

# Story Teller: Sistema de identificación y reproducción de cuentos para niños

Trabajo de Fin de Grado

Ingeniería Informática



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

## **Autor**

Víctor Castilla Garrido

## **Tutores**

Juan Francisco de Paz Santana

Gabriel Villarrubia González

Diego Manuel Jiménez Bravo



## Certificado de los tutores

D. Juan Francisco de Paz Santana, D. Gabriel Villarrubia González y D. Diego Manuel Jiménez Bravo, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

Hacen constar:

Que el trabajo titulado “Story Teller: Sistema de identificación y reproducción de cuentos para niños” que se presenta ha sido realizado por Víctor Castilla Garrido para la superación de la asignatura Trabajo de Fin de Grado de la titulación Grado en Ingeniería Informática de esta universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a        de Julio de 2021

D. Juan Francisco de Paz Santana

D. Gabriel Villarrubia González

D. Diego Manuel Jiménez Bravo



## Resumen

El trabajo de fin de grado “Story Teller: Sistema de identificación y reproducción de cuentos para niños” es una plataforma concebida para que los padres acerquen la literatura a sus hijos desde edades muy tempranas a través de la tecnología.

Permite a cada usuario tener una colección de libros, para los más pequeños cuentos generalmente. Una vez introducidos en la plataforma ésta permitirá la fácil gestión y utilización del material. Asimismo, el usuario podrá acceder a los contenidos desde una aplicación móvil para facilitar la utilización desde cualquier lugar.

A través de la realidad aumentada, proporcionada por la biblioteca Vuforia en la aplicación móvil, se puede usar el reconocimiento de imágenes que se hayan asociado a los libros para iniciar su reproducción de forma sencilla.

Dicha reproducción se realiza a través de la conversión de texto a voz. Este detalle permite una alta personalización del contenido que hará aún más atractivo el uso de la herramienta por los niños.

## Palabras clave

Realidad aumentada, Google, Amazon, Unity, Ruby, Android, móvil, conversor texto a voz, plataforma, cuento, niños, reconocimiento de imágenes.



## Abstract

The degree's final project "Story Teller: System for the identification and reproduction of stories for children" is a platform designed for parents to initiate their children in literature from a very early age through technology.

It allows each user to have a collection of books, or stories when it comes to the children. When the books are in the platform the user can smoothly manage and use the materials. Therefore, the user may access the content from an application and use it anywhere.

The use Vuforia library's augmented reality with the app allows the recognition of the images that have been associated with the books in order to play them readily.

Such play is done through text-to-speech conversion. This detail allows to personalize the content, which will make the use of the tool even more attractive to children.

## Keywords

Augmented reality, Google, Amazon, Unity, Ruby, Android, smartphone, text to speech converter, platform, story, kids, image recognition.





## Índice

1. Introducción.....	17
2. Objetivos.....	20
2.1 Objetivos del sistema .....	20
3. Conceptos teóricos.....	20
3.1 REST .....	21
3.2 CRUD.....	21
3.3 Base de datos noSQL .....	21
3.4 Framework .....	21
3.5 API.....	22
4. Técnicas y herramientas utilizadas .....	22
4.1 Herramientas .....	22
4.1.1 Visual studio code.....	22
4.1.2 Unity.....	23
4.1.3 Postman .....	24
4.1.4 Firebase.....	24
4.1.5 Amazon web services .....	25
4.1.6 Voice RSS .....	26
4.1.7 Ruby on Rails .....	26
4.1.8 Visual paradigm .....	26
4.1.9 EZEstimate.....	26
4.1.10 Microsoft project.....	27
4.1.11 Microsoft visual studio .....	27
4.1.12 Figma.....	28
4.1.13 Patternico.....	28
4.2 Lenguajes .....	29
4.2.1 Ruby .....	29
4.2.2 Javascript.....	29
4.2.3 HTML.....	30
4.2.4 CSS .....	30
4.2.5 C#.....	30
4.2.6 JSON .....	30
4.2.6 UML.....	30

4.3 Herramientas para el despliegue .....	31
4.3.1 Nginx .....	31
4.3.2 Android SDK .....	31
5. Aspectos relevantes del desarrollo .....	31
5.1 Marco de trabajo.....	31
5.2 Planificación temporal.....	32
5.3 Especificación de requisitos.....	34
5.3.1 Objetivos del sistema.....	35
5.3.2 Requisitos de información .....	36
5.3.3 Requisitos no funcionales .....	36
5.3.4 Requisitos funcionales .....	37
5.4 Análisis del sistema .....	38
5.4.1 Modelo del dominio .....	38
5.5 Diseño del sistema .....	39
5.5.1 Patrón MVC .....	40
5.5.2 Subsistemas de diseño.....	40
5.5.3 Realización de casos de uso - diseño.....	41
5.5.4 Despliegue.....	42
5.5 Implementación del sistema.....	42
5.5.1 Pruebas .....	44
5.6 Uso del sistema .....	44
5.6.1 Gestión de usuarios .....	44
5.6.2 Gestión de libros.....	45
5.6.3 Gestión de páginas .....	46
5.6.4 Aplicación móvil .....	47
6. Proyectos similares .....	49
6.1 Audible.....	49
6.2 Scribd .....	50
6.3 Storytel .....	50
7. Conclusiones .....	50
8. Líneas de trabajo futuras.....	52
9. Referencias y Bibliografía .....	52





## Lista de tablas

Tabla 1	Tabla de objetivo gestión de usuarios .....	35
Tabla 2	Tabla de requisitos de información de usuarios .....	36
Tabla 3	Tabla de restricción de nombres de usuario.....	36
Tabla 4	Requisitos no funcionales Usabilidad .....	37
Tabla 5	Caso de uso registrarse .....	38



## Lista de Ilustraciones

Ilustración 1 Entorno de visual studio code .....	23
Ilustración 2 Entorno de unity .....	24
Ilustración 3 Entorno de postman .....	24
Ilustración 4 Entorno de EZEstimate.....	26
Ilustración 5 Entorno de Microsoft project .....	27
Ilustración 6 Entorno de visual studio .....	28
Ilustración 7 Página de Figma.....	28
Ilustración 8 Web de patternico .....	29
Ilustración 9 Calculo de estimación temporal .....	32
Ilustración 10 Calendario de trabajo .....	33
Ilustración 11 Diagrama de Gantt de fase de inicio.....	34
Ilustración 12 Diagrama de caso de uso Gestión de páginas.....	35
Ilustración 13 Diagrama de gestión de usuarios .....	37
Ilustración 14 Modelo del dominio.....	39
Ilustración 15 Diagrama de secuencia de reconocimiento de imagen.....	39
Ilustración 16 Patrón MVC .....	40
Ilustración 17 Paquete de diseño de la aplicación .....	40
Ilustración 18 Diagrama de clases de Gestión de libros .....	41
Ilustración 19 Realización de caso de uso de página anterior .....	41
Ilustración 20 Diagrama de despliegue.....	42
Ilustración 21 Diagrama de funcionamiento Vuforia .....	43
Ilustración 22 Formulario de registro .....	45
Ilustración 23 Colección de libros.....	45
Ilustración 24 Formulario de nuevo libro.....	46
Ilustración 25 Colección de páginas de libro.....	46
Ilustración 26 Formulario de creación de página .....	47
Ilustración 27 Identificación en aplicación móvil .....	48
Ilustración 28 Reconocimiento de imágenes .....	48
Ilustración 29 Reproducción de páginas .....	49





## 1. Introducción

A lo largo de los últimos años se puede observar un cambio en los hábitos y costumbres de la sociedad moderna, con la llegada de equipos electrónicos cada vez más potentes y de menor tamaño, unido a una mayor conectividad la sociedad ha ido tendiendo a una mayor dependencia de estos dispositivos. (REDESTELECOM, REDACCIÓN, 2019)

Todo esto se une al mayor uso de internet y de dichos dispositivos entre los más jóvenes empezando cada vez antes a utilizar estas tecnologías de manera normal y en su día a día. (RTVE.es, 2017)

Por tanto, todos estos datos unidos a un retraso en la edad de maternidad generalizada en España, provocados por diversos motivos entre ellos el tiempo de trabajo provoca que cada vez niños más pequeños comiencen a utilizar estas tecnologías de forma normalizada. (INE, 2019) (Fernández, 2018)

Con todos estos datos en mente se idea Story Teller una aplicación con el objetivo de proporcionar una herramienta útil y didáctica a los padres para que los hijos puedan escuchar un cuento si ellos mismos no pueden contarlos en ese momento. Ello permitirá un contacto temprano con la literatura a los menores a través de la tecnología, independientemente de su edad y del tiempo que puedan disfrutar sus padres con ellos.

La herramienta permite el uso de los libros del hogar, pero también el acceso a una amplia colección asociada a diferentes elementos del entorno para utilizarla en cualquier momento.

La adquisición de información por vía auditiva es la inicial en todos los niños y además es bien sabido que en el proceso de adquisición de conocimientos hay personas con mayor facilidad para asimilar la forma verbal que la escrita. El texto puede ser elaborado enteramente por los padres delimitando exactamente los contenidos que desean transmitir.

Este proyecto pretende unificar los conocimientos adquiridos en el grado de ingeniería informática con los de la experiencia laboral para dar lugar a un producto con potencial.

A lo largo de esta memoria se recoge el trabajo realizado para la creación de esta plataforma, dicha memoria se subdivide en los siguientes apartados:

- **Objetivos**

Dentro de este apartado se recogen todos aquellos objetivos los cuales el proyecto debe satisfacer.

- **Conceptos teóricos**

Se recogen todas aquellas definiciones y definiciones de aquellas tecnologías y patrones utilizados a lo largo del proyecto que pueden que no queden debidamente explicadas a lo largo del documento.

- **Técnicas y herramientas utilizadas**

Durante el desarrollo del proyecto se han ido utilizando diversas herramientas las cuales han facilitado o posibilitado el avance a lo largo de las fases, por tanto, en este apartado se recogen dichas herramientas y se explican brevemente.

- **Aspectos relevantes del desarrollo**

Se recoge en este apartado las diferentes fases por las cuales ha ido evolucionando el proyecto desde su planificación hasta su despliegue final.

- **Conclusiones y líneas de trabajo futuras**

Una vez ha concluido el proyecto se recogen las conclusiones a las que se ha llegado a lo largo de todo este desarrollo y las líneas de trabajo las cuales en caso de desarrollarse mejorarían la calidad del proyecto.

- **Referencias y bibliografía**

Se recogen todas aquellas páginas, enlaces, artículos, estudios los cuales han servido de ayuda o han sido citados en el proyecto.

Además de este documento se adjuntan seis anexos los cuales incluyen lo siguiente:

- **Anexo 1 – Plan de proyecto software**

Este anexo recoge todo aquello que tenga que ver con la planificación del proyecto, desde los cálculos iniciales de estimación hasta el plan de trabajo del proyecto.

- **Anexo 2 – Especificación de requisitos software**

En este anexo se detallan cada uno de los componentes que conformaran el sistema, registrando los objetivos, los requisitos de información y sus restricciones, así como los requisitos funcionales y no funcionales, también los actores y los casos de uso necesarios para el correcto funcionamiento del sistema.

- **Anexo 3 – Análisis del sistema**

A partir de estos objetivos y requisitos, se ha realizado un análisis para poder detallar lo que será una primera versión de la arquitectura del sistema, destacar de este anexo el modelo del dominio y la arquitectura.

- **Anexo 4 – Diseño del sistema**

En este apartado se detalla una posible solución del sistema. Se refina el anexo anterior y se desarrollan los modelos del diseño, la vista arquitectónica y el modelo de despliegue.

- **Anexo 5 – Manual del programador**

En este anexo se detalla todo lo relacionado con el código de la aplicación explicando los procedimientos realizados y así poder facilitar el mantenimiento de la aplicación, así como detallar donde se encuentran cada uno de los componentes claves del sistema, su implementación y su despliegue.

- **Anexo 6 – Manual del usuario**

Se detalla la funcionalidad que tiene la aplicación de tal manera que cualquier usuario pueda entender su propósito y aprender su funcionamiento de forma rápida y sencilla.

## 2. Objetivos

Se recogen en este apartado los objetivos que debe cumplir el sistema y los objetivos personales.

### 2.1 Objetivos del sistema

En primera instancia el sistema pretende servir como herramienta de entretenimiento y educación, para ello proporcionara herramientas de gestión apropiadas para dicho cometido.

- **Gestión de usuarios:** El sistema deberá permitir el alta de nuevos usuarios en el sistema, la modificación de los datos introducidos por el usuario y controlar la identificación de estos, este apartado es crucial debido a que es en base al usuario el cómo se realizan muchas de las tareas para permitir el reconocimiento de sus imágenes y la reproducción de sus libros.
- **Gestión de libros:** El sistema deberá permitir a los usuarios crear los libros que quieran tener y modificarlos cuando ellos lo soliciten. Estos libros creados por el usuario tendrán una imagen que se utilizara para el reconocimiento.
- **Gestión de páginas:** El sistema deberá permitir la creación y edición de las páginas de los libros para que el usuario tenga pleno control sobre su contenido. Contenido el cual será reproducido por la aplicación móvil y a la cual se le deberá enviar en función de la página que el usuario quiera reproducir.
- **Reconocimiento:** El sistema deberá permitir el reconocimiento de las imágenes asociadas a cada libro que el usuario haya introducido en el sistema. Estas imágenes al ser personalizables por el usuario deberán ser cargadas cuando el usuario se identifique en la aplicación.
- **Reproducción:** El sistema deberá permitir el control sobre la reproducción de los libros, permitiendo el paso de página a gusto del usuario.
- **Gestión de favoritos:** El sistema deberá permitir la gestión de los libros favoritos de los usuarios en caso de que así lo indiquen, a su vez el sistema deberá permitir la eliminación de dichos favoritos.
- **Filtrado:** El sistema deberá permitir el filtrado de los libros de los usuarios, permitiéndoles buscar por palabras clave o una parte de estas, haciendo el manejo de la biblioteca más sencilla.
- **Acceso a favoritos:** el sistema debe proporcionar el acceso directo simple y fácil a los favoritos de los usuarios.
- **Progreso:** el sistema debe proporcionar una herramienta simple para indicar el progreso que los usuarios han realizado a la hora de la reproducción de sus libros de tal manera que puedan conocer su progreso con solo ver el libro.

## 3. Conceptos teóricos

En este apartado detallaremos los conceptos teóricos sobre los que se asienta el proyecto realizado.

### 3.1 REST

REST (transferencia de estado representacional) se trata de un estilo de arquitectura para sistemas de hipermedia distribuidos.

Esta arquitectura se basa en los siguientes puntos de diseño fundamentales:

- Cada mensaje HTTP contiene toda la información necesaria para entender la petición realizada.
- Tiene un conjunto de operaciones bien definidas como son "POST", "GET", "PUT", "DELETE".
- Utiliza una sintaxis universal para identificar los recursos.
- Usa hipermedios tanto para la información de la aplicación como para las transiciones de estado de esta, son de forma común "HTML" o "XML".

Una cosa importante en este estilo de arquitectura es la existencia de recursos, estos son elementos que permiten el acceso mediante un identificador global, su comunicación se realiza mediante una interfaz estándar como puedes ser HTTP(Hypertext Transfer Protocol), estos intercambian representaciones de recursos, aunque actualmente esto no se usa en la red común debido a que no está clara la distinción entre los recursos y sus representaciones.

### 3.2 CRUD

CRUD (Crear, leer, actualizar, borrar) es un acrónimo utilizado para referirse a las funciones básicas en las bases de datos o en la capa de persistencia de un software.

Este tipo de patrón suele ir íntimamente ligado con el mencionado en el punto anterior, refiriéndose a menudo conjuntamente.

### 3.3 Base de datos noSQL

Se trata de un tipo de base de datos que difiere completamente de aquellas SQL, siendo un dato destacado el no soportar este lenguaje, así como el no utilizar ningún tipo de estructura fija como tablas.

Este tipo de base de datos por término general se refieren a ellas como almacenamiento estructurado, las más comunes de este tipo son las que almacenan los datos en formato *clave-valor*, aunque existen otros tipos.

### 3.4 Framework

Un *framework* es un entorno de trabajo o marco de trabajo, una estructura abstracta que asiste en el desarrollo de software. Comúnmente incluye soporte de bibliotecas y programas para asistir en dicho desarrollo.

Estos tienen como objetivo principal ofrecer una funcionalidad definida, están contruidos en base a patrones de diseño confiriéndoles una alta cohesión y un bajo acoplamiento.

Sus características principales son:

- Se invierte el control, es decir en un *framework* el flujo de control no está dictado por el programa, sino por el mismo.
- Tienen un comportamiento predeterminado que por defecto debe ser útil, definido e identificable.
- Son escalables, es decir pueden ser ampliados para proporcionar funcionalidades específicas. El *framework* no se supone que deba ser modificado salvo en extensibilidad, es decir los usuarios pueden ampliar sus características, pero no es necesario modificar el código del *framework*.

### 3.5 API

Una API (interfaz de programación de aplicaciones), es un conjunto de funciones y métodos que ofrece una biblioteca para ser utilizado por otro software.

Estas API son un método para conseguir la abstracción entre diversos niveles o capas del software. Al proporcionar funciones de uso general generan grandes beneficios al programador que las utiliza ya que evita que tenga que programar todo desde el principio.

Una API puede estar construida por una organización que la venda al público general para su uso, esto permite a otra organización el uso de las funciones sin necesidad de conocer el funcionamiento interno de la misma, y permitiendo a la organización proveedora el modificarla sin problemas mientras el comportamiento siga siendo el mismo.

## 4. Técnicas y herramientas utilizadas

### 4.1 Herramientas

#### 4.1.1 Visual studio code

Se trata de un editor de código fuente desarrollado por Microsoft para los sistemas de Windows, Linux y macOS. Se caracteriza por incluir soporte de depuración e integración con GIT.

Es un editor bajo licencia MIT, por tanto, sin coste, que cuenta con grandes características de personalización para el usuario, así como muchas herramientas adicionales para el desarrollo en la mayoría de los lenguajes existentes hoy día.

Además, tiene la posibilidad de abrir una consola de comandos en el propio editor lo cual la convierte en una herramienta muy versátil para la programación.

Principalmente ha sido utilizado para el desarrollo de la aplicación web la cual ha estado basada en “Ruby”. (Microsoft, s.f.)

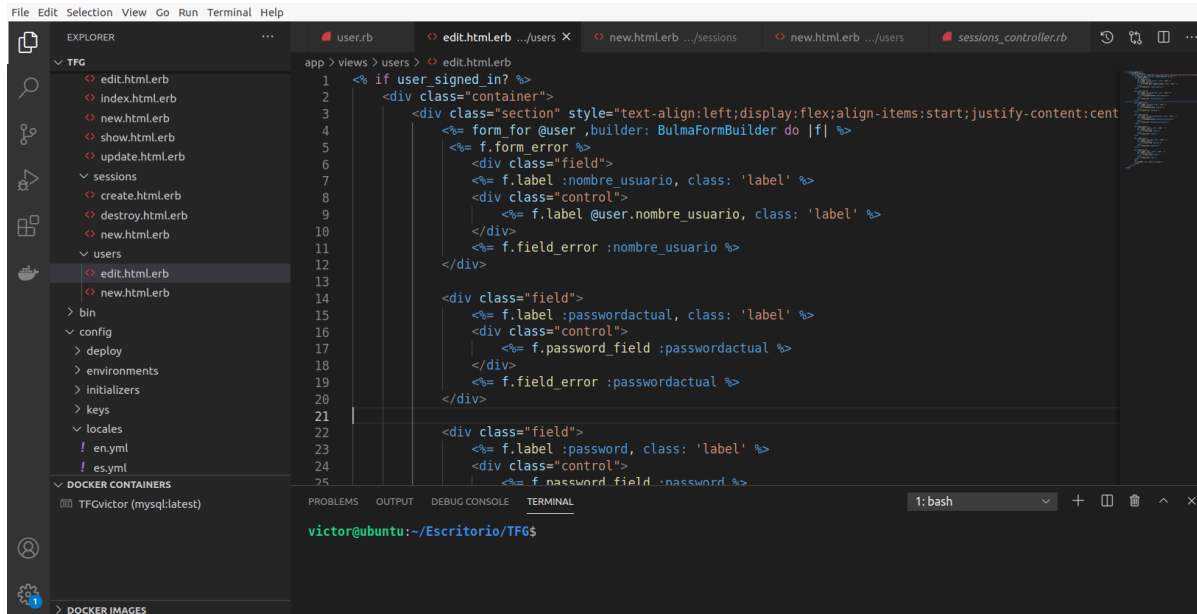


Ilustración 1 Entorno de visual studio code

#### 4.1.2 Unity

Unity se trata de un motor gráfico que permite la compilación de sus aplicaciones para múltiples plataformas como puede ser Android.

Como características destacables, es el soporte multiplataforma que ofrece la posibilidad de integración con múltiples *assets*, ya sean de pago o no y la posibilidad de utilizar múltiples librerías de desarrollo como “Vuforia”. (Unity, s.f.)

Cabe destacar que la parte de scripting se basa en Mono, el cual está diseñado principalmente en “C#”.

En este proyecto se ha utilizado Unity principalmente para el diseño de la interfaz de la aplicación móvil. También se ha incluido la biblioteca “Vuforia” para la creación y el reconocimiento de las imágenes.

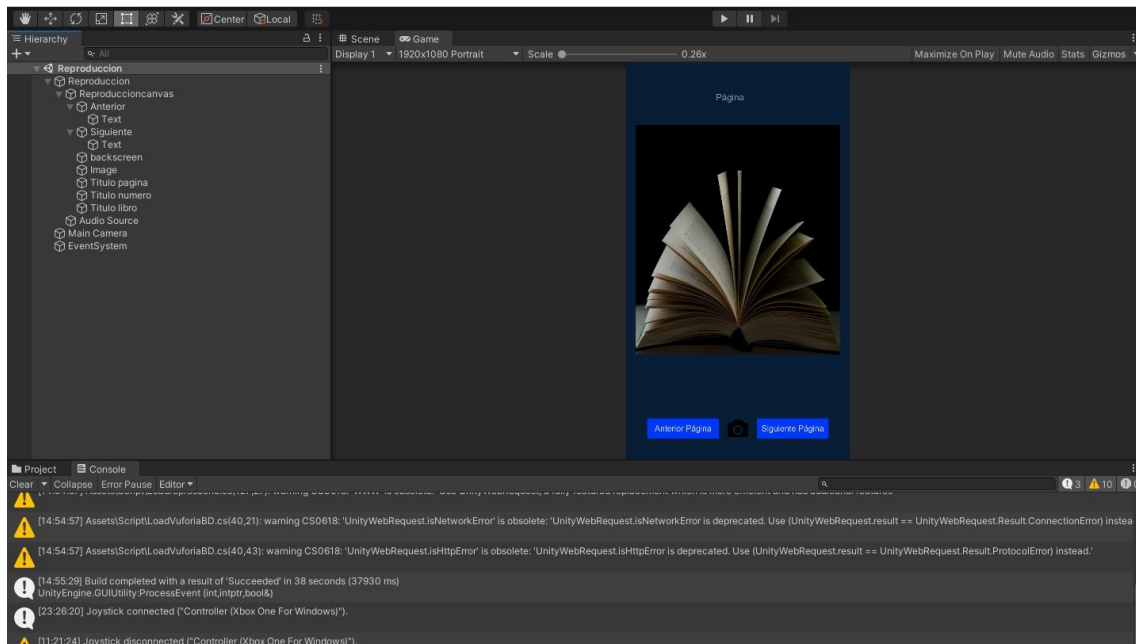


Ilustración 2 Entorno de unity

### 4.1.3 Postman

Para el desarrollo de ciertas partes de la aplicación web, en concreto la parte de conexión entre la aplicación web y móvil se ha utilizado la plataforma Postman.

Es una plataforma de colaboración para el desarrollo de APIS. Esta aplicación nos permite realizar peticiones de todo tipo con los parámetros que necesitemos, en concreto se ha utilizado para realizar peticiones “post” con parámetros concretos como si se tratara de una ejecución real para observar la respuesta generada del servidor. (Postman, s.f.)

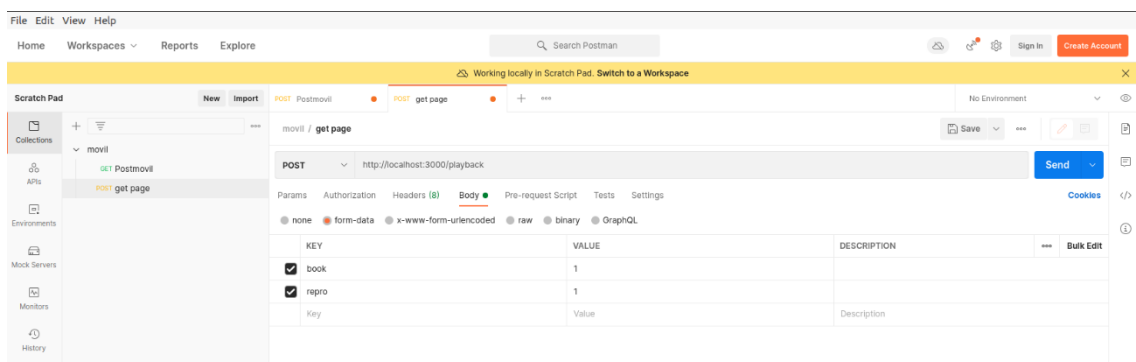


Ilustración 3 Entorno de postman

### 4.1.4 Firebase

Se trata de una plataforma propiedad de Google que se centra en el desarrollo de aplicaciones web y móvil. Está integrada en la nube con el resto de servicios de Google, de los cuales se han utilizado dos. (Google, s.f.)



#### 4.1.4.1 Firestore

Se trata de un servicio de almacenamiento en la nube basado en un sistema de carpetas y archivos, unos dentro de otros, como si de una base de datos noSQL se tratara. Ha servido para el almacenamiento de las páginas de los libros. Se eligió esta plataforma frente a una base de datos común u otras plataformas de la competencia por dos motivos:

- Para una mejor eficiencia frente al almacenamiento común de una base de datos SQL, dado que aquí a diferencia de una base de datos no hay que especificar el tamaño, por tanto, podremos tener hojas de veinte caracteres o de ciento veinte.
- En un principio esta plataforma de Google posee una mejor integración con Unity al poseer un SDK(Software Development Kit) de desarrollo para este motor. El SDK se trata de un kit de desarrollo que ayuda a los programadores a lo largo de la tarea de desarrollar el software en cuestión.

#### 4.1.4.2 Google cloud storage

Al haber seleccionado anteriormente Firebase como la plataforma de desarrollo nos incluye el acceso a este servicio de manera automatizada, dándonos más facilidades de conexión.

#### 4.1.5 Amazon web services

Se trata del servicio de Amazon en la nube que ofrece diferentes servicios tanto de almacenamiento como de hosting, así como otros. (Amazon, s.f.)

##### 4.1.5.1 EC2

EC2(Elastic Compute Cloud) es un servicio de computación en la nube. Se solicita el alquiler de un servidor remoto que se configurará de acuerdo a nuestras necesidades, facilitando así la adaptación a cada organización.

##### 4.1.5.1 RDS

RDS (Relational Database Service) es un servicio que provee de una base de datos relacional, configurable en función a las necesidades de cada organización.

#### 4.1.6 Voice RSS

Se trata de un servicio de conversión de texto a voz basado en la llamada a una función. Ésta nos devolverá el texto convertido a audio. Es una aplicación con una fase gratuita limitada por número de conversiones diarias. (Voice RSS, s.f.)

#### 4.1.7 Ruby on Rails

Ruby on Rails es un *framework* de “Ruby”. Es de código abierto y se centra en disminuir la repetición de código y la generación rápida de una aplicación funcional. (Ruby on rails, s.f.)

#### 4.1.8 Visual paradigm

Visual Paradigm es una herramienta destinada a modelar el sistema utilizando el lenguaje unificado. Permite la creación de diferentes tipos de diagramas. Para este proyecto se han creado diagramas de casos de uso, de secuencia, de despliegue entre otros. (Visual paradigm, s.f.)

#### 4.1.9 EZEstimate

Se trata de una herramienta que nos permite realizar el cálculo de la estimación de esfuerzo de manera sencilla.

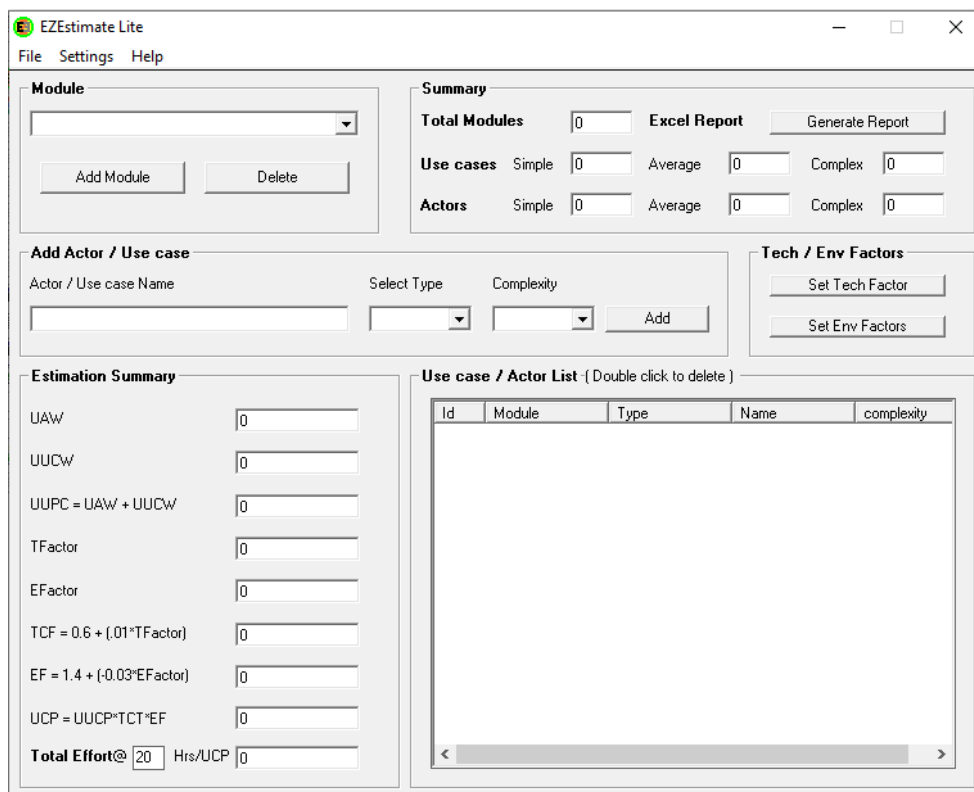


Ilustración 4 Entorno de EZEstimate

#### 4.1.10 Microsoft project

Herramienta creada por Microsoft, nos permite la creación de una planificación temporal estableciendo las tareas a lo largo del tiempo del desarrollo de la aplicación.

Por tanto, podremos establecer la fecha de inicio del proyecto y a partir de ese momento establecemos el tiempo de duración de cada una de las tareas conformando la duración total del proyecto.

Además, nos permite establecer el diagrama de Gantt que nos indicará aquellas tareas que son críticas para el proyecto y que podrían suponer un retraso en el proyecto si cualquiera de ellas fallara.

Como última característica utilizada, nos permite asignar recursos, es decir, indicar quién hará cada tarea. Nuestro objetivo será mantener los recursos siempre al máximo de su capacidad. (Microsoft, s.f.)

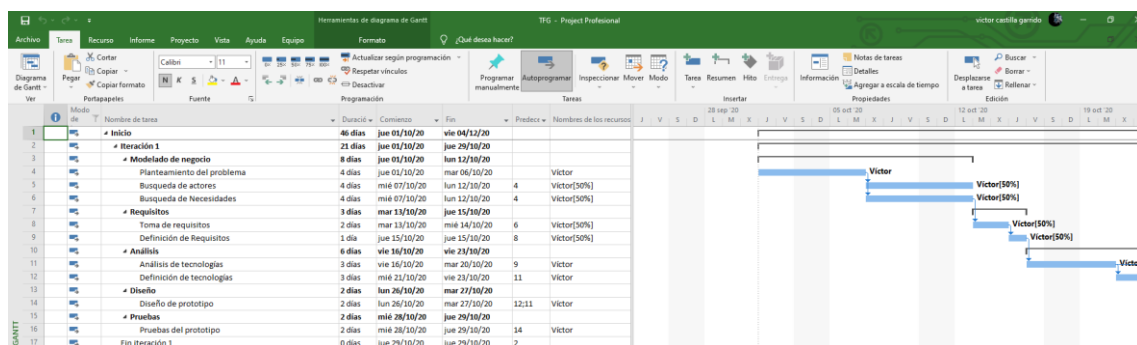


Ilustración 5 Entorno de Microsoft project

#### 4.1.11 Microsoft visual studio

Se trata de una herramienta creada por Microsoft para el desarrollo de software, es un IDE (Entorno de Desarrollo Integrado) completo que incluye depurador. Tiene soporte para diversos lenguajes, así como compatibilidad con distintos frameworks de desarrollo web tales como “Django”.

Además, en sus últimas versiones se le ha ido añadiendo compatibilidad con diversos elementos de la nube de “Azure” convirtiéndolo en un entorno de desarrollo bastante eficaz en función las necesidades de cada usuario.

El uso principal que ha tenido este IDE es el de desarrollar el código ligado a la aplicación móvil y por tanto unido al desarrollo con unity. (Microsoft, s.f.)

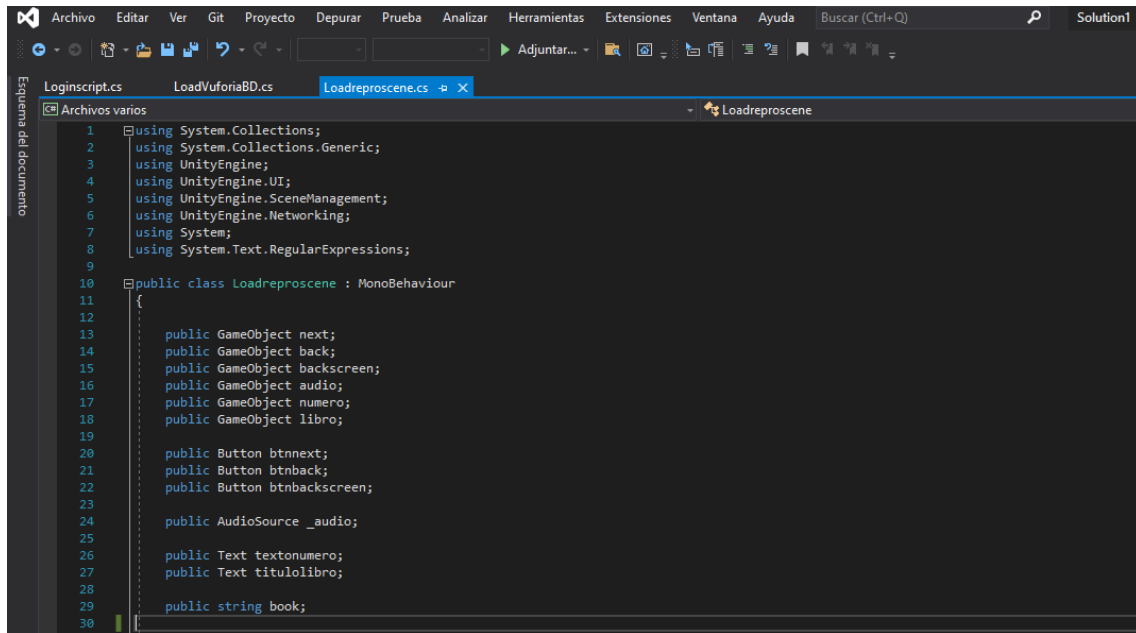


Ilustración 6 Entorno de visual studio

#### 4.1.12 Figma

Figma es una herramienta que permite la edición de gráficos vectoriales y la generación de prototipos de forma rápida y sencilla. Ha sido utilizada para la creación de un prototipo de web, sirviendo de guía para la realización de los elementos visuales de la web. (Figma, s.f.)

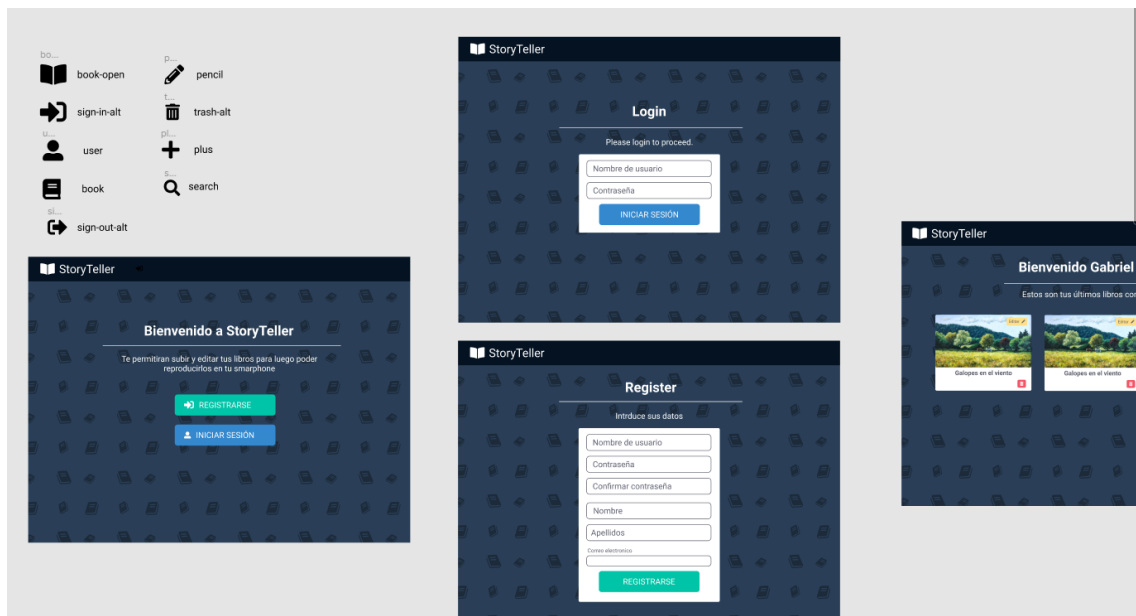


Ilustración 7 Página de Figma

#### 4.1.13 Patternico

Patternico es una web que nos permite la realización de fondos con diferentes patrones en función a nuestras preferencias. En este caso se ha utilizado para la elaboración del fondo web. (Patternico, s.f.)

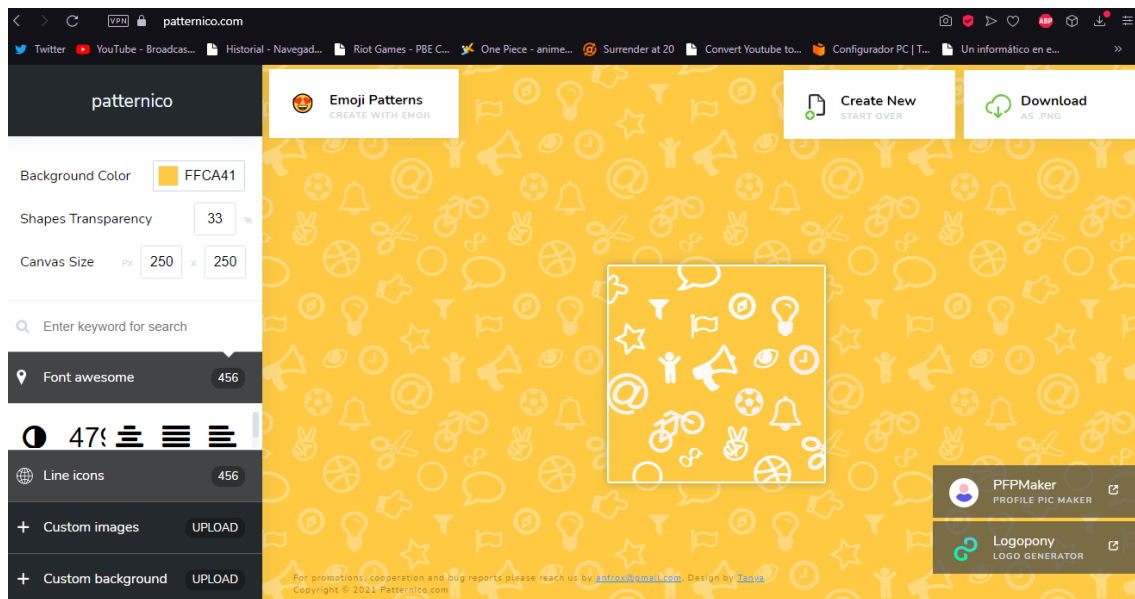


Ilustración 8 Web de patternico

## 4.2 Lenguajes

### 4.2.1 Ruby

Se trata de un lenguaje de programación de código abierto interpretado inspirado en “Python” y “Perl”. Todo esto mezclado con características de programación orientada a objetos.

“Ruby” es un lenguaje popular que cuenta con una gran comunidad que favorece que el lenguaje avance y se simplifiquen ciertos aspectos de las aplicaciones, ya que una de las mayores características de “Ruby” son sus gemas, creadas por los usuarios.

Una gema es semejante a una biblioteca, proporciona funcionalidades específicas simplificando los aspectos a programar, en contrapartida puede resultar que algún elemento no pueda ajustarse a tus necesidades específicas.

Se decidió el uso de “Ruby” en vez de otros lenguajes por su sencillez, su facilidad de programación y su gran comunidad y documentación. (Ruby, s.f.)

### 4.2.2 Javascript

Se trata de un lenguaje de programación interpretado, imperativo y orientado a objetos. Su principal uso es en el desarrollo web, aunque también se utiliza fuera de este tipo de desarrollo, por ejemplo, en Adobe Acrobat.

En el proyecto se ha utilizado para la creación de la visualización de las páginas de los libros, permitiendo el cambio rápido entre ellas sin necesidad de nuevos accesos a páginas.

La decisión de utilizar este lenguaje no fue propia dado que este lenguaje es intrínseco al *framework* de “Ruby” para la realización de ciertas características. (Mozilla, s.f.)

#### 4.2.3 HTML

“HTML” o lenguaje de etiquetas de hipertexto es el componente básico de las webs, define el significado y estructura del contenido web.

Su uso ha estado destinado a estructurar la vista de cada una de las páginas, como característica de “Ruby” este lenguaje se mezcla con el propio lenguaje de “Ruby” en los archivos que definen las vistas. (Mozilla, s.f.)

#### 4.2.4 CSS

“CSS” es un lenguaje de estilos, como tal su nombre es hojas de estilo en cascada. Describe cómo se debe mostrar cada elemento que está estructurado dentro los archivos “HTML”. (Mozilla, s.f.)

Su uso ha sido el de cambiar los estilos a cada uno de los elementos de la interfaz.

#### 4.2.5 C#

“C#” o “C Sharp” es un lenguaje de programación desarrollado por Microsoft, esta deriva en su sintaxis básica de “C” y “C++”, con la característica de que es orientado a objetos. (Microsoft, s.f.)

En el proyecto ha sido utilizado junto con Unity para proveer de la funcionalidad y conexiones necesarias a la aplicación.

#### 4.2.6 JSON

“Json” es un formato de texto utilizado para el intercambio de datos. Su uso es para el intercambio de datos web de manera sencilla e independiente del lenguaje de programación utilizado. (Json, s.f.)

En la aplicación su uso es el de intercambiar datos de los libros y páginas entre la web y la aplicación móvil.

#### 4.2.6 UML

Se trata del lenguaje unificado de modelado. Nos proporciona las herramientas y directrices para especificar, construir y documentar un sistema. Así podemos modelar el sistema y es independiente del lenguaje que se utilice, aunque está ideado sobre los principios de la programación orientada a objetos. (OMG@, s.f.)

En nuestra aplicación será el lenguaje base para construir todos los artefactos y todas las representaciones del sistema.

## 4.3 Herramientas para el despliegue

### 4.3.1 Nginx

Se trata de un servidor web/proxy inverso. Es un software libre y multiplataforma cuya función principal es de servidor web, aunque también se usa de balanceador de carga y proxy de diversos tipos. (Nginx, s.f.)

El propósito del uso de Nginx para nuestra aplicación es que funcione como intermediario entre las peticiones entrantes y nuestro servidor interno de la aplicación.

### 4.3.2 Android SDK

Se trata del software de desarrollo de Android que permite la creación de aplicaciones para dicha plataforma. (Google, s.f.)

Su uso en la aplicación es la de compilador de la aplicación móvil desarrollada en Unity y compilada para ser usada en Android.

## 5. Aspectos relevantes del desarrollo

### 5.1 Marco de trabajo

Para la realización de este trabajo de fin de grado se ha utilizado el proceso unificado, que tiene las siguientes características principales:

- **Dirigido por casos de uso:** estos casos son descripciones de las acciones que realiza un actor dentro del sistema. Con esto constituiremos los requisitos funcionales. Dichos actores pueden ser desde usuarios humanos hasta interacciones que el sistema realice o aplicaciones externas al sistema que interactúen con él.
- **Centrado en la arquitectura:** Se trata de la definición de la estructura del proyecto que permite la realización de todos y cada uno de los casos de uso. Hay que destacar que dicha arquitectura no es estática, sino que va evolucionando con cada interacción para adaptarse y así poder cumplir su función
- **Iterativo e incremental:** El desarrollo de un software es complejo, se precisa de subdividirlo en tareas que se enmarcan en un proceso iterativo. En cada una de las iteraciones se desarrolla una parte de las tareas planificadas conduciéndonos a la consecución de hitos que marcan un incremento y por tanto una nueva versión funcional de la aplicación.

Este proceso tiene diversas fases: inicio, elaboración, construcción, transición. Cada una con distintas iteraciones, dentro de las cuales tienen sus tareas correspondientes.

En el caso de este proyecto se ha utilizado las fases nombradas anteriormente de la siguiente manera:

- **Inicio:** Durante la fase de inicio se realizó el planteamiento del problema, es decir todo aquello relacionado con la propuesta lo cual la aplicación debía de solventar y de cubrir, así como la elección de las tecnologías que se utilizarían en base a lo que sabíamos en aquel momento, en este caso concreto por ejemplo se optó por Ruby y Unity, y al realizar esta elección se escogió Google como proveedor de almacenamiento debido a que ambos soportan y tienen conectividad con dicho proveedor
- **Elaboración:** Esta fue la fase más larga del proyecto, en ella se desarrolló lo que sería la base para la futura solución al problema en forma de aplicaciones web y móvil, según fueron avanzando las iteraciones esta solución fue tomando forma y se desarrollaron los diferentes módulos de gestión y conectividad, así como las pruebas sobre estos.
- **Construcción:** Durante esta fase el trabajo principal fue el de integrar todos y cada uno de los módulos los cuales se habían desarrollado y comprobar que funcionaban correctamente como un conjunto y no como módulos solitarios.
- **Transición:** Durante esta fase el cometido fue desplegar todos y cada uno de los componentes en sus correspondientes proveedores de hosting y asegurar la conectividad entre ellos, conformando así el sistema final.

### 5.2 Planificación temporal

Como se ha dicho antes al estar dirigido por casos de uso, también nos proporciona un método de cálculo de estimaciones. Estas estimaciones se han calculado mediante EZEstimate.

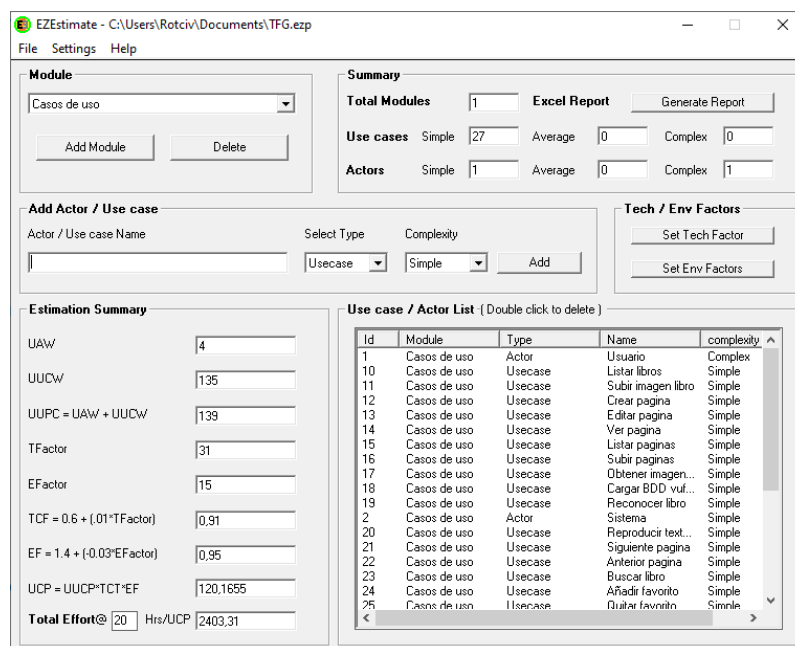


Ilustración 9 Calculo de estimación temporal



Como podemos observar nos encontramos con una estimación de 2403 horas, en base a esto hemos realizado una planificación en el tiempo a lo largo de 8 meses.

Para ello hemos realizado el siguiente calendario laboral el cual subdivide además el tiempo entre las diferentes fases del proceso unificado.



**Leyenda**  
 Festivos: Verde  
 Fase Inicio: Azul  
 Fase Elaboración: Violeta  
 Fase Construcción: Azul oscuro  
 Fase Transición: Rojo

*Ilustración 10 Calendario de trabajo*

En base a esto se ha realizado una subdivisión de las tareas a lo largo de este tiempo, dejándonos un diagrama de Gantt que nos mostrará las tareas críticas:

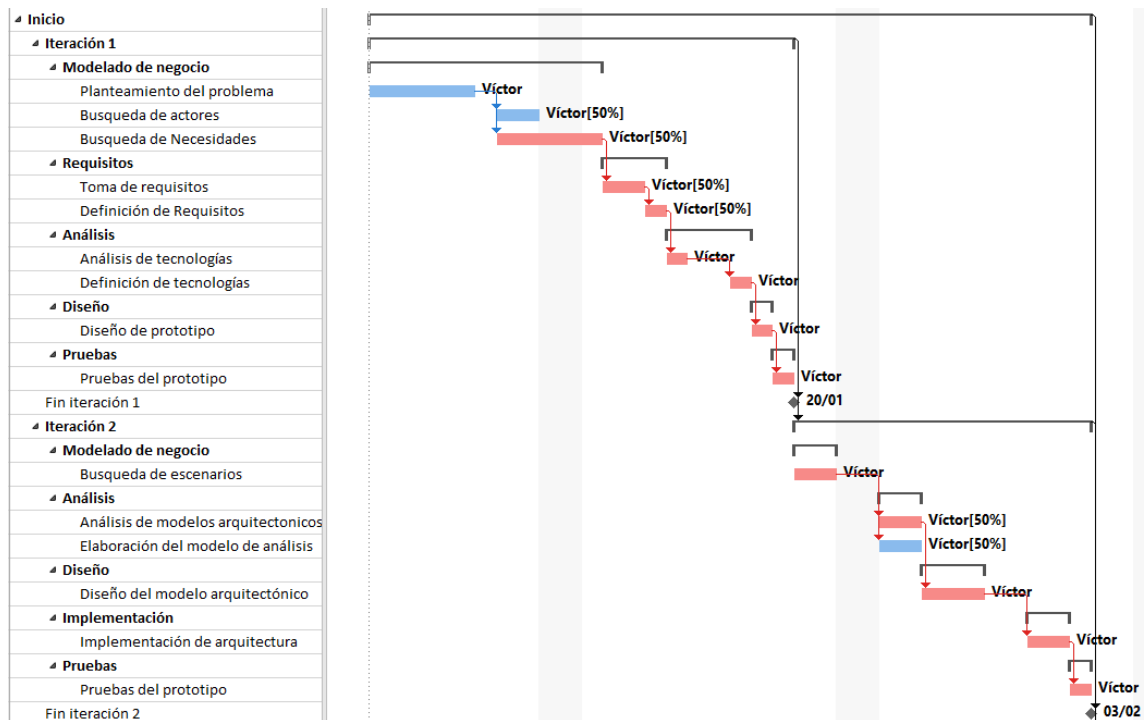


Ilustración 11 Diagrama de Gantt de fase de inicio

Todo se encuentra más detallado en el *Anexo I- Plan de proyecto software*.

### 5.3 Especificación de requisitos

Al ser un proyecto sin cliente no puede realizarse una especificación de requisitos como se haría comúnmente.

Por tanto, lo que se ha hecho para la toma de estos requisitos es realizar un análisis de la propuesta presentada del trabajo de fin de grado, la cual recogía que se debía realizar una plataforma donde se pudieran subir los cuentos y un dispositivo móvil el cual reconozca las imágenes y reproduzca su contenido.

En base a esta descripción se han propuesto los siguientes requisitos basándonos en los puntos críticos anteriormente nombrados y añadiendo algunos indispensables para la realización de estos:

- Creación de una aplicación web que permita el registro de usuarios, creación de libros y páginas.
- Edición de usuarios, libros y páginas mediante la aplicación web.
- Creación de una aplicación móvil que permita la identificación de usuarios.
- Creación de una aplicación móvil que permita el reconocimiento de las imágenes previamente subidas a la aplicación web.
- Creación de una aplicación móvil que reproduzca las páginas del libro reconocido.

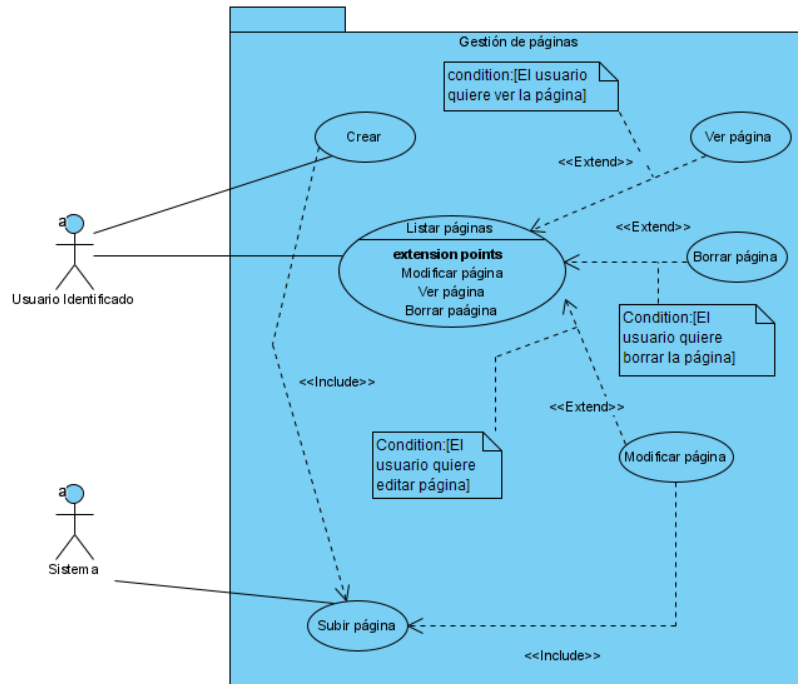


Ilustración 12 Diagrama de caso de uso Gestión de páginas

Estas especificaciones están más detalladas en *Anexo II- Especificación de requisitos software*.

### 5.3.1 Objetivos del sistema

Lo objetivos establecidos en el sistema son los siguientes:

- Gestión de usuarios.
- Gestión de libros.
- Gestión de páginas.
- Aplicación móvil.

Un ejemplo de las tablas que muestran estas especificaciones es el siguiente:

OBJ-0001	Gestión de usuarios
versión	1.0
Autores	Víctor Castilla Garrido
Fuentes	
Descripción	El sistema debe ser capaz de registrar usuarios, permitir su conexión mediante validación y modificar su perfil.
Importancia	Vital
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 1 Tabla de objetivo gestión de usuarios

### 5.3.2 Requisitos de información

El sistema tiene una especificación en función a los requisitos nombrados anteriormente. Con ellos se sabe qué información necesita el sistema, así como las restricciones impuestas sobre el mismo.

Como ejemplo de la realización de estos requisitos de información y restricción se mostrarán las dos siguientes tablas:

IRQ-0001	Información de usuarios
versión	1.0
Autores	Víctor Castilla Garrido
Fuentes	
Dependencias	OBJ-0001
Descripción	El sistema debe ser capaz de almacenar la información de los usuarios registrados en el sistema.
Datos específicos	<ul style="list-style-type: none"> <li>• Nombre de usuario</li> <li>• Nombre</li> <li>• Apellidos</li> <li>• Email</li> <li>• Contraseña</li> </ul>
Importancia	Vital
Estabilidad	Alta
Comentarios	

*Tabla 2 Tabla de requisitos de información de usuarios*

CRQ-0001	Nombres de usuario
versión	1.0
Autores	Víctor Castilla Garrido
Fuentes	
Dependencias	IRQ-0001
Descripción	No pueden existir dos usuarios con el mismo nombre de usuario.
Importancia	Vital
Estabilidad	Alta
Comentarios	

*Tabla 3 Tabla de restricción de nombres de usuario*

### 5.3.3 Requisitos no funcionales

La aplicación contiene requisitos no funcionales que no afectan a la funcionalidad del sistema:

NFR-0001	Usabilidad
versión	1.0
Autores	Víctor Castilla Garrido
Fuentes	
Dependencias	
Descripción	El sistema debe ser lo más sencillo posible para cualquier usuario para que pueda usarlo de forma rápida y simple con un mínimo de aprendizaje. Todo esto teniendo en cuenta que los usuarios no tienen por qué tener conocimientos informáticos.
Importancia	Vital
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 4 Requisitos no funcionales Usabilidad

### 5.3.4 Requisitos funcionales

Estos requisitos afectan directamente a la funcionalidad del sistema. Se representan mediante diagramas y tablas de especificación. Se mostrarán un ejemplo de ambos.

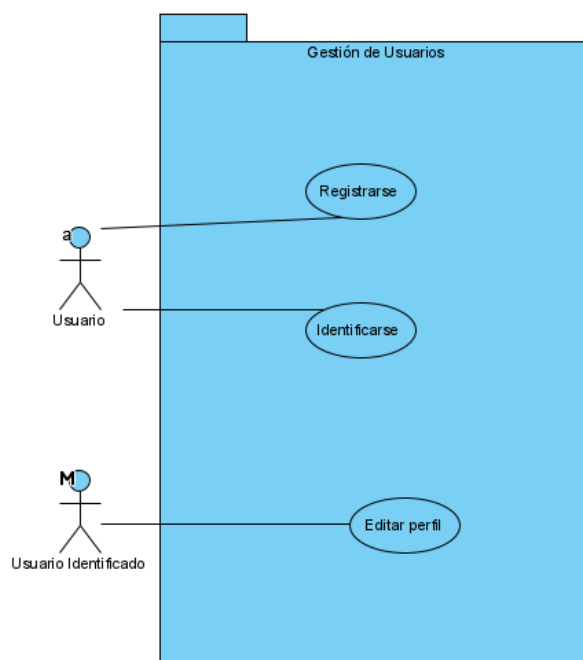


Ilustración 13 Diagrama de gestión de usuarios

UC-0001	Registrarse
Versión	1.0
Autor	Víctor Castilla Garrido
Fuentes	
Dependencias	
Descripción	El sistema debe comportarse según se describe en este caso de uso al registrar un usuario.
Precondición	El usuario no está registrado.
Secuencia normal	<ol style="list-style-type: none"> <li>1. El usuario da los datos asociados a la cuenta</li> <li>2. El sistema procesa los datos del usuario</li> <li>3. Se almacenan los datos en la base de datos</li> </ol>
Postcondiciones	El usuario queda registrado en la aplicación.
Excepciones	2. Si los datos son inválidos o el nombre usuario está ocupado se notifica al usuario y este caso de uso queda sin efecto.
Rendimiento	
Frecuencia esperada	
Importancia	Vital
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 5 Caso de uso registrarse

## 5.4 Análisis del sistema

En este apartado se incluirán aquellos elementos relacionados con el análisis del sistema, este se encuentra más detallado en el *Anexo III- análisis del sistema*.

Para realizar este análisis se han realizado los diferentes paquetes de análisis los cuales nos muestran las estructuras de los diferentes subsistemas, un diagrama de arquitectura para poder visualizar la estructura del sistema y la realización de casos de uso para entender las interacciones de los diferentes componentes.

### 5.4.1 Modelo del dominio

Se va a mostrar el diagrama del modelo del dominio incluido en el anexo anterior. Este diagrama es una representación de los conceptos del mundo real que se verán involucrados en la aplicación para su manejo.

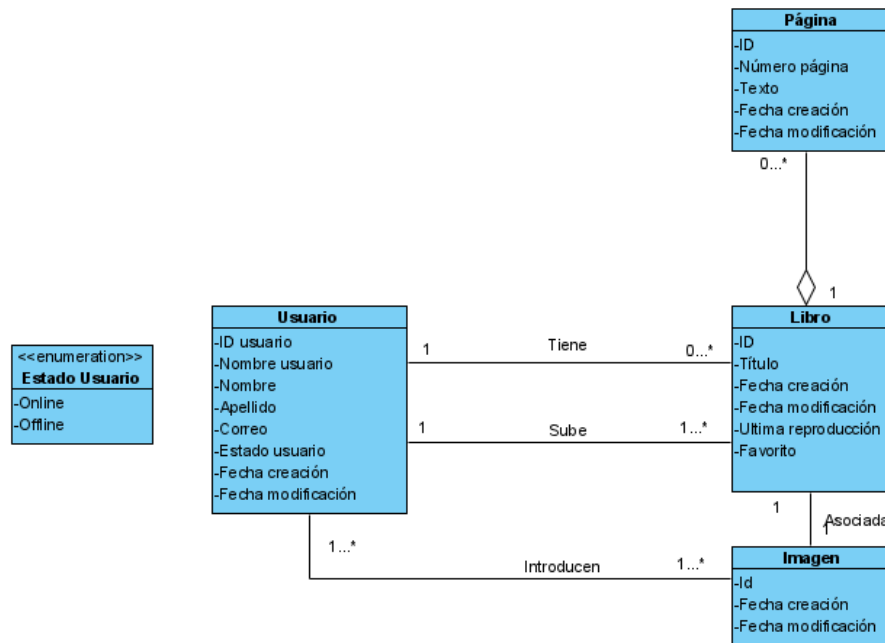


Ilustración 14 Modelo del dominio

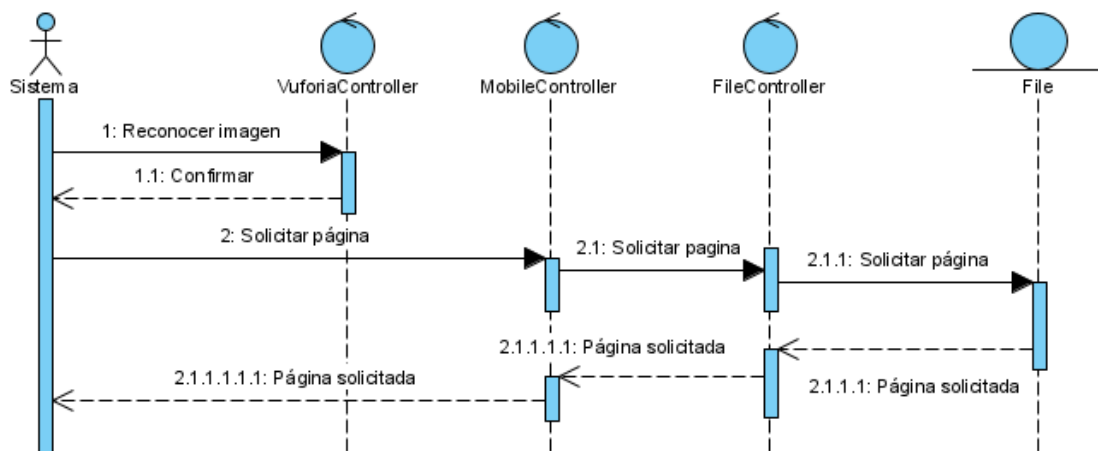


Ilustración 15 Diagrama de secuencia de reconocimiento de imagen

### 5.5 Diseño del sistema

En este apartado se mostrará el diseño del sistema, es decir, aquellos aspectos relevantes que intervienen en el diseño del sistema y por tanto en el dominio de la solución. Este apartado está más extendido en el *Anexo IV- Diseño del sistema*.

### 5.5.1 Patrón MVC

El patrón utilizado en el sistema es el patrón MVC (Modelo, Vista, Controlador) que corresponde al siguiente diagrama.

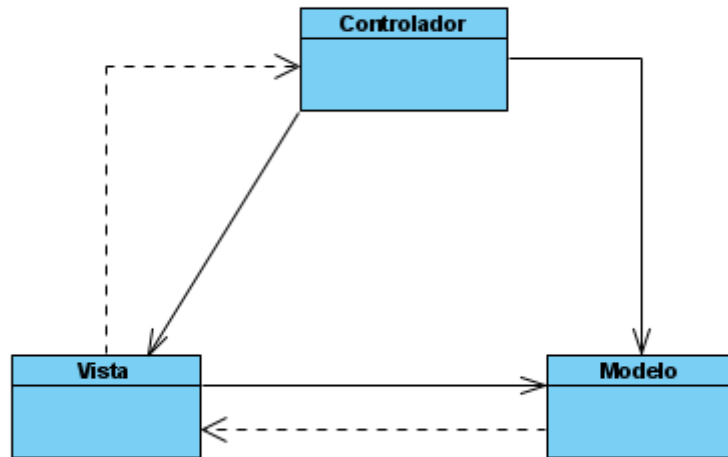


Ilustración 16 Patrón MVC

### 5.5.2 Subsistemas de diseño

Se mostrará a continuación un ejemplo de un diagrama de un subsistema de diseño que muestran las relaciones entre las diferentes partes del sistema.

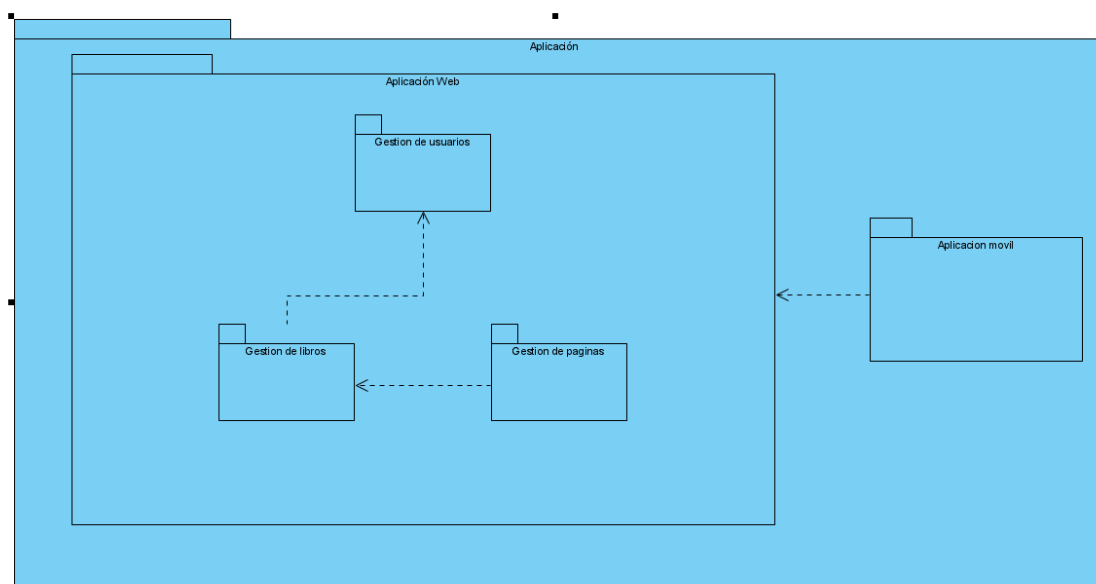


Ilustración 17 Paquete de diseño de la aplicación



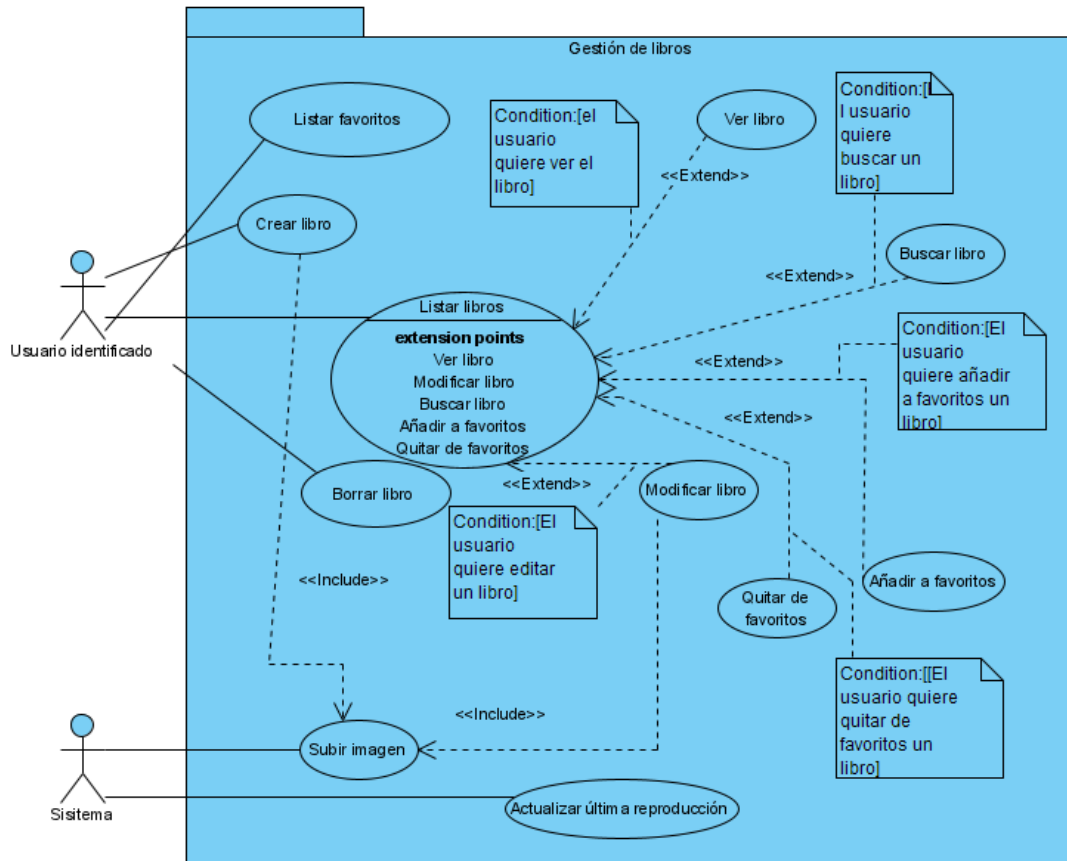


Ilustración 18 Diagrama de clases de Gestión de libros

### 5.5.3 Realización de casos de uso - diseño

Se mostrará un ejemplo de un caso de uso que se aproxima más al dominio de la solución que el diagrama visto anteriormente.

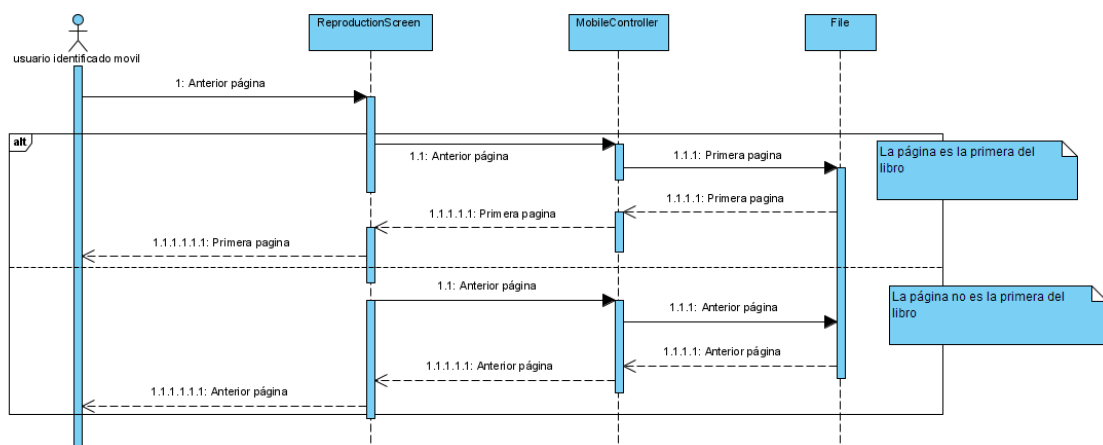


Ilustración 19 Realización de caso de uso de página anterior

### 5.5.4 Despliegue

Se mostrará el diagrama de despliegue del sistema, el cual muestra como estarán subdivididos los componentes una vez el sistema esté en funcionamiento.

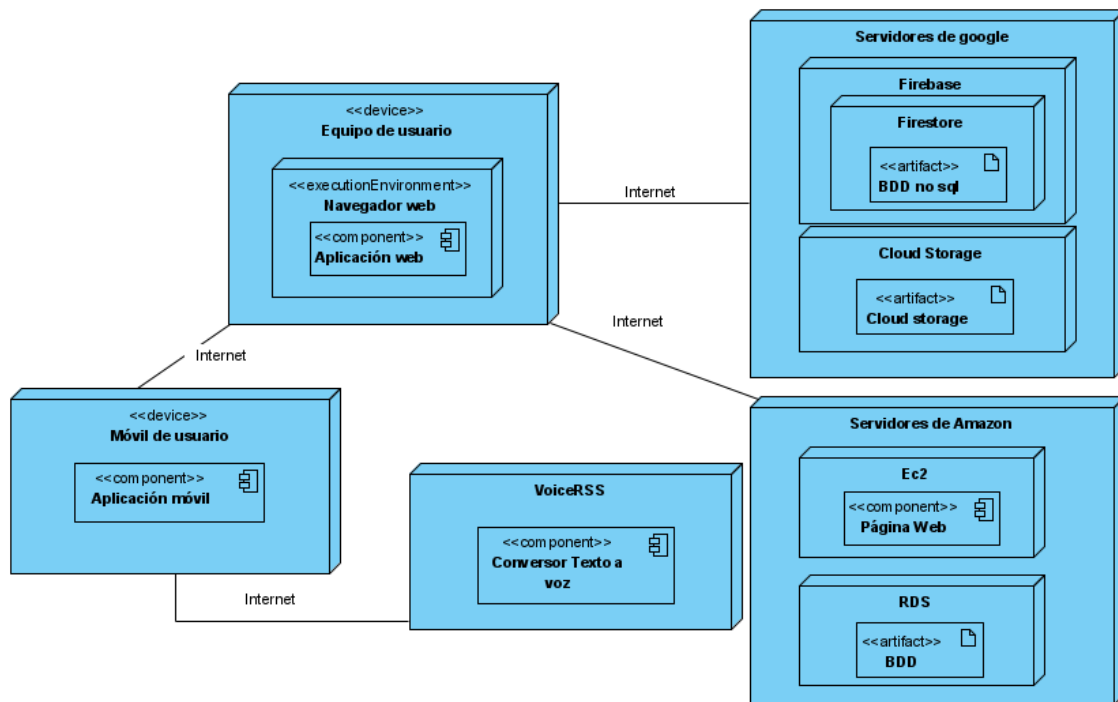


Ilustración 20 Diagrama de despliegue

### 5.5 Implementación del sistema

Se va a describir brevemente aquellos aspectos relevantes de la construcción e implementación del sistema aquí expuesto, esto corresponde en mayor detalle al *Anexo V- Manual del programador*, donde se detalla cada una de las partes del código y como desplugarlo.

Uno de los requisitos que plantea al sistema es el reconocimiento de imágenes de los usuarios, esto si bien puede ser relativamente asequible gracias a la biblioteca de "Vuforia", esta funciona de la siguiente manera:

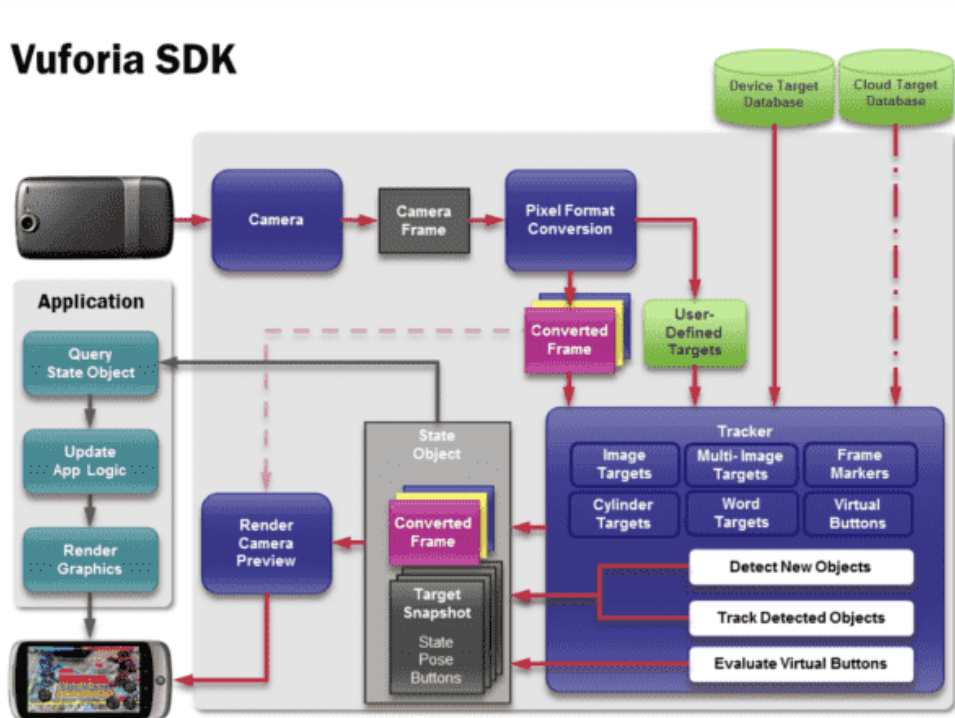


Ilustración 21 Diagrama de funcionamiento Vuforia

El reconocimiento supuso un gran reto ya que para el buen uso de la aplicación las imágenes que se debían convertir en objetos reconocibles debían ser solo las del usuario en cuestión para evitar duplicidad.

Aunque no solventa el problema de la duplicidad lo aligera bastante. Para ello se tuvo que crear una identificación en el móvil para reconocer al usuario que iba a conectarse y localizar sus fotos. Todo ello intentando que el tiempo de espera para el usuario sea el menor posible.

Además, en el sistema se ha requerido una implementación de adquisición de imágenes en tiempo de ejecución, haciendo que una vez el usuario se conecte las imágenes que le pertenezcan pasen a ser objetos reconocibles en el momento en el que sepamos cuáles son.

Para realizar este procedimiento se realiza una búsqueda entre los libros para identificar aquellos los cuales pertenecen a dicho usuario, una vez los tenemos debemos buscar su imagen asociada en la base de datos, esta información la enviaremos a la aplicación móvil.

Una vez llega esta información se genera un bucle en el cual recuperamos de nuestro almacenamiento en la nube esas imágenes convirtiéndolas en objetos reconocibles.

Otro problema encontrado es que una vez tengamos dichas imágenes y podamos reconocerlas la biblioteca en cuestión se usa principalmente para la realidad aumentada. Es decir, su uso principal es la de mostrar objetos en la pantalla a la hora de reconocer una imagen, como por ejemplo, una esfera en tres dimensiones.

Por tanto, para solventar este inconveniente se tuvo que modificar internamente la biblioteca para que a la hora de realizar la detección realizara un salto en vez de mostrar algo en pantalla.

Cabe destacar que la realización de la página web se desarrolló de forma simple sin ningún gran contratiempo importante salvo en el despliegue debido a la necesidad de instalaciones y configuraciones adicionales.

Estas configuraciones de despliegue son específicas para cada proyecto creado en Ruby en el aspecto de que requiere de una clave secreta generada en esa máquina para el correcto funcionamiento en el entorno de producción.

Además, para el entorno de producción se requiere de la precompilación de *assets* los cuales permiten que la página suministre de forma más rápida las vistas a los usuarios que las solicitan.

### 5.5.1 Pruebas

A la vez que se iban desarrollando los diferentes componentes del sistema se iban realizando test sobre ellos. Primero fueron test unitarios de prueba de cada uno de los componentes, para luego pasar a test de integración.

Todas estas pruebas se realizaron primero pensando en cómo debía de realizar el funcionamiento normal el sistema, es decir, el correcto funcionamiento del sistema y una interacción normal, para luego forzarlo con interacciones anormales, por ejemplo, introduciendo datos no válidos, introduciendo datos de otro tipo en campos que requerían uno concreto, con peticiones de formularios incorrectas, etc.

Por último, se entregó la aplicación móvil y se concedió acceso a la aplicación web a usuarios no involucrados en el desarrollo para que experimentaran con la aplicación y ver qué resultados proporcionaban.

## 5.6 Uso del sistema

La aplicación tiene como uso principal el permitir reproducir aquellos libros los cuales el usuario haya introducido en el sistema.

Se va a detallar en este apartado parte del funcionamiento del sistema. Se encuentra más detallado en el *Anexo VI- Manual de usuario*.

### 5.6.1 Gestión de usuarios

El sistema permite al usuario no registrado, realizar el registro y acceder a todos los aspectos de la funcionalidad que ofrece el sistema. Para ello deberá rellenar el siguiente formulario:

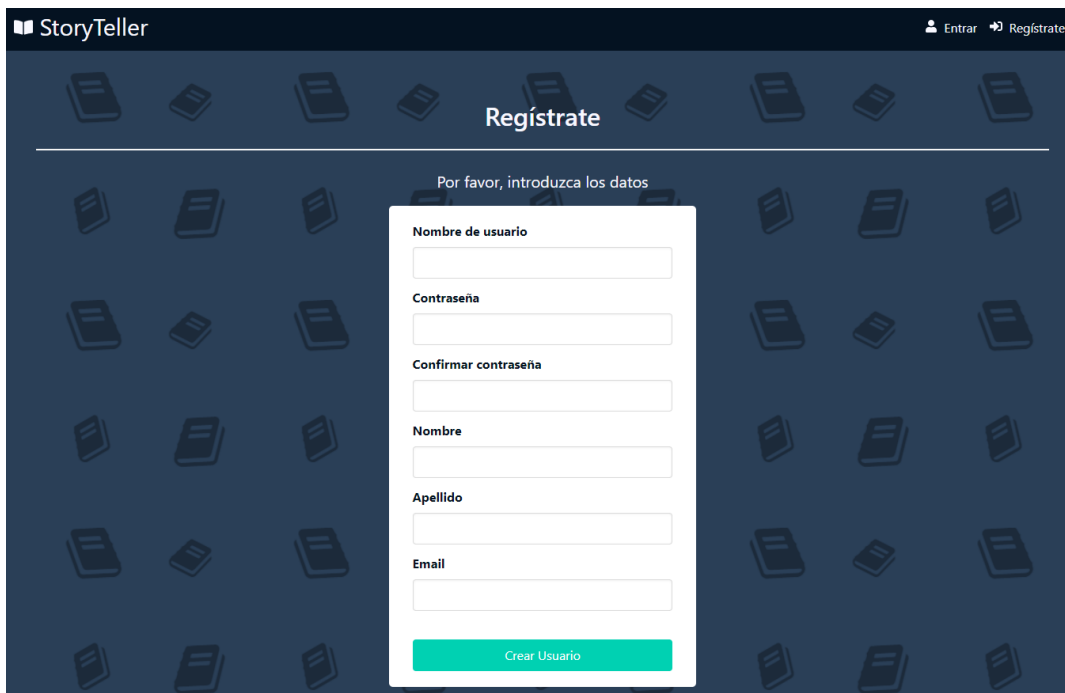


Ilustración 22 Formulario de registro

Este mismo formulario se tiene que rellenar en caso de querer actualizar alguno de los datos aquí mostrados, además de confirmar la identidad con la contraseña actual.

### 5.6.2 Gestión de libros

El sistema permite tanto la creación como la edición de los libros que el usuario desee, estos libros se muestran en la colección de cada usuario.

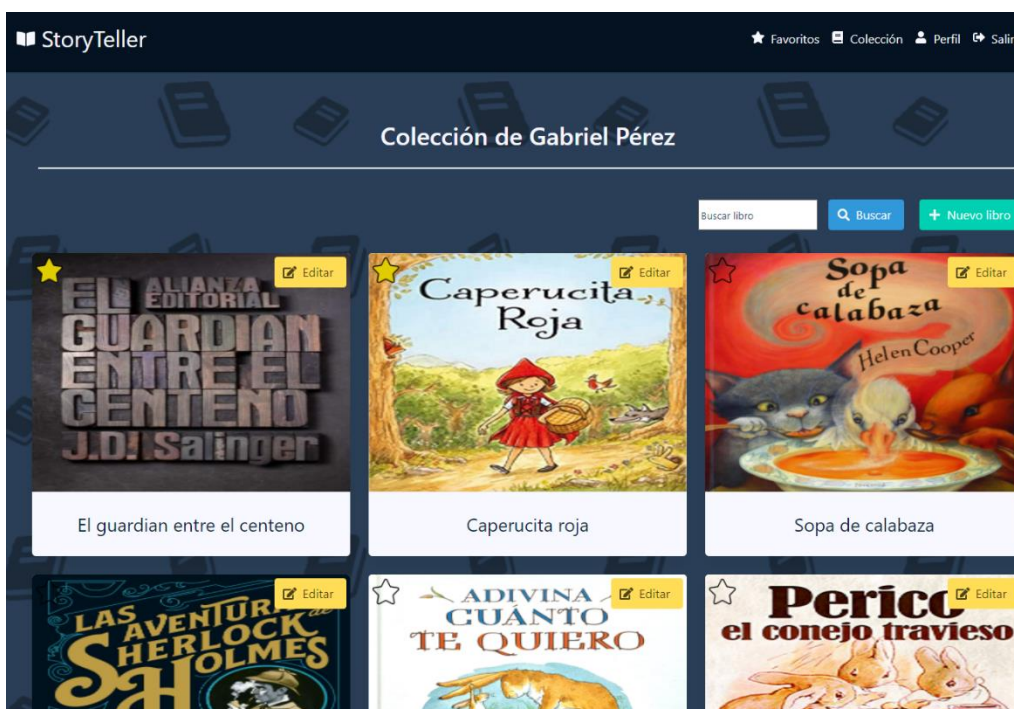


Ilustración 23 Colección de libros

Esta colección muestra nueve libros por página, después de esa cifra debemos cambiar de página para ver los siguientes nueve. Para la creación de un libro solo es necesario rellenar el siguiente formulario:

The screenshot shows the 'Nuevo libro' form in the StoryTeller application. The form is centered on a dark blue background with a repeating book icon pattern. It contains a title input field, a photo selection button, and a 'Crear Libro' (Create Book) button.

Ilustración 24 Formulario de nuevo libro

Para la edición de los libros únicamente se deberá rellenar un formulario similar cambiando los datos deseados.

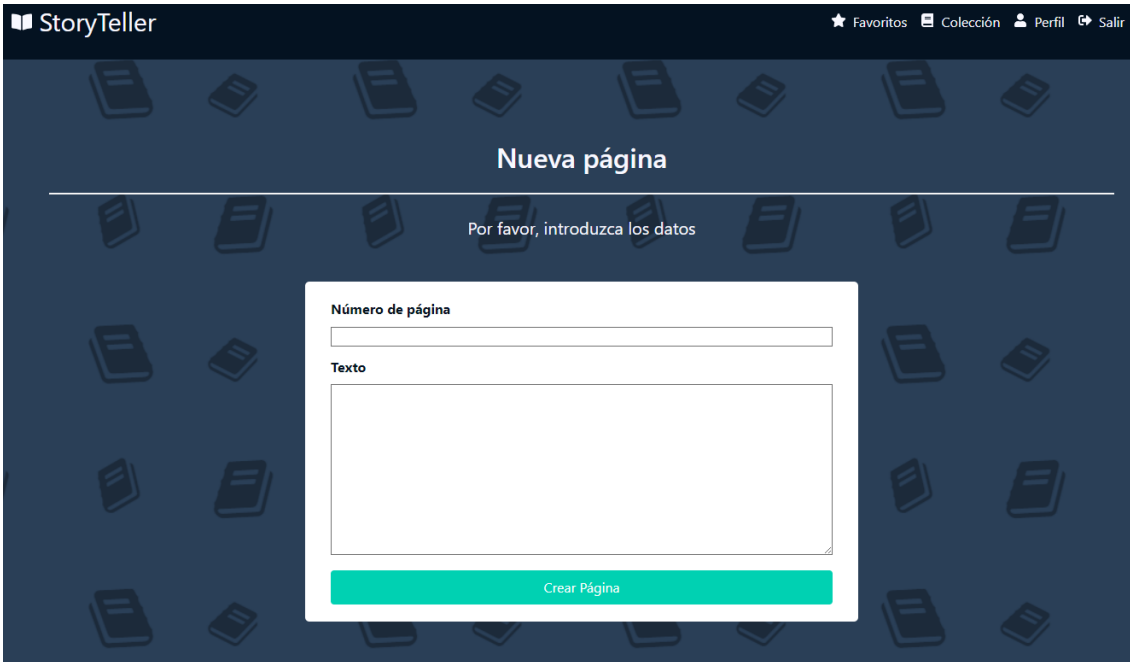
### 5.6.3 Gestión de páginas

Para la gestión de las páginas de cada libro existe una colección interna en cada uno donde se listan todas las páginas que este posee. Se trata de una colección similar a la de los libros con nueve páginas por hoja de paginación.

The screenshot shows the 'Páginas de El guardian entre el centeno' collection in the StoryTeller application. The page features a progress bar indicating the user is on page 1 of 2. There are buttons for 'Nueva página' (New page), 'Borrar libro' (Delete book), 'Borrar' (Delete), and 'Editar' (Edit). The main content area displays the text of the first page.

Ilustración 25 Colección de páginas de libro

Para crear una nueva página se debe rellenar el siguiente formulario:



The screenshot shows the 'StoryTeller' mobile application interface. At the top, there is a dark blue header with the 'StoryTeller' logo on the left and navigation icons for 'Favoritos', 'Colección', 'Perfil', and 'Salir' on the right. The main content area has a dark blue background with a repeating pattern of smartphone icons. In the center, there is a white form titled 'Nueva página'. Below the title, there is a prompt: 'Por favor, introduzca los datos'. The form contains two input fields: a text input field labeled 'Número de página' and a larger text area labeled 'Texto'. At the bottom of the form is a green button labeled 'Crear Página'.

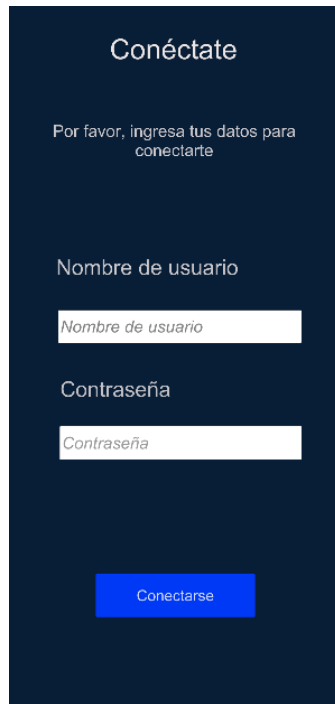
*Ilustración 26 Formulario de creación de página*

Las páginas no deben ser muy extensas porque imposibilitan la conversión de texto a VOZ.

Para la edición de una de estas páginas hay que rellenar un formulario como el anterior cambiando los datos que consideremos necesarios.

#### 5.6.4 Aplicación móvil

Se explica el uso de la aplicación móvil. Una vez abierta lo primero que debemos hacer es realizar la identificación:



*Ilustración 27 Identificación en aplicación móvil*

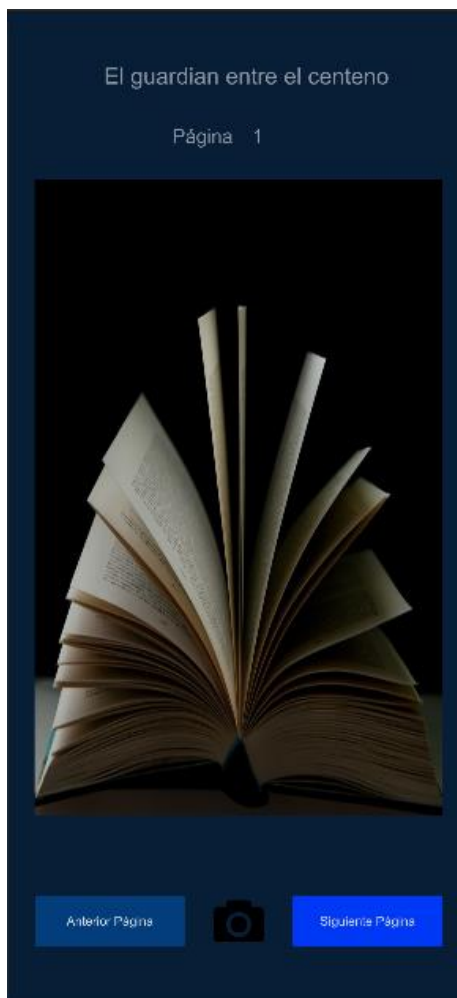
La identificación puede tardar unos instantes, después nos encontraremos en la pantalla con el logotipo de “Vuforia”, indicando que hemos accedido a la pantalla de reconocimiento.



*Ilustración 28 Reconocimiento de imágenes*



Debemos apuntar dicha imagen hacia un elemento que sabemos que es la portada de uno de nuestros libros. Así la pantalla volverá a cambiar mostrándonos la que será la pantalla de reproducción:



*Ilustración 29 Reproducción de páginas*

En esta pantalla se nos mostrara el título, la página donde nos encontramos y los botones de cambio de página del libro. Los botones estarán activos o no en función de si tenemos páginas anteriores o siguientes o no.

## 6. Proyectos similares

### 6.1 Audible

Audible es un proyecto de Amazon que se centra en ofrecer al usuario una selección de audiolibros narrados por personas famosas. Ofrece la posibilidad de escuchar los libros que se ofertan a cambio de una suscripción.

Se trata de un servicio que, si bien te ofrece una gran colección de libros que podría ser una alternativa al proyecto que aquí se presenta, no ofrece ninguna capacidad de personalización ni de control.

## 6.2 Scribd

Scribd es plataforma con una oferta amplia de libros, audiolibros, revistas, podcast y algunos servicios más. Es de suscripción mensual como la anterior, pero ofrece algunos servicios extra, aunque tampoco permite personalización de los usuarios, pero sí un mayor catálogo.

## 6.3 Storytel

Se trata de una plataforma como las dos anteriores. Ofrece a cambio de una suscripción mensual un catálogo de audiolibros para añadir a tu biblioteca y también planes familiares y educativos.

Aunque se asemeja más al proyecto presentado, una vez más no permite la personalización de libros, aunque sí un mayor catálogo.

## 7. Conclusiones

De la realización de este proyecto se extraen diversas conclusiones tanto a nivel personal, profesional y técnico.

A nivel técnico el proyecto me ha permitido alcanzar los siguientes puntos:

- Aprender un nuevo lenguaje de programación.
- Poner en práctica lo aprendido en la carrera.
- Ver la utilidad práctica de todos los diagramas y estimaciones.
- Seguir mejorando mis habilidades de investigación y de documentación.
- Refinar aquellos conocimientos sobre redes y peticiones aprendidos durante mis prácticas de empresa.

A nivel de proyecto se han cumplido con aquellos objetivos que se establecieron al inicio del proyecto, estos comprendían:

- Desarrollo de una aplicación web de gestión para los usuarios de la plataforma donde se les permitiera la gestión de sus libros y páginas.
- Desarrollo de una interfaz que permitiera el fácil uso de la aplicación web.
- Desarrollo de una aplicación móvil que permitiera el reconocimiento de una imagen introducida por un usuario.
- Desarrollo de una aplicación móvil que permitiera la reproducción de los textos introducidos por los usuarios,

- Desarrollo de una aplicación general que permitiera a cada usuario tener su propia biblioteca de libros y su reproducción con independencia del dispositivo Android utilizado.

A nivel tecnológico el proyecto ha planteado diferentes retos:

- Vuforia permite el reconocimiento de las imágenes las cuales nosotros cargamos aun si son parciales, no obstante, se ve bastante reducida esta efectividad en el reconocimiento.
- Ruby permite una gran sencillez a la hora de la programación y generación de aplicaciones, sin embargo, está sencillez provoca que en ocasiones no se adapte todo perfectamente a nuestras necesidades.
- Las gemas que Ruby pone a disposición de los usuarios creadas por la comunidad son una gran fuente de ayuda para todos los programadores, aunque no todas se encuentran bien documentadas ni se adaptan a las necesidades específicas de cada uno.
- Si bien es una gran ventaja de cara al usuario el incremento de la velocidad al precompilar *assets* visuales sucede que a la hora de cambios estéticos radicales se requiera de una nueva compilación para poner en fruncimiento estos nuevos elementos.

Desde un punto de vista personal este proyecto me ha exigido que ponga dedicación diaria para poder sacarlo adelante, un ritmo de trabajo constante, compromiso y disciplina.

Pasando a un punto de vista profesional me ha ayudado mucho a comprender las magnitudes de los proyectos que se realizan ya que, pese a ser un proyecto modesto implica muchas horas de trabajo, de planificación y de esfuerzo.

Por tanto, creo que me va a ayudar mucho de cara a futuros proyectos debido a que, aunque no es común realizar proyectos en solitario, me ha permitido comprender lo que implica y la importancia de las estimaciones tanto de esfuerzo como de costes.

Al ser un proyecto realizado desde el inicio hasta una fase de utilidad (aunque aún pueda desarrollarse más) me ha permitido comprender el compromiso que requiere cualquier proyecto y entender por qué se requiere que cada proyecto sea lo más perfecto que pueda llegar a ser tanto a nivel técnico, visual y de facilidad de uso para que alcance al mayor número de personas y tenga el mayor impacto posible.

## 8. Líneas de trabajo futuras

De cara al futuro de la aplicación se han propuesto las siguientes líneas de investigación a corto y medio plazo:

- Mejorar la gestión de libros y páginas para permitir un completo control de forma aún más sencilla.
- Mejorar la introducción de datos de las páginas de tal manera que se pueda introducir un archivo de texto para cada página.
- Mejorar la conversión de texto a voz haciéndola más natural y que se permita seleccionar una voz masculina o femenina.
- Añadir una onda de sonido para mejorar la estética de la aplicación móvil.
- Añadir compatibilidad con mayores dispositivos.

De cara a las proyecciones a largo plazo la aplicación podría tomar las siguientes líneas de desarrollo:

- Introducir la posibilidad de generar listas de reproducción
- Ampliar la oferta a la reproducción de audios como por ejemplo canciones para darle mas versatilidad a la aplicación.
- Ampliar las plataformas en las que se ejecuta la aplicación móvil extendiendo la aplicación a iOS.
- Mejorar la interfaz gráfica continuamente para hacerla más agradable al usuario.
- Introducir la capacidad de compartición de elementos de contenido.
- Introducir filtros de lenguaje para evitar que mediante la compartición puedan llegar a usarse palabras mal sonantes y otro tipo de vocabulario el cual no es apto para niños.
- Introducir un filtro de imágenes para impedir que se suban o en su defecto se mantengan imágenes las cuales no son adecuadas para menores de edad y o que contengan lenguaje no apropiado.

## 9. Referencias y Bibliografía

Amazon. (s.f.). *AWS documentation*. Obtenido de <https://docs.aws.amazon.com>

Fernández, P. (4 de Mayo de 2018). *guiainfantil.com*. Obtenido de <https://www.guiainfantil.com/blog/educacion/nuevas-tecnologias/el-tiempo-de-uso-de-la-tablet-o-movil-en-ninos-segun-su-edad-resumido-en-una-tabla/#:~:text=Según%20la%20Academia%20Americana%20de,contacto%20directo%20con%20las%20personas>.

Google. (s.f.). *Android developer*. Obtenido de <https://developer.android.com/studio>

Google. (s.f.). *Firebase*. Obtenido de <https://firebase.google.com/docs?hl=es>

- INE. (9 de Abril de 2019). *ine*. Obtenido de [ine: https://www.ine.es/prensa/ef\\_2018\\_d.pdf](https://www.ine.es/prensa/ef_2018_d.pdf)
- Json. (s.f.). *Json.org*. Obtenido de <https://www.json.org/json-es.html>
- Microsoft. (s.f.). *Microsoft docs*. Obtenido de <https://docs.microsoft.com/es-es/dotnet/csharp/>
- Microsoft. (s.f.). *Microsoft project*. Obtenido de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- Microsoft. (s.f.). *Visual studio*. Obtenido de <https://visualstudio.microsoft.com/es/vs/>
- Microsoft. (s.f.). *Visual studio code*. Obtenido de <https://code.visualstudio.com>
- Mozilla. (s.f.). *MDN web docs*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mozilla. (s.f.). *MDN web docs*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- Mozilla. (s.f.). *MDN web docs*. Obtenido de <https://developer.mozilla.org/es/docs/Web/CSS>
- Nginx. (s.f.). *Nginx*. Obtenido de <https://www.nginx.com>
- OMG@. (s.f.). *UML*. Obtenido de <https://www.uml.org>
- Postman. (s.f.). *Posman*. Obtenido de <https://www.postman.com>
- REDESTELECOM, REDACCIÓN. (29 de Julio de 2019). *REDESTELECOM*. Obtenido de REDESTELECOM: <https://www.redestelecom.es/comunicaciones/noticias/1113522000303/espanoles-cada-vez-mas-adictos-al-movil.1.html>
- RTVE.es. (5 de Octubre de 2017). *RTVE*. Obtenido de RTVE: <https://www.rtve.es/noticias/20171005/mas-del-88-ninos-10-anos-usa-ordenadores-accede-internet/1626076.shtml#:~:text=Las%20niñas%20aventajan%20a%20los%20niños%20en%20el%20uso%20del%20móvil&text=Un%2093%2C9%20%25%20de%20las,%2C9%20%25%20de%20los%20hombres>.
- Ruby on rails. (s.f.). *Ruby on rails api*. Obtenido de <https://api.rubyonrails.org>
- Ruby. (s.f.). *ruby-lang*. Obtenido de <https://www.ruby-lang.org/es/>
- Unity. (s.f.). *Unity*. Obtenido de <https://unity.com>
- Visual paradigm. (s.f.). *Visual paradigm*. Obtenido de <https://www.visual-paradigm.com>
- Voice RSS. (s.f.). *Voice RSS api*. Obtenido de <http://www.voicerss.org/api/>

Ingeniería del Software I. Francisco José García Peñalvo.

Ingeniería del Software II. María N. Moreno García.

Gestión de Proyectos. María N. Moreno García.