



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Desarrollo de un prototipo de seguimiento de hábitos de lectura (BookPlanning)

*Development of a prototype for monitoring reading habits
(BookPlanning)*

Luciana Belén Varela Díaz

La Laguna, 8 de septiembre de 2022

D. **Francisco Javier Rodríguez González**, con N.I.F. 43.618.712-V profesor Asociado de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Alejandro Pérez Nava**, con N.I.F. 43.821.179-S profesor Asociado de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Desarrollo de un prototipo de seguimiento de hábitos de lectura (BookPlanning)"

ha sido realizada bajo su dirección por D. **Luciana Belén Varela Díaz**, con N.I.F. 45.941.226-Z.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de septiembre de 2022

Agradecimientos

A mi familia y amigos, que me han dado su apoyo incondicional ahora y siempre. A mi tutor, Javier, que me ha brindado libertad creativa y ha sido un gran guía para el desarrollo de este trabajo.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial-SinObraDerivada 4.0 Internacional.

Resumen

Con el reciente aumento de la oferta electrónica, gran parte de los dispositivos tradicionales se han ido sustituyendo poco a poco por las opciones tecnológicas modernas. Los libros no podían ser menos. Estos también disponen de una alternativa: los libros electrónicos (E-books) y lectores de libros electrónicos (E-readers). Estas tecnologías de lectura están cobrando cada vez más fuerza. Durante la pandemia fueron una herramienta fundamental para el desarrollo de actividades académicas, pero el reto de desarrollar un hábito de lectura sigue existiendo en ambos formatos. En una época en la que el entretenimiento es fácilmente adquirible a través de cualquier aplicación, ¿Por qué acudir a la lectura?

*Tras estudiar el estado del arte de las aplicaciones de lectura así como la tendencia del mercado de libros electrónicos, se ha llegado a la conclusión de que muy pocas de ellas sirven para propiciar el desarrollo de un hábito de lectura. En este prototipo se propone una aplicación con la que podamos iniciar el camino para obtenerlo. Por un lado disponemos de nuestra aplicación móvil, desarrollada con **Expo, React Native** y el framework **Native Base con Typescript** . Este framework junto a **React** y **Next.js** nos permitirá desarrollar una versión web. Además, dispondremos de una API desarrollada en **Nest.js** que conectará con nuestra base de datos en **SQLite** y con distintas herramientas de **Firebase**.*

En el quinto apartado se ha realizado un estudio de la viabilidad económica de la aplicación, la duración estimada del proyecto es de 9 meses y el retorno de la inversión se obtendría aproximadamente a los 16 meses de su comercialización.

Palabras clave: Libros electrónicos, lectura, aplicación, planificación.

Abstract

Nowadays there are a lot of electronic devices, they are taking over our daily tasks. Books are not left out of this tendency, they have an electronic substitute, ebooks and ereaders. These reading technologies are getting stronger day by day. They were a fundamental tool during the pandemic because they helped many students and teachers to still be able to perform their academic activities. However, developing a reading habit is not that simple, in our current world we have a lot of easier options to entertain ourselves. Why would we choose to read?

*After studying the market of reading applications and the ebook market growth, we have come to the conclusion that very few of them help us to develop a reading habit. In this prototype we will be building an application to start creating one. We will have a mobile application made with **Expo**, **React Native** and **Native Base** framework using **Typescript**. This framework but combined with **React** and **Next.js** will allow us to develop a web version. We will also develop an API made with **NestJS** that will connect to our **SQLite** database and different **Firestore** tools.*

On the fifth chapter we made an economic feasibility study of the application, the estimated duration of the project is 9 months and the return of the investment will be met 16 months after its commercialization.

Keywords: EBooks, Reading, Application, Planning.

Índice general

1. Introducción	1
1.1. Definición del problema	1
1.2. Justificación	1
1.3. Estado del arte	2
1.3.1. Aplicaciones multiuso	2
1.3.2. Lectura de libros clásicos	4
1.3.3. Seguimiento de lectura	4
1.3.4. Objetivos de lectura	5
1.4. Objetivos	7
2. Estudio Previo	9
2.1. Importancia de la lectura en el desarrollo cognitivo	9
2.1.1. El cerebro lector	9
2.2. Tendencia en el mercado de libros	10
2.2.1. Breve historia del <i>E-Book</i>	11
2.2.2. Ventajas y desventajas del <i>E-Book</i>	11
2.2.3. Tendencia del mercado del <i>E-Book</i>	13
3. Desarrollo de arquitecturas	15
3.1. Elección de arquitecturas	15
3.1.1. Lenguajes de programación	15
3.1.2. Herramientas adicionales	19
3.2. Web Scraper	20
3.2.1. ¿Qué es el Web Scraping? Utilidad y controversia	20
3.2.2. Utilidad en nuestro prototipo	21
3.3. Aplicación móvil	22
3.3.1. Retos de lectura	27
3.4. Página web	30
3.5. Plataforma de back-end	34
3.5.1. Firebase	34
3.5.2. Base de datos	34
3.5.3. API	37
3.6. Resultados del desarrollo	39
4. Estudio de la viabilidad económica del proyecto	40
4.1. Modelo de comercialización	41
4.1.1. Versión gratuita	42

4.1.2. Versión <i>premium</i>	42
4.2. Desarrollo del proyecto	42
4.3. Coste y duración del proyecto	45
4.4. Estimación del retorno de la inversión	45
5. Conclusiones y líneas futuras	50
6. Summary and Conclusions	52
7. Presupuesto	53

Índice de Figuras

1.1. Índice de personas que leen en España 2022.	1
1.2. Pagina inicial de Goodreads.	3
1.3. Pagina inicial de Story Graph.	3
1.4. Varias imágenes de My Books en un dispositivo iOS.	4
1.5. Varias imágenes de Reading List en un dispositivo iOS.	5
1.6. Varias imágenes de Basmo en un dispositivo iOS.	6
1.7. Imágenes de Bookly en un dispositivo iOS.	7
2.1. Área de brocas en modelo 3D del cerebro.	10
2.2. Evolución de lectores en formato digital en España.	11
2.3. Evolución de la tasa de crecimiento de la venta de libros electrónicos.	14
3.1. Logo de Python.	16
3.2. Logos de <i>Typescript</i> , <i>React</i> (y <i>React Native</i>), <i>Expo</i> y <i>Next.js</i> respectivamente.	17
3.3. Logos de <i>Node JS</i> , <i>Nest JS</i> y <i>Firebase</i> respectivamente.	18
3.4. Logo de SQLite.	19
3.5. Logos de <i>Visual Studio Code</i> y <i>Github</i> respectivamente.	20
3.6. Re-Captcha común en las páginas web.	21
3.7. Login de la aplicación.	23
3.8. Página inicial de la aplicación al hacer login.	23
3.9. Página para realizar la búsqueda.	24
3.10. Página en la que vemos las estadísticas del usuario.	25
3.11. Pestaña de favoritos.	25
3.12. Pestaña del perfil del usuario.	26
3.13. Elegir fecha para finalizar el libro.	27
3.14. Libro añadido a la página de inicio.	28
3.15. Añadimos un nuevo registro.	29
3.16. Elegir fecha para finalizar el libro.	30
3.17. Elegir fecha para finalizar el libro.	30
3.18. Login en versión web.	31
3.19. Página de inicio en versión web.	31
3.20. Imágenes de la búsqueda en versión web.	32
3.21. Establecer reto del libro guardado en web.	32
3.22. Añadir nuevo día leído.	32
3.23. Estadísticas en la versión web.	33
3.24. Comentarios en la versión web.	33
3.25. Perfil en la versión web.	34
3.26. Esquema de la base de datos.	36
3.27. Árbol de directorios de un elemento en la API.	38

3.28	Captura de pantalla de <i>Swagger UI</i>	39
4.1.	Tareas definidas para el desarrollo del proyecto.	44
4.2.	Primera parte del diagrama de Gantt.	44
4.3.	Segunda parte del diagrama de Gantt.	44
4.4.	Tabla de recursos materiales y humanos para el desarrollo del proyecto. . .	45
4.5.	Primeras filas de estimación de crecimiento de usuarios.	46
4.6.	Primeras filas del ingreso por publicidad.	47
4.7.	Primeras filas del ingreso por suscripciones.	47
4.8.	Primeras filas de gastos de la aplicación.	48
4.9.	Primeras filas del cálculo de ingresos contra gastos.	49
4.10	Punto del retorno de la inversión.	49
4.11	Gráfica representando los gastos y beneficios.	49

Capítulo 1

Introducción

1.1. Definición del problema

La lectura es una actividad beneficiosa para el ser humano. Es una de las que podemos ejecutar sin mucho esfuerzo y a pesar de ello es un entrenamiento muy completo para el cerebro. Sin embargo, un 35 % [1] de la población española asegura que no lee. Ya sea por falta de tiempo, porque no tiene ningún interés en la lectura o prefieren realizar otras actividades. Además, de ese 65 % de lectores restante, el 20 % lo hace por obligación, es decir, por trabajo o estudios. Es difícil encontrar una motivación que nos impulse leer y también mantenerla día tras día, ya que si queremos crear un hábito hace falta ser constante. Por mucho que influya en nuestro desarrollo cerebral y enriquecimiento personal, a veces, aunque se quiera y se intente, es complicado.

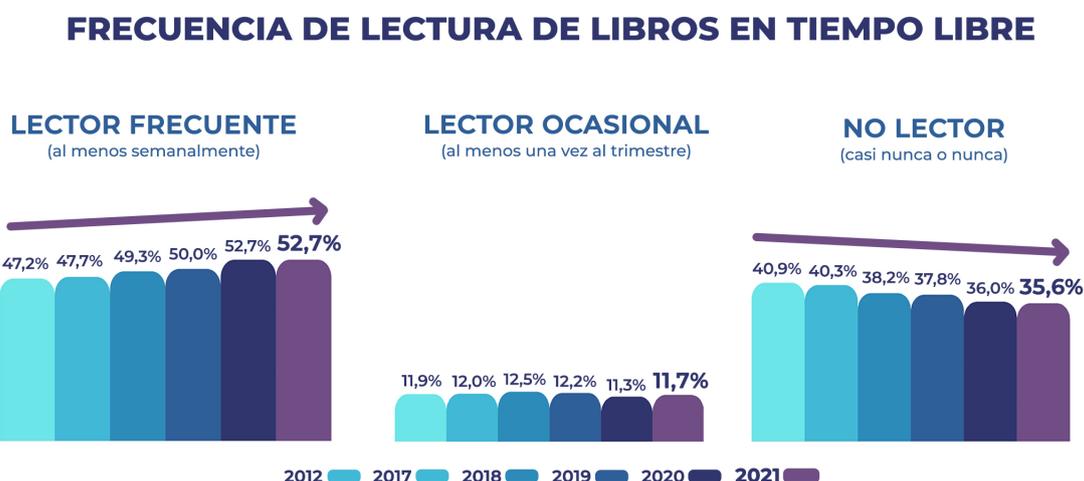


Figura 1.1: Índice de personas que leen en España 2022.

1.2. Justificación

La lectura es una actividad beneficiosa, sí. Pero es una destreza que tenemos que adquirir, y un gusto que hay que desarrollar. Muchas personas a lo largo de su vida intentan hacerlo. Se marcan el propósito de leer un libro al mes, por ejemplo, o alcanzar

un número de páginas al día. Estos objetivos son válidos, pero en muchas ocasiones decirlo no es suficiente. Nos olvidamos de hacerlo y enterramos el propósito que nos marcamos.

Es por ello que se propone una aplicación que pueda ayudar a desarrollar ese hábito. Ya sea para iniciarse en una nueva afición o por interés en alguna materia. También puede ser útil a la hora de programar la lectura de un texto que necesitemos para un proyecto o trabajo. Tener una herramienta que nos recuerde que debemos dedicar tiempo a la lectura diaria, o donde registrar aquello que hemos leído o queremos leer en el futuro, nos permitirá ser más organizados con nuestro objetivo.

Por tanto podemos identificar las siguientes necesidades:

- Definición de objetivos.
- Guardar lecturas actuales (de diversas fuentes).
- Lista de favoritos.
- Posibles recordatorios.

1.3. Estado del arte

En la actualidad existen herramientas que nos permiten realizar acciones similares a las descritas, pero muchas pierden el foco con demasiadas funcionalidades. No se centran en propiciar la creación de un hábito, sino en brindarnos información de nuevos libros o foros de discusión acerca de los mismos, lo cual solo será útil si ya tenemos nuestro hábito creado.

Como se mencionó antes, las tendencias del mercado se centran en aplicaciones con múltiples usos. Al final se convierten en *blogs*¹ de lectura o recomendaciones de novedades. También existen bibliotecas de libros antiguos que pueden distribuirse libremente ya que su *copyright*² se ha perdido y ahora es una obra de propiedad pública.

1.3.1. Aplicaciones multiuso

Existen diversas aplicaciones que se han convertido en este modelo. Una de las más famosas es *Goodreads*³. Sin duda es una herramienta muy útil para aquellas personas que leen de forma regular y quieren mantenerse actualizados de las novedades, o estar en contacto con otros lectores. Pueden comentar qué les ha parecido un libro, puntuarlos, guardarlos, o seguir a sus autores favoritos. Por ello el formato es más similar a un *blog*.

¹**Blogs:** Página web, generalmente de carácter personal, con una estructura cronológica que se actualiza regularmente y que se suele dedicar a tratar un tema concreto.

²**Copyright:** Derecho exclusivo de un autor, editor o concesionario para explotar una obra literaria, científica o artística durante cierto tiempo.

³**Goodreads:** Es una red social o comunidad específica de lectores. Actualmente es una empresa subsidiaria de Amazon. Dispone de una página web y una aplicación tanto en Android como en iOS.

Deciding what to read next?

You're in the right place. Tell us what titles or genres you've enjoyed in the past, and we'll give you surprisingly insightful recommendations.

What are your friends reading?

Chances are your friends are discussing their favorite (and least favorite) books on Goodreads.

Figura 1.2: Pagina inicial de Goodreads.

Con una interfaz diferente tenemos la aplicación de *Story Graph*⁴. En la que se implementan retos de lectura, pero estos son destinados a personas que de por sí leen mucho, y buscan expandir su biblioteca. Ya que son retos por géneros, autores, o países. También tienen una comunidad para compartir tu valoración personal de los libros. Dispone de una funcionalidad interesante en la que podemos importar nuestra biblioteca de *Goodreads*, que ha sido previamente exportada.

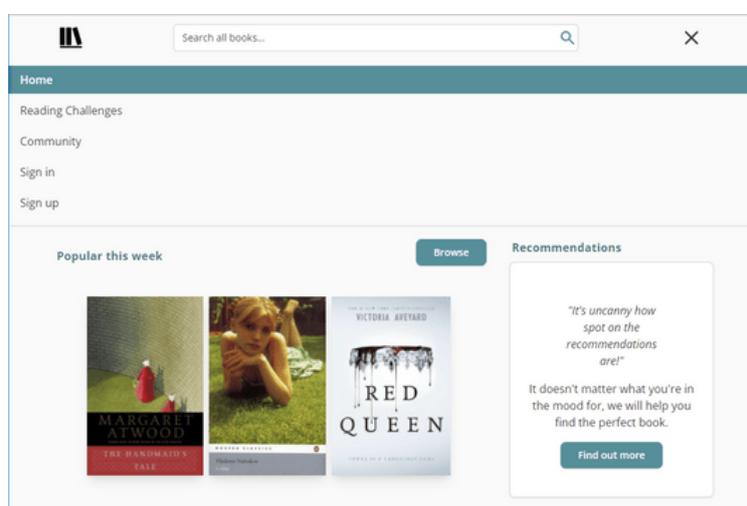


Figura 1.3: Pagina inicial de Story Graph.

⁴**Story Graph**: Su traducción es similar a gráfico de la historia o del relato. Dispone de una página web y una aplicación tanto en Android como en iOS.

1.3.2. Lectura de libros clásicos

Otro tipo de aplicaciones que se han encontrado son las que tienen un catálogo formado por obras de dominio público. Es decir, pueden ser editadas, reproducidas o difundidas. Por ejemplo, libros escritos bajo licencias públicas o aquellos que debido al paso del tiempo sus derechos de autor han expirado.

Un ejemplo puede ser la aplicación de *My Books*⁵. Tiene audio libros⁶ y también una extensa biblioteca de libros clásicos. Están clasificados por géneros. Nosotros podemos buscar y guardar los que nos interesen. Además, tenemos obras recomendadas, ya sean las más leídas (o escuchadas) o los últimos agregados a la biblioteca por los usuarios.

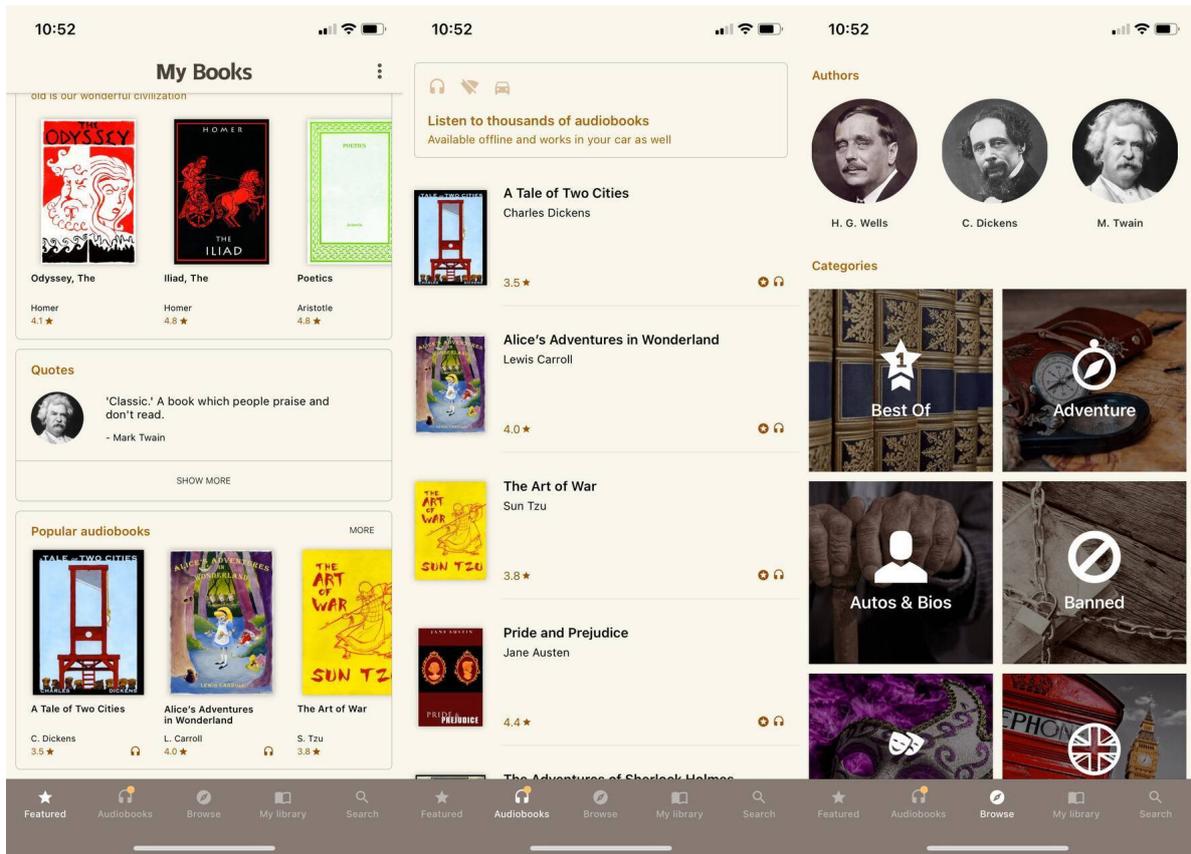


Figura 1.4: Varias imágenes de My Books en un dispositivo iOS.

1.3.3. Seguimiento de lectura

Estas aplicaciones son interesantes porque tienen funcionalidades más similares a las que buscamos implementar.

Hay aplicaciones como *BookBuddy*⁷, que nos permiten tener una biblioteca en nuestro dispositivo móvil. De esta forma podremos acceder en cualquier momento a los libros que tenemos en nuestra biblioteca tanto física como virtual, teniendo los datos centralizados en una aplicación. En ella podremos apuntar los libros que tenemos, si los tenemos porque

⁵**My Books:** Aplicación móvil, su traducción es "Mis libros". Se encuentra disponible en Android y en iOS.

⁶**Audiobook:** Grabación que contiene la lectura en voz alta de un libro.

⁷**BookBuddy:** Aplicación móvil, también existe una versión para Android e iOS, su traducción es compañero de lectura.

son un préstamo o si hemos prestado uno de nuestros libros, además de apuntar si los estamos leyendo y otros datos acerca del lugar en el que los adquirimos.

Otras, son más simples como *Reading List*⁸. En ella podemos tener una lista de libros que estamos leyendo, libros que hemos finalizado, organizarlos en distintas carpetas y descubrir libros nuevos. Ya sea porque son recientes o ver los más leídos por categorías.

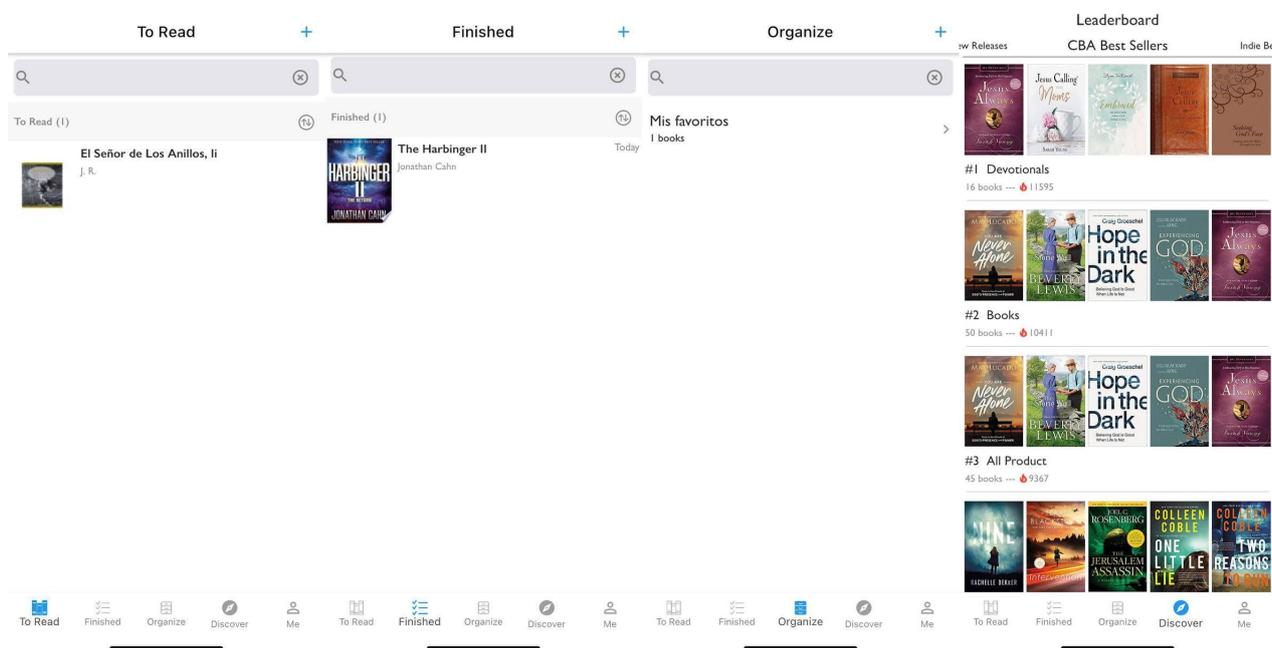


Figura 1.5: Varias imágenes de Reading List en un dispositivo iOS.

1.3.4. Objetivos de lectura

Estas son las aplicaciones que más nos interesan para el desarrollo del prototipo para este TFG. Varias aplicaciones que pertenecen a grupos anteriores tienen funcionalidades similares a las que describiremos a continuación, pero no son su foco principal.

En especial, hemos encontrado dos que concentran las funcionalidades que nos gustaría abordar.

Primero tenemos una aplicación llamada *Basmo*⁹. Tiene una interfaz muy limpia, en ella podemos agregar libros a nuestra biblioteca y a partir de ahí, establecer objetivos de lectura diaria. Por ejemplo, minutos diarios o días a la semana que queremos leer. Además, si tenemos la versión *premium*¹⁰ podemos desbloquear estadísticas de los libros que hayamos añadido y leído.

⁸ **Reading List**: Aplicación móvil disponible para iOS, su traducción lista de lectura.

⁹ **Basmo**: Aplicación móvil dedicada a definir objetivos de lectura.

¹⁰ **Premium**: Dicho de un producto o servicio, que ofrece unas ventajas exclusivas para la persona que lo tiene.

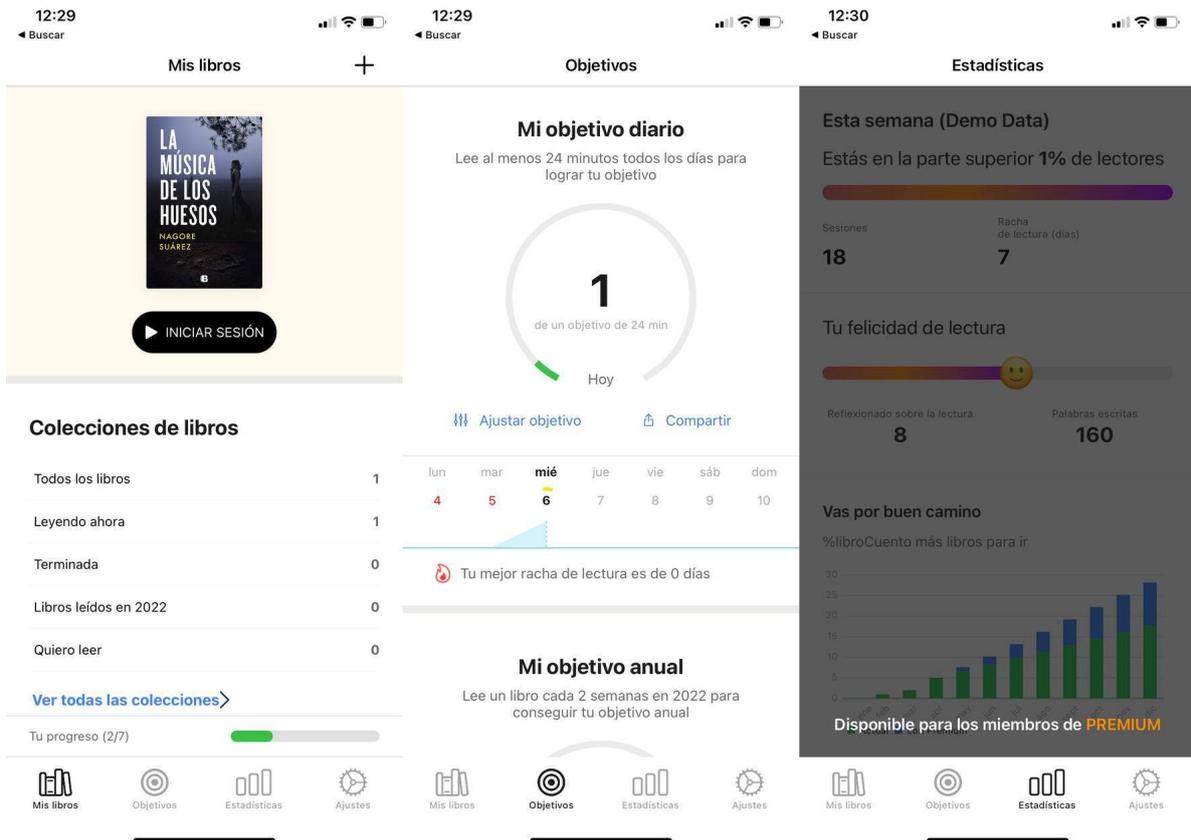


Figura 1.6: Varias imágenes de Basmo en un dispositivo iOS.

La otra aplicación que nos interesa y es más extensa es *Bookly*¹¹. Esta aplicación profundiza mucho más, llegando a disponer de un asistente que irá subiendo de nivel a medida que nosotros registramos datos de nuestros hábitos de lectura. Pero no sólo eso, si dispones de la versión *premium* también sus funcionalidades irán aumentando, tiene una tienda para que personalizemos el avatar del asistente.

De aquí podemos extraer datos interesantes, no solo en diseño. Por ejemplo, vemos que tener un *banner*¹² que nos recuerde que tener el *premium* podría ser beneficioso, pero también que es importante disponer de ciertas estadísticas sin aumentar nuestra suscripción. Esto nos motiva a querer desbloquear más características.

¹¹ **Bookly**: Aplicación móvil dedicada a los objetivos de lectura.

¹² **Banner**: Espacio publicitario insertado en una página de internet.

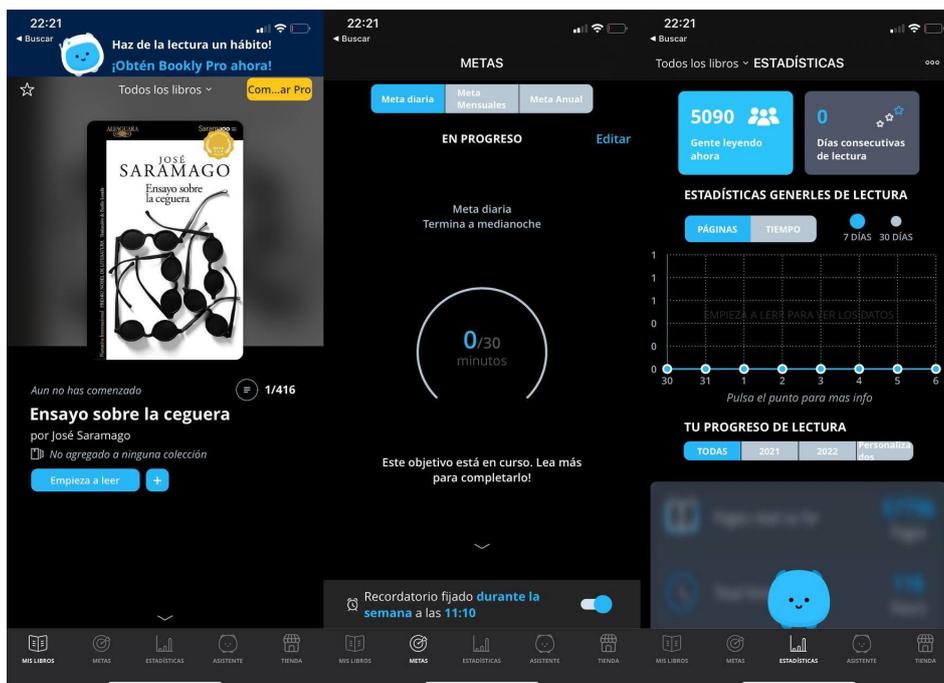


Figura 1.7: Imágenes de Bookly en un dispositivo iOS.

Contar con opciones de personalización también nos podrían ayudar en el Retorno de la Inversión¹³.

1.4. Objetivos

El principal objetivo es desarrollar un prototipo¹⁴ que esté centrado en la definición de retos de lectura. Y que nos permita realizar esta tarea de forma cómoda.

Podemos discernir las siguientes metas:

- Investigación de las aplicaciones actuales.
- Estudio del *hardware*¹⁵, *software*¹⁶ y herramientas que utilizaremos.
- Análisis del desarrollo del prototipo
- Desarrollo:
 - Desarrollo móvil y web
 - Diseño e implementación de base de datos.
 - Desarrollo *Back-end*¹⁷.

¹³**Retorno de la Inversión:** Abreviado ROI. Es una métrica que nos indica el rendimiento económico que se obtiene al realizar una inversión. Nos señala el punto en el que los gastos iniciales se compensan a través de los beneficios obtenidos.

¹⁴**Prototipo:** Primer ejemplar que se fabrica de una figura, un invento u otra cosa.

¹⁵**Hardware:** Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

¹⁶**Software:** Término informático que hace referencia a un programa o conjunto de programas de cómputo.

¹⁷**Back-end:** Parte del desarrollo web que se encarga de que toda la lógica de una página web funcione.

- Desarrollo *Front-end*¹⁸.
- Realizar pruebas de funcionamiento
- Crear un Diagrama de *Gantt*¹⁹ con todas las tareas que se realizarían en un escenario real, los recursos estimados, así como la asignación de tiempo y recursos a las tareas.
- Estimar costes del proyecto real.
- Realizar un estudio acerca del Retorno de la Inversión de una aplicación en el escenario real.

¹⁸**Front-end**: Parte del desarrollo web que se dedica a la parte frontal de un sitio web, es decir de su diseño.

¹⁹**Diagrama de Gantt**: Herramienta para organizar el desarrollo de un proyecto, dividiendo su desarrollo en tareas bien definidas.

Capítulo 2

Estudio Previo

Una vez introducida la problemática podemos hablar del ecosistema que rodea a la misma. Veremos primero por qué es tan importante la lectura, en especial en el desarrollo infantil y por qué es beneficioso para nosotros crear un hábito de lectura. A raíz de esto, veremos cómo se reflejan estas rutinas y tendencias en el mercado.

2.1. Importancia de la lectura en el desarrollo cognitivo

Desde que somos pequeños se nos quiere inculcar un hábito lector. Escuchamos con frecuencia frases como "*Mi hijo no lee*", "*Si leyeras, serías más listo*", ya sea por parte de familiares o en las aulas. Además, existe un esfuerzo por parte de los planes educativos de hacer que los niños se lean, al menos, dos o tres libros en el curso escolar. Alguna vez con éxito y otras no tanto, porque cuando queremos forzar un hábito en alguien que no quiere desarrollarlo, lo normal es fracasar. Pero, ¿A qué se debe tanta insistencia? ¿Es realmente la lectura una práctica beneficiosa?

2.1.1. El cerebro lector

Leer es una actividad compleja. Para aprender a hacerlo, debemos saber procesar sonidos y repetirlos. Combinándolos podremos comenzar a decir palabras. Estas conformarán el vocabulario que nos permitirá formular y comprender frases. Será a partir de ese momento en el que comenzaremos a dar sentido a la lectura y, poco a poco, mejorar nuestra comprensión lectora, para dominar el lenguaje y tener una velocidad aceptable de lectura. Solo así podremos desarrollar otras habilidades de procesamiento de texto más avanzadas.

Es por ello que la respuesta, es que sí, la lectura es una práctica beneficiosa para el ser humano. Varias zonas del cerebro se involucran para que sea posible leer. Entre ellas el lóbulo temporal, que nos permite decodificar y distinguir los sonidos. El área de Brocas en el lóbulo frontal consigue que podamos producir y comprender el lenguaje [2]. Otro componente crucial es la materia blanca del cerebro, que son fibras nerviosas. A través de estas fibras viaja la información en nuestro cerebro. En un estudio llevado a cabo en la Universidad de *Stanford* (y otras universidades colaboradoras) reportó que esta materia se desarrolla más en niños con mejor capacidad de lectura [3].

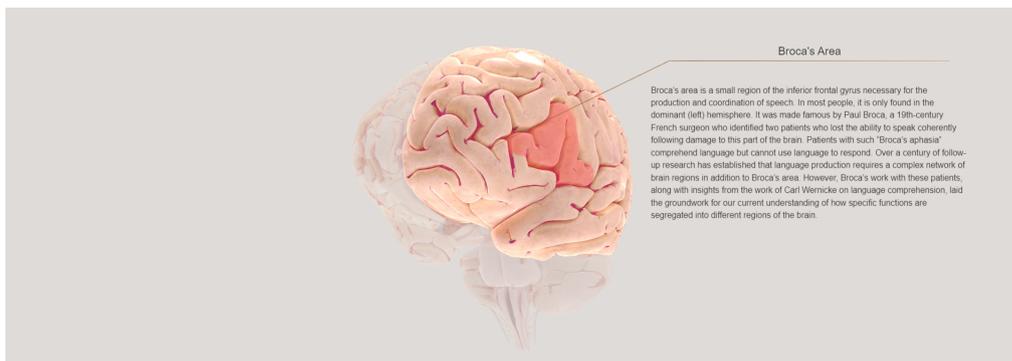


Figura 2.1: Área de brocas en modelo 3D del cerebro.

Este desarrollo cerebral implica una mayor capacidad de obtener vocabulario, más facilidad para comunicarse, etc. Aunque muchas personas piensen que no es relevante y podemos ignorar esta actividad completamente, puede traernos consecuencias.

Un estudio realizado en personas de más de sesenta y cinco años; de las cuales algunas recibieron cuatro años o menos de educación, reveló que aquellas que no sabían leer ni escribir eran más propensas a iniciar la investigación con una demencia incipiente, o a desarrollarla durante el mismo [4]. También se ha estudiado su contraparte, un programa de lectura y otras actividades estimulantes mejoraron el desempeño cognitivo de pacientes con demencia [5]. Pero no hace falta acudir a pacientes de avanzada edad, una investigación realizada en estudiantes universitarios, probó que la literatura recreativa ayudaba a reducir los niveles de angustia que estos sufrían a lo largo del curso escolar [6].

Además, existe un término llamado analfabeto funcional [7] que está siendo estudiado actualmente. Se define como aquella persona que sí dispone de la capacidad de leer y escribir correctamente, pero las tareas que impliquen usar esos conocimientos se vuelven difíciles. Por ejemplo, comprender un texto o rellenar un formulario. Esto se debe a que su comprensión lectora es muy baja y aunque se disponga de esas habilidades, en la práctica no pueden ser usadas correctamente. En este campo no se incluyen personas con afecciones como la dislexia o personas que no sepan leer ni escribir. Podría ser corregido con la práctica, cuanto más leemos, más podremos desarrollar estas capacidades.

Con estos indicios, podemos afirmar que la lectura al involucrar diversas partes del cerebro es una actividad muy positiva para nosotros, y en mayor o menor medida puede ser la clave para una mejor calidad de vida.

2.2. Tendencia en el mercado de libros

El libro convencional está muy lejos de desaparecer. Su índice de ventas sigue siendo superior al de sus alternativas electrónicas, ya que muchas personas prefieren el formato físico. Estudios defienden que existe una ventaja, en cuanto a comprensión se refiere, al leer la edición en papel [8]. Sin embargo, desde la aparición del primer libro electrónico y, en especial en los últimos años, con la situación extraordinaria que hemos vivido todos, la venta de los *E-Readers*¹ y los *E-Book*² ha aumentado [9] y es un factor que se

¹ **E-Reader**: Dispositivo electrónico diseñado especialmente para la lectura, emulando la lectura en papel

² **E-Book**: Libro en formato digital.

mantiene constante, ganando cada vez más fuerza debido a las grandes ventajas que ofrece. Lectores que en un principio se mostraron reticentes a su uso debido al factor de la nostalgia, son fieles seguidores del libro electrónico.

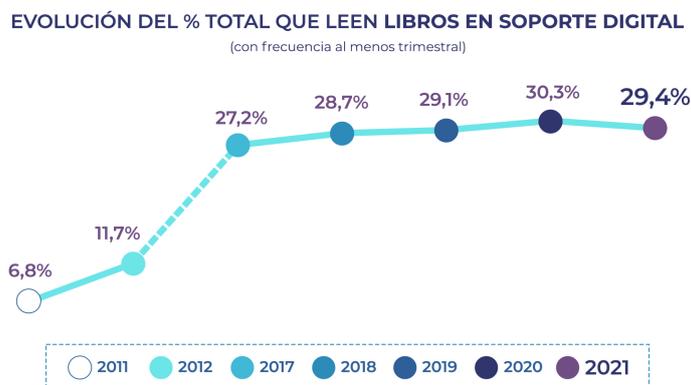


Figura 2.2: Evolución de lectores en formato digital en España.

2.2.1. Breve historia del *E-Book*

El primer *E-Book* surgió en 1971, iniciado gracias al **Proyecto Gutenberg** [10] fundado por Michael Hart, en el que un gran número de obras de dominio público se convirtieron en la primera librería digital. El primer documento que se unió a este proyecto fue la Declaración de Independencia de Estados Unidos. A través de la red local que disponía la Universidad de *Illinois* se compartieron los primeros libros.

Con la llegada del internet, la distribución de estos se hizo cada vez mayor. Dando lugar no sólo al fenómeno del *E-Book*, sino también a la creación de los primeros dispositivos de lectura electrónica o *E-Readers*. Entre ellos se encuentran los modelos *Data Discman*, *Sony Bookman*, ambos de la marca *Sony*, el dispositivo *Franklin Bookman*, *Rocket Book*...

En 1995 **Amazon**³ comenzó su trayectoria vendiendo libros en formato físico. En el año 2000 comenzó a distribuir también *E-Books*. Comenzó con apenas 1000, aumentando su catálogo paulatinamente. Transcurridos cinco años, *Amazon* adquirió la compañía francesa *Mobipocket* especializada en software de lectura y *E-Books* para los *Personal Digital Assistant* (PDAs)⁴. En el año 2007 *Amazon* lanzó su primer *E-Reader*, llamado *Kindle*, en ese momento se disponía de un catálogo de 80,000 *E-Books* en *Amazon*.

A partir de ahí, surgieron otros dispositivos, como por ejemplo, el competidor del *Kindle*, *Nook*⁵, pero el mercado actual trabaja para la continua mejora de sus tecnologías, como por ejemplo, la tinta electrónica o ***E-Ink***⁶.

2.2.2. Ventajas y desventajas del *E-Book*

Si consideramos el origen del *E-Book* el año 1971 con el proyecto *Gutenberg* mencionado anteriormente, es una tecnología que en 2021 cumplió 50 años, el tiempo que

³ **Amazon**: Compañía de venta de bienes a través de internet. Es decir, dedicada al comercio electrónico.

⁴ **PDA**: Computadora de bolsillo.

⁵ **Nook**: *E-Reader* desarrollado por la compañía Barnes & Noble, la mayor librería de Estados Unidos.

⁶ ***E-Ink***: Tecnología que permite crear pantallas planas, muy finas, que representan la información de forma que pueden ser leídas cómodamente como si fuera un papel. Minimizando la fatiga visual que producen las pantallas comunes.

hemos convivido con ellos no es nada en comparación al libro clásico. Es por ello que su crecimiento y su futuro aún está por verse.

Aun así, podemos discernir una serie de ventajas que son la clave para su éxito y su promoción en los años venideros:

- **Portabilidad:** En un dispositivo se encuentran todos nuestros libros. No dependemos tampoco de un medio, podemos disponer de nuestros *E-Books* en un *pendrive*⁷ o en un *E-Reader*.
- **Medio ambiente:** La producción de un *E-Reader* necesita muchos recursos naturales, pero a la larga, podría reducir el *Global Warming Potential* (Potencial de Calentamiento Global - GWP)⁸ [11].
- **Distribución inmediata:** Cuando el mercado de los libros clásicos a papel se vio afectado por los retrasos en las distribuciones, muchos usuarios decidieron acudir a la versión en formato digital, ya que estas se distribuyen de forma inmediata y no se descatalogan⁹.
- **Personalización:** Podemos personalizar el tamaño de fuente¹⁰, imágenes, elegir formatos de texto.
- Fáciles de corregir y editar. No hay que volver a repetir el procedimiento de impresión.
- Nos permiten tener **anotaciones** en ellos sin ser dañados así como buscar una palabra o frase en el libro de forma instantánea. Si los leemos en un *E-Reader*, probablemente tengamos la opción de traducir de forma sencilla o buscar el significado de cualquier palabra.
- **E-ink:** Esta tecnología avanza cada vez más, ya estando adaptada lo suficiente como para poder leer en ella sin que nos afecte a los ojos como la luz azul de las pantallas comunes, y recientemente sus pantallas a color han alcanzado los 50.000 colores [12].
- Fácilmente **adaptable** a personas con necesidades especiales, por ejemplo, existe una fuente llamada *OpenDyslexic* que facilita la lectura a personas con dislexia.

Existen otras ventajas aunque son más subjetivas. Pero también podemos nombrar algunas desventajas:

- No es un producto independiente: Los *E-Books* necesitan ser leídos en algún medio, ya sea un dispositivo móvil o un *E-Reader*.
- Necesitamos internet para poder comprarlos y descargarlos en nuestros dispositivos.

⁷ **Pendrive:** Memoria portátil que se enchufa a otros dispositivos a través de un puerto USB

⁸ **Potencial de Calentamiento Global:** Indicador utilizado para medir el calentamiento global a lo largo del tiempo.

⁹ **Descatalogar:** Los libros descatalogados son aquellos que no serán editados nuevamente.

¹⁰ **Fuente:** Tipo de letra utilizada para la representación del texto.

- No existe un formato único para los *E-Book*, pero por ahora no disponen de una extensión común, la más conocida es el formato .epub, pero existen otros como .kpf o .mobi para los libros de *Amazon*, o .lit para los *E-Book* de *Microsoft* entre otros. Algunos *E-Readers* no soportan todos los formatos en los que se nos puede presentar un *E-Book* [13].
- No existe un formato estándar para su paginación ya que ésta varía según su tamaño de letra, es por ello que depende de dónde vaya a ser publicado podría variar su número de páginas [14].
- Es más sencillo reproducir ilegalmente un *E-Book*, que un libro físico.

A pesar de sus inconvenientes, podemos ver que tiene amplias ventajas, las cuales le han permitido mantenerse estable y escalar poco a poco en los *rankings*¹¹ de venta desde su aparición.

2.2.3. Tendencia del mercado del *E-Book*

El libro tradicional comparte su cuota de mercado con otros formatos que poco a poco crecerán. Es el caso no sólo de los *E-Book* sino también del formato audiolibro.

El mercado de los *E-Books* comenzó a tomar fuerza a finales de los años 2000, cuando se popularizaron los *E-Readers* y *Amazon* tomó las riendas de este terreno, ya que previamente había muchos esfuerzos de digitalización, pero no un medio común en el que obtener libros digitales. A raíz de esto y con el paso de los años su popularidad ha ido en aumento junto a su cuota de mercado. Aquí podemos ver una gráfica con los últimos años de consumo y una previsión de crecimiento del mercado del *E-Book*, basado en la tendencia de compra. Se prevé una tasa de crecimiento anual del 7% de aquí a 2025 [15].

¹¹**Ranking:** Lista ordenada por un criterio, en este caso será el número de ventas.

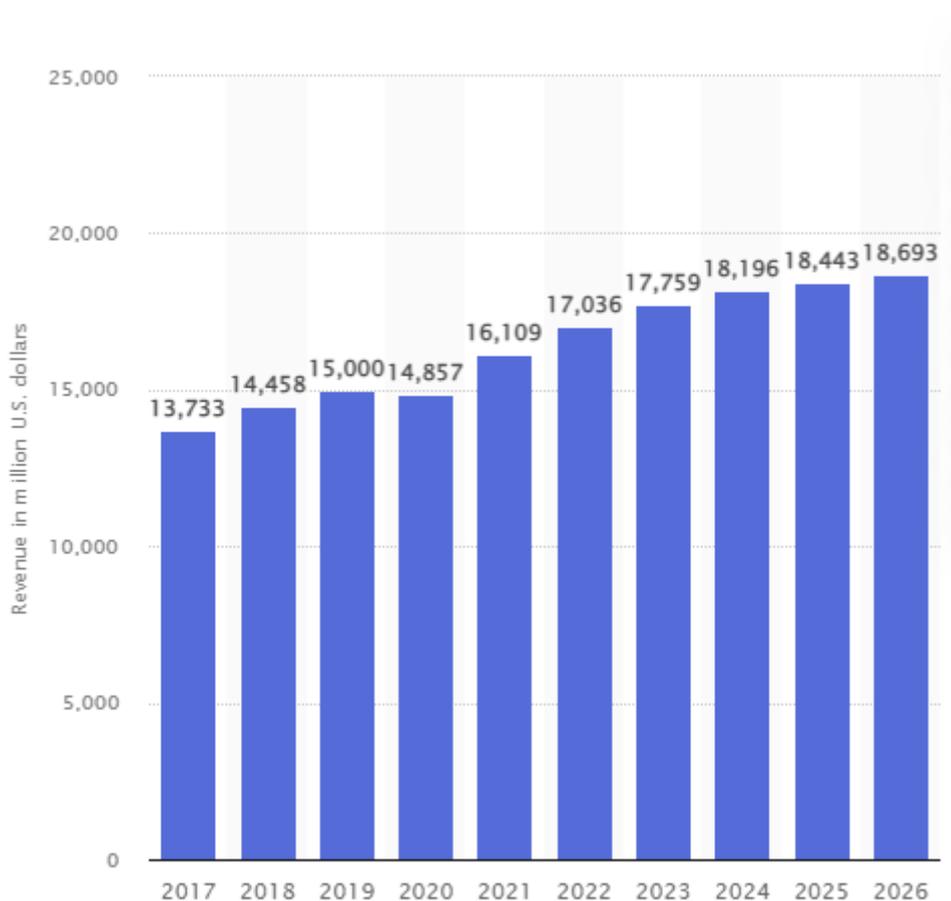


Figura 2.3: Evolución de la tasa de crecimiento de la venta de libros electrónicos.

Es cierto que podemos ver que las ventas han decaído, es un hecho innegable, pero no podemos culpar a los libros electrónicos. La pandemia¹², la creciente inflación¹³, problemas con las cadenas de producción, distribución de papel, la invasión de Ucrania por parte de Rusia y los problemas energéticos derivados de estos acontecimientos, están afectando a todos los mercados [16]. Es decir, aunque veamos un declive con respecto al año anterior, sí que vemos una mejora con respecto a los valores pre-pandemia. Es decir, la tendencia es que el balance sea positivo. Es por ello que el mercado del *E-Book* tomará más fuerza.

Actualmente disponemos de medios para obtener libros electrónicos, muchas personas tienen un *E-Reader* o algún dispositivo inteligente portátil, como una tableta¹⁴ o un teléfono móvil en el que leer. Este dato es importante, ya que a principios de los 2000 no era así. No es descabellado pensar que en los años venideros tomarán mucha más importancia. No sólo a nivel de usuario, si no también de autor, no tener que depender de una editorial para compartir nuestras creaciones literarias es una razón de peso para trasladarnos al campo electrónico.

¹²**Pandemia:** Referencia a la enfermedad del Covid-19 que ha afectado en gran medida los últimos años.

¹³**Inflación:** Proceso económico que ocurre cuando hay un desbalance entre la oferta y la demanda, provoca la subida de los precios.

¹⁴**Tableta:** Computadora portátil que normalmente es de un tamaño superior a un teléfono móvil pero inferior al de un ordenador.

Capítulo 3

Desarrollo de arquitecturas

Ahora procederemos a hablar de la elección de las arquitecturas de nuestro prototipo, es decir, las herramientas que han sido empleados para desarrollar cada uno de los programas que lo componen.

3.1. Elección de arquitecturas

3.1.1. Lenguajes de programación

Existen muchos lenguajes de programación y todos tienen sus ventajas y sus desventajas, ya que cada uno está orientado a realizar unas tareas concretas.

En nuestro caso queremos implementar varias arquitecturas:

- Web Scraper (Definido en la sección 3.2)
- Aplicación móvil
- Aplicación web
- Plataforma de *Back-End*

Web Scraper

Se barajaron diversos lenguajes para el desarrollo del *web scraper*, pero nos decantamos por utilizar **Python**¹ [17]. Esto se debe a que es un lenguaje de alto nivel que es muy legible, no necesitamos una configuración muy extensa para ponerlo a funcionar y sobre todo, es uno de los lenguajes más empleados en el mundo para el procesamiento de datos. Además disponemos de librerías que nos facilitan mucho el trabajo, en nuestro caso lo empleamos junto a la librería **Selenium**² [18], que hace este proceso mucho más sencillo. Otra librería recomendada puede ser **Beautiful Soup**⁴.

¹**Python**: Lenguaje de código abierto de alto nivel interpretado, utilizado en muchos ámbitos como la ciencia de datos, el aprendizaje automático...

²**Selenium**: Librería de *Python* para facilitar la interacción con el navegador. Nos permite crear tests automatizados y crear *bots*³ para obtener información de las páginas web (*Web Scraping*)

⁴**Beautiful Soup**: Biblioteca de *Python* que sirve para analizar documentos html.

Otro lenguaje que se planteó fue **JavaScript**⁵ [19], *JavaScript* es el lenguaje de la web y junto a **Node.js**⁶ [20] es otra alternativa válida para poder realizar un programa que realice *web scraping*. Se descartó ya que no está tan enfocado al procesamiento de datos y las librerías no disponían del mismo número de funcionalidades que las de **Python**.

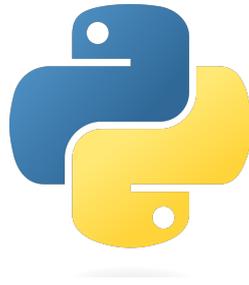


Figura 3.1: Logo de Python.

Aplicación móvil y aplicación web

Como debíamos desarrollar no solo la aplicación móvil, sino también una versión web de la misma, se quiso buscar una herramienta que nos sirviera para ambos casos. Además, teníamos un requisito extra. Es necesario que pueda ejecutarse tanto en **Android**⁷ como en **iOS**⁸.

Estas son algunas de las plataformas más famosas que nos permiten desarrollar para ambos sistemas operativos:

- **Flutter**: Es un *Software Development Kit* (SDK)⁹ desarrollado por **Google**¹⁰, está basado en el lenguaje de programación *Dart*¹¹.
- **React Native: Framework**¹² [21] de código abierto basado en la librería *React*¹³ [22], con él se puede usar *Javascript* o *Typescript*¹⁴ [23].
- **Xamarin**: Plataforma que nos permite crear aplicaciones para *Android*, *iOS* y *Windows* empleando la plataforma de *Microsoft .NET*¹⁵, en la que se pueden desarrollar múltiples y muy diversas aplicaciones.

⁵**JavaScript**: *JavaScript* es un lenguaje de programación pensado en principio para el lado cliente, es decir, para ser ejecutado en un navegador, está estrechamente relacionado con el desarrollo web.

⁶**Node.js**: Entorno que nos permite ejecutar *JavaScript* en el lado servidor del programa, en lugar de en el lado web.

⁷**Android**: Sistema Operativo que se utiliza en dispositivos móviles creado por la empresa *Android Inc*, que fue adquirida por *Google*

⁸**iOS**: Sistema Operativo desarrollado por la empresa *Apple* para sus dispositivos, como el *Iphone*, el *Ipad*...

⁹**Software Development Kit**: Conjunto de herramientas proporcionado usualmente por el fabricante de una plataforma de hardware, un sistema operativo (SO) o un lenguaje de programación

¹⁰**Google**: Compañía estadounidense especializada en productos y servicios relacionados con el internet. Conocidos por su motor de búsqueda con el mismo nombre.

¹¹**Dart**: Es un lenguaje de programación de código abierto, desarrollado por Google.

¹²**Framework**: Herramienta que ofrece componentes o soluciones para desarrollar de forma más rápida.

¹³**React**: Librería del lenguaje *JavaScript* para crear interfaces de usuario de forma sencilla.

¹⁴**Typescript**: Lenguaje de programación creado por *Microsoft* basado en *JavaScript*, pero introduce los tipos de variables y otras mejoras. Este se compila, es decir se traduce a *JavaScript* original.

¹⁵**Microsoft .NET**: *Framework* de *Microsoft* que permite construir soluciones *software* de forma rápida con independencia de la plataforma hardware donde se vayan a ejecutar.

Todas estas herramientas nos permiten el **desarrollo multiplataforma**¹⁶, pero con algunas restricciones, por ejemplo es necesario disponer de un ordenador de **Apple**¹⁷, como un *Mac* o un *Macbook*, o en su defecto, una licencia de desarrollador de *Apple*, como es el caso de **Flutter** y **Xamarin**, pero sin embargo para *React Native* existe una herramienta llamada **Expo**¹⁸ [24]. *Expo*, nos ayuda con algunos de los problemas que *React Native* nos trae, como por ejemplo, tener que realizar la configuración de los proyectos de forma separada para *Android* e *iOS*. Lo interesante de esta herramienta es que nos permite utilizar nuestro propio *iPhone* para probar la aplicación sin necesidad de una licencia de desarrollador.

Es por esto que finalmente nos decantamos por emplear **React** junto a **Next.js**¹⁹ [25] para la página web y **React Native** para el desarrollo móvil. Además el código se realiza de forma muy similar, gran parte se puede compartir. Con el fin de disponer de un diseño similar tanto en la página web como en la aplicación móvil, se eligió una librería llamada **Native Base** [26], en ella hay distintos componentes y utilidades que nos facilitan el desarrollo.

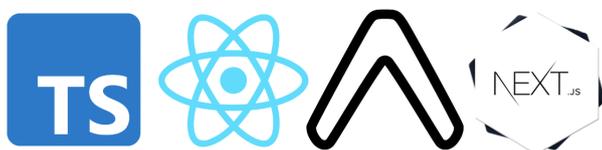


Figura 3.2: Logos de *Typescript*, *React* (y *React Native*), *Expo* y *Next.js* respectivamente.

En el futuro si quisiéramos eliminar la dependencia que tenemos con *Expo*, podríamos utilizar un comando del propio *framework* que separará nuestro proyecto y lo convertirá en un proyecto de *React Native*, pudiendo continuar nuestro desarrollo con normalidad.

Plataforma de *Back-end*

Como plataforma de *Back-end*, nos referimos a la parte del proyecto que el cliente no ve. En este caso será nuestra base de datos y una *API*²⁰ [27] que desarrollaremos para relacionar nuestros proyectos.

Para crear una *API* tenemos varias opciones, ya que estas pueden funcionar con distintos protocolos y arquitecturas. Por ejemplo, existen las *APIs* con un protocolo **SOAP** (*Simple Object Access Protocol* o en español, Protocolo de Acceso a Objetos Simples), cuya comunicación es a través de **XML**²¹, fue un protocolo muy importante pero necesita unos requisitos de seguridad y estructura que lo convierten en una opción lenta y compleja de crear.

¹⁶**Desarrollo multiplataforma:** Proceso mediante el cual se crea una aplicación en un lenguaje de programación que facilita su exportación y visualización en cualquier tipo de dispositivo

¹⁷**Apple:** Empresa estadounidense que produce equipos electrónicos, software y servicios en línea.

¹⁸**Expo:** Plataforma que condensa herramientas que facilitan el desarrollo para *React Native*.

¹⁹**Next.js:** Framework de desarrollo web de código que nos permite construir una web desde cero, utilizando *React* como su interfaz de usuario.

²⁰**API:** Interfaz de programación, nos permite conectar productos y servicios, independientemente de cómo estén implementados.

²¹**XML:** Lenguaje de marcado, similar a *HTML*. La información en *XML* se encuentra entre unas etiquetas que identifican esa información.

Para el desarrollo de aplicaciones móviles se suele utilizar una arquitectura *REST* [28]²², una API que no almacene información entre llamadas, que podamos ampliar los requisitos según necesitemos, y que cada **endpoint o controlador**²³ pueda ser atacado de forma independiente, entre otras características.

Continuando con la línea de trabajar en *Typescript*, se ha decidido utilizar **Node.js**, junto al *framework* de **Nest.js**²⁴. Esto se debe a que *Nest.js* nos permitirá manejar de forma muy sencilla la estructura de nuestra base de datos a la vez que creamos los *endpoints*. Definiendo las entidades de la base de datos creará las tablas y sus relaciones y es muy sencillo utilizar luego estos mismos modelos para trabajar, realizar búsquedas y relacionarlos entre sí. Existen alternativas muy populares como por ejemplo **Express.js**²⁵ para crear una API, pero *Nest.js* está construido encima de *Express*, así que es una herramienta más completa.

Además se utilizó como herramienta de apoyo **Firebase**²⁶, ya que allí podremos utilizar *Firebase Storage*²⁷ de forma gratuita para almacenar las fotos del usuario. Utilizaremos también *Firebase Authentication*²⁸, para manejar y registrar a los usuarios.



Figura 3.3: Logos de *Node JS*, *Nest JS* y *Firebase* respectivamente.

En cuanto a la base de datos, primeramente, enfocado a un desarrollo más profesional se decidió utilizar **PostgreSQL**²⁹, pero al tratarse de un motor más potente afectaba al rendimiento de la máquina ya que debía estar alojado en un servidor, por ello se optó por utilizar una base de datos sencilla en **SQLite**³⁰ para el prototipo ya que nos permite almacenar los datos de forma simple y no ocupa mucho espacio.

²² **REST**: Una API REST es una API que cumple los principios de diseño del estilo de arquitectura REST o transferencia de estado representacional. Le exige a la API cumplir unos requisitos, como por ejemplo que la procedencia de una petición no afecte a su resultado, que no almacenen información entre llamadas, entre otros.

²³ **Endpoint**: Punto de conexión entre la API y el cliente, se identifica un endpoint a través de una URL.

²⁴ **Nest.js**: Framework que nos ayuda a crear aplicaciones del lado servidor utilizando Typescript.

²⁵ **Express.js**: Entorno de trabajo para aplicaciones web para *JavaScript*, y sirve para desarrollar aplicaciones web y APIs.

²⁶ **Firebase**: Plataforma para el desarrollo de aplicaciones web, tiene varias herramientas como bases de datos, opciones de almacenamiento y otros, está desarrollada y mantenida por Google.

²⁷ **Firebase Storage**: Servicio de almacenamiento en la nube.

²⁸ **Firebase Authentication**: Servicio de autenticación.

²⁹ **PostgreSQL**: Sistema de gestión de bases de datos de código abierto.

³⁰ **SQLite**: Herramienta de software libre, es una librería escrita en C que nos permite gestionar una base de datos de forma simple.



Figura 3.4: Logo de SQLite.

3.1.2. Herramientas adicionales

Otras herramientas que han sido útiles para el desarrollo del prototipo son:

- **Visual Studio Code** [29]: Editor de código desarrollado por *Microsoft*, disponible en distintos Sistemas Operativos. En él están integradas múltiples utilidades como **Git**³¹ [30], resaltado de sintaxis... También dispone de una biblioteca de extensiones muy amplia en la que podemos integrar elementos extra como herramientas de formateado de código.
- **GitHub**: Plataforma de desarrollo colaborativo en la que se almacena el código fuente de nuestros proyectos utilizando el sistema de control de versiones llamado Git.
- **Heroku**: Plataforma en la que los desarrolladores pueden realizar el **despliegue**³² de sus aplicaciones sin preocuparse por el mantenimiento que existe detrás, en *Heroku* desplegamos nuestra *API*.
- **Vercel**: Plataforma en la nube que permite a los desarrolladores desplegar sus páginas web de forma instantánea, como nuestra web se desarrolló con *Next.js*, la integración con la misma es muy sencilla.
- **DB Browser for SQLite**: Software que nos permite visualizar el contenido de nuestra base de datos.
- Herramientas de planificación del desarrollo y redacción de la memoria:
 - **Notion**: *Notion* es un software de organización de tareas. En él se pueden realizar apuntes o podemos crear tableros *KanBan*, que es un método de organización de tareas basado en apuntar las tareas en tarjetas y dividir las por el estado en el que se encuentran.
 - **Overleaf**: Editor colaborativo de *LaTeX*³³ basado en la nube que se utiliza para escribir, editar y publicar documentos científicos.

³¹**Git**: Software de control de versiones de código abierto.

³²**Despliegue**: Supone mover aplicaciones de una instalación a otra. En este caso nos referimos a cuando pasamos nuestro entorno de desarrollo a una aplicación disponible en la web.

³³**Latex**: Sistema de composición de textos. Nos permite la creación de documentos escritos que tengan una alta calidad.

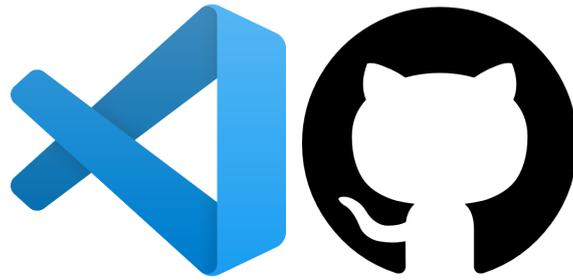


Figura 3.5: Logos de *Visual Studio Code* y *GitHub* respectivamente.

3.2. Web Scraper

Previo al desarrollo del prototipo se utilizó un *web scraper* para disponer de datos con los que trabajar y usar de modelos.

3.2.1. ¿Qué es el Web Scraping? Utilidad y controversia

Un *Web Scraper* es un software que se dedica a recopilar información de cualquier página de internet de forma similar a la que lo haría un humano. Estos programas entran en la página web, seleccionan y almacenan la información que nos interesa de cada página para posteriormente poder trabajar con ella [31]. Normalmente se emplean para realizar tareas que son sencillas pero repetitivas. En lugar de hacer nosotros una búsqueda en una página web 10 veces, ejecutamos el programa y este seleccionará la información por nosotros. Luego podemos procesarla mediante otro programa o realizar con ella la tarea que queramos.

Utilidad

Además de permitirnos una gran automatización, tiene muchas funcionalidades. Entre ellas destacan:

- **Detectar novedades:** Buscar información de la que no dispongamos en una página web, por ejemplo, nuevas noticias. Y almacenar estos datos nuevos en nuestra base de datos, o actualizarlos.
- **Obtención de información relevante:** Comprobar si un elemento que hemos guardado se mantiene igual que hace unas horas o días. Por ejemplo comprobar si el precio de un artículo ha variado de un día a otro (comparadores de precios) o hacer alertas.
- **Posicionamiento en los motores de búsqueda:** Podemos monitorizar qué palabras clave identifican a cada web y así mantener nuestra web en lo alto de las búsquedas.

Controversia

Aunque es usado por múltiples plataformas o empresas podemos ver que registrar o monitorizar páginas que no nos pertenecen puede fácilmente pasar a ser una actividad

ilegal. Hay que tener en cuenta los derechos de autor del contenido que estamos viendo, además de no recuperar datos que no sean visibles para el usuario final, o aquellos datos sensibles de usuarios de la otra web.

Si nosotros formamos parte de un sector muy competitivo, no nos interesa que la competencia pueda disponer de información de nuestra página. Es por ello que normalmente se verifica si el usuario es o no un robot³⁴:

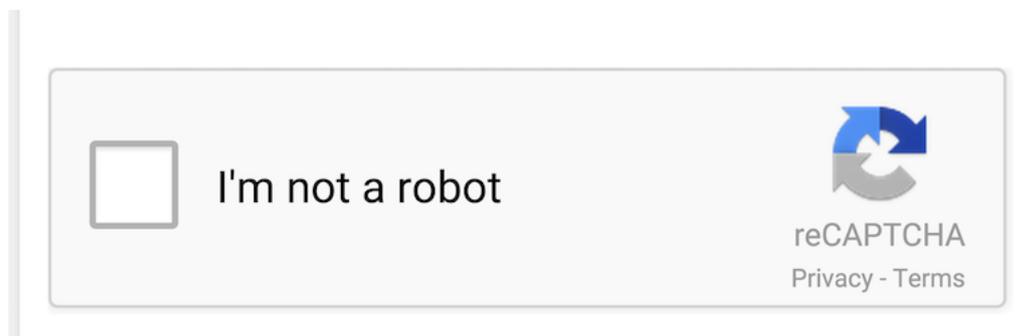


Figura 3.6: Re-Captcha común en las páginas web.

De esta forma se intenta combatir los programas de *Web Scraping*.

Existe también un fichero llamado `robots.txt` que se puede colocar en nuestra página web [31]. Sirve para gestionar estos software de rastreo. Podemos indicar en él por ejemplo, que no deseamos que nuestra página sea rastreada [32]. Sin embargo, este fichero no impedirá que estos programas funcionen.

En resumen, es una herramienta muy útil, pero puede llegar a ser problemática. En principio no es un proceso ilegal, pero podría llegar a convertirse en uno si no tenemos cuidado.

3.2.2. Utilidad en nuestro prototipo

Como primera aproximación a nuestra base de datos de libros, queríamos recopilar datos de una búsqueda en *Amazon*. Es por ello que creamos un *Web Scraper* que simulará a un usuario que busca "Libros de fantasía". Como nos podremos imaginar, esta búsqueda dará como resultado múltiples páginas.

A nosotros nos interesaba recoger los siguientes datos para poder utilizarlos de forma básica y posteriormente emplearlos para recuperar más datos:

- Nombre del libro.
- Autor o Autores.
- ISBN-10³⁵.
- ISBN-13.

³⁴**Captchas:** Son pruebas desafío-respuesta controladas por máquinas que son utilizadas para determinar cuándo el usuario es un humano o un programa automático.

³⁵**ISBN:** Identificador único de un libro, fue ampliado de 10 dígitos, a 13 dígitos, es por ello que existen dos tipos de identificador.

- Editorial.
- *URL*³⁶ de la imagen de la portada del libro.

El proceso de *scraping* que seguimos se compone de estos pasos:

- Recuperar en un fichero todos los enlaces a las páginas que nos devuelve la búsqueda.
- Abrir página a página recuperando y almacenando los enlaces a todos los libros de la misma.
- De cada libro extraemos la información deseada y lo almacenamos en un fichero *.csv*³⁷.
- Procesar el fichero para generar un *JSON*³⁸ que contendrá los datos del *CSV* que cumplan los requisitos (deben tener como mínimo el nombre, el autor y alguno de los ISBN).
- Posteriormente se creó un pequeño programa en *Javascript* que leyendo ese *JSON* añade la información a nuestra base de datos llamando a nuestra *API*.

3.3. Aplicación móvil

La aplicación móvil es el primer prototipo que realizamos. Se compone de las siguientes pantallas:

- **Login:** En esta pantalla podemos iniciar sesión o registrarnos en la aplicación mediante un correo electrónico y una contraseña. En el login veremos también el logotipo de la aplicación.

³⁶**URL:** Uniform Resource Locator o en español Localizador de recursos uniforme. Es la dirección única que se asigna a cada uno de los recursos disponibles de internet para que puedan ser localizados por el navegador.

³⁷**CSV:** En inglés *Comma-Separated Values*. Documento de formato abierto cuyos datos representan una tabla. Las columnas se separan por comas y las filas por saltos de línea.

³⁸**JSON:** *JavaScript Object Notation* o en español Notación de Objetos de JavaScript, es un formato ligero de intercambio de datos, a pesar de su nombre, es independiente del lenguaje de programación.

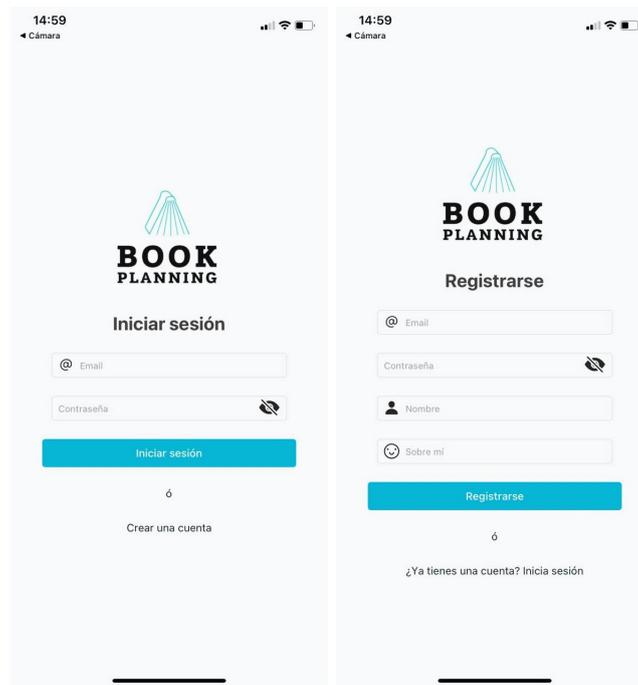


Figura 3.7: Login de la aplicación.

- Inicio:** En la pantalla de inicio disponemos de un apartado superior con nuestra foto de perfil y nuestro nombre. Aquí podremos ver una lista de los libros de los que estamos haciendo un seguimiento ahora mismo. Se presentan en un formato de lista, con los siguientes datos: Fecha de inicio de lectura, la fecha estimada de finalización, cuántas páginas tiene, cuántas hemos leído, progreso de lectura. Aquí podremos guardar el día y cuántas páginas hemos leído, también podremos marcarlos como finalizados presionando el icono de tic en la esquina superior derecha. Los libros que hemos finalizado están disponibles en la pestaña de *Terminados* allí veremos título del libro y el tiempo que nos llevó leerlo.

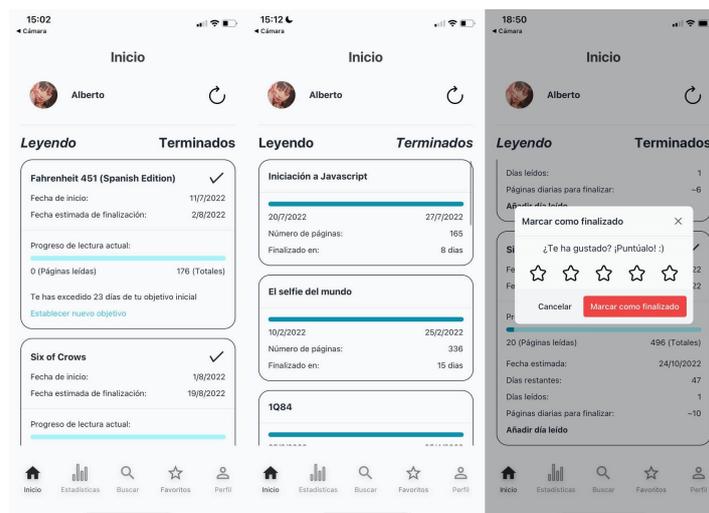


Figura 3.8: Página inicial de la aplicación al hacer login.

- Búsqueda:** Al entrar por primera vez en la pestaña de búsqueda nos pedirá que

aceptemos los permisos de acceso a la cámara. Esto se debe a que para realizar la búsqueda, podemos utilizar el nombre del libro o el autor, pero también disponemos de la opción de usar nuestra cámara para buscar un libro que tengamos en formato físico. Se realiza una llamada a la **Google Books API**³⁹ y nos dará un listado de los libros que ha encontrado con ese título, o el libro con el ISBN que hemos escaneado. También disponemos de un pequeño historial de las últimas búsquedas que hemos hecho. Cuando aparezca el listado, veremos su título, una imagen de portada si tiene y podremos guardarlo presionando el icono de marcador que aparece en la esquina superior derecha.

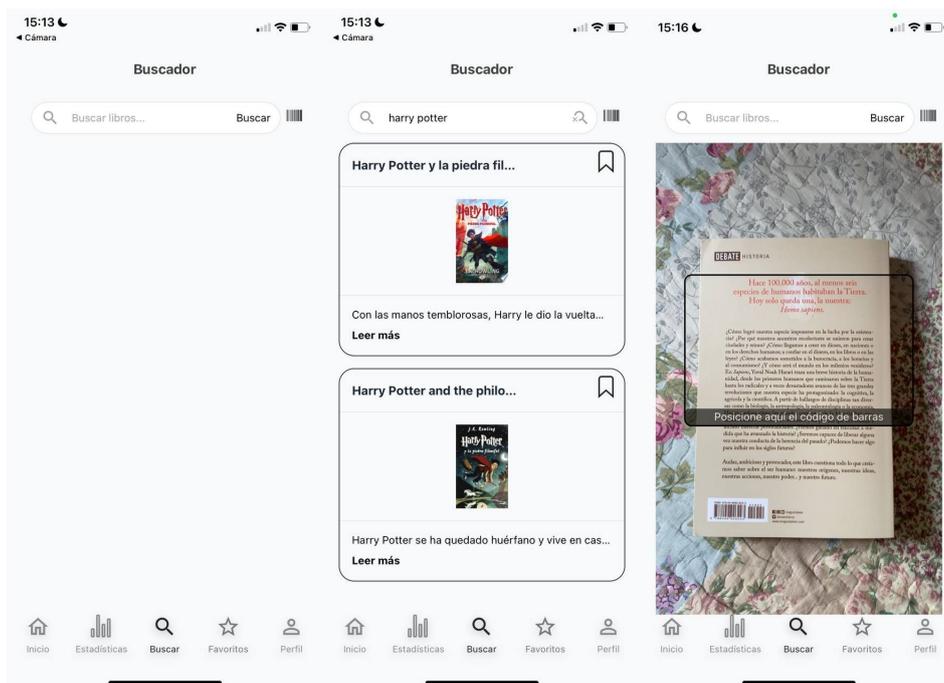


Figura 3.9: Página para realizar la búsqueda.

- **Estadísticas:** La pantalla de estadísticas contiene una colección de datos del usuario. Son los datos que se han recogido con el uso de la aplicación, número de libros leídos, páginas leídas, una media de páginas leídas al mes, media de libros al año... Y unas gráficas que representan los libros leídos cada mes, así como los libros que hemos leído en distintos años. Se añade un pequeño aviso al presionar el botón de **Cargar más**, para simular que nos encontramos en la versión gratuita de la aplicación.

³⁹ **Google Books API:** API de Google dedicada exclusivamente al manejo de los libros que se encuentran en la aplicación de Google Books.

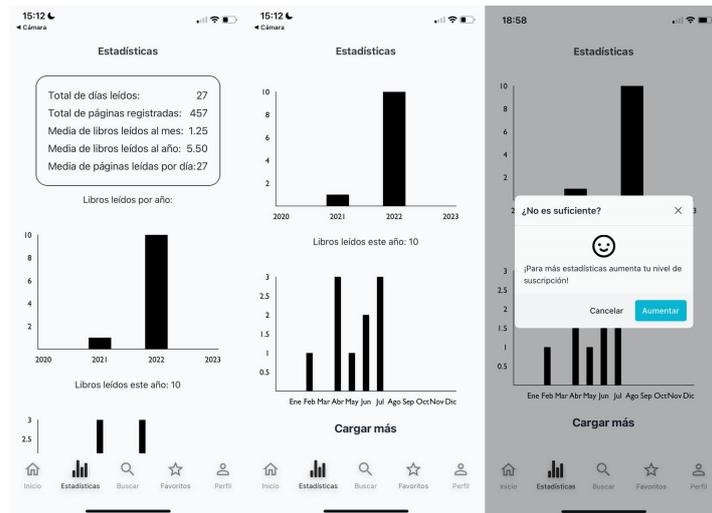


Figura 3.10: Página en la que vemos las estadísticas del usuario.

- Favoritos:** En esta pantalla disponemos de todos los libros que hemos marcado como favoritos. Es decir, aquellos que estamos leyendo, que ya hemos finalizado, etc. Para eliminar uno de estos libros debemos presionar en el botón con forma de marcador que se encuentra arriba a la derecha. Aquí es donde podremos establecer un objetivo de lectura. Presionando en el título del libro, nos preguntará en qué fecha queremos finalizar el libro, pero también hará una estimación de cuántas páginas diarias nos recomienda basado en los datos que hemos introducido en la aplicación. Cuantos más datos añadimos, más precisa será la estimación. Cuando aceptemos ese objetivo, el libro estará disponible en la página inicial para que podamos comenzar a añadir nuestra información.

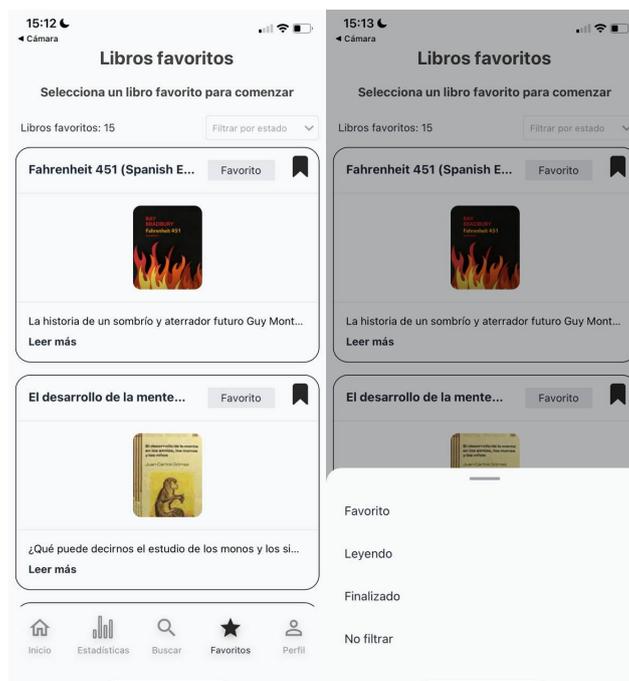


Figura 3.11: Pestaña de favoritos.

- **Perfil:** Aquí podremos realizar tareas relacionadas con la configuración de la aplicación. Disponemos de un apartado de editar perfil, en el que podremos subir una nueva foto y cambiar nuestro nombre, *email* o descripción y cerrar la sesión actual. También podremos cambiar la contraseña que dimos por primera vez a *Firebase*. Existen unas pestañas de sincronización con plataformas famosas de lectura en dispositivos electrónicos como *Amazon* o *Google Books*, pero no tienen funcionalidad. También disponemos de una opción de poner la aplicación en modo noche. El apartado de **contáctanos** podemos dejar un comentario para reportar un error u ofrecer ideas de mejora.

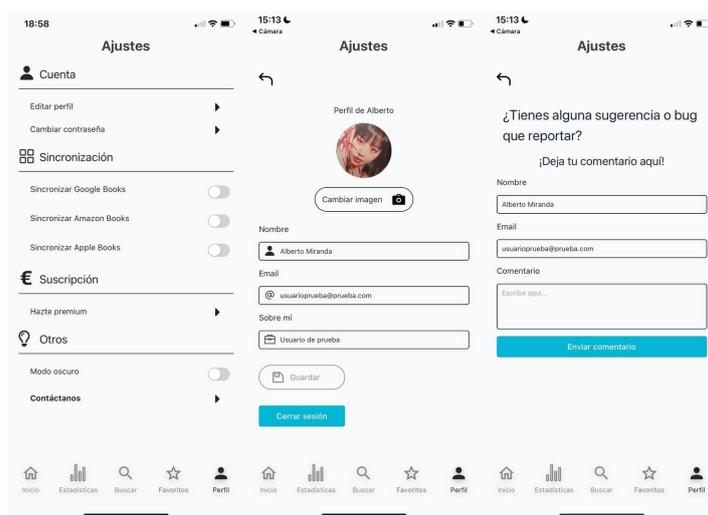


Figura 3.12: Pestaña del perfil del usuario.

3.3.1. Retos de lectura

Para establecer un nuevo reto de lectura, nos desplazamos a la pestaña de favoritos. Una vez allí, presionamos el título del libro y se nos abre un modal en el que podremos elegir en qué fecha queremos finalizar nuestro libro, o en su defecto, cuántas páginas queremos leer al día.

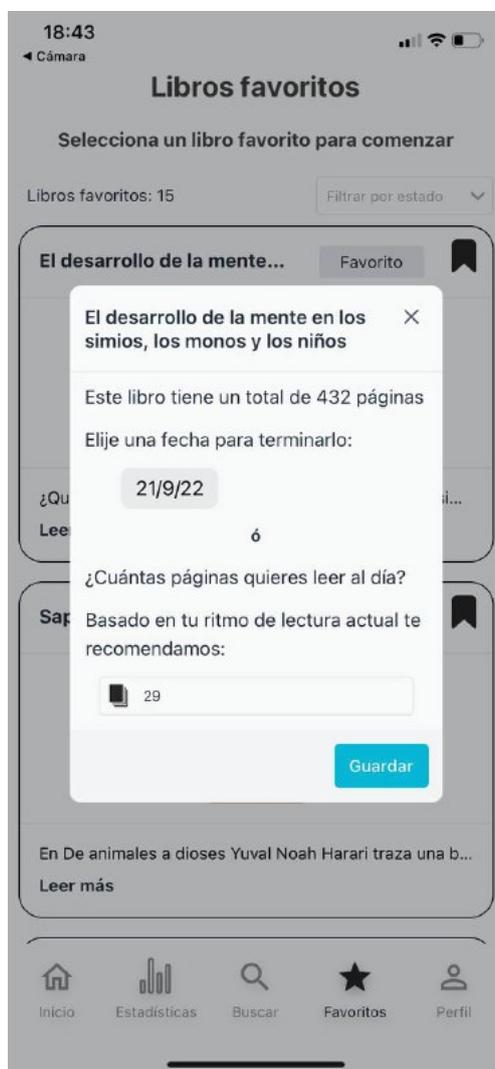


Figura 3.13: Elegir fecha para finalizar el libro.

Ahora veremos nuestro libro en la página de inicio, se nos dará una estimación de cuántas páginas al día debemos leer para finalizarlo en el tiempo que marcamos previamente.

El desarrollo de la mente en los simios, lo... ✓

Fecha de inicio: 6/9/2022

Fecha estimada de finalización: 21/9/2022

Progreso de lectura actual:

0 (Páginas leídas) 432 (Totales)

Fecha estimada: 21/9/2022

Días restantes: 15

Días leídos: 0

Páginas diarias para finalizar: ~29

Añadir día leído

Figura 3.14: Libro añadido a la página de inicio.

Si presionamos la opción de **Añadir día leído**, se nos abrirá un calendario y podremos seleccionar el día de hoy. Entonces, podremos escribir cuántas páginas han sido leídas. Por defecto se encuentran las que elegimos al inicio, pero podemos cambiarlas sin problema.



Figura 3.15: Añadimos un nuevo registro.

Una vez hecho esto se actualizarán los valores de la página. Ahora podemos ver cómo cambió el progreso del libro, así como las páginas diarias que debemos leer para finalizarlo en nuestra fecha deseada.

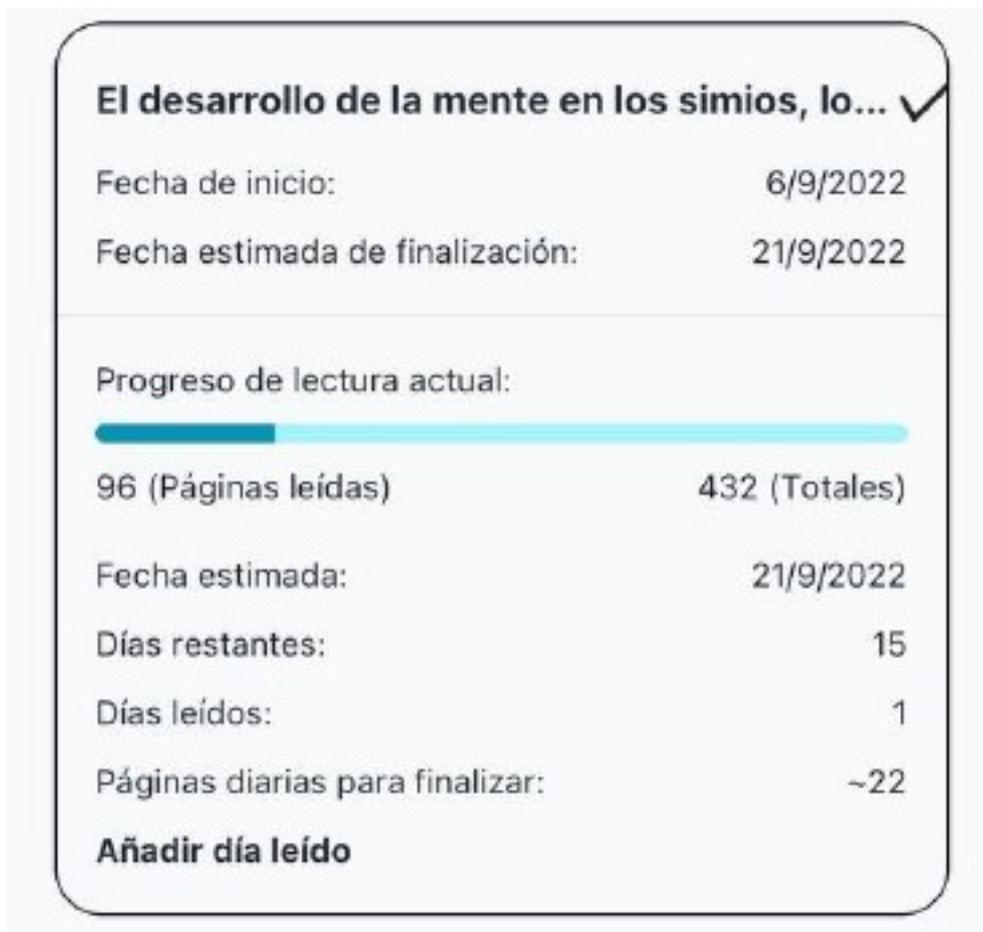


Figura 3.16: Elegir fecha para finalizar el libro.

Además, esta es la información que necesita el programa para poder calcular tu ritmo de lectura, es por ello que al querer iniciar otro reto, dispondremos de otra sugerencia de lectura, cuantos más datos apuntemos, mejor será la estimación.



Figura 3.17: Elegir fecha para finalizar el libro.

3.4. Página web

La página web, es una adaptación de la aplicación móvil. Es decir, existen las mismas pantallas, con una distribución ligeramente diferente, además de modificar ciertas

opciones para poder utilizarlas en *React*.

Algunas imágenes de la web:

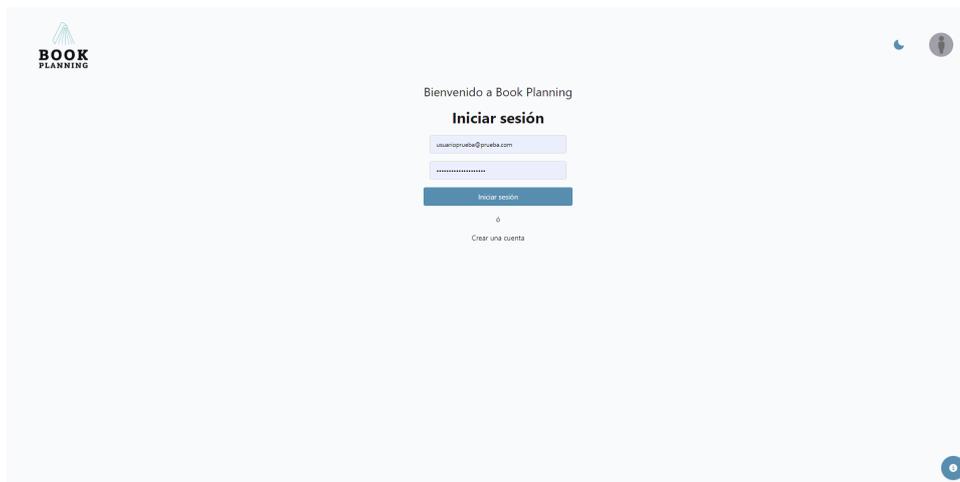


Figura 3.18: Login en versión web.

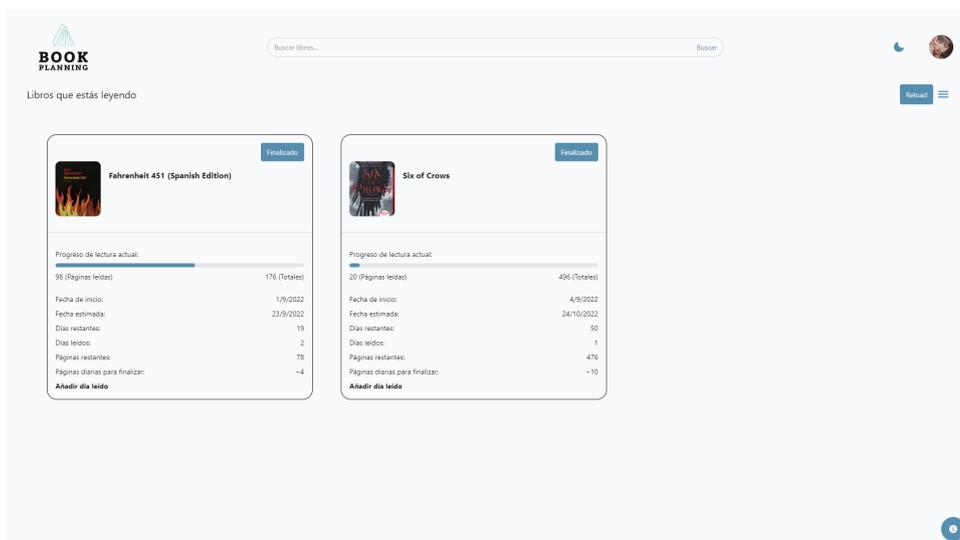


Figura 3.19: Página de inicio en versión web.

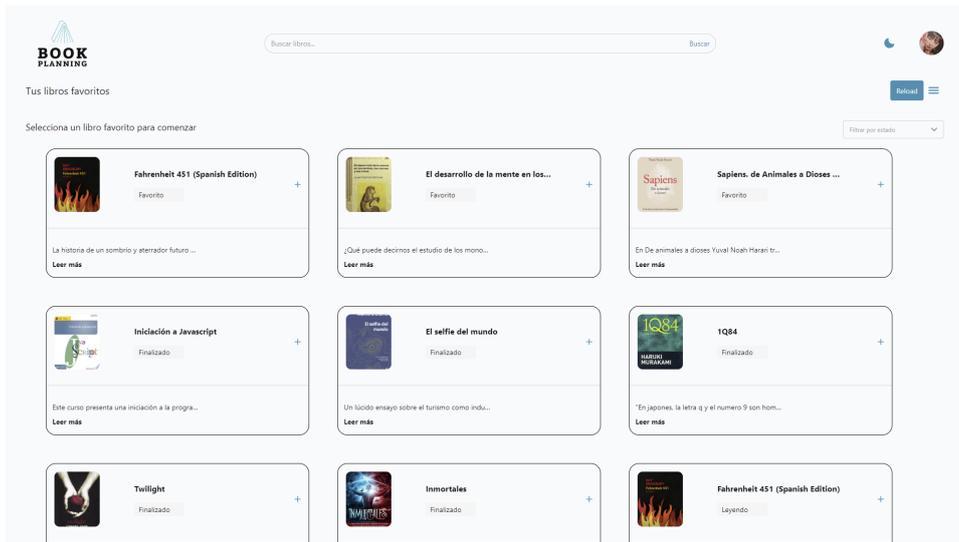


Figura 3.20: Imágenes de la búsqueda en versión web.

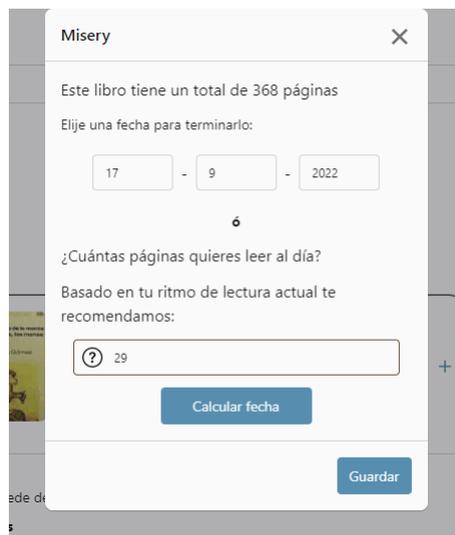


Figura 3.21: Establecer reto del libro guardado en web.

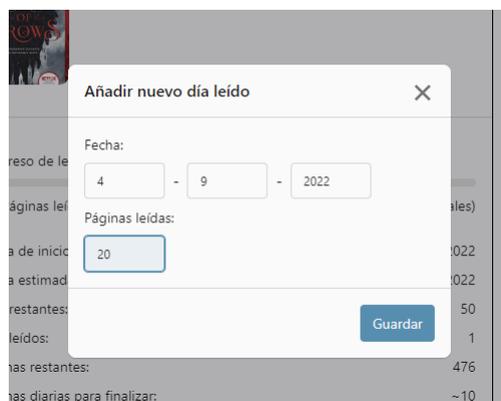


Figura 3.22: Añadir nuevo día leído.

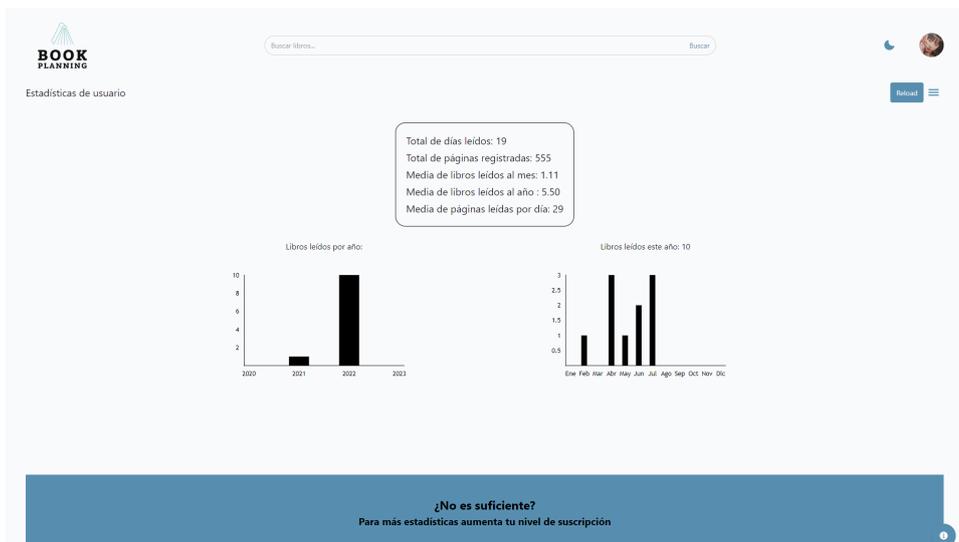


Figura 3.23: Estadísticas en la versión web.

CONTACTA CON NOSOTROS

Nombre

Alberto Test

Email

usuario@prueba@prueba.com

Comentario

Deja aquí tu comentario...

Enviar

Figura 3.24: Comentarios en la versión web.

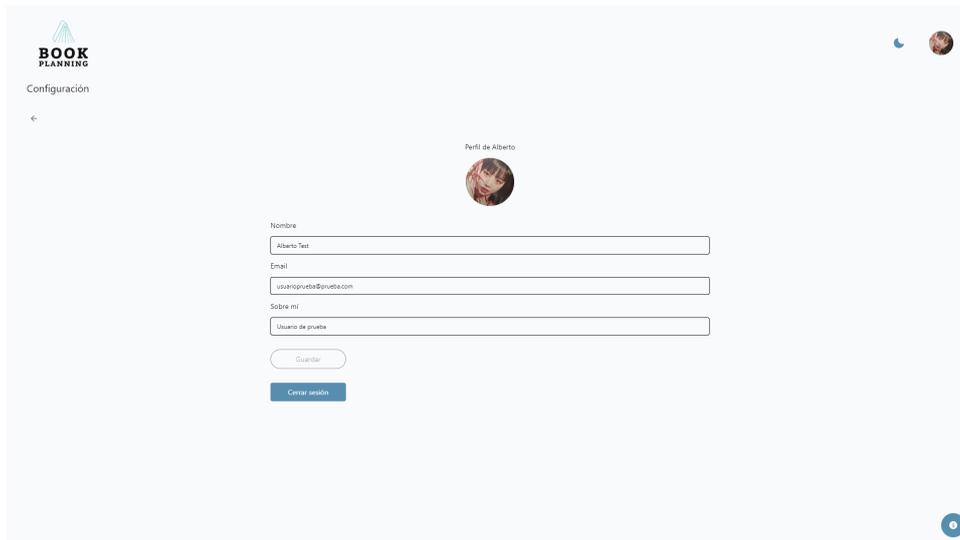


Figura 3.25: Perfil en la versión web.

Como se mencionó previamente la web es una adaptación, así que existen algunas funcionalidades la aplicación móvil que no están presentes en la página web:

- Calendario para seleccionar el día leído.
- Cambio de imagen de perfil.

3.5. Plataforma de back-end

3.5.1. Firebase

Firebase es una plataforma para desarrollo de aplicaciones web y móviles propiedad de *Google*. Tiene muchos componentes que podemos utilizar pero a nosotros nos interesan dos de ellos.

El primero es *Firebase Authentication*, con él realizaremos el login de los usuarios. Disponemos de una autenticación a través de correo electrónico y contraseña.

El segundo es *Firebase Storage*, con esta herramienta podemos incorporar un servicio de almacenamiento en la nube. Allí se alojarán las imágenes que los usuarios establezcan como foto de perfil. Este servicio guardará la imagen de forma segura y nos devolverá una *URL* con la que podremos acceder a ella.

3.5.2. Base de datos

Anteriormente mencionamos que como motor de base de datos hemos elegido *SQLite* ya que se trata de una base de datos con una implementación muy sencilla. Tiene sus limitaciones ya que los tipos que permite almacenar son reducidos, pero a cambio, disponemos de una base de datos que se almacena como un simple fichero con extensión *.db*.

Tabla de Libros

Aquí se almacenan los libros que los usuarios registran. De esta forma pueden ser reutilizados sin necesidad de realizar una búsqueda de nuevo. En esta tabla se almacenan los siguientes datos:

- **Id:** Identificador del libro generado automáticamente por nuestra base de datos.
- **Id de Google:** Identificador del libro según se identifica en la *Google Books API*.
- **Datos del Libro:** Información del volumen en formato *JSON*. Contiene datos como nombre, autor o autores, páginas, imágenes relacionadas con el libro, entre otros.

Tabla de Usuarios

En esta tabla guardamos datos básicos de los usuarios, para poder identificarlos tanto en nuestra aplicación como en *Firebase*.

- **Id:** Identificador del usuario, es el mismo código alfanumérico que asigna *Firebase* al usuario.
- **Nombre:** Nombre del usuario, se introduce al registrarse por primera vez y puede ser cambiado en la configuración.
- **Email:** Correo electrónico del usuario.
- **Descripción:** Breve descripción del usuario, no es obligatoria.
- **Foto de perfil:** Enlace a la foto de perfil que ha subido el usuario.

Tabla de libros que están leyéndose

En esta tabla se relacionan los usuarios con los libros que han guardado en su perfil.

- **Id:** Identificador generado por nuestra base de datos.
- **Fecha de inicio:** Fecha en la que comenzamos a realizar el seguimiento de la lectura del libro.
- **Fecha de finalización:** Fecha en la que finalizamos el libro.
- **Fecha estimada:** Fecha aproximada de finalización del libro.
- **Puntuación del usuario:** Puntuación que le dieron al libro.
- **Comentarios:** Comentario que el usuario deja al libro finalizado.
- **Estado:** Si el libro está guardado en favoritos, si ya se ha comenzado su seguimiento o si ya está finalizado.
- **Clave foránea⁴⁰ - Id Usuario:** Id del usuario.

⁴⁰**Clave foránea:** Columna en una tabla que hace referencia a una columna de otra tabla. Sirve para establecer relaciones entre ellas.

- **Clave foránea - Id Libro:** Id del libro.
- **Fechas leídas:** Fechas que el usuario ha guardado un número de páginas leídas.
- **Páginas diarias a leer:** Páginas que el usuario debería leer al día para finalizar el libro a tiempo.
- **Páginas leídas actualmente:** Número actual de páginas leídas.

Tabla de Comentarios

En esta tabla almacenamos los comentarios que el usuario ha enviado.

- **Id:** Identificador del comentario, generado automáticamente por la base de datos.
- **Nombre:** Nombre del usuario.
- **Email:** Correo electrónico del usuario.
- **Comentario:** Texto que quiere enviar el usuario.
- **Fecha:** Día y hora en la que se guardó el comentario.

Aquí podemos ver un esquema de las distintas entidades que componen nuestra base de datos.

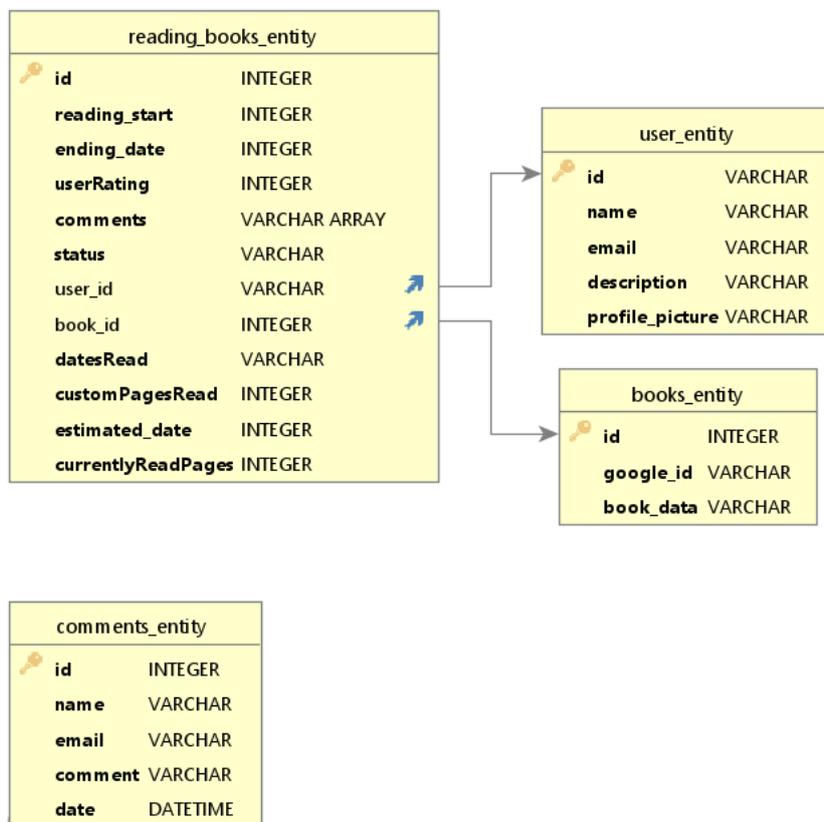


Figura 3.26: Esquema de la base de datos.

La creación de las distintas tablas se realizó mediante la API, como veremos a continuación.

3.5.3. API

La API es nuestro servidor *back-end*. Es la encargada de hacer de intermediaria entre las aplicaciones y nuestra base de datos u otras *APIs*. *Nest.js* nos permite utilizar un proceso llamado **Code First**⁴¹ para crear nuestra base de datos a partir de las **entidades**⁴², que en nuestro caso son clases. Esto es posible haciendo uso de **Type ORM**⁴³. Nosotros estableceremos un módulo que será el que conectará con nuestra base de datos, pero a partir de ahí, no trabajaremos directamente con ella, sino con las entidades antes mencionadas[33].

Ahora veremos un ejemplo de una entidad que representa un usuario:

```
1 // Fichero UserEntity.ts
2 import { Column, Entity, PrimaryColumn } from "typeorm";
3 @Entity()
4 export class UserEntity {
5     @PrimaryColumn()
6     id: string;
7
8     @Column()
9     name: string;
10
11     @Column()
12     email: string;
13
14     @Column()
15     description: string;
16
17     @Column({name: 'profile_picture'})
18     profilePicture: string;
19 }
```

⁴¹**Code First**: O también código primero. Es una metodología para el desarrollo de aplicaciones en las que no se tiene una base de datos creada. Esta será creada a partir de un modelo representado mediante una clase. El nombre de la clase es el equivalente al nombre de la tabla, y sus atributos serán los campos de la misma.

⁴²**Entidades**: Tipo de recurso que se asocia directamente con las tablas en nuestra base de datos, y nos permite utilizarlo en nuestra aplicación.

⁴³**Type ORM**: Un ORM (*Object-Relational Mapping*) es un mecanismo que nos permite trabajar con distintos objetos de forma abstracta, sin importarnos el origen de los datos que los forman. Por ejemplo, sin importarnos si su base de datos cambia. *Type ORM* es un tipo de ORM compatible con *Node.js*

La API se ha diseñado con un estilo de arquitectura hexagonal. Este busca separar el interior y el exterior del software. En el interior definimos el dominio que son nuestros modelos. Los modelos intentan representar la aplicación que se va a crear. Este tipo de diseño es el Diseño guiado por el dominio (*Domain Driven Design*⁴⁴). En nuestro caso tenemos dominios claros como libros o usuarios. El exterior está compuesto por puntos de entrada, como los controladores, y puntos de salida, como la base de datos.

Es por ello que cada elemento está organizado de la siguiente forma:

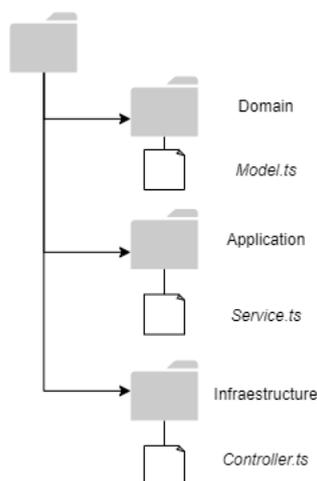


Figura 3.27: Árbol de directorios de un elemento en la API.

- **Dominio:** En el dominio se encuentran los modelos y las entidades que representan al elemento.
- **Aplicación:** Aquí exportamos los servicios que tenemos. Definimos por ejemplo, las acciones que realizaremos con estos datos, crear un nuevo dato, buscarlo...
- **Infraestructura:** En la infraestructura tenemos la cara externa, por ejemplo, los controladores. Estos utilizarán los servicios para realizar las acciones que sean pertinentes.

Los controladores que se han definido son los siguientes:

- **Books:** En este conjunto de *endpoints* se manejan los libros, se pueden añadir libros a la base de datos y recuperarlos para usarlos en las aplicaciones.
- **Comments:** Los comentarios que se envían a través de las aplicaciones llaman a este *endpoint* para ser almacenados.
- **Reading books:** Permiten el manejo de los libros que guarda el usuario, para poder añadirlos, buscar los que están relacionados con el usuario, actualizarlo y también eliminarlos.
- **Search:** Nos permite realizar una búsqueda de un libro en la *Google Books API*.

⁴⁴**Domain Driven Design:** Abreviado DDD. Es una aproximación al desarrollo de software que busca relacionar el desarrollo con los conceptos del negocio y sus modelos.

- **Statistics:** Estadísticas del usuario, a través de este *endpoint* obtenemos unas estadísticas básicas de la lectura que ha realizado este usuario, cuántas páginas ha leído, cuántos libros, en qué años etc.
- **User:** Manejo de los usuarios, crear uno nuevo y actualizarlo.

Al haber sido desarrollada con *Nest.js* nos permite configurar con facilidad un módulo llamado **Swagger UI**⁴⁵. En él podemos ver los controladores que hemos creado e interactuar con ellos desde una interfaz gráfica.

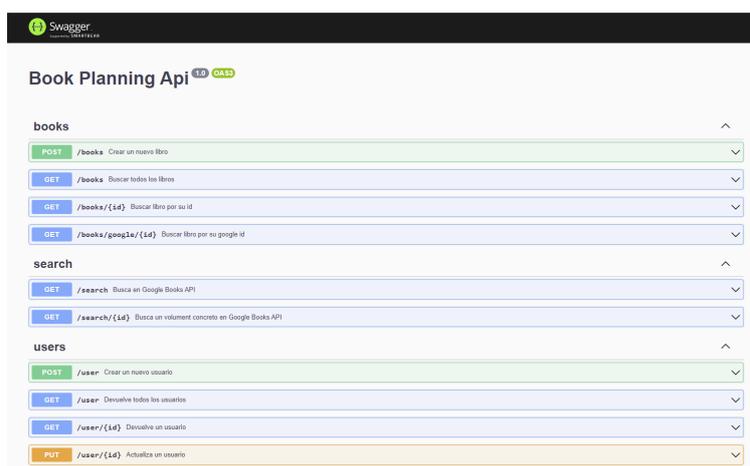


Figura 3.28: Captura de pantalla de *Swagger UI*.

3.6. Resultados del desarrollo

Aquí encontramos una serie de enlaces a los recursos desarrollados para el prototipo.

- **Repositorio Aplicación móvil:** <https://github.com/luzvare/book-planning>
- **Repositorio Aplicación web:** <https://github.com/luzvare/book-planning-web>
- **Repositorio API:** <https://github.com/luzvare/book-planning-api>
- **Repositorio del Web Scraper:** <https://github.com/luzvare/book-planning-scraping>
- **Aplicación web:** <https://book-planning-web.vercel.app/>

⁴⁵ **Swagger:** Swagger es un conjunto de herramientas que nos permiten construir, documentar y utilizar servicios *REST*. UI es la abreviación de *User Interface*, Interfaz de usuario en español.

Capítulo 4

Estudio de la viabilidad económica del proyecto

El prototipo es una primera muestra de la idea que existe detrás de la aplicación, pero para poder realizar un estudio económico con un poco más de profundidad simularemos que disponemos de recursos ilimitados su desarrollo. De esta forma, se nos pueden ocurrir múltiples ideas que mejoren la calidad de la misma. Estas son algunas de ellas:

- Accesibilidad.
- Conexión y sincronización con *E-readers* actuales.
- Inteligencia artificial.
- Ludificación de los retos de lectura.
- Convertir la aplicación en una red social.

Explotar la accesibilidad es fundamental para una aplicación que quiere atraer a la mayor cantidad de personas posibles, una aplicación de estas características debe tener un diseño claro e inclusivo, facilitar la lectura de sus apartados ofreciendo contrastes, fuentes, tamaños en la interfaz adecuados, opciones de **text to speech**¹ que otras aplicaciones no tienen. Esto último enriquece la experiencia atrayendo a un público que es desplazado por muchos diseñadores.

Tener que trasladar la información de un lado a otro porque no existe una integración entre dos tecnologías es una actividad muy tediosa. Si se facilita este proceso, dispondríamos de nuestra lista de lectura al día sin ningún problema. Es por ello que integrar las aplicaciones más famosas, como *Amazon Kindle*, *Google Books*, o *Nook* con la nuestra, sería una motivación más para utilizarla. Podríamos importar la lista de lectura, junto a sus puntuaciones, sus lecturas pendientes etc. El usuario sólo tendría que iniciar sesión con su cuenta, aceptar que se realice esta sincronización y ya podría disfrutar del uso de ambas. A su vez podemos aprovechar otras herramientas, como los relojes inteligentes para integrar las notificaciones y al igual que nos recuerdan que bebamos agua o nos levantemos, nos recuerden también que nos toca leer unas páginas al día.

Teniendo a *Amazon* como el portador de la mayor plataforma de libros electrónicos, no podemos olvidarnos de aprovechar el **Big Data**² a nuestro favor. Toda la información que

¹**Text to speech**: Tecnología que convierte el texto a voz natural.

²**Big Data**: Como su nombre indica, es una cantidad de datos de grandes dimensiones relacionada con un tema, que debe ser procesada para inferir de estos datos nueva información.

podamos recuperar y procesar, ya sea a través del *web scraping* o del propio uso que se haga de la aplicación, serán los indicadores que nos dirán qué buscan los usuarios, qué es lo que más se lee, cual es el objetivo que más se quiere cumplir... Toda esa información que obtengamos se verá reflejada en el futuro de la aplicación, pero también en un **Asistente Virtual**³ que se podría desarrollar para la misma. Al comienzo sólo nos guiará en el uso de la aplicación, pero poco a poco se irá acomodando al lector, ofreciéndole las mejores recomendaciones para él, tanto en libros como autores, también adecuará los objetivos de forma personalizada. Este punto es importante, porque aplicaciones como **Tik Tok**⁴ nos han demostrado que nos encanta que una aplicación nos muestre justo lo que queremos ver.

El cometido de la aplicación es ofrecer una serie de herramientas para poder crear un hábito lector. Ampliar el catálogo de objetivos podría ser un primer paso, no sólo páginas diarias o una fecha como tiempo límite, sino otro tipo de objetivos como tiempo de lectura diario, o capítulos diarios para tener opciones que le puedan gustar más a cada tipo de usuarios. Pero en un proyecto profesional es posible explotar un campo más interesante: la **ludificación**⁵ de nuestra aplicación. Primero que los usuarios puedan seguirse y ver alguna información de los demás. Disponer de un sistema de puntos, que aumente según completamos retos, leamos libros, alcancemos objetivos... Para así obtener recompensas. Esto podría ser muy beneficioso para el aprendizaje, ya que esta estrategia puede mantenernos motivados y mejorar nuestro rendimiento. Además nos permite establecer una clasificación de jugadores, generando una competición entre los mismos. Esto nos daría la posibilidad también de intercambiar estos puntos obtenidos por algunas ventajas, ya sea dentro de la aplicación o fuera.

Por último, un paso más que enriquecería el proyecto, sería que fuera más allá de ser simplemente de ayuda para los lectores. A partir de la aplicación, se podría crear una red social cuyo punto común sea la lectura. Sí, en nuestra aplicación dispondríamos de usuarios. Pero, por qué no ampliar esto a tener distintos tipos de usuarios, no sólo podríamos ser consumidores, sería muy atractivo que los autores pudieran crear cuentas, distribuir novedades, actualizar a sus seguidores de nuevos libros, enlaces de compra a sus libros, a otras redes sociales en las que actualicen, pudiendo estar al tanto de las novedades de forma casi directa. Siendo más ambiciosos, no sólo autores podrían formar parte de la red, sino otro tipo de usuarios que ofrezcan distintos servicios para los autores. Editores por ejemplo o traductores. Todos estos perfiles profesionales podrían relacionarse con otras redes también de profesionales, como **LinkedIn**⁶, para acreditar sus cualidades.

4.1. Modelo de comercialización

El modelo de comercialización, es aquello que nos va a permitir obtener ingresos con el uso de la aplicación. Para ello tendremos dos versiones de la aplicación y cada uno

³**Asistente Virtual**: Programa o software que ayuda a los usuarios a realizar tareas de forma más sencilla.

⁴**Tik Tok**: Red social para compartir vídeos cortos y en formato vertical propiedad de la empresa china ByteDance.

⁵**Ludificación**: Aplicar técnicas relacionadas con juegos a programas o actividades que no sean recreativas para mejorar la participación o el aprendizaje.

⁶**LinkedIn**: Red social dedicada al uso laboral, pudiendo compartir nuestra experiencia y contactar con otros profesionales.

tendrá su modo de comercialización.

4.1.1. Versión gratuita

En el modelo gratuito dispondremos de una versión limitada de la misma. Será lo suficientemente avanzada como para poder probar la aplicación, evitando las funcionalidades de mayor coste. Obtendremos beneficios a través de publicidad. Como por ejemplo, haciendo uso de **Google AdSense**⁷ o **Google AdMob**⁸.

4.1.2. Versión premium

El modo *premium* dispondrá de todas las ventajas de uso de la aplicación. Y funcionará como una suscripción que se podrá pagar por rangos de tiempo.

- Plan mensual: 4.99 €.
- Plan semestral: 19.99 €.
- Plan anual: 35.99 €.

Podemos ver que según sube el plan, nos saldrá mas rentable la suscripción. De esta forma muchos usuarios podrán probar la versión gratuita y decidir si les interesa cambiar a la modalidad *premium*.

Posteriormente, si la aplicación tuviese éxito, podrían existir afiliaciones con otras empresas de mayor envergadura.

4.2. Desarrollo del proyecto

En este apartado vamos a ver cómo será el proceso de desarrollo del proyecto, qué recursos estarán involucrados en el mismo, ya sea materiales o humanos, y de qué forma se divide este desarrollo. Esto se hará utilizando una herramienta llamada **ProjectLibre**⁹. Esta nos permitirá crear un proyecto y definir sus recursos, costes y tareas, cada una con sus responsables y su duración.

El proyecto se encuentra dividido en cuatro fases:

- **Análisis:** La primera etapa del desarrollo consistirá en definir el problema que se quiere resolver y definir el alcance del sistema que se va a crear. En esta fase se reunirán los requisitos del producto, qué funcionalidades tendrá y cuáles no. Así como las necesidades que va a cubrir. En este periodo se realizará un Acta de Constitución del Proyecto.
- **Diseño:** En esta fase no sólo se realizará un diseño de la interfaz del usuario o de qué casos de uso¹⁰ existirán en la aplicación. También se definirá con qué herramientas se va a realizar el proyecto. Se llevará a cabo un análisis del hardware y del software

⁷**Google AdSense:** Producto de la red de publicidad de Google. Nos permite obtener ingresos colocando anuncios en nuestros sitios web.

⁸**Google AdMob:** Producto de la red de publicidad de Google. Nos permite obtener ingresos colocando anuncios en nuestra aplicación móvil.

⁹**ProjectLibre:** Software de administración de proyectos de código abierto.

¹⁰**Casos de uso:** Descripción de una actividad o sistema.

que se usará para la realización de la aplicación móvil, web y la API. Además se elegirá un sistema de bases de datos y se diseñará la estructura de la misma. Es decir se definirá el proceso completo de desarrollo.

- **Desarrollo:** Cuando tengamos el proyecto ya diseñado y todas las tareas a realizar, comienza la fase de desarrollo. En ella se implementarán las arquitecturas definidas. En este proceso de desarrollo se creará un entorno de pruebas para probar el software creado y su desempeño. Normalmente se emularán los procesos que se llevarán a cabo cuando la aplicación pase a producción para comprobar que el funcionamiento es el esperado.
- **Pruebas:** A lo largo del desarrollo se irá comprobando que el funcionamiento es correcto, pero aun así es necesaria una fase de pruebas final. En ella no sólo se probará el funcionamiento, sino cómo se desempeña el hardware y el software. Esta fase se realizará en un entorno de pruebas. Normalmente un es una copia del entorno de producción, en el que podemos probar el funcionamiento de todos los sistemas antes de que estén disponibles de forma pública.
- **Despliegue:** El software desarrollado pasará a un entorno de producción. Esto significa que aumentarán las prestaciones de nuestros servidores y la aplicación, la página web y todos los servicios pasarán a poder ser utilizados por el público general.
- **Mantenimiento:** Cuando la aplicación se encuentre abierta de cara al público, no termina su ciclo de vida. En la fase de mantenimiento se seguirán resolviendo problemas que surjan con el uso, se monitorizará el desempeño de los servidores y otras herramientas que se usen, se realizarán las mejoras necesarias o se añadirán funcionalidades extra. También se actualizará el software necesario reducir el número de posibles incidencias.

Para realizar la organización del proyecto utilizamos el programa anteriormente mencionado **ProjectLibre**, en él se definen todas las tareas y se le asignan los recursos necesarios para ser completadas y así poder estimar una duración del proyecto, así como un coste aproximado del mismo.

En la siguiente figura podremos ver las tareas a realizar:

	Costo	Nombre	Duración
1	134706,49 €	BookPlanning	183 days?
2	8808,00 €	Fase de Análisis	32 days?
3	2320,00 €	Análisis de requisitos	10 days
4	4248,00 €	Análisis de funcionalidades	15 days
5	2240,00 €	Acta de Constitución del Proyecto	7 days?
6	35840,00 €	Fase de Diseño	49 days?
7	14280,00 €	Diseño de arquitectura	21 days?
8	1960,00 €	Arquitectura de la API	7 days
9	1960,00 €	Arquitectura página Web	7 days
10	1960,00 €	Arquitectura aplicación móvil	7 days
11	4200,00 €	Arquitectura Web scraper	15 days?
12	4200,00 €	Arquitectura asistente con IA	15 days
13	8120,00 €	Diseño estructura de BBDD	29 days?
14	4200,00 €	Estructura bases de datos	15 days?
15	1960,00 €	Motor de bases de datos	7 days
16	1960,00 €	Servidor de base de datos	7 days
17	11200,00 €	Diseño UI/UX	42 days
18	5600,00 €	Casos de uso - Diagramas UML	14 days
19	1400,00 €	Diseño página web	7 days
20	1400,00 €	Diseño aplicaciones móviles	7 days
21	2800,00 €	Accesibilidad	14 days
22	2240,00 €	Plan de dirección del proyecto	7 days
23	40168,99 €	Fase de Desarrollo e Implementación	97 days?
24	15120,00 €	Implementación del FrontEnd	63 days
25	1680,00 €	Control de usuarios	7 days
26	1680,00 €	Estadísticas	7 days
27	1680,00 €	Métodos de búsqueda	7 days
28	1680,00 €	Manejo de libros	7 days
29	3360,00 €	Asistente virtual	14 days
30	1680,00 €	Objetivos y ludificación	7 days
31	3360,00 €	Accesibilidad	14 days
32	17688,99 €	Implementación del BackEnd	40 days?
33	1680,00 €	Desarrollo del web scraper	7 days?
34	4200,00 €	Desarrollo de la IA del asistente	15 days
35	3920,00 €	Implementación y despliegue de la BBDD	14 days
36	7888,99 €	Desarrollo de la API	33 days
37	1680,00 €	Manejo de usuarios	7 days
38	3360,00 €	Manejo de Libros	14 days
39	1680,00 €	Estadísticas	7 days
40	1168,99 €	Incorporar APIs externas	5 days
41	3360,00 €	Unificación FrontEnd + BackEnd	14 days
42	4000,00 €	Fase de Pruebas	20 days
43	4000,00 €	Desempeño del hardware y software	20 days
44	1200,00 €	Fase de Despliegue	5 days
45	1200,00 €	Pasar infraestructura a producción	5 days
46	0,00 €	Fase de Explotación y Mantenimiento	0 days

Figura 4.1: Tareas definidas para el desarrollo del proyecto.

Estas tareas y su orden se ven reflejadas en el **diagrama de Gantt**¹¹.

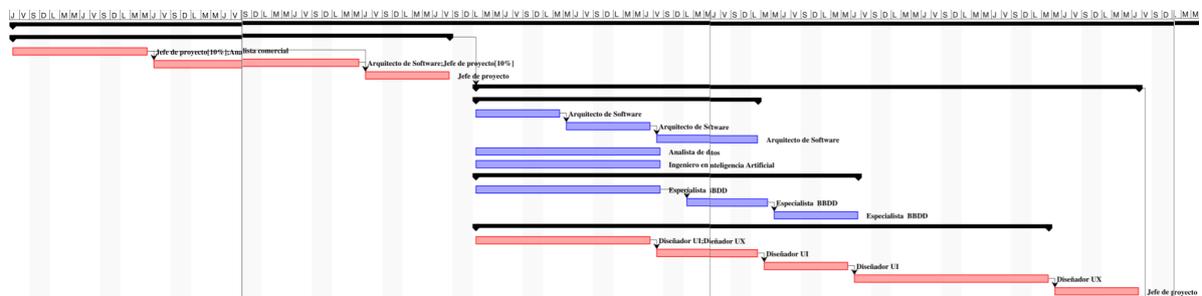


Figura 4.2: Primera parte del diagrama de Gantt.

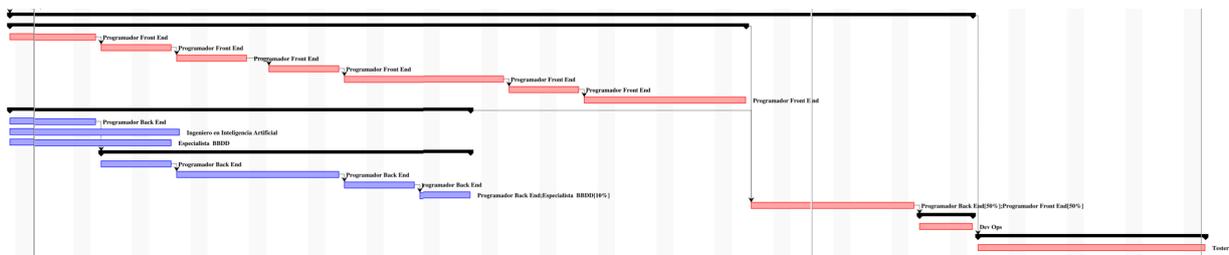


Figura 4.3: Segunda parte del diagrama de Gantt.

¹¹ **Diagrama de Gantt**: Herramienta gráfica que expone el tiempo de dedicación que se prevé para el desarrollo de una actividad o un conjunto de actividades.

4.3. Coste y duración del proyecto

Una vez asignadas las tareas y sus responsables, podemos ver el coste total del proyecto.

	Ⓜ	Nombre	Tipo	Etiqueta material	Iniciales	Tasa Estandar	Tasa sobretiempo	Devengado en	Calendario Base
1	👤	Analista de datos	Trabajo		A	35,00 €/hora	61,25 €/hora	Prorrateado	Estandar
2	👤	Arquitecto de Software	Trabajo		AS	35,00 €/hora	61,25 €/hora	Prorrateado	Estandar
3	👤	Jefe de proyecto	Trabajo		JP	40,00 €/hora	70,00 €/hora	Prorrateado	Estandar
4	👤	Diseñador UI	Trabajo		DUX	25,00 €/hora	43,75 €/hora	Prorrateado	Estandar
5	👤	Programador Back End	Trabajo		PBE	30,00 €/hora	52,50 €/hora	Prorrateado	Estandar
6	👤	Programador Front End	Trabajo		PFE	30,00 €/hora	52,50 €/hora	Prorrateado	Estandar
7	👤	Especialista BBDD	Trabajo		PBDD	35,00 €/hora	61,25 €/hora	Prorrateado	Estandar
8	👤	Tester	Trabajo		T	25,00 €/hora	43,75 €/hora	Prorrateado	Estandar
9	👤	Programacion - Muebles Varios	Material		P	357,00 €		Prorrateado	
10	👤	Oficina - Muebles	Material		MU	1200,00 €		Prorrateado	
11	👤	Oficina - Varios	Material		O	166,00 €		Prorrateado	
12	👤	Mac	Material		M	5499,00 €		Prorrateado	
13	👤	MacBook Pro 2021	Material		MP	2499,00 €		Prorrateado	
14	👤	Programacion - Varios	Material		V	401,00 €		Prorrateado	
15	👤	Microsoft Teams	Material		MT	10,50 €		Prorrateado	
16	👤	Dev Ops	Trabajo		D	30,00 €/hora	52,50 €/hora	Prorrateado	Estandar
17	👤	Analista comercial	Trabajo		A	25,00 €/hora	43,75 €/hora	Prorrateado	Estandar
18	👤	Ingeniero en Inteligencia Artificial	Trabajo		I	35,00 €/hora	61,25 €/hora	Prorrateado	Estandar
19	👤	Diseñador UX	Trabajo		D	25,00 €/hora	43,75 €/hora	Prorrateado	Estandar

Figura 4.4: Tabla de recursos materiales y humanos para el desarrollo del proyecto.

Se estima que la duración del proyecto será de 183 días laborables, 267 días naturales, teniendo una duración aproximada de 9 meses, con una jornada de 8 horas diarias de lunes a viernes. Tendrá un coste de 134.706,49 €.

4.4. Estimación del retorno de la inversión

Una vez la aplicación ha sido desarrollada y se encuentra en el mercado, hemos de saber aproximadamente en qué punto alcanzaremos el **Retorno de la Inversión**.

Este cálculo se realiza poniendo en común los gastos que ha supuesto su desarrollo, los gastos de mantenimiento, mensuales o anuales y los ingresos que se obtienen por su uso. Estimando de esta forma una semana en la que se recuperará la inversión realizada.

Para este cálculo, se asume normalmente que la afluencia de usuarios será exponencial. En nuestro caso se hicieron varios cálculos comenzando por los usuarios.

Usuarios

Como se mencionó con anterioridad tenemos usuarios gratuitos y *premium*. Del tipo de usuarios *premium* tenemos cuentas que se renuevan mensualmente, semestralmente y anualmente. Partimos de la base de que los usuarios irán de forma descendiente siguiendo ese orden ya que la mayoría de usuarios utilizarán la aplicación con un plan gratuito. En base a eso simulamos un crecimiento de los mismos.

A	B	C	D	E	F	G	H	I
Usuarios								
Semana	Gratuito	Mensual	Acumulado mensual	Bienal	Acumulado bienal	Anual	Acumulado anual	Usuarios acumulados
1	20	5	5	1	1	1	1	7
2	124	7	12	1	2	1	2	16
3	297	10	22	1	3	1	3	28
4	540	14	36	2	5	1	4	45
5	851	20	55	2	7	1	5	67
6	1232	27	80	3	10	1	6	96
7	1683	35	111	3	13	2	8	132
8	2202	44	148	4	17	2	10	175
9	2791	55	192	4	21	2	12	225
10	3449	67	243	5	26	2	14	283
11	4176	80	301	6	32	2	16	349
12	4973	94	365	7	39	3	19	423
13	5839	110	437	8	47	3	22	506
14	6774	127	515	9	56	3	25	596
15	7779	145	600	11	67	3	28	695
16	8852	165	692	12	79	4	32	803
17	9995	185	790	13	92	4	36	918
18	11207	207	894	15	107	4	40	1041
19	12489	230	1004	16	123	5	45	1172
20	13840	255	1121	18	141	5	50	1312

Figura 4.5: Primeras filas de estimación de crecimiento de usuarios.

Para dar más realismo a los cálculos, al obtener los usuarios acumulados, hemos estimado que una parte de los mismos se pierde con cada renovación, es decir, de un mes a otro, se perderá un porcentaje de usuarios mensuales. Cada seis meses se perderán usuarios semestrales, etc. De esta forma simulamos que una parte de los usuarios renueva su suscripción y otra no.

Ingresos publicitarios

Una vez sabemos la afluencia de usuarios, comenzamos con el cálculo de ingresos a través de publicidad, que será obtenida mediante los usuarios gratuitos. Utilizamos las calculadoras de *Google AdSense* para la página web y de *Google AdMob* para las aplicaciones móviles.

Para cada uno de estos cálculos se asume que el número de visualizaciones es de 50.000 usuarios al mes. Además asumimos que la relación es directa, para poder calcular el beneficio que nos ofrece un usuario semanalmente en cada plataforma.

- **AdSense:** Se obtendrán unos 1590 € al año. Calculando el beneficio semanal de un usuario de web obtenemos 0,0006625 €.
- **AdMob IOS:** Se obtendrán unos 2992 € al año. Calculando el beneficio semanal de un usuario de IOS obtenemos 0,001246666667 €
- **AdMob Android:** Se obtendrán unos 3743 € al año. Calculando el beneficio semanal de un usuario de *Android* obtenemos 0,001559583333 €

Ahora podemos utilizar estos datos para calcular un beneficio aproximado semanal por cada usuario gratuito, hemos asumido además, que el 50 % de usuarios utilizarán *Android*, el 30 % de usuarios utilizará un dispositivo *iOS* y el 20 % restante utilizará la versión web.

Ingresos publicitarios					
Web (20%) Google Ad Sense	IOS (30%) Google AdMob	Android (50%) Google AdMob	Total publicidad semanal	Total publicidad acumulado	
0,00265	0,00748	0,01559583333	0,02572583333	0,02572583333	
0,01643	0,046376	0,09669416667	0,1595001667	0,185226	
0,0393525	0,111078	0,231598125	0,382028625	0,567254625	
0,07155	0,20196	0,4210875	0,6945975	1,261852125	
0,1127575	0,318274	0,6636027083	1,094634208	2,356486333	
0,16324	0,460768	0,9607033333	1,584711333	3,941197667	
0,2229975	0,629442	1,312389375	2,164828875	6,106026542	
0,291765	0,823548	1,71710125	2,83241425	8,938440792	
0,3698075	1,043834	2,176398542	3,590040042	12,52848083	
0,4569925	1,289926	2,689501458	4,436419958	16,96490079	
0,55332	1,561824	3,25641	5,371554	22,33645479	
0,6589225	1,859902	3,877903958	6,396728458	28,73318325	
0,7736675	2,183786	4,553203542	7,510657042	36,24384029	
0,897555	2,533476	5,28230875	8,71333975	44,95718004	
1,0307175	2,909346	6,065999375	10,00606288	54,96324292	
1,17289	3,310648	6,902715833	11,38625383	66,34949675	
1,3243375	3,73813	7,794017708	12,85648521	79,20598196	
1,4849275	4,191418	8,739125208	14,41547071	93,62145267	
1,6547925	4,670886	9,738818125	16,06449663	109,6859493	
1,8338	5,17616	10,79231667	17,80227667	127,488226	

Figura 4.6: Primeras filas del ingreso por publicidad.

Ingresos por suscripciones

Para los ingresos por suscripciones se establecieron los precios según la duración de las mismas. Como se mencionó anteriormente, el precio es de 4.99 €, 19.99 €, 35.99 € para las modalidades mensual, semestral y anual respectivamente. Utilizando los usuarios acumulados de cada tipo, podemos obtener nuestros beneficios semanales.

Ingresos Premium			
Mensual Acumulado	Bienal Acumulado	Anual Acumulado	Total Premium Acumulado
24,95 €	19,99 €	39,99 €	84,93 €
59,88 €	39,98 €	79,98 €	179,84 €
109,78 €	59,97 €	119,97 €	289,72 €
179,64 €	99,95 €	159,96 €	439,55 €
274,45 €	139,93 €	199,95 €	614,33 €
399,20 €	199,90 €	239,94 €	839,04 €
553,89 €	259,87 €	319,92 €	1.133,68 €
738,52 €	339,83 €	399,90 €	1.478,25 €
958,08 €	419,79 €	479,88 €	1.857,75 €
1.212,57 €	519,74 €	559,86 €	2.292,17 €
1.501,99 €	639,68 €	639,84 €	2.781,51 €
1.821,35 €	779,61 €	759,81 €	3.360,77 €
2.180,63 €	939,53 €	879,78 €	3.999,94 €
2.569,85 €	1.119,44 €	999,75 €	4.689,04 €
2.994,00 €	1.339,33 €	1.119,72 €	5.453,05 €
3.453,08 €	1.579,21 €	1.279,68 €	6.311,97 €
3.942,10 €	1.839,08 €	1.439,64 €	7.220,82 €
4.461,06 €	2.138,93 €	1.599,60 €	8.199,59 €
5.009,96 €	2.458,77 €	1.799,55 €	9.268,28 €
5.593,79 €	2.818,59 €	1.999,50 €	10.411,88 €

Figura 4.7: Primeras filas del ingreso por suscripciones.

Gastos de la aplicación

En el cuadro de gastos, tenemos la tarifa de desarrollador de *Apple*, fuera de esta imagen se encuentran los 25€ anuales de la cuota de desarrollador de *Google* también. Estas cuotas nos sirven para poder publicar nuestra aplicación en las tiendas de aplicaciones y otras utilidades. También disponemos de la cuota de Publicidad de *Google*, primero la versión más básica y posteriormente la versión *Premium*, que nos permitirá aparecer mejor posicionados en las búsquedas en el navegador, entre otras ventajas. También se añadieron cuotas del Plan de *Azure*¹², en este plan se incluyen despliegue de la Web, la

¹²**Azure**: Servicio de computación en la nube creado por *Microsoft* para construir, probar, desplegar y administrar aplicaciones y servicios mediante el uso de sus centros de datos.

API, la base de datos etc. Y se añadió un recargo por el número de usuarios activos, ya que según aumenten las transacciones por usuario, los accesos a la base de datos, ya sea para escritura o lectura, o se necesiten ampliar los recursos de cada servicio, aumentará su costo. Por último, se eligió una empresa que se encargará de realizar otras tareas de marketing, dirigidas a redes sociales y a imagen de producto que tienen un cargo mensual.

Para realizar los cálculos se **prorratearon**¹³ algunos gastos de forma semanal.

Gastos						
IOS Developer Anual	Publicidad google	Gastos Mantenimiento (Plan Azure) + BBDD	Marketing Digital	Gastos Semanales	Gastos semanales Acumulados	Gastos Acumulados
100	75,00 €	302,71 €	1.200,00 €	396,51 €	396,51 €	135.103,00 €
				396,51 €	793,02 €	135.499,51 €
				396,51 €	1.189,53 €	135.896,02 €
				396,51 €	1.586,04 €	136.292,53 €
	75,00 €	305,71 €	1.200,00 €	395,18 €	1.981,22 €	136.687,71 €
				395,18 €	2.376,40 €	137.082,89 €
				395,18 €	2.771,58 €	137.478,07 €
				395,18 €	3.166,75 €	137.873,24 €
	75,00 €	313,61 €	1.200,00 €	397,15 €	3.563,91 €	138.270,40 €
				397,15 €	3.961,06 €	138.667,55 €
				397,15 €	4.358,21 €	139.064,70 €
				397,15 €	4.755,36 €	139.461,85 €
	75,00 €	327,66 €	1.200,00 €	400,67 €	5.156,03 €	139.862,52 €
				400,67 €	5.556,69 €	140.263,18 €
				400,67 €	5.957,36 €	140.663,85 €
				400,67 €	6.358,02 €	141.064,51 €
	400,00 €	348,26 €	1.200,00 €	487,07 €	6.845,09 €	141.551,58 €
				487,07 €	7.332,15 €	142.038,64 €
				487,07 €	7.819,22 €	142.525,71 €
				487,07 €	8.306,28 €	143.012,77 €
	400,00 €	383,57 €	1.200,00 €	495,89 €	8.802,18 €	143.508,67 €
				495,89 €	9.298,07 €	144.004,56 €
				495,89 €	9.793,96 €	144.500,45 €
				495,89 €	10.289,85 €	144.996,34 €
	400,00 €	412,32 €	1.200,00 €	503,08 €	10.792,93 €	145.499,42 €
				503,08 €	11.296,01 €	146.002,50 €
				503,08 €	11.799,09 €	146.505,58 €
				503,08 €	12.302,17 €	147.008,66 €

Figura 4.8: Primeras filas de gastos de la aplicación.

Podemos observar que aprovechamos a calcular los gastos acumulados, en estos se contempla también la inversión inicial que realizamos cuando se desarrolló el proyecto.

Ingresos frente a gastos

Este apartado es el que nos interesa para calcular el Retorno de la Inversión. Para ello, restamos los los gastos acumulados, es decir, teniendo en cuenta la inversión inicial, a los ingresos que vamos generando.

¹³**Prorrateo:** Repartir de forma proporcional una cantidad.

Ingresos vs Gastos			
Ingreso Semanal Acumulado	Ingresos - Gastos	Índice ROI	Semana
84,96 €	-135.018,05 €	-99,94%	1
180,03 €	-135.319,49 €	-99,87%	2
290,29 €	-135.605,74 €	-99,79%	3
440,81 €	-135.851,72 €	-99,68%	4
616,69 €	-136.071,02 €	-99,55%	5
842,98 €	-136.239,91 €	-99,39%	6
1.139,79 €	-136.338,28 €	-99,17%	7
1.487,19 €	-136.386,05 €	-98,92%	8
1.870,28 €	-136.400,12 €	-98,65%	9
2.309,13 €	-136.358,41 €	-98,33%	10
2.803,85 €	-136.260,85 €	-97,98%	11
3.389,50 €	-136.072,35 €	-97,57%	12
4.036,18 €	-135.826,33 €	-97,11%	13
4.734,00 €	-135.529,19 €	-96,62%	14
5.508,01 €	-135.155,84 €	-96,08%	15
6.378,32 €	-134.686,19 €	-95,48%	16
7.300,03 €	-134.251,55 €	-94,84%	17
8.293,21 €	-133.745,43 €	-94,16%	18
9.377,97 €	-133.147,74 €	-93,42%	19
10.539,37 €	-132.473,41 €	-92,63%	20

Figura 4.9: Primeras filas del cálculo de ingresos contra gastos.

Si logramos recuperar la inversión inicial, en algún punto esta operación pasará a dar un resultado positivo, indicando así que a partir de ese punto se ha igualado nuestra inversión y pasaremos a obtener beneficios. En nuestro caso este punto se obtiene en la semana **67**.

144.361,32 €	-22.867,21 €	-13,67%	61
149.282,64 €	-18.708,44 €	-11,14%	62
154.199,48 €	-14.554,15 €	-8,62%	63
159.091,94 €	-10.424,25 €	-6,15%	64
164.020,08 €	-6.281,91 €	-3,69%	65
168.904,03 €	-2.183,76 €	-1,28%	66
173.808,85 €	1.935,25 €	1,13%	67
178.649,65 €	5.990,25 €	3,47%	68
183.431,51 €	9.961,88 €	5,74%	69
188.189,53 €	13.909,65 €	7,98%	70
192.863,81 €	17.773,69 €	10,15%	71
197.474,43 €	21.574,07 €	12,26%	72

Figura 4.10: Punto del retorno de la inversión.

Si agrupamos esta información en una gráfica, el punto del ROI se ve de la siguiente manera:

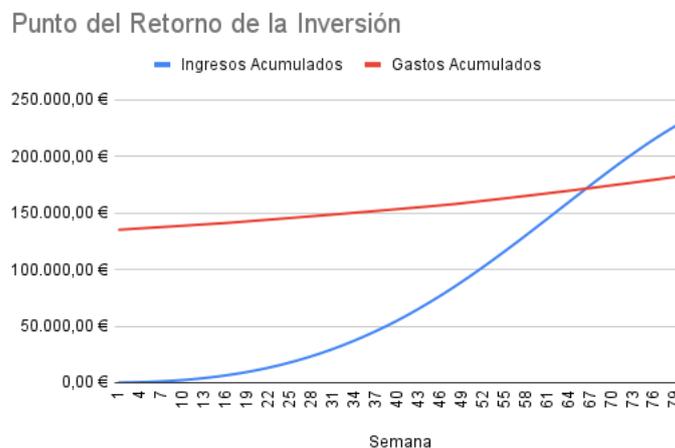


Figura 4.11: Gráfica representando los gastos y beneficios.

Capítulo 5

Conclusiones y líneas futuras

Tras finalizar el proyecto recapitularemos en los objetivos planteados al principio de esta memoria, para comprobar si han sido completados y cómo podría mejorarse en el futuro.

Se propuso la realización de un prototipo que se centre en la definición de objetivos para la lectura con el fin de mejorar nuestros hábitos. Para ello se realizó una investigación previa en la que se demostró que leer es una actividad saludable y beneficiosa para nosotros. Se estudiaron las tendencias de lectura y cuál es la relación actual de la población española con la misma.

Centrándonos en el desarrollo del prototipo, se estudió el estado del arte de las aplicaciones actuales, clasificando las mismas según sus cualidades. Posteriormente se realizó una comparación de las herramientas que emplearíamos para crear nuestro prototipo, seleccionando las que más se ajustaron a nuestras necesidades y su desempeño según las tareas que queríamos realizar.

Se creó una aplicación móvil, que nos permite iniciar sesión, buscar los libros que queramos, o incluso escanear su código ISBN con nuestra cámara si disponemos de ellos en formato físico. A cada libro podemos asignarle una fecha en la que queremos finalizarlo, y si quisiéramos, marcar cada día que leemos cuántas páginas leemos para que posteriormente nos recomiende una fecha estimada de finalización, tenemos también unas pequeñas estadísticas de lectura, etc. Además, una aplicación web en la que podemos realizar las mismas acciones. Se implementó también un *back-end*, compuesto por una base de datos, la utilización de *Firebase*, *Google Books API* y una API creada por nosotros.

Además, se realizó un estudio económico, en el que trasladamos nuestra aplicación a un proyecto real, en él definimos qué funcionalidades extra podríamos añadir, creamos las actividades a realizar, un equipo que llevaría a cabo este proyecto y definimos el tiempo y el coste del mismo.

Los objetivos planteados se han conseguido, pero es cierto que el prototipo en sí mismo podría mejorarse. Como lo vimos en el apartado 4, el Estudio de la viabilidad económica, algunas de esas ideas sólo podrían realizarse en un desarrollo profesional, pero existen otras que se podrían añadir al prototipo para enriquecerlo. Como por ejemplo añadir más objetivos, añadir una lista de amigos para poder seguir a otros lectores, añadir retos entre ellos, implementar un sistema de notificaciones...

En definitiva, a través de este proyecto se ha demostrado que leer es una actividad beneficiosa, se ha desarrollado un prototipo para promover el desarrollo de este hábito y

se ha realizado un estudio económico en el que vemos que sería posible llevar a cabo este desarrollo en la vida real y posiblemente obtener beneficios del mismo.

Capítulo 6

Summary and Conclusions

Once we finished the project, we are going to refer to the start of this memory to check if we have accomplished every objective and how it can be improved in the future.

Our main goal was to create a prototype with the idea of helping the user to create a reading habit. In order to do so we investigated the benefits of reading, this revealed that reading can be a really healthy activity. We also studied the spanish society's reading tendencies, to get to know the reason behind people's lack of interest.

Focusing on the prototype development, another study was made, this one was related to understand the state of the art of current reading applications, and classificate them by their characteristics. Afterwards we chose the tools we used to create the prototype, choosing the ones that matched our needs taking into consideration the tasks we were going to perform.

We developed a mobile application, it allows us to login, search the books we want or even scan their ISBN tag if we have them physically. We are able to assign an estimated ending date, or a number of pages we want to read. Once we have done that, we can mark each day we have read and how many pages to keep track of our progress. If we do that, the estimated date will be calculated based on our previously read books. The application has some statistics that get calculated based on our saved information. We also developed a web page that is able to perform the same tasks as the mobile application. An API of our own was created to manage our databases, Firebase, Google Books API.

To know if this project was profitable, an economic study was made. We translated this prototype to a real world project. In order to do so we decided to add extra functionalities to it, and divided the project into smaller tasks and designed a work environment. With the help of a project management tool (ProjectLibre) we calculated the time and the cost of the project.

The objectives were achieved, but the prototype itself could be improved. Some of the ideas we used for the Economic Feasibility Study, on chapter 4 could be added to the prototype. For example: increase the number of reading objectives, adding a friend's list and being able to challenge them or to be able to add push notifications to our prototype.

All in all, through this project we demonstrated that reading is a beneficial activity, we created a prototype to promote the development of this habit and an economic study has been carried out in which we see that it would be possible to go through whit this development in real life and possibly profit from it.

Capítulo 7

Presupuesto

Hemos visto el coste de llevar a cabo esta aplicación en un escenario real, ahora realizaremos un pequeño presupuesto para el prototipo que hemos creado.

Tareas	Horas dedicadas	Coste
Estudio del estado del arte	30 horas	16 €/hora
Estudio de las arquitecturas (tecnologías, hardware, software...)	50 horas	16 €/hora
Diseño de las arquitecturas	35 horas	16 €/hora
Implementación <i>Back-End</i>	60 horas	16 €/hora
Implementación aplicación móvil	125 horas	16 €/hora
Implementación aplicación web	70 horas	16 €/hora
Total	345 horas	5920 €

Tabla 7.1: Presupuesto del desarrollo del prototipo.

Para realizar este proyecto se necesitaron 345 horas de trabajo, con un precio final de 5920 €.

Bibliografía

- [1] F. de Gremios de Editores de España. «El porcentaje de lectores frecuentes alcanzado durante la pandemia se consolida.» (), dirección: <https://www.culturaydeporte.gob.es/actualidad/2022/02/220223-barometro-habitos-lectura.html>.
- [2] S. for Neuroscience. «The Brain.» (), dirección: <http://www.brainfacts.org/3D-Brain#intro=true>.
- [3] J. Daniel Yeatman, R. Dougherty, M. Ben-Shachar y B. Wandell. «Development of white matter and reading skills.» (), dirección: https://www.researchgate.net/publication/235613863_Development_of_white_matter_and_reading_skills.
- [4] M. A. Rentería, J. M. Vonk, G. Felix y col., «Illiteracy, dementia risk, and cognitive trajectories among older adults with low education,» *Neurology*, vol. 93, n.º 24, e2247-e2256, 2019.
- [5] R. Kawashima, K. Okita, R. Yamazaki y col., «Reading Aloud and Arithmetic Calculation Improve Frontal Function of People With Dementia,» *The Journals of Gerontology: Series A*, vol. 60, n.º 3, págs. 380-384, mar. de 2005, issn: 1079-5006. doi: 10.1093/gerona/60.3.380. eprint: <https://academic.oup.com/biomedgerontology/article-pdf/60/3/380/9733021/380.pdf>. dirección: <https://doi.org/10.1093/gerona/60.3.380>.
- [6] S. L. Levine, S. Cherrier, A. C. Holding y R. Koestner, «For the love of reading: Recreational reading reduces psychological distress in college students and autonomous motivation is the key,» *Journal of American College Health*, vol. 70, n.º 1, págs. 158-164, 2022, PMID: 32150516. doi: 10.1080/07448481.2020.1728280. eprint: <https://doi.org/10.1080/07448481.2020.1728280>. dirección: <https://doi.org/10.1080/07448481.2020.1728280>.
- [7] R. Vágvolgyi, A. Coldea, T. Dresler, J. Schrader y H.-C. Nuerk, «A review about functional illiteracy: Definition, cognitive, linguistic, and numerical aspects,» *Frontiers in psychology*, pág. 1617, 2016.
- [8] F. Jabr. «The Reading Brain in the Digital Age: The Science of Paper versus Screens.» (), dirección: <https://www.scientificamerican.com/article/reading-paper-screens/>.
- [9] R. Wolman. «Hindsight in 2020: The eBook Market During Covid-19.» (), dirección: <https://www.writtenwordmedia.com/ebook-market-covid/>.
- [10] M. Lebert, *A short history of ebooks*, 2009.
- [11] E. Amasawa, T. Ihara y K. Hanaki, «Role of e-reader adoption in life cycle greenhouse gas emissions of book reading activities,» *The International Journal of Life Cycle Assessment*, vol. 23, n.º 9, págs. 1874-1887, 2018.

- [12] A. Halstead. «E Ink Launches E Ink Gallery 3 Color ePaper for Sustainable Digital Reading.» (), dirección: <https://www.businesswire.com/news/home/20220425005064/en/E-Ink-Launches-E-Ink-Gallery-3-Color-ePaper-for-Sustainable-Digital-Reading>.
- [13] Descubierta. «Cuáles son los formatos más habituales para eBooks.» (), dirección: <https://descubierta.es/cuales-son-los-formatos-mas-habituales-para-ebooks>.
- [14] M. Eguaras. «El concepto de página de los ebooks: una idea a desterrar.» (), dirección: <https://marianaeguaras.com/el-concepto-de-pagina-de-los-ebooks-una-idea-a-desterrar/>.
- [15] Statista. «eBooks - Europe.» (), dirección: <https://es.statista.com/outlook/dmo/digital-media/epublishing/ebooks/europe>.
- [16] E. Nawotka. «Book Sales Slide Across Europe in First Half of 2022.» (), dirección: <https://www.publishersweekly.com/pw/by-topic/international/international-book-news/article/89830-book-sales-slide-across-europe-in-first-half-of-2022.html>.
- [17] «Python.» (), dirección: <https://www.python.org/>.
- [18] «Selenium.» (), dirección: <https://www.selenium.dev/>.
- [19] «JavaScript.» (), dirección: <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- [20] «Node.js.» (), dirección: <https://nodejs.org/es/>.
- [21] «React Native.» (), dirección: <https://reactnative.dev/>.
- [22] «React.» (), dirección: <https://es.reactjs.org/>.
- [23] «TypeScript.» (), dirección: <https://www.typescriptlang.org/>.
- [24] «Expo.» (), dirección: <https://expo.dev/>.
- [25] «Next.js by Vercel.» (), dirección: <https://nextjs.org/>.
- [26] «Native Base.» (), dirección: <https://nativebase.io/>.
- [27] R. Hat. «¿Qué es una API?» (), dirección: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>.
- [28] IBM. «¿Qué es una API REST?» (), dirección: <https://www.ibm.com/es-es/cloud/learn/rest-apis>.
- [29] «Visual Studio Code.» (), dirección: <https://code.visualstudio.com/>.
- [30] «Git.» (), dirección: <https://git-scm.com/>.
- [31] codigofacilito. «¿Qué es un web Scraper?» (), dirección: <https://codigofacilito.com/articulos/web-scraper>.
- [32] G. Developers. «Introducción a los archivos robots.txt.» (), dirección: <https://developers.google.com/search/docs/advanced/robots/intro?hl=es>.
- [33] Nest. «Database.» (), dirección: <https://docs.nestjs.com/techniques/database>.