

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**APLICACIÓN WEB PARA GESTIÓN DE NOTAS
PERSONALES**

Autora:

Directores:

Septiembre 2017

Memoria Técnica



UNIVERSIDAD DE CÓRDOBA

Índice

Índice de Figuras	8
Índice de Tablas.....	9
I. Introducción al proyecto	1
1. Introducción	2
2. Definición del problema	3
2.1. Definición del problema real.....	3
2.2. Definición del problema técnico	3
2.2.1. Funcionamiento	4
2.2.2. Entorno.....	4
2.2.3. Vida esperada.....	4
2.2.4. Ciclo de mantenimiento	5
2.2.5. Competencia	5
2.2.6. Aspecto externo	6
2.2.7. Estandarización	6
2.2.8. Calidad y fiabilidad	6
2.2.9. Programa de tareas	7
2.2.10. Pruebas.....	9
2.2.11. Seguridad.....	9
3. Objetivos	10
4. Antecedentes	11
4.1. Evernote	11
4.2. Tomboy y Tomdroid	11
4.3. Google Keep	12
4.4. Dropbox Paper	12
5. Restricciones	13
5.1. Factores dato.....	13
5.2. Factores estratégicos	13
6. Recursos	15
6.1. Recursos humanos	15
6.2. Recursos software	15
6.3. Recursos hardware.....	16
II. Análisis del sistema	17
7. Especificación de Requisitos	18

7.1.	Requisitos Funcionales	18
7.1.1.	Requisitos recogidos del análisis previo	18
7.1.2.	Diagramas de Casos de uso	19
7.1.2.1.	Diagrama de casos de uso de Sistema de Autenticación	19
7.1.2.2.	Diagrama de casos de uso de Sistema de Carpetas	23
7.1.2.3.	Diagrama de casos de uso de Sistema de Nota.....	25
7.1.2.4.	Relación requisitos de Usuario con casos de Uso	29
7.2.	Requisitos No Funcionales	29
7.2.1.	Requisitos del Producto	29
7.2.1.1.	Usabilidad	29
7.2.1.2.	Funcionalidad	30
7.2.1.3.	Rendimiento y Escalabilidad	30
7.2.1.4.	Mantenimiento	30
7.2.1.5.	Seguridad.....	30
7.2.1.6.	Confiabilidad	30
7.2.1.7.	Portabilidad	31
7.2.2.	Requisitos Organizacionales.....	31
7.2.2.1.	Software	31
7.2.2.2.	Hardware.....	31
7.2.3.	Requisitos externos	31
7.2.3.1.	Estándares	31
8.	Especificación del Sistema.....	32
8.1.	Especificación de la Información – Modelo Entidad-Relación	32
8.1.1.	Entidades y Jerarquías.....	33
8.1.1.1.	Entidad User	33
8.1.1.2.	Entidad Persona	33
8.1.2.	Relaciones entre las entidades del sistema	34
8.1.2.1.	Relación User-Persona	34
8.1.3.	Modelo Entidad-Relación.....	35
8.2.	Modelo de datos Relacional.....	35
8.2.1.	Tablas del sistema	36
8.2.1.1.	Tabla User.....	36
8.2.1.2.	Tabla Persona	36
8.2.2.	Esquema Relacional.....	37

8.2.3.	Diagrama Relacional.....	37
III.	Diseño de la aplicación.....	38
9.	Diseño del sistema	39
9.1.	Arquitectura del Sistema.....	39
9.1.1.	Servidor Web.....	39
9.1.2.	Cliente Web.....	40
9.1.3.	Gestor de notas	40
9.2.	Gestor de notas	40
9.3.	Backend	43
9.3.1.	Rutas.....	44
9.4.	Frontend.....	45
9.4.1.	Rutas.....	45
9.5.	Diseño procedimental	47
9.5.1.	Sincronización con Dropbox.....	47
9.5.2.	Crear notas	48
9.5.3.	Listado de notas	49
9.5.4.	Eliminación de notas	50
9.5.5.	Edición de notas	51
9.6.	Interfaz del usuario	52
9.6.1.	Estructura básica de la interfaz	52
9.6.2.	Descripción de Registro e Inicio de Sesión.....	53
9.6.3.	Descripción de Perfil.....	54
9.6.4.	Descripción de Editar Perfil	55
9.6.5.	Descripción de Mis Carpetas.....	56
9.6.6.	Descripción de Crear Notas.....	57
9.6.7.	Descripción de Mis Notas.....	58
IV.	Pruebas.....	61
10.	Pruebas del Sistema	62
10.1.	Pruebas de casos de uso	62
10.1.1.	Caso de prueba registrar a un usuario	62
10.1.2.	Caso de prueba iniciar sesión a un usuario	63
10.1.3.	Caso de prueba cerrar sesión de un usuario.....	63
10.1.4.	Caso de prueba editar perfil de un usuario.....	64
10.1.5.	Caso de prueba ver perfil de un usuario	64

10.1.6.	Caso de prueba crear carpeta a un usuario	65
10.1.7.	Caso de prueba ver carpetas de un usuario.....	65
10.1.8.	Caso de prueba borrar una carpeta de un usuario	66
10.1.9.	Caso de prueba elegir una carpeta	66
10.1.10.	Caso de prueba crear una nota a un usuario	67
10.1.11.	Caso de prueba borrar una nota de un usuario	67
10.1.12.	Caso de prueba listar notas de un usuario.....	68
10.1.13.	Caso de prueba leer una nota de un usuario	68
10.1.14.	Caso de prueba editar nota de un usuario.....	69
V.	Conclusiones.....	70
11.	Conclusiones.....	71
11.1.	Conclusiones sobre los objetivos planteados	71
11.2.	Conclusiones de la fase de pruebas	72
11.3.	Futuras mejoras.....	72
	Bibliografía	73

Índice de Figuras

Figura 1: Diagrama del Sistema de Autenticación	19
Figura 2: Diagrama del Sistema de Carpetas	23
Figura 3: Diagrama del Sistema de Notas.....	25
Figura 4: Entidad User.....	33
Figura 5: Entidad Persona	33
Figura 6: Relación User-Persona	34
Figura 7: Diagrama Entidad-Relación	35
Figura 8: Diagrama Relacional.....	37
Figura 9: Estructura de ficheros.....	41
Figura 10: Diagrama de secuencias: Sincronización con Dropbox.....	47
Figura 11: Diagrama de secuencias: Creación de notas	48
Figura 12: Diagrama de secuencias: Listado de notas.....	49
Figura 13: Diagrama de secuencias: Eliminación de notas.....	50
Figura 14: Diagrama de secuencias: Edición de notas.....	51
Figura 15: Interfaz básica	52
Figura 16: Interfaz de Registro.	53
Figura 17: Interfaz de Inicio de Sesión.	53
Figura 18: Interfaz de Perfil.	54
Figura 19: Interfaz de Editar Perfil.....	55
Figura 20: Interfaz de Mis Carpetas.....	56
Figura 21: Interfaz de Carpeta Borrada.	56
Figura 22: Interfaz de Crear Notas, elegir carpeta.	57
Figura 23: Interfaz de Crear Notas, elegir nombre.....	57
Figura 24: Interfaz de Mis Notas, elegir carpeta.....	58
Figura 25: Interfaz de Mis Notas	59
Figura 26: Interfaz de Mis Notas, lectura de nota.....	59
Figura 27: Interfaz de Mis Notas, edición de nota.	60
Figura 28: Interfaz de Mis Notas, borrado de nota.....	60

Índice de Tablas

Tabla 1: Distribución temporal del proyecto	8
Tabla 2: Especificación del caso de uso Registrarse	20
Tabla 3: Especificación del caso de uso Iniciar Sesión.....	21
Tabla 4: Especificación del caso de uso Cerrar Sesión	22
Tabla 5: Especificación del caso de uso Editar Perfil.....	22
Tabla 6: Especificación del caso de uso Ver Perfil.....	23
Tabla 7: Especificación del caso de uso Crear Carpeta.....	24
Tabla 8: Especificación del caso de uso Ver Carpetas	24
Tabla 9: Especificación del caso de uso Borrar Carpeta	25
Tabla 10: Especificación del caso de uso Elegir Carpeta.....	26
Tabla 11: Especificación del caso de uso Crear Nota	26
Tabla 12: Especificación del caso de uso Borrar Nota.....	27
Tabla 13: Especificación del caso de uso Listar Notas.....	27
Tabla 14: Especificación del caso de uso Leer Nota	28
Tabla 15: Especificación del caso de uso Editar Nota.....	28
Tabla 16: Relación Requisitos Análisis previo con Casos de Uso	29
Tabla 17: Entidad User	33
Tabla 18: Entidad Persona	34
Tabla 19: Esquema Relacional.....	37
Tabla 20: Mapeo de URLs en Django	45
Tabla 21: Mapeo de URLs en Django.....	46
Tabla 22: Tabla Caso de Pruebas registrar a un usuario	62
Tabla 23: Tabla Caso de Pruebas iniciar sesión de un usuario	63
Tabla 24: Tabla Caso de Pruebas cerrar sesión de un usuario.....	63
Tabla 25: Tabla Caso de Pruebas editar perfil de un usuario	64
Tabla 26: Tabla Caso de Pruebas ver perfil de un usuario	64
Tabla 27: Tabla Caso de Pruebas crear carpeta a un usuario	65
Tabla 28: Tabla Caso de Pruebas ver carpetas de un usuario	65
Tabla 29: Tabla Caso de Pruebas borrar una carpeta de un usuario	66
Tabla 30: Tabla Caso de Pruebas elegir una carpeta.....	66
Tabla 31: Tabla Caso de Pruebas crear una nota a un usuario	67
Tabla 32: Tabla Caso de Pruebas borrar una nota de un usuario	67
Tabla 33: Tabla Caso de Pruebas listar notas de un usuario	68
Tabla 34: Tabla Caso de Pruebas leer una nota de un usuario.....	68
Tabla 35: Tabla Caso de Pruebas editar nota de un usuario	69

I. Introducción al proyecto

1. Introducción

Como bien sabemos, las personas siempre estamos apuntando cosas que tenemos pendientes y/o necesitamos recordar, como por ejemplo: ideas que se nos ocurren y no queremos que se nos olviden; datos académicos, como son las notas de exámenes; recetas; precios de materiales que queramos comprar; listas de regalos; películas que queremos ver; contraseñas y muchas cosas más que se nos van viniendo a la mente y no queremos que se nos olviden.

Antiguamente íbamos apuntando las ideas en papeles, hoy en día hemos cambiado los papeles por los dispositivos digitales. Sin embargo, las anotamos en sitios diferentes, lo que nos hace no saber luego donde las hemos apuntado.

Hoy en día, el uso de dispositivos móviles es generalizado y puesto que la mayoría de veces tenemos a mano alguno de nuestros dispositivos, sería ideal poder utilizarlos para anotar todo aquello que queramos. Además, ya que estos dispositivos nos permiten tener una conexión a internet continua, sería útil que estas notas se sincronizasen en todos los demás dispositivos que tengamos para mantenerlo todo actualizado.

Como respuesta a esto, nos proponemos crear una aplicación de notas de entorno web, que pueda ser utilizada desde cualquier dispositivo, con sistemas diferentes. Para que tengamos autonomía en nuestra información y poder decidir qué hacer con nuestra información en cada momento, cada nota estaría en un archivo independiente, el cual podremos llevarlo a donde queramos. Además nuestras notas tendrán un formato de texto plano, con poco TAG.

Con esta idea pretendemos facilitar a todos los tipos de personas, tanto en el ámbito laboral como en el ámbito personal, a que puedan guardar datos importantes en cualquier momento y lugar.

2. Definición del problema

En este punto se tratara de mostrar detalladamente el problema al que se pretende dar solución con la realización de este proyecto. Para definir el problema, se distingue entre el problema real, que se refiere a la visión que ofrece el usuario, y el problema técnico, que define el problema desde el punto de vista de la ingeniería.

2.1. Definición del problema real

En el día a día se nos van planteado miles de ideas, fechas importantes, anotaciones del trabajo o del estudio, en resumen miles de datos que debemos recordar durante una corta temporada, una larga temporada o incluso para siempre.

En este casos, siempre recurrimos a anotar nuestros datos en el lugar más cercano, antes de que se nos olviden para no perderlos. A día de hoy es menos usado el método de anotarlo en un papel pero aun así este método sigue utilizándose. El problema de este método es que podemos, en cualquier momento, perder ese papel, y, por tanto, perder los datos anotados. Si los datos tienen que perdurar a lo largo del tiempo debemos tener bien guardado ese papel si no queremos perder tal información.

Otro problema de apuntar nuestra información en una nota de papel, es que si llegamos a perder dicha nota, ésta podría llegar hasta otras personas que utilizaran esa información sin nuestro permiso.

Sin embargo, ahora es más habitual tener a mano un dispositivo móvil que un trozo de papel. De manera que una solución a dicho problema sería poder utilizar nuestro teléfono móvil para guardar nuestras notas y que podamos visualizarlas e incluso editarlas. Además, también podríamos realizar esto por medio de otros dispositivos como un ordenador o tablet.

2.2. Definición del problema técnico

Para llevar a cabo la identificación del problema técnico se empleará una técnica de ingeniería denominada PDS (Product Design Specification). Esta técnica permite realizar un análisis de los principales condicionantes técnicos del problema mediante la respuesta de una serie de cuestiones básicas.

2.2.1. Funcionamiento

En lo que respecta al funcionamiento de nuestra aplicación, lo que pretendemos es que el usuario cree notas permitiéndole editarlas en todo momento, bien para añadir información nueva o bien para borrarla en caso de que no le sea de utilidad.

No obstante, el usuario podrá visualizar la nota sin realizar ningún cambio a modo de recordatorio o alguna situación similar.

Cada nota tendrá un título representativo, el cual nos ayudará a distinguir una nota de otra. También queremos estructurar dichas notas, permitiendo añadir distintos encabezados y de esta forma cada usuario organizará las notas como más le beneficie. En las notas se podrá poner listas para que el usuario pueda enumerar su información.

Finalmente, se pretende llevar a cabo una clasificación de las notas en distintas categorías, y así poder asignar cada nota a una categoría creada. Asimismo, se permite cambiar la categoría de una nota por otra categoría distinta.

2.2.2. Entorno

La aplicación se podrá utilizar en cualquier entorno. Su público podrá ser de cualquier edad porque su función será útil para cualquier persona y para poder utilizar la aplicación solo deberá tener o un ordenador, móvil o tablet.

La aplicación tendrá una fácil usabilidad y mucha versatilidad para poder ser utilizada en cualquier lugar que posea conexión a internet de cualquier tipo. Esto nos ayudará para que el usuario pueda anotar todos sus datos en cualquier momento y lugar en el que se encuentre sin dificultad.

2.2.3. Vida esperada

La vida esperada de una aplicación es complicada de saberla ya que influyen muchos factores, pero se esperaría que la aplicación durara bastante tiempo porque la necesidad de anotar datos es algo que las personas llevan haciendo durante toda su vida y seguirán haciendo en el futuro. Por ello la aplicación cumplirá su objetivo hasta que llegue el momento en que las nuevas actualizaciones y nuevas aplicaciones la dejasen obsoleta. Esto podría ocurrir a lo largo de aproximadamente 5 años.

2.2.4. Ciclo de mantenimiento

El ciclo de mantenimiento se identifica con el conjunto de modificaciones que resultan necesarias para realizar una determinada aplicación y que esta pueda hacer frente a diferentes circunstancias que puedan surgir en cualquier momento, además de a nuevas exigencias procedentes del propio sistema o del usuario final que utilizara la aplicación.

El mantenimiento puede realizarse para cubrir varios tipos de necesidades generadas como pueden ser:

- *Adaptación:* Conjunto de actividades que se realizaran para poder adaptar nuestra aplicación al entorno tecnológico, que puede producir cambios por su actualizaciones.
- *Corrección:* Conjunto de actividades que se llevan a cabo para realizar las correcciones de errores que no han sido detectados en la aplicación hasta cierto momento.
- *Perfeccionamiento:* Conjunto de funciones que su propósito es la mejora en el funcionamiento de la aplicación o agregación de funciones requeridas que hacen que esta evolucione para ser mejorada. Dichas mejoras pueden consistir en aumentar la facilidad del mantenimiento de la aplicación para posibles modificaciones en el futuro, para funciones que realiza, para la eficiencia de la aplicación, entre otras cosas.

En principio, de cara al futuro de la aplicación no se han planeado seguir un mantenimiento determinado. Sin embargo, la aplicación tendrá un desarrollo que utilizara las mejores técnicas para que se pueda adaptar a cambios de librerías de una manera fácil y rápida, además de que se puedan añadir nuevas funciones.

2.2.5. Competencia

Actualmente existen varios proyectos en el mercado, con sus aplicaciones, que intentan resolver este problema como podrían ser: Evernote [1], Tomboy [2], Tomdroid [3], Google Keep [4] y Dropbox Paper [5]. Sin embargo, todas estas aplicaciones tienen carencias que hacen que no sean totalmente eficientes para los usuarios finales. Veremos en profundidad las competencias en el apartado de "Antecedentes".

2.2.6. Aspecto externo

La interfaz será totalmente intuitiva, de uso fácil, amigable, bien estructurado y con una buena calidad gráfica de cara al usuario final. Para que al usuario le sea fácil su manejo debe permitir un aprendizaje y uso rápido y atractivo.

2.2.7. Estandarización

Se ha decidido desarrollar la aplicación web con un framework llamado Django, codificado en Python, por lo que nos permite integrar las librerías y la reutilización de código.

La aplicación como se ha comentado anteriormente es una aplicación web, eso nos lleva a seguir estándares como HTML5 y CSS para garantizar el funcionamiento en cualquier navegador web, ya que en este proyecto se le da mucha importancia que la aplicación sea multiplataforma.

2.2.8. Calidad y fiabilidad

En la actualidad calidad y fiabilidad del software son dos características importantes que debemos tener en cuenta para el desarrollo del proyecto software, puesto que deben proporcionar al usuario final garantías que le permitan depositar su confianza en el producto que va a utilizar, porque en caso contrario podrían surgir reacciones adversas al ser utilizado.

La calidad de cualquier aplicación se asocia al hecho de que durante su ejecución no se produzcan errores que perjudiquen al usuario. Por otro lado, la fiabilidad hace referencia a la capacidad del sistema para proporcionar datos reales, asegurando que las acciones que se van a realizar durante el proceso resultan correctas y se llevan a cabo de la mejor manera posible.

El uso de lenguajes muy estandarizados para el desarrollo de la aplicación y la utilización de framework permite garantizar una calidad alta y un desarrollo rápido. Si se encontrasen fallos serían debido a interrupciones en la conexión con el servidor. Además también se pueden producir fallos por la propia conexión a la red del dispositivo móvil que tenga la aplicación. La probabilidad de fallo se minimizará optimizando los elementos que componen la aplicación, así como realizando un plan de pruebas que verifique y valide la aplicación.

2.2.9. Programa de tareas

El programa de tareas se define como el conjunto de etapas y actividades que constituyen el proceso de desarrollo de una aplicación. El proyecto quedará dividido en las siguientes fases, las cuales nos ayudarán a organizar y llevar a cabo el proyecto de una manera ordenada. Las fases son las siguientes:

- *Fase de aprendizaje y familiarización de las tecnologías:* en cuanto a esta fase se va a estudiar y a realizar varias pruebas diferentes sobre la tecnología que se va a utilizar en el proyecto.
- *Fase de investigación y planteamiento:* Se va a investigar los antecedentes de nuestro proyecto, ya que nos servirá para implementar nuestro proyecto de forma más eficiente y satisfactoria y haremos un plan de ejecución.
- *Fase de desarrollo e implementación:* Realizaremos el desarrollo completo del software. Crearemos las funciones una vez tengamos nuestro diseño y estudio sobre las diferentes tecnologías que se va a utilizar. Además, comprobaremos el funcionamiento de los sistemas de almacenamiento en la nube con nuestro software de entorno web. Nuestro software debe cumplir las funcionalidades básicas de la aplicación de notas, además de la sincronización con la nube.
- *Fase de diseño de la interfaz:* Una vez superado los pasos anteriores pasaremos al diseño de nuestra interfaz. Ésta deberá ser fácil de utilizar para el usuario además de ser lo más óptima posible.
- *Fase de pruebas y verificación:* realizaremos varias pruebas para poder comprobar que el funcionamiento de nuestro software es correcto y evaluarlo, además, de su estabilidad. Con esto podremos encontrar posible errores que pueda contener. Para finalizar pasaremos a la fase de corrección de errores para que todo funcione correctamente.
- *Fase de documentación y diseño:* En esta fase se va a realizar, de manera continua durante la realización de nuestro proyecto, toda la documentación, de forma clara y sencilla, sobre el proceso de desarrollo del mismo además de su diseño. Para analizar y construir nuestro software utilizaremos la metodología de ingeniería de software.

Esta será la distribución temporal del proyecto:

<i>Tareas</i>	<i>Tiempo (h)</i>
Aprendizaje y familiarización de las tecnologías a usar en el proyecto	45
+ Django + Python	15
+ HTML + CSS	10
+ Phonegap	10
+ API Dropbox para Python	10
Investigación y planteamiento	40
+ Investigar nuevas técnicas y posibilidades	20
+ Plantear el software	20
Desarrollo e implementación	135
+ Diseño	45
- Diseño de la página web	25
- Diseño del entorno móvil	20
+ Programación de funcionalidades	90
- Desarrollo para entorno responsivo	20
- Desarrollo para la sincronización de notas	20
- Desarrollo de funcionalidades básicas para notas	20
- Sistema de etiquetado	15
- Configuración del usuario	15
Pruebas	40
Documentación	40
Total	300

Tabla 1: Distribución temporal del proyecto

2.2.10.Pruebas

Las pruebas se definen como el conjunto de acciones y datos que son utilizados para poder depurar la aplicación desarrollada, además también del medio para demostrar la funcionalidad y la utilidad de la aplicación.

Por tanto, se realizaran pruebas como por ejemplo: la comprobación de la carga que soportará la aplicación, conectando a 100 personas para comprobar si nuestro sistema lo soporta.

Los tipos de pruebas que se realizarán los veremos más adelante con mayor profundidad, pero serán pruebas de caja negra, caja blanca, etc.

2.2.11.Seguridad

Este apartado es muy importante dentro de la aplicación. Las comunicaciones entre usuario, como los datos personales de los mismos, deberían ir encriptados, y por una vía segura. En este tema se tendrá cuidado realizando unas condiciones de privacidad y de uso que favorezcan totalmente al cliente en el uso de los servicios de la aplicación. Una parte de la seguridad dependerá del servicio de almacenamiento que se va a utilizar. Por ello, se debe garantizar una comunicación segura con el medio de almacenamiento y garantizar la seguridad de las claves de acceso al medio.

3. Objetivos

El proyecto a desarrollar trata de la creación de una aplicación de notas para entorno web. Para el almacenamiento se usará el sistema en la nube de Dropbox, permitiendo así la sincronización de notas en distintos entornos web.

El objetivo principal es implementar una aplicación para notas en un entorno web que cumpla lo siguiente:

- Permitir la lectura de las notas en diferentes dispositivos mediante diseño responsive.
- Permitir que el usuario pueda crear notas, modificar o borrar notas desde cualquier dispositivo.
- Compatible con diferentes navegadores de sistemas como Linux, Android, IOS o Windows.
- Permitir la sincronización de las notas a través de al menos un sistema de almacenamiento en la nube, como Dropbox.
- Cada nota se almacenará en un archivo en formato XHTML Basic 1.1
- Opciones básicas de formato para el usuario a la hora de crear una nota (títulos, negrita, cursiva y listas).
- Se podrán archivar las notas en diferentes carpetas.
- Tendremos una barra para acceder a las diferentes funciones de la aplicación.
- La aplicación podrá ser utilizada por cualquier tipo de usuario ya que estará realizada de tal forma que su uso será sencillo para que el usuario pueda trabajar sin dificultades con ella.

4. Antecedentes

Actualmente podemos observar que existen varias posibilidades de aplicaciones para la utilización y gestión de notas, además de los métodos tradicionales de anotar en un simple papel. Se va a ver en este capítulo varias aplicaciones que podemos encontrar para la gestión de notas.

Las aplicaciones de notas que podemos encontrar a día de hoy tienen bastantes carencias. Tienen usos limitados por lo que muchas cosas son de pago y aun así no nos proporcionan con claridad los recursos que necesitamos, haciendo difícil su utilización por el usuario.

Nuestra aplicación pretende cubrir las principales carencias que poseen los sistemas que a continuación nombramos. Nos permite un uso más eficiente y fácil de manejar, también nos permite sincronizar notas entre todos los diferentes sistemas operativos mediante las principales plataformas de almacenamiento de la nube. Además, será software libre y permitirá a los usuarios el acceso a sus datos fácilmente, incluso sin la aplicación, al estar almacenada cada nota como un archivo en un sistema de ficheros de algún servicio que ya tienen y poderse ver las notas con un editor de texto sin formato.

4.1. Evernote

Es una aplicación informática cuyo objetivo es la organización de información personal mediante el archivo de notas. Existen versiones para sistemas operativos y una versión web. Evernote para web funciona mediante un servicio de suscripción y se sincroniza con las aplicaciones locales. En resumen, Evernote es muy completa pero sus servicios tienen una cuota para poder utilizarlos totalmente. [1]

4.2. Tomboy y Tomdroid

Tomboy es la aplicación de escritorio, esta no nos permite sincronizar notas entre versiones distintas y las notas no se pueden ver con un editor de texto plano. Además, Tomboy tiene versión de escritorio pero no posee una versión de entorno web para poderla utilizar en otros dispositivos móviles. Por otro lado, Tomdroid es la versión usada en Android, la cual lleva mucho tiempo sin ser actualizada y no nos permite editar las notas. [2][3]

4.3.Google Keep

Google Keep es una aplicación desarrollada por Google Inc. que permite organizar la información personal a través del archivo de notas. Está disponible para Android y como aplicación web en Google Drive. En resumen, Google Keep tiene ventaja en poner sincronizar las notas entre todos los dispositivos, pero su rendimiento es menor y puede llegar a funcionar mal en dispositivos antiguos. [4]

4.4.Dropbox Paper

Dropbox Paper es una aplicación desarrollada por Dropbox para crear y editar documentos en grupo de trabajo que tienen cuentas en Dropbox. Es un servicio de edición de texto colaborativo, es decir, cuando creemos un texto, se va a ir modificando y otras personas pueden editarlo al mismo tiempo. Para poder usar esta plataforma tienes que estar registrado en Dropbox. [5]

5. Restricciones

Se va a exponer todas las restricciones, o factores limitativos, que existen en el ámbito del diseño y que condicionan la elección de una u otra alternativa para la aplicación. Tenemos dos grupos para representarlos que veremos en los siguientes apartados.

5.1. Factores dato

En el desarrollo de este proyecto se van a considerar los factores dato que nombramos a continuación:

- Limitación de tiempo, contamos con 300 horas para la realización del proyecto. Además de poder tenerlo realizado para junio o septiembre, por la realización de la matrícula.
- Todas las plataformas y librerías escogidas son libres y de código abierto. Esto será para la programación y explotación de las herramientas, puesto que el proyecto va a hacer gratuito y no tiene ninguna inversión inicial.

5.2. Factores estratégicos

Estos representan variables de diseño que permiten que elijamos entre diferentes opciones. Según lo que elijamos podrá alterarse el proceso del desarrollo y el propio producto final obtenido.

Para la plataforma web, en la parte del backend, se ha decidido usar el framework de Django, el cual usa Python como lenguaje de programación. Para el frontend, es decir, las plantillas que darán lugar a la interfaz del usuario, se ha decidido usar HTML y CSS como estándar de páginas webs, para presentarle un diseño más atractivo y cómodo al usuario.

Además, utilizaremos Summernote [14], el cual es un simple WYSIWYG editor en Bootstrap y Summernote es una librería JavaScript que ayuda a crear editores WYSIWYG en línea. Ésta nos ayudara a modificar la información de las notas.

Para el entorno de desarrollo se ha escogido el sistema operativo Ubuntu, una distribución de Linux, que es un sistema gratuito, seguro y estable. Para la codificación se ha trabajado con el editor de texto de Ubuntu, tanto para el *backend* como para el *frontend*.

El código se encuentra alojado en un sistema de alojamiento llamado GitHub, que ofrece varias ventajas, la principal es que tiene un servicio de control de versiones, pudiendo registrar los cambios a la hora del desarrollo y volver a un estado anterior en cualquier momento, otra ventaja es que permite el desarrollo colaborativo, permitiendo a varios desarrolladores trabajar conjuntamente sobre el mismo repositorio de código.

Por otro lado, la aplicación ha sido alojada en un hosting llamado Dinahosting[17], así el usuario podrá hacer uso de la aplicación por medio de cualquier navegador. Además, usaremos Phonegap[16] para la creación de la Apk en Android[13].

6. Recursos

En este capítulo se presentan de forma clara los recursos humanos, software y hardware necesarios para la realización de este proyecto. Los recursos son definidos como aquellos medios de los que se dispone para abordar el proceso de desarrollo del proyecto.

6.1. Recursos humanos

Son aquellos que están constituidos por toda persona que intervenga en el proceso de desarrollo del sistema, por ello el equipo de trabajo encargado de la implementación de esta aplicación está formado por las siguientes personas:

- *Directores del proyecto:* Prof. Dr. Antonio Araúzo Azofra, Profesor del Área de Proyectos de Ingeniería del Departamento de Ingeniería Rural de la Universidad de Córdoba y Prof. Dra. María Luque Rodríguez, Profesora del departamento de Informática y Análisis Numérico, Universidad de Córdoba serán los encargados de la orientación al autor del mismo durante su proceso de desarrollo. Los directores se encargaran de supervisar las tareas de desarrollo para comprobar que los resultados obtenidos se corresponden con los requisitos planteados. Además, facilitarán aquellos recursos materiales que resulten necesarios para abordar el proceso de desarrollo exitosamente.
- *Autor del proyecto:* La autora, Pilar Arjona Galindo, alumna de Grado de Ingeniería Informática de la Universidad de Córdoba. Será la encargada de realizar la planificación inicial y posterior desarrollo del sistema software, así como de plantear y ejecutar las pruebas previstas para el mismo. por último, deberá generar la documentación del proyecto, que estará constituida por el Manual Técnico, el Manual de Usuario y el Manual de Código.

6.2. Recursos software

Conjunto de programas informáticos necesarios durante el proceso de desarrollo del proyecto, cuya ejecución se realiza mediante los recursos hardware disponibles.

El software necesario para el desarrollo de la aplicación es el siguiente:

- Framework de desarrollo web Django[6] - Es de código abierto y está escrito en Python [7](Versión 3.5.2) , este usa el patrón de diseño de Modelo-Vista-Controlador.
- Lenguaje de programación HTML5[8]
- Lenguaje de programación CSS[9]
- Aplicaciones Dropbox [10]- Para la comunicación de las plataformas.
- Phonegap[16] - Para la adaptación a móvil.

6.3.Recursos hardware

Conjunto de equipos informáticos utilizados durante el proceso de realización del proyecto.

El equipo en el cual se ha desarrollado la aplicación es un ordenador portátil Asus X53S con las siguientes características:

- Procesador Intel(R) Core(™) i5-2450M CPU @ 2.50GHz
- Memoria RAM de 6,00 GB
- Sistema operativo Ubuntu 16.04 de 64 bits

Además, también se utilizara para el desarrollo un dispositivo móvil Nexus 5 con las siguientes características:

- Procesador CPU: Qualcomm Snapdragon™ 800, 2,26 GHz , GPU:
- Adreno 330 de 450 MHz
- Memoria de 32 GB y Memoria RAM de 2 GB
- Sistema operativo Android 6.0.1

II. Análisis del sistema

7. Especificación de Requisitos

En este apartado se va a describir lo que debe hacer el sistema de forma técnica.

La especificación de requisitos (ERS) se apoya en los datos obtenidos durante su análisis. Se va a dar una mayor importancia a esta fase, ya que influye en gran parte en el desarrollo y en la calidad de un producto software.

En la ERS se definen los datos y toda la información que el software debe manejar y transformar. Además, se definirán los procesos encargados de la transformación y como esta deberá mostrarse al usuario.

Utilizaremos el lenguaje unificado UML [12] para realizar una especificación formal de los requisitos de la aplicación, aplicando plantillas para definir los casos de uso.

7.1.Requisitos Funcionales

Para la especificación de los requisitos funcionales se desarrollaran uno por uno recogidos de un análisis previo, después se utilizará un conjunto de diagramas, llamados casos de uso, apropiados para mostrar su comportamiento.

Finalmente realizaré la matriz de relación entre los requisitos previos y los diagramas de casos de uso.

7.1.1. Requisitos recogidos del análisis previo

En cuanto a la anotación de notas las funciones que pueden realizar los usuarios con la aplicación son las siguientes:

RF-1. El sistema debería permitir crear, visualizar y elegir carpetas para la organización de sus notas.

RF-2. El sistema debería permitir borrar carpetas para eliminar información que ya no es necesaria.

RF-3. El sistema debería permitir crear y listar notas para apuntar cualquier tipo de información.

RF-4. El sistema debería permitir borrar notas que ya no son necesarias.

RF-5. El sistema debería permitir editar notas para realizar cambios en ellas.

RF-6. El sistema debería permitir visualizar la información de las notas guardadas.

RF-7. El sistema debería permitir ver información sobre su usuario, además, de poder editarlo.

RF-8. El sistema debería permitir registrarse, iniciar y cerrar sesión dentro de la aplicación.

7.1.2. Diagramas de Casos de uso

7.1.2.1. Diagrama de casos de uso de Sistema de Autenticación

En este diagrama mostramos detalladamente los casos de uso que pertenecen a la parte de autenticación (inicio de sesión y registro) dentro de la aplicación.

Seguidamente podemos observar los escenarios pertinentes detallando cada caso de uso.

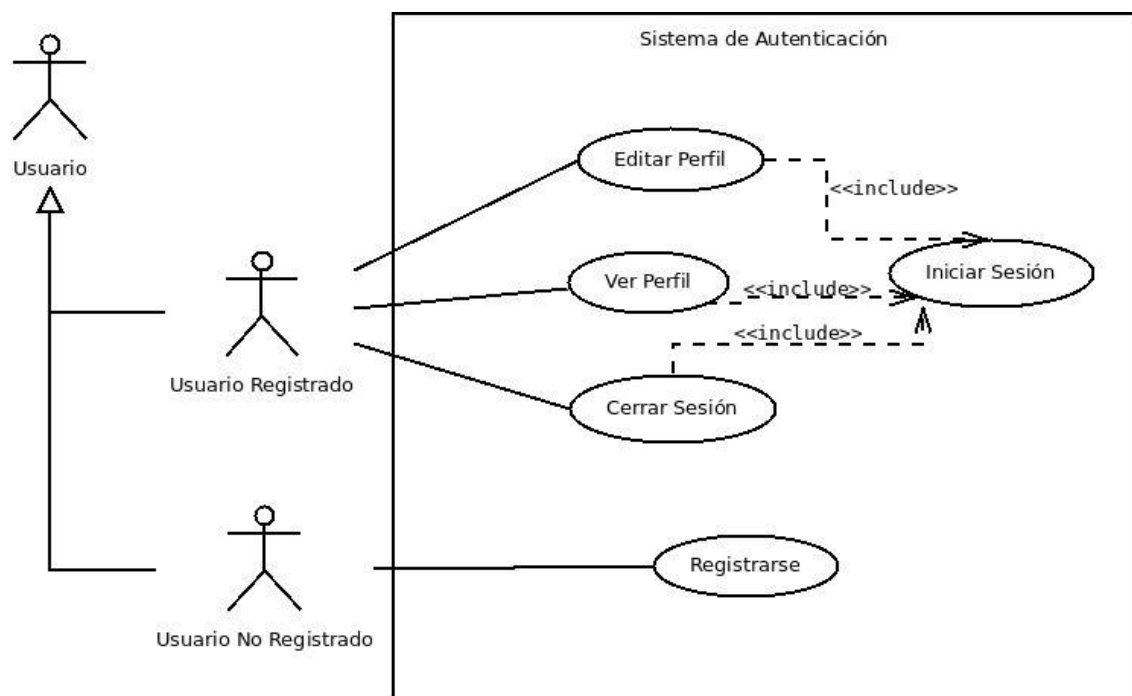


Figura 1: Diagrama del Sistema de Autenticación

Especificaciones de los Casos de Uso

Nombre del caso de uso	Registrarse
ID	1
Descripción	El usuario rellena los datos de registro y el sistema lo registra dentro de la base de datos de la aplicación.
Actor Principal	Usuario No Registrado
Actor Secundario	Ninguno
Precondición	Ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de registrarse. 2. El usuario rellena los datos correctamente en el formulario. 3. El sistema comprueba al usuario en la base de datos. 4. El sistema registra al usuario en la base de datos. 5. El sistema redirige al usuario a la página de iniciar sesión.
Post-Condición	El usuario queda registrado.
Flujo Alternativo	<p>3. El sistema comprueba al usuario en la base de datos.</p> <p>3.1. No se han completado los campos obligatorios: El sistema comprueba que el usuario no ha introducido los campos obligatorios y lanza un aviso.</p> <p>3.2. No se han completado los campos con los caracteres permitidos: El sistema comprueba que el usuario ha introducido caracteres no permitidos y lanza un aviso.</p>

Tabla 2: Especificación del caso de uso Registrarse

Nombre del caso de uso	Iniciar Sesión
ID	2
Descripción	El usuario rellena los datos de iniciar sesión y el sistema le abre una sesión en el servidor.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haberse registrado con anterioridad.

Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de iniciar sesión. 2. El usuario inserta los datos en el formulario. 3. El sistema comprueba al usuario en la base de datos. 4. El sistema autentica e inicia la sesión del usuario.
Post-Condición	El usuario esta dentro de su sesión.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema comprueba al usuario en la base de datos. <ol style="list-style-type: none"> 3.1. No se han completado los campos obligatorios: El sistema comprueba que el usuario no ha introducido los campos obligatorios y lanza un aviso. 4. El sistema autentica e inicia la sesión del usuario. <ol style="list-style-type: none"> 4.1. Los datos introducidos no son correctos: Si se comprueba en la base de datos que los datos no están registrados se lanza un error avisando al usuario. 5. El sistema comprueba si el cliente tiene un token en su usuario <ol style="list-style-type: none"> 5.1. Si el cliente tiene token, se conectará a Dropbox y podrá hacer uso de la aplicación. 5.2. Si el cliente no tiene token, deberá generar uno para poder usar la aplicación.

Tabla 3: Especificación del caso de uso Iniciar Sesión

Nombre del caso de uso	Cerrar Sesión
ID	3
Descripción	El usuario cierra la sesión que tiene iniciada.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe tener una sesión iniciada para poder cerrarla.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de cerrar sesión. 2. El sistema cierra la sesión del usuario.
Post-Condición	El usuario esta fuera de su sesión.

Flujo Alternativo	Ninguno
--------------------------	---------

Tabla 4: Especificación del caso de uso Cerrar Sesión

Nombre del caso de uso	Editar Perfil
ID	4
Descripción	El usuario modifica los datos de su perfil
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	1. El usuario debe tener la sesión iniciada.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de editar perfil. 2. El usuario modifica los datos en el formulario. 3. El usuario accede al link para la obtención de un token. 4. El usuario introduce el token en los datos del formulario. 5. El sistema actualiza los datos nuevos en la base de datos.
Post-Condición	Ninguno
Flujo Alternativo	<ol style="list-style-type: none"> 2. El sistema actualiza los datos nuevos en la base de datos. <ol style="list-style-type: none"> 2.1. No se han completado los campos obligatorios: El sistema comprueba que el usuario no ha introducido los campos obligatorios y lanza un aviso. 2. El sistema conectará con Dropbox al introducir un token. <ol style="list-style-type: none"> 2.1. Si el token es aceptado: El usuario no deberá volver a generar un token. 2.2. Si el token no es aceptado: El usuario deberá volver a generar un token.

Tabla 5: Especificación del caso de uso Editar Perfil

Nombre del caso de uso	Ver Perfil
ID	5
Descripción	El usuario visualiza los datos de su perfil
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	1. El usuario debe tener la sesión iniciada.
Flujo Principal	1. El usuario presiona el botón de Mi Perfil. 2. El sistema muestra los datos del usuario.
Post-Condición	Ninguno
Flujo Alternativo	Ninguno

Tabla 6: Especificación del caso de uso Ver Perfil

7.1.2.2. Diagrama de casos de uso de Sistema de Carpetas

En este diagrama mostramos detalladamente los casos de uso que pertenecen a la parte relacionada con las carpetas dentro de la aplicación.

Seguidamente podemos observar los escenarios pertinentes detallando cada caso de uso.

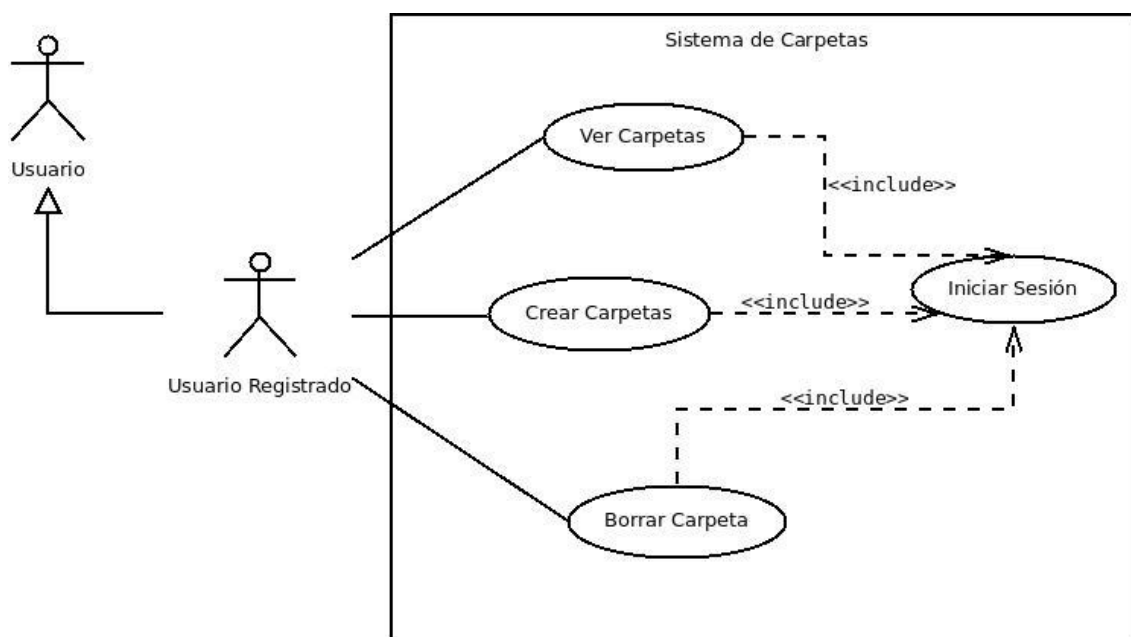


Figura 2: Diagrama del Sistema de Carpetas

Especificaciones de los Casos de Uso

Nombre del caso de uso	Crear Carpeta
ID	6
Descripción	El usuario crea una carpeta para almacenar notas.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	2. El usuario debe tener la sesión iniciada.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de Mis Carpetas. 2. El usuario rellena el formulario para añadir una nueva carpeta. 3. El sistema creará la nueva carpeta.
Post-Condición	Se ha creado una nueva carpeta.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema creará la nueva carpeta. <ol style="list-style-type: none"> 3.1. El nombre introducido para la nueva carpeta ya existe: El sistema comprueba que la carpeta no exista, en caso de existir manda un aviso.

Tabla 7: Especificación del caso de uso Crear Carpeta

Nombre del caso de uso	Ver Carpetas
ID	7
Descripción	El usuario visualiza las carpetas existentes.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	1. El usuario debe tener la sesión iniciada.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de Mis Carpetas. 2. El sistema mostrará las carpetas que contiene la cuenta del usuario.
Post-Condición	Ninguno
Flujo Alternativo	Ninguno

Tabla 8: Especificación del caso de uso Ver Carpetas

Nombre del caso de uso	Borrar Carpeta
ID	8
Descripción	El usuario visualiza las carpetas existentes.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno

Precondición	1. El usuario debe tener la sesión iniciada.
Flujo Principal	1. El usuario presiona el botón de Mis Carpetas. 2. El sistema mostrara las carpetas que contiene la cuenta del usuario. 3. El usuario presionara el botón de Borrar, este se encontrará justo al lado de la carpeta que desee borrar. 4. El sistema borrará la carpeta y notificará al usuario.
Post-Condición	La carpeta queda borrada.
Flujo Alternativo	Ninguno

Tabla 9: Especificación del caso de uso Borrar Carpeta

7.1.2.3. Diagrama de casos de uso de Sistema de Nota

En este diagrama mostramos detalladamente los casos de uso que pertenecen a la parte relacionada con las notas dentro de la aplicación.

Seguidamente podemos observar los escenarios pertinentes detallando cada caso de uso.

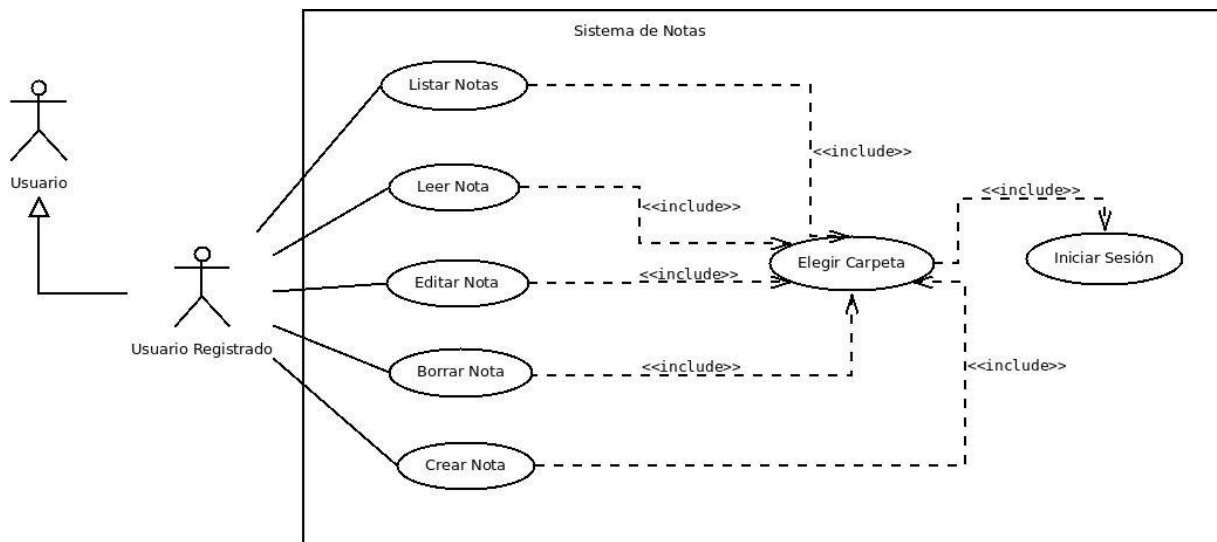


Figura 3: Diagrama del Sistema de Notas

Especificaciones de los Casos de Uso

Nombre del caso de uso	Elegir Carpeta
ID	9
Descripción	El usuario elige una carpeta para guardar la nueva nota.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	1. El usuario debe tener la sesión iniciada.
Flujo Principal	1. El usuario presiona el botón Crear Notas 2. El sistema muestra al usuario todas las carpetas que posee, para elegir en donde quiere guardar la nueva nota. 3. El usuario elige una carpeta.
Post-Condición	1. La carpeta elegida queda seleccionada
Flujo Alternativo	Ninguno

Tabla 10: Especificación del caso de uso Elegir Carpeta

Nombre del caso de uso	Crear Nota
ID	10
Descripción	El usuario crea una nueva nota.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	1. El usuario debe tener la sesión iniciada. 2. El usuario debe haber elegido carpeta (CU-9).
Flujo Principal	1. El usuario rellena el formulario para crear una nueva nota. 2. El sistema guarda la nueva nota.
Post-Condición	La nueva nota queda guardada en la carpeta elegida.
Flujo Alternativo	2. El sistema guarda la nueva nota. 2.1. El nombre introducido para la nueva nota ya existe: El sistema comprueba que la nota no exista, en caso de existir manda un aviso.

Tabla 11: Especificación del caso de uso Crear Nota

Nombre del caso de uso	Borrar Nota
ID	11
Descripción	El usuario borra una nota existente.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe tener la sesión iniciada. 2. El usuario debe haber elegido carpeta (CU-9).
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de borrar, que se encuentra justo al lado de la nota que desee eliminar. 2. El sistema elimina la nota y notifica al usuario.
Post-Condición	La nota queda borrada.
Flujo Alternativo	Ninguno

Tabla 12: Especificación del caso de uso Borrar Nota

Nombre del caso de uso	Listar Notas
ID	12
Descripción	El usuario visualiza todas sus notas existentes.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe tener la sesión iniciada. 2. El usuario debe haber elegido carpeta (CU-9).
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema muestra una lista de todas las notas existentes al usuario.
Post-Condición	Ninguno
Flujo Alternativo	Ninguno

Tabla 13: Especificación del caso de uso Listar Notas

Nombre del caso de uso	Leer Nota
ID	13
Descripción	El usuario lee el contenido de una nota.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe tener la sesión iniciada. 2. El usuario debe haber elegido carpeta (CU-9).
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema muestra una lista de todas las notas existentes al usuario. 2. El usuario presiona el nombre de la nota que desea leer. 3. El sistema muestra el contenido de la nota al usuario.
Post-Condición	Ninguno
Flujo Alternativo	Ninguno

Tabla 14: Especificación del caso de uso Leer Nota

Nombre del caso de uso	Editar Nota
ID	14
Descripción	El usuario edita el contenido de una nota.
Actor Principal	Usuario Registrado
Actor Secundario	Ninguno
Precondición	<ol style="list-style-type: none"> 1. El usuario debe tener la sesión iniciada. 2. El usuario debe haber elegido carpeta (CU-9).
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema muestra una lista de todas las notas existentes al usuario. 2. El usuario presiona el botón de editar nota. 3. El sistema muestra el editor de notas. 4. El usuario rellena la nota con las nuevas modificaciones. 5. El sistema guarda los cambios realizados.
Post-Condición	La nota queda modificada.
Flujo Alternativo	Ninguno

Tabla 15: Especificación del caso de uso Editar Nota

7.1.2.4. Relación requisitos de Usuario con casos de Uso

CU RU	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						X	X		X					
2								X						
3										X		X		
4											X			
5														X
6													X	
7				X	X									
8	X	X	X											

Tabla 16: Relación Requisitos Análisis previo con Casos de Uso

7.2.Requisitos No Funcionales

7.2.1. Requisitos del Producto

7.2.1.1. Usabilidad

RNF-1. La experiencia del usuario debe ser satisfactoria en todos los factores tanto en el manejo de la aplicación como los visuales.

RNF-2. El software será usado por cualquier tipo de persona.

RNF-3. La interfaz deberá ser clara y concisa, lo que supone gran facilidad a la hora de usar la aplicación para todo tipo de personas.

7.2.1.2. Funcionalidad

RNF-4. Se propone unas funcionalidades básicas con buenos resultados para que los usuarios no tengan ningún problema a la hora de utilizar la aplicación en cualquier momento que deseen.

7.2.1.3. Rendimiento y Escalabilidad

RNF-5. El rendimiento se verá condicionado al tipo de Smartphone y tipo de conexión del mismo.

RNF-6. El sistema debe ser rápido y el tiempo de respuesta debe ser el mínimo posible.

7.2.1.4. Mantenimiento

RNF-7. El usuario puede contactar fácilmente con el desarrollador para informar de posibles fallos o mejoras.

RNF-8. La aplicación ofrecerá actualizaciones periódicas a lo largo del tiempo, con las mejoras introducidas durante ese tiempo.

7.2.1.5. Seguridad

RNF-9. El acceso y uso de la aplicación ofrece unas condiciones de privacidad y de uso que el usuario debe aceptar para optar al uso de la aplicación.

7.2.1.6. Confiabilidad

RNF-10. El servicio que ofrece la aplicación estará totalmente disponible desde el momento en que se suba a las distintas tiendas de descarga de las plataformas móviles, teniendo una continuidad en el tiempo mínima de un año, son la deseable posibilidad de aumentar este tiempo dependiendo de los usuarios.

7.2.1.7. Portabilidad

RNF-11. La aplicación será totalmente portable. Siempre que el usuario posea acceso a internet y un Smartphone podrá tener la posibilidad de utilizar la aplicación donde lo desee.

7.2.2. Requisitos Organizacionales

7.2.2.1. Software

RNF-14. Las plataformas que podrán utilizar esta aplicación serán todas aquellas que tenga acceso a un navegador como mínimo.

7.2.2.2. Hardware

RNF-15. Los dispositivos deben disponer de espacio suficiente para la instalación de la aplicación y de conexión a internet para el funcionamiento de la misma.

7.2.3. Requisitos externos

7.2.3.1. Estándares

RNF-16. La aplicación deberá cumplir con los estándares marcados por el www.Consortium (HTML, CSS, etc.)[8][9]

RNF-17. La licencia de uso de software donde se aloja y con el que se realiza la aplicación debe ser lo menos restrictiva posible, totalmente de código abierto.

8. Especificación del Sistema

8.1. Especificación de la Información – Modelo Entidad-Relación

En los apartados anteriores hemos visto la definición del problema, además de los requisitos y casos de uso, estos nos ofrecen toda la información necesaria para definir un conjunto de elementos, los tipos de entidad, en los que se almacenará la información que se maneja en la aplicación.

Una vez que se analiza las necesidades de la aplicación se identifican una serie de entidades que englobaran la base de datos. A continuación, se realizara una breve descripción inicial de estas entidades que nos van a servir para identificar el correcto mantenimiento de la información que tenemos del sistema.

El modelo realizado para la representación es un modelo Entidad/Relación. Los diferentes tipos de entidades que forman parte del sistema llevaran una estructura compuesta por los siguientes apartados.

- *Diagrama*: Representación grafica del tipo de entidad.
- *Descripción*: Definición detallada del tipo de entidad.
- *Restricciones*: Posibles dependencias que posee el tipo de entidad con respecto con otros tipos de entidades, además también se detallara los factores que delimitan al tipo de entidad.
- *Características*: Descripción de los datos y características que posee el tipo de entidad, siguiendo la estructura:
 - ❖ Nombre de entidad
 - ❖ Identificador principal
 - ❖ Identificador alternativo
 - ❖ Numero de atributos
- *Atributos*: Breve descripción de todos los atributos propios que posee el tipo de entidad, los cuales tendrán la siguiente estructura:
 - ❖ Definición
 - ❖ Tipo
 - ❖ Restricciones

En la nomenclatura utilizada, los atributos claves de las entidades van a ser representados con un círculo negro y las claves foráneas con un círculo gris.

8.1.1. Entidades y Jerarquías

8.1.1.1. Entidad User

Tipo de entidad que representa el objeto del mundo real User, el cual almacena los datos principales de los usuarios, los cuales hacen falta para poder iniciar sesión en la aplicación.

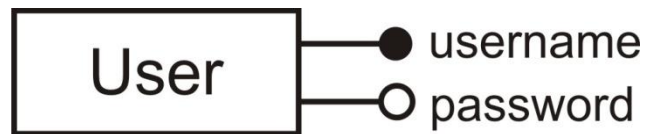


Figura 4: Entidad User

Nombre	Descripción	Obligatorio	Tamaño
Username	Nombre del usuario	X	100 Caracteres
Password	Contraseña del usuario	X	100 Caracteres

Tabla 17: Entidad User

8.1.1.2. Entidad Persona

Tipo de entidad que representa el objeto del mundo real Persona, el cual almacena más información sobre los usuarios.

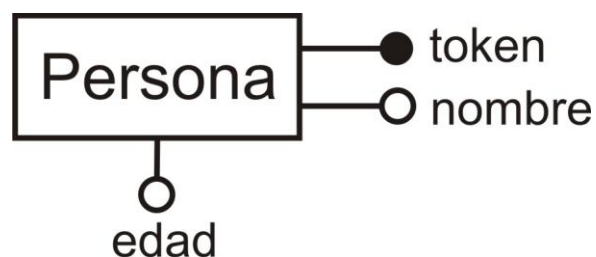


Figura 5: Entidad Persona

Nombre	Descripción	Obligatorio	Tamaño
token	Clave de acceso al Dropbox	X	50 Caracteres
nombre	Nick del usuario		100 Caracteres
edad	Edad del usuario		Entero

Tabla 18: Entidad Persona

8.1.2. Relaciones entre las entidades del sistema

Las entidades detalladas en el punto anterior se encuentran relacionadas entre sí de la siguiente forma.

8.1.2.1. Relación User-Persona

Relaciona los tipos de entidad User y Persona bajo un cardinalidad 1:1, tal y como se muestra en el siguiente diagrama. Se considera que un User tiene una persona, participando el tipo de entidad Persona con cardinalidades (1,1). Por otro lado, una Persona pertenece a un User, por lo que el tipo de entidad User participa con cardinalidad (1,1).

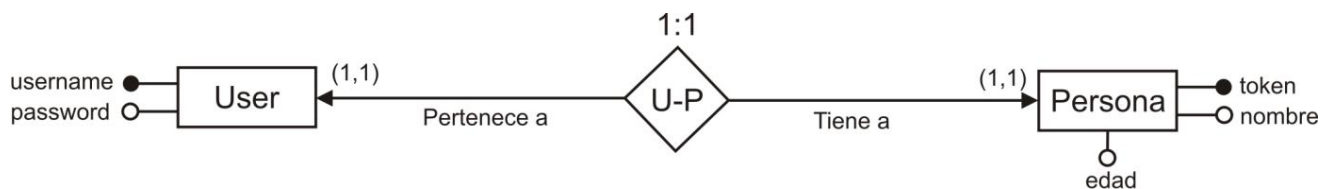


Figura 6: Relación User-Persona

8.1.3. Modelo Entidad-Relación

A continuación en la figura 7, se presenta el modelo Entidad-Relación completo donde se pueden apreciar cada uno de los tipos de entidad con sus respectivas relaciones explicadas anteriormente, podemos apreciarlo en la figura 7.

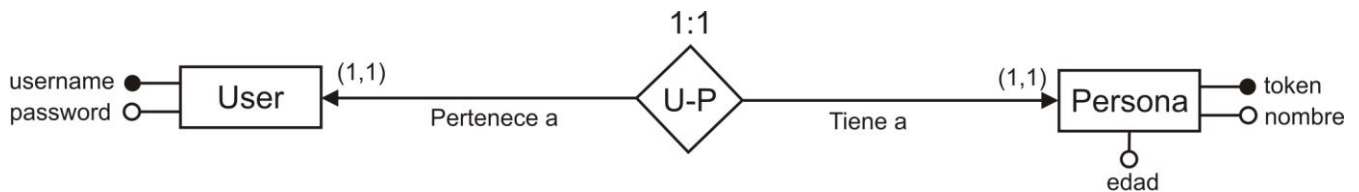


Figura 7: Diagrama Entidad-Relación

8.2. Modelo de datos Relacional

Una de las partes más importantes del sistema puesto que tiene un impacto en la calidad del software a desarrollar, el diseño de datos influirá en la estructura del programa y en su complejidad procedimental.

Para la realización de usara como base el diagrama entidad-relación, desarrollado en el punto anterior, que representa el modelo conceptual de datos. Después de definir la forma completa de los tipos de entidad y relaciones que se pueden ver en el modelo conceptual, tenemos una serie de reglas fáciles de entender y de aplicar para el proceso de traducción de esquemas conceptuales, realizados mediante el modelo E-R a esquemas lógicos basados en el modelo relacional. Estas reglas transformaran los esquemas en objetos pertenecientes a los esquemas lógicos.

Los tipos de entidades se transformaran en tablas y los tipos de relación en relaciones. Además como paso previo a la aplicación de las reglas de transformación de esquemas conceptuales a esquemas relacionales, es favorable la preparación de los esquemas conceptuales mediante la aplicación de unas reglas preparatorias, que garanticen y faciliten la fiabilidad del proceso. Estas reglas se aplicaran en los distintos tipos de entidad, algunos atributos múltiples y/o algún atributo compuesto, o también para eliminar relaciones jerárquicas y otros elementos del esquema conceptual que no dispongan de un mecanismo de representación conveniente en la traducción a su representación bajo el modelo relacional.

Observando el estudio anterior o el modelo conceptual nos damos cuenta que no hay ningún atributo múltiple o compuesto, ni relaciones jerárquicas.

A continuación, todas las entidades presentes en el esquema conceptual se transformaran en tablas manteniendo el número y tipo de atributos. En cuanto a las relaciones entre entidades, todas mantendrán su cardinalidad.

En la nomenclatura utilizada, los atributos claves de la tablas se representaran subrayados y las claves foráneas en negrita.

8.2.1. Tablas del sistema

8.2.1.1. Tabla User

La tabla User se forma a partir del tipo de entidad que lleva su mismo nombre, tomando de ella los atributos username y password.

El identificador de esta tabla es el atributo username. No se considera ninguna clave alterna. Tenemos como clave foránea el atributo token.

User (username, password, **token**)

R2(User).token->**R1**(Persona).token

8.2.1.2. Tabla Persona

La tabla Persona se forma a partir del tipo de entidad que lleva su mismo nombre, tomando de ella los atributos token, nombre y edad.

El identificador de esta tabla es el atributo token. No se considera ninguna clave alterna. Tenemos como clave foránea el atributo username.

Persona (token, nombre, edad, **username**)

R2(Persona).username->**R1**(User).username

8.2.2. Esquema Relacional

A continuación, se presenta el esquema relacional completo a partir de los datos de las tablas anteriores, en la tabla 19.

Nombre de la tabla	Atributos
User	<u>Username</u> , password, token
Persona	<u>Token</u> , nombre, edad, username

Tabla 19: Esquema Relacional

8.2.3. Diagrama Relacional

A continuación en la figura 8, se presenta el modelo relacional completo donde se pueden apreciar todas las tablas y relaciones, previo paso de realizar el código de la base de datos.

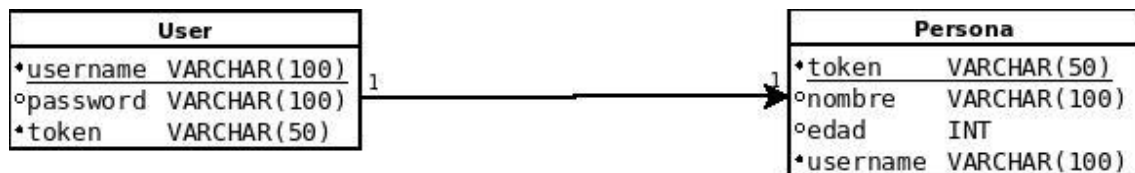


Figura 8: Diagrama Relacional

III. Diseño de la aplicación

9. Diseño del sistema

En este apartado se va a definir la arquitectura general del proyecto, especificando las partes que lo componen y la relación entre ellas.

9.1. Arquitectura del Sistema

El desarrollo se va a centrar en tres partes fundamentales:

- La parte del servidor, que va a ser totalmente transparente para el usuario.
- La parte del cliente, que se encargará tanto de la interfaz gráfica como de la funcionalidad ofrecida al usuario.
- La parte donde se va a controlar la lógica de la aplicación relacionada con la gestión de notas personales.

A continuación se detallará las tres partes en las que se va a organizar la aplicación.

9.1.1. Servidor Web

En esta parte se llevará a cabo el funcionamiento del servidor web, es decir, el *backend*. Para ello se va a utilizar Django, un framework web de código abierto escrito en Python que permite desarrollar aplicaciones webs de forma rápida y sencilla.

Una de las características de Django es que sigue el patrón de diseño MVC (Modelo-Vista-Controlador), donde se distinguen los siguientes módulos:

- El Modelo hace referencia a la base de datos.
- La vista define los datos que se van a mostrar al usuario y cómo se van a mostrar, a través de plantillas.
- El controlador es el que atiende a las peticiones del usuario y ejecuta la acción requerida, permitiendo modificar el modelo.

La vista es lo que se va a corresponder al Cliente Web y el Controlador será la parte funcional para gestionar las notas personales.

9.1.2. Cliente Web

Es la parte dedicada a la interfaz grafica del usuario donde se muestran las funcionalidades que se le ofrece, conocido también como *frontend*. En Django corresponde al módulo de Vista y se implementa a través de plantillas HTML5, CSS para darle estilos de formato a las plantillas y JavaScript para que los datos mostrados tengan un comportamiento dinámico.

La aplicación web generará una interfaz sencilla y atractiva para el usuario, en la que no solo podrá organizar todas sus notas en carpetas, sino que, además, tendrá la posibilidad de modificarlas y actualizarlas posteriormente.

Por otro lado, se usará Summernote, un editor WYSIWYG, para facilitar la modificación de notas al usuario, ya que gracias al editor el usuario tendrá varias opciones a la hora de modificar la información de cada nota.

9.1.3. Gestor de notas

En lo que respecta a esta parte, pertenece al servidor y será la más importante de la aplicación. En ella, se van a desarrollar las funciones para que el sistema se sincronice con Dropbox y nos permita realizar múltiples tareas; por un lado, la creación, listado y borrado de carpetas y notas; y por otro lado, la visualización y edición de notas.

9.2. Gestor de notas

La implementación del proyecto se va a estructurar en directorios. Como podemos ver en la figura 9:

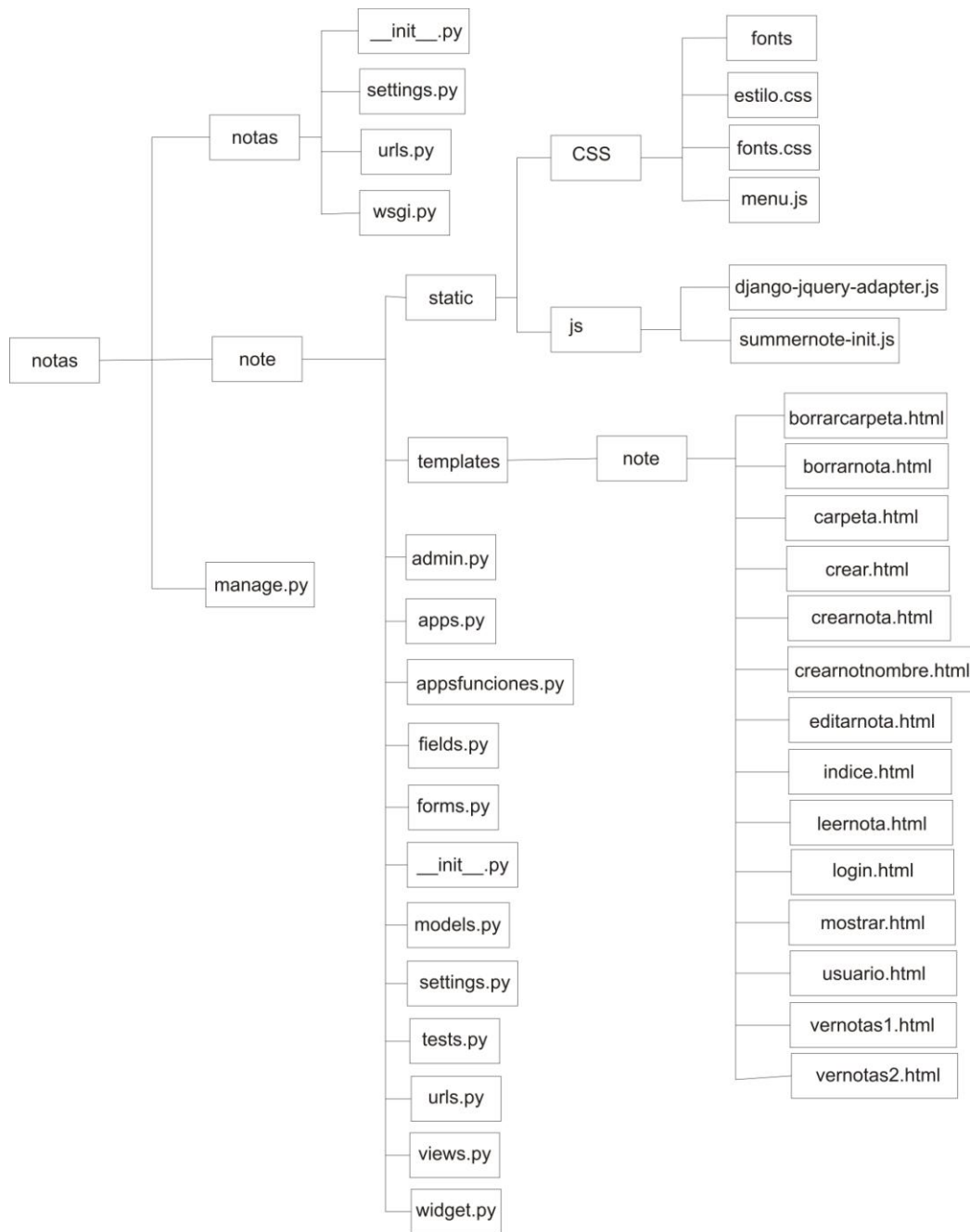


Figura 9: Estructura de ficheros.

El directorio **notas** es el directorio raíz que es el que va a contener el proyecto. De dicho directorio, se va a ver algunos de los siguientes ficheros que cuelgan de él.

- **notas:** es el directorio interno que contiene el paquete Python para el proyecto. El nombre de este directorio es el que se usará para cuando se quiera importar algo de este proyecto.
 - **__init__.py:** es un fichero vacío que indica a Python que el directorio tiene que ser considerado como un paquete de Python.

- **settings.py**: es un fichero de configuración del proyecto Django.
- **urls.py**: es un fichero donde se declaran las direcciones URL del proyecto Django.
- **wsgi.py**: es un fichero de configuración que sirve como punto de entrada para servidores web compatibles con WSGI (Web Server Gateway Interface).
- **note**: en Django se pueden desarrollar diferentes aplicaciones dentro del mismo proyecto, en este caso se va a crear una única aplicación llamada note y desde este directorio cuelgan todos los archivos necesarios para implementar la aplicación.
 - **static**: en este directorio es donde se declaran los ficheros estáticos, en este caso los ficheros que van a definir las vistas para cada bloque HTML.
 - **css**: este directorio contiene los ficheros que dan estilos a las paginas HTML.
 - **estilo.css**: este fichero contiene el código CSS personalizado para dar estilos a las paginas.
 - **menu.js**: este fichero contiene el código para la aparición y desaparición del menú.
 - **js**: este directorio contiene los ficheros con el código JavaScript.
 - **summernote-init.js**: en este fichero se encuentra parte del código que hace que funcione el editor de notas.
 - **templates**: en este directorio es donde se declaran los ficheros que sirven a Django como plantillas.
 - **borrarcarpeta.html**: fichero con la plantilla HTML de borrar una carpeta.
 - **borrarnota.html**: fichero con la plantilla HTML de borrar una nota.
 - **carpeta.html**: fichero con la plantilla HTML de mostrar las carpetas existentes y crear nuevas carpetas.
 - **crear.html**: fichero con la plantilla HTML para editar los datos de usuario.
 - **crearnota.html**: fichero con la plantilla HTML para crear notas en donde se elige la carpeta.
 - **crearnotnombre.html**: fichero con la plantilla HTML para crear notas en donde se elige el nombre.
 - **editarnota.html**: fichero con la plantilla HTML para editar una nota.
 - **indice.html**: fichero con la plantilla HTML inicial, es decir, la página web principal.
 - **leernota.html**: fichero con la plantilla HTML para leer la información de una nota.

- **login.html**: fichero con la plantilla HTML para iniciar sesión.
 - **mostrar.html**: fichero con la plantilla HTML en donde se muestra el perfil del usuario.
 - **usuario.html**: fichero con la plantilla HTML para registrar un nuevo usuario.
 - **vernotas1.html**: fichero con la plantilla HTML para ver notas, en donde primero se elige una carpeta.
 - **vernotas2.html**: fichero con la plantilla HTML para ver notas una vez elegida la carpeta.
-
- **apps.py**: en este fichero se encuentra nuestra clase Carpeta.
 - **appsfunciones.py**: en este fichero se encuentran las funciones necesarias para realizar las tareas de la aplicación.
 - **fields.py**: en este fichero se encuentra mas parte del código para el funcionamiento del editor de notas.
 - **forms.py**: en este fichero se encuentran los formularios.
 - **models.py**: en este fichero se encuentra el modelo de nuestra aplicación.
 - **views.py**: archivo para definir el controlador en Django (backend) del modelo MVC.
 - **widget.py**: este fichero contiene más código para nuestro editor de notas.
-
- **manage.py**: es un fichero que permite interactuar en línea de comandos con el proyecto de Django.

9.3. Backend

Aquí es donde se va a implementar la funcionalidad que corresponde al servidor web y al gestor de notas. Se implementara las siguientes tareas:

- Responder a las peticiones que hacen los usuarios mediante la aplicación web.
- Crear y borrar carpetas que pida el usuario.
- Crear y borrar notas que pida el usuario.
- Posibilidad de editar notas ya existentes.
- Sincronizar los archivos con Dropbox.

Para ofrecer estas funcionalidades al cliente web (*frontend*) se va a usar el framework de Django.

Django usa el patrón de diseño MVC (Modelo-Vista-Controlador), para esta aplicación necesitaremos un modelo para guardar los datos del usuario. La parte de la Vista correspondería a la plantillas HTML. Por último, la parte Controlador será la parte más importante, es donde se va a implementar la funcionalidad principal del proyecto para gestionar las notas personales.

9.3.1. Rutas

Para que el backend atienda a las peticiones que reciba del usuario a través del frontend, se usará el módulo de rutas de Django, que permite hacer un mapeo de URLs. Este mapeo permite pasar cualquier variables y devolver a la Vista algún dato, en este caso al frontend.

Además, el mapeo de URLs permite realizar diferentes tipos de peticiones, como:

- **GET:** Para obtener un recurso del servidor.
- **POST:** Para crear un recurso del servidor.
- **PUT:** Para actualizar un recurso del servidor.
- **DELETE:** Para eliminar un recurso del servidor.

A continuación en la tabla 20 detallaremos las rutas URLs que mapearán el servidor, así como los parámetros que recibe y la información que deberá devolver al final.

Ruta	Entrada	Salida
/nuevoUser	Información del usuario	Registro de un usuario nuevo en la base de datos.
/login	Información del usuario	Autenticación a un usuario para poder acceder a la aplicación.
/mostrar	Información del usuario	Se muestra información del usuario registrado.
/crear	Nueva información del usuario	Modificación de los datos del usuario.
/carpeta	Información de las carpetas existentes, y información para crear nuevas	Muestra un listado de carpetas existentes y permite la creación de carpetas nuevas.

/crearnota	Listado de carpetas	Muestra un listado de carpetas a elegir.
/crearnotnombre	Información para crear nota y carpeta seleccionada	Crea una nueva nota.
/listarcarpetaN	Listado de carpetas	Elección de una carpeta.
/listarnota	Carpeta elegida	Muestra un listado de notas.
/leernota	Carpeta elegida y nota	Muestra el contenido de la nota.
/editarnota	Carpeta elegida y nota	Muestra un editor para modificar la nota.
/borrarnota	Nota elegida	Queda borrada la nota.
/borrarcarpeta	Carpeta elegida	Queda borrada la carpeta.

Tabla 20: Mapeo de URLs en Django

9.4. Frontend

El frontend definirá la interfaz grafica con la que interactuará el usuario a través de una página web, en ella es dónde se implementará la funcionalidad para realizar peticiones al servidor (*backend*). Se implementará las siguientes tareas:

- Mostrar los datos de usuario.
- Mostrar las carpetas que tiene creadas el usuario.
- Mostrar las notas que tiene creadas el usuario.
- Mostrar la información que contiene las notas.

Para implementar la parte del *frontend* se va a usar HTML5, junto con CSS.

9.4.1. Rutas

En este caso, se va a utilizar el mapeo de URLs para gestionar las peticiones y mostrar las vistas correspondientes al usuario, por lo que se va a trabajar con rutas para atender a las diferentes peticiones.

A continuación en la tabla 21 mostramos las rutas que se van a implementar:

Ruta	Vista	Descripción
/persona	Usuario.html	Se muestra un formulario para que un usuario nuevo se registre.
/entrar	Login.html	Se muestra un formulario para iniciar sesión y poder acceder a la aplicación.
/perfil	Mostrar.html	Permite mostrar información del usuario registrado.
/crea	Crear.html	Permite modificar los datos del usuario.
/carpeta	Carpeta.html	Muestra un listado de carpetas existentes y permite la creación de carpetas nuevas.
/crearnota	Crearnota.html	Nos muestra un listado de carpetas a elegir.
/crearnotnombre	Crearnotnombre.html	Permite rellenar el formulario de creación de notas.
/listarcarpetaN	Vernotas1.html	Nos muestra un listado de carpetas a elegir.
/listarnota/..	Vernotas2.html	Nos muestra un listado de notas.
/leernota/./..	Leernota.html	Permite visualizar el contenido de una nota.
/editarnota/./..	Editarnota.html	Permite editar notas.
/borrarnota/./..	Borrarnota.html	Permite borrar una nota.
/borrarcarpeta/..	Borrarcarpeta.html	Permite borrar una carpeta.

Tabla 21: Mapeo de URLs en Django

9.5. Diseño procedimental

En este apartado se va a detallar algunas de las actividades y operaciones necesarias que realiza la aplicación.

Para ello se especificará los flujos y algoritmos con la ayuda de diagramas de UML.

9.5.1. Sincronización con Dropbox

En la Figura 10 se especifica mediante un diagrama de secuencias la sincronización que se realiza entre la aplicación y Dropbox.

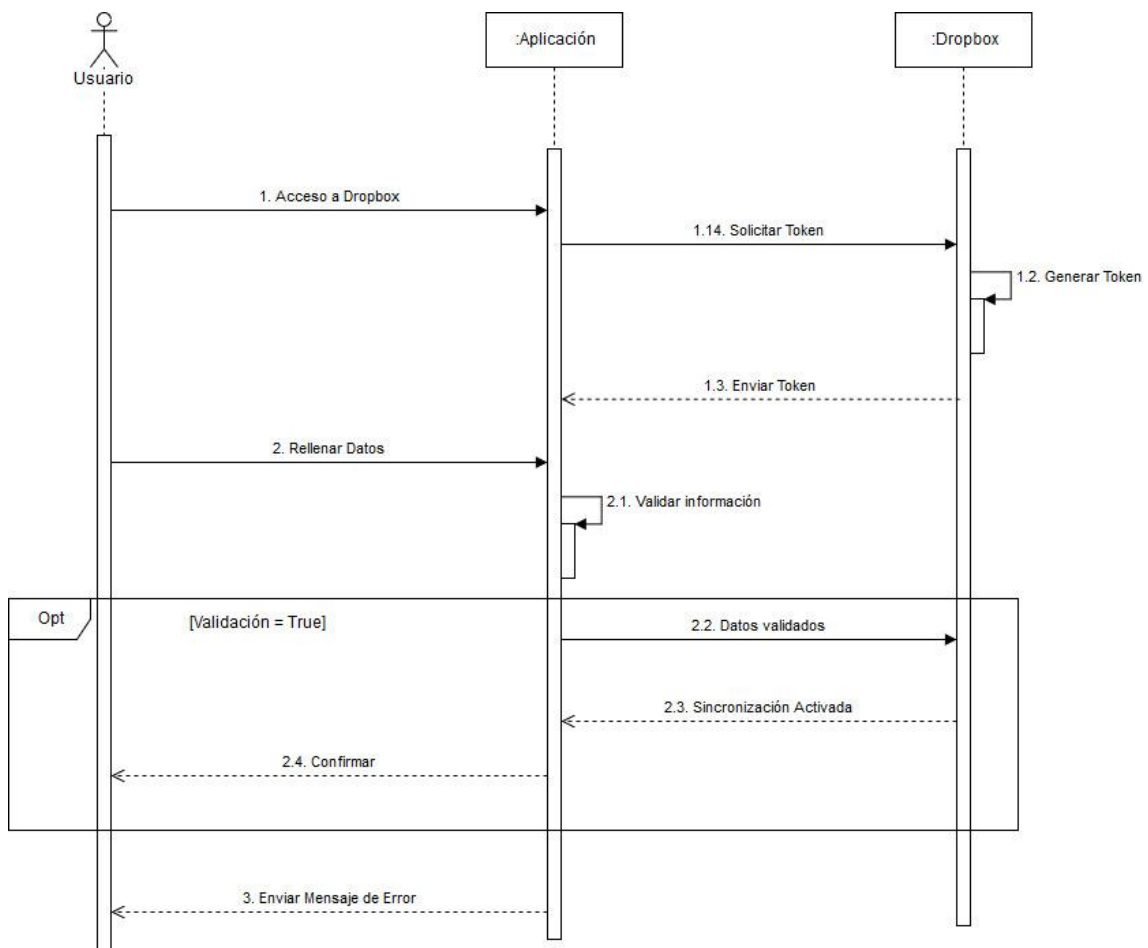


Figura 10: Diagrama de secuencias: Sincronización con Dropbox

9.5.2. Crear notas

En la Figura 11 se muestra un diagrama de secuencias, donde encontramos los pasos que se siguen para realizar la creación de una nueva nota en la aplicación y como se sincroniza con Dropbox.

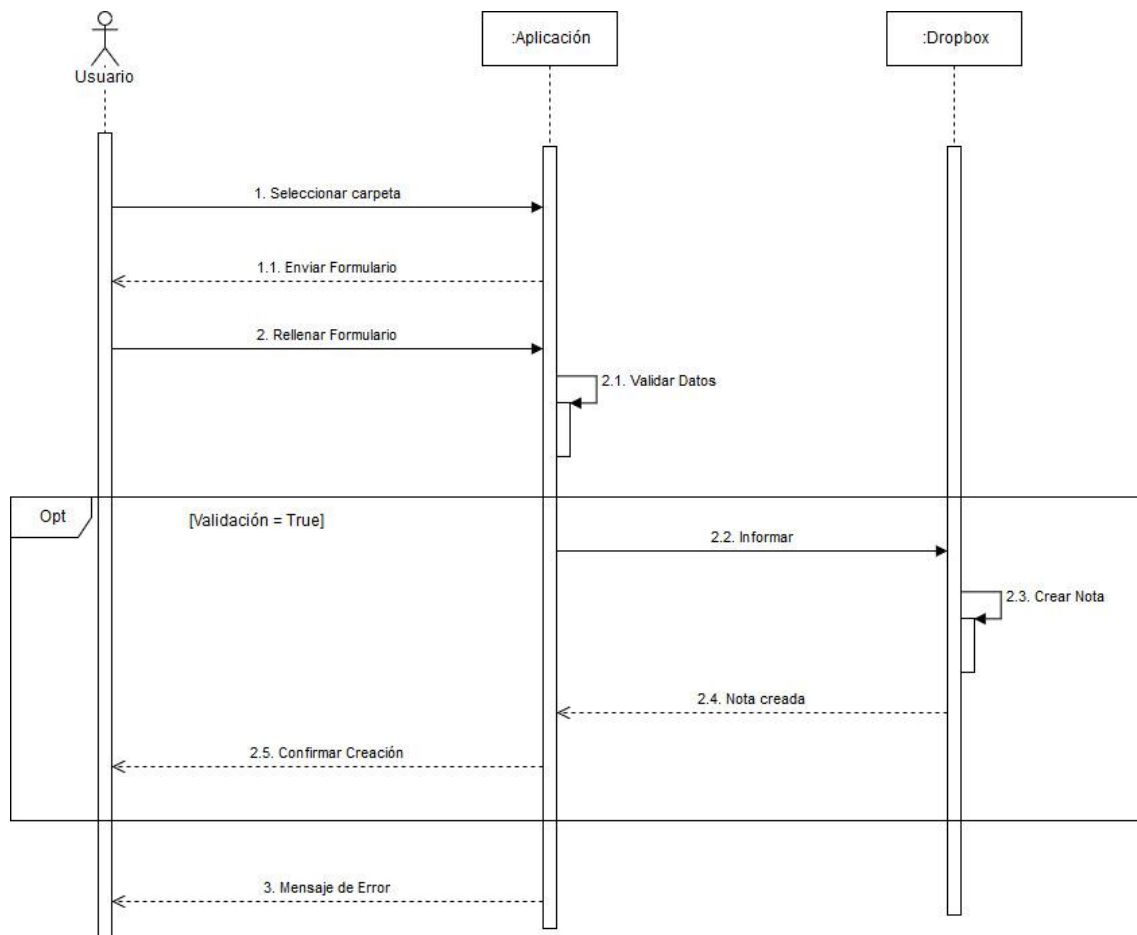


Figura 11: Diagrama de secuencias: Creación de notas

9.5.3. Listado de notas

En la Figura 12 se encuentra un diagrama de secuencias, que representa los pasos que sigue la aplicación para mostrar un listado de las notas que contiene la carpeta seleccionada.

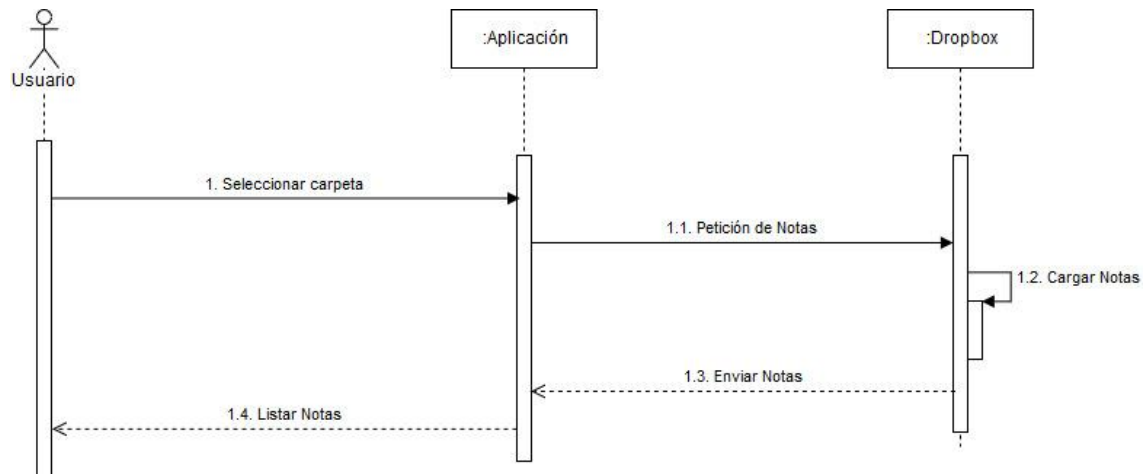


Figura 12: Diagrama de secuencias: Listado de notas

9.5.4. Eliminación de notas

En la figura 13 se ve un diagrama de secuencias, donde se puede ver como se realiza la eliminación de notas y su sincronización con Dropbox.

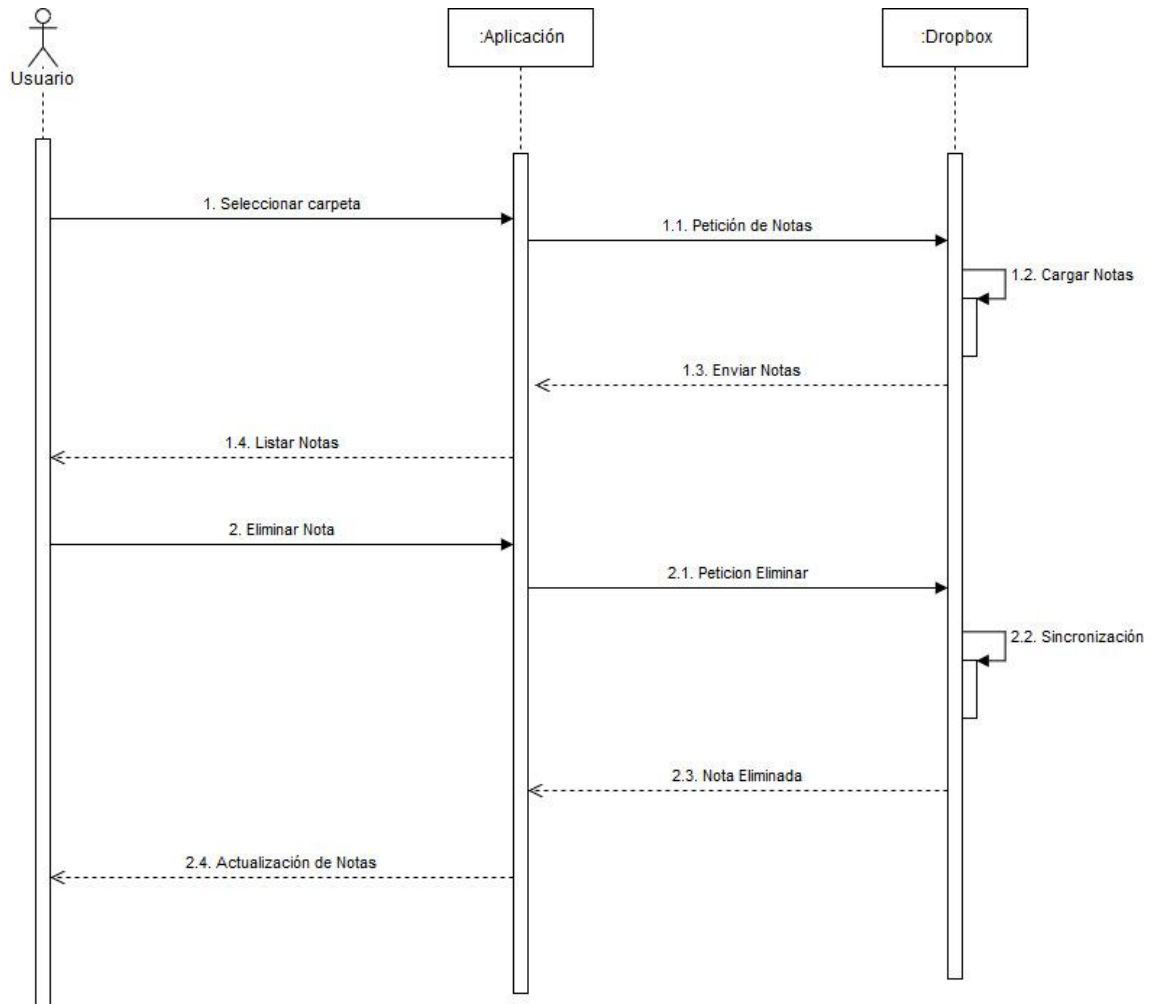


Figura 13: Diagrama de secuencias: Eliminación de notas

9.5.5. Edición de notas

En la figura 14 se encuentra un diagrama de secuencias, el cual muestra como se realiza la edición de notas con su respectiva sincronización con Dropbox.

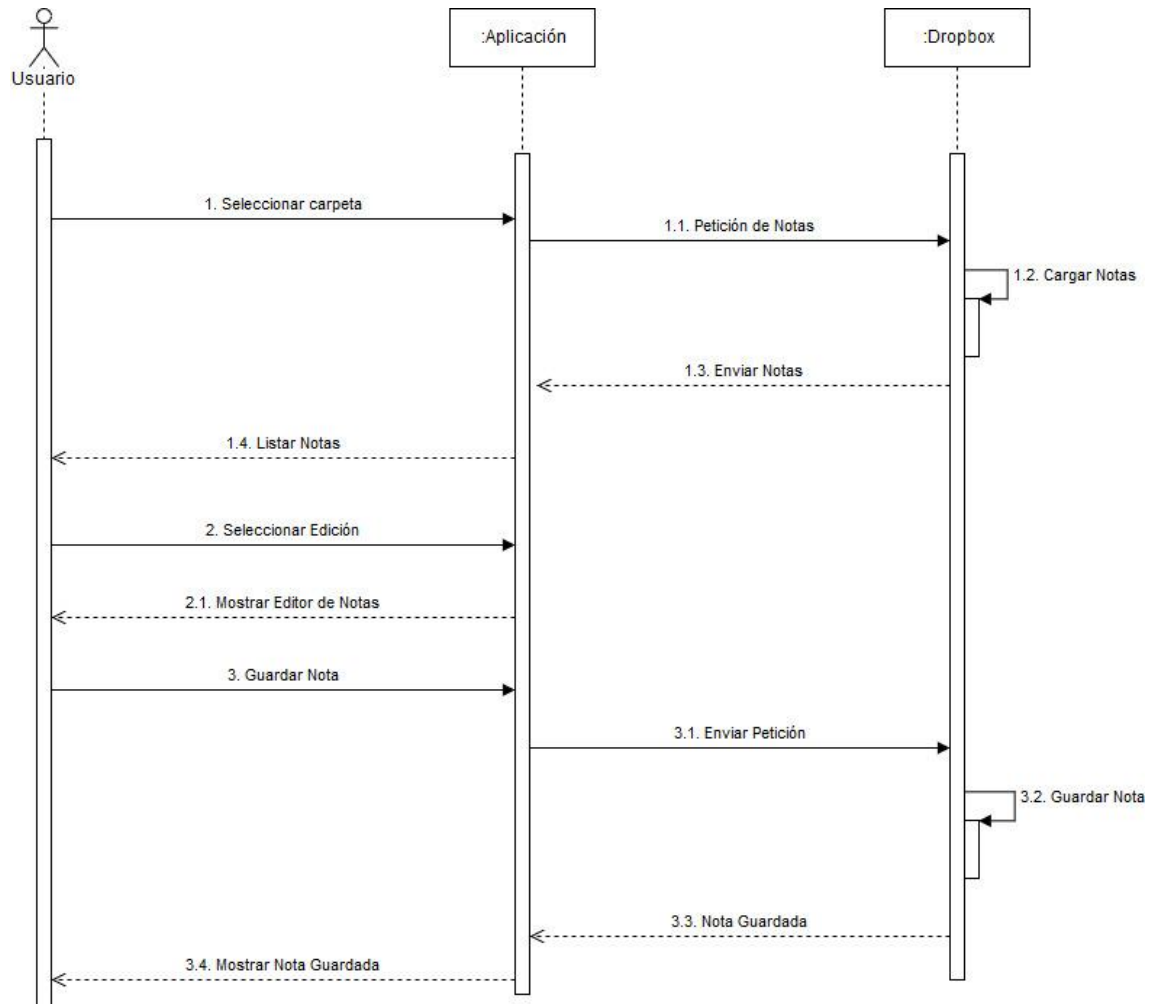


Figura 14: Diagrama de secuencias: Edición de notas

9.6. Interfaz del usuario

En este apartado se describirá el aspecto de la interfaz de usuario de la aplicación web. Esta descripción se centrará en mostrar el diseño de la página. Además, se describirá las funcionalidades que ofrece la aplicación web al usuario.

9.6.1. Estructura básica de la interfaz

La estructura de la app es bastante sencilla. Se divide en dos partes:

- Cabecera: Aquí hallamos el nombre de la aplicación, según el tamaño de la ventana encontramos el menú en su forma completa o responsive, y por último, encontramos el título de cada página.
- Contenido: En esta sección podemos encontrar todos los objetos o componentes que forman cada una de las páginas. Además, de un botón para volver atrás.

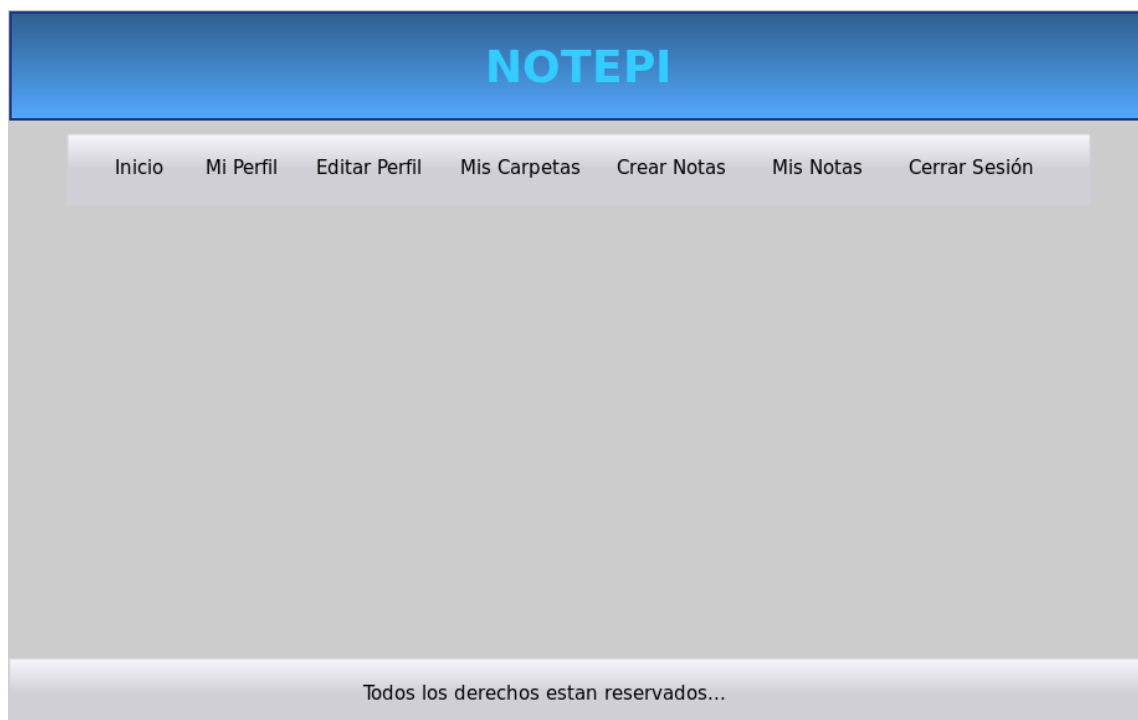
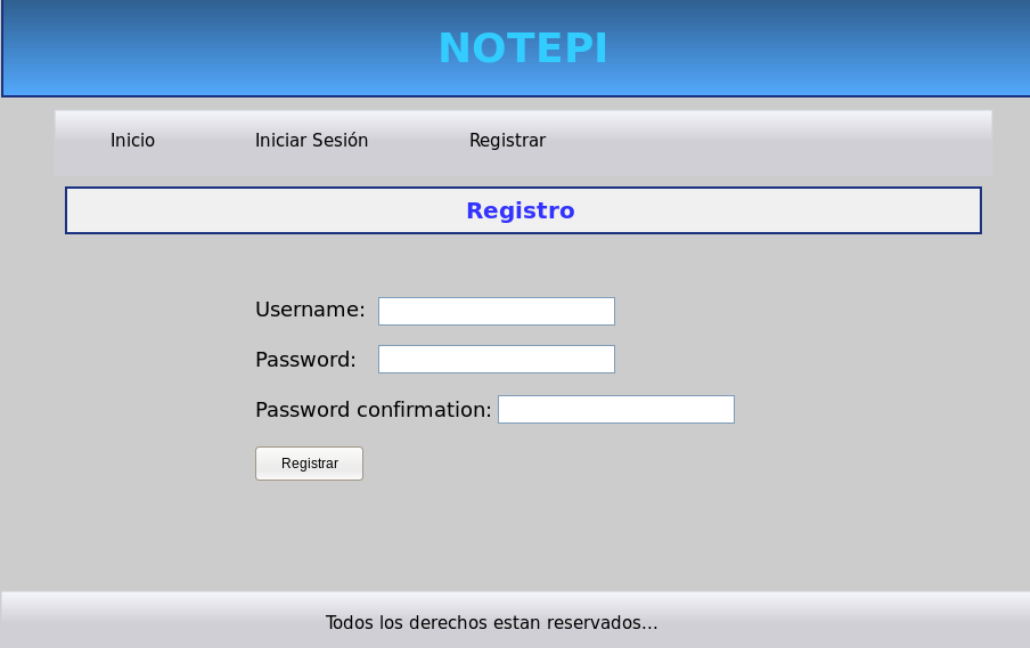


Figura 15: Interfaz básica

9.6.2. Descripción de Registro e Inicio de Sesión

En este apartado se muestra la interfaz de registro e inicio de sesión. Por un lado, en la interfaz de registro, aparece un formulario para que el nuevo usuario pueda registrarse en la aplicación. Por otro lado, tenemos nuestra interfaz de inicio de sesión, en donde se encuentra otro formulario, que el usuario deberá rellenar para acceder a su cuenta.



The screenshot shows the registration interface for NOTEPI. At the top is a blue header with the text "NOTEPI". Below the header is a navigation bar with three links: "Inicio", "Iniciar Sesión", and "Registrar". A white box with a blue border and the text "Registro" is centered below the navigation bar. The registration form consists of three input fields: "Username:", "Password:", and "Password confirmation:". Below the input fields is a button labeled "Registrar". At the bottom of the page, there is a footer with the text "Todos los derechos estan reservados...".

Figura 16: Interfaz de Registro.



The screenshot shows the login interface for NOTEPI. At the top is a blue header with the text "NOTEPI". Below the header is a navigation bar with three links: "Inicio", "Iniciar Sesión", and "Registrar". A white box with a blue border and the text "Iniciar Sesión" is centered below the navigation bar. The login form consists of two input fields: "Username:" and "Password:". Below the input fields is a button labeled "Iniciar Sesión". At the bottom of the page, there is a footer with the text "Todos los derechos estan reservados...".

Figura 17: Interfaz de Inicio de Sesión.

9.6.3. Descripción de Perfil

En este apartado se va a ver la interfaz de "Mi Perfil", en ella se encuentra la información que pertenece al usuario, así como, su nombre, edad y token.



Figura 18: Interfaz de Perfil.

9.6.4. Descripción de Editar Perfil

En este apartado hallamos la interfaz de "Editar Perfil", donde podemos ver unas instrucciones para que el usuario obtenga su token y seguidamente un formulario para que el usuario rellene.

The screenshot displays the 'Editar Perfil' (Edit Profile) page of the NOTEPI application. At the top, a blue header bar contains the application name 'NOTEPI' in white. Below this, a navigation bar with a light gray background lists several menu items: 'Inicio', 'Mi Perfil', 'Editar Perfil' (which is the active page), 'Mis Carpetas', 'Crear Notas', 'Mis Notas', and 'Cerrar Sesión'. The main content area has a light gray background and features a title box at the top with the text 'Editar Datos de Usuario'. Below the title box, there is a section titled 'Para añadir Token:' followed by a placeholder text 'Texto con intrucciones...'. The form contains three input fields: 'Nombre:' with a text input, 'Edad:' with a numeric input showing '0' and a dropdown arrow, and 'Token:' with a text input. A 'Volver atrás' (Go back) button is located at the bottom left of the form area. At the very bottom of the page, a footer bar states 'Todos los derechos estan reservados...'.

Figura 19: Interfaz de Editar Perfil.

9.6.5. Descripción de Mis Carpetas

En este apartado se va a ver la interfaz de "Mis Carpetas", en ella encontramos todas las carpetas pertenecientes al usuario enumeradas en una lista, además de tener la opción de borrar dichas carpetas con un botón habilitado para tal efecto. Seguidamente, tenemos un apartado para añadir nuevas carpetas, en donde solo se debe rellenar un pequeño formulario para su creación.

The screenshot shows the 'NOTEPI' application interface. At the top is a blue header with the text 'NOTEPI'. Below the header is a navigation bar with links: Inicio, Mi Perfil, Editar Perfil, Mis Carpetas, Crear Notas, Mis Notas, and Cerrar Sesión. The main content area is titled 'Mis Carpetas' in a blue box. Below this, there is a list of folders: 'Carpeta1', 'Carpeta2', and 'Carpeta3'. To the right of each folder name is a 'Borrar' (Delete) button. Below the list of folders is a section titled 'Añadir Nueva Carpetas' (Add New Folders) in a blue box. This section contains a label 'Nombre Carpeta:' followed by a text input field. Below the input field is a 'Crear Carpeta' (Create Folder) button. At the bottom left of the main content area is a 'Volver atrás' (Go Back) button. At the very bottom of the page is a footer that says 'Todos los derechos están reservados...' (All rights reserved...).

Figura 20: Interfaz de Mis Carpetas.

The screenshot shows the 'NOTEPI' application interface. At the top is a blue header with the text 'NOTEPI'. Below the header is a navigation bar with links: Inicio, Mi Perfil, Editar Perfil, Mis Carpetas, Crear Notas, Mis Notas, and Cerrar Sesión. The main content area is titled 'Carpeta Borrada' (Folder Deleted) in a blue box. Below this, the area is mostly empty. At the very bottom of the page is a footer that says 'Todos los derechos están reservados...' (All rights reserved...).

Figura 21: Interfaz de Carpeta Borrada.

9.6.6. Descripción de Crear Notas

En este apartado encontramos las interfaces de "Crear Notas", en este caso tenemos dos partes para la creación de una nota. En primer lugar, se nos muestra un listado de las carpetas existentes, en donde deberemos elegir una para en ella guardar la nota. En segundo lugar y ultimo, encontramos un pequeño formulario para rellenar. Con esto tendríamos nuestra nueva nota creada.



The screenshot shows the 'NOTEPI' application interface. At the top is a blue header with the text 'NOTEPI'. Below the header is a navigation bar with links: Inicio, Mi Perfil, Editar Perfil, Mis Carpetas, Crear Notas, Mis Notas, and Cerrar Sesión. The main content area has a title 'Crear Nueva Notas' in a blue box. Below this, there are three links: 'Carpeta1', 'Carpeta2', and 'Carpeta3'. At the bottom left is a button labeled 'Volver atrás'. At the bottom center is the text 'Todos los derechos estan reservados...'.

Figura 22: Interfaz de Crear Notas, elegir carpeta.



The screenshot shows the 'NOTEPI' application interface. At the top is a blue header with the text 'NOTEPI'. Below the header is a navigation bar with links: Inicio, Mi Perfil, Editar Perfil, Mis Carpetas, Crear Notas, Mis Notas, and Cerrar Sesión. The main content area has a title 'Crear Nueva Nota' in a blue box. Below this, there is a form with the label 'Nombre Nota:' followed by a text input field. Below the input field is a button labeled 'Crear Nota'. At the bottom left is a button labeled 'Volver atrás'. At the bottom center is the text 'Todos los derechos estan reservados...'.

Figura 23: Interfaz de Crear Notas, elegir nombre.

9.6.7. Descripción de Mis Notas

En este apartado, encontraremos las interfaces de "Mis Notas". Aquí, podemos señalar varias partes, las cuales iremos mencionando. En primer lugar, se va a encontrar con una lista de nuestras carpetas existentes, en donde tenemos que elegir la carpeta que contiene nuestra nota. En segundo lugar, tenemos un listado de todas nuestras notas en el interior de la carpeta elegida, en este caso tenemos tres botones para elegir qué hacer con esa nota.

- Leer: Podemos seleccionar el botón de lectura, en donde se mostrara nuestra nota solo para lectura.
- Editar: Si seleccionamos este botón accederemos a la interfaz donde nos muestra el editor de notas, en ella podremos modificar nuestra nota y guardarla.
- Borrar: Para borrar la nota solo tendremos que presionar este botón y la nota quedará eliminada.



Figura 24: Interfaz de Mis Notas, elegir carpeta.

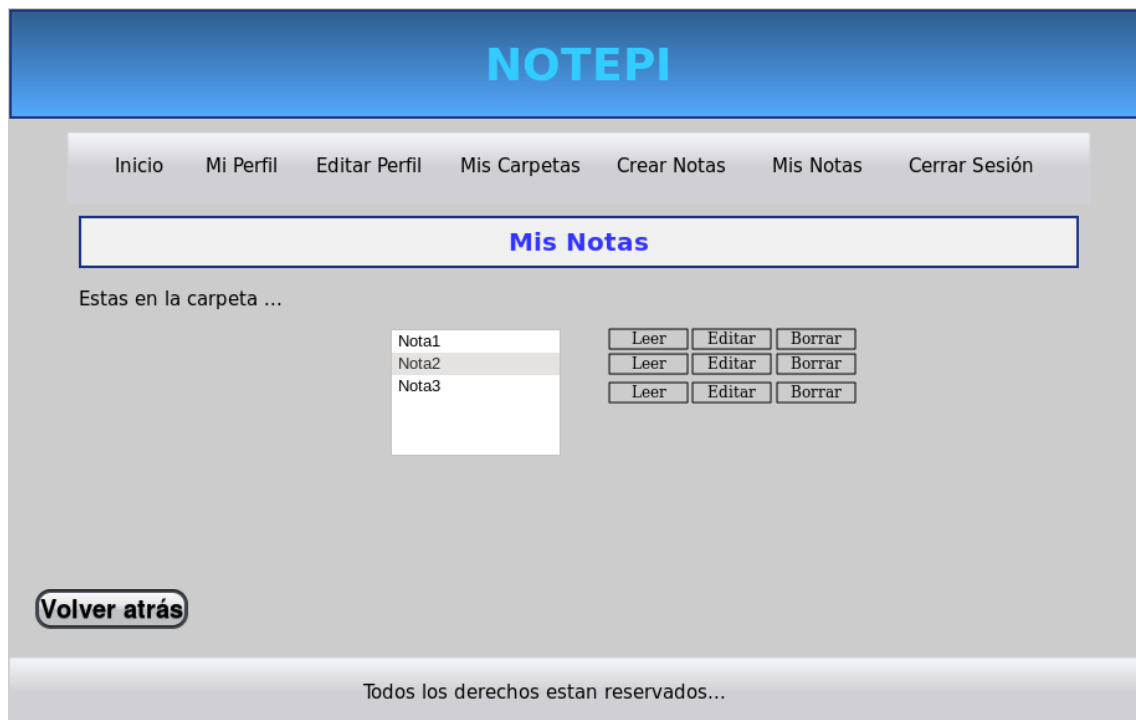


Figura 25: Interfaz de Mis Notas



Figura 26: Interfaz de Mis Notas, lectura de nota.

NOTEPI

[Volver atrás](#)

Nota: ...

OptionOptionOptionOption

Editor de notas..

Guardar Nota

Todos los derechos estan reservados...

Figura 27: Interfaz de Mis Notas, edición de nota.

NOTEPI

[Inicio](#)[Mi Perfil](#)[Editar Perfil](#)[Mis Carpetas](#)[Crear Notas](#)[Mis Notas](#)[Cerrar Sesión](#)

Nota Borrada

Todos los derechos estan reservados...

Figura 28: Interfaz de Mis Notas, borrado de nota.

IV. Pruebas

10. Pruebas del Sistema

El objetivo principal de este apartado es comprobar que el sistema ha sido construido correctamente, comprobando que se cumplen todos los requisitos, que las interfaces se muestran correctamente y la funcionalidad de la aplicación se ejecuta correctamente produciendo el resultado esperado en el prototipo final.

Para llevar a cabo este punto, es necesario comprobar que la visualización de los resultados son los esperados, probar la funcionalidad de la misma, y que todo realmente realiza lo que se esperaba de cada parte del sistema, además, de comprobar la sincronización de los elementos. Todo esto nos permitirá verificar y revelar la calidad del producto.

Para ello se realizarán pruebas de los casos de uso, los procedimientos y el comportamiento de la interfaz de usuario para finalmente contrastar el resultado con el esperado. Las pruebas se harán sobre la aplicación web en el modo desarrollo que ofrece el framework Django.

10.1. Pruebas de casos de uso

Para llevar a cabo las pruebas de casos de uso se definirán casos de prueba para comprobar que se cumplen con las especificaciones que se detallan en cada caso de uso.

10.1.1. Caso de prueba registrar a un usuario

Nombre	Registrar a un usuario
Condiciones	El usuario habrá tenido que rellenar un formulario para su registro.
Salida esperada	Se espera que el usuario quede registrado en la aplicación.
Salida obtenida	El usuario queda registrado en la aplicación.
Resultado	Correcto

Tabla 22: Tabla Caso de Pruebas registrar a un usuario

10.1.2. Caso de prueba iniciar sesión a un usuario

Nombre	Iniciar sesión a un usuario
Condiciones	El usuario habrá rellenado el formulario de inicio de sesión.
Salida esperada	Se espera que el usuario haya podido acceder a la aplicación sin problemas.
Salida obtenida	El usuario accede a la aplicación.
Resultado	Correcto

Tabla 23: Tabla Caso de Pruebas iniciar sesión de un usuario

10.1.3. Caso de prueba cerrar sesión de un usuario

Nombre	Cerrar sesión de un usuario
Condiciones	El usuario debe tener una sesión iniciada para poder cerrarla.
Salida esperada	Se espera que el usuario pueda cerrar su sesión sin problemas.
Salida obtenida	El usuario cierra su sesión.
Resultado	Correcto

Tabla 24: Tabla Caso de Pruebas cerrar sesión de un usuario

10.1.4. Caso de prueba editar perfil de un usuario

Nombre	Editar perfil de un usuario
Condiciones	El usuario habrá tenido que rellenar el formulario de modificación de perfil completamente.
Salida esperada	Se espera que los datos del usuario se guarden correctamente en el sistema.
Salida obtenida	Los datos del usuario se guardan correctamente en el sistema.
Resultado	Correcto

Tabla 25: Tabla Caso de Pruebas editar perfil de un usuario

10.1.5. Caso de prueba ver perfil de un usuario

Nombre	Ver perfil de un usuario
Condiciones	El usuario debe tener una sesión iniciada y haber modificado su perfil para visualizar los datos.
Salida esperada	Se espera que la información del usuario se muestre en pantalla.
Salida obtenida	Se muestra en pantalla la información del usuario.
Resultado	Correcto.

Tabla 26: Tabla Caso de Pruebas ver perfil de un usuario

10.1.6. Caso de prueba crear carpeta a un usuario

Nombre	Crear carpeta a un usuario
Condiciones	El usuario habrá tenido que rellenar el formulario para crear una nueva carpeta.
Salida esperada	Se espera que la carpeta haya sido creada correctamente.
Salida obtenida	La carpeta ha sido creada correctamente.
Resultado	Correcto

Tabla 27: Tabla Caso de Pruebas crear carpeta a un usuario

10.1.7. Caso de prueba ver carpetas de un usuario

Nombre	Ver carpetas de un usuario
Condiciones	El usuario debe haber creado con anterioridad alguna carpeta.
Salida esperada	Se espera que se muestre una lista con las carpetas existentes del usuario.
Salida obtenida	Se muestra en una lista las carpetas que tiene dicho usuario.
Resultado	Correcto

Tabla 28: Tabla Caso de Pruebas ver carpetas de un usuario

10.1.8. Caso de prueba borrar una carpeta de un usuario

Nombre	Borrar una carpeta de un usuario
Condiciones	El usuario debe tener alguna carpeta.
Salida esperada	Se espera que se elimina la carpeta seleccionada por el usuario.
Salida obtenida	La carpeta seleccionada por el usuario queda eliminada.
Resultado	Correcto

Tabla 29: Tabla Caso de Pruebas borrar una carpeta de un usuario

10.1.9. Caso de prueba elegir una carpeta

Nombre	Elegir una carpeta
Condiciones	El usuario habrá tenido que hacer alguna carpeta anteriormente y elegir una de ellas.
Salida esperada	Se espera que se seleccione una carpeta para poder realizar la siguiente operación.
Salida obtenida	Se selecciona una carpeta correctamente.
Resultado	Correcto

Tabla 30: Tabla Caso de Pruebas elegir una carpeta

10.1.10. Caso de prueba crear una nota a un usuario

Nombre	Crear una nota a un usuario
Condiciones	El usuario habrá tenido que elegir con anterioridad una carpeta para crear la nota. Además, deberá añadir el nombre que quiere darle a la nota.
Salida esperada	Se espera que se cree una nueva nota correctamente.
Salida obtenida	Se crea una nueva nota.
Resultado	Correcto

Tabla 31: Tabla Caso de Pruebas crear una nota a un usuario

10.1.11. Caso de prueba borrar una nota de un usuario

Nombre	Borrar una nota de un usuario
Condiciones	El usuario habrá tenido que crear alguna nota con anterioridad y seleccionar cual quiere borrar.
Salida esperada	Se espera que la nota quede eliminada.
Salida obtenida	La nota ha sido eliminada correctamente.
Resultado	Correcto

Tabla 32: Tabla Caso de Pruebas borrar una nota de un usuario

10.1.12. Caso de prueba listar notas de un usuario

Nombre	Listar notas de un usuario
Condiciones	El usuario habrá tenido que crear alguna nota anteriormente y haber elegido una carpeta.
Salida esperada	Se espera que aparezca una lista con todas las notas existentes del usuario.
Salida obtenida	Se muestra una lista de todas las notas que posee el usuario.
Resultado	Correcto

Tabla 33: Tabla Caso de Pruebas listar notas de un usuario

10.1.13. Caso de prueba leer una nota de un usuario

Nombre	Leer una nota de un usuario
Condiciones	El usuario habrá tenido que crear alguna nota anteriormente y haber elegido una nota.
Salida esperada	Se espera que se muestre el contenido de la nota seleccionada por el usuario.
Salida obtenida	Se muestra en pantalla el contenido de la nota seleccionada.
Resultado	Correcto

Tabla 34: Tabla Caso de Pruebas leer una nota de un usuario

10.1.14. Caso de prueba editar nota de un usuario

Nombre	Editar nota de un usuario
Condiciones	El usuario habrá tenido que crear alguna nota anteriormente y haber elegido una nota.
Salida esperada	Se espera que se muestre en pantalla el editor de texto para poder modificar la nota.
Salida obtenida	Se muestra el editor de texto para poder modificar la nota elegida.
Resultado	Correcto

Tabla 35: Tabla Caso de Pruebas editar nota de un usuario

V. Conclusiones

11. Conclusiones

En este apartado se tratará de exponer las conclusiones a las que se han llegado a cabo una vez finalizadas las fases de desarrollo de la aplicación. Las conclusiones estarán relacionadas con los objetivos planteados inicialmente y con los obtenidos después de realizar la fase de pruebas. También se tratará de llegar a una conclusión sobre qué mejoras se podrían desarrollar en un futuro.

11.1. Conclusiones sobre los objetivos planteados

Se puede decir que todos los objetivos que fueron planteados en el apartado 3 de la documentación se han cumplido satisfactoriamente.

El objetivo principal consistía en desarrollar una aplicación para notas en un entorno web que cumpliera una serie de requisitos, los cuales se van a ir nombrándolos a continuación.

- Se ha desarrollado una aplicación que cumple con el objetivo de permitir al usuario crear, modificar o borrar notas. Además, estas notas se almacenarán con un formato XHTML Basic 1.1 y podrán ser archivadas en diferentes carpetas.
- En cuanto a las notas, podremos hacer una lectura de ellas en diferentes dispositivos mediante un diseño responsive. Por otro lado, a la hora de editar las notas en un primer momento se propuso que el editor dispondría de un formato con título, negrita, cursiva y listas; sin embargo, se ha ampliado y tiene muchas más opciones en lo concerniente en lo que se refiere al formato del texto de las notas.
- Se permitirá la sincronización de nuestras notas a través del sistema de almacenamiento en la nube de Dropbox. Además, la aplicación será compatible con los diferentes navegadores de sistema como podrán ser, Linux, Android, IOS o Windows.
- La aplicación consta de un menú responsive para acceder a las diferentes funciones de la aplicación. Por último, su diseño y facilidad de uso hace que cualquier tipo de usuario pueda utilizarla sin ningún tipo de problema.

11.2. Conclusiones de la fase de pruebas

En la fase de pruebas se han ejecutado una serie de pruebas para comprobar que el sistema ofrece la funcionalidad esperada y que no se producen errores, ni por parte de la interacción del usuario con la aplicación ni por parte de la comunicación con el servidor.

Aunque tengamos estos resultados, no se puede definir con total rotundidad que la aplicación esté libre de errores. En el uso de la aplicación, podrían ocurrir situaciones que no hayan sido contempladas, las cuales ocasionarían algunos errores que deberían ser solucionados en el futuro.

11.3. Futuras mejoras

La aplicación cumple con todos los objetivos planteados pero se podría añadir más funcionalidades y mejorar el sistema de cara a próximas versiones.

En la aplicación se gestionan notas personales y su organización, por ello podríamos ofrecer otras funcionalidades como:

- Creación de subcarpetas, para mayor organización.
- Introducir la opción de buscador por notas y/o carpetas.
- Creación de apk en otros sistemas.

Bibliografía

[1] Evernote, artículo de Wikipedia "Evernote", web consultada 2016-12-15 :

<https://es.wikipedia.org/wiki/Evernote>

[2] Tomboy, artículo de Wikipedia "Tomboy (software)", web consultada 2016-12-15 :

[https://es.wikipedia.org/wiki/Tomboy_\(software\)](https://es.wikipedia.org/wiki/Tomboy_(software))

[3] Tomdroid, web consultada 2016-12-15 :

<https://play.google.com/store/apps/details?id=org.tomdroid>

[4] Google Keep, artículo de Wikipedia "Google Keep", web consultada 2016-12-15 :

https://es.wikipedia.org/wiki/Google_Keep

[5] Dropbox Paper, artículo de "El Android libre", web consultada 2017-3-8 :

<https://www.elandroidelibre.com/2016/08/dropbox-paper.html>

[6] Django, artículo de Wikipedia "Django (framework)", web consultada 2016-12-10

[https://es.wikipedia.org/wiki/Django_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework))

[7] Python, artículo de Wikipedia "Python ", web consultada 2016-12-10

<https://es.wikipedia.org/wiki/Python>

[8] HTML5, artículo de Wikipedia "HTML5 ", web consultada 2016-12-15

<https://es.wikipedia.org/wiki/HTML5>

[9] CSS, pagina web w3schools, web consultada 2016-12-15

<https://www.w3schools.com/css/>

[10] Dropbox, pagina web Dropbox, web consultada 2017-2-27

<https://www.dropbox.com/developers/documentation/python#install>

[11] UML (Lenguaje Unificado de Modelado), web consultada 2017-7-18

https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

[12] Icomoon, web consultada 2017-7-28

<https://icomoon.io/>

[13] Android Studio, web consultada 2017-7-29

<https://developer.android.com/studio/install.html>

[14] Summernote, web consultada 2017-6-27

<http://summernote.org/>

[15] Menú responsive, web consultada 2017-7-28

<http://www.falconmasters.com/web-design/menu-de-navegacion-responsive/>

[16] Phonegap, web consultada 2017-7-29

<http://victorroblesweb.es/2016/01/09/crear-apk-con-cordova/>

[17] Dinahosting, web consultada 2017-8-2

<https://dinahosting.com/>