



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA

Grado en Ingeniería Informática
de Servicios y Aplicaciones

Notes4All

Sistema distribuido de compartición de apuntes

*Alumno: Yaser El Dabete Arribas
Tutor: Fernando Díaz Gómez*

Agradecimientos

En primer lugar, agradecer a mi familia por todo el apoyo y el cariño recibido durante estos años, sin ellos hubiese sido más complicado.

A mis amigos por estar siempre ahí sacándome una sonrisa y haciéndome pensar en otra cosa en los momentos malos.

Por último y no menos importante agradecimiento especial a mi tutor, Fernando Díaz Gómez por prestarse a guiarme en estos meses de desarrollo del proyecto.

Resumen

El principal objetivo de este Trabajo Final de Grado es el de dar facilidades a los usuarios para compartir conocimientos de una forma intuitiva, rápida y segura. De esta forma, nace *Notes4all*, un sistema distribuido de compartición de apuntes que cuenta con un validador autónomo de documentos publicados, aportando así una fiabilidad a los usuarios que deseen leer dicha documentación.

Palabras clave: Android, Angular, apuntes, validación, documentos.

Abstract

The main purpose of this Final Degree Project is to provide users with facilities to share knowledge in an intuitive, fast and secure way. In this way, *Notes4all* is born, a distributed system of sharing of notes that has an autonomous validator of published documents, thus providing reliability to users who wish to read said documentation.

Keywords: Android, Angular, notes, validation, documents.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción	9
1.1 Introducción	9
1.2 Motivación	9
1.3 Objetivos	10
1.4 Alcance	10
1.6 Entorno tecnológico	13
1.6.1 Plataformas o <i>frameworks</i>	13
1.6.1 Herramientas de soporte.....	17
1.7 Organización del documento	18
1.8 Contenido del CD-ROM	19
Capítulo 2 – Estado del arte	20
Capítulo 3 – Planificación	22
3.1 Metodología de trabajo	22
3.2 Planificación temporal	23
3.2 Análisis	25
3.2.1 Coste de los componentes hardware	25
3.2.2 Coste de los componentes software.....	25
3.2.3 Coste del personal.....	26
3.2.4 Método de puntos de función	26
3.2.5 Estimación por COCOMO	29
3.2.6 Comparativa de estimaciones.....	32
3.2.7 Presupuesto final.....	32
Capítulo 4 – Análisis	34
4.1 Descripción de los actores	34
4.2 Requisitos de usuario	35
4.3 Casos de uso	37
4.4 Especificación de los requisitos de usuario	39
4.4 Requisitos de información	53

4.4.1 Diagrama entidad-relación.....	53
4.4.1 Diccionario de datos.....	54
4.5 Requisitos no funcionales	55
Capítulo 5 – Diseño e implementación	56
5.1 Arquitectura lógica.....	56
5.2 Arquitectura física	58
5.3 Diagrama de clases.....	59
5.4 Modelo lógico de la base de datos.....	61
5.5 Implementación	62
5.6 Diseño de la interfaz de usuario	63
5.6.1 Aplicación móvil	63
5.6.2 Aplicación web.....	67
Capítulo 6 – Pruebas	72
6.1 Pruebas de caja blanca	72
6.2 Pruebas de caja negra.....	72
Capítulo 7 – Documentación del producto	78
7.1 Manual de despliegue	78
7.2 Manual de usuario.....	80
7.2.1 Aplicación web.....	80
7.2.2 Aplicación móvil	83
7.2 Manual de administradores	87
Capítulo 8 – Conclusiones y mejoras.....	91
Referencias	93
Índice de tablas.....	96

Capítulo 1 - Introducción

1.1 Introducción

Los últimos datos dicen que el 59% de la población mundial somos consumidores de Internet, esto se traduce en que 4.540 millones de personas somos usuarios de Internet y empleamos una media de 6 horas y 43 minutos diarios en este cometido.

En consecuencia, cada día cobran más fuerza los servicios online; revisamos nuestro correo diario, compramos, vemos la televisión, hacemos la compra e incluso estudiamos online.

Todo esto se traduce en que estos sistemas son cada vez más robustos, más sofisticados, más seguros y más sencillos de usar para el usuario. Pero sobre todo, lo que nos permite dichos servicios es la posibilidad y facilidad de compartir lo que queramos con la persona que vive debajo de nosotros o con la que vive a cientos o miles de kilómetros.

1.2 Motivación

Teniendo en cuenta lo expuesto anteriormente, este TFG surge para dar la posibilidad tanto a alumnos como a autodidactas de ceder su conocimiento compartiendo sus apuntes.

Muchas veces, en época de exámenes, hemos querido tener siempre los mejores apuntes, los más completos o quizá simplemente los más atractivos de cara a pasar mucho tiempo estudiando; pero esta búsqueda de los “apuntes de oro” normalmente se reduce a los compañeros de clase o como mucho a compañeros de otros cursos.

Por ello surge *Notes4all*, una plataforma tanto web como móvil en la que los usuarios podrán publicar sus apuntes y así poder ayudar a otra persona que esté interesada en la misma materia. De esta manera se podrá compartir el conocimiento entre personas de diferentes niveles, entre diferentes centros educativos o incluso entre compañeros que se sientan juntos cada día, sin tener que hacer una foto a cada folio o pasando horas y horas pasando a “limpio” apuntes.

1.3 Objetivos

A nivel personal el principal objetivo es el de convertir una simple idea en una plataforma totalmente estable, funcional, ¿y por qué no?, de uso diario para ciertas personas que encuentren en ella lo que necesitan.

A nivel funcional, el principal objetivo de este trabajo se centra en el diseño e implementación de una plataforma tanto web como móvil que permita la compartición de apuntes o documentos, validados automáticamente por el sistema, entre los usuarios del mismo.

Para llevar a cabo este objetivo general, descrito anteriormente, se han de implementar los objetivos más específicos que se encuentran desglosados en la siguiente tabla (*Tabla 1*):

ID	Objetivo
OBJ-1	Facilitar la obtención de nuevos conocimientos a través de las publicaciones realizadas por los usuarios del sistema.
OBJ-2	Implementar una <i>API REST*</i> que actuará como núcleo entre la aplicación web y móvil, por lo que deberá estar totalmente desacoplada de las mismas.
OBJ-3	Implementar dos plataformas, una web y otra móvil, que tengan un núcleo común descrito en el objetivo anterior.
OBJ-4	Implementar un sistema automático que valide dichas publicaciones.
OBJ-5	Permitir a los usuarios registrados en el sistema realizar publicaciones en las diferentes categorías disponibles en el sistema.
OBJ-6	Permitir que los usuarios puedan consultar y/o descargar los documentos disponibles en el sistema.
OBJ-7	Implementar un sistema fácilmente escalable con módulos totalmente independientes y así dejarlo preparado para futuras mejoras.
OBJ-8	Implementar un módulo de administración sencillo e intuitivo para los administradores del sistema.

Tabla 1 - Tabla de objetivos

1.4 Alcance

A continuación y como preámbulo de lo que se va a tratar en este apartado, se muestra el árbol de características* del sistema (*Figura 1*), que servirá de gran ayuda para identificar, representar y entender de una forma directa y sencilla las distintas funcionalidades que comparten tanto la aplicación web como la aplicación móvil.

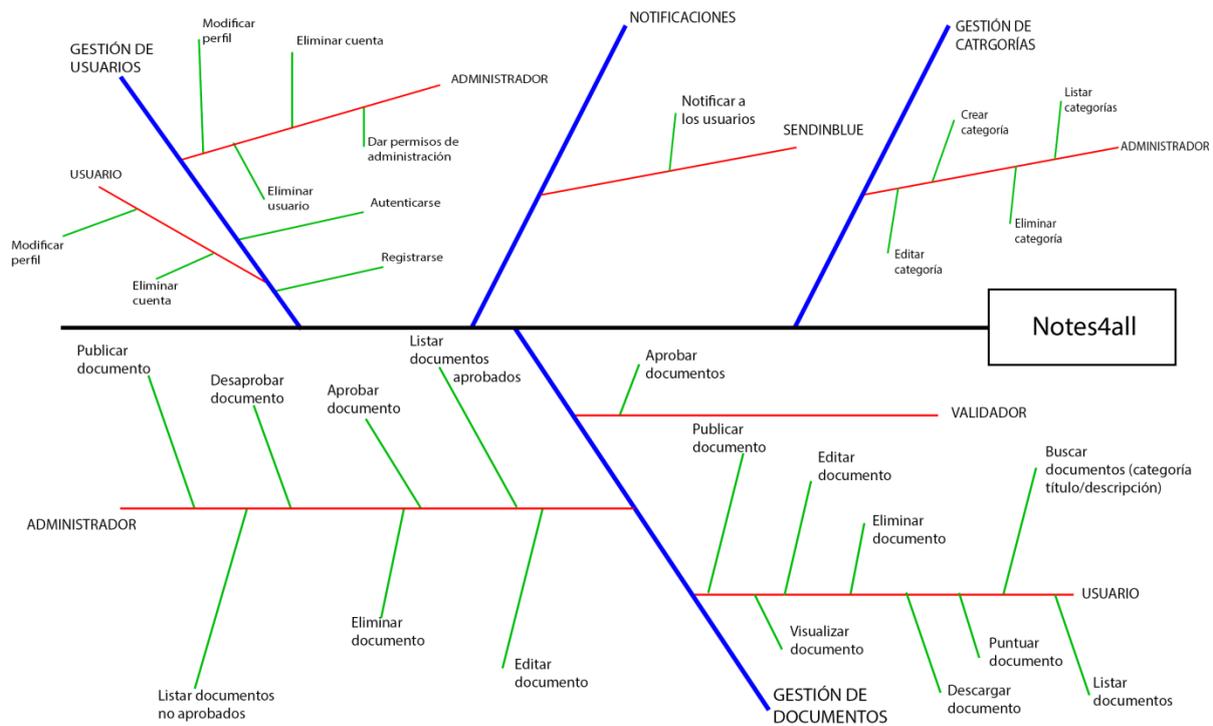


Figura 1 - Árbol de características

Se procede a explicar el árbol de características, que como se puede observar, cuenta con un total de tres componentes o características principales, que son:

- **GESTIÓN DE USUARIOS:** Es el módulo dedicado a los usuarios del sistema, en el cual, las acciones estarán restringidas por tipo de usuario.
 - **ADMINISTRADOR:** Los usuarios del sistema que tengan permisos de administración podrán:
 1. **Modificar su perfil.**
 2. **Eliminar su cuenta.**
 3. **Eliminar usuarios.**
 4. **Dar permisos de administración a otros usuarios.**
 - **USUARIO:** Los usuarios que no tengan permisos de administración podrán:
 1. **Modificar su perfil.**
 2. **Eliminar su cuenta.**
- **GESTIÓN DE CATEGORÍAS:** Es el módulo dónde se administran las categorías disponibles en el sistema. Únicamente los administradores podrán acceder a este módulo y tendrán disponibles las siguientes acciones:
 1. **Listar categorías.**
 2. **Crear categorías.**
 3. **Editar categorías.**
 4. **Eliminar categorías.**

- **GESTIÓN DE DOCUMENTOS:** Es el módulo más importante de la plataforma ya que se corresponde con el modelo de negocio de la misma. A continuación se detallan las acciones que se podrán llevar a cabo:
 - **VALIDADOR:** Se trata del proceso autónomo encargado de validar los documentos publicados en el sistema, su única acción es la de **validar documentos**.
 - **ADMINSTRADORES:** Podrán realizar las siguientes acciones:
 1. **Listar documentos aprobados.**
 2. **Listar documentos no aprobados.**
 3. **Aprobar documentos.**
 4. **Desaprobar documentos.**
 5. **Publicar documentos.**
 6. **Editar documentos.**
 7. **Eliminar documentos.**
 - **USUARIOS:** Los usuarios no administradores podrán:
 1. **Publicar documentos.**
 2. **Actualizar sus documentos.**
 3. **Eliminar sus documentos.**
 4. **Buscar documentos (por título y/o categoría).**
 5. **Visualizar documentos.**
 6. **Descargar documentos.**
 7. **Puntuar otros documentos.**
 8. **Listar documentos.**

1.6 Entorno tecnológico

En este apartado se presentarán las herramientas, plataformas o *frameworks** empleados para poder desarrollar este proyecto final de grado.

1.6.1 Plataformas o *frameworks*

- **Visual Studio Code**

Visual Studio Code es un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*) desarrollado por Microsoft y puede ser utilizado en Windows, Linux y macOS. Permite programar sobre una gran cantidad de lenguajes e instalar infinidad de extensiones para cada uno de ellos.

Esta herramienta es esencial ya que la he utilizado como entorno principal y es donde se ha construido la parte web del proyecto.



Figura 2 - Logo de Visual Studio Code

- **Node.js**

Node.js es un entorno de ejecución en tiempo real de *JavaScript** y fue creado por los desarrolladores de *JavaScript*. Node.js se basa en un modelo de entrada y salida no bloqueantes, es decir, peticiones y respuestas. Dicho modelo se controla por eventos lo que le hace ser muy ligero y eficiente; por ello dónde realmente brilla Node.js es en la creación de aplicaciones de red rápidas debido a su capacidad de gestionar una gran cantidad de conexiones simultáneas, lo que se traduce en una alta escalabilidad.



Figura 3 - Logo de Node.js

- **Angular (versión 8)**

Angular es un potente *framework* basado en *JavaScript*. El desarrollo en Angular se realiza con *TypeScript*. *TypeScript* se puede describir como un súper conjunto de *JavaScript* que ofrece muchas herramientas adicionales al lenguaje.

A lo que a Angular se refiere, éste ofrece una base para el desarrollo aplicaciones robustas, escalables y optimizadas en la parte del cliente utilizando el estilo *Single Page Application (SPA*)*.



Figura 4 - Logo de Angular

- **Node Package Manager (NPM)**

Node.js se encuentra estructurado en módulos por lo que, a medida que avanza un desarrollo, se podrán agregar estos módulos en función de las necesidades. *Node Package Manager*, o simplemente NPM es el gestor de paquetes encargado de gestionar estos módulos en nuestros desarrollos. NPM permite agregar dependencias de una forma simple, distribuir paquetes y administrar de una forma eficaz el proyecto que desarrollemos.



Figura 5 - Logo de NPM

- **MongoDB**

MongoDB es una base de datos no relacional, distribuida y orientada a documentos, lo que quiere decir que no almacena los datos en registros sino en documentos. Estos documentos son almacenados en *BSON*, que es una representación binaria de *JSON* (*JavaScript Object Notation*).

Una de las diferencias primordiales con las bases de datos relacionales, es que no es necesario seguir un esquema en los documentos que almacena, es decir, diferentes documentos de una misma colección (concepto similar a una tabla en una base de datos relacional) pueden tener esquemas diferentes.

Es recomendable utilizar Mongo en entornos que requieran escalabilidad horizontal, en entornos con recursos limitados o entornos con una alta disponibilidad.



Figura 6 - Logo de MongoDB

- **Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para implementación de aplicaciones Android, está basado en IntelliJ IDEA, lo que quiere decir que cuenta con un potente editor de código y una amplia variedad de herramientas destinadas a los desarrolladores. Además de esto, Android Studio ofrece una serie de funciones destinadas a aumentar la productividad del desarrollador, como por ejemplo:

- Emulador integrado en el IDE.
- Integración con GitHub.
- Herramienta de *Lint* para identificar problemas de rendimiento, compatibilidad de la versión o usabilidad entre otros.



Figura 7 - Logo de Android Studio

- **GitHub**

GitHub es una plataforma de desarrollo software colaborativa que se encarga del control de versiones de un proyecto, para ello utiliza Git como sistema de control de versiones. GitHub cómo plataforma permite almacenar proyectos o código de forma tanto pública como privada. Aunque existen otras plataformas similares, gracias a GitHub podemos mantener un repositorio donde almacenar de forma segura, con control de cambios y respaldados por copias nuestros proyectos.

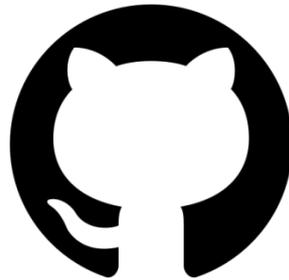


Figura 8 - Logo de GitHub

- **Google Cloud Platform (GCP)**

Google Cloud Platform es un conjunto de servicios que funcionan en la misma infraestructura que utiliza Google. Las herramientas que proporciona GCP abarca *Cloud Computing, Networking, Data Storage, Machine Learning, etc.* Google Cloud Platform se va a utilizar en este TFG ya que proporciona un servicio de un servidor virtual privado (*virtual private server*) gratuito de un año o el uso equivalente a 300 dólares. Este servidor será vital para poder desplegar las aplicaciones.



Google Cloud Platform

Figura 9 - Logo de Google Cloud Platform

- **NGINX**

Servidor web utilizado para ejecutar **Notes4all**. Se trata de un conocido software de código abierto el cual se puede emplear como servidor web HTTP, proxy inverso, balanceador de carga o proxy de correo electrónico.

NGINX surgió como intento de solucionar el resto de gestionar diez mil conexiones a la vez en el año 2002. Hoy en día los servidores web deben de soportar un mayor número de conexiones por eso, NGINX tiene una arquitectura asíncrona y controlada por eventos.



Figura 10 - Logo de NGINX

1.6.1 Herramientas de soporte

- **Draw.io:** editor de diagramas gratuito con versión en la nube y de escritorio que permite exportar las figuras a múltiples formatos como PDF o PNG.
- **Microsoft Word 2010:** es un procesador de textos utilizado para realizar esta documentación.
- **Photoshop:** aplicación de escritorio utilizada para la edición de imágenes. Ha sido empleada para la composición de ciertas imágenes presentes en la documentación.
- **GanttProject:** es un software de código abierto con licencia GPL cuyo objetivo es la administración de proyectos empleando el diagrama de Gantt.
- **Ninjamock:** es una herramienta que facilita la creación de diseños o mocks para diferentes dispositivos, por ejemplo un móvil o un navegador.

1.7 Organización del documento

En esta sección, se describe la organización de los capítulos que conforman esta documentación. La memoria, consta de ocho capítulos, las referencias utilizadas y dos índices de figuras y tablas contenidas en el documento. A continuación se realizará una breve descripción de cada uno de los capítulos de la memoria.

- **Capítulo 1: Introducción.** En este capítulo se detallan los motivos que han llevado a la elección de este proyecto, además de definir el alcance del sistema y los objetivos que se pretenden con su desarrollo.
- **Capítulo 2: Estado del arte.** En este capítulo se presenta el estudio realizado sobre las aplicaciones similares a la que se va a desarrollar y sobre el entorno tecnológico que la rodea.
- **Capítulo 3: Planificación.** En esta sección se realizará un estudio detallado de la metodología empleada para llevar a cabo la planificación y estimación del proyecto. Además, se presentará un presupuesto del desarrollo.
- **Capítulo 4: Análisis.** En este capítulo se exponen todas las características representativas de cada uno de los sistemas que se van a desarrollar. Entre dichas características, se pueden encontrar los requisitos funcionales o los actores del sistema.
- **Capítulo 5: Diseño e implementación.** En esta sección se representarán las arquitecturas, los aspectos relevantes del proceso de implementación y las interfaces de usuario.
- **Capítulo 6: Pruebas.** En este capítulo se recogen las pruebas que se han realizado en los sistemas para verificar su correcto funcionamiento.
- **Capítulo 7: Manuales.** En este capítulo se detallan las guías que permitirán a los diferentes usuarios del sistema emplear su función correctamente. En ellos se recogen desde la instalación de la plataforma hasta el uso de la misma.
- **Capítulo 8: Conclusiones y mejoras.** En esta sección se realizará una retrospectiva a modo de resumen de lo que ha sido el desarrollo del proyecto y se analizará el resultado del mismo concluyendo con un listado de posibles mejoras.
- **Referencias.** En este apartado se recogen las diferentes referencias webgráficas utilizadas durante el desarrollo del proyecto.

1.8 Contenido del CD-ROM

Junto con este documento, se adjunta el CD-ROM del proyecto el cual contiene los siguientes archivos:

- **Documentación.pdf:** Documento en formato .pdf que contiene la memoria del proyecto.
- **notes4all_front:** Carpeta que contiene el código de la aplicación cliente, es decir, el *frontend*.
- **notes4all_back:** Carpeta que contiene el código del servidor, es decir, el *backend*.
- **notes4all:** Carpeta que contiene el código de la aplicación móvil.
- **notes4all.apk:** Fichero de instalación de la aplicación móvil.
- **deploy.sh:** Fichero con el código necesario para la realización del despliegue de la aplicación web.

Capítulo 2 – Estado del arte

A día de hoy existen múltiples webs donde es posible subir tus apuntes y descargar los de otros usuarios. Investigando cuáles son las más reconocidas o utilizadas he destacado las tres siguientes:

Whuolah

Whuolah quizá sea la aplicación más llamativa a nivel de diseño, pero deja un poco que desear en cuanto a su funcionalidad, tanto la búsqueda como la publicación de documentos son poco intuitivas.

La parte más destacada de esta aplicación es que cuenta con un sistema de calificación de publicadores bastante efectiva, además de un sistema de recomendación de los usuarios más populares y activos del sistema.

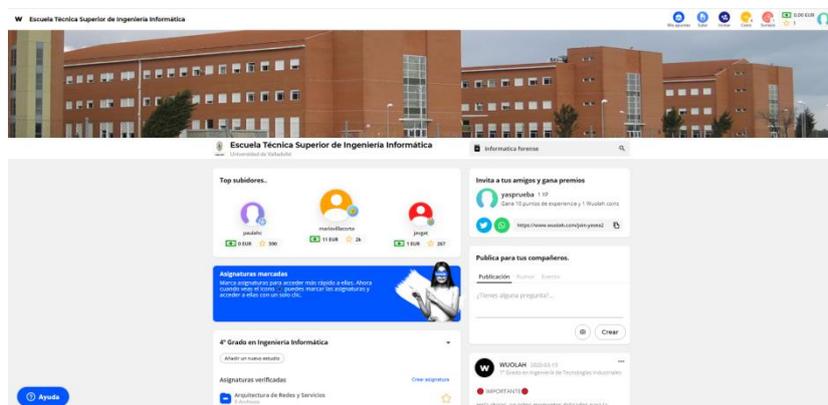


Figura 11 - Whuolah

Doccity

Doccity tiene un perfil similar a Whuolah, también cuenta con un efectivo sistema de puntuación de usuarios, pero la búsqueda de documentos también es un poco ineficiente ya que muestra un listado interminable de documentos, categorías y títulos que en ocasiones confunden un poco a la vista.

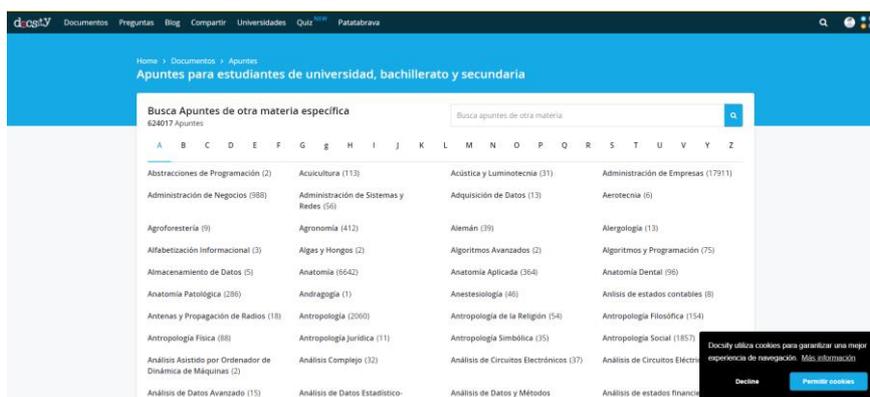


Figura 12 - Doccity

Rincón del vago

El rincón del vago, quizá sea el sistema más antiguo y reconocido de los que hay en la actualidad por la cantidad de años que lleva disponible. Se puede apreciar que se ha quedado un poco anticuado en cuanto al diseño y funcionalidades, ya que no cuenta con un perfil de usuario, ni de puntuación. También las búsquedas de documentos son algo incómodas ya que muestra un listado interminable de títulos o categorías.



Figura 13 - Rincón del vago

A continuación se muestra una comparativa de las aplicaciones analizadas frente a la propuesta de este trabajo final de grado. *Notes4all* cuenta con la mayoría de funcionalidades analizadas, e incluso, las mejora o aporta otras nuevas.

	<i>Whuolah</i>	<i>Docsity</i>	<i>Rincón del vago</i>	<i>Notes4all</i>
Publicación de documentos	✓	✓	✓	✓
Calificación de documentos	✓	✗	✗	✓
Validación automática de documentos	✗	✗	✗	✓
Usabilidad intuitiva	✗	✗	✗	✓
Calificación de usuarios	✓	✓	✗	✓
Sistema de recomendación	✓	✓	✗	✗
Monetización de documentos	✓	✓	✗	✗

Tabla 2 - Comparativa de aplicaciones

Capítulo 3 – Planificación

3.1 Metodología de trabajo

Después de realizar un análisis exhaustivo de las diferentes metodologías de desarrollo, he seleccionado el modelo iterativo incremental ya que es el más adecuado para llevar a cabo este proyecto.

Esta metodología se basa en realizar pequeñas etapas o iteraciones de forma repetitiva y las ventajas que tiene sobre otro tipo de metodologías son: la reducción del tiempo de desarrollo del sistema y la posibilidad de añadir requisitos nuevos en cada iteración, si se diese el caso de que estos no se han detectado en las etapas de análisis anteriores. Cada iteración está formada por las siguientes fases:

- **Análisis:** durante esta fase se hace una extracción, lo más completa posible, de los requisitos del sistema para realizar un correcto desarrollo del mismo.
- **Diseño:** durante esta fase, se establecen los diseños de las aplicaciones a nivel de arquitectura, funcionalidad y estilo.
- **Implementación:** en esta fase, se desarrolla el código de las aplicaciones partiendo del análisis y el diseño realizado anteriormente.
- **Pruebas:** al terminar la fase de implementación, se realiza una batida de pruebas con el objetivo de comprobar que el sistema cumple con los objetivos propuestos en la iteración correspondiente.
- **Documentación:** última fase de cada iteración, que consiste en plasmar todo lo realizado anteriormente en un documento.

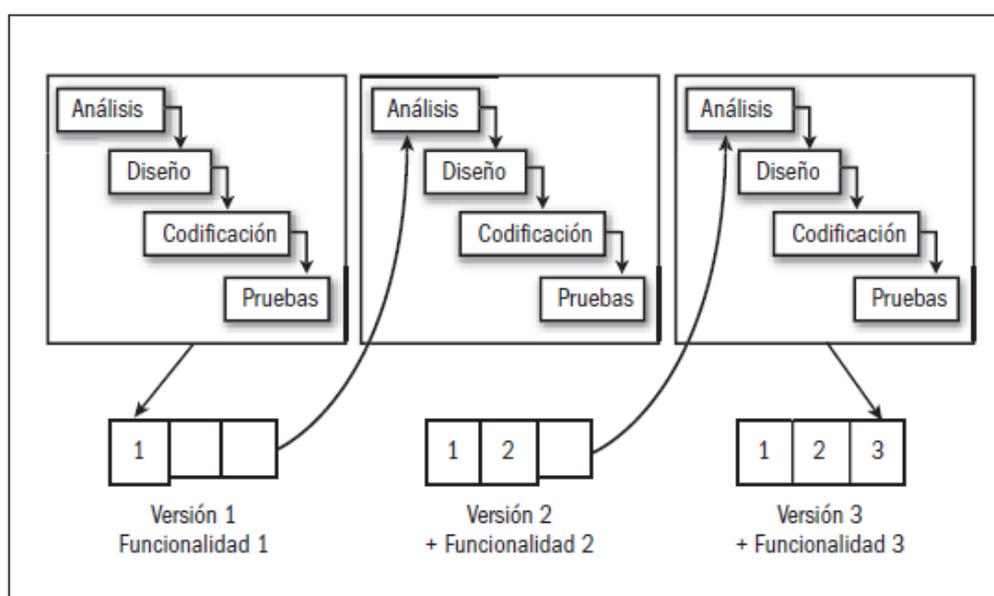


Figura 14 - Modelo iterativo incremental

3.2 Planificación temporal

Una vez seleccionada la metodología a emplear durante el desarrollo del proyecto, se va a realizar la planificación de las tareas que se van a llevar a cabo con el objetivo de estimar la fecha de finalización del proyecto.

La planificación inicial del proyecto es desde el 18 de noviembre de 2019 hasta el 30 de junio de 2020. Durante este tiempo se llevarán a cabo 5 iteraciones y se establece una jornada de trabajo de 4 horas diarias de lunes a jueves durante este periodo de 7 meses y medio, lo que hace un total de 682 horas de trabajo.

En las primeras iteraciones se dedicará más tiempo a las fases de análisis, diseño y documentación que a la implementación con el objetivo de definir una correcta estructura del proyecto y así poder dedicar más tiempo a la implementación y las pruebas en las siguientes iteraciones.

A continuación se puede apreciar la distribución temporal de las fases en cada iteración:

[-] ● Notes4all	18/11/19	30/06/20
[-] ● Desarrollo del backend	18/11/19	31/12/19
● Análisis	18/11/19	3/12/19
● Diseño	3/12/19	17/12/19
● Implementación	17/12/19	24/12/19
● Pruebas	24/12/19	24/12/19
● Documentación	24/12/19	31/12/19
[-] ● Frontend del usuario no autenticado	31/12/19	12/02/20
● Análisis	31/12/19	10/01/20
● Diseño	10/01/20	21/01/20
● Implementación	21/01/20	5/02/20
● Pruebas	5/02/20	5/02/20
● Documentación	6/02/20	12/02/20
[-] ● Frontend del usuario autenticado	12/02/20	31/03/20
● Análisis	12/02/20	24/02/20
● Diseño	24/02/20	4/03/20
● Implementación	4/03/20	17/03/20
● Pruebas	18/03/20	25/03/20
● Documentación	25/03/20	31/03/20
[-] ● Validador de documentos	31/03/20	12/05/20
● Análisis	31/03/20	2/04/20
● Diseño	3/04/20	7/04/20
● Implementación	8/04/20	27/04/20
● Pruebas	27/04/20	5/05/20
● Documentación	6/05/20	12/05/20
[-] ● Aplicación móvil	12/05/20	30/06/20
● Análisis	12/05/20	14/05/20
● Diseño	15/05/20	20/05/20
● Implementación	21/05/20	11/06/20
● Pruebas	11/06/20	22/06/20
● Documentación	23/06/20	30/06/20

Figura 15 - Planificación temporal

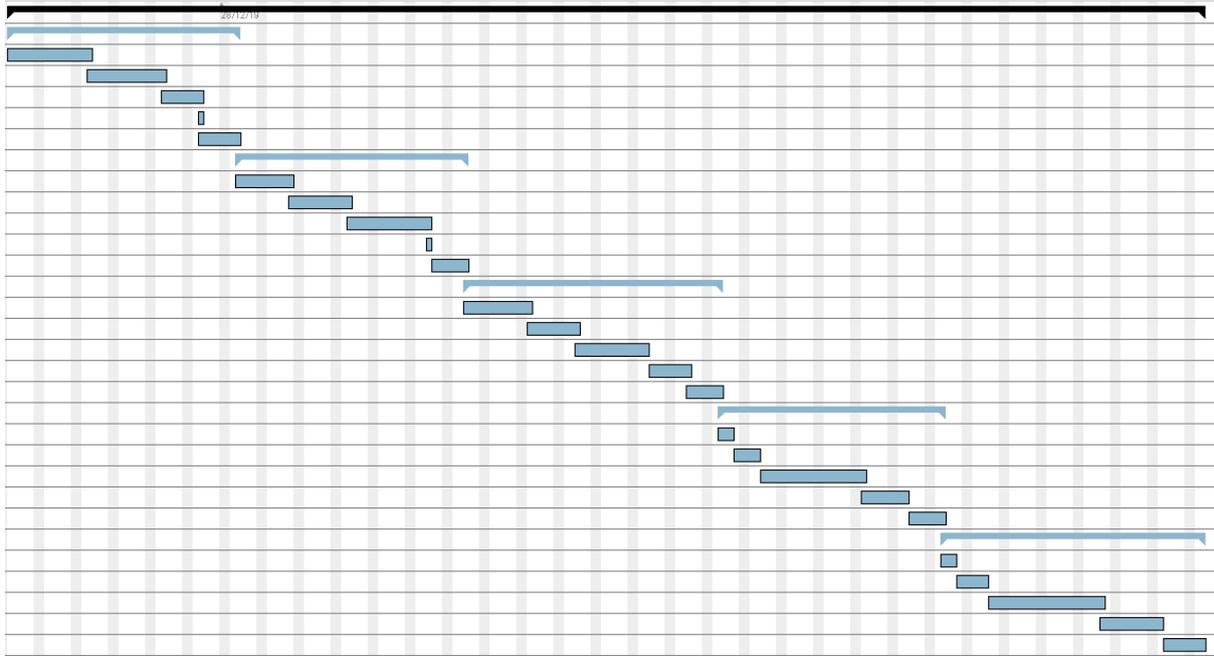


Figura 16 - Diagrama de Gantt de la planificación temporal

3.2 Análisis

3.2.1 Coste de los componentes hardware

En la tabla siguiente, se muestran los costes de los componentes hardware que se han empleado para el desarrollo de las aplicaciones. También, se ha tenido en cuenta el tiempo de uso que se ha dado a cada componente en función del tiempo de vida estimado. Así, se calcula el coste real de cada componente durante el desarrollo del sistema.

Componente	Precio	Vida útil	% uso	Coste real
Ordenador	1500,00€	5 años	25%	375,00€
Monitor 1	160,00€	5 años	20%	40,00€
Monitor 2	150,00€	5 años	20%	30,00€
Ratón	50,00€	2 años	20%	10,00€
Teclado	75,00€	2 años	20%	15,00€
Móvil	140,00€	2 años	50%	70,00€
Conexión a internet	50,00€/mes	8 meses	-	400,00€
TOTAL				940,00€

Tabla 3 - Costes de componentes hardware

3.2.2 Coste de los componentes software

En la siguiente tabla se muestran los costes de los componentes software que se han utilizado para desarrollar el proyecto (aplicaciones y documentación). Para hallar el coste de cada componente, se ha empleado el mismo método que en el cálculo de los costes hardware.

Componente	Precio	Vida útil	% uso	Coste real
Ubuntu 18.4	0,00€	4años	25%	0,00€
Windows 10	80,00€	4 años	25%	20,00€
Microsoft Office 2010	70,00€	5 años	20%	10,00€
Draw.io	0,00	-	-	0,00€
Servidooor GCP	4,00€/mes	8 meses	100%	32,00€
Visual Studio Code	0,00€	-	100%	0,00€
Android Studio	0,00	-	100%	0,00€
Github	0,00€	8 meses	100€	0,00€
Photoshop	30,00€/mes	5 meses	-	150,00€
TOTAL				212,00€

Tabla 4 - Costes de componentes software

3.2.3 Coste del personal

El desarrollo del proyecto ha sido llevado a cabo por una única persona desarrollando diferentes roles que son: analista, desarrollador y tester; dependiendo de cada tipo de tarea. En la siguiente tabla, se puede apreciar los costes de cada rol en función del coste por hora y las horas dedicadas a cada tarea teniendo en cuenta la planificación inicial:

Rol	Tiempo (horas)	Coste/hora	Coste real
Analista	240	16,20€/h	3.888,00€
Desarrollador	350	13,00€/h	4.550,00€
Tester	92	11,50€/h	1.058,00€
TOTAL			9496,00€

Tabla 5 - Costes de personal

3.2.4 Método de puntos de función

La estimación mediante puntos de función, aproxima el coste del proyecto a través de la evaluación de todas sus funciones, ya que dependiendo de la complejidad de cada una de ellas, así será el coste de su desarrollo.

La aplicación se divide en cinco grupos con el objetivo de establecer los puntos de función; cada uno de estos grupos, contiene los diferentes componentes a implementar y, en función del grupo, se asignará una complejidad. Los cinco grupos son:

- **Entradas de usuario:** son los datos que el usuario introduce en el sistema.
- **Salidas de usuario:** son los datos que el sistema muestra al usuario.
- **Consultas externas:** son peticiones que realiza el usuario a un sistema externo el cual interactúa con la aplicación.
- **Ficheros lógicos internos:** es la propia base de datos de la aplicación.
- **Ficheros lógicos externos:** son bases de datos externas al sistema las cuales pueden ser utilizadas por la aplicación.

Una vez se han repartido las diferentes funcionalidades del sistema en los diferentes grupos, se hará uso de la siguiente tabla para poder establecer la complejidad de cada una de ellas.

Para Ficheros Lógicos Internos y Ficheros Externos de Interfaz				Para Salidas Externas y Consultas Externas				Para Entradas Externas			
Nº de tipos de registro	Tipos de Datos distintos			Nº de ficheros referenc	Tipos de Datos distintos			Nº de ficheros referenc	Tipos de Datos distintos		
	1-19	20-50	+51		1-5	6-19	+20		1-4	5-15	+16
1	Baja	Baja	Media	0 - 1	Baja	Baja	Media	0 - 1	Baja	Baja	Media
2 - 5	Baja	Media	Alta	2 - 3	Baja	Media	Alta	2 - 3	Baja	Media	Alta
+6	Media	Alta	Alta	+4	Media	Alta	Alta	+3	Media	Alta	Alta

Tabla 6 – Complejidad

A continuación, se definen las funcionalidades asociadas a cada uno de los grupos definidos anteriormente.

- **Entradas del usuario**

ENTRADAS	COMPLEJIDAD
Formulario de registro	Media
Formulario de inicio de sesión	Media
Formulario de búsqueda	Media
Formulario de publicación	Alta
Validación de un documento	Alta
Formulario de creación de categorías	Alta
Formulario de edición de categorías	Baja
Formulario de edición de documentos	Baja
Formulario de edición del perfil	Media
Formulario de puntuación de documentos	Media

Tabla 7 - Entradas de usuario

- **Salidas del usuario**

SALIDAS	COMPLEJIDAD
Listado de documentos	Alta
Listado de la búsqueda	Alta
Listado de categorías	Media
Listado de usuarios	Media
Listado de documentos auto aprobados	Media
Información de un documento	Baja
Información del usuario	Baja
Información de una categoría	Baja

Tabla 8 - Salida del usuario

- **Fichero lógicos internos**

FICHEROS LÓGICOS INTERNOS	COMPLEJIDAD
Base de datos mongo	Alta
Autenticación con <i>json web token</i>	Alta

Tabla 9 - Ficheros lógicos internos

Una vez que se han establecido los puntos de función sin ajustar (PFNA), se realiza la suma de todos ellos con ayuda de una ponderación en función del grupo al que pertenece cada funcionalidad y su complejidad.

En la siguiente tabla se representa la suma de los puntos de función sin ajustar y la ponderación correspondiente a cada grupo y complejidad:

	Peso de la complejidad baja * n°	Peso de la complejidad media * n°	Peso de la complejidad alta * n°
Entradas	3x2 = 6	4x5 = 20	6x3 = 18
Salidas	3x3 = 9	4x3 = 12	6x2 = 12
Consultas externas	4x0 = 0	5x0 = 0	7x0 = 0
Ficheros internos	7x0 = 0	10x0 = 0	15x2 = 30
Ficheros externos	5x0 = 0	7x0 = 0	10x0 = 0
TOTAL = 14 + 32 + 60 = 106 PFNA			

Tabla 10 - Puntos de función sin ajustar (PFNA)

Una vez se ha obtenido el total de los puntos de función sin ajustar, es necesario hallar el Factor de Ajuste (FA). Para poder hallarlo, hay que basarse en 14 factores que miden la funcionalidad y la complejidad general del sistema. Así que es necesario ponderar cada uno de estos factores en una escala de cero a cinco, siendo cero el valor más bajo y cinco el más alto. Dicha ponderación se representa en la siguiente tabla:

Factor de ajuste	Complejidad
Comunicación de datos	4
Funciones distribuidas	3
Prestaciones	3
Gran uso de la configuración	1
Velocidad de las transacciones	3
Entrada online de datos	5
Diseño para la eficiencia del usuario final	3
Actualización de datos online	5
Complejidad de procesos lógicos internos	4
Reusabilidad del código	3
Facilidad de instalación	3
Facilidad de operación	4
Localizaciones múltiples	1
Facilidad de cambios	3
TOTAL: 45	

Tabla 11 - Factores de ajuste

Para hallar el factor de ajuste (FA), se debe emplear la siguiente fórmula:

$$FA = 0,65 + 0,01 * Total del factor de ajuste$$

$$FA = 0,65 + 0,01 * 45$$

$$FA = 0,65 + 0,45$$

$$FA = 1,1$$

Por los tanto, se puede afirmar que con un factor de ajuste de 1,1, se puede calcular los puntos de función ajustados. Para ello se aplica la siguiente fórmula:

$$PFA = PFNA * FA$$

$$PFA = 106 * 1,1$$

$$PFA = 116,6$$

Para acabar, es necesario hallar el cálculo de la estimación de tiempo. Para ello tenemos en cuenta, que en este caso un punto de función ajustado equivale a cuatro horas de trabajo

$$Tiempo = PFA * horas * PFA$$

$$Tiempo = 116,6 * 4$$

$$Tiempo = 466,4 \text{ horas de duración}$$

Una vez se ha obtenido la estimación de la duración del proyecto, es necesario hallar de nuevo los costes de personal. Para ello, se utilizará la misma estimación temporal que se utilizado anteriormente, de forma que las estimaciones quedan de la siguiente forma: 51,3% el desarrollador, 35,2% el analista y el 13,5% el tester

Rol	Tiempo (horas)	Coste/hora	Coste real
Analista	164,2	16,20€/h	2.660,04€
Desarrollador	239,3	13,00€/h	3.110,90€
Tester	62,9	11,50€/h	723,35€
TOTAL			6494,29€

Tabla 12 - Costes de personal (puntos de función)

3.2.5 Estimación por COCOMO

COCOMO es un modelo de estimación que calcula el presupuesto en función del número de líneas de código. Dicho presupuesto se calculará a partir de los puntos de función ya obtenidos. Para llevar a cabo la estimación, es necesario elegir uno de los modelos en base a las características del proyecto. Los modelos de COCOMO son:

- **Orgánico:** son proyectos de pequeño tamaño, sin gran complejidad, con requisitos poco rígidos y por un equipo pequeño.
- **Empotrado:** son proyectos grandes, de gran complejidad y con requisitos muy rígidos.
- **Semiempotrado:** se trata de un modelo intermedio entre los dos anteriores, en el que los equipos se coordinan para realizar el desarrollo.

Analizando cada uno de los modelos, se llega a la conclusión de que el modelo semiempotrado es el que mejor se adapta para este proyecto.

Para realizar la estimación por COCOMO, lo primero es estimar el número de líneas de código que tiene el proyecto. Los lenguajes empleados para desarrollar son Java para la aplicación Android y Typescript para la aplicación web, teniendo en cuenta que Typescript está basado en Javascript, podemos adoptar una media de 55 líneas de código por cada punto de función. Por tanto las líneas de código del proyecto son:

$$LDC = PF * estimación\ de\ líneas$$

$$LDC = 106 * 55$$

$$LDC = 5830$$

Ahora es necesario aplicar un factor de esfuerzo basado en la siguiente tabla:

Factores	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad requerida	0.75	0.88	1.00	1.15	1.4	
Tamaño de la base de datos		0.94	1.00	1.08	1.16	
Complejidad del software	0.7.	0.85	1.00	1.15	1.30	1.65
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de memoria			1.00	1.06	1.21	1.56
Volatilidad del hardware		0.87	1.00	1.15	1.30	
Restricciones de tiempo de respuesta		0.87	1.00	1.07		
Calidad de los analistas	1.46	1.19	1.00	0.86	0.71	
Experiencia con el tipo de aplicación	1.29	1.13	1.00	0.91	0.82	
Experiencia con el hardware	1.21	1.10	1.00	0.90		
Experiencia con el lenguaje de programación	1.14	1.07	1.00	0.95		
Calidad de los programadores	1.42	1.17	1.00	0.86	0.70	
Técnicas modernas de programación	1.24	1.10	1.00	0.91	0.82	
Empleo de herramientas	1.24	1.10	1.00	0.91	0.83	
Restricciones a la duración del proyecto	1.23	1.08	1.00	1.04	1.10	

Tabla 13 - Factores de ajuste de COCOMO

	Factor	Valor
Fiabilidad requerida	Alto	1,15
Tamaño de la base de datos	Bajo	0,94
Complejidad del software	Alto	1,15
Restricciones de tiempo de ejecución	Medio	1,00
Restricciones de memoria	Medio	1,00
Volatilidad del hardware	Medio	1,15
Restricciones de tiempo de respuesta	Alto	1,07
Calidad de los analistas	Medio	1,00
Experiencia con el tipo de aplicación	Alto	0,91
Experiencia con el hardware	Alto	0,90
Experiencia con el lenguaje de programación	Alto	0,95
Calidad de los programadores	Medio	1,00
Técnicas modernas de programación	Alto	0,91
Empleo de herramientas	Alto	0,91
Restricciones a la duración del proyecto	Medio	1,00
Total de la media		1,003

Tabla 14 - Factor de ajuste de COCOMO

A continuación, es necesario tener en cuenta el conjunto de factores de ajuste de la siguiente tabla para poder calcular el esfuerzo con las siguientes fórmulas:

MODELO	a	b	c	d
Semiempotrado	3,00	1,12	2,50	0,35

Tabla 15 - Ponderaciones de COCOMO

$$\text{Factor de ajuste} = \prod_{i=0}^n \text{valorFA}(i)$$

Factor de ajuste

$$= 1,15 * 0,94 * 1,15 * 1,00 * 1,00 * 1,15 * 1,07 * 1,00 * 0,91 * 0,90 * 0,95 * 1,00 * 0,91 * 0,91 * 1,00$$

$$\text{Factor de ajuste} = \mathbf{0,985}$$

$$\text{Esfuerzo (E)} = a * (\text{KLDC})^b * \text{factor de ajuste}$$

$$E = 3 * 5,830^{1,12} * 0,985$$

$$\mathbf{E = 21,28 personas/mes}$$

$$\text{Tiempo de desarrollo (Tdev)} = c * E^d$$

$$Tdev = 2,50 * 21,28^{0,35}$$

$$\mathbf{Tdev = 7,29 meses}$$

Una vez que se ha obtenido el esfuerzo y el tiempo de desarrollo, es necesario calcular el esfuerzo nominal, es decir, el número de personas que serían necesarias para realizar el proyecto en 7 meses y medio aproximadamente.

$$N = \frac{E}{Tdev}$$

$$N = \frac{21,28}{7,29}$$

$$\mathbf{N = 2,91 \cong 3 personas}$$

Por tanto, serían necesarias 3 personas para realizar el proyecto en unos 7 meses y medio. Como en este caso, el proyecto ha sido realizado por una única persona, serían necesarios 22 meses para la realización del proyecto, lo que equivale a 1760 horas de trabajo.

Como en la estimación inicial y la estimación con puntos de función, la estimación temporal en función de cada rol es: 51,3% el desarrollador, 35,2% el analista y el 13,5% el tester. Con estos datos, se calcula el coste del personal:

Rol	Tiempo (horas)	Coste/hora	Coste real
Analista	619,52	16,20€/h	10.036,22€
Desarrollador	902,88	13,00€/h	11.737,44€
Tester	237,6	11,50€/h	2.732,4€
TOTAL			24.506,06€

Tabla 16 - Costes de personal por COCOMO

3.2.6 Comparativa de estimaciones

Ahora que ya se han obtenido las estimaciones empleando el método de COCOMO y el de puntos de función, podemos decir que el que obtiene una estimación más aproximada a la realidad es el método de puntos de función ya que la estimación por COCOMO es excesiva para este proyecto. Esto se debe a que COCOMO está pensado para proyectos con lenguajes de programación de bajo nivel.

Si se compara la estimación por puntos de función con la estimación inicial, se puede apreciar que existe una diferencia entre los presupuestos, que puede ser debido, a una mala planificación inicial por una estimación excesiva de horas de trabajo.

3.2.7 Presupuesto final

Para calcular el presupuesto del proyecto, se tienen que seguir los mismos pasos que se han realizado en el análisis, pero empleando los datos que se han obtenido a través de la nueva estimación basada en puntos de función.

Los costes de los componentes software y hardware se mantienen debido a que la duración total (en meses) será la misma, pero lo que debe variar son las horas de trabajo diarias, que se obtienen de la siguiente forma:

$$\text{Horas de trabajo} = \frac{4 * 466,4}{682}$$

$$\text{Horas de trabajo} = 2,74$$

Para el coste del personal, se utilizará el coste obtenido en la estimación por puntos de función ya que el tiempo de desarrollo del proyecto, es el que más se asemeja a la realidad. Por lo que el coste del personal serán 6494,29€

Para obtener el presupuesto final del proyecto, se ha de sumar todos los costes que se han obtenido.

$$\textit{Presupuesto} = \textit{Costes de SW} + \textit{Costes de HW} + \textit{Costes de personal}$$

$$\textit{Presupuesto} = 212,00 + 940,00 + 6494,29$$

$$\textbf{\textit{Presupuesto} = 7646,29€}$$

Capítulo 4 – Análisis

4.1 Descripción de los actores

Se puede definir como actor a todo agente externo al sistema que interactúa con él, considerándose tanto personas como máquinas.

Partiendo de esta definición, a continuación se exponen los actores que interactúan con el software desarrollado en este TGF.

- **Usuario no autenticado**: usuario que interactúa con la aplicación sin autenticarse, el cual solo podrá consultar los documentos.
- **Usuario autenticado**: usuario que se ha registrado en el sistema y ha iniciado sesión, se trata de un actor ficticio ya que no existe por sí mismo, sino que representa de forma genérica, un usuario autenticado (común o administrador).
- **Usuario común**: usuario que interactúa con la plataforma con la posibilidad de realizar publicaciones y puntuar las que consulta.
- **Usuario administrador**: usuario que interactúa con el sistema con los permisos de gestionar documentos, categorías y usuarios.
- **API Rest**: sistema encargado de procesar las peticiones y enviar las respuestas al cliente.
- **Sendinblue**: sistema externo a la plataforma que proporciona un sistema de envío de emails a los usuarios.

En la siguiente figura, se puede observar la jerarquía existente entre los diferentes actores del sistema:

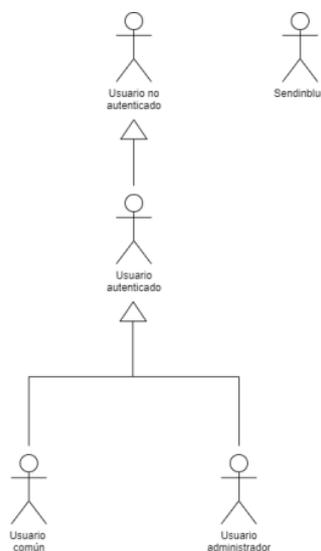


Figura 17 - Jerarquía de los actores del sistema

4.2 Requisitos de usuario

En este apartado, se van a describir los requisitos de usuario, que podemos describirlos como la capacidad o capacidades que tiene un actor en el sistema, es decir, la especificación de acciones que puede llevar a cabo.

En las siguientes tablas se detallan los requisitos de usuario del sistema, tanto de la plataforma web como de la móvil respectivamente.

Requisito	Descripción
RU-01	Un usuario podrá registrarse en el sistema.
RU-02	Un usuario podrá autenticarse en el sistema.
RU-03	Un usuario podrá cerrar sesión.
RU-04	Un usuario podrá realizar búsquedas de documentos.
RU-05	Un usuario podrá visualizar documentos.
RU-06	Un usuario podrá publicar documentos.
RU-07	Un usuario podrá editar documentos.
RU-08	Un usuario podrá eliminar documentos.
RU-09	Un usuario podrá descargar documentos.
RU-10	Un usuario podrá puntuar documentos.
RU-11	Un usuario podrá listar documentos
RU-12	Un usuario podrá aprobar documentos
RU-13	Un usuario podrá desaprobar documentos
RU-14	Un usuario podrá ver su perfil
RU-15	Un usuario podrá editar cuenta
RU-16	Un usuario podrá eliminar cuenta
RU-17	Un usuario podrá dar permisos de administración a usuarios
RU-18	Un usuario podrá listar usuarios
RU-19	Un usuario podrá eliminar usuarios
RU-20	Un usuario podrá listar categorías
RU-21	Un usuario podrá crear categorías
RU-22	Un usuario podrá editar categorías
RU-23	Un usuario podrá eliminar categorías
RU-24	Notificar a los usuarios

Tabla 17 - Tabla de requisitos de usuario de la aplicación web

Requisito	Descripción
RU-01	Un usuario podrá registrarse en el sistema.
RU-02	Un usuario podrá autenticarse en el sistema.
RU-03	Un usuario podrá cerrar sesión.
RU-04	Un usuario podrá realizar búsquedas de documentos.
RU-05	Un usuario podrá visualizar documentos.
RU-06	Un usuario podrá publicar documentos.
RU-09	Un usuario podrá descargar documentos.
RU-10	Un usuario podrá puntuar documentos.
RU-11	Un usuario podrá listar documentos
RU-14	Un usuario podrá ver su perfil
RU-24	Notificar a los usuarios

Tabla 18 - Requisitos de usuario de la aplicación móvil

4.3 Casos de uso

Una vez se han enumerado los requisitos de usuario del sistema, se ha de realizar una especificación más detallada de los mismos, es decir, los casos de uso de los requisitos citados. Para ello se realizan los siguientes diagramas en los que se muestra los diferentes requisitos o acciones en función del actor.

El primer diagrama que se va a presentar, representa las acciones permitidas en la plataforma web que pueden llevar a cabo los diferentes tipos de usuarios que tiene el sistema.

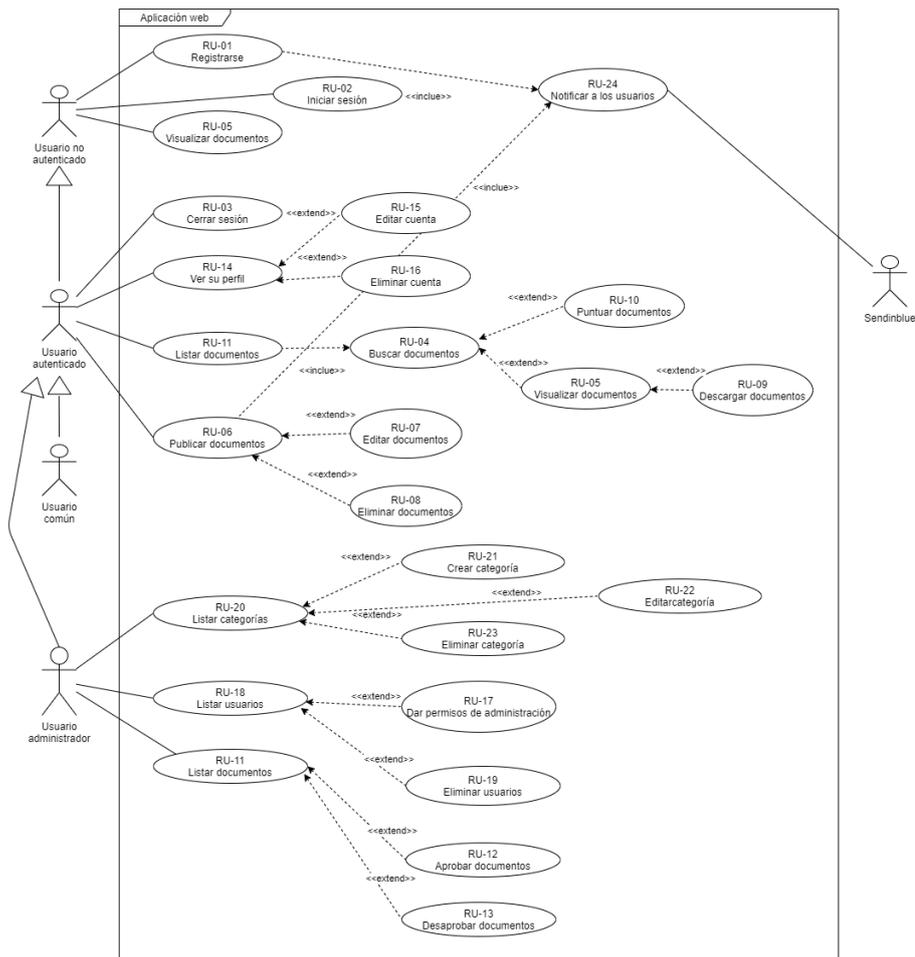


Figura 18 - Diagrama de casos de la aplicación web

Como se puede observar en la figura 11, cada tipo de usuario está relacionado con las acciones que puede llevar a cabo. También se puede ver los diferentes actores que utilizarán el sistema. La gran diferenciación está entre el usuario no autenticado y el usuario autenticado; también tenemos el usuario común y el administrador, que son especificaciones del usuario autenticado.

También se puede observar al usuario encargado de notificar al usuario ciertas acciones (RU-24) como puede ser el registro (RU-01) o la publicación de documentos (RU-05). *Sendinblue* es el actor encargado de notificar dichas acciones a los usuarios a través de un correo electrónico.

En el diagrama siguiente podemos ver las diferentes acciones que puede realizar cada tipo de usuario en la aplicación móvil. Como punto a destacar, podemos ver que en este caso no aparece el usuario administrador ya que su funcionalidad solo está accesible desde la plataforma web.

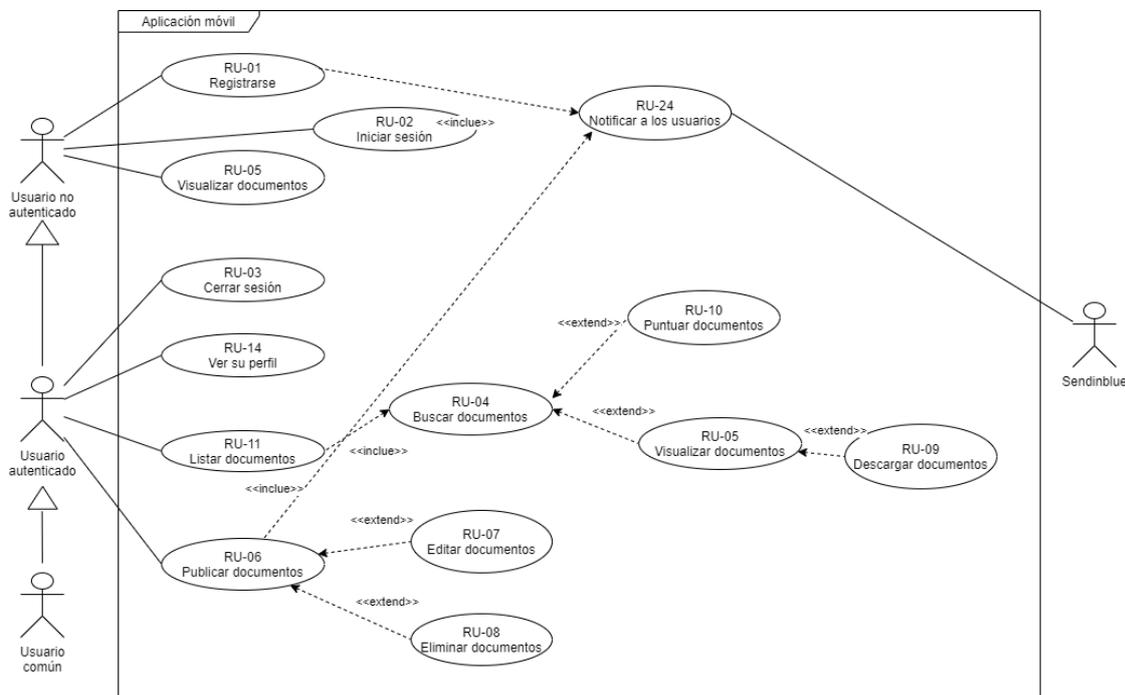


Figura 19 - Diagrama de casos de la aplicación móvil

Al igual que en el diagrama de casos de uso de la aplicación web, vemos al actor *Sendinblue* como el actor encargado de notificar a través de un correo electrónico ciertas acciones que realizan los usuarios.

4.4 Especificación de los requisitos de usuario

CU-01 Registro	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-01 Un usuario podrá registrarse en el sistema
Actor	Usuario no autenticado
Descripción	El sistema permitirá crear al usuario una cuenta para poder acceder al mismo.
Precondiciones	N/A
Flujo normal	<ol style="list-style-type: none"> 1. El usuario solicita crear una cuenta en el sistema. 2. El sistema muestra al usuario los datos que debe rellenar. 3. El usuario rellena dichos datos. 4. El sistema comprueba que los datos sean correctos. 5. El sistema crea un nuevo usuario.
Postcondiciones	La información del usuario se guarda en el sistema y pasa a estar registrado.
Excepciones	<ul style="list-style-type: none"> • Los datos proporcionados ya existen en el sistema, se mostrará un error y se volverá al paso 2. • Los datos proporcionados no son correctos, se mostrará un error y se volverá al paso 2.
Frecuencia	Media
Importancia	Crítica
Prioridad	Alta
Observaciones	N/A

Tabla 19 - Caso de uso "Registro" (CU-01)

CU-02 Iniciar sesión	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-02 Iniciar sesión
Actor	Usuario no autenticado
Descripción	El sistema permitirá al usuario acceder a través de un formulario de inicio de sesión.
Precondiciones	El usuario debe estar registrado en la plataforma.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario solicita acceder a su cuenta. 2. El sistema muestra al usuario los datos que debe rellenar. 3. El usuario envía al sistema los datos rellenados. 4. El sistema verificar que los datos existen y son correctos. 5. El sistema redirige al usuario a la página principal de la plataforma.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Los datos de acceso proporcionados por el usuario son incorrectos. Se mostrará un mensaje de error y se volverá al paso 2.
Frecuencia	Alta
Importancia	Crítica
Prioridad	Alta
Observaciones	

Tabla 20 - Caso de uso "Inicio de sesión" (CU-02)

CU-03 Cerrar sesión	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-03 Cerrar sesión
Actor	Usuario autenticado
Descripción	El sistema permitirá al usuario cerrar la sesión
Precondiciones	El usuario debe estar logueado en la plataforma
Flujo normal	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de cerrar sesión. 2. El sistema procesa la acción eliminando los datos de sesión del usuario. 3. El sistema redirige al usuario a la página principal de la plataforma.
Postcondiciones	El usuario pasa a no estar autenticado en el sistema.
Excepciones	N/A
Frecuencia	Alta
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 21 - Caso de uso "cerrar sesión" (CU-03)

CU-04 Realizar búsquedas	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-04 Un usuario podrá realizar búsquedas de documentos
Actor	Usuario autenticado
Descripción	Un usuario puede realizar búsquedas en el sistema a partir de unos filtros.
Precondiciones	El usuario debe estar autenticado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de búsqueda. 2. El usuario filtra por título y/o categoría. 3. El usuario hace click en el botón “Buscar” 4. El sistema muestra los resultados de la búsqueda
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • No existen resultados de búsqueda: se mostrará un mensaje de error para que el usuario modifique los criterios de búsqueda.
Frecuencia	Media
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 22 - Caso de uso “Realizar búsquedas” (CU-04)

CU-05 Visualizar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-05 Un usuario podrá visualizar documentos
Actor	Usuario no autenticado
Descripción	Un usuario podrá seleccionar un documento para su visualización.
Precondiciones	N/A
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la plataforma. 2. El usuario pulsa el botón de visualizar en un documento. 3. El sistema sirve el documento para su visualización.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • El documento no existe: el sistema mostrará un error informando al usuario de lo acontecido. • El documento no puede mostrarse: el sistema mostrará un error animando al usuario a visualizar otro documento.
Frecuencia	Alta
Importancia	Crítica
Prioridad	Alta
Observaciones	

Tabla 23 - Caso de uso “Visualizar documentos” (CU-05)

CU-06 Publicar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-06 Un usuario podrá publicar documentos
Actor	Usuario autenticado
Descripción	El usuario tiene la capacidad de realizar publicaciones en el sistema.
Precondiciones	El usuario debe estar autenticado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de publicación. 2. El sistema muestra los datos necesarios para la realización de la misma. 3. El usuario rellena los datos requeridos. 4. El sistema verifica que los datos proporcionados son correctos. 5. El sistema crea una nueva publicación.
Postcondiciones	El usuario contará con una publicación más en su contador de publicaciones
Excepciones	<ul style="list-style-type: none"> • Los datos introducidos no son correctos: el sistema informa al usuario a través de un mensaje de error y vuelve al paso 2. • Se produce un error al almacenar la publicación: el sistema informa al usuario a través de un mensaje de error y vuelve al paso 2.
Frecuencia	Alta
Importancia	Crítica
Prioridad	Alta
Observaciones	

Tabla 24 - Caso de uso "Publicar documentos" (CU-06)

CU-07 Editar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-07 Un usuario podrá editar documentos
Actor	Usuario autenticado
Descripción	El usuario tiene la capacidad de editar sus publicaciones.
Precondiciones	El usuario debe estar autenticado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la sección de publicaciones de su perfil. 2. El usuario selecciona una publicación para editar. 3. El sistema muestra los datos que permite editar. 4. El usuario rellena los datos. 5. El sistema verifica que los datos proporcionados son correctos. 6. El sistema actualiza la publicación.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Los datos introducidos no son correctos: el sistema informa al usuario a través de un mensaje de error y vuelve al paso 3. • Se produce un error al actualizar la publicación: el sistema informa al usuario a través de un mensaje de error y vuelve al paso 3.
Frecuencia	Baja
Importancia	Media
Prioridad	Baja
Observaciones	

Tabla 25 - Caso de uso "Editar documentos" (CU-07)

CU-08 Eliminar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-08 Un usuario podrá eliminar documentos
Actor	Usuario autenticado
Descripción	El usuario tiene la capacidad de eliminar sus publicaciones.
Precondiciones	El usuario debe estar autenticado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la sección de publicaciones de su perfil. 2. El usuario selecciona una publicación para eliminar. 3. El sistema pregunta al usuario si quiere eliminar la publicación. 4. El usuario hace click en el botón de aceptar. 5. El sistema elimina la publicación.
Postcondiciones	El usuario contará con una publicación menos en su historial.
Excepciones	<ul style="list-style-type: none"> • Se produce un error al eliminar la publicación: el sistema informara al usuario mediante un mensaje de error y volverá al paso 2.
Frecuencia	Baja
Importancia	Baja
Prioridad	Baja
Observaciones	

Tabla 26 - Caso de uso "Eliminar documentos" (CU-08)

CU-09 Descargar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-09 El usuario podrá descargar documentos.
Actor	Usuario autenticado
Descripción	El sistema permite a los usuarios descargar documentos.
Precondiciones	El usuario debe estar autenticado
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la plataforma. 2. El usuario selecciona un documento para descargar. 3. El sistema recupera el documento y realiza la descarga.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • El documento no existe: el sistema muestra un error e invita al usuario a descargar otro documento. • Se produce un error en la descarga: el sistema muestra un error e invita al usuario a descargar otro documento.
Frecuencia	Alta
Importancia	Alta
Prioridad	Alta
Observaciones	

Tabla 27 - Caso de uso "Descargar documentos" (CU-09)

CU-10 Puntuar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-10 Un usuario podrá puntuar documentos.
Actor	Usuario autenticado
Descripción	El usuario tiene la capacidad de puntuar documentos en el sistema.
Precondiciones	El usuario debe estar autenticado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal del sistema. 2. Pulsa en el botón de puntuar. 3. El sistema muestra los datos de puntuación necesarios. 4. El usuario introduce su puntuación. 5. El sistema verifica y registra la puntuación.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Se produce un error al almacenar la puntuación: el sistema informa mediante un mensaje de error y vuelve al paso 1
Frecuencia	Media
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 28 - Caso de uso "Puntuar documentos" (CU-10)

CU-11 Listar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-11 Un usuario podrá listar documentos
Actor	Usuario autenticado
Descripción	Un usuario tiene la capacidad para listar documentos
Precondiciones	El usuario debe estar autenticado
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la sección de documentos 2. El sistema muestra un listado de los documentos.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Se produce un error al recuperar los documentos: el sistema muestra un mensaje de error.
Frecuencia	Alta
Importancia	Alta
Prioridad	Alta
Observaciones	

Tabla 29 - Caso de uso "Listar documentos" (CU-11)

CU-12 Aprobar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-12 Un usuario podrá aprobar documentos
Actor	Usuario administrador
Descripción	El usuario administrador tendrá la capacidad de aprobar un documento.
Precondiciones	El usuario debe estar logueado como administrador.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de documentos. 2. El sistema muestra un listado de documentos. 3. El usuario selecciona un documento para aprobar. 4. El sistema marca como aprobado el documento.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Se produce un error al listar los documentos: el sistema informa a través de un mensaje. • Se produce un error al aprobar el documento: el sistema muestra un mensaje de error e invita al usuario a que repita la acción.
Frecuencia	Media
Importancia	Alta
Prioridad	Media
Observaciones	

Tabla 30 - Caso de uso "Aprobar documentos" (CU-12)

CU-13 Desaprobar documentos	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-13 Un usuario podrá desaprobar documentos
Actor	Usuario administrador
Descripción	El usuario administrador tendrá la capacidad de desaprobar un documento.
Precondiciones	El usuario debe estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de documentos. 2. El sistema muestra un listado de documentos. 3. El usuario selecciona un documento para desaprobar. 4. El sistema marca como desaprobadado el documento.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Se produce un error al listar los documentos: el sistema informa a través de un mensaje. • Se produce un error al desaprobar el documento: el sistema muestra un mensaje de error e invita al usuario a que repita la acción.
Frecuencia	Media
Importancia	Alta
Prioridad	Media
Observaciones	

Tabla 31 - Caso de uso "Desaprobar documentos" (CU-13)

CU-14 Ver perfil	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-14 Un usuario podrá ver su perfil
Actor	Usuario autenticado (común y administrador)
Descripción	Un usuario tendrá la posibilidad de visualizar su perfil
Precondiciones	El usuario debe de estar autenticado
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de su perfil. 2. El sistema muestra los datos del usuario.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al mostrar los datos del usuario: el sistema muestra un mensaje informativo y redirige al usuario a la página principal del sistema
Frecuencia	Media
Importancia	Alta
Prioridad	Media
Observaciones	

Tabla 32 - Caso de uso "Visualizar perfil" (CU-14)

CU-15 Editar perfil	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-15 Un usuario podrá editar su cuenta/perfil
Actor	Usuario autenticado (común y administrador)
Descripción	Un usuario tendrá la posibilidad de modificar su perfil
Precondiciones	El usuario debe de estar autenticado
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de su perfil. 2. El sistema muestra sus datos y da la opción de editar. 3. El usuario pulsa el botón de editar. 4. El sistema muestra los datos que puede editar. 5. El usuario rellena los datos que quiera editar. 6. El sistema comprueba la validez de los datos y los almacena.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al almacenar los datos a actualizar: el sistema muestra un error informativo y vuelve al paso 4. • Los datos a actualizar no son correctos: el sistema muestra un mensaje de error informando al usuario y vuelve al paso 4.
Frecuencia	Media
Importancia	Alta
Prioridad	Media
Observaciones	

Tabla 33 - Caso de uso "Editar perfil" (CU-15)

CU-16 Eliminar cuenta	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-15 Un usuario podrá eliminar su cuenta
Actor	Usuario autenticado (común y administrador)
Descripción	Un usuario tendrá la posibilidad de eliminar su perfil de la plataforma.
Precondiciones	El usuario debe de estar autenticado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de su perfil. 2. El sistema muestra sus datos y la opción de eliminar la cuenta. 3. El usuario hace click sobre el botón de eliminar cuenta. 4. El sistema pregunta si está seguro de la acción que va a realizar. 5. El usuario confirma dicha acción 6. El sistema elimina los datos de la cuenta del usuario, muestra un mensaje y cierra la sesión del usuario.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Error al eliminar los datos del usuario: el sistema muestra un error informativo e invita al usuario a intentarlo de nuevo.
Frecuencia	Baja
Importancia	Media
Prioridad	Baja
Observaciones	

Tabla 34 - Caso de uso "Eliminar cuenta" (CU-16)

CU-17 Dar permisos de administración	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-16 Un usuario podrá otorgar permisos de administración
Actor	Usuario administrador
Descripción	Un usuario tendrá la capacidad de otorgar permisos de administración a otro.
Precondiciones	El usuario debe de estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de usuarios. 2. El sistema muestra un listado con los usuarios del sistema. 3. El usuario selecciona un usuario para darle los permisos. 4. El sistema almacena los nuevos permisos del usuario.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Se produce un error al mostrar los usuarios: el sistema muestra un mensaje de error informativo. • Se produce un error al almacenar los permisos del usuario: el sistema muestra un error e invita al usuario a intentarlo de nuevo.
Frecuencia	Baja
Importancia	Baja
Prioridad	Baja
Observaciones	

Tabla 35 - Caso de uso "Dar permisos de administración" (CU-17)

CU-18 Listar usuarios	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-17
Actor	Usuario administrador
Descripción	Un usuario administrador tendrá la capacidad de listar los usuarios del sistema.
Precondiciones	El usuario debe de estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de usuarios. 2. El sistema muestra un listado con todos los usuarios de la plataforma.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al recuperar los usuarios: el sistema mostrará un mensaje de error y redirigirá al usuario a la página principal.
Frecuencia	Media
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 36 - Caso de uso "Listar usuarios" (CU-18)

CU-19 Eliminar usuarios	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-18 Un usuario podrá eliminar otros usuarios
Actor	Usuario administrador
Descripción	Un usuario administrador podrá eliminar otros usuarios.
Precondiciones	El usuario debe de estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de usuarios. 2. El sistema muestra un listado con todos los usuarios de la plataforma. 3. El usuario selecciona un usuario a eliminar. 4. El sistema pregunta si realmente quiere realizar la acción. 5. El usuario hace click en el botón de aceptar. 6. El sistema elimina los datos del usuario seleccionado.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al listar los usuarios: el sistema muestra un error y redirige al usuario a la página principal del sistema. • Ocurre un error al eliminar el usuario: el sistema muestra un error informativo e invita al usuario a repetir la acción.
Frecuencia	Baja
Importancia	Baja
Prioridad	Baja
Observaciones	

Tabla 37 - Caso de uso "Eliminar usuarios" (CU-19)

CU-20 Listar categorías	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-19 Un usuario podrá listar categorías
Actor	Usuario administrador.
Descripción	Un usuario administrador podrá listar las categorías del sistema
Precondiciones	El usuario debe de estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de categorías. 2. El sistema muestra un listado con todas las categorías del sistema.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al devolver el listado de categorías: el sistema muestra un error informativo y redirige al usuario a la página principal de la plataforma
Frecuencia	Media
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 38 - Caso de uso "Listar categorías" (CU-20)

CU-21 Crear categorías	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-20 Un usuario podrá crear categorías.
Actor	Usuario administrador.
Descripción	Un usuario administrador tendrá la capacidad de crear categorías en el sistema.
Precondiciones	El usuario debe de estar logueado como administrador.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de categorías 2. El sistema muestra un listado con las categorías de la plataforma. 3. El usuario hace click en el botón de crear. 4. El sistema muestra los datos necesarios para la creación. 5. El usuario rellena los datos requeridos. 6. El sistema verifica los datos introducidos. 7. El sistema crea una categoría nueva.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Los datos introducidos son erróneos: el sistema muestra un mensaje informativo y vuelve al paso 4. • Ocurre un error al crear la categoría: el sistema muestra un mensaje informativo y vuelve al paso 4.
Frecuencia	Alta
Importancia	Alta
Prioridad	Alta
Observaciones	

Tabla 39 - Caso de uso "Crear categorías" (CU-21)

CU-22 Editar categoría	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-21 Un usuario podrá editar categorías.
Actor	Usuario administrador.
Descripción	Un usuario administrador podrá editar categorías.
Precondiciones	El usuario debe de estar logueado como administrador.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de categorías 2. El sistema muestra un listado con las categorías de la plataforma. 3. El usuario selecciona la categoría a editar 4. El sistema muestra los datos que se pueden editar. 5. El usuario rellena los datos. 6. El sistema verifica los datos introducidos. 7. El sistema edita la categoría.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Los datos introducidos son erróneos: el sistema muestra un mensaje informativo y vuelve al paso 4. • Ocurre un error al crear la categoría: el sistema muestra un mensaje informativo y vuelve al paso 4.
Frecuencia	Media
Importancia	Media
Prioridad	Media
Observaciones	

Tabla 40 - Caso de uso "Editar categoría" (CU-22)

CU-23 Eliminar categoría	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-22 Un usuario podrá eliminar categorías
Actor	Usuario administrador
Descripción	Un usuario administrador podrá eliminar categorías del sistema.
Precondiciones	El usuario debe de estar logueado como administrador
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página del listado de categorías 2. El sistema muestra un listado con las categorías de la plataforma. 3. El usuario selecciona la categoría a eliminar 4. El sistema pregunta si realmente quiere realizar dicha acción. 5. El usuario hace click en el botón aceptar. 6. El sistema elimina la categoría.
Postcondiciones	N/A
Excepciones	<ul style="list-style-type: none"> • Ocurre un error al eliminar la categoría: el sistema muestra un mensaje informativo y vuelve al paso 2
Frecuencia	Baja
Importancia	Baja
Prioridad	Baja
Observaciones	

Tabla 41 - Caso de uso "Eliminar categoría" (CU-23)

CU-24 Notificar a los usuario	
Versión	1.0
Autor	Yaser El Dabete Arribas
Requisitos asociados	RU-24 Notificar a los usuarios
Actor	Sendinblue
Descripción	Sendinblue puede notificar vía email a los usuarios.
Precondiciones	<ul style="list-style-type: none"> • Un usuario se registre en el sistema. • Un usuario realice una publicación. • Un administrador elimine una cuenta de usuario. • Un administrador apruebe/desapruebe una publicación.
Flujo normal	<ol style="list-style-type: none"> 1. Un usuario realiza cualquiera de las 4 acciones citadas anteriormente. 2. Un usuario materializa la acción haciendo click en el botón referente a la acción. 3. El sistema procesa dicha acción. 4. El sistema realiza una petición a Sendinblue con los datos necesarios para enviar el email. 5. Sendinblue notifica al usuario recibido en el cuerpo de la petición anteriormente citada.
Postcondiciones	N/A
Excepciones	<ol style="list-style-type: none"> 1. Sendinblue se encuentra sobrecargado, la petición se encolará y se resolverá cuando sea posible. 2. Sendinblue se ha caído, se mostrará un mensaje de error informando de la situación.
Frecuencia	Alta: las acciones que requieren una notificación son mínimas pero entre ellas se encuentra la publicación de documentos, pilar del sistema.
Importancia	Alta
Prioridad	Media
Observaciones	N/A

Tabla 42 - Caso de uso "Notificar a los usuarios" (CU-24)

4.4 Requisitos de información

A continuación se detallan los requisitos de información del sistema, podemos definir requisitos de información como el conjunto de datos con sus especificaciones necesarios para que el sistema funcione correctamente.

Requisito	Descripción
RI-01	El sistema almacenará la información del usuario cuando realice el registro.
RI-02	El sistema almacenará la información de los documentos publicados.
RI-03	El sistema almacenará la información de las categorías creadas.
RI04	El sistema almacenará las puntuaciones de documentos y usuarios.
RI-05	El sistema permitirá guardar un registro de eventos, errores y acciones de los usuarios.

Tabla 43 - Requisitos de información

4.4.1 Diagrama entidad-relación

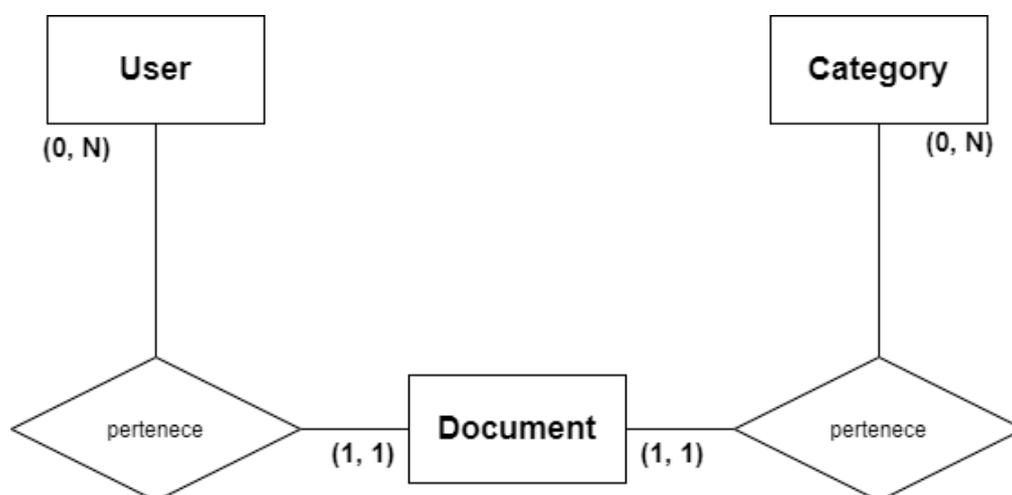


Figura 20 - Diagrama de entidad relación

Pese a la complejidad de la aplicación, el diagrama de documentos, se reduce a tan solo tres entidades. La primera entidad que nos encontramos es la de “User”, no es la entidad más importante del sistema pero si es crítica ya que almacena la información del registro de usuarios.

Si seguimos hacia la derecha, tenemos la entidad “Category” en la que se recoge la información relacionada con cada una de las categorías existentes en el sistema.

Por último tenemos la entidad “Document” cuya importancia es crítica en el sistema ya que es la encargada de almacenar la información de las publicaciones existentes en la aplicación. Como se puede observar la entidad “Document” se relaciona tanto con “User” como con “Category” ya que es necesario saber en todo momento quién es el autor de una publicación y en qué categoría está asignada dicha publicación.

4.4.1 Diccionario de datos

Las tablas que se ven a continuación, hacen referencia a cada uno de las entidades de la base de datos.

Atributo	Tipo	Único (unique)	Puede ser nulo (required)
_id	String	SI	NO
name	String	NO	NO
description	String	NO	SI
email	String	SI	NO
password	String	NO	NO
reputation	Number	NO	SI
numRates	Number	NO	SI
contributions	Number	NO	SI
admin	Boolean	NO	NO
signupDate	Date	NO	NO
image	String	SI	SI

Tabla 44 - Diccionario de datos de "Users"

Atributo	Tipo	Único (unique)	Puede ser nulo (required)
_id	String	SI	NO
name	String	SI	NO
description	String	NO	NO
keyWords	Array<String>	NO	NO
image	String	NO	SI

Tabla 45 - Diccionario de datos de "Categories"

Atributo	Tipo	Único (unique)	Puede ser nulo (required)
_id	String	SI	NO
title	String	NO	NO
description	String	NO	SI
topic	String	NO	NO
autoApproval	Boolean	NO	SI
manuApproval	Boolean	NO	SI
approved	Boolean	NO	SI
file	String	NO	NO
uploadDate	Date	NO	NO
rate	Number	NO	SI
validationRate	Number	NO	SI
numRates	Number	NO	SI
Status	String	NO	SI
idAuthor	String	NO	NO
idCategory	String	NO	NO

Tabla 46 - Diccionario de datos de "Documents"

4.5 Requisitos no funcionales

Se puede definir requisitos no funcionales como aquellas restricciones del sistema que afectan a su comportamiento o desarrollo. A continuación, se muestran los grupos más habituales con su correspondiente función en el sistema.

Seguridad

- **RNF-01:** Los datos de configuración sensibles se guardan en un fichero de variables de entorno.
- **RNF-02:** Las contraseñas deberán tener un mínimo de 8 caracteres incluyendo números, letras mayúsculas o minúsculas y además se almacenarán cifradas.
- **RNF-03:** Tanto la aplicación web como la móvil implementan el protocolo de seguridad SSL.

Usabilidad

- **RNF-04:** El sistema deberá ser fácil de usar.
- **RNF-05:** El sistema proporcionará mensajes de error claros.

Escalabilidad

- **RNF-06:** El producto final estará dividido en módulos independientes.
- **RNF-07:** La lógica de negocio esta encapsulada en componentes que interactúan entre sí.
- **RNF-08:** El sistema deberá ser escalable y soportar la concurrencia de usuarios.
- **RNF-09:** El sistema será desplegado en un servidor Node para facilitar su escalabilidad.

Disponibilidad

- **RNF-10:** El sistema deberá ser capaz de tener una disponibilidad de 24 horas 7 días a la semana.

Capítulo 5 – Diseño e implementación

5.1 Arquitectura lógica

Podemos definir arquitectura lógica como la organización de los componentes dentro de una aplicación especificando las relaciones que existen entre ellos. En este caso particular se detallarán la arquitectura lógica tanto de la aplicación web como de la móvil.

Como punto común entre ambas aplicaciones, podemos decir que están diseñadas bajo el patrón Modelo-Vista-Controlador (MVC) y que constan de dos partes claramente diferenciadas, un *frontend*, que consta de la aplicación móvil y la web que realizan la función de cliente y un *backend* alojado en un servidor Node que se encarga de procesar las peticiones de los clientes y almacenar la información.

Aplicación web

Partiendo de que la aplicación consta de dos partes claramente diferenciadas, como se ha expuesto anteriormente, se procede a describir estas partes:

- **Ciente web o *frontend*:** es la primera parte del sistema, la capa de presentación, es decir, con la que el usuario interactúa directamente a través de un navegador. Esta capa se encuentra desarrollada en Angular que básicamente es HTML, CSS y TypeScript, que es una especie de JavaScript mejorado y con una serie de funcionalidades destinadas a facilitar el desarrollo.
- **Capa de negocio o *backend*:** Esta es la capa dónde se encuentra la lógica de negocio y se encarga de mantener la comunicación entre el cliente web y la base de datos.
- **Capa de datos:** Es la capa encargada de hacer persistir los datos una vez estos hayan procesado. Cabe resaltar que estos datos son almacenados en objetos *BSON*, que es una extensión binaria de objetos *JSON*.

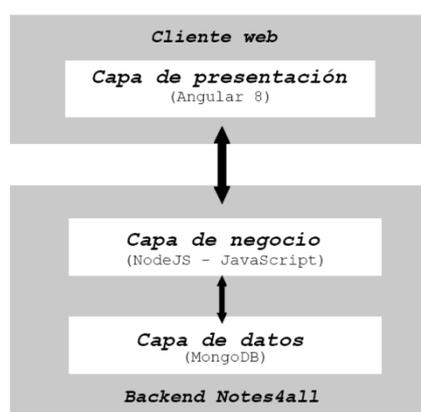


Figura 21 - Arquitectura lógica de la aplicación web

Aplicación móvil

Para hablar de la arquitectura lógica de la aplicación móvil, también se debe tener en cuenta la clara diferenciación en dos grandes partes del sistema. A continuación se detalla la arquitectura de la aplicación móvil:

- En este caso, el lado cliente lo compone únicamente el dispositivo móvil del usuario, el cual deberá tener la aplicación instalada. Una vez inicie la aplicación, se visualizará la capa de presentación, formada por un conjunto de ficheros XML (*Extensible Markup Languaje*) generados a partir de las capas de negocio y datos. Además la aplicación móvil cuenta con otras tres capas que llamaremos: Capa de Seguridad, Capa de configuración y Capa de Control de Errores, las cuales existen para asegurar el correcto funcionamiento de la aplicación.
- En el lado del servidor se encuentra nuestro el servidor NodeJs con las capas de negocio y datos. Funciona como una API Rest haciendo uso del protocolo HTTP y utiliza la comunicación para enviar datos *JSON* que darán forma a la aplicación. La API Rest será la encargada de controlar las peticiones realizadas por los diferentes usuarios.

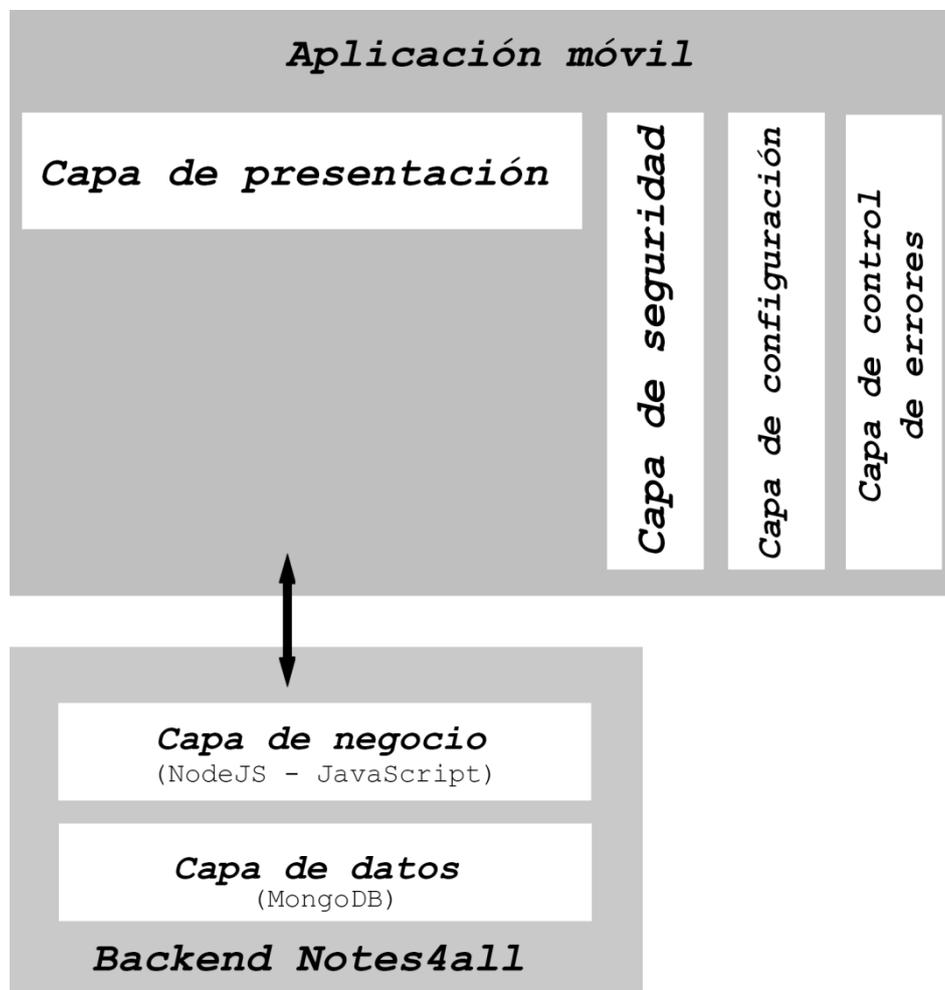


Figura 22 - Arquitectura lógica de la aplicación móvil

5.2 Arquitectura física

Podemos definir la arquitectura física como la encargada de representar la adecuación de todos los componentes de carácter físico que intervienen en un sistema para su correcto funcionamiento y despliegue.

En este caso concreto, tenemos por un lado el cliente o *frontend*, que es el encargado de mostrar al usuario la información y procesar la información que introduce el usuario para enviarla al servidor. Este lado cliente se divide en dos partes, una es el navegador web que utilizará la plataforma web y otro es el dispositivo móvil del usuario que se encargará de utilizar la aplicación móvil.

Por otro lado, se encuentra el servidor o *backend*, es decir, el servidor NodeJS que además de ser la unión de las dos plataformas anteriormente descritas, también será el encargado de procesar las peticiones que lleguen y del almacenamiento de los datos vitales para el correcto funcionamiento del sistema global.

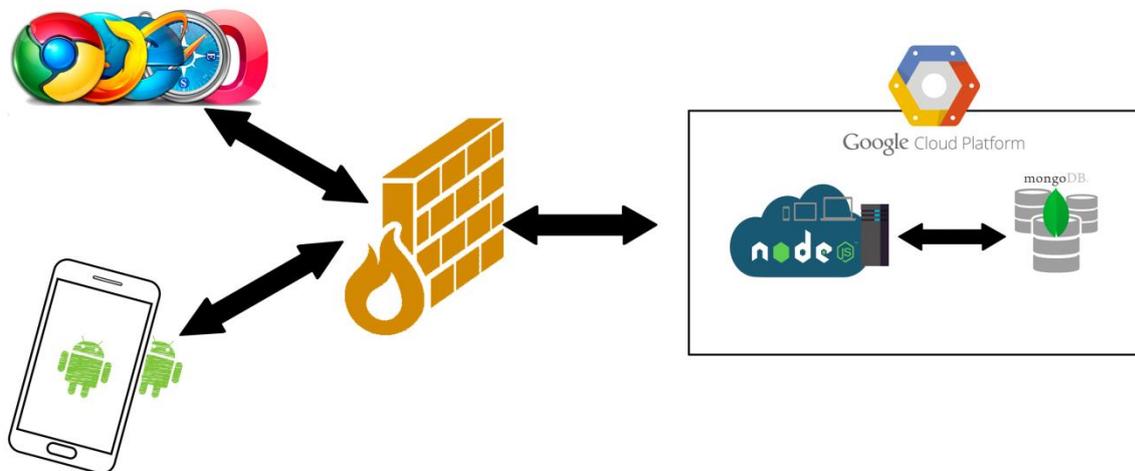


Figura 23 - Arquitectura física del sistema

5.3 Diagrama de clases

Si observamos la figura siguiente, se puede apreciar el diagrama de clases en el que se basa la aplicación móvil, en él nos encontramos con las entidades alojadas en la aplicación con sus atributos y métodos necesarios para el correcto funcionamiento de la misma.



Figura 24 - Diagrama de clases de la aplicación móvil

Ahora se muestra el diagrama de clases de la aplicación web. Como se puede apreciar en la figura, consta de una entidad más y un mayor número de métodos en las otras dos entidades ya que la aplicación web alberga una mayor cantidad de funcionalidades, entre ellas, todas las que desempeña el usuario administrador que sobre todo tienen que ver con la gestión de las categorías de los documentos.

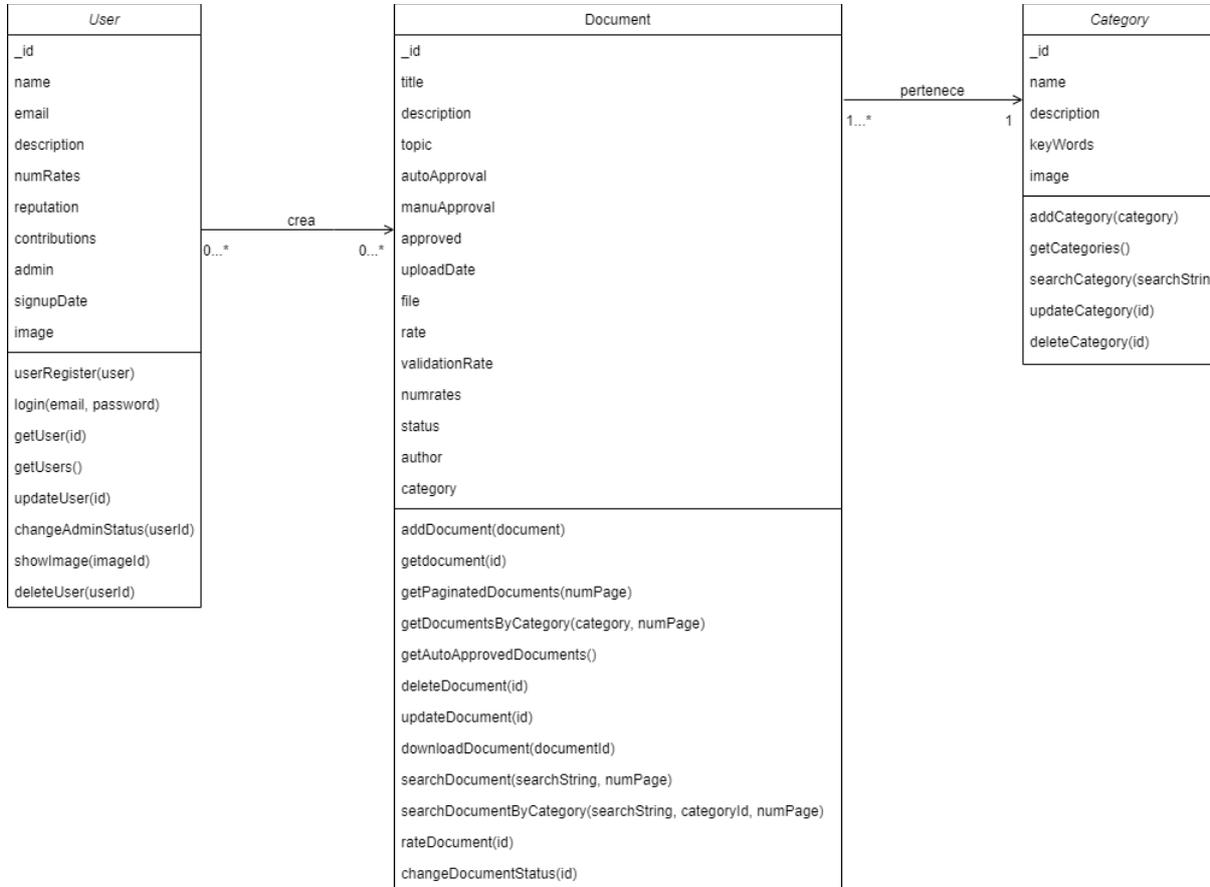


Figura 25 - Diagrama de clases de la aplicación web

5.4 Modelo lógico de la base de datos

Notes4All utiliza por defecto el motor de bases de datos MongoDB, pero nunca llega a emplear lógica a tan bajo nivel ya que se introduce un agente intermedio llamado *mongoose*. Podemos definir este agente como un modelo de programación que permite definir estructuras de una base de datos sobre una estructura lógica de entidades (ORM, *Object Relational Mapping*). Con este agente intermedio se facilita la actuación entre el sistema y la base de datos ya que nos provee de una serie de funcionalidad con la que interactuar con la base de datos, por lo que en la mayoría de casos, no es necesario realizar las consultas específicas de Mongo.

Según lo expuesto en el párrafo anterior, podemos decir entonces que **Notes4All** es agnóstico respecto al sistema de bases de datos ya que gracias a esta capa intermedia, la aplicación desconoce la base de datos que se encuentra por debajo.



Figura 26 - Base de datos MongoDB

Como se puede observar en la figura anterior, los datos se almacenan en documentos que tienen formato *JSON*. También se puede ver la relación entre los documentos, en este caso cada publicación (*Documents*) tiene un autor que se corresponde con el identificador del usuario que ha creado dicho documento y una categoría que representa el identificador de la categoría a la que se ha asignado la publicación.

5.5 Implementación

Anteriormente se han representado las entidades que componen el sistema con sus atributos y sus métodos. Como hemos podido observar, la mayoría de funciones del sistema, se emplean para realizar consultas, solo las funciones que tienen que ver con la autenticación (registro e inicio de sesión), las de edición o creación de categorías, la edición del perfil y la puntuación de documentos, son funciones que insertan datos en el sistema.

Una de las funcionalidades que inserta datos en el sistema y que es necesario destacar es la de realizar una publicación en el sistema ya que en ella está involucrado el validador de documentos. Dicho validador analiza el documento que se ha subido al sistema, más concretamente, elimina las palabras más comunes del lenguaje como son las preposiciones, artículos, pronombres, etc. y después analiza las palabras restantes y las contrasta con una lista de palabras clave asociadas a cada categoría, obteniendo el porcentaje de apariciones de las mismas. Según el valor obtenido calificará el documento como:

- Aprobado si el valor del porcentaje es mayor o igual a 60.
- Cuestionable si el valor del porcentaje se encuentra entre 40 y 60.
- Rechazado si el valor del porcentaje se encuentra por debajo de 40.

Además, el validador es capaz de mejorar su análisis ya que una vez ha limpiado el documento de palabras comunes, obtiene e inserta las palabras más repetidas en el listado de palabras clave de la categoría correspondiente.

La validación de documentos es una tarea que lleva cierto tiempo realizar, es decir, es una función asíncrona por lo que no se puede obtener una respuesta instantánea, es por esto que cuando un usuario realiza una publicación, recibe una respuesta genérica (de error o éxito) para que pueda seguir navegando por la aplicación y su experiencia no sea negativa esperando que termine la validación. Una vez la validación haya terminado, el sistema enviará una notificación al usuario con el resultado de dicha validación.

Como se ha podido ver en los puntos anteriores, un documento puede ser identificado de dos formas, una de ellas es por el autor que lo ha publicado y otra es a través de la categoría a la que pertenece ya que un documento siempre va a tener una categoría asociada y un usuario autor de la publicación.

5.6 Diseño de la interfaz de usuario

A continuación se muestran los *mocks* o diseños de cada pantalla realizados previamente a la construcción de las aplicaciones que conforman el sistema

5.6.1 Aplicación móvil

Inicio de sesión

Se trata de la pantalla inicial de la aplicación. En ella aparece el formulario con los datos necesarios para la autenticación del usuario y un *link* a la pantalla de registro por si el usuario no tuviese creada una cuenta en el sistema.



Figura 27 - Diseño de la pantalla "Inicio de sesión"

Registro:

Se trata de la pantalla a través de la cual un usuario es capaz de crear una cuenta en el sistema. En ella se muestra el formulario con los datos necesarios para realizar dicho registro y además un *link* que lleva a la pantalla de inicio de sesión.



Figura 28 - Diseño de la pantalla "Registro"

Inicio:

Es la pantalla a la que se accede una vez que el usuario ha iniciado sesión en el sistema. En ella se muestra el listado de documentos publicados y aprobados en la plataforma con un pequeño resumen de cada uno de ellos.



Figura 29 . Diseño de la pantalla “Inicio”

Menú:

Una vez que el usuario esté autenticado, podrá visualizar el menú donde dispondrá de las funciones principales que puede realizar en la aplicación y podrá acceder a cada una de ellas.

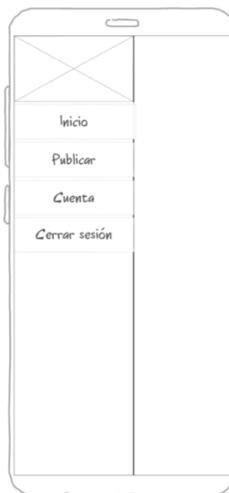


Figura 30 - Diseño del menú de la aplicación

Publicar:

Es la pantalla a través de la cual un usuario puede realizar una publicación en el sistema. En ella se muestran los campos obligatorios para poder realizar dicha publicación.



The image shows a mobile application wireframe for the 'Publicar' (Publish) screen. It features a vertical list of input fields: 'Título' (Title), 'Tema' (Topic), 'Descripción' (Description), and 'Categoría' (Category). Below these fields is a button labeled 'Selecciona el documento' (Select the document). At the bottom of the screen is a large button labeled 'Publicar' (Publish).

Figura 31 - Diseño de la pantalla "Publicar"

Cuenta:

Es la pantalla en la que un usuario puede ver los datos relevantes de su cuenta como por ejemplo, las publicaciones que ha realizado o la puntuación que tiene en el sistema.



The image shows a mobile application wireframe for the 'Cuenta' (Account) screen. At the top is a square placeholder for a profile picture with an 'X' inside. Below this is a section titled 'Información personal' (Personal information) containing three fields: 'Nombre: xxxx' (Name), 'Email: xxxx@xxxx.xx' (Email), and 'Descripción: xxxxxx' (Description). Below this is another section titled 'Información de la cuenta' (Account information) containing two fields: 'Publicaciones: X' (Publications) and 'Puntuación: X' (Score).

Figura 32 - Diseño de la pantalla "Cuenta"

Cerrar sesión:

Pop up o alerta que se muestra cuando el usuario selecciona la opción de cerrar sesión en el menú. Sirve para confirmar si se quiere cerrar sesión o no.



Figura 33 - Diseño del pop up "Cerrar sesión"

5.6.2 Aplicación web

El diseño de la aplicación está optimizado para ser utilizada en un navegador de un monitor de unas 22 pulgadas. Sin embargo, el diseño se ha realizado con una base *responsive* o adaptativa, lo que quiere decir que la aplicación es totalmente funcional en otro tipo de dispositivos con diferentes tamaños o resoluciones de pantalla.

A continuación se van a detallar las interfaces de las secciones más importantes de la aplicación ya que la mayoría tienen una base común; por ejemplo los formularios de creación o edición tienen la misma estructura y solo cambian los campos editables dependiendo de cada modelo, por ello para este tipo de interfaces se va a utilizar una plantilla genérica.

Inicio de sesión

Es la pantalla que permite autenticarse en la web, y en ella aparece el formulario con los datos de acceso.

A screenshot of a web browser window showing a login form. The browser's address bar is empty. The form is centered on the page and contains the following elements: a label 'Email' followed by a text input field with the placeholder text 'Introduzca su email'; a label 'Password' followed by a text input field with the placeholder text 'Introduzca su contraseña'; a line of text '(*) Campos requeridos'; and a 'Iniciar sesión' button at the bottom.

Figura 34 - Diseño de la pantalla "Inicio de sesión"

Registro

Es la pantalla que permite a un usuario crear una cuenta en el sistema. En ella aparece el formulario con los datos necesarios para la creación de dicha cuenta.

A screenshot of a web browser window showing a registration form. The browser's address bar is empty. The form is centered on the page and contains the following elements: a label 'Email' followed by a text input field with the placeholder text 'Introduzca su email'; a label 'Password' followed by a text input field with the placeholder text 'Introduzca su contraseña'; a line of text '(*) Campos requeridos'; and a 'Iniciar sesión' button at the bottom.

Figura 35 - Diseño de la pantalla "Registro"

Inicio

Es la pantalla principal de la web, la que se carga al iniciar la aplicación. En ella se muestra el listado paginado de todas las publicaciones aprobadas que existen en el sistema.

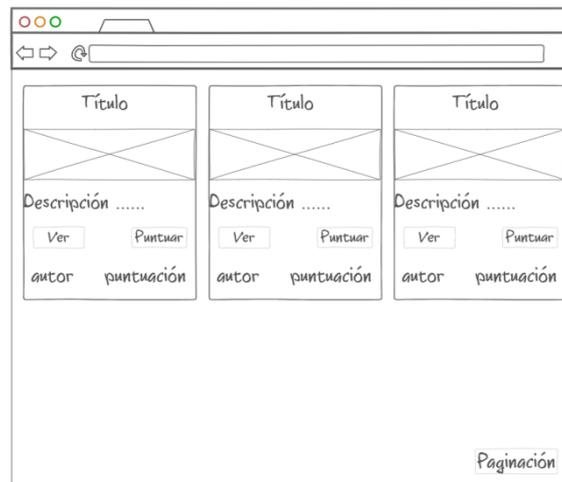


Figura 36 - Diseño de la pantalla "Inicio"

Buscar

Es la pantalla en la que se permite buscar publicaciones o bien, filtrando por el título del documento, o bien, haciéndolo por la categoría a la que pertenece.

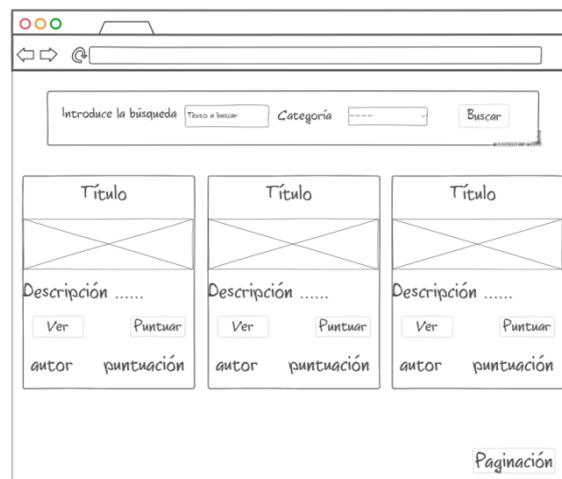


Figura 37- Diseño de la pantalla "Buscar"

Publicar

Es la pantalla en la que se le permite a un usuario autenticado realizar publicaciones en el sistema. En ella aparece un formulario con los datos necesarios para llevar a cabo dicha publicación.

Selección el documento a publicar

Título -
Introduce el título

Descripción -
Introduce la descripción

Tema -

Categoría -
Selecciona una categoría

(*) Campos requeridos

Publicar

Figura 38 - Diseño de la pantalla "Publicar"

Mi perfil

Es la pantalla correspondiente a los datos del usuario, en ella podrá ver sus datos como por ejemplo la puntuación que tiene o las publicaciones que ha realizado en el sistema y además un listado de los documentos publicados con sus datos más relevantes.

Nombre
Tienes un total de X puntos
Tienes un total de X publicaciones
Biografía

Editar Eliminar

Introduce la búsqueda

Título	Descripción	Categoría	Estado	Puntuación	Acciones

Figura 39 - Diseño de la pantalla "Mi perfil"

A continuación se muestra el diseño de las secciones únicamente accesibles para los usuarios de tipo administrador.

Usuarios

Es la pantalla en la que se muestra el listado completo de los usuarios registrados en el sistema con sus datos más relevantes. En ella se podrá eliminar un usuario o darle permisos de administración.

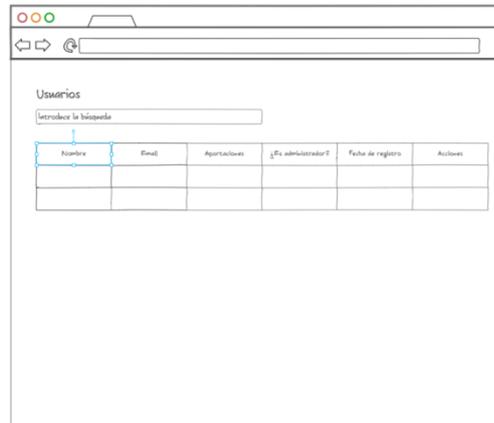


Figura 40 - Diseño de la pantalla "Usuarios"

Categorías

Pantalla en la que se muestra el listado completo de las categorías existentes en el sistema con sus datos más importantes. En ella se podrá dar de alta una nueva categoría, editar y eliminar categorías.



Figura 41 - Diseño de la pantalla "Categorías"

Documentos

Es la pantalla en la que se muestra el listado completo de los documentos publicados en el sistema con sus datos más importantes y diferenciados por los documentos aprobados por el validador y los documentos que necesitan una revisión manual. En ella se podrá aprobar o desaprobar un documento, visualizarlo o eliminarlo.

Documentos a revisar

Introduce la búsqueda

Título	Categoría	¿Aprobado?	Autor	Puntuación del sistema	Acciones

Documentos auto-aprobados

Introduce la búsqueda

Título	Categoría	¿Aprobado?	Autor	Puntuación del sistema	Acciones

Figura 42 - Diseño de la pantalla "Documentos"

Por último vamos a ver el diseño genérico del formulario que se emplea para crear o editar ciertos datos como por ejemplo las categorías, un documento o el perfil del usuario.

Crear / Editar

Es el formulario utilizado para crear o editar los campos de los diferentes modelos, la estructura es la misma en todos y lo único que cambia son los campos que se pueden editar.

Añadir/Editar

Campo 1

Introduce el campo 1

Campo 2

Introduce el campo 2

Campo 3

Introduce el campo 3

Campo 4

Introduce el campo 4

Añadir/Editar

Figura 43 - Diseño del formulario de creación o edición de datos

Capítulo 6 – Pruebas

En este capítulo se muestran las pruebas que se han realizado al sistema para asegurar de que tiene un funcionamiento correcto en el caso de que se produzca un comportamiento o una respuesta no esperada. Para ello se van a realizar dos tipos de pruebas.

6.1 Pruebas de caja blanca

Las pruebas de caja blanca o pruebas estructurales, son aquellas que se basan en los detalles procedimentales del sistema y por tanto están muy ligadas al código desarrollado. Para llevar a cabo este tipo de pruebas, se va a probar que la ejecución de cada una de las funcionalidades de las aplicaciones funcionan de forma correcta, es decir, que tengan una respuesta correcta ante una petición concreta. Este tipo de pruebas se realizan durante la implementación del sistema y se han realizado para las dos aplicaciones:

- Validación de la entrada de datos de todos los formularios.
- Validación de comunicación con la base de datos.
- Control de sesiones.
- Validación del funcionamiento de las operaciones.
- Validación de la disposición de información en tiempo real.
- Control ante un posible error de conexión entre el cliente y el servidor.
- Control de las respuestas del servidor.

6.2 Pruebas de caja negra

Este tipo de pruebas son aquellas en las que no importar el código, ni el procesamiento interno de los datos, por lo que se emplean unos datos de entrada y se comprueba que los datos de salida son los esperados. En resumen podemos decir que las pruebas de caja negra son puramente funcionales y se dedican a “mirar” el exterior del sistema que se prueba. A continuación se detallan algunas de las pruebas que se han realizado en los dos sistemas:

PCN-01 Registro de usuario	
Objetivo	Comprobar que se puede crear un usuario común en el sistema..
Precondiciones	N/A
Datos de entrada	Email: yaser.94ae@gmail.com Contraseña: yaser94 Nombre: Yaser El Dabete Arribas
Acción esperada	Al pulsar “Registro” el sistema notificará la creación del usuario y redirigirá a la pantalla de inicio de sesión.
Resultado	Correcto

Tabla 47 - PCN-01 Registro de usuario

PCN-02 Iniciar sesión	
Objetivo	Comprobar que un usuario puede iniciar sesión en la aplicación.
Precondiciones	El usuario debe de estar registrado en el sistema.
Datos de entrada	Email: yaser.94ae@gmail.com Contraseña: yaser94
Acción esperada	Al pulsar “Iniciar sesión” el sistema notificará el inicio de sesión y redirigirá al usuario a la pantalla de inicio.
Resultado	Correcto

Tabla 48 - PCN-02 Inicio de sesión

PCN-03 Inicio	
Objetivo	Comprobar que el usuario puede visualizar todos los documentos disponibles en el sistema.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón “Inicio” del menú, el sistema redirigirá al usuario a la sección de inicio donde podrá consultar todos los documentos aprobados del sistema.
Resultado	Correcto

Tabla 49 - PCN-03 Inicio

PCN-04 Mi perfil	
Objetivo	Comprobar que el usuario puede ver los datos de su cuenta.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación.
Datos de entrada	N/A
Acción esperada	Al pulsar en el botón “Cuenta” del menú, el sistema redirigirá al usuario a la sección de su perfil dónde podrá consultar sus datos.
Resultado	Correcto

Tabla 50 - PCN-04 Mi perfil

PCN-05 Publicar	
Objetivo	Comprobar que el usuario puede realizar una publicación.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación.
Datos de entrada	Documento: prueba.pdf Título: prueba Tema: prueba Categoría: prueba Descripción: prueba
Acción esperada	Al pulsar el botón de publicar, el sistema notificará al usuario de dicha publicación y le redirigirá a la pantalla de inicio.
Resultado	Correcto

Tabla 51 - PCN-05 Publicar

PCN-06 Puntuar un documento	
Objetivo	Comprobar que el usuario puede puntuar documentos.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación.
Datos de entrada	Documento: prueba.pdf Puntuación: 5
Acción esperada	Al pulsar el botón correspondiente con la puntuación seleccionada, el sistema notificará al usuario del resultado de la acción y redirigirá a la pantalla de inicio con los documentos actualizados.
Resultado	Correcto

Tabla 52 - PCN-06 Puntuar un documento

PCN-07 Crear / editar una categoría	
Objetivo	Comprobar que el usuario administrador puede crear o editar una categoría.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación y ser administrador.
Datos de entrada	Nombre: Informática Descripción: Categoría que engloba los documentos relacionados con la informática Palabras clave: informática, programación, Angular, NodeJs, Mongo...
Acción esperada	Al pulsar el botón de “Crear” o “Editar”, el sistema creará o editará la categoría y aparecerá en el listado.
Resultado	Correcto

Tabla 53 - PCN-07 Crear o editar categorías

PCN-08 Aprobar / desaprobar un documento	
Objetivo	Comprobar que el usuario administrador puede aprobar o desaprobar un documento.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de aprobar o desaprobar un documento, el sistema modificará el estado de dicho documento.
Resultado	Correcto

Tabla 54 - PCN-08 Aprobar o desaprobar un documento

PCN-09 Listado de usuarios	
Objetivo	Comprobar que el administrador puede visualizar el listado de los usuarios registrados en el sistema.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al acceder a la pantalla de gestión de usuarios, se mostrará una tabla con el listado de los usuarios registrados en el sistema y la posibilidad de eliminar un usuario o dar permisos de administrador.
Resultado	Correcto

Tabla 55 - PCN-09 Listado de usuarios

PCN-10 Listado de categorías	
Objetivo	Comprobar que el administrador puede visualizar el listado de las categorías del sistema.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al acceder a la pantalla de gestión de categorías, se mostrará una tabla con el listado de las categorías registradas en el sistema y la posibilidad de eliminarlas o editarlas.
Resultado	Correcto

Tabla 56 - PCN-10 Listado de categorías

PCN-11 Listado de documentos	
Objetivo	Comprobar que el administrador puede visualizar el listado de los documentos del sistema.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al acceder a la pantalla de gestión de documentos, se mostrarán dos tablas, una con el listado de documentos auto-aprobados por el validador y otra de los documentos a revisar. Podrá eliminar documentos, visualizarlo y aprobarlos o desaprobados.
Resultado	Correcto

Tabla 57 – PCN-11 Listado de documentos

PCN-12 Eliminar un usuario	
Objetivo	Comprobar que un usuario administrador puede eliminar a otro usuario registrado en el sistema
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de eliminar usuario, se eliminará del sistema y dicho usuario no podrá acceder a la plataforma.
Resultado	Correcto

Tabla 58 - PCN-12 Eliminar un usuario

PCN-13 Dar permisos de administración	
Objetivo	Comprobar que un usuario administrador puede dar permisos de administración a otro usuario.
Precondiciones	El usuario debe de haber iniciado sesión en la aplicación como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de dar permisos de administración, el usuario que los recibe podrá administrar el sistema.
Resultado	Correcto

Tabla 59 - PCN-13 Dar permisos de administración

PCN-15 Buscar documentos	
Objetivo	Comprobar que un usuario puede buscar documentos.
Precondiciones	El usuario debe de haber iniciado sesión en el sistema.
Datos de entrada	Existen tres opciones: <ul style="list-style-type: none"> • El usuario introduce un texto a buscar, por ejemplo: “Programación web”, • El usuario selecciona una de las categorías disponibles en el sistema, por ejemplo “Informática”. • El usuario introduce un texto a buscar y selecciona una de las categorías disponibles, por ejemplo: “Programación web” e “Informática”
Acción esperada	Al pulsar el botón “Buscar” aparecerán los resultados obtenidos para la búsqueda solicitada.
Resultado	Correcto

Tabla 60 - PCN-15 Buscar documentos

PCN-16 Detalles de usuario	
Objetivo	Comprobar que el usuario administrador puede ver el detalle de cada usuario.
Precondiciones	El usuario debe de haber iniciado sesión en el sistema como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón “Detalles”, se muestran los datos más importantes del usuario.
Resultado	Correcto

Tabla 61 - PCN-16 Detalles de usuario

PCN-17 Eliminar un documento	
Objetivo	Comprobar que el usuario administrador puede eliminar uno de los documentos registrados en el sistema.
Precondiciones	El usuario debe de haber iniciado sesión en el sistema como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón “Eliminar” el documento dejará de estar disponible en el sistema.
Resultado	Correcto

Tabla 62 - PCN-17 Eliminar documentos

PCN-18 Eliminar categorías	
Objetivo	Comprobar que el usuario administrador puede eliminar categorías del sistema.
Precondiciones	El usuario debe de haber iniciado sesión en el sistema como administrador.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón “Eliminar” la categoría y los documentos asociados a ella dejarán de estar disponibles en el sistema.
Resultado	Correcto

Tabla 63 - PCN-18 Eliminar categorías

PCN-19 Eliminar cuenta	
Objetivo	Comprobar que un usuario puede eliminar su cuenta.
Precondiciones	El usuario debe de haber iniciado sesión en el sistema.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón “Eliminar” la cuenta se dará de baja y el usuario no podrá acceder al sistema.
Resultado	Correcto

Tabla 64 - PCN-19 Eliminar cuenta

Capítulo 7 – Documentación del producto

En este capítulo se mostrarán tres manuales con el objetivo de orientar a los posibles administradores del sistema o usuarios cómo poder montar el sistema y/o utilizarlo de forma correcta.

7.1 Manual de despliegue

Para poder desplegar **Notes4all** en un servidor productivo es necesario seguir cuidadosamente las siguientes indicaciones:

- Instalar/actualizar *NodeJs* y *npm*, en concreto serán necesarias las versiones 12.16.1 y 6.13.4 respectivamente. Para realizar la instalación se ejecutan los siguientes comandos:

```
1 curl -sL https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh -o install_nvm.sh
2 bash install_nvm.sh
3 source ~/.profile
4 nvm install 12.16.1
```

Figura 44 - Instalación NodeJs y npm

- Descargar los repositorios, para ello ejecutaremos:

```
1 git clone https://github.com/vasmanets/notes4all_front.git
2 git clone https://github.com/vasmanets/notes4all_back.git
3
```

Figura 45 - Instalación de los repositorios

- Instalar Angular en su versión 8.3.21, para ello ejecutaremos el siguiente comando:

```
1
2 sudo npm i -g @angular/cli@8.3.21
3
```

Figura 46 - Instalación de Angular 8.3.21

- Instalar el sistema de gestión de bases de datos, en este caso MongoDB, para ello se ejecutarán los siguientes comandos:

```
2 sudo apt install -y mongodb
3 sudo systemctl status mongodb
```

Figura 47- Instalación de MongoDB

- Instalar los certificados SSL y configurar NGINX para su correcto funcionamiento, para ello se ejecutarán los siguientes comandos:

```

2  sudo add-apt-repository ppa:certbot/certbot
3  sudo apt install python-certbot-nginx
4  sudo nano /etc/nginx/sites-available/notes4all.tk
5  .....
   server_name notes4all.tk www.notes4all.tk;
6  sudo nginx -t
7  sudo systemctl reload nginx
8  sudo certbot --nginx -d notes4all.tk -d www.notes4all.tk
9

```

Figura 48 - Instalación de los certificados SSL

- Instalar *PM2* e iniciar el servidor web, para ello ejecutaremos los siguientes comandos:

```

2  sudo npm install pm2@latest -g
3  cd
4  pm2 start notes4all_back/index.js -name server -l /home/ubuntu/logs/server.log
5

```

Figura 49 - Instalación de PM2 e inicio del servidor

- Crear el fichero que contiene las variables de entorno en la raíz de la carpeta que se ha generado al clonar el repositorio. Dicho fichero debe llamarse **.env** y debe tener las siguientes líneas:

```

NODE_ENV=production
PORT=4201
FRONT_PORT=4200
TOKENSECRET=**treesala**pApY2019$
MONGO=mongodb://localhost:27017/notes4all
TOKENSECRET=**secret$notes4all**
EMAIL_API_KEY=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
MONGO_SERVER=localhost:27017
MONGO_USER=notes4all
MONGO_PWE=n0tes4all2020$
MONGO_DB=notes4all

```

Figura 50 - Fichero de variables de entorno

Por último se ha de crear un fichero llamado **deploy.sh** y deberemos darle permisos de ejecución, esto lo podemos hacer con el siguiente comando:

```

1
2  sudo chmod +x deploy.sh
3

```

Figura 51 - Permisos de ejecución

Dicho fichero es el encargado de ejecutar todos los procesos necesarios para poder realizar el despliegue de la aplicación, descargará posibles *commits* realizados a los repositorios, actualizará las librerías a través del fichero *package.json*, realizará la construcción del código alojado en el repositorio del *frontend* y finalmente relanzará el servidor web con todos los cambios que se hayan realizado. El contenido del fichero de despliegue es el siguiente:

```
GNU nano 2.9.3 deploy.sh

#!/bin/bash

cd /home/ubuntu/notes4all_back

git checkout -f

mkdir -p uploads/images
mkdir -p uploads/documents

git pull origin develop
npm ci

cd /home/ubuntu/notes4all_front
git checkout -f
git pull origin develop
npm ci

ng build --prod

mkdir -p /home/ubuntu/notes4all_back/notes4all
cp -R dist/notes4all/* /home/ubuntu/notes4all_back/notes4all/

mkdir -p /home/ubuntu/logs
pm2 restart server
```

Figura 52 - Contenido del fichero deploy.sh

7.2 Manual de usuario

Con este manual se pretende orientar al usuario en el uso tanto de la aplicación web como de la móvil. Para iniciar la aplicación el usuario tendrá que entrar en la url <https://www.notes4all.tk> o bien iniciar la aplicación desde su dispositivo Android.

7.2.1 Aplicación web

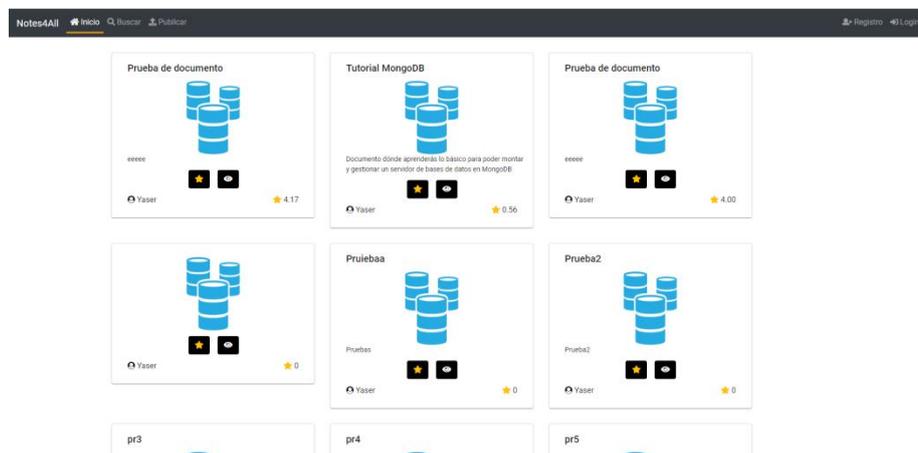


Figura 53 - Pantalla de inicio de la aplicación web

Nada más entrar en la web, el usuario se encontrará en la página de inicio donde podrá visualizar todos los documentos paginados registrados en el sistema. Si desea realizar

más acciones en el sistema deberá iniciar la sesión o bien crearse una cuenta nueva si no la tiene. Para realizar cualquiera de las dos opciones deberá pulsar la opción correspondiente en el menú de la barra superior.



Figura 54 - Pantalla de inicio de sesión

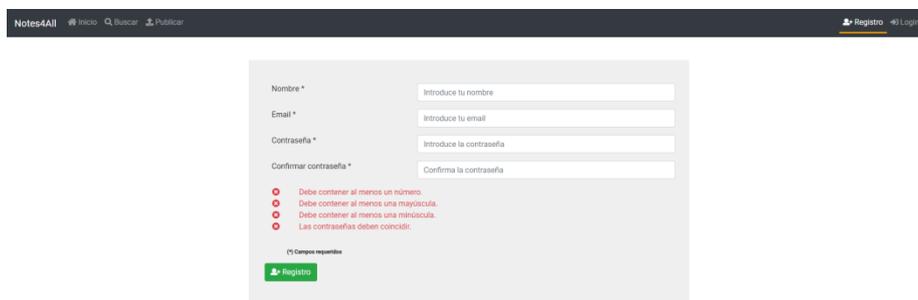


Figura 55 - Pantalla de registro

Una vez haya iniciado sesión será redirigido a la pantalla “Inicio”, que hemos visto anteriormente. Entonces ya podrá realizar las diferentes acciones disponibles como por ejemplo buscar alguna publicación, para ello deberá ir a la pantalla “Buscar” e introducir la búsqueda.

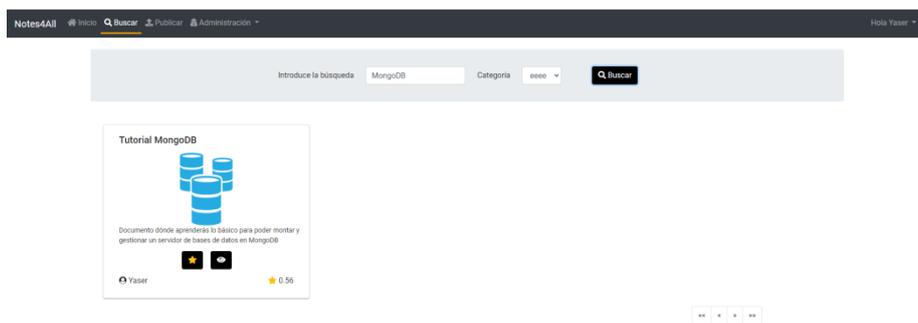


Figura 56 - Pantalla de búsquedas

Después de leer y leer documentos encuentra el que le sirve para su propósito y decide dejar una puntuación, para ello deberá pulsar sobre el botón con el icono de una estrella e introducir su puntuación.

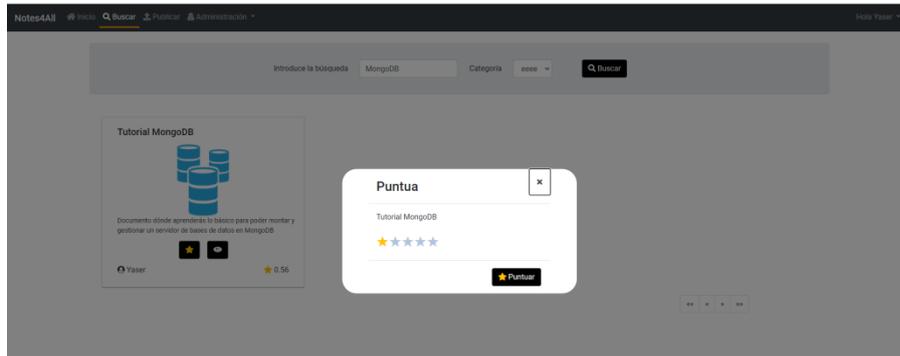


Figura 57 - Pantalla de puntuación de un documento

Si el usuario quiere ver los datos relevantes de su perfil, puede hacerlo desde la sección “Mi perfil” situada en la parte derecha del menú superior, en ella encontrará lo siguiente:

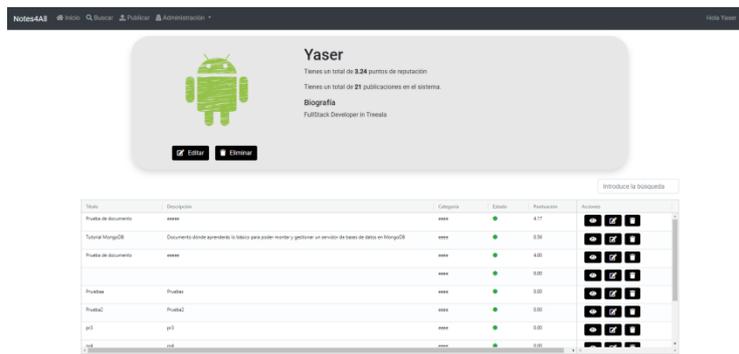


Figura 58 - Pantalla de “Mi perfil”

Cuando el usuario quiera realizar una publicación en el sistema, podrá realizarla desde la sección “Publicar” que es una de las opciones situadas en el menú de la barra superior, en ella encontrará el formulario con los campos necesarios para poder realizar dicha publicación:

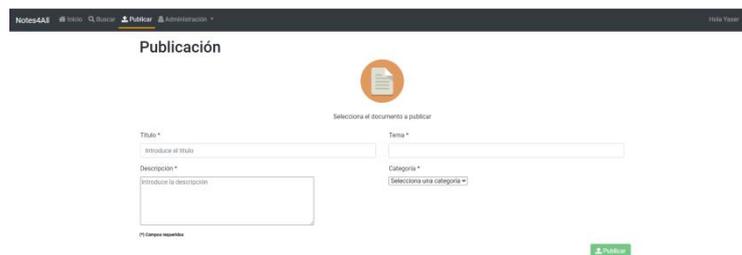


Figura 59 - Pantalla de “Publicar”

7.2.2 Aplicación móvil

Cuando el usuario inicie la aplicación se le mostrará el formulario de inicio de sesión y en el caso de que aún no tenga una cuenta registrada en el sistema, tendrá un enlace a la página de registro.

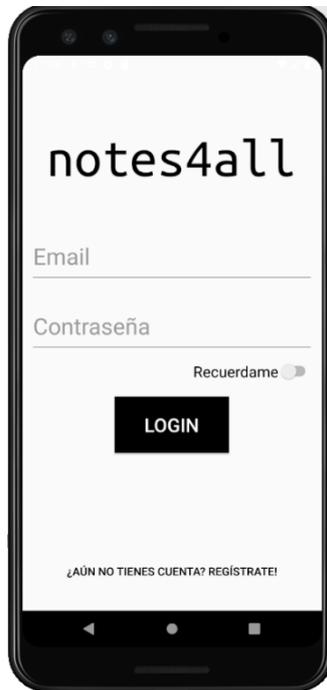


Figura 60 - Pantalla "Inicio de sesión"



Figura 61 - Pantalla "Registro"

Una vez que el usuario haya iniciado sesión, se le mostrará la pantalla de inicio donde aparecerán todos los documentos registrados en el sistema además de un buscador para que pueda filtrarlos.



Figura 62 - Pantalla de inicio

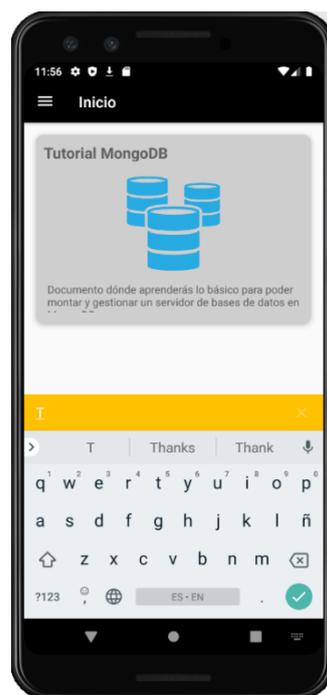


Figura 63 - Pantalla "Inicio" con un filtro aplicado

Una vez haya encontrado el documento que satisface sus necesidades, puede hacer *click* encima de él y aparece la pantalla de los detalles del documento, dónde además, podrá puntuarlo:



Figura 64 - Pantalla "Detalles"



Figura 65 - Pantalla "Puntuar documento"

Para acceder a los detalles de su cuenta, puede pulsar en el botón del menú llamado “Cuenta” y aparecerá la siguiente pantalla:

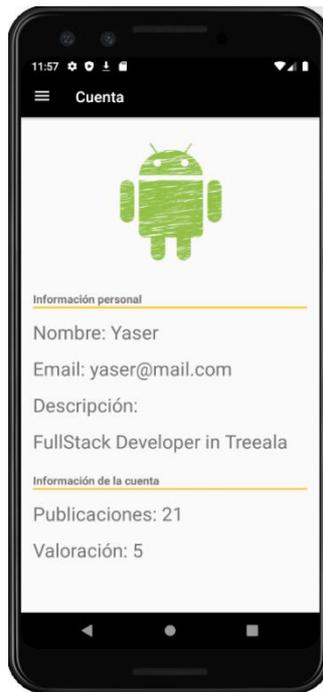


Figura 66 - Pantalla “Mi perfil”

Para realizar una publicación, podrá pulsar en la opción “Publicar” situada en el menú y le aparecerá la pantalla de publicación con el formulario que debe rellenar para poder realizar dicha publicación:



Figura 67 - Pantalla “Publicar”

7.2 Manual de administradores

Con este manual se pretende orientar al usuario administrador en el uso de la aplicación web. Para iniciar la aplicación el usuario tendrá que entrar en la url <https://www.notes4all.tk> y al acceder tendrá que autenticarse.



Figura 68 - Pantalla “Inicio de sesión”

Una vez se haya iniciado sesión en el sistema aparecerá una opción más en el menú llamada “Administración”, al hacer *click* encima, aparecerán las opciones disponibles para administradores.

Entre ellas está la opción “Categorías” desde la que podrá listar las categorías existentes en el sistema, crear una nueva, editar alguna de las existentes o eliminarla.

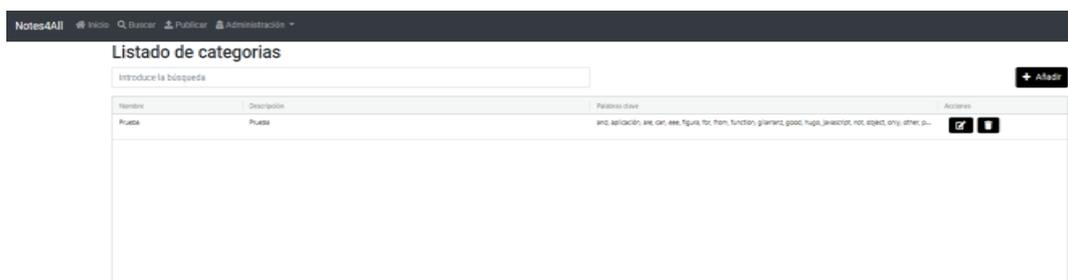


Figura 69 - Pantalla “Categorías”

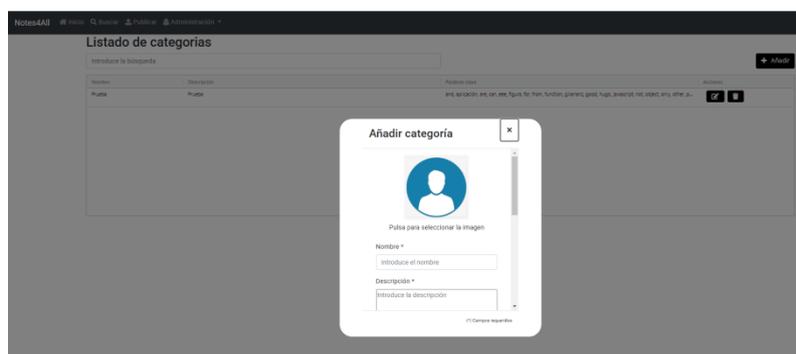


Figura 70 - Pantalla “Crear categoría”

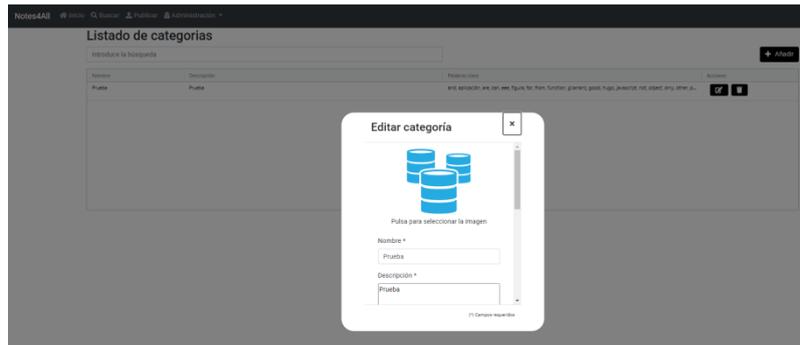


Figura 71 - Pantalla “Editar categoría”

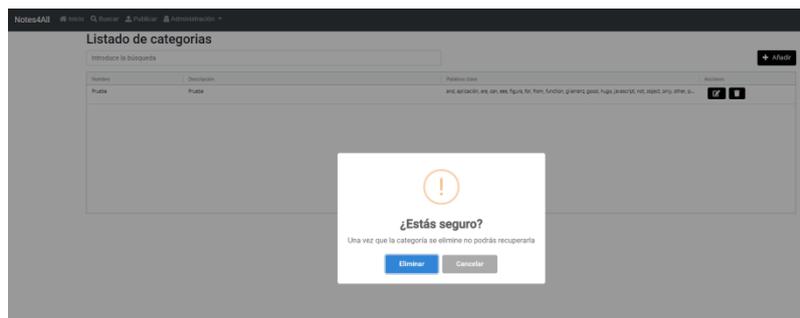


Figura 72 - Pantalla “Eliminar categoría”

Entre las opciones disponibles también está la de “Documentos” en la que podrá encontrar dos listados, uno de ellos con los documentos auto aprobados por el validador y otro con los documentos pendientes de revisión. Además podrá aprobar y/o desaprobado un documento o eliminarlo.

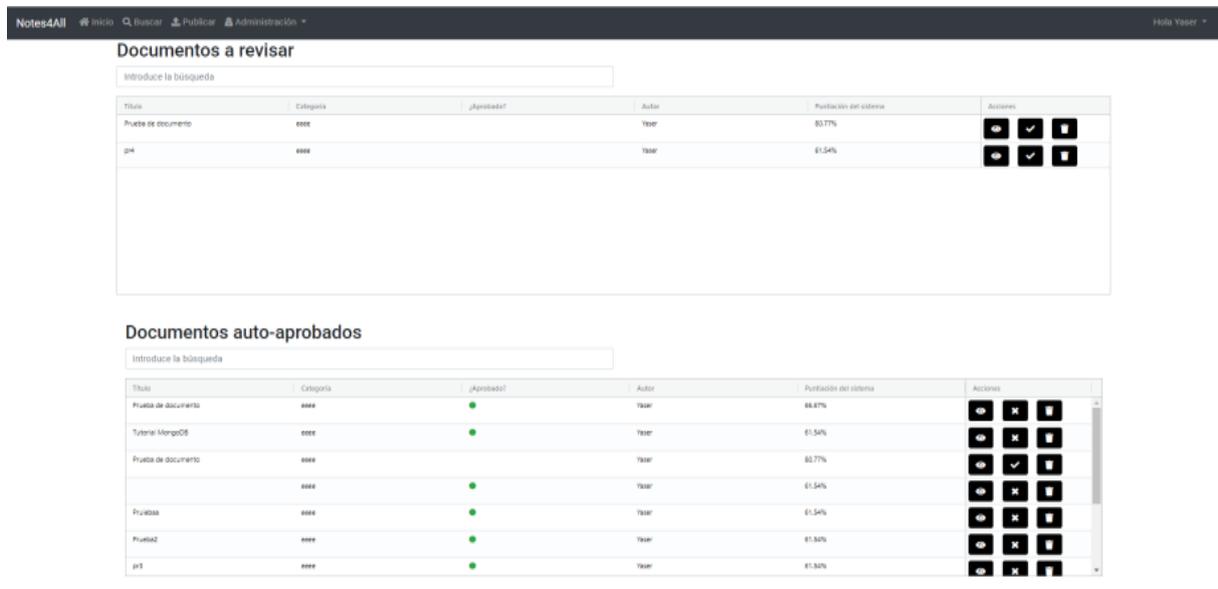


Figura 73 - Pantalla “Documentos”

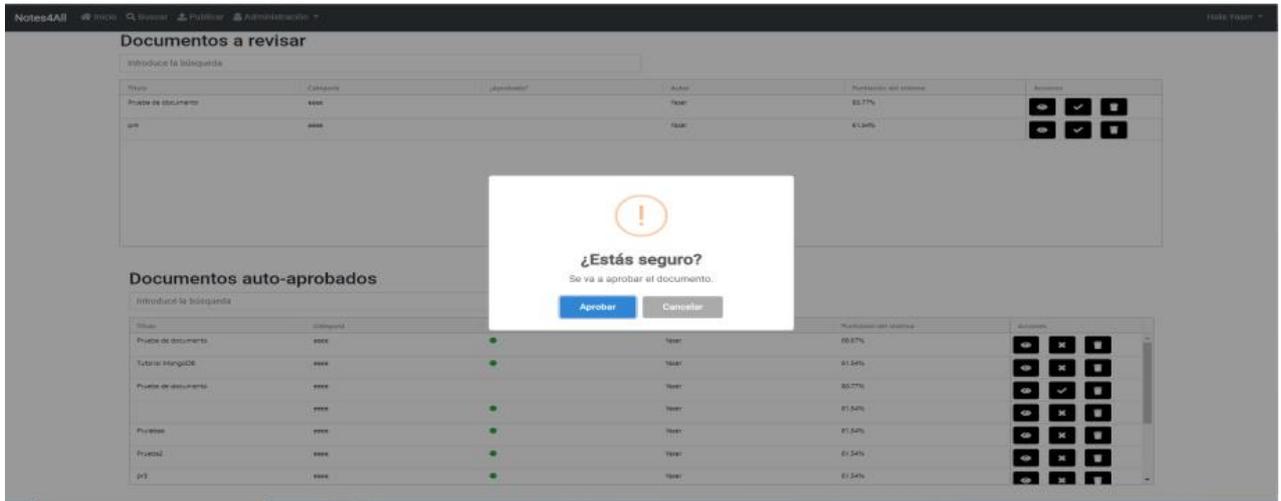


Figura 74 - Pantalla “Aprobar documento”

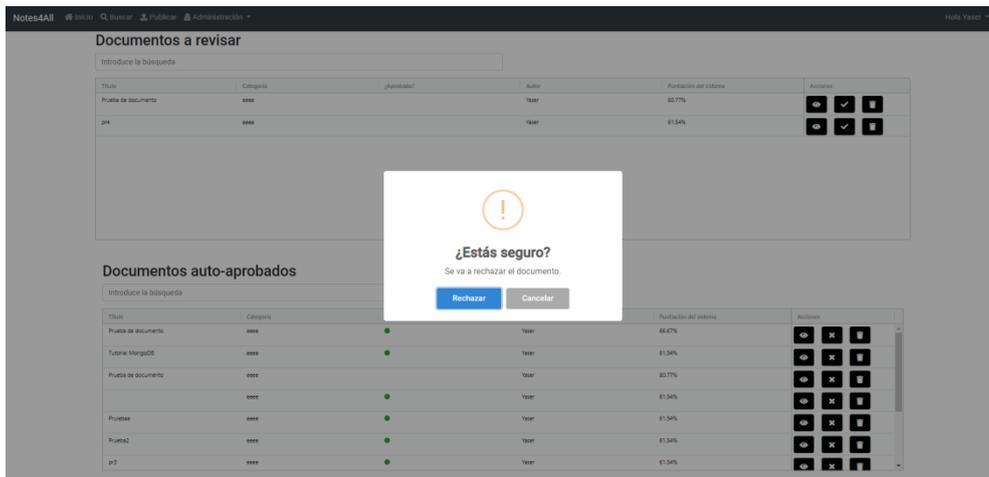


Figura 75 - Pantalla “Rechazar documento”

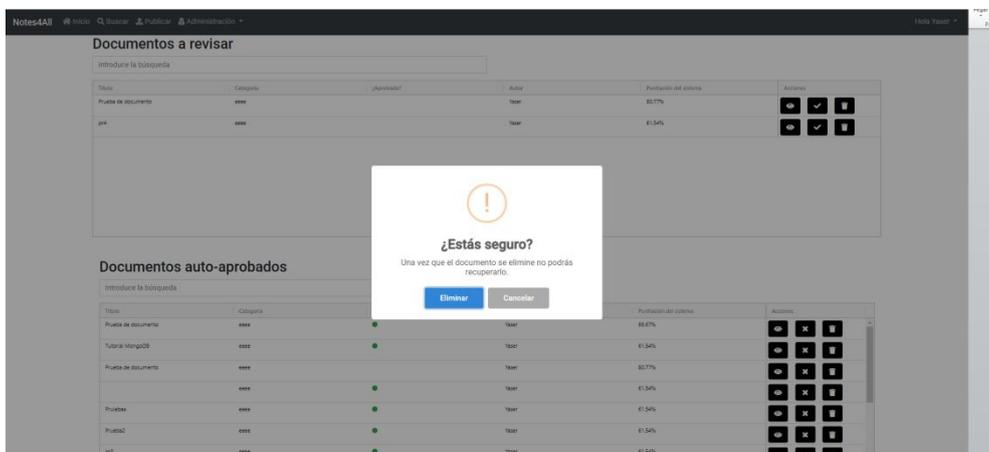


Figura 76 - Pantalla “Eliminar documento”

Por último, el usuario administrador podrá acceder a la sección “Usuarios” dónde dispondrá de un listado con todos los usuarios registrados en el sistema. Además de esto, podrá otorgar permisos de administración, ver los detalles o eliminarlos.

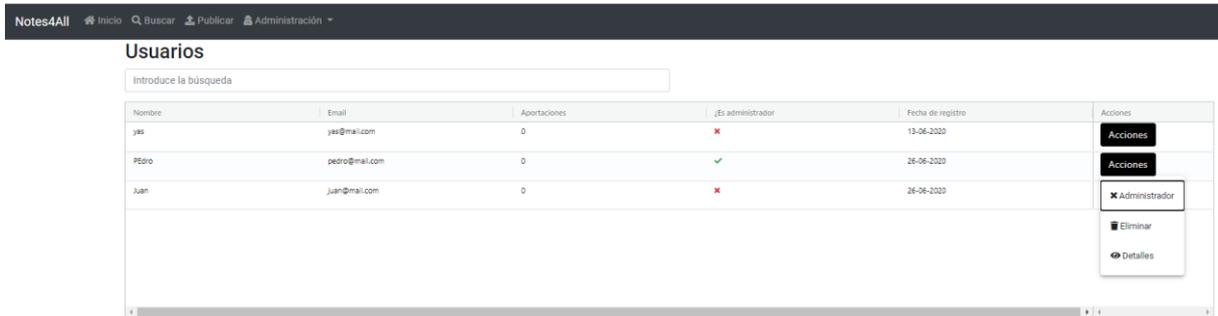


Figura 77 - Pantalla “Usuarios”

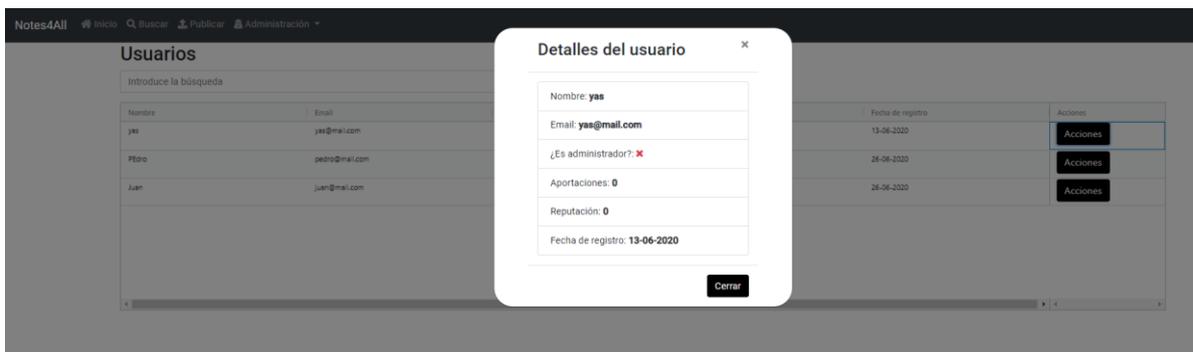


Figura 78 - Pantalla “Detalles”

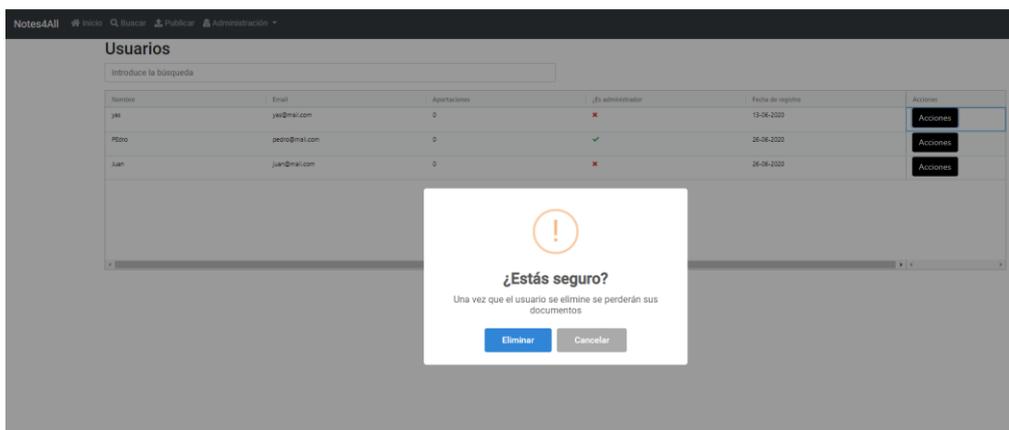


Figura 79 - Pantalla “Eliminar usuario”

Capítulo 8 – Conclusiones y mejoras

Uno de los objetivos principales de este proyecto era el desarrollar un sistema de compartición de apuntes sencillo e intuitivo y sobre todo capaz de validar los documentos que se publican. Considero que se ha conseguido ya que el sistema es capaz de validar los documentos con bastante acierto y “aprender” a base de analizar documentos ya que se aprovechan las palabras más repetidas del documento para añadirlas a la lista de palabras clave de cada categoría.

Otro de los objetivos marcados en la realización de este trabajo final de carrera, es el de intentar ir más allá de lo estrictamente académico y poner en práctica conceptos, metodologías o lenguajes que no se han visto durante los años de carrera, esto conlleva una gran carga de aprendizaje autónomo y autogestión del tiempo que después será vital en el ámbito laboral consecuencia de este apasionante mundo cambiante cómo es la informática.

Durante el desarrollo del sistema han aparecido bastantes complicaciones, creo que destacaría el control del código asíncrono de *Javascript* y el *Event Loop* de *NodeJs*, además de intentar desarrollar de la forma más eficiente el validador de documentos. Considero que he sido capaz de solventar dichos inconvenientes a base de esfuerzo y dedicación.

Llegados a este punto, puedo finalizar este proyecto diciendo que ha sido todo un reto y me ha servido para adquirir nuevos conocimientos como por ejemplo *Angular* y *MongoDb* tecnologías de las que partía de cero. También, considero que he desarrollado mi capacidad de resolución de problemas ante situaciones algo frustrantes y mi capacidad de aprendizaje autónomo, aptitudes que considero fundamentales en este mundo en constante cambio como es la informática.

A continuación muestro una serie de futuras mejoras, que por limitación de tiempo y recursos no se han podido implementar en este trabajo de fin de grado pero que sin ninguna duda sería realmente interesante de aplicar en un futuro no muy lejano:

- Mejorar el algoritmo de clasificación de los documentos, quizá implementando un algoritmo de búsqueda de palabras clave o indexación de documentos como pueden ser el algoritmo *RAKE* o el algoritmo *KSAS*.
- Implementar tests unitarios para el *backend* con el objetivo de desarrollar un entorno de pruebas automatizadas.
- Crear una tarea automatizada que ejecute un *script* que elimine los documentos rechazados después de que hayan pasado unos días desde su publicación.
- Mejorar la estructura del *backend* para facilitar su escalabilidad si llegase el momento.
- Permitir la autenticación tanto de la aplicación web como de la móvil a través de servicios como Google o Facebook.
- Implementar un pequeño sistema de recomendación para que los usuarios visualizasen aquellos documentos que les puedan resultar de mayor interés.

Referencias

- **Instalación de NGINX.** Disponible en la url: <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04> [Consultado en abril de 2020].
- **Instalación de certificados SSL en NGINX.** Disponible en la url: <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04> [Consultado en mayo de 2020].
- **Instalación y configuración de PM2.** Disponible en la url: <https://www.digitalocean.com/community/tutorials/como-configurar-una-aplicacion-de-node-js-para-produccion-en-ubuntu-18-04-es> [Consultado en mayo de 2020].
- **Instalación y configuración de la librería dotenv.** Disponible en la url: <https://www.npmjs.com/package/dotenv> [Consultado en mayo de 2020].
- **Uso de Express en NodeJs.** Documentación oficial disponible en la url: <https://expressjs.com/es/4x/api.html#express> [Consultado en mayo de 2020].
- **Pensar asíncronamente en un mundo asíncrono.** Disponible en la url: <https://medium.com/@ulisesGascon/pensar-as%C3%ADncronamente-en-un-mundo-s%C3%ADncrono-8e25cfcfd83> [Consultado en abril de 2020].
- **The NodeJs Event Loop.** Disponible en la url: <https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/> [Consultado en abril de 2020].
- **Documentación de Angular 8.** Disponible en la url: <https://angular.io/> [Consultado en mayo de 2020]
- **Librería axios.** Disponible en la url: <https://www.npmjs.com/package/axios> [Consultado en mayo de 2020].
- **RecyclerView & CardView.** Disponible en la url: <https://medium.com/@bsoto.dev/recyclerview-y-cardview-59277bdcfbe9> [Consultado en junio de 2020].
- **Librería Picasso.** Disponible en la url: <https://github.com/square/picasso> [Consultado en junio de 2020].
- **Librería Retrofit.** Disponible en la url: <https://square.github.io/retrofit/> [Consultado en junio de 2020].
- **Alert Dialogs.** Disponible en la url: <https://developer.android.com/guide/topics/ui/dialogs?hl=es-419> [Consultado en junio de 2020].
- **SharedPreferences.** Disponible en la url: <https://developer.android.com/reference/android/content/SharedPreferences> [Consultado en junio de 2020].
- **Menú lateral con Navigation Drawer.** Disponible en la url: <https://danielme.com/2018/12/19/disenio-android-menu-lateral-con-navigation-drawer/> [Consultado en junio de 2020].

Índice de figuras

Figura 1 - Árbol de características	11
Figura 2 - Logo de Visual Studio Code	13
Figura 3 - Logo de Node.js	13
Figura 4 - Logo de Angular	14
Figura 5 - Logo de NPM	14
Figura 6 - Logo de MongoDB	15
Figura 7 - Logo de Android Studio	15
Figura 8 - Logo de GitHub	16
Figura 9 - Logo de Google Cloud Platform	16
Figura 10 - Logo de NGINX	17
Figura 11 - Whuolah	20
Figura 12 - Docsity	20
Figura 13 - Rincón del vago	21
Figura 14 - Modelo iterativo incremental	22
Figura 15 - Planificación temporal	23
Figura 16 - Diagrama de Gantt de la planificación temporal	24
Figura 17 - Jerarquía de los actores del sistema	34
Figura 18 - Diagrama de casos de la aplicación web	37
Figura 19 - Diagrama de casos de la aplicación móvil	38
Figura 21 - Diagrama de entidad relación	53
Figura 22 - Arquitectura lógica de la aplicación web	56
Figura 23 - Arquitectura lógica de la aplicación móvil	57
Figura 24 - Arquitectura física del sistema	58
Figura 25 - Diagrama de clases de la aplicación móvil	59
Figura 26 - Diagrama de clases de la aplicación web	60
Figura 27 - Base de datos MongoDB	61
Figura 28 - Diseño de la pantalla "Inicio de sesión"	63
Figura 29 - Diseño de la pantalla "Registro"	63
Figura 30 - Diseño de la pantalla "Inicio"	64
Figura 31 - Diseño del menú de la aplicación	64
Figura 32 - Diseño de la pantalla "Publicar"	65
Figura 33 - Diseño de la pantalla "Cuenta"	65
Figura 34 - Diseño del pop up "Cerrar sesión"	66
Figura 35 - Diseño de la pantalla "Inicio de sesión"	67
Figura 36 - Diseño de la pantalla "Registro"	67
Figura 37 - Diseño de la pantalla "Inicio"	68
Figura 38- Diseño de la pantalla "Buscar"	68
Figura 39 - Diseño de la pantalla "Publicar"	69
Figura 40 - Diseño de la pantalla "Mi perfil"	69
Figura 41 - Diseño de la pantalla "Usuarios"	70
Figura 42 - Diseño de la pantalla "Categorías"	70
Figura 43 - Diseño de la pantalla "Documentos"	71
Figura 44 - Diseño del formulario de creación o edición de datos	71
Figura 45 - Instalación NodeJs y npm	78
Figura 46 - Instalación de los repositorios	78
Figura 47 - Instalación de Angular 8.3.21	78
Figura 48- Instalación de MongoDB	78
Figura 49 - Instalación de los certificados SSL	79
Figura 50 - Instalación de PM2 e inicio del servidor	79

<i>Figura 51 - Fichero de variables de entorno</i>	79
<i>Figura 52 - Permisos de ejecución</i>	79
<i>Figura 53 - Contenido del fichero deploy.sh</i>	80
<i>Figura 54 - Pantalla de inicio de la aplicación web</i>	80
<i>Figura 55 - Pantalla de inicio de sesión</i>	81
<i>Figura 56 - Pantalla de registro</i>	81
<i>Figura 57 - Pantalla de búsquedas</i>	81
<i>Figura 58 - Pantalla de puntuación de un documento</i>	82
<i>Figura 59 - Pantalla de "Mi perfil"</i>	82
<i>Figura 60 - Pantalla de "Publicar"</i>	82
<i>Figura 61 - Pantalla "Inicio de sesión"</i>	83
<i>Figura 62 - Pantalla "Registro"</i>	83
<i>Figura 63 - Pantalla de inicio</i>	84
<i>Figura 64 - Pantalla "Inicio" con un filtro aplicado</i>	84
<i>Figura 65 - Pantalla "Detalles"</i>	85
<i>Figura 66 - Pantalla "Puntuar documento"</i>	85
<i>Figura 67 - Pantalla "Mi perfil"</i>	86
<i>Figura 68 - Pantalla "Publicar"</i>	86
<i>Figura 69 - Pantalla "Inicio de sesión"</i>	87
<i>Figura 70 - Pantalla "Categorías"</i>	87
<i>Figura 71 - Pantalla "Crear categoría"</i>	87
<i>Figura 72 - Pantalla "Editar categoría"</i>	88
<i>Figura 73 - Pantalla "Eliminar categoría"</i>	88
<i>Figura 74 - Pantalla "Documentos"</i>	88
<i>Figura 75 - Pantalla "Aprobar documento"</i>	89
<i>Figura 76 - Pantalla "Rechazar documento"</i>	89
<i>Figura 77 - Pantalla "Eliminar documento"</i>	89
<i>Figura 78 - Pantalla "Usuarios"</i>	90
<i>Figura 79 - Pantalla "Detalles"</i>	90
<i>Figura 80 - Pantalla "Eliminar usuario"</i>	90

Índice de tablas

Tabla 1 - Tabla de objetivos	10
Tabla 2 - Comparativa de aplicaciones	21
Tabla 3 - Costes de componentes hardware.....	25
Tabla 4 - Costes de componentes software.....	25
Tabla 5 - Costes de personal.....	26
Tabla 6 – Complejidad.....	26
Tabla 7 - Entradas de usuario.....	27
Tabla 8 - Salida del usuario.....	27
Tabla 9 - Ficheros lógicos internos.....	27
Tabla 10 - Puntos de función sin ajustar (PFNA)	28
Tabla 11 - Factores de ajuste.....	28
Tabla 12 - Costes de personal (puntos de función).....	29
Tabla 13 - Factores de ajuste de COCOMO.....	30
Tabla 14 - Factor de ajuste de COCOMO.....	30
Tabla 15 - Ponderaciones de COCOMO.....	31
Tabla 16 - Costes de personal por COCOMO.....	32
Tabla 17 - Tabla de requisitos de usuario de la aplicación web.....	35
Tabla 18 - Requisitos de usuario de la aplicación móvil.....	36
Tabla 19 - Caso de uso “Registro” (CU-01).....	39
Tabla 20 - Caso de uso “Inicio de sesión” (CU-02).....	40
Tabla 21 - Caso de uso “cerrar sesión” (CU-03).....	40
Tabla 22 - Caso de uso “Realizar búsquedas” (CU-04).....	41
Tabla 23 - Caso de uso “Visualizar documentos” (CU-05)	41
Tabla 24 - Caso de uso “Publicar documentos” (CU-06).....	42
Tabla 25 - Caso de uso “Editar documentos” (CU-07).....	43
Tabla 26 - Caso de uso “Eliminar documentos” (CU-08)	43
Tabla 27 - Caso de uso “Descargar documentos” (CU-09).....	44
Tabla 28 - Caso de uso “Puntuar documentos” (CU-10)	44
Tabla 29 - Caso de uso “Listar documentos” (CU-11).....	45
Tabla 30 - Caso de uso “Aprobar documentos” (CU-12).....	45
Tabla 31 - Caso de uso “Desaprobar documentos” (CU-13).....	46
Tabla 32 - Caso de uso “Visualizar perfil” (CU-14).....	46
Tabla 33 - Caso de uso “Editar perfil” (CU-15).....	47
Tabla 34 - Caso de uso “Eliminar cuenta” (CU-16).....	47
Tabla 35 - Caso de uso “Dar permisos de administración” (CU-17)	48
Tabla 36 - Caso de uso “Listar usuarios” (CU-18).....	48
Tabla 37 - Caso de uso “Eliminar usuarios” (CU-19).....	49
Tabla 38 - Caso de uso “Listar categorías” (CU-20).....	49
Tabla 39 - Caso de uso “Crear categorías” (CU-21).....	50
Tabla 40 - Caso de uso “Editar categoría” (CU-22).....	51
Tabla 41 - Caso de uso “Eliminar categoría” (CU-23).....	51
Tabla 42 - Caso de uso “Notificar a los usuarios” (CU-24).....	52
Tabla 45 - Requisitos de información.....	53
Tabla 46 - Diccionario de datos de “Users”	54
Tabla 47 - Diccionario de datos de “Categories”	54
Tabla 48 - Diccionario de datos de “Documents”	54
Tabla 49 - PCN-01 Registro de usuario	72
Tabla 50 - PCN-02 Inicio de sesión.....	73

<i>Tabla 51 - PCN-03 Inicio</i>	<i>73</i>
<i>Tabla 52 - PCN-04 Mi perfil</i>	<i>73</i>
<i>Tabla 53 - PCN-05 Publicar</i>	<i>73</i>
<i>Tabla 54 - PCN-06 Puntuar un documento</i>	<i>74</i>
<i>Tabla 55 - PCN-07 Crear o editar categorías</i>	<i>74</i>
<i>Tabla 56 - PCN-08 Aprobar o desaprobar un documento</i>	<i>74</i>
<i>Tabla 57 - PCN-09 Listado de usuarios</i>	<i>74</i>
<i>Tabla 58 - PCN-10 Listado de categorías</i>	<i>75</i>
<i>Tabla 59 - PCN-11 Listado de documentos.....</i>	<i>75</i>
<i>Tabla 60 - PCN-12 Eliminar un usuario</i>	<i>75</i>
<i>Tabla 61 - PCN-13 Dar permisos de administración</i>	<i>75</i>
<i>Tabla 62 - PCN-15 Buscar documentos</i>	<i>76</i>
<i>Tabla 63 - PCN-16 Detalles de usuario</i>	<i>76</i>
<i>Tabla 64 - PCN-17 Eliminar documentos</i>	<i>76</i>
<i>Tabla 65 - PCN-18 Eliminar categorías</i>	<i>77</i>
<i>Tabla 66 - PCN-19 Eliminar cuenta.....</i>	<i>77</i>

