puerta a futuras exploraciones y aprendizajes en este ámbito.

En resumen, este proyecto ha sido una experiencia de aprendizaje significativa, que no solo me ha permitido crear una aplicación web que podría beneficiar a otros, sino que también ha fortalecido mis habilidades técnicas y personales. Estoy satisfecha con lo que he logrado y entusiasmada por las posibilidades que estos conocimientos me permitirán explorar.

7.2 Objetivos Cumplidos y Logros obtenidos

El desarrollo del sistema de recomendación personalizado cumple con el objetivo de mejorar la experiencia de los usuarios en la plataforma, facilitando el descubrimiento de libros basados en sus gustos. A través de un *test* de preferencias literarias, el sistema evalúa los intereses y hábitos de lectura de cada usuario, permitiendo generar recomendaciones precisas y personalizadas. Este cuestionario ha sido viable gracias a la integración de un algoritmo basado en el Teorema de Bayes, que además utiliza la API de Google Books para obtener información detallada de los títulos sugeridos.

La implementación del foro ha cumplido con éxito los objetivos establecidos, proporcionando un espacio interactivo donde los usuarios pueden participar activamente en debates y discusiones relacionadas con el mundo de la lectura. El foro ha facilitado la creación de hilos de discusión sobre temas diversos, como libros, autores o tendencias literarias, lo que

enriquece la experiencia de los usuarios al ofrecerles la oportunidad de compartir sus opiniones. Este componente fomenta una comunidad activa de lectores, promoviendo una mayor interacción entre los miembros de la plataforma.

La integración de la red social ha permitido cumplir con el objetivo de ofrecer un espacio donde los usuarios puedan conectarse y comunicarse. Ahora, los usuarios pueden seguirse mutuamente, enviarse mensajes privados y explorar los perfiles de otros lectores, lo que ha facilitado la interacción entre ellos. Además, los usuarios tienen la capacidad de ver las actividades recientes y las listas de libros descargados de otros, fomentando así un entorno social. Con ello, se ha logrado crear una comunidad activa basada en el interés común por la literatura.

Uno de los principales objetivos planteados para este proyecto fue habilitar la posibilidad de escuchar los libros en formato audiolibro, lo cual ha sido completamente implementado. Esta funcionalidad ofrece a los usuarios la opción de disfrutar de sus lecturas de una manera diferente, adaptándose a las necesidades de aquellos que prefieren escuchar la lectura mientras realizan otras actividades o que tienen dificultades para leer de manera tradicional. El sistema permite controlar la reproducción del contenido en audio a través de los controles integrados, con opciones de play, pausa, y avance en la barra de progreso, lo que garantiza una experiencia fluida y accesible.

Destaca la precisión y la rapidez del sistema para sincronizar el texto con el audio, sin depender de tiempos de carga que podrían resultar tediosos para los usuarios. Además, el sistema es capaz de recalcular el tiempo de lectura al cambiar de capítulo, lo que favorece la coherencia de la reproducción. Esta opción de escuchar los libros amplía considerablemente las posibilidades de uso del lector digital, ofreciendo una experiencia más inclusiva y adaptada a diferentes tipos de usuarios.

7.3 Problemas Encontrados

Uno de los principales problemas encontrados durante el desarrollo del recomendador fue la imposibilidad de depender completamente de la API de Google Books para generar sugerencias personalizadas. Además, no se pudo encontrar una fuente externa o base de datos con preferencias de usuarios que permitiera entrenar un algoritmo de recomendaciones más robusto. Como solución, se tuvo que simplificar la funcionalidad del recomendador,

utilizando una lista predefinida de libros populares para diferentes perfiles de lectores, que luego son recomendados utilizando el algoritmo basado en el teorema de Bayes.

Durante el desarrollo de la funcionalidad de audio, uno de los principales desafíos fue la implementación de una característica que permitiera a cada usuario grabar su propia voz para la lectura personalizada de los libros. La idea inicial era que cada usuario pudiera grabar unos segundos de su voz y que el sistema generara una voz de lectura basada en esas grabaciones, o que los usuarios pudieran elegir entre algunas voces ya generadas (Ábalos, 2023; Microsoft Research, 2024). Sin embargo, debido a problemas técnicos y limitaciones en la integración de tecnologías de síntesis de voz personalizada, esta funcionalidad no pudo ser implementada como se había previsto, recurriendo finalmente al uso de una voz de lectura predeterminada.

7.4 Líneas futuras

Dado que el proyecto cuenta con un gran potencial, existen diversas funcionalidades que no han podido ser desarrolladas en esta primera versión. Además, otras áreas funcionales podrían beneficiarse de mejoras sustanciales para optimizar la experiencia del usuario.

Estas funcionalidades representan oportunidades clave para ampliar el alcance del proyecto, incrementar su usabilidad y personalización, y responder de forma más activa a las necesidades de los usuarios. La evolución de estas características no solo podría enriquecer el ecosistema actual del proyecto, sino también abrir nuevas posibilidades de interacción y adaptación tecnológica que mejoren significativamente su desempeño.

7.4.1 Lectura de Audio Personalizada

La intención inicial era permitir que los usuarios grabaran unos segundos de su propia voz para generar un lector personalizado. Esto haría que la experiencia de lectura de audio fuera más natural y única para cada usuario. Se contempló el uso de tecnologías avanzadas de clonación de voz como Monoceros o VALL-E para lograr este objetivo (Monoceros Lab, 2023; Microsoft, 2024).

Otras funciones interesantes consistirían en agregar un controlador de volumen y un controlador de velocidad de audio. Con el controlador de volumen, los usuarios podrían ajustar la intensidad según sus preferencias o el entorno en que se encuentre, sin necesidad

de modificar el volumen general del dispositivo. Por otra parte, el controlador de velocidad permitiría ajustar la rapidez de reproducción del texto. Algunos usuarios pueden preferir una lectura más lenta para comprender mejor el contenido, mientras que otros pueden optar por una velocidad más rápida, permitiendo avanzar rápidamente a través de la lectura.

Otra opción que se debatió fue la creación de un feed de RSS para contener los capítulos del libro, lo que permitiría su integración en aplicaciones de podcasts. RSS, o *Really Simple Syndication*, es un formato de distribución de contenido web que permite a los usuarios suscribirse a un contenido de forma automática. Con un feed RSS para los capítulos, los usuarios podrían recibir notificaciones de nuevos capítulos en sus aplicaciones de podcast, facilitando el acceso y la gestión del contenido sin necesidad de visitar la aplicación (Mafra, 2021).

Aunque no han podido ser implementados en esta versión del proyecto, serían excelentes mejoras en el futuro, proporcionando una experiencia más personalizada y envolvente.

7.4.2 Base de Datos de Preferencias para las Recomendaciones

Otra posible mejora sería integrar el recomendador con una base de datos de preferencias y gustos de los usuarios. Actualmente, el recomendador está enlazado con la API de Google Books, de forma que es posible realizar búsquedas de libros indicando prácticamente cualquiera de sus atributos. El principal problema es que, dado que no se contaba con una base de datos con preferencias lectoras actualizada y que tuviera una amplia variedad de libros, no fue posible dicha implementación.

Con esta mejora se eliminaría la necesidad de introducir manualmente el nombre de los libros, permitiendo que el sistema sugiera automáticamente libros que podrían interesar de forma más precisa al usuario en base a las respuestas proporcionadas en el cuestionario y las ya existentes en la base de datos.

7.4.3 Notificaciones mejoradas

En esta versión, existen tres tipos de notificaciones posibles. Un objetivo futuro sería ampliar este rango para diversas acciones del usuario, de esta forma, se generaría más

interacción entre los usuarios. Algunas de estas acciones podrían ser cuando un usuario guarda un libro en su biblioteca o cuando publica la calificación del libro.

Dado que tal cantidad de notificaciones puede llegar a ser molesta para ciertos usuarios, se daría la opción de configurar qué tipo de notificaciones se desea recibir, brindando un mayor control sobre la experiencia del usuario en la plataforma.

7.4.4 Perfiles Públicos y Privados

Actualmente, los perfiles de los usuarios son todos públicos, lo que podría suponer una limitación para ciertas personas a la hora de querer participar en la comunidad. Por ello, otra importante mejora sería implementar la opción de tener perfiles privados, permitiendo a los usuarios elegir el nivel de privacidad que prefieren para su perfil.

7.4.5 Cambio de Idioma en la Lectura de Voz

Sería interesante incorporar la posibilidad de que los usuarios puedan cambiar el idioma de la lectura de voz en el lector digital. Dado que esta versión de la aplicación está orientada principalmente a un público de habla hispana, se optó por utilizar el español como idioma predeterminado en la reproducción de voz. Sin embargo, esta decisión limita la lectura de libros que no estén en español, lo que reduce el rango de opciones disponibles para los usuarios. Este cambio mejoraría la accesibilidad y adaptabilidad del sistema a usuarios que prefieren leer en diferentes idiomas.

7.4.6 Acceso de Mensajería

Ciertamente, ahora es un poco tedioso el acceso a los mensajes directos con otros usuarios. Una forma más atractiva para el usuario podría ser agregar un acceso a los mensajes a través de la barra de navegación donde se encuentran los accesos al foro o al recomendador, reuniendo así los mensajes de todos los usuarios en un mismo espacio.

Referencias

Cárdenas, L (2024). *calibre-web*. GitHub. <https://github.com/luciacp1312/calibre-web>

Calibre. (2024). *Calibre*. <https://calibre-ebook.com/es>

Janeczko, J. (2024). *Calibre-Web*. GitHub. <https://github.com/janeczku/calibre-web>

Canva. (2024). *Canva Design School*. Canva. https://www.canva.com/es_es/

Gitmind. (2024). *Gitmind: Free Mind Mapping Tool*. Gitmind. <https://gitmind.com/>

Microsoft. (2024). *Visual Studio Code Documentation*. Microsoft.

<https://code.visualstudio.com/docs>

Oracle. (2024). *Oracle SQL Developer Data Modeler*. Oracle.

<https://docs.oracle.com/en/database/>

Werkzeug. (2007). *Werkzeug Documentation (3.0.x)*. Pallets Projects.

<https://werkzeug.palletsprojects.com/en/3.0.x/>

Bootstrap. (2024). *Bootstrap: Introduction*. Bootstrap.

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Flask-Principal. (2010). *Flask-Principal Documentation (0.4.x)*. Flask-Principal.

<https://pythonhosted.org/Flask-Principal/>

Flask. (2010). *Flask Documentation (3.0.x)*. Flask.

<https://flask.palletsprojects.com/en/3.0.x/>

Flask-WTF. (2010). *Flask-WTF Documentation (1.3.x)*. Flask-WTF. [https://flask-](https://flask-wtf.readthedocs.io/en/1.2.x/quickstart/)

[wtf.readthedocs.io/en/1.2.x/quickstart/](https://flask-wtf.readthedocs.io/en/1.2.x/quickstart/)

requests. (2024). *Requests Documentation (2.32.x)*. Requests. [https://docs.python-](https://docs.python-requests.org/en/latest/)

[requests.org/en/latest/](https://docs.python-requests.org/en/latest/)

SQLAlchemy. (2024). *SQLAlchemy Documentation (2.1.x)*. SQLAlchemy.

<https://docs.sqlalchemy.org/en/20/orm/quickstart.html>

Goodreads. (2024). *Goodreads*. <https://www.goodreads.com/>

Google. (2024). *Books API*. Google Developers.

<https://developers.google.com/books?hl=es-419>

GitHub. (2024). *GitHub*. GitHub. <https://github.com/>

Clarke, E. (2020). *Building Akinator with Python using Bayes' Theorem*. Medium. <https://medium.com/analytics-vidhya/building-akinator-with-python-using-bayes-theorem-216253c98daa>

Chaves, R. (2024). *Naive Bayes Classifiers - Algoritmo de Bayes*. Guru99. <https://www.guru99.com/es/naive-bayes-classifiers.html>

CNumPy Developers. (2008). *NumPy Documentation (2.1.x)*. NumPy. <https://numpy.org/doc/stable/>

Erkalović, A. (2022). *EbookLib Documentation (0.19.x)*. EbookLib. <https://pypi.org/project/EbookLib/>

Richardson, L. (2023). *BeautifulSoup Documentation (4.13.x)*. BeautifulSoup. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Ábalos, N. (2023). *Tecnología para convertir texto a voz*. Monoceros Labs. <https://monoceroslabs.com/blog/tecnologia-para-convertir-texto-a-voz/>

Microsoft Research. (2024). *VALL-E X: Neural Codec Language Model for Text to Speech Synthesis*. Microsoft Research. <https://www.microsoft.com/en-us/research/project/vall-e-x/vall-e/>

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., & Wei, F. (2023). Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. [arXiv preprint arXiv:2301.02111].

Bidelman, E. (2020). *Web apps that talk: Introduction to the Speech Synthesis API*. Google Chrome Developers. <https://developer.chrome.com/blog/web-apps-that-talk-introduction-to-the-speech-synthesis-api?hl=es-419>

MDN Web Docs. (2018). *SpeechSynthesisUtterance interface*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesisUtterance>

MDN Web Docs. (2018). *Document.getElementById()*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById>

MDN Web Docs. (2018). *EventTarget.addEventListener()*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>

Font Awesome. (2024). *Font Awesome Documentation*. Font Awesome. <https://fontawesome.com/docs>

Python Software Foundation. (2024). *unittest — Unit testing framework*. Python Software

Foundation. <https://docs.python.org/3/library/unittest.html>

Test & Code. (2017). *Flask-Testing Documentation*. Test & Code. <https://flask-testing.readthedocs.io/en/latest/>

Mafra, É (2021). *Feed RSS: descubre cómo este recurso de difusión puede ayudarte a interactuar con la audiencia de tu blog*. Rock Content - ES.

<https://rockcontent.com/es/blog/feed-rss/>

Apéndice A

Manual de Despliegue

Requerimientos generales

Entorno de desarrollo

Se debe tener un entorno de desarrollo instalado. Se recomienda utilizar Visual Studio Code.

Aplicación Calibre

Se debe instalar la aplicación de Calibre y asignar una carpeta como Biblioteca de Calibre. Esto último puede hacerse dentro de la aplicación propia Calibre (Calibre, 2024).

Python

Se necesita tener instalado Python para poder ejecutar correctamente la aplicación y realizar otras instalaciones requeridas. Esto es posible a través de la página oficial “python.org”. Debe asegurarse de marcar la opción “Add Python to PATH” durante la instalación.

Verificación. Puede verificarse su correcta instalación comprobando la versión:

- Windows: `python --version`
- macOS/Linux: `python3 --version`

Pip

En caso de no haberse instalado pip junto con Python, se puede descargar e instalar independientemente como se muestra a continuación.

Descarga e instalación (Windows/macOS/Linux)

- `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
- `python get-pip.py`

Verificación (Windows/macOS/Linux)

- `pip --version`

Clonación e instalación de dependencias

Clonar el Repositorio y Configurar el Entorno

Se debe clonar el repositorio de calibre-web desde GitHub (Cárdenas, 2024).

A continuación, desde una terminal, acceda a la carpeta del proyecto: `cd calibre-web`

Una vez dentro del proyecto, es necesario crear un entorno virtual:

- Windows: `python -m venv venv`
- macOS/Linux: `python3 -m venv venv`

Se necesita activar dicho entorno:

- Windows: `.\venv\Scripts\activate`
- macOS/Linux: `source venv/bin/activate`

Si surge un error en este paso que impida ejecución de scripts en el sistema. Se debe:

- Abrir PowerShell como administrador.
- Comprobar la política de ejecución actual: `Get-ExecutionPolicy`
- En caso de ser *Restricted*, se debe cambiar a *RemoteSigned*: `Set-ExecutionPolicy RemoteSigned`

RemoteSigned

- Confirmar el cambio y activar el entorno virtual.

Otras instalaciones necesarias

Algunas dependencias requieren otras instalaciones adicionales para su correcto funcionamiento.

Microsoft Visual C++ Build Tools (para Windows)

Durante la instalación, seleccionar el componente *Desktop development with C++* (Desarrollo para escritorio con C++).

Asegurarse de que al menos las siguientes opciones estén marcadas:

- MSVC v143 - VS 2022 C++ x64/x86 build tools
- Windows 10 SDK (o la versión apropiada)

Instalación de dependencias

Dada la cantidad de dependencias, se ha creado un fichero requirements.txt con todas las dependencias necesarias: `pip install -r requirements.txt`

Una vez esté todo instalado, es posible iniciar la aplicación: `python cps.py`

Este paso iniciará Calibre-Web en el puerto predeterminado 8083. Puede acceder a la interfaz a través de un navegador web en la dirección: `http://localhost:8083`

Instalación de Paquetes para Pruebas

Instalación de Selenium y pytest

Se debe instalar Selenium y pytest en el entorno de desarrollo. Primero, debe asegurarse de que el entorno virtual esté activado.

- Instalación de pytest: `pip install pytest`
- Verificación: `pytest --version`
- En caso de error: `python -m pip install pytest`

- Instalación de selenium: `pip install selenium`
- Verificación: `python -c "import selenium; print(selenium.__version__)"`
- En caso de error, otra forma de instalarlo es: `python -m pip install selenium`

Instalación de un WebDriver

El proyecto incluye el archivo chromedriver, pero se debe añadir la ubicación del archivo a las variables de entorno (PATH).

En Windows:

- Acceder a la ventana de Variables de entorno.
- En la sección "Variables del sistema", buscar "Path" y seleccionar "Editar".
- Agregar la ruta al directorio que contiene chromedriver.exe.

En macOS/Linux:

- Añadir la ruta al "PATH" en el archivo de perfil:

```
export PATH=$PATH:/path/to/chromedriver
```

- Aplicar cambios: `source ~/.bash_profile` # o el archivo correspondiente

Ejecución de *Tests*

- Ejecutar un test específico: `pytest -k nombre_del_test`
- Ejecutar todos los tests: `pytest test_selenium.py`

Precauciones para los *Tests*

- Se debe tener en cuenta que algunas pruebas pueden borrar elementos o necesitar la creación previa de otros. Asegúrese de ejecutarlos en el orden correcto, ejecutando siempre los *tests* de creación antes.

- Los *tests* y *drivers* están configurados para el navegador web Google Chrome.
- Tenga en cuenta que los *tests* de notificaciones dependen de que otro usuario haya realizado cierta actividad para generar la notificación. Esto puede realizarse:

- Siguiendo al usuario a estudiar. Esta acción genera la notificación de seguimiento, que permite ver el perfil del seguidor.

- Mandando un mensaje al usuario. Esto crea una notificación de mensaje, y permite acceder al *chat* con el usuario en cuestión.

- Participando en el foro. Si el usuario a estudiar sigue a otro usuario y este crea una publicación en el foro, se genera la notificación de publicación, que permite ver el *post* en el foro.

- Para los *tests* relacionados con el audio, es necesaria la descarga del libro “El gran Gatsby”. Este puede encontrarse en la página Elejandría, que contiene libros de dominio público.

Apéndice B

Manual de Usuario

Configuración Inicial

Inicio de sesión

Cada vez que se acceda a la aplicación, se requerirá realizar el inicio de sesión. Ingrese con las siguientes credenciales de administrador, ya que el registro de nuevos usuarios no está activado inicialmente:

- Nombre de usuario: admin
- Contraseña: admin123

Configuración de la Biblioteca

Al iniciar sesión por primera vez, es necesario configurar la ubicación de la biblioteca de Calibre.

1. Seleccione el directorio donde estén los archivos de libros electrónicos gestionados por Calibre, generado en el paso de instalación de Calibre.
2. Guarde los cambios haciendo clic en el botón *Save*.

En este punto, ya se tiene acceso a las funcionalidades de la aplicación a través de la cuenta por defecto.

Configuración de la Cuenta

Personalización del Usuario

El usuario puede cambiar el idioma, el correo o la contraseña, así como el nombre de usuario desde el perfil *admin*, que se encuentra en la barra superior, a la izquierda de *Logout*.

Configuración de la Aplicación

Acceda a la pestaña *Admin* para gestionar la moderación de usuarios, configuraciones de la base de datos o del servidor de correo electrónico. Esta pestaña se encuentra a la izquierda del perfil *admin*.

Puede seleccionar *Edit Calibre Database Configuration* para cambiar la ubicación de la biblioteca de Calibre.

Configuraciones adicionales

- “Editar la configuración básica”. Permite modificar elementos de seguridad y activar funciones ocultas como el registro de nuevos usuarios. Active esta última opción para permitir la creación de cuentas.

- “Editar la configuración de la interfaz de usuario”. Contiene diversas herramientas de configuración. Active varias opciones para nuevos usuarios como “Permitir cambiar la contraseña”, “Permitir visor de libros”, opción que permite la lectura de audiolibros, y, opcionalmente, cambie el *Default Language* a español.

Configuración del Correo Electrónico

Cambios en la Configuración del Correo

Para el registro es necesario el uso de correo electrónico, puesto que se deben recibir confirmaciones a través de este y se proporcionará una contraseña temporal a cada usuario.

1. En el perfil de usuario, cambie el correo del administrador por uno propio, distinto al correo utilizado en la configuración del siguiente paso.
2. Acceda a la página de configuraciones del administrador.
3. Pulse el botón “Cambiar parámetros de correo”.
4. Modifique los siguientes datos:
 - a. Servidor SMTP: smtp.gmail.com
 - b. Puerto SMTP: 587
 - c. Encriptado: STARTTLS
 - d. Login SMTP: correo_ejemplo@gmail.com
 - e. Contraseña SMTP: contraseña

f. Desde el correo: corre_ejemplo@gmail.com

5. Al realizar estos cambios, pulse *Save and Send Test Email*. Con esta opción, el administrador debe recibir un correo de prueba desde corre_ejemplo@gmail.com. Si el correo es recibido, la aplicación está lista para el registro de usuarios mediante correo electrónico.

Detalles de los Parámetros de Correo

correo_ejemplo@gmail.com debe ser distinto al correo del administrador. El correo indicado en los parámetros anteriores debe ser Gmail, ya que se ha configurado para aceptar únicamente dicho tipo, pero los demás correos pueden ser de cualquier servicio.

La contraseña proporcionada debe ser la empleada en la cuenta de Google. En caso de tener activada la verificación en dos pasos, esta no es válida, y se necesita obtener una contraseña de aplicación. Si no se tiene la verificación en dos pasos activada y se quiere emplear la contraseña de aplicación para mayor seguridad, a continuación se muestran los pasos para activarla:

1. Acceda al apartado de Seguridad de Google.
2. Busque el apartado “Cómo inicias sesión en Google” y seleccione “Verificación en dos pasos”.
3. Siga los pasos que se indiquen en la página.

A continuación, se muestran los pasos necesarios para obtener la contraseña de aplicación:

1. Acceda al apartado de Seguridad de Google.
2. Escriba “contraseñas de aplicación” en la barra de búsqueda y seleccione el apartado que se corresponda con la búsqueda.
3. Indique el nombre de la aplicación para la cual se quiera generar la contraseña (por ejemplo, calibre-web).
4. Aparecerá una ventana emergente con la contraseña generada. Esta se debe ingresar en los parámetros de configuración.

Avisos

En ciertas ocasiones, cuando se ha utilizado la aplicación y se cierra sin cerrar la sesión correctamente, el navegador puede guardar la información de dicha sesión. Esto implica que, al volver a abrir la aplicación, el navegador, seguirá intentando usar la sesión guarda, incluso si no es válida. Como resultado, podría mostrarse una pantalla de inicio de sesión continuamente, y no se podría acceder a la aplicación.

Para solucionar este problema, se deben eliminar los datos de navegación:

1. Abra Google Chrome.
2. Haga clic en el menú y seleccione “Eliminar datos de navegación”.
3. En la ventana que aparece, asegúrese de seleccionar “Cookies y otros datos de sitios”.
4. Pulse “Eliminar datos”
5. Recargue la página donde tenga alojada la aplicación y vuelva a intentar el inicio de sesión.



UNIVERSIDAD
DE MÁLAGA | uma.es

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga