

# UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

## TRABAJO FIN DE GRADO

“Interfaz web para la manipulación de ficheros pdf”

**JUNIO 2021**

DIRECTOR: Miguel Riesco Albizu

AUTOR: Enrique Avello Díaz



# Agradecimientos

Primeramente, a mis padres y a mi hermana por su cariño, por creer en mí y darme fuerzas cuando más lo necesitaba. Sin vosotros todo esto hubiese sido mucho más difícil.

También agradecer a mis abuelos su continuo apoyo en estos años, su confianza y el haberme acompañado en todo este camino. A mi tía Maribel, por haber estado día tras día a mi lado durante esta etapa y al resto de la familia por estar siempre ahí cuando se necesita.

Además, dar las gracias a los compañeros, a los que he conocido en estos años y a los de siempre. Me quedo con todos los momentos que hemos compartido juntos y los que nos quedan por compartir.

Por último, dar las gracias a todo el personal docente, y en especial a mi tutor, por la excelente formación recibida en este periodo.

# Resumen

El principal objetivo de este Trabajo Fin de Grado es la creación de una herramienta web para la manipulación de ficheros PDF. Estos ficheros tienen gran número de ventajas: ofrecen la posibilidad de bloquearlos con contraseña, compatibilidad entre los sistemas operativos principales, tienen gran popularidad... No obstante, el formato presenta un problema, y es que manipularlos no es una tarea simple.

En este proyecto se ha analizado, diseñado y construido el sistema descrito en la primera línea de este resumen, con el objetivo de permitir al usuario realizar tareas con documentos PDF como, por ejemplo, la extracción o eliminación de páginas de un fichero, recortar el contenido de las páginas de un documento, unir dos o más PDF o modificar el orden de las páginas.

El resultado final, además de este documento, en el que se recoge todo el proceso de realización del proyecto a través de las distintas fases, es una aplicación web desplegada con las funcionalidades identificadas anteriormente, además de algunas otras. Esta aplicación está formada por dos subsistemas principales: un *backend*, implementado sobre Django REST Framework, que expone una API REST, y un *frontend*, con el que interactúa el usuario y que se ha implementado sobre Vue.js.

# Palabras clave

Manipulación de documentos PDF, Aplicación de una Sola Página, API Rest, Django, Vue.

# Abstract

The main goal of this Final Degree Project is the creation of a web tool that allows the manipulation of PDF files. This kind of file has a large number of advantages: they offer the chance of locking them with a password, compatibility between the main operating systems, it is a very popular format... However, there is also a problem, which is that manipulating them is not a simple task.

Throughout this project, the described system has been analysed, designed and built to allow users to carry out tasks with PDF documents, such as, extracting or deleting pages from a file, trimming the content of the pages, joining two or more documents into one or changing the page order.

The final result, in addition to this document, with the whole process of carrying out the project through the different phases is described in, is a web application, deployed with the functionalities identified above and some more else. This application is made up of two main sub-systems: a backend, implemented in Django REST Framework, which exposes a REST API, and a frontend, which the user interacts with and it has been implemented in Vue.js.

# Keywords

PDF document manipulation, Single Page Application, Rest API, Django, Vue.

# Índice General

<b>Capítulo 1. Memoria del proyecto .....</b>	<b>17</b>
1.1 Resumen de la motivación, objetivos y alcance del proyecto .....	17
1.2 Resumen de todos los aspectos .....	17
1.3 Originalidad de la obra .....	19
<b>Capítulo 2. Introducción .....</b>	<b>20</b>
2.1 Justificación del proyecto.....	20
2.2 Objetivos del Proyecto .....	20
2.3 Estudio de la situación actual.....	21
2.3.1 Estudio previo.....	21
2.3.2 Evaluación de alternativas .....	25
<b>Capítulo 3. Aspectos teóricos .....</b>	<b>27</b>
3.1 Conceptos.....	27
3.1.1 REST .....	27
3.1.2 MVC.....	27
3.1.3 SPA.....	28
3.2 Tecnologías.....	28
3.2.1 Django .....	29
3.2.2 Vue .....	29
3.2.3 PyPDF2.....	29
3.2.4 BootstrapVue .....	30
<b>Capítulo 4. Planificación del Proyecto y Presupuesto iniciales .....</b>	<b>31</b>
4.1 Planificación inicial .....	31
4.1.1 Estudios previos .....	32
4.1.2 Documentación inicio proyecto .....	33
4.1.3 Formación.....	33
4.1.4 Planificación y presupuesto inicial .....	34



4.1.5	Análisis.....	34
4.1.6	Diseño.....	35
4.1.7	Implementación .....	35
4.1.8	Realización de pruebas.....	37
4.1.9	Despliegue de la aplicación .....	37
4.1.10	Documentación final .....	38
4.2	Presupuesto de costes inicial .....	39
4.2.1	Modelo de empresa .....	39
4.2.2	Desarrollo del presupuesto de costes inicial.....	43
<b>Capítulo 5. Análisis .....</b>		<b>53</b>
5.1	Definición del Sistema.....	53
5.1.1	Determinación del alcance del sistema.....	53
5.2	Requisitos del sistema.....	53
5.2.1	Obtención de los requisitos del sistema .....	53
5.2.2	Identificación de actores del sistema.....	59
5.2.3	Especificación de Casos de uso .....	59
5.3	Identificación de los subsistemas en la Fase de Análisis.....	63
5.3.1	Descripción de los subsistemas.....	63
5.3.2	Descripción de los interfaces entre subsistemas .....	65
5.4	Diagrama de Clases Preliminar del Análisis.....	66
5.4.1	Diagrama de clases.....	66
5.4.2	Descripción de las clases .....	68
5.5	Análisis de Interfaces de usuario.....	82
5.5.1	Descripción de la interfaz.....	82
5.5.2	Descripción del comportamiento de la interfaz.....	88
5.5.3	Diagrama de navegabilidad.....	89
5.6	Especificación del Plan de Pruebas .....	92
<b>Capítulo 6. Diseño del sistema.....</b>		<b>93</b>

6.1	Arquitectura del sistema .....	93
6.1.1	Diagrama de paquetes .....	93
6.1.2	Diagramas de despliegue .....	96
6.2	Diseño de clases .....	97
6.2.1	Diagrama de clases.....	97
6.3	Diagramas de Interacción.....	100
6.3.1	Ordenar documento.....	100
6.3.2	Consultar historial de documentos generados .....	100
6.4	Diseño de la base de datos.....	101
6.4.1	Descripción del SGBD usado.....	101
6.4.2	Integración del SGBD en nuestro sistema.....	102
6.4.3	Diagrama E-R.....	103
6.5	Diseño de la interfaz.....	103
6.5.1	Pantalla de inicio de sesión .....	103
6.5.2	Pantalla de registro .....	104
6.5.3	Pantalla de inicio .....	104
6.5.4	Historial de documentos .....	105
6.5.5	Eliminar/Extraer páginas.....	105
6.5.6	Mezclar documentos.....	106
6.5.7	Ordenar .....	106
6.5.8	Recortar.....	107
6.5.9	Dividir .....	107
6.5.10	Descarga de fichero.....	108
6.6	Especificación técnica del plan de pruebas.....	108
6.6.1	Pruebas funcionales .....	108
6.6.2	Pruebas de usabilidad .....	116
<b>Capítulo 7. Implementación del sistema.....</b>		<b>120</b>
7.1	Estándares y Normas Seguidos .....	120

7.1.1	JSON .....	120
7.1.2	Vue Style Guide .....	120
7.2	Lenguajes de programación .....	120
7.2.1	Python 3 .....	120
7.2.2	JavaScript.....	121
7.2.3	Otros lenguajes .....	121
7.3	Herramientas y Programas Usados para el desarrollo.....	122
7.3.1	Visual Studio Code.....	123
7.3.2	PowerShell.....	123
7.3.3	Git.....	124
7.3.4	Postman.....	124
7.3.5	Google Chrome.....	125
7.4	Creación del sistema .....	125
7.4.1	Problemas Encontrados .....	125
7.4.2	Descripción Detallada de las clases.....	128
<b>Capítulo 8. Desarrollo de las Pruebas .....</b>		<b>129</b>
8.1	Pruebas funcionales .....	129
8.1.1	Pruebas “manuales” .....	129
8.1.2	Pruebas unitarias automatizadas.....	139
8.2	Pruebas de usabilidad .....	142
8.2.1	Resultados de los cuestionarios de evaluación.....	143
8.2.2	Análisis de los resultados de los cuestionarios .....	150
<b>Capítulo 9. Manuales del Sistema.....</b>		<b>152</b>
9.1	Manual de Instalación.....	152
9.2	Manual de Ejecución y Despliegue.....	152
9.2.1	Instalación de Python y Node.js.....	152
9.2.2	Puesta en marcha del backend .....	153
9.2.3	Puesta en marcha del frontend.....	155

9.2.4	Despliegue de la aplicación .....	156
9.3	Manual de Usuario .....	157
9.3.1	Barra de navegación .....	158
9.3.2	Registro .....	158
9.3.3	Inicio de sesión .....	159
9.3.4	Página de inicio.....	160
9.3.5	Mezclar/Unir documentos .....	161
9.3.6	Ordenar .....	163
9.3.7	Eliminar extraer páginas.....	165
9.3.8	Recortar.....	166
9.3.9	Dividir .....	167
9.3.10	Historial .....	168
9.4	Manual del Programador .....	171
9.4.1	Herramientas de gestión de dependencias/paquetes .....	171
9.4.2	Librerías y frameworks más importantes.....	171
9.4.3	Estructura de directorios del proyecto .....	172
9.4.4	Ejemplo de creación de una nueva funcionalidad .....	175
<b>Capítulo 10. Conclusiones y ampliaciones.....</b>		<b>176</b>
10.1	Conclusiones.....	176
10.2	Ampliaciones .....	176
10.2.1	Añadir más acciones a la aplicación .....	177
10.2.2	Encadenamiento de acciones.....	177
10.2.3	Emplear ficheros del historial para realizar acciones.....	177
10.2.4	Permitir la selección de documentos de distintas carpetas.....	177
10.2.5	Permitir la selección de la página de referencia para el recorte de documentos 177	
10.2.6	Mejorar las opciones de usuarios .....	178
10.2.7	Internacionalización de la aplicación .....	178
<b>Capítulo 11. Planificación del Proyecto y Presupuesto finales.....</b>		<b>179</b>

11.1	Planificación final .....	179
11.1.1	Estudios previos .....	180
11.1.2	Documentación inicio proyecto .....	181
11.1.3	Formación.....	181
11.1.4	Planificación y presupuesto inicial .....	182
11.1.5	Análisis.....	182
11.1.6	Diseño.....	183
11.1.7	Implementación .....	184
11.1.8	Realización de pruebas.....	186
11.1.9	Despliegue de la aplicación .....	186
11.1.10	Documentación final .....	186
11.2	Presupuesto de costes final .....	188
11.2.1	Desarrollo del presupuesto de costes final .....	188
<b>Capítulo 12. Referencias Bibliográficas .....</b>		<b>198</b>
<b>Capítulo 13. Apéndices .....</b>		<b>202</b>
13.1	Glosario .....	202
13.2	Contenido entregado en el archivo adjunto .....	203
13.3	Código fuente.....	203

# Índice de figuras

Figura 2.1 Interfaz del programa PDFtk .....	22
Figura 2.2 Pantalla principal de la página ilovepdf .....	23
Figura 2.3 Interfaz de la página PDFCandy.....	24
Figura 2.4 Limitaciones de PDFCandy .....	24
Figura 3.1 Imagen de REST .....	27
Figura 3.2 Diagrama explicativo del patrón MVC.....	28
Figura 3.3 Logo de Django.....	29
Figura 3.4 Logo De Vue .....	29
Figura 3.5 Logo de BootstrapVue.....	30
Figura 4.1 Actividades y diagrama de Gantt de alto nivel.....	32
Figura 5.1 Diagrama de casos de uso .....	60
Figura 5.2 Diagrama de clases de análisis del subsistema “Servidor web” .....	67
Figura 5.3 Diagrama de “clases” de análisis del subsistema aplicación web.....	68
Figura 5.4 Mockup de pantalla de inicio de sesión .....	83
Figura 5.5 Mockup de pantalla de registro .....	84
Figura 5.6 Mockup de pantalla principal.....	84
Figura 5.7 Mockup del historial de documentos.....	85
Figura 5.8 Mockup de la página de eliminación/extracción de páginas .....	85
Figura 5.9 Mockup de la página de unión de documentos .....	86
Figura 5.10 Mockup de la página de ordenación de ficheros .....	86
Figura 5.11 Mockup de la página de recorte de ficheros.....	87
Figura 5.12 Mockup de la página de descarga de ficheros .....	87
Figura 5.13 Imagen representativa de una acción llevada a cabo con éxito .....	88
Figura 5.14 Imagen representativa de acción no llevada a cabo con éxito .....	89
Figura 5.15 Diálogo de confirmación de acción .....	89
Figura 5.16 Diagrama de navegabilidad de la aplicación .....	91
Figura 6.1 Diagrama de paquetes de ambos subsistemas .....	94
Figura 6.2 Diagrama de despliegue .....	96
Figura 6.3 Diagrama de clases de diseño del servidor web .....	98
Figura 6.4 Diagrama de “clases” de diseño de la aplicación web .....	99
Figura 6.5 Diagrama de interacción para la ordenación de un documento .....	100
Figura 6.6 Diagrama de interacción para la consulta del historial de documentos.....	101
Figura 6.7 Logo de SQLite.....	101
Figura 6.8 Logo de PostgreSQL.....	102
Figura 6.9 Configuración para la integración de PostgreSQL en el sistema.....	103
Figura 6.10 Pantalla de inicio de sesión .....	104
Figura 6.11 Pantalla de registro .....	104
Figura 6.12 Pantalla de inicio .....	105
Figura 6.13 Pantalla Historial de ficheros.....	105
Figura 6.14 Pantalla de eliminación/extracción de páginas.....	106

Figura 6.15 Pantalla de unión de documentos .....	106
Figura 6.16 Pantalla de ordenación de documentos. ....	107
Figura 6.17 Pantalla de recorte de documentos.....	107
Figura 6.18 Pantalla de división de un fichero .....	108
Figura 6.19 Pantalla de descarga de un fichero. ....	108
Figura 7.1 Logo de Python.....	120
Figura 7.2 Logo de JavaScript .....	121
Figura 7.3 Logo de HTML.....	122
Figura 7.4 Logo de CSS .....	122
Figura 7.5 Logo de Visual Studio Code .....	123
Figura 7.6 Logo de PowerShell .....	123
Figura 7.7 Logo de Git .....	124
Figura 7.8 Logo de POSTMAN .....	124
Figura 7.9 Logo de Google Chrome .....	125
Figura 7.10 Configuración de Cuentas y Claves de Azure (Las claves se muestran recortadas).....	127
Figura 7.11 Configuración de Azure. Se establece carpeta por defecto y que no se deben sobrescribir ficheros (se renombrarán en su lugar) .....	128
Figura 8.1 Indicador de tamaño inicial.....	150
Figura 8.2 Indicador de tamaño tras el cambio .....	151
Figura 9.1 Muestra de activación del entorno virtual venv .....	153
Figura 9.2 Distintas configuraciones de base de datos.....	154
Figura 9.3 Indicador de servidor Django arrancado.....	154
Figura 9.4 Página de error del servidor.....	155
Figura 9.5 Configuración de URL de autenticación .....	156
Figura 9.6 Configuración de URL genérica .....	156
Figura 9.7 Indicador servidor Frontend.....	156
Figura 9.8 Creación de nueva app (servidor) en Heroku.....	157
Figura 9.9 Configuraciones de despliegue de Heroku.....	157
Figura 9.10 Barra de navegación de la aplicación .....	158
Figura 9.11 Opciones de un usuario sin sesión iniciada.....	158
Figura 9.12 Opciones de usuario con sesión iniciada.....	158
Figura 9.13 Campos del componente de registro de usuario .....	159
Figura 9.14 Campos del componente de inicio de sesión.....	160
Figura 9.15 Opciones de la página de inicio.....	161
Figura 9.16 Componente selector de ficheros .....	161
Figura 9.17 Selector de ficheros del sistema operativo .....	162
Figura 9.18 Página de unión de PDF una vez seleccionados archivos válidos .....	162
Figura 9.19 Capa de espera .....	163
Figura 9.20 Componente de descarga de ficheros.....	163
Figura 9.21 Presentación de las páginas una vez seleccionado un fichero.....	164
Figura 9.22 Presentación de las páginas al invertirlas .....	164
Figura 9.23 Aspecto inicial de la página de eliminación/extracción de páginas .....	165
Figura 9.24 Aspecto de la página de eliminación/extracción de páginas una vez seleccionado un fichero .....	166

Figura 9.25 Componente de recorte de una página .....	167
Figura 9.26 Página de división de ficheros una vez seleccionado un documento .....	168
Figura 9.27 Historial de ficheros .....	168
Figura 9.28 Descarga de un fichero almacenado .....	169
Figura 9.29 Confirmación de la eliminación de ficheros .....	170
Figura 9.30 Selección de todos los elementos de una página .....	170
Figura 11.1 Actividades y diagrama de Gantt de alto nivel de la planificación final.....	180



# Capítulo 1. Memoria del proyecto

## 1.1 Resumen de la motivación, objetivos y alcance del proyecto

Este Trabajo Fin de Grado consiste en la construcción de un proyecto software, pasando por todos los puntos del ciclo de vida de este: toma de requisitos, análisis, diseño, implementación, pruebas, e incluso despliegue.

El resultado de este proceso, además de la presente documentación, será una aplicación capaz de manipular ficheros PDF a través de una interfaz simple, de modo que cualquier persona que tenga unos conocimientos informáticos a nivel de usuario pueda usarla sin mayores complicaciones. Esta aplicación se encontrará desplegada en una plataforma en la nube.

Obviamente, existen muchos tipos de operaciones que se pueden realizar sobre ficheros PDF, lo que hace que este proyecto pudiese tener un alcance prácticamente ilimitado. Por este motivo, se ha decidido que se implementarán algunas de las que pueden ser más habituales, pero estructurando la aplicación de tal forma que incluir nuevas acciones en el futuro no sea una tarea difícil. Las acciones de esta primera versión de la aplicación serán:

- **Extracción de páginas.**
- **Eliminación de páginas.**
- **Modificación del orden de las páginas.**
- **Unión de varios ficheros.**
- **Recorte de páginas.**

La interfaz web de manipulación de los ficheros estará desacoplada del procesamiento de estos, permitiendo que, si en un futuro se desea cambiar la forma de procesarlos, hacer cambios en la interfaz o crear una nueva, el otro subsistema no se vea afectado. Esto se conseguirá dividiendo la aplicación en dos subsistemas. Un *backend* expondrá los servicios encargados del procesamiento a través de una **API Rest**. El *frontend*, con el que interactuará el usuario, se encargará de recoger y mostrar datos y consumirá esta API para manipular los ficheros.

## 1.2 Resumen de todos los aspectos

En este apartado se recoge un resumen de las secciones que aparecen en este documento.

- **Introducción:** En esta sección se recogen los motivos por los que se aborda el proyecto, así como los objetivos de este. Además, se estudia la situación actual, buscando alternativas ya existentes a la aplicación que se construirá. Por último, se evalúan distintos aspectos para la elaboración del proyecto, principalmente relacionados con las tecnologías a emplear.

- **Aspectos teóricos:** En este apartado se describen brevemente algunos de los conceptos importantes para el desarrollo del proyecto, así como las tecnologías más importantes que se emplearán.
- **Planificación del Proyecto y Presupuesto Iniciales:** Aquí se realiza un estudio de las tareas que deben ser realizadas para completar el proyecto, incluyendo no solo las actividades relacionadas con la construcción del software en sí, sino también aspectos relacionados con la presente documentación. Se agrupan estas tareas en etapas, se estiman, y se genera un cronograma del proyecto teniendo en cuenta los recursos disponibles. En base a esta planificación y el coste de los recursos se realiza un presupuesto inicial.
- **Análisis:** Se estudian los objetivos y se extraen y documentan los requisitos a partir de estos, generando además una serie de casos de uso identificando los actores del sistema. También se identifican y detallan los subsistemas que componen la aplicación y se realiza un primer diagrama “de clases” de cada uno de estos. Se realizan bocetos de la que se pretende que sea la interfaz del sistema y se indica cómo será la navegación entre pantallas. Por último, se define a alto nivel el plan de pruebas.
- **Diseño:** Se incluyen principalmente diagramas de los dos subsistemas de la aplicación: diagrama de paquetes, de “clases” (con mayor nivel de detalle que el de análisis), diagramas de secuencia de algunas de las acciones que se pueden realizar y diagrama de despliegue. También se muestra el resultado de la construcción de la interfaz a partir de los *mockups* realizados en la fase de análisis. Por último, se detallan las pruebas a realizar.
- **Implementación del sistema:** Se recogen los estándares y normas seguidas, así como los lenguajes y herramientas empleadas en la construcción del sistema. Se indican los problemas que aparecieron en la fase de implementación y su solución o la manera de mitigarlos. También se indica la descripción detallada de las clases en ficheros externos, a los que se enlaza desde aquí.
- **Desarrollo de las pruebas:** En este apartado se explica cómo han sido realizadas las pruebas y los resultados obtenidos en estas.
- **Manuales del sistema:** Se realizan distintos manuales del sistema destinados a diversos perfiles: manual de instalación, de ejecución y despliegue, de usuario y del programador.
- **Conclusiones y Ampliaciones:** Se recogen las conclusiones obtenidas a partir de la realización del proyecto y se proponen posibles ampliaciones o mejoras del sistema que no han sido realizadas por escaparse del alcance.
- **Planificación del Proyecto y Presupuesto finales:** Contiene la misma información que el apartado de *Planificación del Proyecto y Presupuesto Iniciales* adaptando los tiempos y las tareas realizadas a lo que realmente se hizo una vez acabado el proyecto, lo que permite ver el desvío entre lo que se planificó inicialmente y el resultado final en relación con la carga de trabajo y presupuesto.
- **Referencias Bibliográficas:** Recoge el conjunto de referencias bibliográficas recogidas en este documento.
- **Apéndices:** Incluye un glosario de términos y se detalla el contenido entregado.

## 1.3 Originalidad de la obra

Este proyecto es una obra original realizada por Enrique Avello Díaz. Las fuentes empleadas han sido citadas y recogidas en el apartado *Referencias Bibliográficas*.

## Capítulo 2. Introducción

### 2.1 Justificación del proyecto

Uno de los formatos de fichero más empleados en el día a día de cualquier persona es el **PDF** (siglas de **Portable Document Format**). Sus características hacen que su uso sea aconsejable para el intercambio de información, ya que es independiente de la plataforma, existen numerosos visores incluso integrados en los navegadores, las principales suites ofimáticas permiten guardar los documentos creados en este formato y mantienen su estructura independientemente de la aplicación.

Uno de sus inconvenientes principales es la dificultad a la hora de manipularlos. A diferencia de lo que ocurre con otros formatos como, por ejemplo, docx, es complicado añadir páginas a un documento PDF, extraer cierta información de este, unir páginas o cualquier tipo de manipulación que implique modificar el contenido de este.

Por este motivo, se ha decidido desarrollar un producto software que facilite esta labor, permitiendo realizar distintas operaciones sobre documentos PDF a través de una interfaz web de fácil comprensión para usuarios, sean cuales sean sus conocimientos a nivel de informática.

### 2.2 Objetivos del Proyecto

El objetivo principal del proyecto es la creación de una aplicación que pueda ser empleada para la manipulación de ficheros PDF, creando tanto una parte de lógica de negocio, que permita ejecutar las operaciones que se detallarán, como una interfaz web a través de la cual los usuarios realizarán dichas manipulaciones de manera simple.

Por tanto, a nivel de operaciones a realizar los objetivos del proyecto son los siguientes:

- Permitir la eliminación de páginas a un usuario que indica un documento y de alguna forma una o más páginas que desea eliminar del fichero.
- Permitir la extracción de páginas de un documento a un usuario que indica un documento y de alguna forma una o más páginas que desea extraer del fichero.
- Dar al usuario la posibilidad de ordenar las páginas de un documento, indicando el orden en que desea que se encuentren y teniendo opciones predefinidas como la inversión del orden del documento.
- Soportar la unión de una serie de documentos PDF, permitiendo al usuario que indique el orden en que quiere unirlos.
- Permitir al usuario recortar las páginas de un fichero, presentando una imagen de referencia en la que pueda especificar sus límites.

Estos documentos deberán ser procesados y almacenados en el servidor, y una vez realizado el procesamiento, el usuario los descargará.

Además, existen también objetivos a nivel de formación que no están relacionados directamente con las funcionalidades que se pretende construir:

- Aprendizaje de nuevas tecnologías de desarrollo que no han sido utilizadas durante los estudios.
- Desplegar una aplicación constituida por dos subsistemas en la nube, consiguiendo la comunicación entre dichos subsistemas.
- Comprender el desarrollo de webs **SPA**.

En definitiva, como resumen, y de forma más genérica, se podría decir que el objetivo de este proyecto es conseguir realizar y desplegar un producto software útil respaldado por una correcta documentación, para así comprender en qué consisten todas las etapas de construcción de un producto software, atravesando todas ellas.

## 2.3 Estudio de la situación actual

Como paso previo a la elaboración del proyecto, se han estudiado sistemas existentes a la fecha de inicio de este que cumplan parte de los objetivos de este proyecto, teniendo en cuenta en el estudio aplicaciones web o de escritorio que manipulen, de alguna manera, ficheros en formato **PDF**.

Se detallarán las funcionalidades existentes de estas aplicaciones, haciendo hincapié en los aspectos más relevantes de cara al proyecto.

Una vez realizado el estudio de los sistemas existentes similares, se planteará una evaluación de alternativas, valorando diferentes aspectos que influyen en la construcción del sistema.

### 2.3.1 Estudio previo

A continuación, se documentan algunos de los sistemas existentes, que, de alguna manera, podrían tener relación con el producto que se va a desarrollar.

Se valorará una aplicación de escritorio, y dos de las múltiples alternativas que existen a nivel de productos web para la manipulación de este tipo de archivos. El motivo para selección de estas dos aplicaciones es que son las que primero aparecen cuando en el buscador de Google se introducen términos como “unir pdf” o “extraer páginas pdf”, considerando que pueden ser representativas del resto de opciones.

#### 2.3.1.1 PDFtk

Es una aplicación de escritorio. En su versión gratuita está pensada para principalmente mezclar diversos documentos PDF o dividir las páginas de estos en diferentes archivos nuevos.

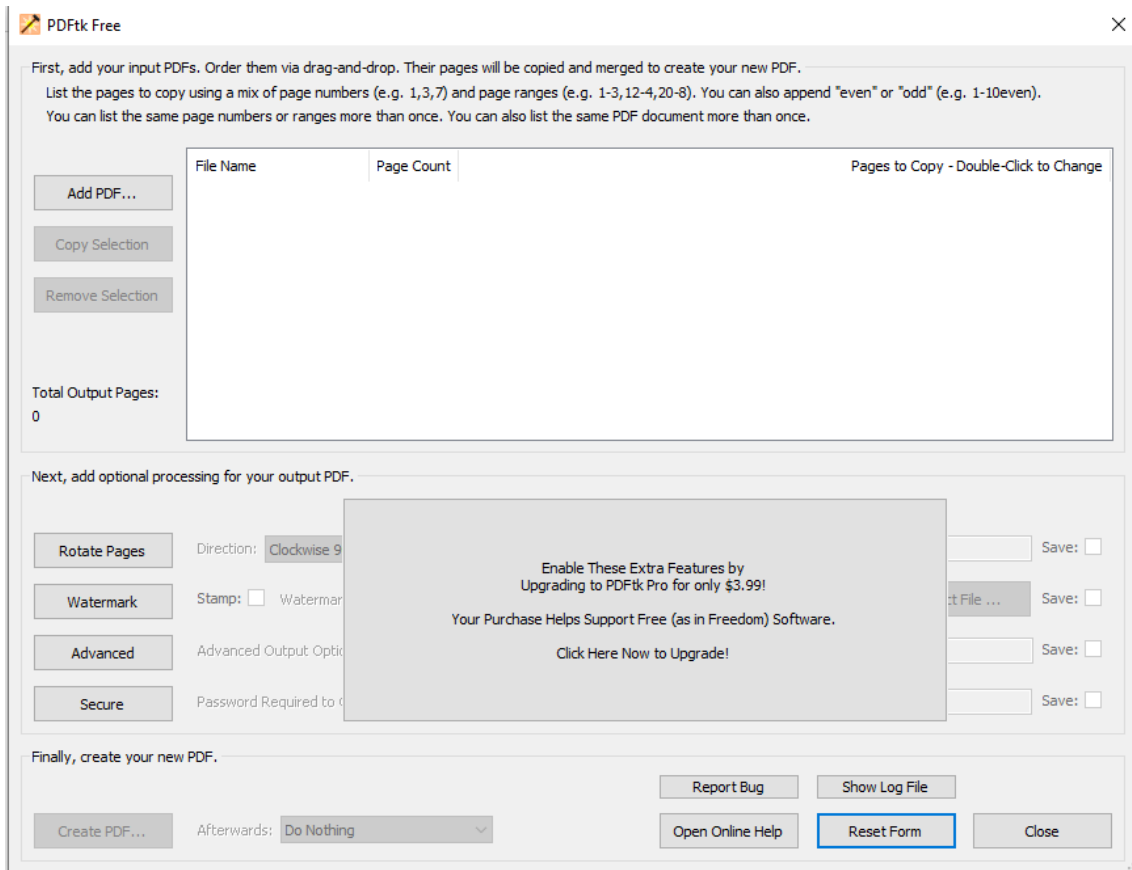


Figura 2.1 Interfaz del programa PDFtk

Por un lado, no tiene todas las funcionalidades que se pretenden conseguir, y además para conseguir alguna de ellas es necesario obtener una versión de pago. Además, se trata de un producto cuya última versión estable data del año 2010. La interfaz gráfica, uno de los aspectos que se van a tener más en cuenta en este proyecto, puede llegar a ser difícil de comprender para usuarios poco experimentados. Como ventaja principal, se puede destacar que, al ser una aplicación de escritorio, el tiempo de espera necesario para realizar una operación es menor, ya que no es necesario enviar los ficheros a ningún servidor.

### 2.3.1.2 *llovepdf.es*

Se trata de una web creada en 2010 por una empresa de Barcelona con un objetivo similar al de este proyecto: proporcionar a sus usuarios una forma sencilla de editar ficheros PDF. Además de esta versión, han creado una versión de escritorio y otra móvil. Según lo que recogen en su web, actualmente cuenta con un equipo de dieciocho personas, de los cuales nueve son ingenieros de software.



Figura 2.2 Pantalla principal de la página ilovepdf

Como ventajas principales destacan el número de funcionalidades que posee y su interfaz clara, que hace simple comprender como se debería trabajar desde la primera vez que se usa.

Como inconvenientes, al tratarse de una aplicación desarrollada por una empresa obviamente tiene fines comerciales, por lo que, para cada operación que se realice se mostrará publicidad. También existen planes de suscripción que otorgan ventajas a los usuarios que pagan una cuota. Además, los documentos generados al realizar las operaciones se borran una vez transcurridas dos horas (en el caso de usuarios registrados en su versión gratuita). Por último, una de las funcionalidades que se pretenden desarrollar, como es el recorte de documentos, no está presente en *ilovepdf*.

Obviamente, el objetivo de este trabajo no es crear una aplicación con todas las funcionalidades que incluye *ilovepdf* por ser algo completamente fuera de alcance. Sin embargo, algunas de las que se desarrollarán sí que serán comunes.

### 2.3.1.3 PDFCandy

Como en el caso anterior, se trata de una aplicación web con un gran número de opciones, no solo para la manipulación, sino también para la conversión en otros formatos.

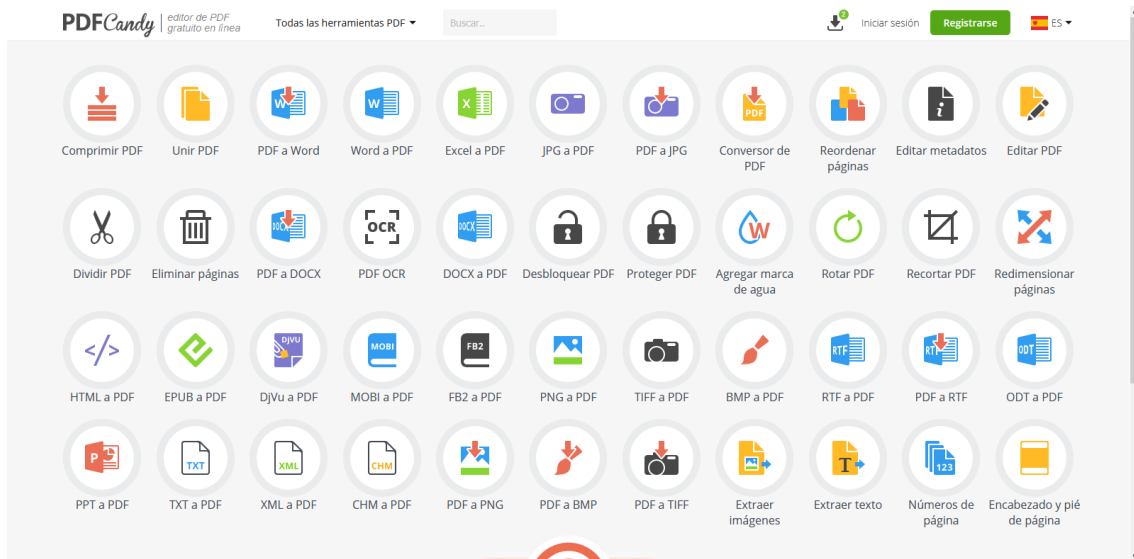


Figura 2.3 Interfaz de la página PDFCandy

De nuevo, la principal ventaja es el número de opciones soportadas, que en este caso sí incluye la funcionalidad del recorte del documento. Sin embargo, la forma de presentarlas no es tan clara. Aparece demasiada información en la pantalla principal y puede generar confusión. Además, una vez seleccionada una opción, la forma de indicar los parámetros de la operación, pese a ser correcta, no es tan intuitiva como en la aplicación anterior.

La principal desventaja viene de nuevo del aspecto comercial, y es incluso más flagrante que en el caso anterior. Todas las opciones de registro tienen un coste asociado, ya sea con un pago único o mensual y los usuarios no registrados solo pueden hacer dos tareas cada dos horas.



Figura 2.4 Limitaciones de PDFCandy

Por último, como ocurría en *ilovepdf* los documentos generados solo se guardan durante dos horas.



## 2.3.2 Evaluación de alternativas

En este apartado se estudiarán las distintas alternativas que han sido tenidas en cuenta para la realización del proyecto y que se consideran de mayor importancia, detallando los motivos que para justificar la selección final.

### 2.3.2.1 Arquitectura del sistema

Para la arquitectura de la aplicación se valoraron principalmente dos opciones: una arquitectura “sin servidor” o una arquitectura cliente-servidor con *frontend* y *backend* diferenciados.

La arquitectura sin servidor tiene como ventaja principal la sencillez de integración. Al tener solo un sistema que desarrollar no habría que preocuparse de la comunicación entre los sistemas y permitiría la obtención de un producto inicial más rápido. No obstante, aunque no se trata de una aplicación especialmente exigente con la base de datos, los ficheros generados y datos de los usuarios se van a tener que guardar en algún lugar, por lo que seguirían existiendo dependencias externas.

La arquitectura cliente-servidor tradicional tiene la principal ventaja de estar muy consolidada, por lo que existen multitud de recursos para consultar buenas prácticas a la hora de aplicarla. Además, permite distribuir de una mejor manera las responsabilidades, facilitando la mantenibilidad.

Teniendo en cuenta estos aspectos, he decidido emplear una arquitectura cliente-servidor.

### 2.3.2.2 Lenguaje de programación backend

Como alternativas para la realización del *backend* se han valorado Java, JavaScript y Python, por ser aquellos en los que el desarrollador tiene conocimiento en mayor o menor medida.

No obstante, para la selección del lenguaje no se ha valorado únicamente el grado de experiencia empleando el mismo, sino que, dada la tipología del proyecto, se ha tenido muy en cuenta la existencia de bibliotecas que permiten manipular ficheros PDF, analizando tanto la facilidad de aprendizaje como la velocidad al realizar determinadas acciones.

Por tanto, en principio se han tenido en cuenta las siguientes combinaciones lenguaje-biblioteca:

- **Java + PDFBox.**
- **Javascript + PDF.js.**
- **Python + pyPDF2.**

La primera alternativa descartada es la de JavaScript + PDF.js, ya que, a pesar de tener la ventaja de ser el lenguaje que empleo día a día en mi trabajo, la librería es la más compleja de comprender de las tres.

Entre las alternativas Java o Python, a nivel de experiencia estoy más habituado a emplear Java, pero la librería parece más simple en el caso de Python.

Finalmente, realizo una comparación de tiempos de ejecución para los procesos de unir dos documentos y extraer varias páginas de uno, para comprobar si alguna de las alternativas penalizaría demasiado el rendimiento, creando pequeños scripts de prueba.

Realizando cada una de las operaciones 10 veces, con los mismos ficheros y mismas páginas obtengo los siguientes tiempos:

Operación	Lenguaje	Tiempo Medio (N=10) (ms)
Unión	PDFBox + Java	2485,54
	Python + Pypdf2	3581,98
Extracción	PDFBox + Java	3545,02
	Python + Pypdf2	1023,34

En la unión es más rápida la librería de Java, mientras que en la extracción lo es la de Python, pero estas diferencias tampoco son muy grandes.

Teniendo en cuenta estas ventajas e inconvenientes de cada combinación y el interés personal en aprender a realizar aplicaciones de cierto tamaño con Python, decido emplear este último lenguaje.

### 2.3.2.3 Frameworks para el frontend

Se ha decidido utilizar un *framework* que permita realizar SPA (*Single Page Applications*) para construir el *frontend*. Las alternativas valoradas son las tres principales para realizar este tipo de aplicaciones hoy en día: Angular, React y Vue.

En el aspecto de experiencia de uso, antes de empezar el proyecto no había trabajado con ninguno de los tres. Según diversas comparaciones consultadas en la web, por ejemplo [1], Vue tiene la curva de aprendizaje más ligera, siendo Angular el más costoso de comprender. React, tiene su propia sintaxis (JSX), hecho que complica la asimilación de conceptos. Este es un punto importante, ya que, en un proyecto con el tiempo bastante limitado, poder aprender una nueva tecnología en un breve periodo de tiempo es fundamental.

En cuanto al tamaño, Vue es el más pequeño de los tres, pesando unas seis veces menos que Angular [2], por lo que su tiempo de carga será el menor también.

Una de las ventajas y a la vez problemáticas de Vue, es que es el más flexible de los tres [1]. Esto hace que no haya que seguir ninguna arquitectura concreta, como sí que obliga Angular. Por este motivo, puede llevar a soluciones de escasa calidad si no se toman las decisiones cuidadosamente.

Teniendo en cuenta estos aspectos, he decidido emplear Vue, dándole especial importancia a la facilidad de aprendizaje.

## Capítulo 3. Aspectos teóricos

En este capítulo recopilare las herramientas, conceptos y tecnologías que tienen importancia en el desarrollo del proyecto, describiéndolas brevemente.

### 3.1 Conceptos

En este subapartado quedaran detallados aquellos conceptos relevantes en el proyecto por determinar patrones a seguir o formas de comunicación entre componentes de la aplicación.

#### 3.1.1 REST



**Figura 3.1 Imagen de REST**

Las siglas provienen del inglés *representational state transfer*, transferencia de estado representacional en español [3]. Constituye un estilo de arquitectura software que, sin ser un estándar, hace uso de distintos estándares, como HTTP o URI [4]. Con esta aproximación, tanto datos como funcionalidad son recursos a los que se accede empleando las mencionadas URI. Los datos transmitidos a través de REST no tienen un formato obligatorio. Podría transmitirse información en JSON, XML o incluso texto simple [4].

Se empleará en el proyecto como sistema de comunicación entre *backend* y *frontend*.

#### 3.1.2 MVC

Se trata de un estilo arquitectónico cuya base de funcionamiento es la separación de los datos (modelo), interfaz de usuario (vista) y lógica de control (controlador), en tres componentes diferenciados, estableciendo un flujo de comunicación entre dichos componentes [5]. Es un estilo con muchos años de historia, y existen gran número de *framework* que implementan el patrón [6].

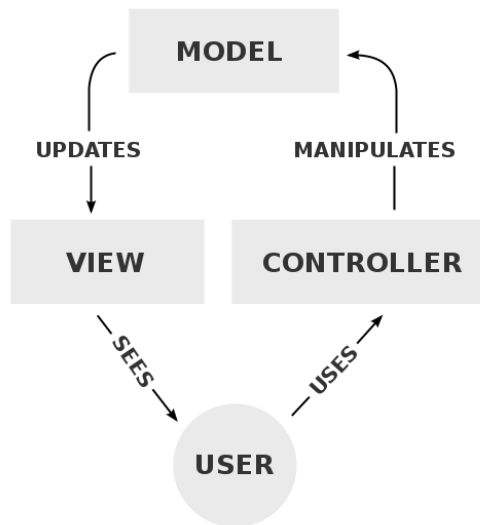


Figura 3.2 Diagrama explicativo del patrón MVC

En el proyecto a desarrollar tanto el *backend* como el *frontend* emplean variantes surgidas de este patrón.

### 3.1.3 SPA

SPA, del inglés “*Single page Application*”, o Aplicación de Página Única en español, es un tipo de aplicación en la web en la cual toda la información se sirve en una sola página. De esta manera, no es necesario cargar a través del servidor las diferentes páginas, sino que los recursos precargados se van mostrando dinámicamente, proporcionando una mejor experiencia de usuario al reducir los tiempos de espera [7].

Existen numerosos *framework* que permiten realizar aplicaciones de este tipo, siendo Vue uno de ellos. Por tanto, en este proyecto, se empleará esta aproximación.

## 3.2 Tecnologías

En este subapartado quedarán detallados los *framework* o librerías empleadas en el desarrollo del proyecto. Obviamente no se recogerán todas las librerías utilizadas, solo aquellas que tienen importancia por ser globales o estar relacionadas con el objetivo del proyecto.

### 3.2.1 Django



**Figura 3.3 Logo de Django**

Django es un *framework* de código abierto para el desarrollo web en el lenguaje Python, construido bajo la idea del patrón MVC. Creado y lanzado en el año 2005 por Adrian Holovaty y Simon Willison, es mantenido por la Django Software Foundation [8].

- Página web: <https://www.djangoproject.com/>
- Versión utilizada: 3.1.4.

En el proyecto ha sido empleado para la implementación de la API Rest que permite realizar operaciones con los ficheros.

### 3.2.2 Vue



**Figura 3.4 Logo De Vue**

Vue es un *framework* del lenguaje Javascript basado en componentes para la creación de SPA [9]. Sigue el patrón arquitectónico Model-view-viewmodel [9]. Fue creado por Evan You en el año 2014 y está en continua evolución desde entonces, siendo mantenido por el conocido como “*The Core Team*”, que incluye al propio Evan [10].

- Página web: <https://vuejs.org/>
- Repositorio GitHub: <https://github.com/vuejs/vue>
- Versión utilizada: 2.5.2.

En el proyecto ha sido empleado para la realización del *frontend* de la aplicación.

### 3.2.3 PyPDF2

PyPDF2 es una librería de Python destinada a la manipulación de ficheros PDF, para realizar distintas acciones como extraer información del documento, recortar páginas, unir distintos

documentos en uno... [11] Fue lanzado a finales de 2011 partiendo de una primera versión, llamada PyPDF, que había sido creada en 2005 por Mathieu Fenniak [12].

- Página web: <http://mstamy2.github.io/PyPDF2/>
- Repositorio GitHub: <https://github.com/mstamy2/PyPDF2>
- Versión utilizada: 1.26.0.

En el proyecto ha sido utilizada para la manipulación de los ficheros PDF.

### 3.2.4 BootstrapVue



**Figura 3.5 Logo de BootstrapVue**

BootstrapVue es una librería que integra la conocida librería Bootstrap con el Vue, ofreciendo componentes que pueden ser empleados para la maquetación de la aplicación, siendo además compatibles con las directivas del *framework* [13].

- Página Web: <https://bootstrap-vue.org/>
- Repositorio GitHub: <https://github.com/bootstrap-vue/bootstrap-vue>
- Versión utilizada: 2.0.11.

En el proyecto se ha utilizado para la maquetación de las distintas vistas de la web, empleando sus componentes predefinidos.

## Capítulo 4. Planificación del Proyecto y Presupuesto iniciales

En este apartado se realizará una planificación temporal del proyecto para posteriormente, en base a dicha planificación, realizar un presupuesto inicial.

### 4.1 Planificación inicial

La duración estimada inicialmente para realizar la totalidad del proyecto es de 320 horas, iniciándose el mismo el 15 de diciembre de 2020 y finalizando el 11 de junio de 2021.

La planificación ha sido realizada empleando la herramienta Microsoft Project y se encuentra en el archivo PlanTFGINicial.mpp, dentro de la carpeta adjunta Planificación y Presupuesto.

En la planificación se ha tenido en cuenta el siguiente calendario laboral:

- Días de trabajo: lunes a viernes.
- Horario: 17:30 a 20:00.

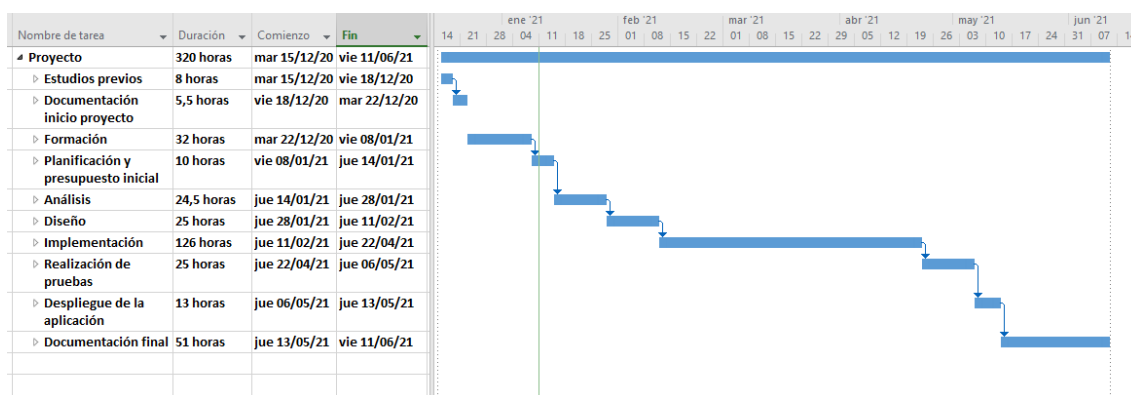
Se ha dividido el proyecto en diez grupos de tareas o etapas, que se detallan en la siguiente tabla, incluyendo duración y fechas de comienzo y fin estimadas en base a esta duración y el calendario laboral mencionado anteriormente:

Nombre de tarea	Duración	Comienzo	Fin
<b>Estudios previos</b>	<b>8 horas</b>	<b>mar 15/12/20</b>	<b>vie 18/12/20</b>
<b>Documentación inicio proyecto</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>mar 22/12/20</b>
<b>Formación</b>	<b>32 horas</b>	<b>mar 22/12/20</b>	<b>vie 08/01/21</b>
<b>Planificación y presupuesto inicial</b>	<b>10 horas</b>	<b>vie 08/01/21</b>	<b>jue 14/01/21</b>
<b>Análisis</b>	<b>24,5 horas</b>	<b>jue 14/01/21</b>	<b>jue 28/01/21</b>
<b>Diseño</b>	<b>25 horas</b>	<b>jue 28/01/21</b>	<b>jue 11/02/21</b>
<b>Implementación</b>	<b>126 horas</b>	<b>jue 11/02/21</b>	<b>jue 22/04/21</b>
<b>Realización de pruebas</b>	<b>25 horas</b>	<b>jue 22/04/21</b>	<b>jue 06/05/21</b>
<b>Despliegue de la aplicación</b>	<b>13 horas</b>	<b>jue 06/05/21</b>	<b>jue 13/05/21</b>

<b>Documentación final</b>	<b>51 horas</b>	<b>jue 13/05/21</b>	<b>vie 11/06/21</b>
----------------------------	-----------------	---------------------	---------------------

Se puede observar en la tabla que existen tres etapas que se realizan antes que la planificación inicial: Estudios previos, Documentación inicio proyecto y Formación. Esto es así porque para poder realizar la planificación, y sobre todo la estimación de las tareas a ejecutar, era necesario tener conocimiento de algunos aspectos como son el alcance, los objetivos, alternativas seleccionadas y por supuesto, cierto nivel de formación en las tecnologías que se van a emplear. Como consecuencia, la duración que aparece para estas etapas no es estimada, sino que es el tiempo real consumido en su realización.

En la siguiente imagen se presenta un diagrama de Gantt de alto nivel del proyecto en base a los grupos de tareas anteriores:



**Figura 4.1 Actividades y diagrama de Gantt de alto nivel**

A continuación, mostraré con mayor nivel de detalle estos grupos de tareas, explicando en que consiste cada uno de ellos y que subtareas los componen.

### 4.1.1 Estudios previos

Contiene aquellas tareas que deben ser ejecutadas antes de realizar cualquier avance en el proyecto, ya que aportan información imprescindible para poder continuar.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1</b>	<b>Proyecto</b>	<b>320 horas</b>	<b>mar 15/12/20</b>	<b>vie 11/06/21</b>
<b>1.1</b>	<b>Estudios previos</b>	<b>8 horas</b>	<b>mar 15/12/20</b>	<b>vie 18/12/20</b>
1.1.1	Estudio de los objetivos del proyecto	2 horas	mar 15/12/20	mar 15/12/20
1.1.2	Búsqueda de información sobre librerías	2 horas	mié 16/12/20	mié 16/12/20



	para modificación de ficheros PDF			
1.1.3	Realización de pruebas con diferentes librerías	4 horas	mié 16/12/20	vie 18/12/20

#### 4.1.2 Documentación inicio proyecto

Hace referencia a las tareas de documentación de la introducción del TFG. Dentro de este grupo se encuentra el estudio de la situación actual, que tiene especial interés, ya que define algunas de las herramientas que se emplearán en el proyecto, las cuales afectan a esta planificación.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.2</b>	<b>Documentación inicio proyecto</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>mar 22/12/20</b>
<b>1.2.1</b>	<b>Introducción</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>mar 22/12/20</b>
1.2.1.1	Justificación	0,5 horas	vie 18/12/20	vie 18/12/20
1.2.1.2	Objetivo	0,5 horas	vie 18/12/20	vie 18/12/20
<b>1.2.1.3</b>	<b>Estudio situación actual</b>	<b>4,5 horas</b>	<b>vie 18/12/20</b>	<b>mar 22/12/20</b>
1.2.1.3.1	Estudio previo	1,5 horas	vie 18/12/20	lun 21/12/20
1.2.1.3.2	Evaluación de alternativas	3 horas	lun 21/12/20	mar 22/12/20

#### 4.1.3 Formación

Contiene el conjunto de tareas relativas a la formación en tecnologías que han sido seleccionadas en la evaluación de alternativas y en las que no se tiene un grado de conocimiento suficiente como para acometer el proyecto sin una introducción previa.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.3</b>	<b>Formación</b>	<b>32 horas</b>	<b>mar 22/12/20</b>	<b>vie 08/01/21</b>
1.3.1	Formación en Vue	12 horas	mar 22/12/20	mar 29/12/20
1.3.2	Formación en Django	12 horas	mar 29/12/20	mar 05/01/21

1.3.3	Formación Integración Vue+Django	8 horas	mar 05/01/21	vie 08/01/21
-------	----------------------------------	---------	--------------	--------------

#### 4.1.4 Planificación y presupuesto inicial

Conformado por las tareas dedicadas a la definición de este WBS, la estimación de las tareas y la realización del presupuesto inicial. En resumen, este capítulo de la documentación.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.4</b>	<b>Planificación y presupuesto inicial</b>	<b>10 horas</b>	<b>vie 08/01/21</b>	<b>jue 14/01/21</b>
1.4.1	Realización del WBS inicial	3 horas	vie 08/01/21	lun 11/01/21
1.4.2	Estimar actividades del WBS	2 horas	lun 11/01/21	mar 12/01/21
1.4.3	Presupuesto inicial	5 horas	mar 12/01/21	jue 14/01/21

#### 4.1.5 Análisis

Contiene las tareas necesarias para realizar el análisis del sistema que se va a construir.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.5</b>	<b>Análisis</b>	<b>24,5 horas</b>	<b>jue 14/01/21</b>	<b>jue 28/01/21</b>
1.5.1	Lectura en detalle de la información aportada por el tutor	1 hora	jue 14/01/21	jue 14/01/21
1.5.2	Establecimiento del alcance del sistema	2 horas	jue 14/01/21	vie 15/01/21
<b>1.5.3</b>	<b>Análisis de requisitos</b>	<b>12 horas</b>	<b>vie 15/01/21</b>	<b>vie 22/01/21</b>
1.5.3.1	Extracción de requisitos	6 horas	vie 15/01/21	mar 19/01/21
1.5.3.2	Definición de requisitos	6 horas	mié 20/01/21	vie 22/01/21
1.5.4	Análisis de casos de uso	5 horas	vie 22/01/21	mar 26/01/21
1.5.5	Identificación de los subsistemas	1 hora	mar 26/01/21	mar 26/01/21

1.5.6	Análisis de interfaces de usuario	2 horas	mar 26/01/21	mié 27/01/21
1.5.7	Análisis del plan de pruebas	1,5 horas	mié 27/01/21	jue 28/01/21

#### 4.1.6 Diseño

Grupo de tareas que han de ser realizadas para establecer el diseño del sistema que se va a construir.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.6</b>	<b>Diseño</b>	<b>25 horas</b>	<b>jue 28/01/21</b>	<b>jue 11/02/21</b>
<b>1.6.1</b>	<b>Definición de la arquitectura del sistema</b>	<b>8 horas</b>	<b>jue 28/01/21</b>	<b>mar 02/02/21</b>
<b>1.6.1.1</b>	<b>Realización de diagramas</b>	<b>8 horas</b>	<b>jue 28/01/21</b>	<b>mar 02/02/21</b>
1.6.1.1.1	Diagrama de paquetes	2 horas	jue 28/01/21	jue 28/01/21
1.6.1.1.2	Diagrama de componentes	4 horas	vie 29/01/21	lun 01/02/21
1.6.1.1.3	Diagrama de despliegue	2 horas	lun 01/02/21	mar 02/02/21
1.6.2	Diseño de clases	12 horas	mar 02/02/21	mar 09/02/21
1.6.3	Diseño de interfaces de usuario	5 horas	mar 09/02/21	jue 11/02/21

#### 4.1.7 Implementación

Conformado por todas las tareas necesarias para codificar la aplicación web, tanto *frontend* como *backend* y tareas de configuración de los entornos para poder realizar dicha implementación.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.7</b>	<b>Implementación</b>	<b>126 horas</b>	<b>jue 11/02/21</b>	<b>jue 22/04/21</b>
1.7.1	Instalación de los entornos de trabajo	2 horas	jue 11/02/21	jue 11/02/21
<b>1.7.2</b>	<b>Inicialización</b>	<b>6 horas</b>	<b>vie 12/02/21</b>	<b>mar 16/02/21</b>
1.7.2.1	Inicialización del proyecto frontend	3 horas	vie 12/02/21	lun 15/02/21

1.7.2.2	Inicialización del proyecto backend	3 horas	lun 15/02/21	mar 16/02/21
1.7.3	Definición de modelos y serializadores backend	3 horas	mar 16/02/21	mié 17/02/21
1.7.4	Pantalla de inicio	3 horas	mié 17/02/21	jue 18/02/21
<b>1.7.5</b>	<b>Sistema de inicio de sesión</b>	<b>8 horas</b>	<b>jue 18/02/21</b>	<b>mar 23/02/21</b>
1.7.5.1	Backend Inicio de sesión	4 horas	jue 18/02/21	lun 22/02/21
1.7.5.2	Frontend inicio de Sesión	4 horas	lun 22/02/21	mar 23/02/21
<b>1.7.6</b>	<b>Sistema de registro</b>	<b>6 horas</b>	<b>mié 24/02/21</b>	<b>vie 26/02/21</b>
1.7.6.1	Frontend sistema de registro	2 horas	mié 24/02/21	mié 24/02/21
1.7.6.2	Backend sistema de registro	4 horas	mié 24/02/21	vie 26/02/21
<b>1.7.7</b>	<b>Eliminación/Extracción de páginas</b>	<b>22 horas</b>	<b>vie 26/02/21</b>	<b>jue 11/03/21</b>
1.7.7.1	Backend Eliminación/Extracción de páginas	12 horas	vie 26/02/21	vie 05/03/21
1.7.7.2	Frontend Eliminación/Extracción de páginas	10 horas	vie 05/03/21	jue 11/03/21
<b>1.7.8</b>	<b>Mezcla de documentos</b>	<b>16 horas</b>	<b>jue 11/03/21</b>	<b>vie 19/03/21</b>
1.7.8.1	Backend Mezcla de documentos	6 horas	jue 11/03/21	lun 15/03/21
1.7.8.2	Frontend mezcla de documentos	10 horas	lun 15/03/21	vie 19/03/21
<b>1.7.9</b>	<b>Ordenación de documentos</b>	<b>18 horas</b>	<b>vie 19/03/21</b>	<b>mar 30/03/21</b>
1.7.9.1	Backend ordenación de documentos	8 horas	vie 19/03/21	mié 24/03/21
1.7.9.2	Frontend ordenación de documentos	10 horas	mié 24/03/21	mar 30/03/21
<b>1.7.10</b>	<b>Recorte de documentos</b>	<b>28 horas</b>	<b>mar 30/03/21</b>	<b>mié 14/04/21</b>

1.7.10.1	Backend recorte de documentos	12 horas	mar 30/03/21	mar 06/04/21
1.7.10.2	Frontend recorte de documentos	16 horas	mar 06/04/21	mié 14/04/21
<b>1.7.11</b>	<b>Historial de operaciones</b>	<b>9 horas</b>	<b>jue 15/04/21</b>	<b>mar 20/04/21</b>
1.7.11.1	Backend historial de operaciones	3 horas	jue 15/04/21	vie 16/04/21
1.7.11.2	Frontend historial de operaciones	6 horas	vie 16/04/21	mar 20/04/21
1.7.12	Sistema de ruteo y navegación	5 horas	mar 20/04/21	jue 22/04/21

#### 4.1.8 Realización de pruebas

Contiene las tareas enfocadas a la realización de pruebas sobre el sistema construido.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.8</b>	<b>Realización de pruebas</b>	<b>25 horas</b>	<b>jue 22/04/21</b>	<b>jue 06/05/21</b>
1.8.1	Pruebas unitarias	10 horas	jue 22/04/21	mié 28/04/21
1.8.2	Resto de pruebas	15 horas	mié 28/04/21	jue 06/05/21

#### 4.1.9 Despliegue de la aplicación

Hace referencia a las tareas que han de ser realizadas para poner la aplicación en un servidor de producción.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.9</b>	<b>Despliegue de la aplicación</b>	<b>13 horas</b>	<b>jue 06/05/21</b>	<b>jue 13/05/21</b>
1.9.1	Estudio de alternativas y formación para el despliegue	3 horas	jue 06/05/21	vie 07/05/21
1.9.2	Despliegue	10 horas	vie 07/05/21	jue 13/05/21

#### 4.1.10 Documentación final

Grupo de tareas que incluye todas aquellas de documentación que se realizarán una vez finalizada la construcción de la aplicación y su puesta en producción. Constituye la mayor parte de esta documentación.

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.10</b>	<b>Documentación final</b>	<b>51 horas</b>	<b>jue 13/05/21</b>	<b>vie 11/06/21</b>
1.10.1	Detallar aspectos teóricos	3 horas	jue 13/05/21	vie 14/05/21
<b>1.10.2</b>	<b>Documentación sobre implementación</b>	<b>7 horas</b>	<b>lun 17/05/21</b>	<b>mié 19/05/21</b>
1.10.2.1	Estándares y normas seguidos	1 hora	lun 17/05/21	lun 17/05/21
1.10.2.2	Lenguajes utilizados	2 horas	lun 17/05/21	mar 18/05/21
1.10.2.3	Herramientas y programas empleados en el desarrollo	2 horas	mar 18/05/21	mar 18/05/21
1.10.2.4	Creación del sistema	2 horas	mié 19/05/21	mié 19/05/21
1.10.3	Documentación sobre el desarrollo de las pruebas	10 horas	mié 19/05/21	mar 25/05/21
<b>1.10.4</b>	<b>Manuales del sistema</b>	<b>11 horas</b>	<b>mar 25/05/21</b>	<b>mar 01/06/21</b>
1.10.4.1	Manual de instalación	2 horas	mar 25/05/21	mié 26/05/21
1.10.4.2	Manual de ejecución	2 horas	mié 26/05/21	jue 27/05/21
1.10.4.3	Manual de usuario	5 horas	jue 27/05/21	lun 31/05/21
1.10.4.4	Manual del programador	2 horas	lun 31/05/21	mar 01/06/21
<b>1.10.5</b>	<b>Conclusiones y ampliaciones</b>	<b>3,5 horas</b>	<b>mar 01/06/21</b>	<b>mié 02/06/21</b>
1.10.5.1	Conclusiones	1,5 horas	mar 01/06/21	mar 01/06/21
1.10.5.2	Ampliaciones	2 horas	mar 01/06/21	mié 02/06/21
1.10.6	Referencias bibliográficas	1 hora	mié 02/06/21	mié 02/06/21

<b>1.10.7</b>	<b>Apéndices</b>	<b>5 horas</b>	<b>jue 03/06/21</b>	<b>vie 04/06/21</b>
1.10.7.1	Glosario	2 horas	jue 03/06/21	jue 03/06/21
1.10.7.2	Contenido entregado	1 hora	jue 03/06/21	vie 04/06/21
1.10.7.3	Índice alfabético	1 hora	vie 04/06/21	vie 04/06/21
1.10.7.4	Código fuente	1 hora	vie 04/06/21	vie 04/06/21
1.10.8	Memoria del proyecto	2 horas	lun 07/06/21	lun 07/06/21
<b>1.10.9</b>	<b>Planificación y presupuesto final</b>	<b>8,5 horas</b>	<b>lun 07/06/21</b>	<b>vie 11/06/21</b>
1.10.9.1	Adaptación WBS	2,5 horas	lun 07/06/21	mar 08/06/21
1.10.9.2	Registro tiempos totales	1 hora	mar 08/06/21	mié 09/06/21
1.10.9.3	Presupuesto final	5 horas	mié 09/06/21	vie 11/06/21

## 4.2 Presupuesto de costes inicial

En este apartado se presentará el cálculo del presupuesto de costes inicial, realizado a partir de la planificación detallada en el apartado anterior.

Para hacer este presupuesto se ha tenido en cuenta lo establecido en la Guía de Aprendizaje de la asignatura DPPI.

### 4.2.1 Modelo de empresa

Para realizar el cálculo del presupuesto de costes se ha establecido un modelo de empresa, con el fin de justificar los costes de precio/hora del perfil que realizará el trabajo. Obviamente, por las características del proyecto, se trata de una simplificación de una empresa orientada a proyectos, con un único trabajador.

#### 4.2.1.1 Personal

Como se ha mencionado existe un único empleado, un Ingeniero de Software con un sueldo bruto anual de 24000€.

A partir de estos datos se calcula el coste salarial de este empleado para la empresa. Este está calculado incrementando un 30% el valor del sueldo bruto anual. Se utiliza este porcentaje para tener en cuenta los gastos derivados del pago de las cuotas de la seguridad social y periodos vacacionales.

Por otra parte, se asigna una productividad al trabajador del 80%. Esto se debe a que podrían existir bloqueos, reuniones o descansos que no computarían dentro del tiempo del trabajo en proyectos del empleado.

A partir de todo esto se calculan los costes directos e indirectos generados por este trabajador, reflejados en la siguiente tabla:

Personal	Número	Sueldo Bruto/año	Coste salarial/año	Productividad	Coste Directo	Coste Indirecto (%)	Coste indirecto
Ingeniero de Software	1	24.000 €	31.200 €	80%	24.960 €	20%	6.240 €

#### 4.2.1.2 Costes indirectos de los medios de producción

Existen una serie de medios de producción, tanto físicos como software, necesarios para la realización de proyectos, que computan como costes indirectos de la empresa. Estos son, en el caso de este TFG, el ordenador portátil empleado y las licencias de Microsoft Office y Microsoft Project. En la siguiente tabla se reflejan estos costes, así como su tipo:

Costes de los medios de producción					
Medio	Unidades	Precio	Coste anual	Tipo	Plazo (años)
Portátil Lenovo IdeaPad	1	699 €	174,8 €	Amortizado	4
Licencia Microsoft Project	1		303,6 €	Alquiler	
Licencia Office 365	1		105,60 €	Alquiler	
<b>TOTAL</b>			<b>583,95 €</b>		

#### 4.2.1.3 Otros costes indirectos

Por último, existen una serie de costes indirectos que se producirían por el funcionamiento de la empresa. Estos costes se recogen en la siguiente tabla:

Otros Costes Indirectos		
Descripción	Coste/mes	Coste/año
Limpieza	200 €	2.400 €
Asesoría	50 €	600 €
Tributos y tasas diversas	300 €	3.600 €



Alquileres	500 €	6.000 €
Gastos de mantenimiento	30 €	360 €
Pequeño utillaje y herramientas	20 €	240 €
Consumos de electricidad	50 €	600 €
Consumo de agua	20 €	240 €
Primas de seguros	200 €	2.400 €
Portes y gastos de transporte	100 €	1.200 €
Gastos en comunicaciones	80 €	960 €
Gatos de viajes/desplazamientos	20 €	240 €
<b>Total</b>	<b>1.570 €</b>	<b>18.840 €</b>

#### 4.2.1.4 Coste total anual

Una vez efectuados los cálculos de los tres apartados anteriores se puede obtener el coste total anual de la empresa, a partir de los costes indirectos, costes directos y beneficios que se espera obtener.

Los costes indirectos están compuestos por los obtenidos a partir de las horas no productivas del empleado, los costes de los medios de producción y los recogidos en la tabla de otros costes indirectos. Por tanto, el resultado final es este:

Costes indirectos	
Concepto	Importe
Costes indirectos salariales	6.240,00 €
Costes indirectos de los medios de producción	583,95 €
Otros costes indirectos	18.840 €
<b>TOTAL</b>	<b>25.663,95 €</b>

Los costes directos son únicamente aquellos que provienen del coste salarial del empleado en sus horas productivas, es decir **24.960 €**.

Sobre el total de estos costes (**50.623,95€**) se calculará el beneficio que espera obtener la empresa, que se supondrá que es de un 25%, y se obtendrá el coste total anual, conformado por la suma de costes directos, costes indirectos y el beneficio esperado. Esto se detalla en la siguiente tabla:

Costes totales	
Concepto	Importe
Costes directos	24.960 €
Costes indirectos	25.663,95 €
Beneficio (sobre costes directos + indirectos)	12.655,99 €
<b>TOTAL</b>	<b>63.279,94 €</b>

#### 4.2.1.5 Precio hora del empleado

Teniendo en cuenta estos datos, se define el precio por hora que se debería cobrar a un cliente para obtener una facturación que iguale o supere los costes calculados.

Con un precio por hora de 40€ se podría conseguir una facturación de **64.256€**, que supera los costes. La facturación se calcula multiplicando el número de horas productivas, basada en la productividad del 80% mencionada anteriormente, por el precio hora. En la siguiente tabla se definen estos datos:

Personal	Productividad	Horas/año	Horas Prod/Año	Precio/Hora	Facturación	Precio/Hora (Sin Beneficios)
Ingeniero de Software	80%	2008	1606,4	40 €	64.256,00 €	30,00 €

La última columna presenta el precio/hora que se empleará para el cálculo del presupuesto de costes. Este valor se calcula como el 75% del precio hora utilizado para obtener la facturación, ya que el 25% corresponde a los beneficios que se esperan obtener, y al tratarse de un presupuesto de costes interno no se tendrán en cuenta.

#### 4.2.1.6 Margen entre coste anual y facturación

Por último, se calcula el margen que hay entre la facturación obtenida con un precio hora de 40€ y los costes totales. Este margen se refleja en la siguiente tabla:

Margen entre costes y facturación		
Concepto	Importe	Margen (%)
Coste total	63.279,94 €	
Facturación	64.256,00 €	
Margen	976,06 €	1,52%

## 4.2.2 Desarrollo del presupuesto de costes inicial

Se han considerado 4 partidas para la realización del presupuesto:

- Partida 1: Puesta en marcha (Estudios y planificación inicial).
- Partida 2: Análisis y Diseño del sistema.
- Partida 3: Implementación, pruebas y despliegue.
- Partida 4: Documentación.

A continuación, se desarrollará cada una de las partidas, mostrando para las tareas y grupos de tareas la carga horaria, el coste por hora y el precio total. El coste por hora, dadas las características del proyecto es, para todas las tareas, el calculado en el apartado anterior sin beneficios (30€/hora).

Las estimaciones horarias para cada trabajo son las que se han definido en la planificación.

Todas las partidas del presupuesto se corresponden con tareas planificadas en el apartado de planificación. Dado que el proyecto será realizado íntegramente desde un mismo lugar de trabajo, y no serán necesarios desplazamientos, no se incluyen otros posibles costes, como podrían ser estos desplazamientos o las dietas derivadas de los mismos.

En cada partida se especifica el nivel de tarea. Por ejemplo, la tarea *Estudios previos* es de nivel 1 (n1) y su coste aparecerá en la columna Subtotal(n1). Esta tiene subtareas de nivel 2 (n2) como *Estudio de los objetivos del proyecto*, cuyo subtotal aparece en la columna Subtotal(n2). En caso de que una tarea tenga subtareas, su subtotal será la suma de los subtotales de estas últimas.

En todas las partidas los costes reflejados no tienen el IVA incluido.

4.2.2.1 Partida 1: Puesta en marcha (Estudios y planificación inicial)

Incluye los estudios realizados con anterioridad al comienzo del proyecto, la realización de la planificación, este presupuesto y un periodo de formación.

<b>Partida 1: Puesta en marcha (Estudios y planificación inicial)</b>						
n1	n2	Descripción	Horas	Precio/Hora	Subtotal(n2)	Subtotal(n1)
1		<b>Estudios previos</b>				<b>240 €</b>
	1	Estudio de los objetivos del proyecto	2	30 €	60 €	
	2	Búsqueda de información de librerías para la modificación de ficheros pdf	2	30 €	60 €	
	3	Realización de pruebas con diferentes librerías	4	30 €	120 €	
2		<b>Formación</b>				<b>960 €</b>
	1	Formación en Vue	12	30 €	360 €	
	2	Formación en Django	12	30 €	360 €	
	3	Formación Integración Vue+Django	8	30 €	240 €	
3		<b>Planificación y presupuesto inicial</b>				<b>300 €</b>
	1	Realización del WBS inicial	3	30 €	90 €	
	2	Estimar actividades del WBS	2	30 €	60 €	
	3	Presupuesto inicial	5	30 €	150 €	
<b>Total</b>						<b>1.500 €</b>

## 4.2.2.2 Partida 2: Análisis y Diseño del Sistema

Esta partida incluye aquellas tareas que pertenecen a la fase de análisis y diseño del sistema:

Partida 2: Análisis y Diseño del sistema										
n1	n2	n3	n4	Descripción	Horas	Precio/Hora	Subtotal(n4)	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1				<b>Análisis</b>						<b>735 €</b>
	1			Lectura en detalle de la información aportada por el tutor	1	30 €			<b>30 €</b>	
	2			Establecimiento del alcance del sistema	2	30 €			<b>60 €</b>	
	3			<b>Análisis de requisitos</b>					<b>360 €</b>	
		1		Extracción de requisitos	6	30 €		<b>180 €</b>		
		2		Definición de requisitos	6	30 €		<b>180 €</b>		
	4			Análisis de casos de uso	5	30 €			<b>150 €</b>	
	5			Identificación de los subsistemas	1	30 €			<b>30 €</b>	
	6			Análisis de interfaces de usuario	2	30 €			<b>60 €</b>	
	7			Análisis del plan de pruebas	1,5	30 €			<b>45 €</b>	
2				<b>Diseño</b>						<b>750 €</b>
	1			<b>Definición de la arquitectura</b>					<b>240 €</b>	
		1		Realización de diagramas				<b>240 €</b>		

		1	Diagrama de paquetes	2	30 €	60 €		
		2	Diagrama de componentes	4	30 €	120 €		
		3	Diagrama de despliegue	2	30 €	60 €		
	2		Diseño de clases	12	30 €			<b>360 €</b>
	3		Diseño de interfaces de usuario	5	30 €			<b>150 €</b>

<b>Total</b>	<b>1.485 €</b>
--------------	----------------

4.2.2.3 Partida 3: Implementación, pruebas y despliegue

Esta partida estará conformada por todas las tareas de implementación del producto, pruebas de distinto tipo y puesta en producción del sistema:

<b>Partida 3: Implementación, pruebas y despliegue</b>								
n1	n2	n3	Descripción	Horas	Precio/Hora	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1			<b>Implementación</b>					<b>3.780 €</b>
	1		Instalación de los entornos de trabajo	2	30 €		<b>60 €</b>	
	2		<b>Inicialización</b>				<b>180 €</b>	
		1	Inicialización del proyecto frontend	3	30 €	90 €		
		2	Inicialización del proyecto backend	3	30 €	90 €		
	3		Definición de modelos y serializadores backend	3	30 €		<b>90 €</b>	
	4		Pantalla de inicio	3	30 €		<b>90 €</b>	

5		<b>Sistema de inicio de sesión</b>				<b>240 €</b>
	1	Backend Inicio de sesión	4	30 €	120 €	
	2	Frontend inicio de sesión	4	30 €	120 €	
6		<b>Sistema de registro</b>				<b>180 €</b>
	1	Frontend sistema de registro	2	30 €	60 €	
	2	Backend sistema de registro	4	30 €	120 €	
7		<b>Eliminación /Extracción de páginas</b>				<b>660 €</b>
	1	Backend Eliminación/Extracción de páginas	12	30 €	360 €	
	2	Frontend Eliminación/Extracción de páginas	10	30 €	300 €	
8		<b>Mezcla de documentos</b>				<b>480 €</b>
	1	Backend Mezcla de documentos	6	30 €	180 €	
	2	Frontend Mezcla de documentos	10	30 €	300 €	
9		<b>Ordenación de documentos</b>				<b>540 €</b>
	1	Backend ordenación de documentos	8	30 €	240 €	
	2	Frontend ordenación de documentos	10	30 €	300 €	
10		<b>Recorte de documentos</b>				<b>840 €</b>
	1	Backend recorte de documentos	12	30 €	360 €	
	2	Frontend recorte de documentos	16	30 €	480 €	

11		<b>Historial de operaciones</b>				<b>270 €</b>
	1	Backend historial de operaciones	3	30 €	90 €	
	2	Frontend historial de operaciones	6	30 €	180 €	
12		Sistema de ruteo y navegación	5	30 €	150	<b>150 €</b>
2		<b>Realización de pruebas</b>				<b>750 €</b>
	1	Pruebas unitarias	10	30 €		<b>300 €</b>
	2	Resto de pruebas	15	30 €		<b>450 €</b>
3		<b>Despliegue de la aplicación</b>				<b>390 €</b>
	1	Estudio de alternativas y formación para el despliegue	3	30 €		<b>90 €</b>
	2	Despliegue	10	30 €		<b>300 €</b>

<b>Total</b>	<b>4.920 €</b>
--------------	----------------

#### 4.2.2.4 Partida 4: Documentación

Esta partida incluye todos los aspectos del proyecto que deben de ser documentados, tanto aquella documentación que se puede realizar antes de comenzar a analizar el sistema como aquella que necesita que el sistema esté implementado.



Partida 4: Documentación										
n1	n2	n3	n4	Descripción	Horas	Precio/Hora	Subtotal(n4)	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1				<b>Documentación inicio proyecto</b>						<b>165 €</b>
	2			<b>Introducción</b>					<b>165 €</b>	
		1		Justificación	0,5	30 €		15 €		
		2		Objetivo	0,5	30 €		15 €		
		3		<i>Estudio situación actual</i>				135 €		
		1		Estudio previo	1,5	30 €	45 €			
		2		Evaluación de alternativas	3	30 €	90 €			
2				<b>Documentación final</b>						<b>1.530 €</b>
	1			Detallar aspectos teóricos	3	30 €			<b>90 €</b>	
	2			<b>Documentación sobre implementación</b>					<b>210 €</b>	
		1		Estándares y normas seguidos	1	30 €		30 €		
		2		Lenguajes utilizados	2	30 €		60 €		
		3		Herramientas y programas empleados en el desarrollo	2	30 €		60 €		
		4		Creación del sistema	2	30 €		60 €		
	3			Documentación sobre el desarrollo de las pruebas	10	30 €			<b>300 €</b>	

4		<b>Manuales del sistema</b>					<b>330 €</b>
	1	Manual de instalación	2	30 €		60 €	
	2	Manual de ejecución	2	30 €		60 €	
	3	Manual de usuario	5	30 €		150 €	
	4	Manual del programador	2	30 €		60 €	
5		<b>Conclusiones y ampliaciones</b>					<b>105 €</b>
	1	Conclusiones	1,5	30 €		45 €	
	2	Ampliaciones	2	30 €		60 €	
6		Referencias bibliográficas	1	30 €			<b>30 €</b>
7		<b>Apéndices</b>					<b>150 €</b>
	1	Glosario	2	30 €		60 €	
	2	Contenido entregado	1	30 €		30 €	
	3	Índice alfabético	1	30 €		30 €	
	4	Código fuente	1	30 €		30 €	
8		Memoria del proyecto	2	30 €			<b>60 €</b>
9		<b>Planificación y presupuesto final</b>					<b>255 €</b>
	1	Adaptación WBS	2,5	30 €		75 €	
	2	Registro tiempos totales	1	30 €		30 €	

		3	Presupuesto final	5	30 €		150 €		
								<b>Total</b>	<b>1.695 €</b>

#### 4.2.2.5 Resumen presupuesto de costes inicial

En resumen, agrupando los costes por partidas, el presupuesto resultante sería el siguiente:

<b>Presupuesto de costes</b>	
<b>Partida</b>	<b>Importe</b>
1. Puesta en marcha (Estudios y planificación inicial)	1.500 €
2. Análisis y Diseño del sistema	1.485 €
3. Implementación, pruebas y despliegue	4.920 €
4. Documentación	1.695 €
<b>TOTAL</b>	<b>9.600 €</b>

## Capítulo 5. Análisis

### 5.1 Definición del Sistema

#### 5.1.1 Determinación del alcance del sistema

En resumen, el sistema a desarrollar será una aplicación que permita realizar distintas operaciones sobre ficheros con extensión pdf. Estas operaciones serán: extraer páginas de un documento, eliminarlas, unir una serie de documentos (2 o más), alterar el orden de las páginas de un fichero y recortar el tamaño de las páginas. Además, los usuarios deberán registrarse en el sistema, de tal manera que, una vez que realicen manipulaciones sobre los documentos, estos se guardarán asociados a cada usuario para su posterior consulta.

El sistema se dividirá en dos partes principales, una aplicación web que se mostrará a los usuarios y en el que podrán introducir los parámetros de las operaciones a desarrollar, y un *backend*, que expondrá una API Rest y será el encargado de realizar el procesado. De esta manera, si en un futuro se decide cambiar la aplicación web o realizar una aplicación móvil, en el servidor no sería necesario realizar cambios.

Cada operación se mostrará en una página independiente en la que se podrán añadir los archivos. Una vez añadidos, se presentarán las opciones específicas relacionadas con la operación. Existirán además páginas para consultar el historial de operaciones, iniciar sesión en el sistema y registrarse en este.

Las pantallas de operaciones serán accesibles tanto desde una página de inicio como de una barra de navegación. El resto de las pantallas serán accesibles a través de una barra de navegación.

### 5.2 Requisitos del sistema

#### 5.2.1 Obtención de los requisitos del sistema

##### 5.2.1.1 Requisitos funcionales

En este apartado se recogerán los requisitos funcionales del sistema, dividiéndolos según el apartado de la aplicación al que pertenecen.

##### 5.2.1.1.1 Requisitos de registro e inicio de sesión

RF-1. Un usuario podrá registrarse en el sistema.

RF-1.1. El usuario debe aportar las credenciales con las que se desea registrar.

RF-1.1.1. Nombre de usuario.

RF-1.1.2. Contraseña.

RF-1.1.3. Repetición de contraseña.

RF-1.2. El sistema comprobará que el nombre de usuario no está registrado.

RF-1.2.1. Si el nombre ya estuviese registrado se avisará al usuario de esta circunstancia.

- RF-1.2.2. Si el nombre ya estuviese registrado no se almacenará ningún dato del usuario.
  - RF-1.3. El sistema verificará que las contraseñas aportadas coinciden.
    - RF-1.3.1. En caso de no coincidir se mostrará un aviso indicándolo.
  - RF-1.4. Se verificará que la contraseña tenga al menos NUM\_CARACTERES\_CONTRASEÑA caracteres.
    - RF-1.4.1. El valor inicial de NUM\_CARACTERES\_CONTRASEÑA será 8.
    - RF-1.4.2. En caso de no verificarse se indicará al usuario la circunstancia.
  - RF-1.5. Si las validaciones anteriores se cumplen el sistema guardará los datos del usuario:
    - RF-1.5.1. Nombre de usuario.
    - RF-1.5.2. Contraseña hasheada.
  - RF-1.6. Una vez completado el proceso de registro, se informará al usuario de que este ha sido completado.
  - RF-1.7. El sistema redirigirá al usuario a la pantalla de inicio de sesión.
- RF-2. Un usuario registrado podrá iniciar sesión en el sistema.
- RF-2.1. El usuario deberá aportar las credenciales.
    - RF-2.1.1. Nombre de usuario.
    - RF-2.1.2. Contraseña.
  - RF-2.2. El sistema verificará que las credenciales son correctas.
    - RF-2.2.1. Se comprobará que el usuario existe en la base de datos.
      - RF-2.2.1.1. Si no existe, se informará al usuario de que no se puede iniciar sesión.
    - RF-2.2.2. Se comprobará que la contraseña aportada coincida con la almacenada para el usuario.
      - RF-2.2.2.1. Si no coincide, se informará que el inicio de sesión no ha sido satisfactorio.
  - RF-2.3. Si se valida correctamente el RF2.2 se guarda en sesión al usuario.
  - RF-2.4. El sistema redirigirá al usuario a la página de inicio.
- RF-3. Un usuario con sesión iniciada podrá salir de sesión.
- RF-3.1. Se eliminará al usuario de la sesión del sistema.
  - RF-3.2. Se informará al usuario de que su sesión ha sido cerrada de forma satisfactoria.
  - RF-3.3. El usuario pasará a tener acceso únicamente a los apartados de la aplicación que no requieren que el usuario tenga sesión iniciada.
- 5.2.1.1.2 [Requisitos de procesamiento de ficheros](#)
- RF-4. El sistema permitirá el guardado de ficheros PDF resultado de procesamiento.
- RF-4.1. El sistema guardará el PDF creado en su sistema de ficheros.
  - RF-4.2. El sistema almacenará los datos del fichero en la base de datos:
    - RF-4.2.1. Id del fichero.
      - RF-4.2.1.1. Este campo lo generará automáticamente el sistema al realizar el guardado.
    - RF-4.2.2. URL del fichero en el sistema de ficheros.
    - RF-4.2.3. Tamaño del fichero.

- RF-4.2.4. Fecha de creación.
  - RF-4.2.5. Usuario creador.
- RF-5. El sistema permitirá al usuario extraer páginas de un fichero PDF.
- RF-5.1. El usuario deberá tener sesión iniciada en el sistema.
  - RF-5.2. El sistema presentará la opción de seleccionar un fichero del equipo local.
    - RF-5.2.1. El sistema comprobará que el fichero seleccionado es un fichero en formato PDF.
      - RF-5.2.1.1. Si el fichero no fuese un PDF el sistema no permitirá realizar la acción.
  - RF-5.3. El sistema obtendrá el número de páginas que tiene el fichero.
  - RF-5.4. El sistema permitirá al usuario definir las páginas que desea extraer.
    - RF-5.4.1. El sistema permitirá la selección de estas páginas como elementos individuales.
    - RF-5.4.2. El sistema permitirá la selección de rangos de páginas.
    - RF-5.4.3. El sistema validará que el formato en el que el usuario indica las páginas a escoger es correcto.
    - RF-5.4.4. La representación de las páginas que se extraerán será a través de una cadena de texto.
      - RF-5.4.4.1. La cadena de texto deberá cumplir la siguiente expresión regular:  $/^{\left(\left([1-9] + [0-9]^*\right)\left(\left(,\right)\left(-\right)\left([1-9]+[0-9]^*\right)\right)^*\right)\$}/$ .
        - RF-5.4.4.1.1. En el caso de que la cadena no cumpla la expresión no se podrá realizar la operación.
  - RF-5.5. El sistema validará que no se seleccionen páginas cuyo número sea mayor que el de páginas totales del documento.
    - RF-5.5.1. Si existen páginas que no cumplen esto simplemente no se añadirán al PDF resultante.
  - RF-5.6. El sistema creará un nuevo PDF que contenga las páginas seleccionadas.
  - RF-5.7. El sistema guardará los datos del fichero tal y como se indica en RF-4.
  - RF-5.8. El sistema permitirá al usuario descargar el fichero resultado de la extracción.
- RF-6. El sistema permitirá al usuario eliminar páginas de un fichero PDF.
- RF-6.1. El usuario deberá tener sesión iniciada en el sistema.
  - RF-6.2. El sistema presentará la opción de seleccionar un fichero del equipo local.
    - RF-6.2.1. El sistema comprobará que el fichero seleccionado es un fichero en formato PDF.
      - RF-6.2.1.1. Si el fichero no fuese un PDF el sistema no permitirá realizar la acción.
  - RF-6.3. El sistema obtendrá el número de páginas que tiene el fichero.
  - RF-6.4. El sistema permitirá al usuario definir las páginas que desea eliminar.
    - RF-6.4.1. El sistema permitirá la selección de estas páginas como elementos individuales.
    - RF-6.4.2. El sistema permitirá la selección de rangos de páginas.
    - RF-6.4.3. El sistema validará que el formato en el que el usuario indica las páginas a escoger es correcto.

- RF-6.4.4. La representación de las páginas que se extraerán será a través de una cadena de texto.
    - RF-6.4.4.1. La cadena de texto deberá cumplir la siguiente expresión regular:  $/^{\wedge}(((1-9) + [0-9] *) ((,|-)((1-9)+[0-9]*))^*)\$/$ .
      - RF-6.4.4.1.1. En el caso de que la cadena no cumpla la expresión no se podrá realizar la operación.
  - RF-6.5. El sistema validará que no se seleccionen páginas cuyo número sea mayor que el de páginas totales del documento.
    - RF-6.5.1. Si existen páginas que no cumplen esto simplemente no se añadirán al PDF resultante.
  - RF-6.6. El sistema creará un nuevo PDF que contenga las páginas del documento inicial excluyendo las seleccionadas.
  - RF-6.7. El sistema guardará los datos del fichero tal y como se indica en RF-4.
  - RF-6.8. El sistema permitirá al usuario descargar el fichero resultante tras la eliminación de páginas.
- 
- RF-7. El sistema permitirá al usuario la unión de documentos PDF.
    - RF-7.1. El usuario deberá tener sesión iniciada en el sistema.
    - RF-7.2. El sistema presentará la opción de seleccionar ficheros del equipo del usuario.
      - RF-7.2.1. Todos los ficheros deberán pertenecer al mismo directorio en el equipo del usuario.
      - RF-7.2.2. El sistema validará que todos los ficheros seleccionados sean de formato PDF.
        - RF-7.2.2.1. En caso de que alguno de los ficheros no sea PDF, el sistema no permitirá realizar la acción.
      - RF-7.2.3. El sistema verificará que al menos se han seleccionado dos ficheros.
        - RF-7.2.3.1. En caso de seleccionar menos de dos ficheros el sistema no permitirá realizar la operación.
    - RF-7.3. El sistema permitirá al usuario seleccionar el orden el que desea unir los ficheros.
      - RF-7.3.1. Por defecto, este orden será en el que el usuario ha seleccionado los ficheros de su equipo.
    - RF-7.4. El sistema creará un nuevo fichero uniendo los ficheros seleccionados en el orden estipulado por el usuario.
    - RF-7.5. El sistema guardará el fichero creado tal y como se indica en RF-4.
    - RF-7.6. El sistema permitirá al usuario la descarga del fichero resultado de la unión.
  - RF-8. El sistema permitirá al usuario la ordenación de las páginas de un fichero PDF.
    - RF-8.1. El usuario deberá tener sesión iniciada en el sistema.
    - RF-8.2. El sistema presentará la opción de seleccionar un fichero del equipo del usuario.
      - RF-8.2.1. El sistema comprobará que el fichero seleccionado es un fichero en formato PDF.
        - RF-8.2.1.1. Si el fichero no fuese un PDF el sistema no permitirá realizar la acción.
    - RF-8.3. El sistema cargará el número de páginas que tiene el fichero.



- RF-8.4. El sistema presentará N elementos, cada uno de los cuales representará un número de página.
    - RF-8.4.1. N será el número de páginas del fichero.
    - RF-8.4.2. Cada elemento tendrá una posición X que representará la posición final de la página en el documento.
      - RF-8.4.2.1. Inicialmente, posición y número de página coincidirán.
        - RF-8.4.2.1.1. La página 1 estará en la posición 1, la página n-1 en la posición n-1 y así para todas las páginas del fichero.
  - RF-8.5. El usuario podrá cambiar el orden de los elementos.
    - RF-8.5.1. Cualquier elemento podrá pasar de su posición X a otra posición Y.
  - RF-8.6. El usuario tendrá la opción predefinida de invertir el orden del documento.
    - RF-8.6.1. En este caso, el elemento 0 pasaría a la posición N y viceversa, el elemento 1 a la posición N-1 y viceversa, y así sucesivamente.
  - RF-8.7. El sistema creará un nuevo fichero con el orden especificado por el usuario.
    - RF-8.7.1. Se recorrerá la lista de elementos ordenados generada por el usuario.
      - RF-8.7.1.1. Cada elemento contendrá el número de página que se desea colocar en la posición del elemento.
        - RF-8.7.1.2. Se añadirá la página en la posición indicada por el elemento.
  - RF-8.8. El sistema guardará el fichero creado tal y como se indica en RF-4.
  - RF-8.9. El sistema permitirá al usuario la descarga del fichero resultado de la ordenación.
- 
- RF-9. El sistema permitirá al usuario recortar el área visible de un documento PDF.
    - RF-9.1. El usuario deberá tener sesión iniciada en el sistema.
    - RF-9.2. El sistema presentará la opción de seleccionar un fichero del equipo del usuario.
      - RF-9.2.1. El sistema comprobará que el fichero seleccionado es un fichero en formato PDF.
        - RF-9.2.1.1. Si el fichero no fuese un PDF el sistema no permitirá realizar la acción.
    - RF-9.3. El sistema cargará una previsualización de la primera página del documento.
    - RF-9.4. El sistema permitirá al usuario seleccionar los límites por los que desea cortar el documento, basándose en la previsualización anterior. Se definirá:
      - RF-9.4.1. Límite superior izquierdo.
      - RF-9.4.2. Anchura.
      - RF-9.4.3. Altura.
    - RF-9.5. El resultado de la selección del recorte deberá tener un tamaño inferior que el del documento original.
    - RF-9.6. El sistema transformará los datos de la selección a coordenadas:
      - RF-9.6.1. Se obtendrá la coordenada (x,y) del punto situado en la parte superior izquierda del recuadro definido.
      - RF-9.6.2. Se obtendrá la anchura del recuadro definido.
      - RF-9.6.3. Se obtendrá la altura del recuadro definido.
    - RF-9.7. Con los datos obtenidos en el requisito RF-9.6 se recorrerán las páginas del documento.
      - RF-9.7.1. Para cada página se establecerá un nuevo punto superior izquierdo.
      - RF-9.7.2. Para cada página se establecerá un nuevo punto inferior derecho.

- RF-9.7.3. Se añadirán las páginas con la nueva caja definida a un nuevo documento.
- RF-9.8. El sistema guardará el fichero creado tal y como se indica en RF-4.
- RF-9.9. El sistema permitirá al usuario la descarga del fichero resultado de la ordenación.

#### 5.2.1.1.3 Listado de documentos

- RF-10. Un usuario podrá consultar el listado de documentos que ha generado usando la aplicación.
  - RF-10.1. El usuario deberá haber iniciado sesión en el sistema.
  - RF-10.2. Un usuario únicamente podrá ver los ficheros que han sido generados por él.
  - RF-10.3. Los resultados se mostrarán paginados.
  - RF-10.4. Se mostrará información de los ficheros:
    - RF-10.4.1. Nombre.
    - RF-10.4.2. Tamaño.
    - RF-10.4.3. Tiempo que hace que el fichero se ha añadido.
  - RF-10.5. Un usuario podrá descargar un fichero del listado.
    - RF-10.5.1. Se obtendrá la URL del fichero dentro del sistema de ficheros.
    - RF-10.5.2. Se descargará el fichero desde esa URL.
  - RF-10.6. Un usuario podrá eliminar ficheros desde el listado.
    - RF-10.6.1. El usuario podrá seleccionar uno o más ficheros para eliminar.
      - RF-10.6.1.1. Existirá una opción para seleccionar todos los ficheros de una página.
    - RF-10.6.2. Antes de realizar el borrado se solicitará confirmación al usuario para realizar esta acción.
    - RF-10.6.3. Para cada fichero seleccionado, el sistema borrará la información asociada:
      - RF-10.6.3.1. Se borrará la entrada en la base de datos asociada al fichero.
      - RF-10.6.3.2. Se borrará el propio fichero del sistema de ficheros del servidor.
  - RF-10.7. El sistema mostrará el tamaño total de los ficheros almacenados.

#### 5.2.1.2 Requisitos no funcionales

- RNF-1. Los tiempos de carga de las pantallas que no requieran comunicación con el *backend* serán inferiores a dos segundos.
- RNF-2. El tiempo de procesamiento de un fichero de menos de 100 páginas en la ordenación ha de ser menor de 15 segundos.
- RNF-3. El *backend* de la aplicación estará implementado en el lenguaje Python en la versión 3.8.1.
  - RNF -3.1. Se empleará el *framework* Django, versión 3.1.4 para la realización de la API Rest.

RNF-4. El *frontend* de la aplicación se implementará usando el *framework* Vue.js, en su versión 2.5.2.

RNF-5. La aplicación podrá ser utilizada en los navegadores actuales con mayor cuota de mercado:

- RNF -5.1. Google Chrome en su versión 89.0.4389.128 y posteriores.
- RNF -5.2. Mozilla Firefox en su versión 87.0 y posteriores.
- RNF -5.3. Microsoft Edge en su versión 90.0.818.42 y posteriores.

RNF-6. Ambos subsistemas estarán desplegados en un servidor en la nube.

## 5.2.2 Identificación de actores del sistema

En el sistema únicamente existirán dos tipos de actores, usuario con sesión iniciada y usuario sin sesión iniciada.

- **Usuario sin sesión iniciada:** Engloba tanto a los usuarios que tienen una cuenta en la aplicación, pero no tienen sesión iniciada, como a aquellos que no están registrados. Tendrán acceso a la parte pública de la aplicación, es decir, podrán consultar las opciones que ofrece, iniciar sesión si ya tienen una cuenta, o registrarse si aún no la tienen.
- **Usuario con sesión iniciada:** Además de las opciones que se presentan para un usuario que no tiene sesión iniciada, salvo la de registrarse, tendrán acceso a todas las funcionalidades que impliquen la manipulación de ficheros, así como consultar los ficheros que han ido generado con el uso de la aplicación.

## 5.2.3 Especificación de Casos de uso

En este apartado se presentarán los casos de uso que tienen mayor relevancia en la aplicación, mostrándolos inicialmente en un diagrama para posteriormente detallar cada uno de ellos.

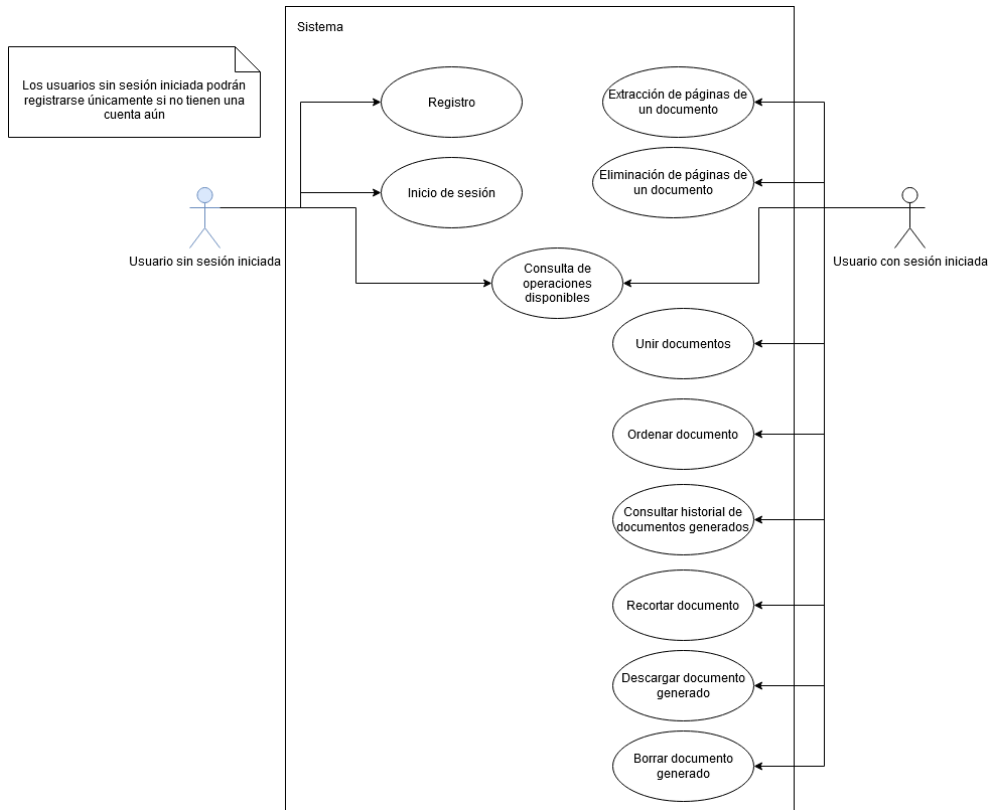


Figura 5.1 Diagrama de casos de uso

<b>Nombre del caso de uso</b>
Registro
<b>Descripción</b>
Un usuario sin sesión iniciada, y que aún no tenga cuenta en el sistema, podrá registrarse en el sistema proporcionando unas credenciales: nombre de usuario y contraseña. Esta última dos veces para verificar que no se ha cometido ningún error al mecanizarla.

<b>Nombre del caso de uso</b>
Inicio de sesión
<b>Descripción</b>
Un usuario sin sesión iniciada y que tenga una cuenta registrada, podrá iniciar sesión en el sistema introduciendo su combinación de credenciales usuario/contraseña. Al realizar esto pasará a comportarse como un usuario con sesión iniciada y tendrá acceso a todas las opciones existentes para este tipo de usuarios.

<b>Nombre del caso de uso</b>
-------------------------------

Consulta de operaciones disponibles
-------------------------------------

<b>Descripción</b>
--------------------

Cualquier usuario, independientemente del estado de su sesión, podrá consultar, desde la página de inicio de la aplicación, qué operaciones se ofrecen para la manipulación de ficheros pdf.
--

<b>Nombre del caso de uso</b>
-------------------------------

Extracción de páginas de un documento
---------------------------------------

<b>Descripción</b>
--------------------

Un usuario con sesión iniciada podrá extraer páginas de un documento pdf que tenga almacenado en su equipo. El sistema cargará la información del documento aportado y presentará una forma de seleccionar las páginas que desea extraer. El programa, con la información del documento y conjunto de páginas a extraer generará un nuevo documento, que incluirá únicamente las páginas seleccionadas, y se lo devolverá al usuario, pudiendo este último descargarlo.
---

<b>Nombre del caso de uso</b>
-------------------------------

Eliminación de páginas de un documento
--

<b>Descripción</b>
--------------------

Un usuario con sesión iniciada podrá eliminar páginas de un documento que tenga almacenado en su equipo. El sistema cargará la información del documento y presentará una forma de seleccionar las páginas que desea eliminar. El sistema, con la información del documento y conjunto de páginas a eliminar generará un nuevo documento, que incluirá todas las páginas del documento aportado excepto las indicadas, y se lo devolverá al usuario para que proceda a su descarga si así lo desea.
---

<b>Nombre del caso de uso</b>
-------------------------------

Unir documentos
-----------------

<b>Descripción</b>
--------------------

Un usuario en sesión podrá unir dos o más ficheros que tenga almacenados en su equipo local en un único documento. El sistema proporcionará la opción de seleccionar el orden en el que desea que se unan. Posteriormente, tomará los documentos y el orden especificado para
---

generar uno nuevo que incluya todos los anteriores en ese orden. Una vez realizado el proceso, el usuario podrá descargar el documento generado.

<b>Nombre del caso de uso</b>
-------------------------------

Ordenar documento
-------------------

<b>Descripción</b>
--------------------

Un usuario con sesión iniciada podrá alterar el orden de las páginas que conforman un documento que tenga guardado. El sistema obtendrá el número de páginas del documento y permitirá que se especifique el orden en el que se desea que aparezcan las páginas. La aplicación tomará el documento y el orden especificado para generar un nuevo documento que las incluya en ese orden. Una vez realizado el proceso, se presentará la opción de descargar el fichero generado.

<b>Nombre del caso de uso</b>
-------------------------------

Recortar documento
--------------------

<b>Descripción</b>
--------------------

Un usuario en sesión podrá recortar el área visible de un documento. El sistema, una vez seleccionado el fichero, mostrará una previsualización de la primera página del documento con un recuadro superpuesto que representará el área de recorte. Se obtendrán las coordenadas del recuadro seleccionado y, a partir de esto, el sistema modificará los límites del documento para que pasen a ser coincidentes con el recuadro para todas las páginas. Una vez realizada la modificación de los límites, el usuario podrá descargar el documento generado.

<b>Nombre del caso de uso</b>
-------------------------------

Consultar historial de documentos generados
---

<b>Descripción</b>
--------------------

Un usuario con sesión iniciada podrá visualizar los documentos que ha generado con el uso de la aplicación en cada operación realizada y que aún no haya borrado. Para cada archivo se mostrará:

- Nombre del documento en el sistema.
- Tamaño del fichero.
- Fecha en la que se generó.

<b>Nombre del caso de uso</b>
Borrar documento generado
<b>Descripción</b>
Desde el listado de documentos generados mencionado en el caso de uso “Consultar historial de documentos generados”, un usuario con sesión iniciada tendrá la opción de eliminar uno o más documentos de este historial.

<b>Nombre del caso de uso</b>
Descargar documento generado
<b>Descripción</b>
Además de poder descargar el archivo en el momento en el que se realiza la operación, un usuario con sesión activa tendrá la posibilidad de descargar aquellos ficheros que ha generado, siempre y cuando no hayan sido eliminados. Esta operación se podrá realizar desde el listado de documentos generados.

## 5.3 Identificación de los subsistemas en la Fase de Análisis

En este apartado se explicarán, desde el punto de vista del análisis, los diferentes subsistemas que componen el sistema que se pretende construir. En primer lugar, se describirán estos subsistemas, para posteriormente especificar las interfaces que permiten la comunicación entre ellos.

### 5.3.1 Descripción de los subsistemas

En un primer análisis de alto nivel se identifican claramente dos subsistemas. Por un lado, el *backend* o servidor web de la aplicación, que se construirá empleando en lenguaje Python y el *framework* Django, y por otro lado el *frontend*, construido sobre Vue.js.

Se presentarán estos subsistemas, descomponiendo a su vez cada uno de ellos en los módulos que lo componen.

#### 5.3.1.1 Servidor web

Los módulos del servidor web se ven influidos por el *framework* utilizado, que propone una arquitectura concreta siguiendo un patrón MVT [14]. No obstante, el sistema no se limita a seguir esta arquitectura, se añaden elementos que la complementan. A continuación, se detallan estos módulos.

#### 5.3.1.1.1 Settings

Se encarga, como su nombre indica, de la configuración del servidor, tratando aspectos como activación/desactivación del modo de depuración, gestión de políticas CORS, establecimiento de las librerías que se utilizarán en el proyecto, configuración sobre las bases de datos, autenticación, validaciones e internacionalización.

#### 5.3.1.1.2 Router

Contiene los elementos que gestionan el ruteo, asociando una URL a una vista, concepto que se explicará a continuación.

#### 5.3.1.1.3 Vistas

Una vista es un fragmento de código que recibe una petición o *request* y devuelve una respuesta o *response*. Este fragmento de código tiene una URL asociada que hace que se ejecute y puede ser desde una simple función, hasta una clase que responde a distintos tipos de peticiones HTTP. En este proyecto se gestionarán con *Viewsets*, elemento proporcionado por Django-REST [15]. Son similares a lo que en otros marcos de trabajo se conoce como Controlador.

#### 5.3.1.1.4 Procesadores

Este no es un elemento del *framework*, siendo específico para esta aplicación. Es el subsistema o paquete encargado de lo que es propiamente el procesamiento de ficheros. Recibirá la información de la *request* y devolverá el fichero generado junto a otra información que emplearán los serializadores, que se exponen a continuación.

#### 5.3.1.1.5 Serializadores

Otro elemento de Django. Son transformadores de datos que convierten instancias de modelos (se explicará este concepto posteriormente) o resultados de consultas en tipos de datos nativos de Python y viceversa, creando instancias de estos modelos a partir de datos básicos, pudiendo realizar validaciones en el proceso [16].

#### 5.3.1.1.6 Modelos

Los modelos son la representación de los datos que se almacenarán en la base de datos del sistema. En un modelo se definirán los campos que deben persistirse de un objeto y el tipo que tendrán estos [17]. Por lo tanto, se podría considerar que un modelo corresponde con una tabla si estamos hablando de bases de datos relacionales. En el sistema no tendrán gran relevancia porque no se trata de una aplicación que haga uso intensivo de la base de datos, pero sí se utilizarán para almacenar la información de los pdf generados, así como de los usuarios.

### 5.3.1.2 Aplicación web SPA

En el caso de la aplicación web, los distintos módulos en los que se dividirá no están tan influidos por Vue.js, ya que otorga mayor libertad para que el desarrollador decida cómo plantear la arquitectura. Aun así, el empleo de este *framework* obliga a que sus archivos tengan una estructura. Por ejemplo, los ficheros de Vue incluyen tanto HTML, como código JavaScript para la gestión y, en algunos casos, CSS para los estilos. A continuación, se detallan estos módulos.



#### 5.3.1.2.1 Componentes

Los componentes son la base no solo de Vue si no de una de las arquitecturas más empleadas hoy en día, que es la arquitectura de componentes web [18]. Son básicamente fragmentos de código, que incluyen de manera general plantilla, comportamiento y estilos, pudiendo ser reutilizados. En el sistema se hará uso, de forma general, de la librería BootstrapVue, que contiene gran cantidad de componentes preparados construidos sobre Bootstrap. No obstante, también se crearán algunos componentes personalizados, basados en los proporcionados por dicha librería.

#### 5.3.1.2.2 Vistas

Serán las pantallas de la aplicación. Estarán conformadas por distintos componentes y en ellas se gestionará todo lo relacionado con la recogida, muestra de datos y las transformaciones para las llamadas a los servicios, módulo que se explica a continuación.

#### 5.3.1.2.3 Servicios

Son, desde el punto de vista del *frontend*, el punto de conexión entre este y el servidor web. Recibirán datos desde las pantallas y se encargarán de enviarlos, a través de una petición HTTP, al *backend*. Una vez recibida una respuesta le devolverán la información a las pantallas para que estas la gestionen como sea necesario.

#### 5.3.1.2.4 Routing

Módulo encargado de la navegación en la aplicación web, efectuando distintas tareas como asociar URL a las diferentes vistas, controlar el modo de ruteo, o ejecutar ciertas acciones antes de acceder a una pantalla o justo antes de salir de ella.

#### 5.3.1.2.5 Store

Será una fuente de información única para toda la aplicación. Se construirá siguiendo el patrón Redux, más concretamente la implementación para Vue, Vuex [19]. El objetivo de este módulo es centralizar información que debe ser accedida desde distintos puntos del sistema, como pueden ser los datos de autenticación.

### 5.3.2 Descripción de los interfaces entre subsistemas

La comunicación entre módulos desde dentro de cada uno de los grandes sistemas (servidor web y aplicación web) es a través de llamadas a métodos o por configuración. Este último es el caso que asociará URL a vistas, por ejemplo. Si que merecen especial atención dos casuísticas, una en cada subsistema, que tienen algo más de complejidad, como son la comunicación con la base de datos en el *backend* y la gestión de la información centralizada en el store en el *frontend*:

- Para la comunicación con la base de datos Django ofrece una abstracción de esta en forma de API, que permite la creación, actualización, borrado y obtención de información almacenada. Esta información, de manera más detallada, se podrá consultar en [20].

- El uso de Vuex para la gestión del estado implica seguir unos patrones para la manipulación de datos que se almacenan en este. Se explica claramente en [19]. Existen tres componentes principales, que de forma resumida son:
  - **State:** Objeto que almacena los datos. Se puede obtener información de él, pero no debe ser modificado directamente, esto se hará a través de mutaciones.
  - **Mutaciones:** Son métodos encargados de modificar el State. Para llamar a uno de estos métodos se ejecuta un *commit* sobre el objeto que representa el State, pasándole como parámetro el nombre de la mutación.
  - **Acciones:** Básicamente funcionan de la misma forma que las mutaciones. No obstante, no modifican directamente el State, si no que llaman a mutaciones encargadas de realizar estos los cambios. Además, permiten operaciones asíncronas.

La comunicación entre los dos grandes subsistemas será realizada a través de un API Rest. El servidor recibirá peticiones a través de URLs que, como se ha explicado, estarán asociadas a vistas. Estas peticiones se efectuarán en la aplicación web desde el módulo de servicios, empleando el cliente HTTP Axios, específicamente preparado para consumir este tipo de APIs y que permite tanto enviar la petición como procesar la respuesta [21].

## 5.4 Diagrama de Clases Preliminar del Análisis

En este apartado se presentarán las clases identificadas durante el análisis que deberían estar presentes en el sistema. Se mostrará en primer lugar el diagrama de los dos subsistemas, para posteriormente, describir los elementos identificados y métodos propuestos.

Se debe de tener en cuenta que estos diagramas se han realizado en la fase de análisis, en la que pese a haber realizado cursos de introducción, aún no se han utilizado las tecnologías en un proyecto real, por lo que podrían aparecer cambios a lo largo del proyecto si se descubren nuevas formas de solucionar ciertos aspectos.

### 5.4.1 Diagrama de clases

Se ha realizado un diagrama para cada uno de los subsistemas por separado para permitir que el resultado sea más legible.

5.4.1.1 Diagrama de clases del servidor web

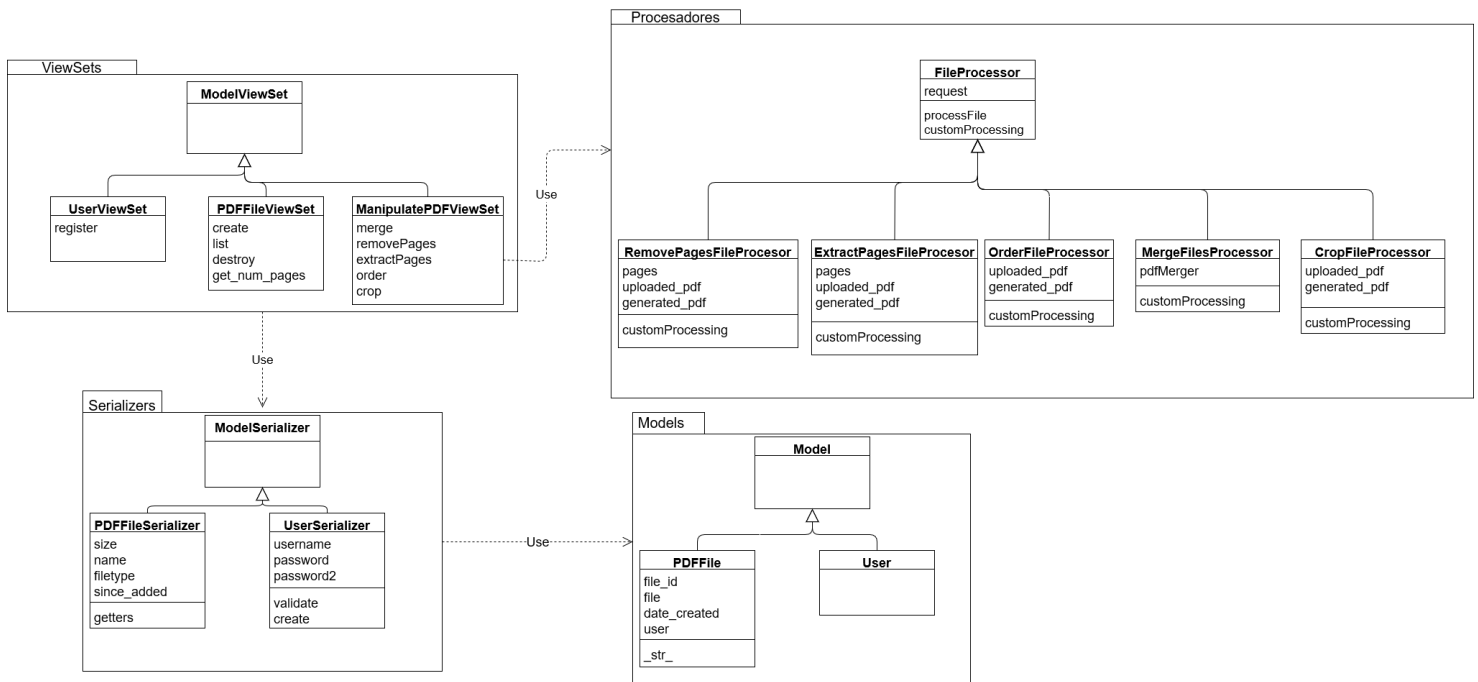


Figura 5.2 Diagrama de clases de análisis del subsistema “Servidor web”

5.4.1.2 Diagrama de clases de la aplicación web

En el caso del *frontend* no van a existir clases debido a que Vue.js no las emplea. No obstante, los ficheros Vue se componen básicamente de una plantilla HTML, propiedades y métodos, por lo que a la hora de realizar el diagrama se considerarán como “clases”. También se hará con los ficheros JavaScript existentes.

En el diagrama no se detallan los componentes que se van a utilizar por que se considera que se trata de un aspecto de diseño, por lo que únicamente se enlaza el paquete de vistas con el de componentes, para representar que harán uso de estos.

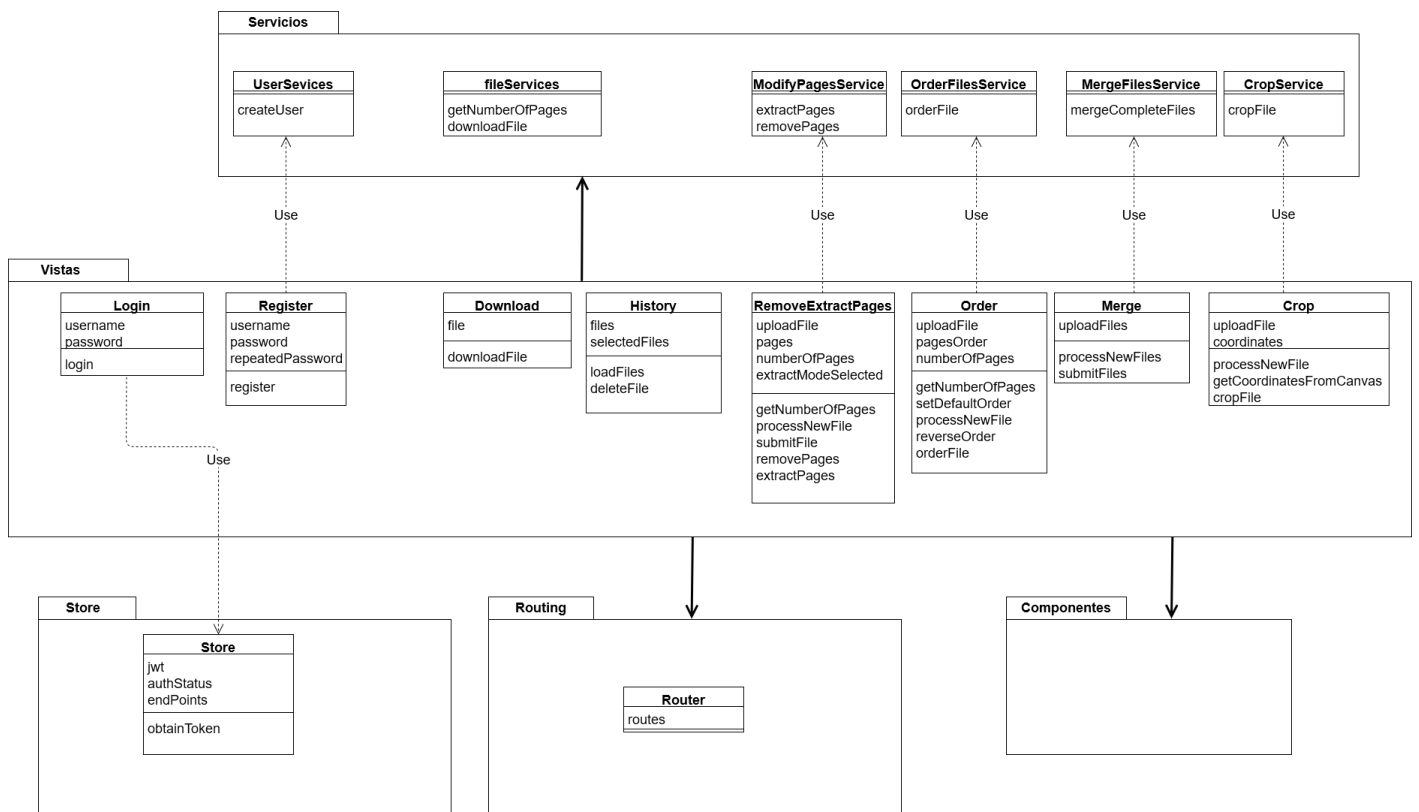


Figura 5.3 Diagrama de “clases” de análisis del subsistema aplicación web

## 5.4.2 Descripción de las clases

En este apartado se detallarán las responsabilidades de las clases representadas en los diagramas, tanto de la propia clase en sí como de los atributos y métodos propuestos para ella.

### 5.4.2.1 Servidor web

A continuación, se recogen las clases que conforman el servidor web, separadas por paquetes.

#### 5.4.2.1.1 Viewsets

Como se ha mencionado anteriormente, el concepto de *viewset* proviene de Django. Viene a ser equiparable a lo que es un controlador en otros *framework* que implementan el patrón MVC. Cada uno de ellos tendrá métodos que enlazarán una petición HTTP con una acción, como puede ser añadir un fichero, eliminarlo o realizar un listado de los documentos asociados a un usuario.

En el diagrama se añade una clase *ModelViewSet* que no está implementada en el sistema, sino que es parte del propio *framework*. De ella heredarán su comportamiento los distintos *viewset*.

Los atributos y métodos de los que se podrá hacer uso pero que pertenezcan a esta superclase no serán registrados en este apartado. Se pueden consultar en [22].

Este paquete contendrá las siguientes clases:

<b>Nombre de la clase</b>
UserViewSet
<b>Responsabilidades</b>
Es la clase encargada de enlazar las URL para la gestión de usuarios con las acciones que permiten realizar esta gestión, además de responder a las peticiones que llegan a través de dichas URL. En una primera versión únicamente se podrán crear usuarios por lo que esta será su responsabilidad.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>register:</b> Recogerá los datos procedentes de la petición, y hará uso del serializer asociado para que este valide, y en caso de que sea satisfactoria la validación, cree un usuario. Enviará la respuesta a la petición según lo que haya sucedido en los pasos anteriores.</li> </ul>

<b>Nombre de la clase</b>
PDFFileViewSet
<b>Responsabilidades</b>
Encargada de gestionar la asociación de peticiones con acciones que permiten modificar u obtener información de los ficheros PDF que se encuentran guardados en el sistema.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>create:</b> Permitirá añadir un nuevo documento al sistema recibido a través de una petición POST. Hará uso del serializador correspondiente para validar, guardar y enviará una respuesta a la petición dependiendo del resultado de la validación.</li> <li>• <b>list:</b> Obtendrá una lista de todos los documentos guardados para el usuario que está en sesión y responderá con esta información.</li> <li>• <b>destroy:</b> Eliminará el fichero o ficheros cuyo identificador venga en la petición DELETE asociada a este método.</li> <li>• <b>get_num_pages:</b> Obtendrá para el pdf que reciba en la petición el número de páginas que lo componen y responderá con estos datos.</li> </ul>

<b>Nombre de la clase</b>
---------------------------

ManipulatePDFViewSet

**Responsabilidades**

Encargada de establecer la relación entre peticiones que pretenden generar un nuevo PDF a partir de uno o más documentos de este tipo y una información adicional (distinta para cada caso), y las acciones encargadas de generar estos nuevos documentos.

**Atributos propuestos****Métodos propuestos**

- **merge:** Recibirá a través de una petición POST una serie de documentos pdf en un orden y llamará al objeto encargado de realizar el procesamiento necesario para generar uno nuevo uniendo estos documentos. Finalmente, hará uso del serializador correspondiente para validar y guardar este nuevo documento. Por último, envía una respuesta en función de si se ha realizado o no correctamente esta operación.
- **removePages:** Recibirá a través de una petición POST un fichero y un listado de páginas que se pretenden eliminar. Llamará al objeto encargado de crear un nuevo fichero que no incluya estas páginas y una vez obtenido lo validará y guardará si es posible, enviando una respuesta a la petición informando si se ha llevado a cabo o no correctamente la acción.
- **extractPages:** Recibirá la misma información que **removePages** pero en este caso el procesador asociado generará el pdf incluyendo únicamente el listado de páginas recibidas. Hará uso del serializador y enviará una respuesta exactamente igual que los métodos anteriores.
- **order:** Recibirá, en una petición POST, un pdf y un listado de páginas en la que cada índice contendrá un número, información que usará el procesador concreto para generar el documento con las páginas en este orden. El uso del serializador y respuesta será común al resto de métodos.
- **crop:** Recibirá una petición POST con un pdf y unas coordenadas y enviará esta información al procesador concreto. Con el fichero generado realizará una validación y guardado y responderá exactamente igual que el resto de los métodos.

## 5.4.2.1.2 Serializadores

De nuevo, las clases que componen este paquete o subsistema provienen de un concepto de Django. Permiten la transformación bidireccional entre los tipos de datos que usa Python y elementos del *framework*, como querysets o modelos, de forma transparente. Además, incluyen utilidades para la validación de estos datos.

Los serializadores empleados heredan de la clase ModelSerializer. Esta se representa en el diagrama para dar contexto, pero no ha sido implementada en el sistema por ser parte de Django. Se puede encontrar su documentación en [23].

Para las subclases solo se detallarán los métodos que deban ser redefinidos.

<b>Nombre de la clase</b>
UserSerializer
<b>Responsabilidades</b>
Clase encargada de la representación de los datos que se van a poder obtener y guardar para un usuario y la “traducción” entre los datos obtenidos de la petición y el modelo que se va a guardar, así como efectuar la validación de estos datos.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>username:</b> Nombre con el que el usuario se identifica en la aplicación, se validará su obligatoriedad y que sea único en el sistema (no habrá dos usuarios con mismo nombre).</li> <li>• <b>password:</b> Contraseña asociada al usuario. Será un campo requerido para la creación del usuario.</li> <li>• <b>password2:</b> Repetición de la contraseña asociada al usuario. Será un campo requerido para la creación del usuario.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>validate:</b> Además de las validaciones ya definidas en los atributos se validará que los atributos <b>password</b> y <b>password2</b> tengan el mismo valor, levantando una excepción si no lo tuviesen.</li> <li>• <b>create:</b> Creará un usuario a partir del nombre de usuario y le asociará la contraseña, para posteriormente guardarlo empleando el modelo correspondiente.</li> </ul>

<b>Nombre de la clase</b>
PDFFileSerializer
<b>Responsabilidades</b>
Clase encargada de la representación de los datos que se van a poder obtener y guardar de un fichero, además de mapear propiedades que se encuentran dentro del campo de tipo FileField (del modelo PDFFile) para poder representar los datos como el nombre o el tamaño del fichero que se encuentran encapsulados dentro de este campo.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>size:</b> Tamaño del fichero.</li> <li>• <b>name:</b> Nombre del fichero.</li> <li>• <b>filetype:</b> Tipo de fichero (extensión).</li> <li>• <b>since_added:</b> Fecha de creación.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>getters:</b> Cada uno de los atributos propuestos tendrá un <i>getter</i> asociado que obtendrá del campo correspondiente del modelo esta información.</li> </ul>

#### 5.4.2.1.3 Models

Los modelos son también un concepto propio de Django. Son la representación de los datos que se van a almacenar en la base de datos, y cada modelo se mapea a una tabla de esta. Esta y otra información se puede consultar en [17].

En el diagrama se han representado, para contextualizar, tanto la clase Model como la clase User, que proceden del *framework*. La información de esta última se puede obtener en [24]. La información sobre la clase Model está referenciada en el párrafo anterior.

<b>Nombre de la clase</b>
PDFFile
<b>Responsabilidades</b>
Representa la información que va a ser mapeada en la tabla de base de datos de los ficheros.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>file_id:</b> Identificador del elemento, atributo que se generará automáticamente.</li> <li>• <b>file:</b> Campo de tipo FileField, contendrá información del fichero.</li> <li>• <b>date_created:</b> Fecha en la que se creó el fichero, por defecto tomará la fecha actual.</li> <li>• <b>user:</b> Foreign Key que asociará el fichero a un usuario.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>_str_:</b> Representación en forma de string del fichero. En principio será su nombre.</li> </ul>

#### 5.4.2.1.4 Procesadores

En este caso no se trata de un elemento de Django. Es un módulo propio creado para la manipulación de la información de documentos PDF.

<b>Nombre de la clase</b>
FileProcessor
<b>Responsabilidades</b>
Clase base que definirá la secuencia de pasos genérica para realizar cualquier manipulación en un fichero y que será redefinida por cada uno de los procesadores concretos. Se encargará de la inicialización de datos, establecer la plantilla de pasos a seguir y realizar la implementación de los que son comunes a todos los procesadores o a la mayoría de ellos, como puede ser la creación de un fichero a partir de los datos o preparar los datos en la estructura que “comprende” el serializador.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>request:</b> Información de la petición procedente del viewset.</li> </ul>



**Métodos propuestos**

- **processFile:** Establece la plantilla de pasos a seguir para procesar un fichero, incluyendo un apartado para que cada procesador redefina con su propio comportamiento.
- **customProcessing:** Método abstracto que redefinirá cada uno de los procesadores.

**Nombre de la clase**

RemovePagesFileProcessor

**Responsabilidades**

Establecerá el algoritmo concreto empleado para el borrado de páginas de un documento.

**Atributos propuestos**

- **pages:** almacenará el listado de páginas a borrar.
- **uploaded\_pdf:** objeto de la librería pyPdf2 lector del fichero que llegará en la petición.
- **generated\_pdf:** objeto de la librería pyPdf2 que permite generar un nuevo PDF.

**Métodos propuestos**

- **customProcessing:** Recorrerá el listado de páginas proporcionadas en la petición, añadiendo tan solo al pdf generado aquellas que no se encuentran en el listado.

**Nombre de la clase**

ExtractPagesFileProcessor

**Responsabilidades**

Establecerá el algoritmo concreto empleado para la extracción de páginas de un documento.

**Atributos propuestos**

- **pages:** almacenará el listado de páginas a extraer.
- **uploaded\_pdf:** objeto de la librería pyPdf2 lector del fichero que llegará en la petición.
- **generated\_pdf:** objeto de la librería pyPdf2 que permite generar un nuevo PDF.

**Métodos propuestos**

- **customProcessing:** Recorrerá el listado de páginas proporcionadas en la petición, añadiendo tan solo al pdf generado aquellas que se encuentran en el listado.

<b>Nombre de la clase</b>
OrderFileProcessor
<b>Responsabilidades</b>
Establecerá el algoritmo concreto empleado para ordenar las páginas dentro de un documento.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploaded_pdf</b>: objeto de la librería pyPdf2 lector del fichero que llegará en la petición.</li> <li>• <b>generated_pdf</b>: objeto de la librería pyPdf2 que permite generar un nuevo PDF.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>customProcessing</b>: Obtendrá un listado con las páginas en el orden que se desean añadir al nuevo documento y recorrerá este listado, añadiendo una a una las páginas al objeto <b>generated_pdf</b>.</li> </ul>

<b>Nombre de la clase</b>
MergeFilesProcessor
<b>Responsabilidades</b>
Establecerá el algoritmo concreto empleado para unir dos o más documentos PDF en un orden concreto.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>pdfMerger</b>: objeto de la librería pyPdf2 que permite unir ficheros.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>customProcessing</b>: Obtendrá los ficheros en orden y los irá añadiendo al objeto pdfMerger.</li> </ul>

<b>Nombre de la clase</b>
CropFileProcessor
<b>Responsabilidades</b>
Establecerá el algoritmo concreto empleado para recortar el contenido visible de un PDF.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploaded_pdf</b>: objeto de la librería pyPdf2 lector del fichero que llegará en la petición.</li> </ul>

<ul style="list-style-type: none"> <li>• <b>generated_pdf</b>: objeto de la librería pyPdf2 que permite generar un nuevo PDF.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>customProcessing</b>: Obtendrá las coordenadas (punto superior izquierdo, anchura y altura) y modificará la “caja contenedora” del documento para adaptarlo al nuevo tamaño.</li> </ul>

#### 5.4.2.2 Aplicación web

En este apartado, se detallarán las “clases” que conforman la aplicación web. Se recuerda que no se trata de clases como tal, sino que son ficheros de Vue o JavaScript para los que se hace esta abstracción.

##### 5.4.2.2.1 Servicios

Dentro de la aplicación web los servicios serán los encargados de enlazar esta misma con el servidor web, enviando peticiones y proporcionando la información recibida a las vistas.

<b>Nombre de la clase</b>
UserServices
<b>Responsabilidades</b>
Contendrá los servicios que manipulen o consulten información sobre los usuarios del sistema.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>createUser</b>: Enviará una petición al servicio encargado de la creación de usuarios, pasando como parámetro un objeto que contiene el nombre de usuario y 2 veces la contraseña del usuario que se desea crear.</li> </ul>

<b>Nombre de la clase</b>
FileServices
<b>Responsabilidades</b>
Contendrá los servicios que realicen operaciones comunes sobre los ficheros o que consulten datos sobre estos.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>

- **getNumberOfPages:** Enviará una petición al servicio que devuelve el número de páginas de un fichero, mandándole este como parámetro y retornará dicho número.
- **downloadFile:** Obtendrá del sistema de ficheros la información del documento a descargar y a partir de la respuesta creará un enlace sobre el que se hará click programáticamente, descargando el archivo.

<b>Nombre de la clase</b>
ModifyPagesService
<b>Responsabilidades</b>
Contendrá los servicios que impliquen cambios en las páginas de un fichero, ya sea extracción o eliminación de estas.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>extractPages:</b> Llamará al servicio del servidor encargado de la extracción de páginas de un fichero, pasándole como información el propio fichero y el listado de páginas. Retornará la información que le devuelva el servidor.</li> <li>• <b>removePages:</b> Llamará al servicio del servidor encargado de la eliminación de páginas de un fichero, pasándole como información el propio fichero y el listado de páginas. Retornará la información que le devuelva el servidor.</li> </ul>

<b>Nombre de la clase</b>
OrderFilesService
<b>Responsabilidades</b>
Contendrá el servicio encargado de conectar con el servidor para la ordenación de las páginas de un documento PDF.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>orderFile:</b> Enviará una petición al servicio encargado de ordenar las páginas de un documento, pasando como información el propio PDF y las páginas en el orden que se desea que aparezcan. Retornará la información devuelta por el servidor.</li> </ul>

<b>Nombre de la clase</b>
MergeFilesService
<b>Responsabilidades</b>
Contendrá el servicio que enviará la petición al servidor para unir una serie de documentos pdf.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>mergeCompleteFiles:</b> Enviará una petición al servicio encargado de unir varios documentos en uno, incluyendo en la petición el listado de documentos, y retornará la información devuelta por el servidor.</li> </ul>

<b>Nombre de la clase</b>
CropService
<b>Responsabilidades</b>
Contendrá el servicio que enviará la petición al servidor para recortar el área visible de un documento.
<b>Atributos propuestos</b>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>cropFile:</b> Enviará una petición al servicio encargado de recortar un PDF, incluyendo en la petición el fichero y las coordenadas que se desea que tenga el área visible y retornará la información devuelta por el servidor.</li> </ul>

#### 5.4.2.2.2 Vistas

Las vistas serán ficheros de Vue. Básicamente, contendrán el código que interactúa con el DOM, muestra información procedente de otros paquetes y envía datos introducidos por el usuario a los servicios para su procesamiento.

<b>Nombre de la clase</b>
Login
<b>Responsabilidades</b>

Presentará al usuario un formulario para introducir sus credenciales de inicio de sesión, y gestionará esta información enviándola al store encargado de almacenar la sesión.

**Atributos propuestos**

- **username:** Identificador del usuario introducido en el formulario.
- **password:** Contraseña introducida en el formulario.

**Métodos propuestos**

- **login:** Introducirá en un objeto la información recogida del formulario y conectará con el store para gestionar el estado de la sesión. Si el inicio de sesión es correcto, redirigirá al usuario a la página de inicio. En otro caso, mostrará un mensaje de error.

**Nombre de la clase**

Register

**Responsabilidades**

Presentará al usuario un formulario para introducir sus datos de registro en el sistema. Gestionará esta información enviándola al servicio encargado de conectar con el servidor.

**Atributos propuestos**

- **username:** Identificador del usuario introducido en el formulario.
- **password:** Contraseña introducida en el formulario.
- **repeatedPassword:** Segunda contraseña introducida en el formulario.

**Métodos propuestos**

- **register:** Introducirá en un objeto la información recogida en el formulario y conectará con el servicio createUser, enviándole esta información. Si el registro es correcto redirigirá al usuario a la pantalla de Login. Si no, mostrará un mensaje de error.

**Nombre de la clase**

Download

**Responsabilidades**

Presentará al usuario un enlace para descargar un fichero y al calcar en el enlace descargará este fichero.

**Atributos propuestos**

- **file:** Objeto que contiene el fichero.

**Métodos propuestos**

- **downloadFile:** Llamará al servicio de descarga de ficheros.

<b>Nombre de la clase</b>
History
<b>Responsabilidades</b>
Mostrará al usuario el listado de ficheros que ha generado con el uso de la aplicación. Permitirá el borrado de archivos del sistema.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>files:</b> Objeto que contiene los ficheros que se mostrarán en el historial.</li> <li>• <b>selectedFiles:</b> Contendrá los PDF seleccionados del historial por el usuario.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>loadFiles:</b> Cargará los ficheros llamando al servicio que los obtiene para un usuario concreto.</li> <li>• <b>deleteFile:</b> Llamará al servicio encargado de borrar ficheros y se actualizará el listado en función de los que se eliminen.</li> </ul>

<b>Nombre de la clase</b>
RemoveExtractPages
<b>Responsabilidades</b>
Permitirá al usuario seleccionar un fichero y un listado de páginas que quiere extraer o eliminar del fichero.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploadFile:</b> Contendrá el fichero seleccionado por el usuario.</li> <li>• <b>pages:</b> Contendrá el listado de páginas seleccionadas por el usuario.</li> <li>• <b>numberOfPages:</b> Almacenará el número de páginas que tiene el fichero actual.</li> <li>• <b>extractModeSelected:</b> Recoge el tipo de procesamiento seleccionado: extracción o borrado.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>getNumberOfPages:</b> Obtendrá el número de páginas del PDF seleccionado, con una llamada al servicio correspondiente.</li> <li>• <b>processNewFile:</b> Se ejecutará al subir un nuevo archivo, validará y obtendrá la información necesaria.</li> <li>• <b>submitFile:</b> Recogerá la información introducida por el usuario y llamará al método encargado de realizar el procesamiento en función del modo seleccionado.</li> </ul>

- **extractPages:** Llamará al servicio encargado de la extracción de páginas y procesará la respuesta.
- **removePages:** Llamará al servicio encargado del borrado de páginas y procesará la respuesta.

<b>Nombre de la clase</b>
Order
<b>Responsabilidades</b>
Permitirá al usuario seleccionar un fichero y modificar el orden de páginas de este.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploadFile:</b> Contendrá el fichero seleccionado por el usuario.</li> <li>• <b>numberOfPages:</b> Almacenará el número de páginas que tiene el fichero actual.</li> <li>• <b>pagesOrder:</b> Recoge el listado de las páginas en el orden que el usuario desea que estén.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>getNumberOfPages:</b> Obtendrá el número de páginas del archivo seleccionado con una llamada al servicio correspondiente.</li> <li>• <b>processNewFile:</b> Se ejecutará al subir un nuevo fichero, validará y obtendrá la información necesaria.</li> <li>• <b>setDefaultOrder:</b> Inicializará la propiedad <b>pagesOrder</b>.</li> <li>• <b>reverseOrder:</b> Invertirá el orden del array <b>pagesOrder</b>.</li> <li>• <b>orderFile:</b> Recogerá los datos introducidos por el usuario y con ellos llamará al servicio encargado de ordenar las páginas. Finalmente, procesará la respuesta de este servicio.</li> </ul>

<b>Nombre de la clase</b>
Merge
<b>Responsabilidades</b>
Permitirá al usuario seleccionar una serie de ficheros y unirlos en un único fichero en el orden deseado.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploadFiles:</b> Almacenará los ficheros que ha introducido el usuario en el orden especificado.</li> </ul>
<b>Métodos propuestos</b>



- **processNewFiles:** Se ejecutará al subir los ficheros. Validará que son PDF y obtendrá la información correspondiente.
- **submitFiles:** Obtendrá el listado de ficheros y con él hará una llamada al servicio correspondiente, procesando posteriormente la respuesta.

<b>Nombre de la clase</b>
Crop
<b>Responsabilidades</b>
Permitirá al usuario seleccionar un fichero, establecer un nuevo recuadro para recortar sobre una previsualización de este y hacer únicamente visible lo que se encuentra en el recuadro.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>uploadFile:</b> Contendrá el fichero subido por el usuario.</li> <li>• <b>coordinates:</b> Almacenará las coordenadas del recuadro seleccionado por el usuario.</li> </ul>
<b>Métodos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>processNewFile:</b> Se ejecutará al subir el fichero, validando que se trata de un fichero pdf</li> <li>• <b>getCoordinatesFromCanvas:</b> Obtendrá los puntos seleccionados por el usuario en el Canvas.</li> <li>• <b>cropFile:</b> Obtendrá la información introducida por el usuario, llamará al servicio correspondiente y procesará el resultado.</li> </ul>

#### 5.4.2.2.3 Store

El store será un punto de información único basado en el concepto de Vuex. Podrá ser accedido desde cualquier punto de la aplicación web. Contendrá la información del usuario en sesión.

<b>Nombre de la clase</b>
Store
<b>Responsabilidades</b>
Será el punto en el que se almacene la información de la sesión del usuario que tiene sesión iniciada en la aplicación.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>jwt:</b> Almacenará el token de sesión del usuario.</li> <li>• <b>authStatus:</b> Almacenará el estado en que se encuentra la sesión.</li> <li>• <b>endpoints:</b> Contendrá las url de los servicios necesarios para la gestión de la sesión.</li> </ul>
<b>Métodos propuestos</b>

- **obtainToken:** Llamará al servicio encargado de proporcionar un token al usuario y almacenará la información de la sesión en base a la respuesta.

#### 5.4.2.2.4 Router

Empleando la librería oficial de enrutamiento para Vue, en este módulo se asociará a cada una de las vistas un *path* y un nombre. A través de cualquiera de estos se podrá navegar por la aplicación.

<b>Nombre de la clase</b>
Router
<b>Responsabilidades</b>
Almacenará la información de enrutamiento del sistema.
<b>Atributos propuestos</b>
<ul style="list-style-type: none"> <li>• <b>routes:</b> Array de objetos en el que cada objeto estará conformado por tres propiedades (<i>path</i>, <i>name</i> y <i>component</i>) que vincularán el componente (vista en el caso de esta aplicación) con el <i>path</i> y <i>name</i> especificado.</li> </ul>
<b>Métodos propuestos</b>

#### 5.4.2.2.5 Componentes

Los componentes serán utilizados en las vistas y permitirán reutilizar fragmentos de estas. No se detalla ninguno en el diagrama de análisis ya que se decidirá qué componentes se pueden usar a la hora de realizar el diseño. Además, la mayoría no serán implementados desde cero en el sistema, sino que se aprovecharán los de BootstrapVue.

## 5.5 Análisis de Interfaces de usuario

Las interfaces de usuario juegan un papel muy importante en el sistema, ya que trata de facilitar la manipulación de ficheros PDF a usuarios sin conocimientos avanzados de informática. Por tanto, se tratará de ofrecer una simbología y dinámicas de interacción intuitivas.

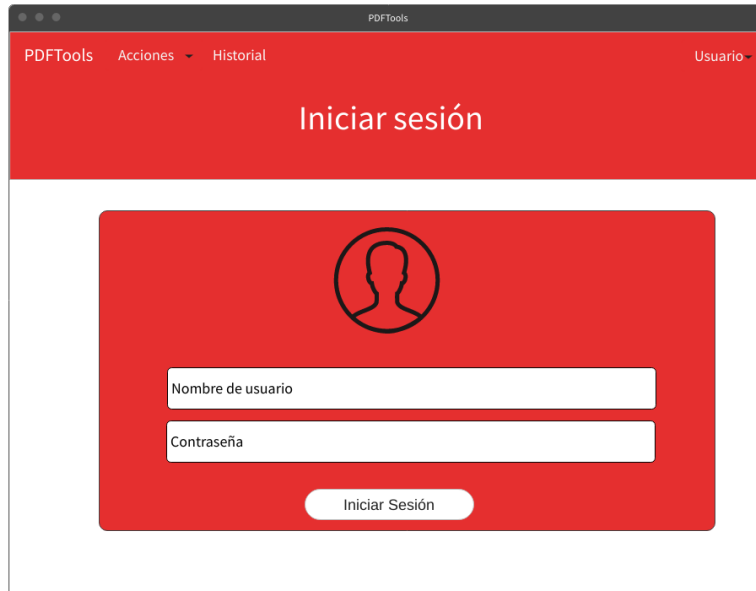
### 5.5.1 Descripción de la interfaz

Se han realizado una serie de *mockups* de todas las pantallas de la aplicación, pretendiendo que sirvan de guía a la hora de realizar las distintas interfaces. Estos se han realizado, tras una primera versión a lápiz y papel, con la herramienta MockFlow [25].

Existen una serie de elementos comunes a todas las pantallas. Estos son la barra de navegación y el indicador de la pantalla actual.

A continuación, se presentan estos diseños iniciales.

#### 5.5.1.1 Pantalla de inicio de sesión



**Figura 5.4 Mockup de pantalla de inicio de sesión**

### 5.5.1.2 Pantalla de registro

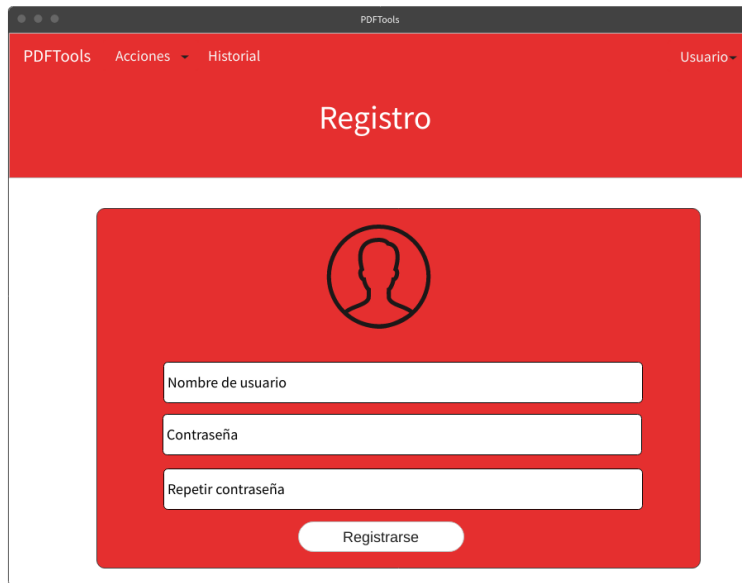


Figura 5.5 Mockup de pantalla de registro

### 5.5.1.3 Pantalla de inicio

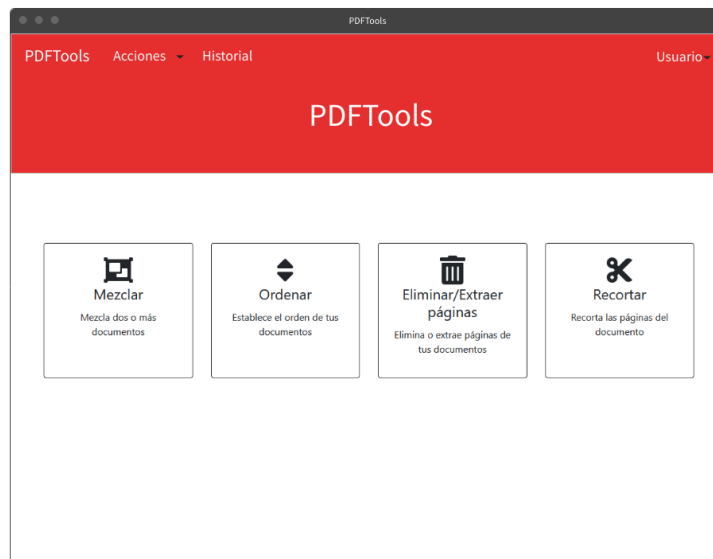


Figura 5.6 Mockup de pantalla principal

5.5.1.4 Historial de documentos

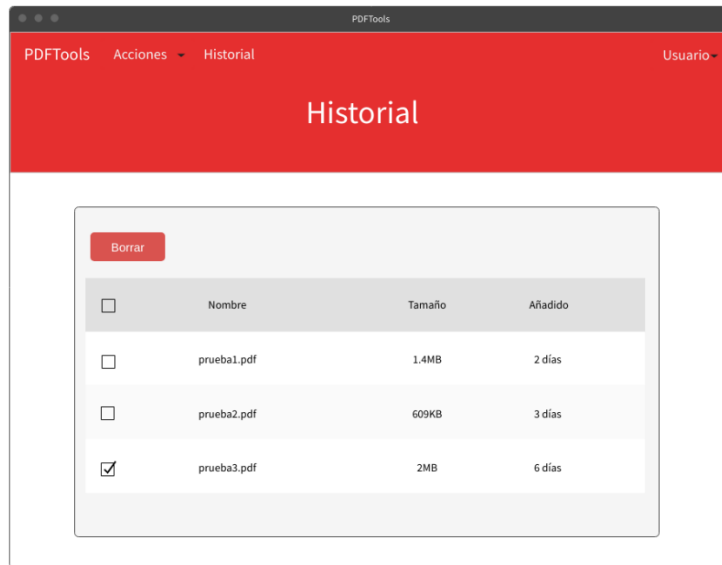


Figura 5.7 Mockup del historial de documentos

5.5.1.5 Eliminar/extraer páginas

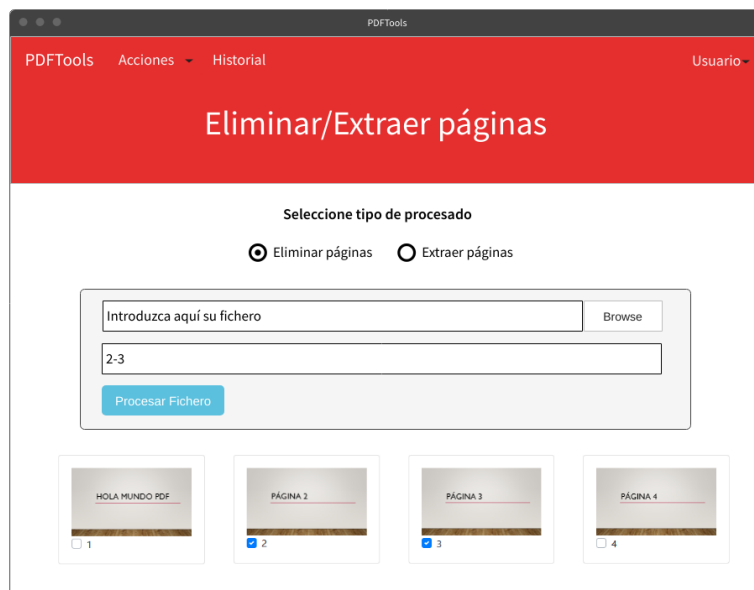


Figura 5.8 Mockup de la página de eliminación/extracción de páginas

### 5.5.1.6 Mezclar ficheros



Figura 5.9 Mockup de la página de unión de documentos

### 5.5.1.7 Ordenar ficheros

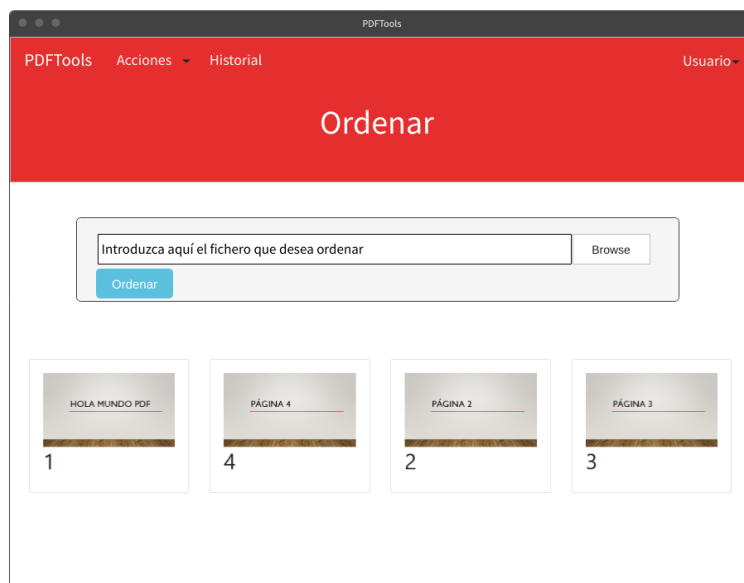


Figura 5.10 Mockup de la página de ordenación de ficheros

### 5.5.1.8 Recortar fichero

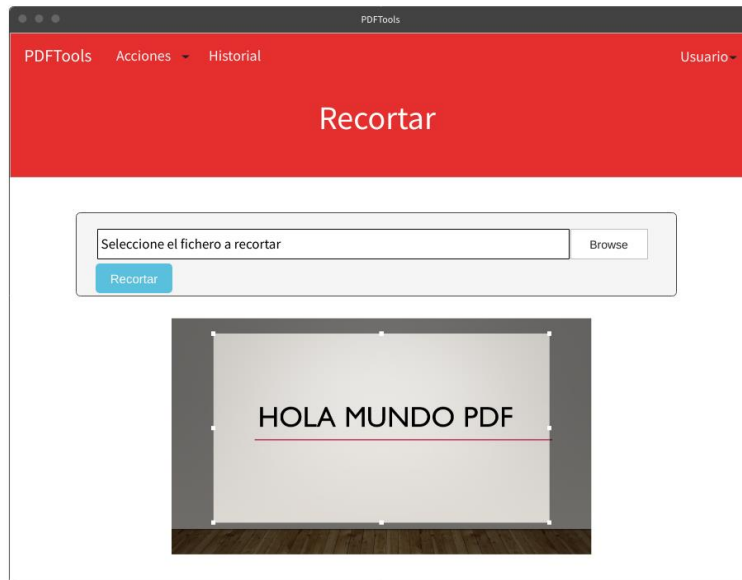


Figura 5.11 Mockup de la página de recorte de ficheros

### 5.5.1.9 Descarga de fichero

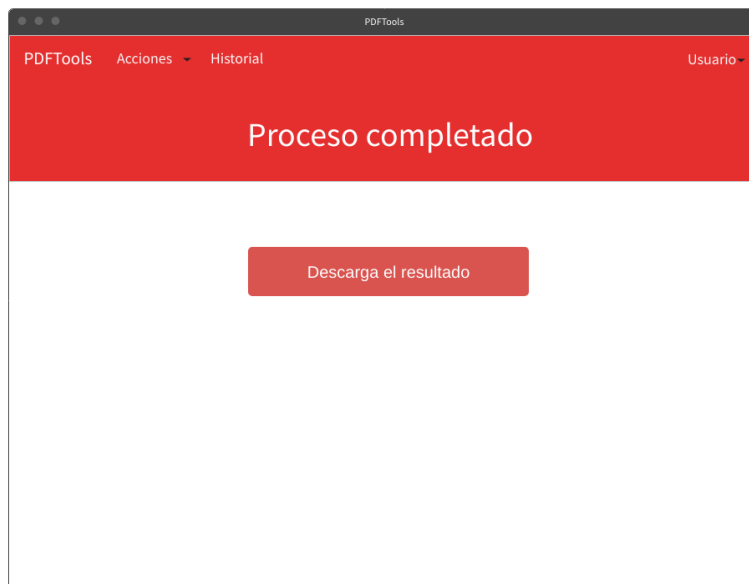


Figura 5.12 Mockup de la página de descarga de ficheros

## 5.5.2 Descripción del comportamiento de la interfaz

En todas las pantallas de la interfaz puede verse que existe una división en dos zonas diferenciada. La primera zona incluye la barra de navegación y la información de la página en la que se nos encontramos. En la segunda zona aparecen los componentes con los que el usuario va a interactuar para realizar las distintas operaciones o consultar información.

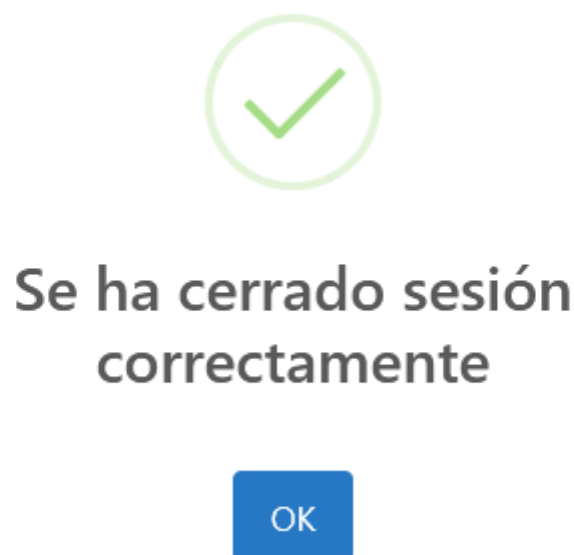
Los bocetos presentados en el apartado anterior ofrecen una visión inicial estática de las diferentes pantallas de la aplicación. No obstante, existen una serie de comportamientos dinámicos que no son apreciables en los mockups.

Las páginas de extracción/eliminación de páginas, unión de ficheros, ordenación y recorte mostrarán inicialmente tan solo la primera zona y de la segunda zona el componente que permite seleccionar los ficheros. Los elementos con los que puede interactuar aparecerán una vez que se cargue la información de los archivos, ya sea el número de páginas, imágenes, nombres...

Además, desde estas páginas se navegará, de forma transparente para el usuario, a la página de descarga del fichero una vez que se haya completado cualquiera de las operaciones.

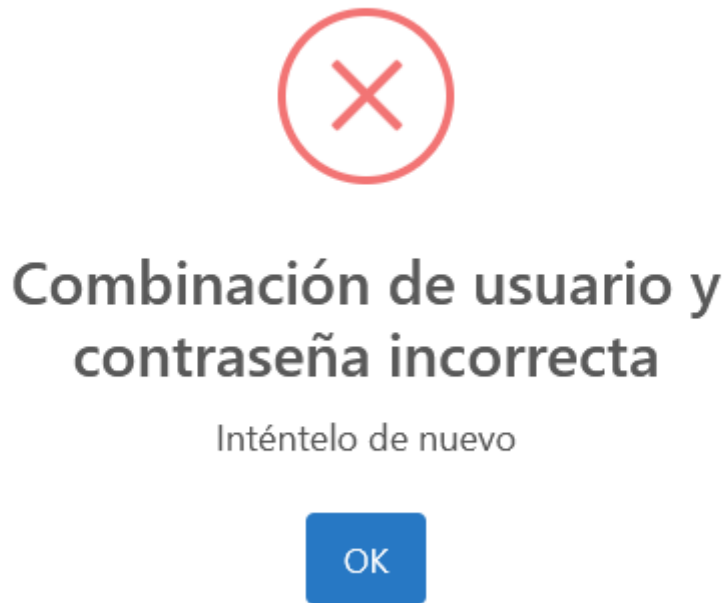
Al realizar cualquier operación se mantendrá al usuario informado de lo que está ocurriendo, intentando que esta información sea fácilmente comprensible y rápida de asimilar. En estos términos, cuando se esté realizando cualquier tipo de procesamiento en la aplicación que pueda demorarse un tiempo, se mostrará una capa de espera que hará ver al usuario que su petición está siendo procesada. Además, se informará al usuario de aquellas acciones que tengan importancia o errores que se produzcan durante la ejecución de estas.

Ejemplos:



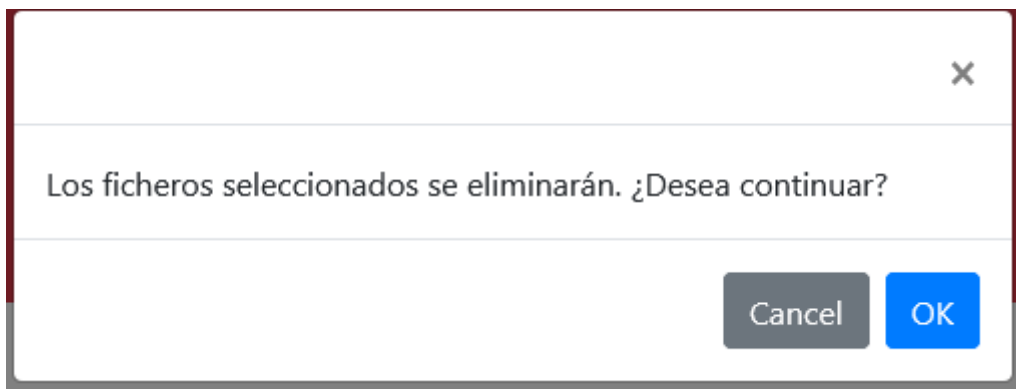
**Figura 5.13** Imagen representativa de una acción llevada a cabo con éxito





**Figura 5.14** Imagen representativa de acción no llevada a cabo con éxito

Por otra parte, se pedirá confirmación al usuario para aquellas acciones críticas que puedan implicar pérdida de datos, como puede ser el borrado de uno o más documentos.



**Figura 5.15** Diálogo de confirmación de acción

### 5.5.3 Diagrama de navegabilidad

Se presentará a continuación el diagrama de navegabilidad de la aplicación. Los nodos estarán constituidos por cada una de las pantallas identificadas. Además, existirá uno especial que representará la barra de navegación, ya que al ser un componente presente en todas las páginas y que articula todo el sistema de navegación se considera que debe ser representado de forma independiente, evitando así relaciones bidireccionales entre prácticamente todos los nodos.

Los nodos están unidos por flechas que representan el sentido de la navegación y la acción que genera este cambio de pantalla, descrita en pseudocódigo. Esta acción puede ser

desencadenada directamente por el usuario, por ejemplo, hacer click en un botón, o una acción programática, como desplazarse a la pantalla de descarga de fichero una vez que termina el procesamiento de este.

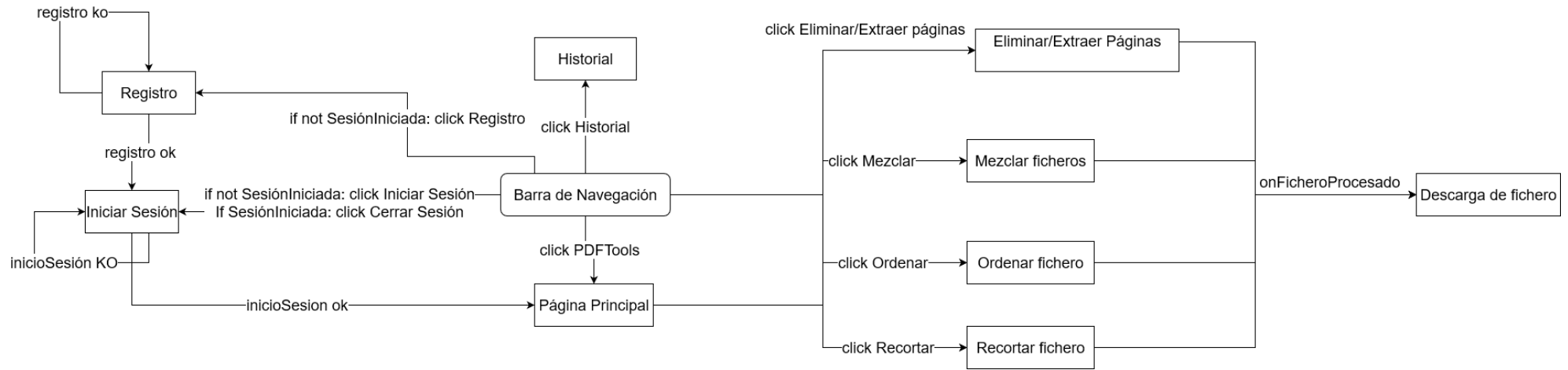


Figura 5.16 Diagrama de navegabilidad de la aplicación

## 5.6 Especificación del Plan de Pruebas

Se harán distintos tipos de pruebas que se recogen a continuación:

- **Pruebas funcionales:** Se realizarán pruebas del funcionamiento de la aplicación, tanto **automatizadas** como **no automatizadas**. Las pruebas no automatizadas tienen gran importancia y servirán para probar el sistema de forma más precisa, ya que, por las características de la aplicación existen algunas casuísticas difíciles de probar de forma automatizada. Por ejemplo, si se extrajesen tres páginas de un fichero las pruebas automatizadas podrían detectar fácilmente que el nuevo documento tiene efectivamente tres páginas, pero la verificación de que las tres páginas son las que deben ser, se deberá realizar de forma manual. De forma automática se probarán, por tanto, los métodos encargados de la manipulación de ficheros en el *backend*. El resto de las funcionalidades se probarán de forma no automática, empleando dos vías: realización de peticiones HTTP empleando el cliente Postman y a través de la propia aplicación web.
- **Pruebas de usabilidad:** Para realizar este tipo de pruebas se seleccionarán una serie de personas, tres en principio, con distinto nivel de conocimiento informático y de diversas edades. Se solicitará a estas personas que realicen una serie de tareas en el sistema y que rellenen un cuestionario, en el que se les pedirá opinión tanto de la dificultad para realizar las tareas solicitadas como preguntas generales sobre la experiencia de uso. Este cuestionario se detallará más adelante. Serán realizadas en la fase final del proyecto, una vez que haya desplegada una versión del sistema.

## Capítulo 6. Diseño del sistema

En este capítulo se realizará el diseño del sistema, partiendo del análisis anterior. Se incluye una nueva funcionalidad que no había sido analizada, la división de un PDF en páginas individuales. Como tiene un comportamiento y una estructura similar al resto de funcionalidades de la aplicación, no se actualizará el análisis ni la planificación inicial y se comenzarán a recoger en esta etapa de diseño.

### 6.1 Arquitectura del sistema

#### 6.1.1 Diagrama de paquetes

A continuación, se presenta el diagrama de paquetes de la aplicación, indicando en él ambos subsistemas y las dependencias con las librerías más importantes.

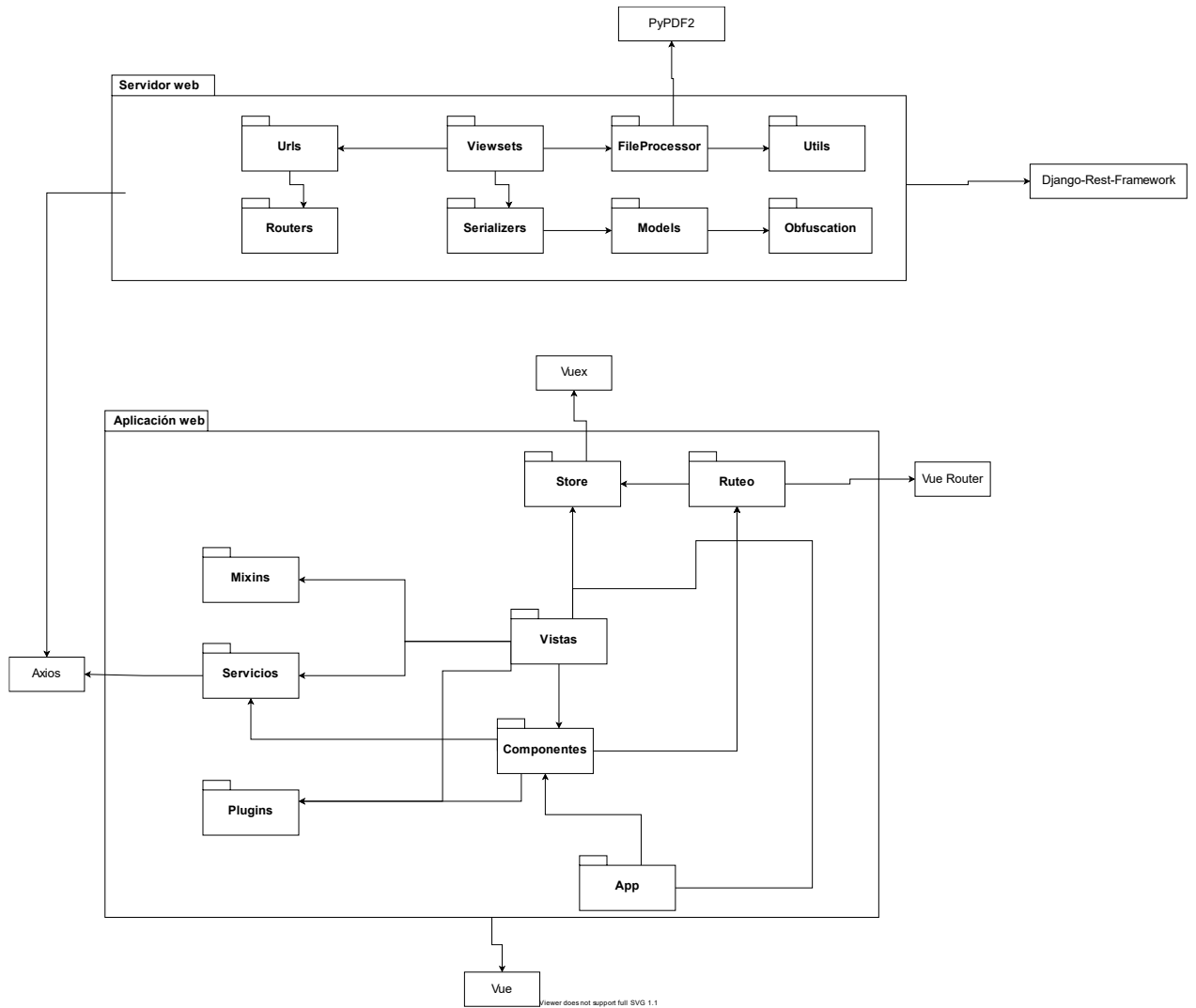


Figura 6.1 Diagrama de paquetes de ambos subsistemas

Las librerías que se detallan, en la parte externa de cada uno de los subsistemas, son las siguientes:

- **Django Rest Framework:** Se trata de un popular *framework* de Python basado en Django para realizar APIs REST [26].
- **PyPDF2:** Librería del lenguaje Python que permite realizar gran cantidad de operaciones con documentos PDF, tolerando trabajar con manipulación de ficheros en memoria [11].
- **Vue:** *Framework* de JavaScript orientado a la realización de aplicaciones SPA [9].
- **Vue Router:** Librería de enrutamiento oficial de Vue. Permite la navegación entre pantallas de la aplicación y efectuar distintos controles al navegar [27].
- **Vuex:** Librería oficial de Vue para el manejo del estado. Permite tener un lugar central en el que se almacenará información que podrá ser accedida desde cualquier parte de la aplicación [19].
- **Axios:** Cliente HTTP empleado para la comunicación entre los dos subsistemas de la aplicación [21].

Los paquetes son básicamente los mismos que se habían detallado en la descripción de los subsistemas, añadiendo algunos otros por necesidades que han sido detectadas. A continuación, se describen brevemente sus responsabilidades:

- **Servidor web:**
  - **Urls:** Efectuará la asociación entre una URL base y el conjunto de **routers** que la emplean como referencia.
  - **Routers:** Asocian una URL, partiendo de la base ya definida en el paquete Urls, a un **ViewSet**.
  - **Viewsets:** Recibirán los datos de las **peticiones** y se los enviarán al **procesador** correspondiente. Una vez realizado el procesamiento, emplearán los **serializers** para guardar información en la base de datos. Finalmente, retornarán una respuesta a la petición recibida.
  - **Serializers:** Transformarán los datos recibidos desde el **viewset** en datos que pueden asociarse a un **modelo**.
  - **FileProcessor:** Contendrá las clases encargadas de realizar el procesamiento específico sobre un fichero: eliminar páginas, ordenarlas, recortar el documento...
  - **Models:** Servirán como modelo de la base de datos, representando los datos que se guardan para cada *entidad* y los *triggers* que se pueden ejecutar al efectuar cambios en estas entidades.
  - **Utils:** Incorporará funciones de utilidad para el procesamiento de ficheros, principalmente asociadas a la transformación de datos.
  - **Obfuscation:** Proporcionará una forma de ofuscar ciertos contenidos para dificultar el acceso a estos, como pueden ser las carpetas del sistema de ficheros en las que se guardarán los ficheros generados por cada usuario.
- **Aplicación web:**
  - **App:** Más que un paquete en sí mismo, se trata de los ficheros de Vue y JavaScript que inicializan la aplicación, importan y registran las librerías que se usarán de forma general. Además, establecerá una plantilla en la que se incrustarán las vistas, gestionada por el sistema de ruteo.
  - **Vistas:** Realizarán la definición de las diversas pantallas de la aplicación, tanto plantilla HTML como comportamiento y estilos. Serán el punto de entrada de las interacciones del usuario con la aplicación, procesarán los datos introducidos por estos y se los comunicarán al resto de paquetes.
  - **Componentes:** Serán similares a las vistas, pero no serán pantallas completas, sino fragmentos de estas con la intención de que sean reutilizables o aplicarles estilos específicos.
  - **Plugins:** Contendrá funcionalidades globales a toda la aplicación. En principio la declaración de mensajes que se usarán en las distintas pantallas.
  - **Store:** Servirá como punto único de consulta para datos que deben ser accedidos desde múltiples puntos de la aplicación. La idea es que almacene los datos de la sesión del usuario.
  - **Mixins:** Concepto de Vue. Contendrá funcionalidades que pueden ser empleadas en distintas vistas o componentes con el objetivo de evitar la

duplicación de código. Por ejemplo, se empleará para la gestión de capas de espera.

- **Servicios:** Serán el punto de conexión de la aplicación web con el servidor. Recibirán datos de las vistas y los enviarán en peticiones HTTP empleando la librería Axios.
- **Ruteo:** Gestionará el sistema de navegación de la aplicación web, permitiendo los cambios entre vistas o realizar acciones al “salir” o “entrar” de alguna pantalla.

### 6.1.2 Diagramas de despliegue

Existen tren “servidores” que se comunican entre sí para que el conjunto del sistema funcione correctamente:

- **Azure Blob Storage:** Se trata de un servicio de almacenamiento en la nube de Azure [28] que almacenará los ficheros que se generan con el uso de la aplicación.
- **Servidor Heroku Backend:** *Dyno* [29] de Heroku que contendrá el *backend* de la aplicación. Además de la API Rest de Django, se encontrará en él la base de datos del sistema. Recibirá datos del servidor del *frontend* y enviará consultas al servicio de almacenamiento.
- **Servidor Heroku Frontend:** Contendrá la aplicación *frontend* desplegada, siendo el punto de entrada del usuario en el sistema. Enviará y recibirá datos del servidor del *backend*.

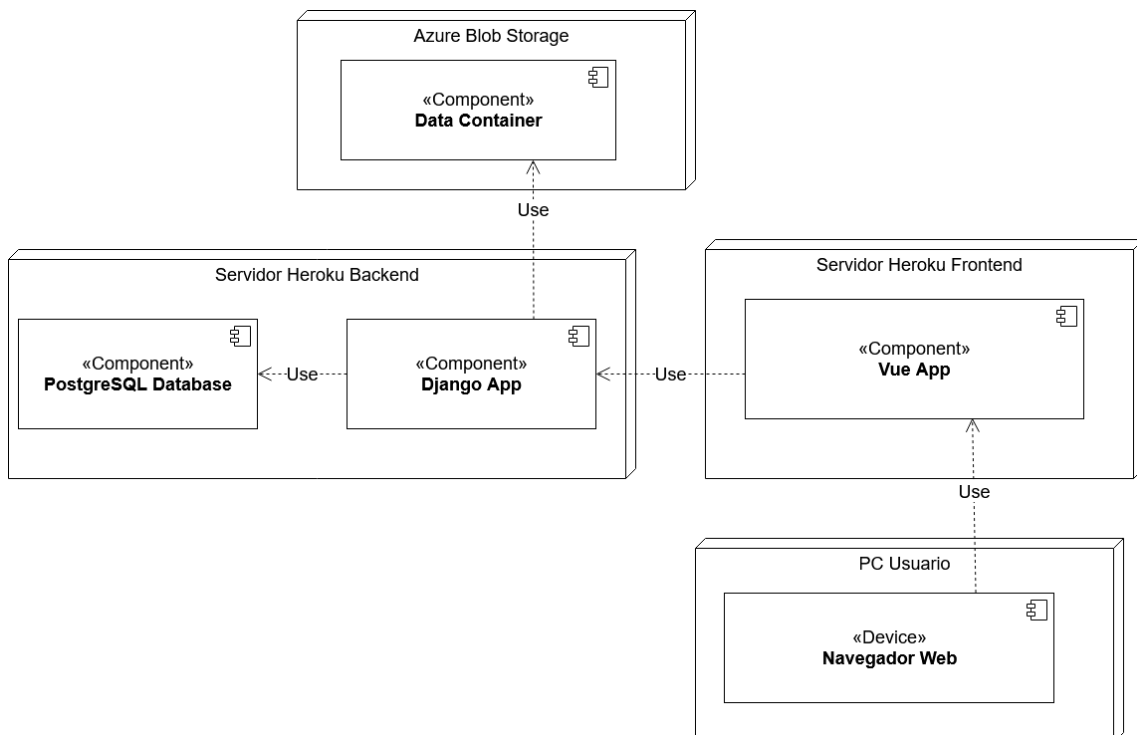


Figura 6.2 Diagrama de despliegue



## 6.2 Diseño de clases

### 6.2.1 Diagrama de clases

A continuación, se presentará los diagramas de clases de la aplicación, uno para cada uno de los dos subsistemas.

En términos generales, ambos estarán basados en los diagramas creados durante la fase de análisis, ya que el uso de *frameworks*, sobre todo en el caso de Django, “obliga” a seguir una determinada estructura de codificación según el patrón definido. No obstante, tendrá mayor nivel de detalle y se añadirán responsabilidades identificadas durante esta fase.

Se recuerda que no se trata de un diagrama de clases puro, ya que en el *frontend* se emplean componentes y ficheros *.js*, que realmente no son clases. Además, en el *backend* existirán funciones de utilidad que no estarán asociadas a ninguna clase concreta. En este último caso, se indicará en el diagrama que se trata de un fichero con funciones y no de una clase.

En las clases se omiten los atributos o métodos propios del *framework*. Además, algunos paquetes que definen únicamente configuraciones tampoco se incluyen.

Ambos diagramas, además de en las siguientes páginas, se entregan adjuntos en la carpeta **Diagramas**. Los archivos son **DiagramaClasesBack.svg** y **DiagramaClasesFront.svg**.

Por estar en este formato puede hacerse zoom sin perder resolución, tanto en los ficheros como en este documento.

6.2.1.1 Servidor web

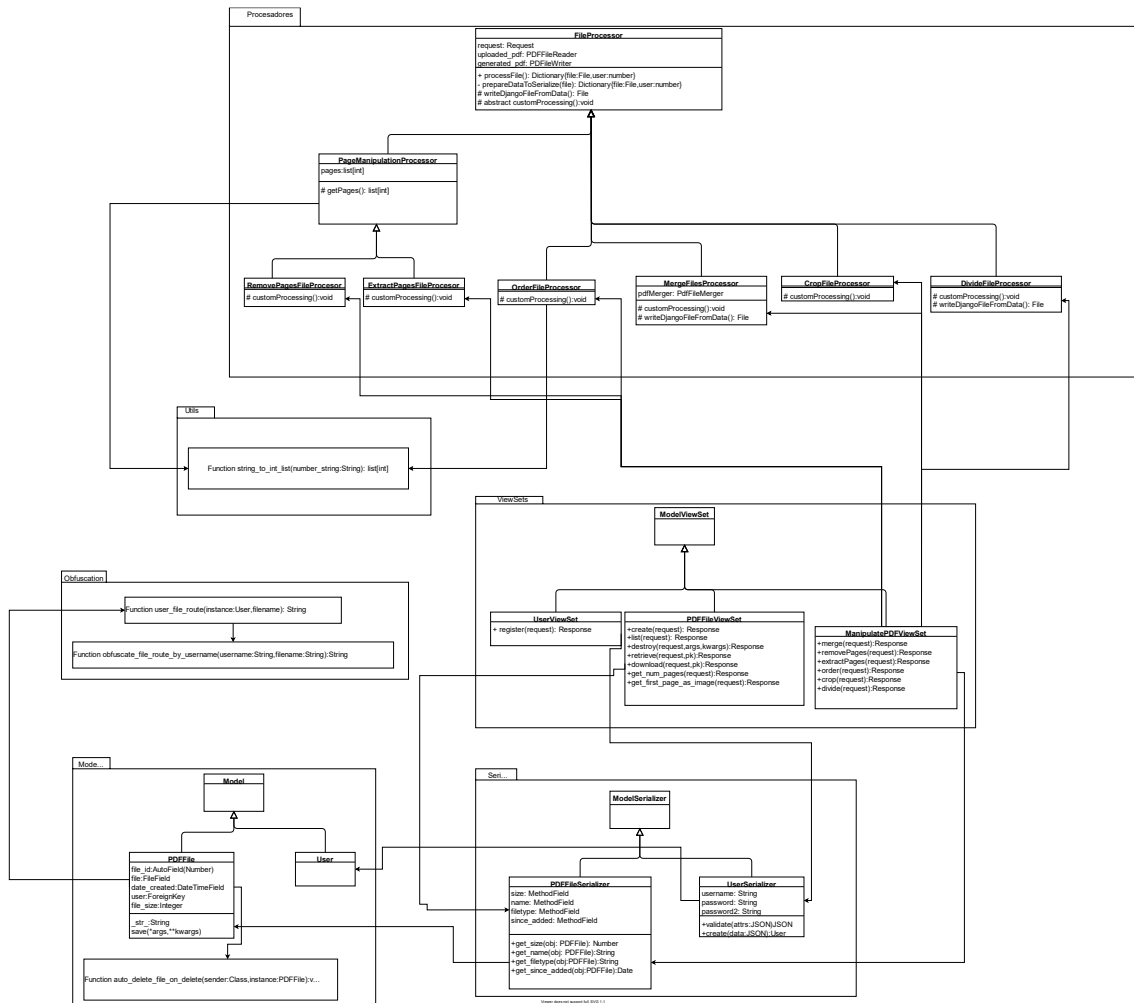


Figura 6.3 Diagrama de clases de diseño del servidor web

6.2.1.2 Aplicación web

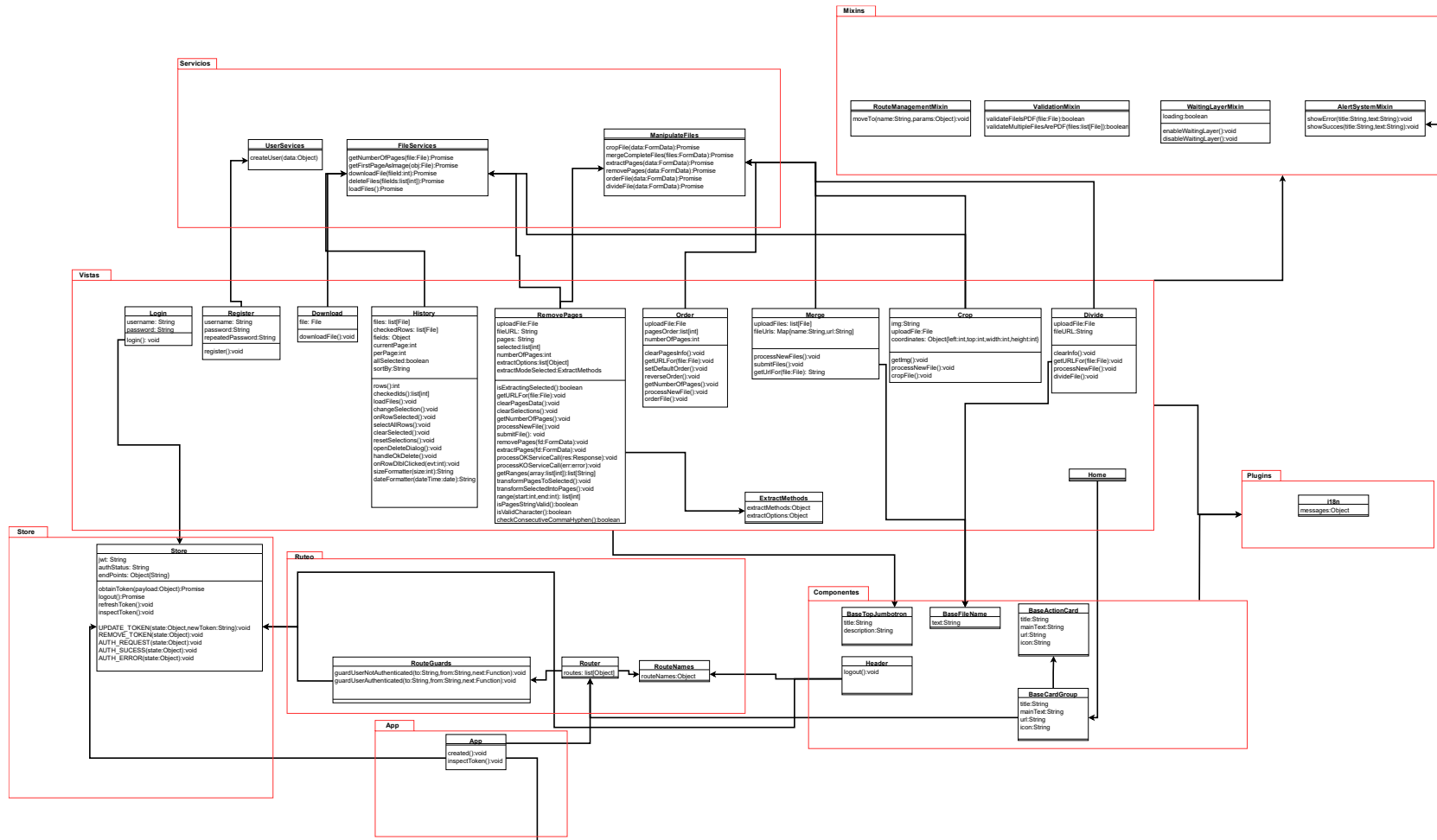


Figura 6.4 Diagrama de “clases” de diseño de la aplicación web

## 6.3 Diagramas de Interacción

En este apartado se presentarán los diagramas de interacción de dos tareas representativas del funcionamiento de la aplicación.

Se detallarán los siguientes casos de uso:

- Ordenar documento.
- Consultar historial de documentos generados.

Partiremos de la base de que el comportamiento es correcto y se ejecutan sin fallar ninguna validación.

### 6.3.1 Ordenar documento

La representación del diagrama se inicia en el momento en que el usuario ha establecido ya un orden en su interacción con la pantalla y calca el botón de “ordenar”.

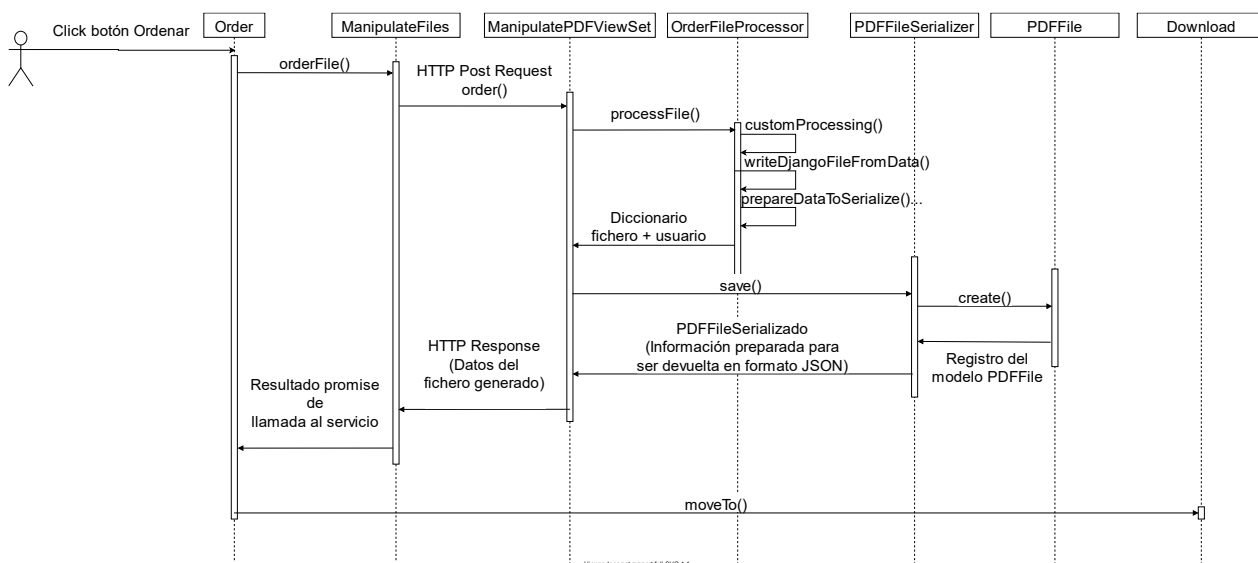


Figura 6.5 Diagrama de interacción para la ordenación de un documento

### 6.3.2 Consultar historial de documentos generados

Aquí se detallará un escenario más simple como es la consulta del historial de ficheros generados.

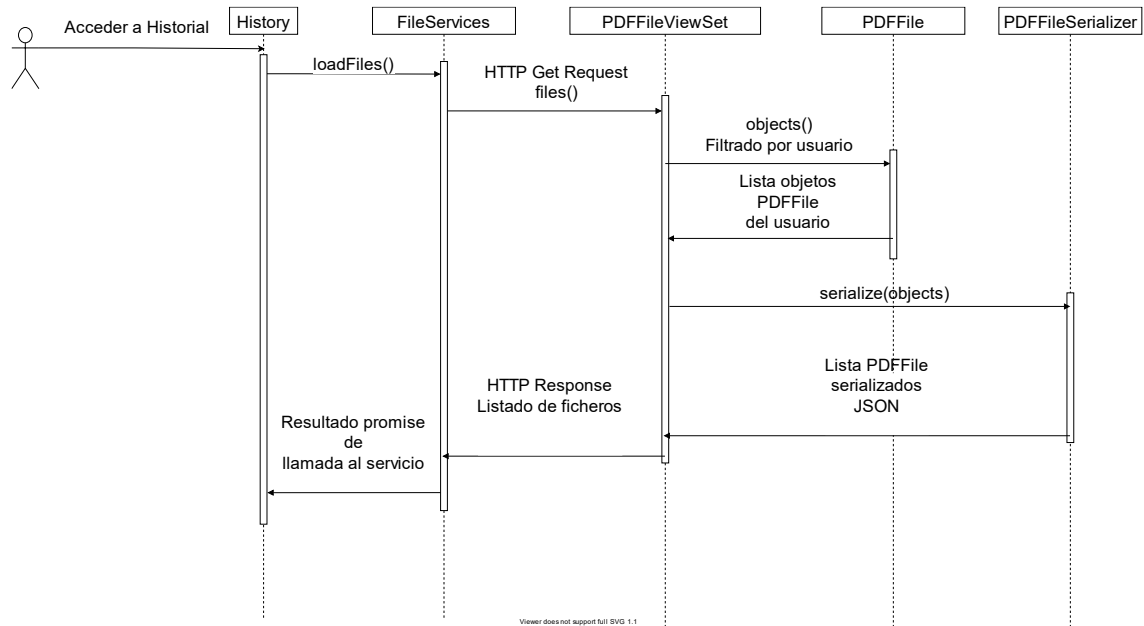


Figura 6.6 Diagrama de interacción para la consulta del historial de documentos

## 6.4 Diseño de la base de datos

En este apartado se reflejará muy brevemente la información relativa a la base de datos. No se considera necesario explicar este tema con gran nivel de detalle, puesto que no se trata de una aplicación que haga uso intensivo de esta, al ser empleada únicamente para almacenar la información de los ficheros generados y los usuarios del sistema.

### 6.4.1 Descripción del SGBD usado

Se han utilizado dos SGBD distintos a lo largo del ciclo de vida del proyecto.

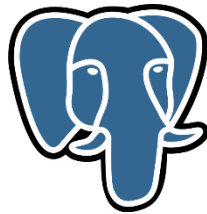
Durante la mayor parte del desarrollo se ha utilizado **SQLite**. Se trata de un SGBD relacional muy simple que no necesita de un servidor y se integra en el programa que lo utiliza. Toda la información que se encuentra en la base de datos se almacena en un archivo del sistema de ficheros de la máquina que aloja el programa [30].



Figura 6.7 Logo de SQLite

Tiene algunas desventajas como las limitaciones de tamaño y concurrencia. Por estas características podría provocar cuellos de botella en aplicaciones con mucho tráfico de peticiones HTTP. Por tanto, no es aconsejable su uso en producción en sistemas distribuidos.

En una fase final del desarrollo del proyecto previa a la puesta en producción, se pasó a emplear **PostgreSQL**, otro sistema gestor de bases de datos relacionales de código abierto [31]. En este caso los datos no se almacenan en un fichero del sistema y se necesita tener un servidor de bases de datos funcionando. Su uso es más recomendable en el caso de aplicaciones distribuidas en producción.



**Figura 6.8** Logo de PostgreSQL

Se trata de uno de los SGBD de código abierto más empleados y está en continuo mantenimiento, lanzándose versiones cada poco tiempo. Además, la aplicación se desplegará en Heroku, que ofrece integración con este sistema de forma nativa.

#### 6.4.2 Integración del SGBD en nuestro sistema

Una de las ventajas de usar SQLite al principio del desarrollo es que no es necesario realizar ningún tipo de integración, ya que Django genera los proyectos directamente empleándolo.

Dentro del proyecto se generará un fichero `db.sqlite3` en el que se almacenará toda la información de la base de datos.

Cuando se desea añadir nuevas tablas o columnas no es necesario modificar nada más que el código de la aplicación. Se añade el modelo (o campos del modelo) en la clase dedicada a ello y se ejecutan los comandos **`python manage.py makemigrations`** y **`python manage.py migrate`**. Con esto se redefinirá la estructura de la base de datos.

Para realizar consultas, si bien se puede emplear código SQL, Django proporciona, integrada en el *framework*, una API que sirve como interfaz para realizar operaciones sobre la base de datos de forma programática. Como no será necesario realizar consultas complejas, en principio todas ellas serán realizadas empleando este ORM.

El cambio para pasar a emplear PostgreSQL tampoco implica grandes complicaciones. Simplemente habrá que tener una base de datos a la que conectarse y en la configuración del proyecto Django cambiar los parámetros de SQLite por los datos de conexión de este otro sistema, tal que así:

```
DATABASES={
  'default':{
    'ENGINE': 'django.db.backends.postgresql_psycopg2',
    'NAME': 'pdftools',
    'USER': 'postgres',
    'PASSWORD': '',
    'HOST': '127.0.0.1',
    'PORT': '5432',
  }
}
```

Figura 6.9 Configuración para la integración de PostgreSQL en el sistema

El *engine* indicado tendrá que ser instalado como dependencia en el entorno virtual del proyecto.

### 6.4.3 Diagrama E-R

Como la base de datos es muy simple y solo va a estar conformada por dos tablas, **User** y **PDFFile**, y la relación entre ellas es que un **User** puede tener  $n$  **PDFFile** y cada **PDFFile** pertenece a un **User**, se considera que no es necesaria la representación de esta característica en un diagrama entidad-relación.

## 6.5 Diseño de la interfaz

A continuación, se mostrará el resultado del diseño de la interfaz, una vez trasladados los mockups construidos en la fase de análisis a código HTML y CSS.

En esta fase se ha decidido integrar información que puede servir de ayuda en las cabeceras de la página, para que el usuario sepa, además del lugar en que se encuentra, las acciones que puede realizar desde ese lugar.

### 6.5.1 Pantalla de inicio de sesión

En esta pantalla se ha incluido, además de lo ya definido en el mockup, un botón de redirección a la página de registro, para que los usuarios puedan acceder a esta si es que aún no están registrados.

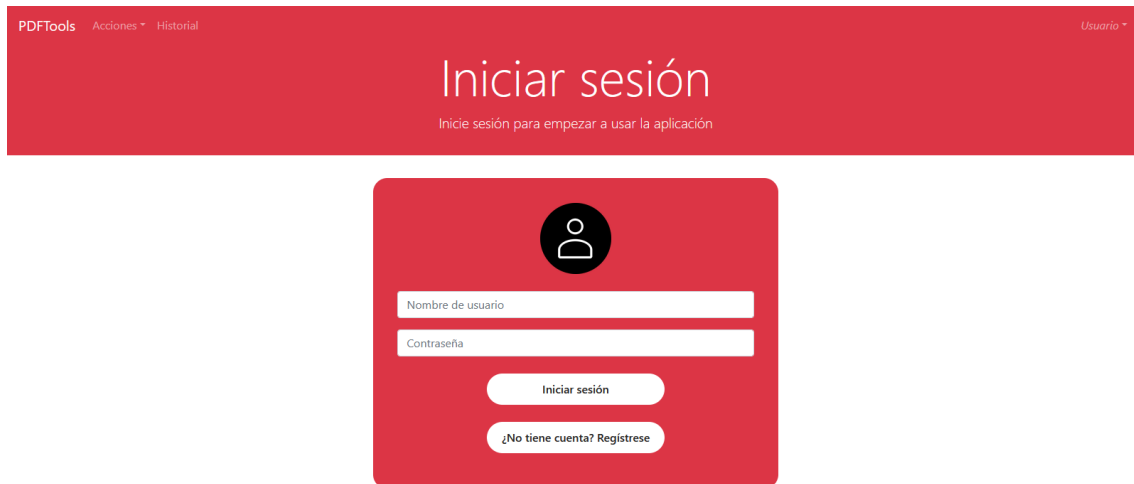


Figura 6.10 Pantalla de inicio de sesión

## 6.5.2 Pantalla de registro

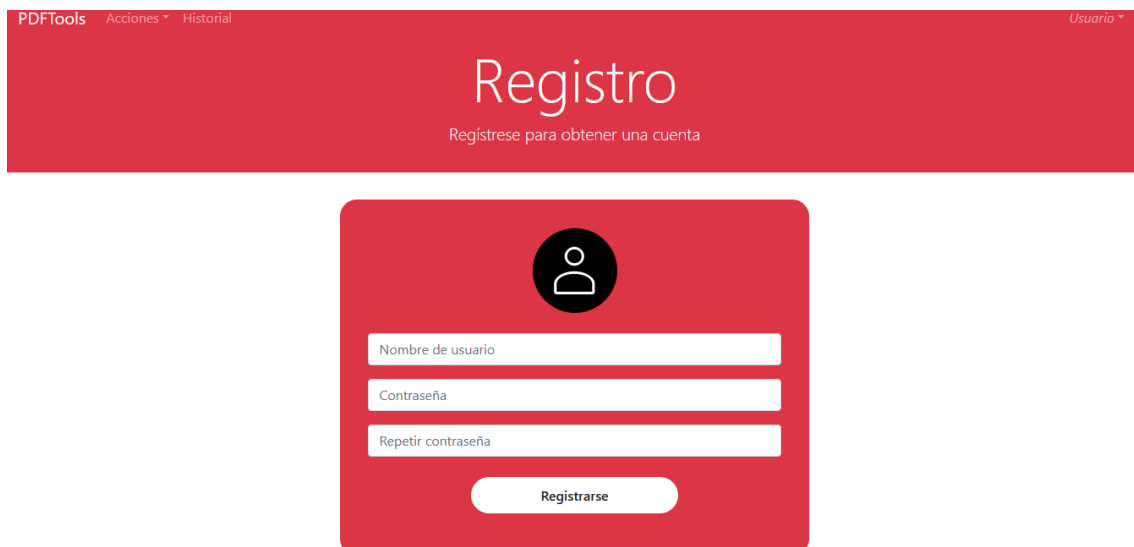


Figura 6.11 Pantalla de registro

## 6.5.3 Pantalla de inicio

Esta pantalla también ha sufrido modificaciones con respecto a lo planteado en el *mockup*. Esto se debe a que se añadió la opción de dividir un PDF en un momento posterior a la realización del diseño inicial. Por tanto, hubo que adaptar la distribución para agregar esta opción, siendo el resultado el siguiente:



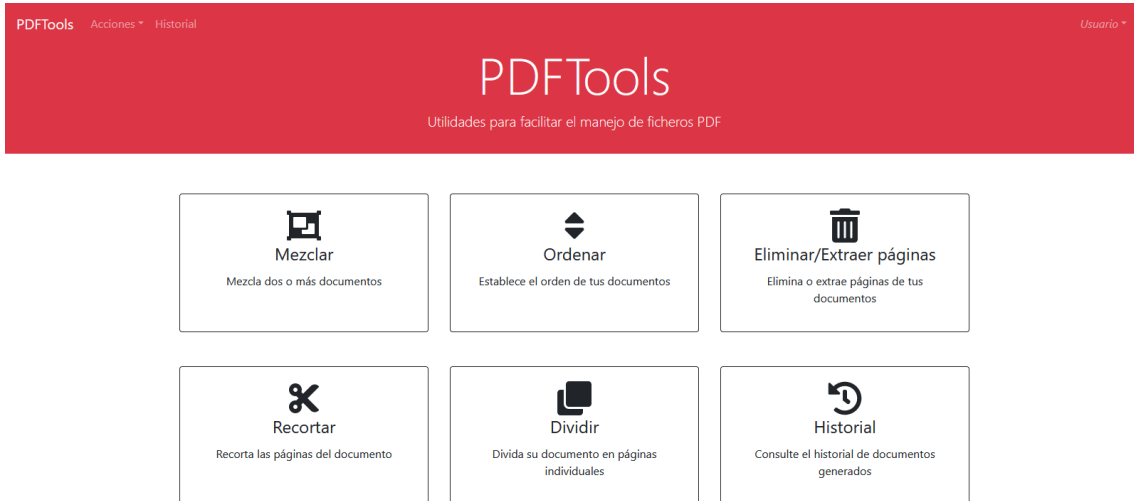


Figura 6.12 Pantalla de inicio

### 6.5.4 Historial de documentos

La pantalla de historial también ha sufrido una modificación, aunque en este caso pequeña. Se ha incluido en la parte superior derecha un indicador del tamaño total de los ficheros almacenados.

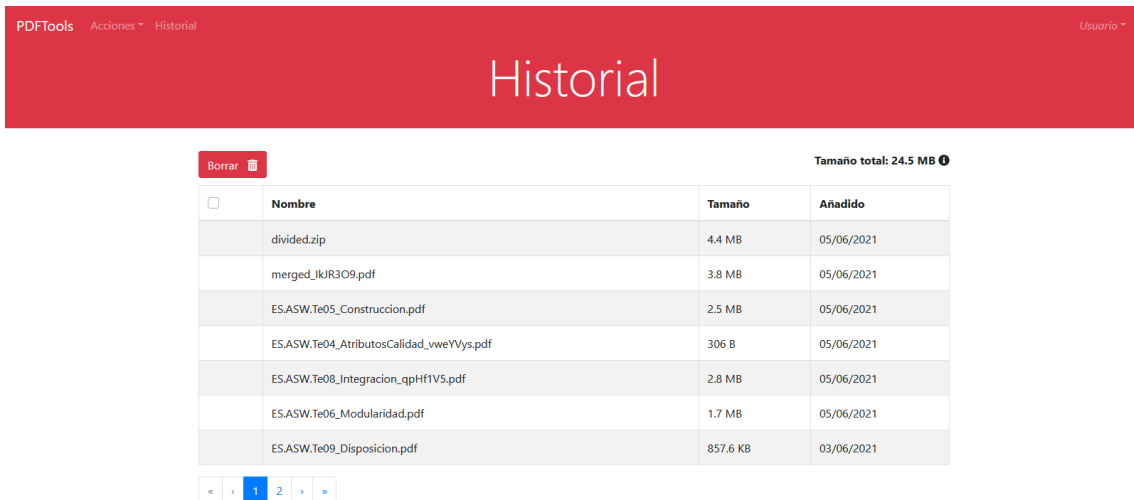


Figura 6.13 Pantalla Historial de ficheros

### 6.5.5 Eliminar/Extraer páginas

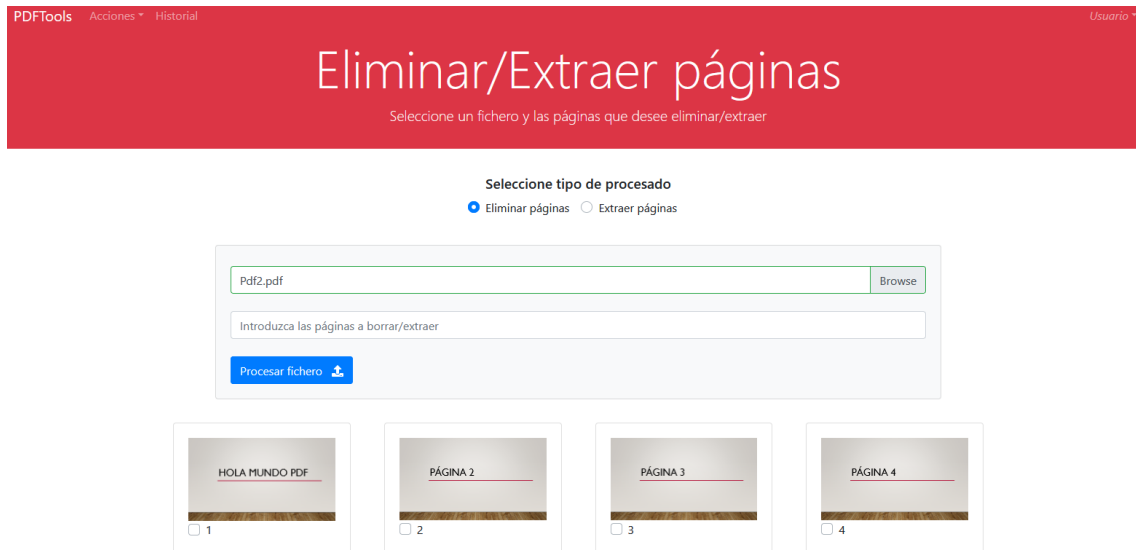


Figura 6.14 Pantalla de eliminación/extracción de páginas

### 6.5.6 Mezclar documentos



Figura 6.15 Pantalla de unión de documentos

### 6.5.7 Ordenar

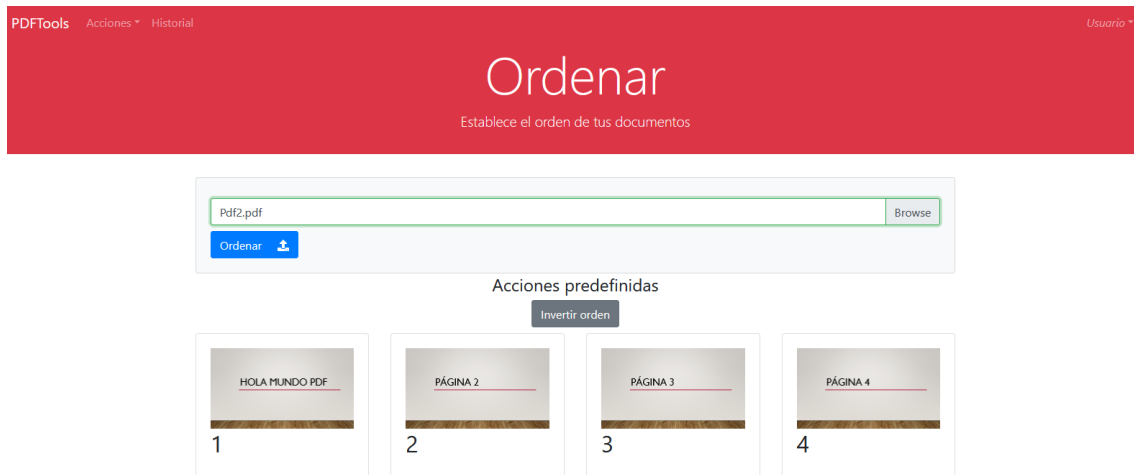


Figura 6.16 Pantalla de ordenación de documentos.

## 6.5.8 Recortar



Figura 6.17 Pantalla de recorte de documentos

## 6.5.9 Dividir

Esta pantalla no está incluida en los mockups puesto que es una funcionalidad que no estaba prevista al inicio del desarrollo. Fue añadida una vez que el proyecto estaba en un estado avanzado y se pasó directamente al diseño.

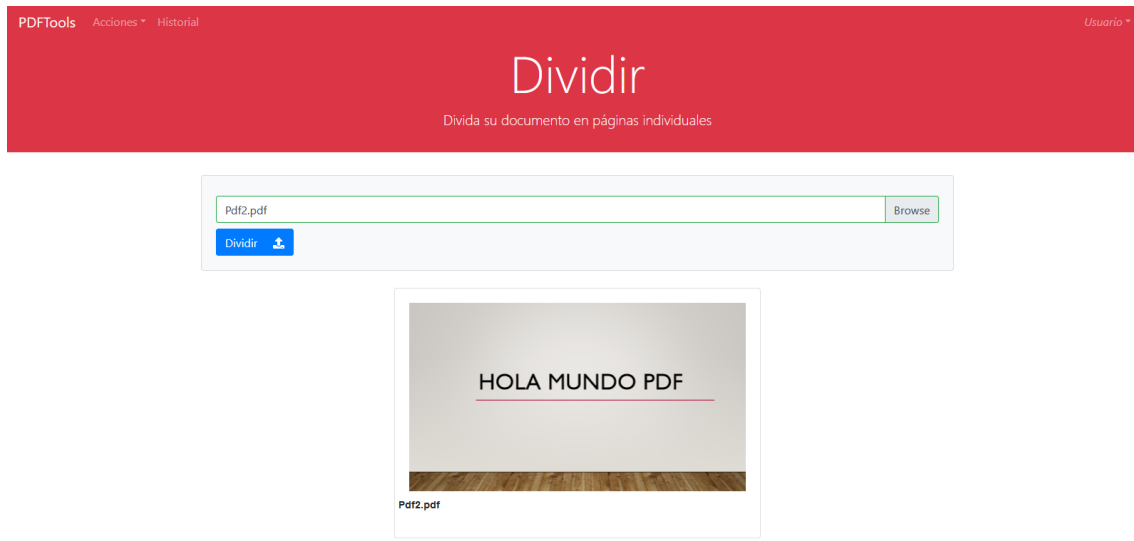


Figura 6.18 Pantalla de división de un fichero

### 6.5.10 Descarga de fichero



Figura 6.19 Pantalla de descarga de un fichero.

## 6.6 Especificación técnica del plan de pruebas

A continuación, se definen las distintas pruebas que se realizarán sobre el sistema.

### 6.6.1 Pruebas funcionales

Se detallan las pruebas funcionales a realizar en cada uno de los casos de uso. La forma de probar estos aspectos se detallará en el apartado *Desarrollo de las Pruebas*.

<b>Registro</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Se trata de registrar un usuario cuyo nombre ya se emplea en el sistema.	Error en la petición HTTP. Se avisa al usuario de que su nombre ya está en uso y no se registra.
Se trata de registrar un usuario con alguno de los campos "en blanco".	Error en la petición HTTP. Se indica al usuario que campo está en blanco (si hay varios el primero de ellos) y no se registra.
Se trata de registrar un usuario con una contraseña de menos de 8 caracteres.	Error en la petición HTTP. Se indica al usuario que la contraseña debe tener al menos 8 caracteres y no se registra.
Se trata de registrar un usuario cuyas contraseñas no coinciden.	Error en la petición HTTP. Se indica al usuario que sus contraseñas no coinciden y no se registra.
Caso límite inválido: Se intenta registrar a un usuario con todos los valores correctos y una contraseña de 7 caracteres.	Error en la petición HTTP. Se indica al usuario que la contraseña debe tener al menos 8 caracteres y no se registra.
Caso límite válido: Se intenta registrar a un usuario con todos los valores correctos y una contraseña de justo 8 caracteres.	Petición HTTP Correcta. Se crea e informa al usuario, se le muestra que el registro ha sido satisfactorio y se le redirige a la pantalla de inicio de sesión.

<b>Inicio de sesión</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Alguno o ambos campos de inicio de sesión en blanco.	Error en la petición HTTP. Se muestra mensaje que indica que la combinación es incorrecta.
Nombre de usuario correcto y contraseña incorrecta.	Error en la petición HTTP. Se muestra mensaje que indica que la combinación es incorrecta.
Combinación correcta.	Petición HTTP Correcta. Se recibe un token en la petición. Se redirige al usuario a la página principal.

<b>Consulta de operaciones disponibles</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Acceder a la página de Inicio.	Se muestran las operaciones que se pueden realizar en el sistema.
Desplegar menú Acciones de la barra de navegación.	Se muestran las operaciones que se pueden realizar en el sistema.
Acceder a cada una de las páginas de las opciones desde el Inicio.	Se muestra en cada caso la página correspondiente.
Acceder a cada una de las páginas de las opciones desde el menú de navegación.	Se muestra en cada caso la página correspondiente.

<b>Casos comunes a todas las operaciones en el Frontend</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Se selecciona un fichero o ficheros (caso de unión) que no es PDF.	Se muestra un mensaje de error indicando que el fichero seleccionado debe de ser un PDF y que ninguno puede superar los X MB, siendo X en principio 10 MB.
Se selecciona uno o más ficheros de más de X MB de tamaño, siendo X en principio 10 MB.	Se muestra un mensaje de error indicando que el fichero seleccionado debe de ser un PDF y que ninguno puede superar los X MB, siendo X en principio 10 MB.

<b>Extracción de páginas de un documento</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
<i>Frontend:</i> Carga de un documento PDF de tamaño correcto.	Se obtiene su número de páginas a través de una petición HTTP y se muestra una previsualización de todas las páginas, cada una de ellas tiene un checkbox.
<i>Frontend:</i> Seleccionar una única página desde el componente de texto.	Se marca el checkbox de la página correspondiente.
<i>Frontend:</i> Borrar la página desde el componente de texto.	Se desmarca el checkbox de la página correspondiente.
<i>Frontend:</i> Seleccionar una única página desde el checkbox.	Se muestra la página en el componente de texto.

<i>Frontend:</i> Desmarcar la página desde el checkbox.	Se borra la página del componente de texto.
<i>Frontend:</i> Se introducen varias páginas consecutivas separadas por comas en el componente de texto. Por ejemplo: 1,2,3,4,5.	Se convierten en un rango (1-5) y se marcan los checkbox correspondientes.
<i>Frontend:</i> Se introducen varias páginas no consecutivas separadas por comas en el componente de texto. Por ejemplo: 1,3,7.	Se quedan separadas por comas y se marcan los checkbox correspondientes.
<i>Frontend:</i> Se desmarcan todos los checkbox.	Se borran las páginas del componente de texto.
<i>Frontend:</i> Se marcan páginas consecutivas en el checkbox: 1 a 5 por ejemplo.	Se convierten en un rango (1-5) en el componente de texto.
<i>Frontend:</i> Se marcan páginas consecutivas en el checkbox: 1 ,3 y 5 por ejemplo.	Aparecen separadas por comas en el componente de texto.
<i>Frontend:</i> Se introducen dos comas seguidas en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.
<i>Frontend:</i> Se introducen dos guiones seguidos en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.
<i>Frontend:</i> Se introduce una coma y un guion seguidos en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.
<i>Frontend:</i> Se introduce un carácter que no es numérico “,” o “-”.	No se añade al componente de texto.
<i>Frontend:</i> Se introduce una página o rango de páginas en el componente de texto que no tiene el fichero.	No ocurre nada.
<i>Frontend:</i> Se seleccionan varios rangos de páginas en los checkbox. Por ejemplo 1-3,5-7.	Se traslada correctamente esta información al componente de texto (1-3,5-7).
<i>Frontend:</i> Se seleccionan varios rangos de páginas en el componente de texto.	Se traslada correctamente esta información a los checkbox.
<i>Frontend:</i> Se trata de procesar fichero con el listado de páginas vacío.	Se muestra un mensaje de error indicando que el formato no es válido.
<i>Procesamiento:</i> Se extrae un listado de páginas separadas por comas 1,3,5,9 Por ejemplo (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene 4 páginas y son las esperadas.
<i>Procesamiento:</i> Se extrae un listado de páginas separadas por guion (rango 1-4) por ejemplo (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene 4 páginas y son las esperadas.

<i>Procesamiento:</i> Caso límite, se extraen todas las páginas.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas del original.
<i>Procesamiento:</i> Caso límite. Se indica únicamente una página dentro de las seleccionadas que no tiene el fichero.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante es un fichero vacío.
<i>Procesamiento:</i> Caso límite. Se indica una página dentro de las seleccionadas que no tiene el fichero y otras que sí que existen.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante contiene las páginas que existían en el fichero únicamente.

<b>Eliminación de páginas de un fichero</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
<i>Frontend:</i> Todos los casos cubiertos en la extracción, ya que comparten componentes.	Revisar casos de prueba <b>Extracción de páginas de un fichero.</b>
<i>Procesamiento:</i> Se elimina un listado de páginas separadas por comas 1,3,5,9 por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas menos las que se eliminan.
<i>Procesamiento:</i> Se elimina un listado de páginas separadas por guion (rango 1-4) por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas menos las indicadas.
<i>Procesamiento:</i> Caso límite, se eliminan todas las páginas.	Se procesa correctamente, se redirige a la página de descarga, el fichero resultante es un fichero vacío.
<i>Procesamiento:</i> Caso límite. Se indica únicamente una página dentro de las seleccionadas que no tiene el fichero.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante es un fichero con todas las páginas originales.
<i>Procesamiento:</i> Caso límite. Se indica una página dentro de las seleccionadas que no tiene el fichero y otras que sí que existen.	Se procesa correctamente. Se redirige a la página de descarga. El fichero tendrá todas las páginas originales excepto las que si existían y se indicaron.

<b>Unir ficheros</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>



<i>Frontend:</i> Seleccionar un único fichero.	Se muestra un mensaje indicando que al menos deben de ser dos ficheros para mezclar/unir.
<i>Frontend:</i> Seleccionar más de 5 ficheros.	Se muestra un mensaje indicando que el número máximo de ficheros menos para mezclar/unir es 5.
<i>Frontend:</i> Se introduce un número correcto de ficheros.	Se muestran una vista previa de los ficheros y se puede variar el orden arrastrando uno sobre el otro.
<i>Procesamiento:</i> Se mezclan dos ficheros sin cambiar el orden	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene tantas páginas como la suma de los dos y se ha unido en el orden correcto.
<i>Procesamiento:</i> Se mezclan dos ficheros alterando el orden	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene tantas páginas como la suma de los dos y se ha unido en el orden indicado.

<b>Ordenar</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
<i>Frontend:</i> Seleccionar un fichero.	Se carga una previsualización de todas las páginas del fichero en el orden inicial, indicando para cada uno el número de página. Se muestran las acciones predefinidas.
<i>Frontend:</i> Pulsar el botón de invertir el orden sin realizar ninguna modificación.	Se invierte el orden del documento, pasando a ser la última página la primera y la primera la última.
<i>Frontend:</i> Pulsar el botón de invertir el orden dos veces consecutivas.	El orden de las páginas es el mismo que el inicial.
<i>Frontend:</i> Intercambiar el orden de dos páginas consecutivas, 5 y 6 por ejemplo.	La página 5 pasa a ocupar la posición 6 y la página 6 la posición 5.
<i>Frontend:</i> Mover el orden de dos páginas no consecutivas. Por ejemplo, página 3 a la posición 6.	La página 3 pasa a ocupar la posición 6. Páginas 4,5 y 6 se mueven a una posición inferior
<i>Procesamiento:</i> Calcar el botón de ordenar sin ningún tipo de cambio.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante

	tiene las mismas páginas y en el mismo orden que el inicial.
<i>Procesamiento:</i> Calcar el botón de ordenar justo tras invertir el orden.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene las mismas páginas que el inicial, pero en el orden inverso.
<i>Procesamiento:</i> Alterar la posición de algunas páginas manualmente y calcar en el botón ordenar.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene las mismas páginas que el inicial, en el orden indicado por el usuario.

<b>Recortar</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
<i>Frontend:</i> Seleccionar un fichero.	Se muestra una previsualización de la primera página de un fichero, con una rejilla encima que se puede desplazar y cambiar sus límites.
<i>Frontend:</i> Mover la rejilla.	La rejilla se traslada de la forma esperada.
<i>Frontend:</i> Cambiar la posición de cada uno de los ocho puntos de la rejilla.	La rejilla pasa a tener el tamaño esperado.
<i>Frontend:</i> Aumentar el tamaño de la rejilla al máximo posible.	En ningún momento supera el tamaño del contenedor del documento.
<i>Procesamiento:</i> Recortar con distintos tamaños de rejilla.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene visible tan solo el contenido de la rejilla.
<i>Procesamiento:</i> Recortar un fichero rotado.	Se procesa correctamente a pesar de esto, coincidiendo el área seleccionada por el usuario con el área que se recorta.

<b>Dividir</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
<i>Frontend:</i> Seleccionar un fichero.	Se muestra una previsualización de la primera página junto con el nombre del fichero.
<i>Procesamiento:</i> Dividir el fichero.	Se procesa correctamente y se redirige a descargas. Si descargamos el resultado es un

	zip, con tantos PDF dentro como páginas tenía el fichero y cada uno de ellos tiene una página.
--	--

<b>Historial</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Acceder al historial de ficheros sin tener ficheros procesados.	La tabla se muestra vacía y el tamaño total de los ficheros es 0 Bytes.
Acceder al historial de ficheros teniendo ficheros procesados.	Se muestra una entrada en la tabla para cada fichero, indicando nombre, tamaño y fecha en la que fue añadido. Están ordenados por fecha descendente. El tamaño total es la suma del tamaño de los ficheros.
Acceder al historial de ficheros teniendo gran número de ficheros procesados.	Se muestran los ficheros en distintas páginas, con la información que se indicó en el apartado anterior.

<b>Descargar</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Descargar un fichero desde la página del historial.	El navegador descarga el fichero correctamente.

<b>Borrar documento</b>	
<b>Caso de prueba</b>	<b>Resultados esperados</b>
Seleccionar un documento y borrarlo.	El documento desaparece de la tabla, de la base de datos y del sistema de ficheros.
Seleccionar más de un documento y borrarlos.	Los documentos desaparecen de la tabla, de la base de datos y del sistema de ficheros.
Seleccionar todos los documentos de una página y borrarlos cuando solo hay una página.	La tabla queda vacía. Los documentos se borran de la base de datos y del sistema de ficheros.
Seleccionar todos los documentos de una página y borrarlos cuando hay más de una página.	En la tabla pasa a mostrarse la información de la siguiente página.

## 6.6.2 Pruebas de usabilidad

Para realizar este tipo de pruebas se seleccionará una serie de usuarios (3 en total) de diferentes edades y distinto nivel informático. Se les solicitará que realicen un grupo de tareas en la aplicación y que, una vez realizadas, rellenen un cuestionario con información sobre estas. Se hacen con el fin de evaluar una serie de aspectos:

- **Curva de aprendizaje:** Se tratará de comprobar cuanto tiempo le lleva a un usuario aprender a usar el sistema de forma correcta, buscando los puntos en los que más problemas aparecen.
- **Satisfacción:** Se estudiará el grado de satisfacción a nivel general de los usuarios al usar la aplicación.
- **Detección de errores:** Estas pruebas también permitirán detectar errores, defectos, o aspectos con los que los usuarios no se encuentran cómodos.
- **Sugerencias de mejora:** Se solicitará a los usuarios una opinión, pidiéndoles sugerencias de mejora para futuras ampliaciones del sistema.

### 6.6.2.1 Cuestionario de evaluación de la usabilidad

El cuestionario se dividirá en dos partes:

1. Se harán preguntas al usuario que no tienen nada que ver con el sistema implementado, con el objetivo de conocer su nivel informático y características personales.
2. Se solicitará al usuario que realice una serie de tareas, indicándole únicamente cuál es el propósito de la aplicación y las tareas que debe realizar. Cuando acabe estas, se le realizarán preguntas sobre las mismas y sobre la aplicación en general, una vez que tiene una visión global del sistema.

A continuación, se presenta el cuestionario que se pedirá a los usuarios que rellenen.

#### 6.6.2.1.1 Preguntas iniciales

<b>Edad</b>
<b>Profesión</b>
<b>¿Con qué frecuencia utiliza un ordenador o dispositivo similar?</b>
<ul style="list-style-type: none"> <li>• Todos los días.</li> <li>• Más de una vez por semana.</li> <li>• Raras veces.</li> </ul>
<b>¿Cómo calificaría su nivel informático?</b>

<ul style="list-style-type: none"><li>• Bajo.</li><li>• Medio.</li><li>• Alto.</li></ul>
<b>Cuando lo hace, ¿Para qué emplea un ordenador? (Puede seleccionar varias opciones)</b>
<ul style="list-style-type: none"><li>• Navegar.</li><li>• Consultar el correo electrónico.</li><li>• Ocio (jugar a videojuegos, ver películas o vídeos...).</li><li>• Es mi herramienta de trabajo principal.</li><li>• Hago uso de aplicaciones de ofimática.</li></ul>
<b>¿Sabe qué es un documento PDF?</b>
<ul style="list-style-type: none"><li>• Sí, los uso en mi día a día.</li><li>• Sí, y los uso de vez en cuando.</li><li>• Sé lo que es, pero nunca he usado uno.</li><li>• No.</li></ul>
<b>¿Alguna vez ha tratado de manipular ficheros PDF?</b>
<ul style="list-style-type: none"><li>• Sí.</li><li>• No.</li></ul>
<b>¿De entre estos aspectos, cuál valoraría más en una aplicación?</b>
<ul style="list-style-type: none"><li>• Que sea rápida.</li><li>• Que se comprendan rápido todas sus funcionalidades.</li><li>• Que tenga gran número de funcionalidades.</li></ul>

#### 6.6.2.1.2 Realización de tareas y preguntas sobre el sistema

Se solicitará al usuario que realice una serie de tareas:

- Registro en el sistema.
- Inicio de sesión.
- Extraer páginas de un documento.
- Unir documentos.
- Recortar páginas.
- Ordenar un documento.
- Dividir un documento.
- Consultar el historial de documentos.
- Borrar un documento.

Una vez realizadas se le pedirá que rellene el siguiente cuestionario.

<b>¿Ha conseguido realizar todas las tareas?</b>
--

- Sí, con facilidad.
- Sí, pero con complicaciones.
- No.

**Si ha tenido complicaciones o no ha podido completarlas indique aquí el motivo**

**¿Considera que la aplicación es rápida?**

- Sí.
- Sí al navegar, pero no los tiempos de espera a la hora de procesar documentos.
- No.

**¿Se ajusta la forma en la que hay que desarrollar cada tarea a lo que pensaba que tendría que hacer?**

- Sí, en todas.
- En la mayoría.
- Solo en algunas.
- En ninguna.

**¿Es capaz de navegar rápidamente por las distintas pantallas?**

- Sí.
- No del todo, pero sí habitualmente.
- No, la navegación no es intuitiva.

**Conoce en qué página se encuentra**

- En todo momento.
- Algunas veces.
- Nunca.

**¿Le resulta fácil el uso del sistema en conjunto?**

- Sí.
- No.

**¿Considera que los tamaños de letra son correctos?**

- Sí, en toda la aplicación.
- En algunas pantallas.
- No.

**¿Considera que la combinación de colores empleada es correcta?**

- Sí.
- No.

**¿Considera el diseño en general atractivo?**

- Sí.
- No.

**¿Cumple el sistema con su propósito?**

- Sí.
- No.

**Observaciones. Indique en este cuadro de texto aquellos aspectos que le han resultado extraños, sugerencias de mejora, funcionalidades que considera interesante añadir...**

# Capítulo 7. Implementación del sistema

Dentro de este apartado se recogerán aspectos relacionados con la implementación del sistema: herramientas empleadas, *stack* tecnológico, normativa seguida...

## 7.1 Estándares y Normas Seguidos

A continuación, se detallan algunos de los estándares, normas o prácticas que se han utilizado en la implementación del sistema.

### 7.1.1 JSON

**JSON**, que proviene de **JavaScript Object Notation**, es un formato ideado para el intercambio de datos buscando que sea ligero y comprensible [32].

En la aplicación se utiliza como medio de comunicación en las peticiones entre la aplicación web y el servidor web, tanto para el envío como la recepción de datos de la parte frontal.

Este estándar está recogido por **ECMA** en [33].

### 7.1.2 Vue Style Guide

Sin tratarse de un estándar, desde la propia página de Vue proporcionan una serie de recomendaciones aplicables al código de un proyecto que se realiza utilizando este marco de trabajo. En la implementación del sistema se han intentado seguir estas recomendaciones de la forma más fiel posible para evitar caer en malas prácticas o antipatrones. Esta guía de estilo se puede encontrar en [34].

## 7.2 Lenguajes de programación

En la implementación del sistema se han empleado dos lenguajes, Python y JavaScript, cada uno en su subsistema correspondiente.

### 7.2.1 Python 3



Figura 7.1 Logo de Python



Es un lenguaje interpretado, de tipado fuerte y dinámico, que soporta múltiples paradigmas [35]. Se trata de uno de los más empleados hoy en día, concretamente el segundo a mayo de 2021, según [36].

Sus características hacen que sea un lenguaje de programación de propósito general, ya que se puede emplear para realizar aplicaciones de muchos campos distintos (juegos, desarrollo web, *machine learning*...) [37].

Python, en su versión **3.8.1**, ha sido el lenguaje con el que se ha construido el servidor web del sistema, empleando **Django**, y más concretamente su variante **Django Rest Framework**, para exponer una API que permite, entre otras cosas, la manipulación de ficheros PDF.

## 7.2.2 JavaScript

Al igual que Python es un lenguaje interpretado y multiparadigma. En cuanto a su sistema de tipos es de tipado débil y dinámico. Está basado en el estándar ECMAScript [38]. Su “cuota de mercado” es también bastante alta, estando según [36] en el puesto 7 de lenguajes más usados en mayo de 2021.



Figura 7.2 Logo de JavaScript

Javascript, junto con el *framework* **Vue**, ha sido el lenguaje utilizado para implementar la parte frontal del sistema, con la que el usuario interactuará directamente y que se comunicará con el servidor web para proporcionar la funcionalidad completa.

## 7.2.3 Otros lenguajes

En este subapartado se recogen lenguajes que, si bien no son lenguajes de programación, son parte importante de la codificación del sistema y se considera relevante reflejar brevemente su empleo.

### 7.2.3.1 HTML

**HTML** del inglés *Hypertext Markup Language*, es un lenguaje de marcado que define la estructura y el significado de un sitio web. Con las marcas o etiquetas se estructuran los diferentes contenidos que aparecerán en la web [39].



Figura 7.3 Logo de HTML

La versión 5 de HTML se ha empleado dentro del ámbito de Vue, para definir las “plantillas” de las distintas vistas o componentes.

#### 7.2.3.2 CSS

**CSS**, que proviene del inglés *Cascading Style Sheets* u **Hojas de Estilo en Cascada** en español, es un lenguaje que permite establecer estilos para definir el diseño visual de documentos XML [40].



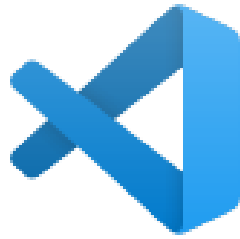
Figura 7.4 Logo de CSS

El uso que se le da en la aplicación explícitamente es limitado ya que se emplea la librería Bootstrap para la maquetación de las distintas páginas y no el lenguaje directamente. No obstante, en algunas vistas y componentes se usa para hacer pequeños ajustes en los estilos.

## 7.3 Herramientas y Programas Usados para el desarrollo

En este subapartado se mencionará a rasgos generales la infraestructura empleada para el desarrollo del proyecto, herramientas que no forman parte de este, pero son primordiales para construirlo.

### 7.3.1 Visual Studio Code



**Figura 7.5 Logo de Visual Studio Code**

Visual Studio Code es un ligero editor de código multiplataforma, con soporte nativo para Javascript, TypeScript y Node.js, pero con numerosas extensiones que permiten el desarrollo en muchos otros lenguajes [41].

Ha sido desarrollado por Microsoft, pero es gratuito y de código abierto, el cual está disponible en GitHub. Una de sus principales características es la gran cantidad de extensiones que se pueden añadir al editor facilitando el proceso de desarrollo [41].

- Página Web: <https://code.visualstudio.com/>
- Repositorio GitHub: <https://github.com/microsoft/vscode>

En el proyecto ambos subsistemas han sido desarrollados empleando este editor.

### 7.3.2 PowerShell



**Figura 7.6 Logo de PowerShell**

Se trata de una interfaz de consola, desarrollada por Microsoft, que permite entre muchas otras cosas la ejecución de comandos. Información más detallada se encuentra en [42].

Durante el desarrollo se ha empleado para ejecutar diversas tareas, como instalar dependencias de npm, arrancar los servidores de desarrollo de ambos subsistemas, ejecutar comandos de git...

### 7.3.3 Git



Figura 7.7 Logo de Git

Git es un sistema de control de versiones libre y de código abierto, cuyo objetivo es mantener distintas versiones del código de una aplicación de una forma eficiente y confiable a través del registro de los cambios en los archivos en un repositorio de código [43].

Su funcionamiento se basa en la ramificación, lo que permite trabajar y realizar cambios sobre una rama de una forma totalmente independiente a lo que se está desarrollando en otras ramas [43].

- Página web: <https://git-scm.com/>

Durante el desarrollo se usará Git como sistema de control de versiones del código.

### 7.3.4 Postman



Figura 7.8 Logo de POSTMAN

Postman es una plataforma que proporciona herramientas muy útiles en el desarrollo de APIs REST. Uno de los principales usos que tiene es el funcionamiento como cliente de estas API, sin necesidad de realizar ningún tipo de desarrollo que consuma los servicios [44].

- Página web: <https://www.postman.com/>

En el sistema se ha empleado para comprobar, en momentos en los que aún no estaba construido el *frontend*, que el subsistema servidor se comportaba correctamente y realizaba las operaciones esperadas en cada caso. Una vez en fase de pruebas, se empleó para realizar pruebas funcionales del sistema aislando el servidor web.

### 7.3.5 Google Chrome



**Figura 7.9 Logo de Google Chrome**

Se trata del navegador web con mayor cuota de mercado [45]. Proporciona las funcionalidades típicas de cualquier navegador y, además, aporta al desarrollador una serie de herramientas para depurar, detectar errores o comprobar la adaptabilidad de la aplicación a distintos tamaños entre otras.

- Página web: <https://www.google.com/intl/es/chrome/>

En el desarrollo se ha usado para comprobar el correcto funcionamiento de las distintas versiones de la aplicación que se han ido generando. Las versiones empleadas han sido la 89.0.4389.128 y posteriores.

## 7.4 Creación del sistema

Se recogen en esta sección los aspectos que ido apareciendo durante la creación del sistema.

### 7.4.1 Problemas Encontrados

Durante todo el proyecto y sobre todo en la etapa de desarrollo, han aparecido distintos problemas, los cuales se han ido subsanando o mitigando a medida que eran detectados. Se recogen a continuación algunos de ellos junto al tratamiento aplicado en cada caso.

#### *7.4.1.1 Falta de experiencia con el stack tecnológico seleccionado*

El enfoque que se le dio inicialmente al proyecto fue, además de obviamente la realización de este, obtener conocimientos en tecnologías que durante el grado no habían sido empleadas y que consideraba interesantes. Esto trajo consigo consecuencias, ya que ralentizó la etapa de desarrollo.

Si bien es cierto que los lenguajes empleados en el desarrollo (Python y Javascript) no eran completamente desconocidos al inicio del desarrollo, tampoco eran en mi caso particular lenguajes en los que tuviese un conocimiento avanzado, como puede ser el caso de Java, al ser el lenguaje más utilizado durante el grado.

A esto se le sumó la selección de los *framework* base para los dos subsistemas, **Vue.js** para el *frontend* y **Django Rest Framework** para el *backend*; que no había utilizado nunca y en los que partía de cero en la curva de aprendizaje.

Por estos motivos, la dificultad que ya existe implícita en el desarrollo de un producto software se vio notablemente incrementada por la necesidad de asimilar estos nuevos conocimientos.

No ha existido como tal una solución a este problema, más bien una forma de mitigarlo, al realizar cursos online de las tecnologías utilizadas antes de comenzar la implementación del sistema. Considero que fue de gran ayuda, especialmente en las primeras etapas del desarrollo.

#### 7.4.1.2 Problemas de dependencias en paquetes npm

En el *frontend* se ha usado un gran número de librerías para facilitar el desarrollo de distintas tareas, como por ejemplo realizar la maquetación, permitir la navegación entre páginas de la aplicación web, establecer la comunicación entre subsistemas o permitir el *drag and drop*.

Las versiones de estas librerías o paquetes que son instaladas usando el gestor de paquetes npm presentan a veces incompatibilidades entre sí, haciendo imposible desplegar el servidor de desarrollo.

Este problema se presentó únicamente al principio del desarrollo, en el periodo en el que se estaban seleccionando los elementos externos que podrían facilitar el trabajo. Una vez que se tuvo un conjunto de librerías estable no volvió a ocurrir.

Para solucionar este problema se buscó información en los repositorios de Github o en StackOverflow, que suelen tener registradas este tipo de incidencias. La solución final en estos casos ha sido corregir, normalmente usando una versión anterior estable, las versiones empleadas en el fichero `packages.json`.

#### 7.4.1.3 Problemas con el sistema de ficheros de Django

##### 7.4.1.3.1 Seguridad en el acceso a ficheros

Este defecto fue detectado en una fase avanzada del desarrollo del sistema.

La forma estándar de guardar ficheros en un programa desarrollado con Django Rest es empleando su sistema de ficheros. Para esto, en la clase modelo que representa una tabla de base de datos se declara un campo (equivalente a una columna) de tipo `FileField`.

Al guardar un fichero este campo indicará la ruta de este dentro del sistema de ficheros de Django, que está definida en un fichero de configuración.

El enfoque que se había dado en un principio para la descarga de ficheros era el siguiente: se enviaba una petición GET para descargar un fichero y se consultaba a partir del id cuál era su ruta a través del campo `FileField`, devolviendo como respuesta a la petición una URL en la que se incluía la ruta completa del fichero dentro del servidor.

Al no haber alterado el comportamiento por defecto del sistema de ficheros, todos los PDF generados por los usuarios eran almacenados en un mismo directorio, por lo que una vez que

un usuario tuviese la URL de un fichero, con simplemente conocer el nombre de otro archivo (de cualquier usuario), tendría acceso a este.

Para solucionar este problema se adoptaron inicialmente dos medidas:

1. Por una parte, se establece un subdirectorio para cada usuario. Dentro de la carpeta *media*, en la que anteriormente se almacenaban todos los ficheros, habrá una carpeta por cada usuario del sistema.

Esto se consigue a través del campo `FileField`, que permite definir una ruta concreta en la que se guardará el fichero. Para que cada usuario tenga una ruta única, esta es obtenida mediante una técnica de ofuscación, la cual a partir del nombre de usuario y una sal, genera un hash MD5. Este hash será el nombre de la subcarpeta del usuario. Por tanto, para acceder a un fichero de otra persona ya no hay que conocer únicamente el nombre del documento, también se debe conocer el hash asociado a su nombre.

2. Por otra parte, se dejó de enviar la URL como respuesta a la petición de descarga para que el usuario descargase directamente el fichero, pasando a enviar un fichero binario. A partir de este, el *frontend* genera una URL temporal con el método `URL.createObjectURL()` [46].

#### 7.4.1.3.2 Almacenamiento de ficheros en producción

Una vez que el sistema iba a ser desplegado, se detectó además que los ficheros no podían ser almacenados en un directorio del proyecto ni en una carpeta del sistema de Heroku, ya que las máquinas de este borran los archivos guardados cada cierto tiempo. Se explica esto para el SGBD SQLite en este artículo [47], pero aplica también a los ficheros.

Como solución a este problema se optó por usar un sistema de almacenamiento de ficheros distribuido, decantándome por la opción ofrecida por Azure, Azure Blob Storage [28].

Para adaptarse a este nuevo sistema de ficheros se hicieron cambios en el código de la aplicación de Django: se especificaron las propiedades de conexión con Azure y se estableció como sistema de almacenamiento por defecto, customizando la forma en la que se comporta:

```
DEFAULT_FILE_STORAGE = 'tfg.custom_azure.AzureMediaStorage'  
AZURE_ACCOUNT_NAME = 'tfgpdftools'  
AZURE_ACCOUNT_KEY = 'rVi5RvJsN+0h84vzN4jnUcHvV0a8nAM12Us9CBWLM'  
AZURE_CONTAINER='pdftools'  
AZURE_CONNECTION_STRING='DefaultEndpointsProtocol=https;Accou'
```

Figura 7.10 Configuración de Cuentas y Claves de Azure (Las claves se muestran recortadas)

```
from storages.backends.azure_storage import AzureStorage

class AzureMediaStorage(AzureStorage):
    location = 'media'
    file_overwrite = False
```

**Figura 7.11 Configuración de Azure. Se establece carpeta por defecto y que no se deben sobrescribir ficheros (se renombrarán en su lugar)**

De esta forma se aumenta también la seguridad, ya que para acceder a un fichero de este sistema se necesita la clave de la cuenta o el *string* de conexión.

#### 7.4.1.3.3 Conclusión

Con estos cambios, el usuario final de la aplicación deja de tener un conocimiento directo de la ruta en la que se encuentran los archivos que se almacenan en el sistema de ficheros, por lo que solo tendrá acceso a ellos desde la aplicación y no podrá acceder a los archivos de otros usuarios, aumentando la seguridad del sistema.

## 7.4.2 Descripción Detallada de las clases

En este subapartado se recoge la documentación del código fuente de la aplicación de ambos subsistemas. Esta documentación ha sido generada a partir de herramientas y se indicarán las rutas a los documentos en los que se pueden consultar.

### 7.4.2.1 Descripción subsistema Servidor Web

Se ha generado la documentación empleando la herramienta Sphinx, y se encuentra, dentro del proyecto del *backend*, en la ruta **pdftools/docs/\_build/html/index.html**.

### 7.4.2.2 Descripción subsistema Aplicación Web

Se ha tratado de generar la documentación empleando la herramienta documentation.js [48]. No obstante, no funcionaba bien con algunos archivos del proyecto y se completó manualmente. Se encuentra, tanto en formato HTML como markdown, dentro del proyecto del *frontend* en el directorio **docs**.



## Capítulo 8. Desarrollo de las Pruebas

En este capítulo se recogerán los resultados de la realización de las pruebas del sistema, tanto funcionales como de usabilidad.

### 8.1 Pruebas funcionales

Estas pruebas han sido realizadas en base a los casos de prueba extraídos en el apartado *Especificación técnica del plan de pruebas*. Se han realizado de diversas formas y en distintos momentos del proyecto.

Durante el desarrollo, se han ido probando mediante la interacción directa con el sistema o realizando peticiones HTTP con la herramienta Postman. Además, se han implementado para las funcionalidades que se consideran más importantes (procesamiento de los ficheros) pruebas unitarias automatizadas.

Una vez desplegado el sistema se han ejecutado las pruebas unitarias y, manualmente, se han probado todos los casos de uso registrados en el sistema en producción.

A continuación, se recogen en tablas los resultados de las pruebas, indicando si el resultado ha coincidido con el esperado o existe alguna variación. Si hubiese alguna variación se señala cuál ha sido esta y la corrección. Estas variaciones han podido aparecer en cualquier momento del desarrollo, no se refieren a errores detectados una vez que el sistema se desplegó.

#### 8.1.1 Pruebas “manuales”

Registro		
Caso de prueba	Resultados esperados	Prueba satisfactoria
Se trata de registrar un usuario cuyo nombre ya se emplea en el sistema.	Error en la petición HTTP. Se avisa al usuario de que su nombre ya está en uso y no se registra.	<b>Sí</b>
Se trata de registrar un usuario con alguno de los campos “en blanco”.	Error en la petición HTTP. Se indica al usuario que campo está en blanco (si hay varios el primero de ellos) y no se registra.	Se detecta que no se hace la comprobación, pero falla el registro. Se añade una comprobación para “avisar al usuario.

Se trata de registrar un usuario con una contraseña de menos de 8 caracteres.	Error en la petición HTTP. Se indica al usuario que la contraseña debe tener al menos 8 caracteres y no se registra.	<b>Sí</b>
Se trata de registrar un usuario cuyas contraseñas no coinciden.	Error en la petición HTTP. Se indica al usuario que sus contraseñas no coinciden y no se registra.	<b>Sí</b>
Caso límite inválido: Se intenta registrar a un usuario con todos los valores correctos y una contraseña de 7 caracteres.	Error en la petición HTTP. Se indica al usuario que la contraseña debe tener al menos 8 caracteres y no se registra.	<b>Sí</b>
Caso límite válido: Se intenta registrar a un usuario con todos los valores correctos y una contraseña de justo 8 caracteres.	Petición HTTP Correcta. Se crea e informa al usuario, se le muestra que el registro ha sido satisfactorio y se le redirige a la pantalla de inicio de sesión.	<b>Sí</b>

#### Inicio de sesión

<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
Alguno o ambos campos de inicio de sesión en blanco.	Error en la petición HTTP. Se muestra mensaje que indica que la combinación es incorrecta.	<b>Sí</b>
Nombre de usuario correcto y contraseña incorrecta.	Error en la petición HTTP. Se muestra mensaje que indica que la combinación es incorrecta.	<b>Sí</b>
Combinación correcta.	Petición HTTP Correcta. Se recibe un token en la petición. Se redirige al usuario a la página principal.	<b>Sí</b>

#### Consulta de operaciones disponibles

<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
Acceder a la página de Inicio.	Se muestran las operaciones que se pueden realizar en el sistema.	<b>Sí</b>

Desplegar menú Acciones de la barra de navegación.	Se muestran las operaciones que se pueden realizar en el sistema.	Sí
Acceder a cada una de las páginas de las opciones desde el Inicio.	Se muestra en cada caso la página correspondiente.	Sí
Acceder a cada una de las páginas de las opciones desde el menú de navegación.	Se muestra en cada caso la página correspondiente.	Se detecta que la página de ordenar redirige a la de eliminar/extraer páginas. Se corrige el comportamiento.

Casos comunes a todas las operaciones en el Frontend		
Caso de prueba	Resultados esperados	Prueba satisfactoria
Se selecciona un fichero o ficheros (caso de unión) que no es PDF.	Se muestra un mensaje de error indicando que el fichero seleccionado debe de ser un PDF y que ninguno puede superar los X MB, siendo X en principio 10 MB.	Sí
Se selecciona uno o más ficheros de más de X MB de tamaño, siendo X en principio 10 MB.	Se muestra un mensaje de error indicando que el fichero seleccionado debe de ser un PDF y que ninguno puede superar los X MB, siendo X en principio 10 MB.	Sí

Extracción de páginas de un documento		
Caso de prueba	Resultados esperados	Prueba satisfactoria
<i>Frontend:</i> Carga de un documento PDF de tamaño correcto.	Se obtiene su número de páginas a través de una petición HTTP y se muestra una previsualización de todas las páginas, cada una de ellas tiene un checkbox.	Sí

<i>Frontend:</i> Seleccionar una única página desde el componente de texto.	Se marca el checkbox de la página correspondiente.	<b>Sí</b>
<i>Frontend:</i> Borrar la página desde el componente de texto.	Se desmarca el checkbox de la página correspondiente.	Se detecta error de sincronización. Se soluciona
<i>Frontend:</i> Seleccionar una única página desde el checkbox.	Se muestra la página en el componente de texto.	<b>Sí</b>
<i>Frontend:</i> Desmarcar la página desde el checkbox.	Se borra la página del componente de texto.	<b>Sí</b>
<i>Frontend:</i> Se introducen varias páginas consecutivas separadas por comas en el componente de texto. Por ejemplo: 1,2,3,4,5.	Se convierten en un rango (1-5) y se marcan los checkbox correspondientes.	<b>Sí</b>
<i>Frontend:</i> Se introducen varias páginas no consecutivas separadas por comas en el componente de texto. Por ejemplo: 1,3,7.	Se quedan separadas por comas y se marcan los checkbox correspondientes.	<b>Sí</b>
<i>Frontend:</i> Se desmarcan todos los checkbox.	Se borran las páginas del componente de texto.	Se detecta error de sincronización. Se soluciona
<i>Frontend:</i> Se marcan páginas consecutivas en el checkbox: 1 a 5 por ejemplo.	Se convierten en un rango (1-5) en el componente de texto.	<b>Sí</b>
<i>Frontend:</i> Se marcan páginas consecutivas en el checkbox: 1 ,3 y 5 por ejemplo.	Aparecen separadas por comas en el componente de texto.	<b>Sí</b>
<i>Frontend:</i> Se introducen dos comas seguidas en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.	<b>Sí</b>
<i>Frontend:</i> Se introducen dos guiones seguidos en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.	<b>Sí</b>
<i>Frontend:</i> Se introduce una coma y un guion seguidos en el componente de texto.	Cuando el componente pierde el foco se muestra un aviso y se borran.	<b>Sí</b>

<i>Frontend:</i> Se introduce un carácter que no es numérico “,” o “-”.	No se añade al componente de texto.	Se pueden añadir. Se corrige este comportamiento.
<i>Frontend:</i> Se introduce una página o rango de páginas en el componente de texto que no tiene el fichero.	No ocurre nada.	<b>Sí</b>
<i>Frontend:</i> Se seleccionan varios rangos de páginas en los checkbox. Por ejemplo 1-3,5-7.	Se traslada correctamente esta información al componente de texto (1-3,5-7).	<b>Sí</b>
<i>Frontend:</i> Se seleccionan varios rangos de páginas en el componente de texto.	Se traslada correctamente esta información a los checkbox.	<b>Sí</b>
<i>Frontend:</i> Se trata de procesar fichero con el listado de páginas vacío.	Se muestra un mensaje de error indicando que el formato no es válido.	<b>Sí</b>
<i>Procesamiento:</i> Se extrae un listado de páginas separadas por comas 1,3,5,9 por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene 4 páginas y son las esperadas.	Se detecta que las páginas no son las esperadas (Siendo 2,4,5 y 10). Se corrige teniendo en cuenta que si el usuario escoge la página 1 es la 0.
<i>Procesamiento:</i> Se extrae un listado de páginas separadas por guion (rango 1-4) por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene 4 páginas y son las esperadas.	Mismo problema que en el caso anterior. Se corrige.
<i>Procesamiento:</i> Caso límite, se extraen todas las páginas.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas del original.	<b>Sí</b>
<i>Procesamiento:</i> Caso límite, se indica únicamente una página dentro de las seleccionadas que no tiene el fichero.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante es un fichero vacío.	<b>Sí</b>
<i>Procesamiento:</i> Caso límite, se indica una página dentro de las seleccionadas que no tiene el	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante contiene las	<b>Sí</b>

fichero y otras que sí que existen.	páginas que existían en el fichero únicamente.	
-------------------------------------	--	--

<b>Eliminación de páginas de un fichero</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
<i>Frontend:</i> Todos los casos cubiertos en la extracción, ya que comparten componentes.	Revisar casos de prueba <b>Extracción de páginas de un fichero.</b>	Indicado en el apartado anterior.
<i>Procesamiento:</i> Se elimina un listado de páginas separadas por comas 1,3,5,9 por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas menos las que se eliminan.	<b>Sí</b>
<i>Procesamiento:</i> Se elimina un listado de páginas separadas por guion (rango 1-4) por ejemplo. (Caso límite primera página).	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante tiene todas las páginas menos las indicadas.	<b>Sí</b>
<i>Procesamiento:</i> Caso límite. Se eliminan todas las páginas.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante es un fichero vacío.	<b>Sí</b>
<i>Procesamiento:</i> Caso límite. Se indica únicamente una página dentro de las seleccionadas que no tiene el fichero.	Se procesa correctamente, se redirige a la página de descarga. El fichero resultante es un fichero con todas las páginas originales.	<b>Sí</b>
<i>Procesamiento:</i> Caso límite. Se indica una página dentro de las seleccionadas que no tiene el fichero y otras que sí que existen.	Se procesa correctamente, se redirige a la página de descarga. El fichero tendrá todas las páginas originales excepto las que sí existían y se indicaron.	<b>Sí</b>

<b>Unir ficheros</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
<i>Frontend:</i> Seleccionar un único fichero.	Se muestra un mensaje indicando que al menos deben de ser dos ficheros para mezclar/unir.	Inicialmente no se mostraba y se quedaba el fichero

		tal y como estaba. Se añade este comportamiento.
<i>Frontend:</i> Seleccionar más de 5 ficheros.	Se muestra un mensaje indicando que el número máximo de ficheros menos para mezclar/unir es 5.	<b>Sí</b>
<i>Frontend:</i> Se introduce un número correcto de ficheros.	Se muestran una vista previa de los ficheros y se puede variar el orden arrastrando uno sobre el otro.	<b>Sí</b>
<i>Procesamiento:</i> Se mezclan dos ficheros sin cambiar el orden.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene tantas páginas como la suma de los dos y se ha unido en el orden correcto.	<b>Sí</b>
<i>Procesamiento:</i> Se mezclan dos ficheros alterando el orden.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene tantas páginas como la suma de los dos y se ha unido en el orden indicado.	<b>Sí</b>

<b>Ordenar</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
<i>Frontend:</i> Seleccionar un fichero.	Se carga una previsualización de todas las páginas del fichero en el orden inicial, indicando para cada uno el número de página. Se muestran las acciones predefinidas.	Las acciones predefinidas se muestran antes. Se modifica para mostrarlas tan solo cuando hay un fichero.
<i>Frontend:</i> Pulsar el botón de invertir el orden sin realizar ninguna modificación.	Se invierte el orden del documento, pasando a ser la última página la primera y la primera la última.	<b>Sí</b>
<i>Frontend:</i> Pulsar el botón de invertir el orden dos veces consecutivas.	El orden de las páginas es el mismo que el inicial.	<b>Sí</b>

<i>Frontend:</i> Intercambiar el orden de dos páginas consecutivas, 5 y 6 por ejemplo.	La página 5 pasa a ocupar la posición 6 y la página 6 la posición 5.	<b>Sí</b>
<i>Frontend:</i> Mover el orden de dos páginas no consecutivas. Por ejemplo, página 3 a la posición 6.	La página 3 pasa a ocupar la posición 6. Páginas 4,5 y 6 se mueven a una posición inferior.	<b>Sí</b>
<i>Procesamiento:</i> Calcar el botón de ordenar sin ningún tipo de cambio.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene las mismas páginas y en el mismo orden que el inicial.	<b>Sí</b>
<i>Procesamiento:</i> Calcar el botón de ordenar justo tras invertir el orden.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene las mismas páginas que el inicial, pero en el orden inverso.	<b>Sí</b>
<i>Procesamiento:</i> Alterar la posición de algunas páginas manualmente y calcar en el botón ordenar.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene las mismas páginas que el inicial, en el orden indicado por el usuario.	<b>Sí</b>

<b>Recortar</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
<i>Frontend:</i> Seleccionar un fichero.	Se muestra una previsualización de la primera página de un fichero, con una rejilla encima que se puede desplazar y cambiar sus límites.	<b>Sí</b>
<i>Frontend:</i> Mover la rejilla.	La rejilla se traslada de la forma esperada.	<b>Sí</b>
<i>Frontend:</i> Cambiar la posición de cada uno de los ocho puntos de la rejilla.	La rejilla pasa a tener el tamaño esperado.	<b>Sí</b>
<i>Frontend:</i> Aumentar el tamaño de la rejilla al máximo posible.	En ningún momento supera el tamaño del contenedor del documento.	<b>Sí</b>



<i>Procesamiento:</i> Recortar con distintos tamaños de rejilla.	Se procesa correctamente, se redirige a la página de descarga. El documento resultante tiene visible tan solo el contenido de la rejilla.	<b>Sí</b>
<i>Procesamiento:</i> Recortar un fichero rotado.	Se procesa correctamente a pesar de esto, coincidiendo el área seleccionada por el usuario con el área que se recorta.	El área de recorte no coincide con lo seleccionado del usuario. Se soluciona mostrando la imagen de selección rotada.

<b>Dividir</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
<i>Frontend:</i> Seleccionar un fichero.	Se muestra una previsualización de la primera página junto con el nombre del fichero.	<b>Sí</b>
<i>Procesamiento:</i> Dividir el fichero.	Se procesa correctamente y se redirige a descargas. Si descargamos el resultado es un zip, con tantos PDF dentro como páginas tenía el fichero y cada uno de ellos tiene una página.	Inicialmente se procesaba correctamente, pero se descargaba un PDF corrupto. Esto se debía a que el <i>frontend</i> esperaba una respuesta con un BLOB tipo PDF y no un fichero de cualquier tipo. Corrigiendo esto, quedó solucionado el problema.

<b>Historial</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>

Acceder al historial de ficheros sin tener ficheros procesados.	La tabla se muestra vacía y el tamaño total de los ficheros es 0 Bytes.	<b>Sí</b>
Acceder al historial de ficheros teniendo ficheros procesados.	Se muestra una entrada en la tabla para cada fichero, indicando nombre, tamaño y fecha en la que fue añadido. Están ordenados por fecha descendente. El tamaño total es la suma del tamaño de los ficheros.	Para el tamaño se muestra un número entero de bytes sin unidades. Se transforma añadiendo la unidad correspondiente.
Acceder al historial de ficheros teniendo gran número de ficheros procesados.	Se muestran los ficheros en distintas páginas, con la información que se indicó en el apartado anterior.	<b>Sí</b>

<b>Descargar</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
Descargar un fichero desde la página del historial.	El navegador descarga el fichero correctamente.	<b>Sí</b>

<b>Borrar documento</b>		
<b>Caso de prueba</b>	<b>Resultados esperados</b>	<b>Prueba satisfactoria</b>
Seleccionar un documento y borrarlo.	El documento desaparece de la tabla, de la base de datos y del sistema de ficheros.	<b>Sí</b>
Seleccionar más de un documento y borrarlos.	Los documentos desaparecen de la tabla, de la base de datos y del sistema de ficheros.	<b>Sí</b>
Seleccionar todos los documentos de una página y borrarlos cuando solo hay una página.	La tabla queda vacía. Los documentos se borran de la base de datos y del sistema de ficheros.	<b>Sí</b>
Seleccionar todos los documentos de una página y borrarlos cuando hay más de una página.	En la tabla pasa a mostrarse la información de la siguiente página.	<b>Sí</b>

## 8.1.2 Pruebas unitarias automatizadas

En combinación con las anteriores pruebas “manuales”, se han realizado pruebas unitarias automáticas para aquellas funcionalidades que se considera más relevante tener este tipo de pruebas. Se han implementado principalmente para probar los distintos procesamientos de los ficheros.

Estas pruebas se encuentran en el archivo **tests.py** del directorio **pdftools** del backend.

<b>RemovePagesFileProcessorTest</b>		
<b>Nombre método</b>	<b>Descripción</b>	<b>Defectos encontrados</b>
<b>test_remove_one_page</b>	Comprueba que, al eliminar una página, el archivo tiene una menos.	Ninguno.
<b>test_remove_no_page</b>	Comprueba que si no se indican páginas en la <i>request</i> no se elimina ninguna página.	Se producía un error no controlado. Se controla este comportamiento.
<b>test_remove_multiple_pages</b>	Comprueba que al eliminar varias páginas el número disminuye en tantas páginas como las indicadas.	Ninguno.
<b>test_remove_all_pages</b>	Comprueba que al borrar todas las páginas el número de páginas resultantes es 0.	El número resultante era 1 debido a que la página “0” no se estaba teniendo en cuenta. Se corrigió este comportamiento.

<b>ExtractPagesFileProcessorTest</b>		
<b>Nombre método</b>	<b>Descripción</b>	<b>Defectos encontrados</b>
<b>test_extract_one_page</b>	Comprueba que al extraer una única página el fichero resultante tiene efectivamente una página.	Ninguno.
<b>test_extract_no_page</b>	Comprueba que si no se indican páginas en la <i>request</i> no se extrae ninguna página. Resultaría un fichero vacío.	Se detectó que se estaba retornando un fichero corrupto y se

		corrigió este comportamiento.
<b>test_extract_multiple_pages</b>	Comprueba que al extraer varias páginas el número de páginas del fichero es el mismo que el de páginas extraídas.	Ninguno.
<b>test_extract_all_pages</b>	Comprueba que al extraer todas las páginas el fichero resultante tiene las mismas páginas que el inicial.	Ninguno.

#### OrderPagesFileProcessorTest

Nombre método	Descripción	Defectos encontrados
<b>test_order_without_variation</b>	Comprueba que, si el orden de páginas es el mismo que el inicial, el documento no pierde ninguna página.	Ninguno.
<b>test_reverse_order</b>	Comprueba que, si el orden de páginas es el inverso que el inicial, el documento no pierde ninguna página.	Ninguno.
<b>test_change_order</b>	Comprueba que, si el orden de páginas ha sido manipulado manualmente, el documento no pierde ninguna página.	Ninguno.
<b>test_exception_wrong_pagesOrder</b>	Comprueba que, si el listado de páginas a borrar no es completo, se lance una excepción de tipo ValueError.	No se estaba controlando este comportamiento, lo que provocaba un IndexError no controlado al recorrer el array.

#### MergeFilesProcessorTest

Nombre método	Descripción	Defectos encontrados
<b>test_merge_only_one_file</b>	Comprueba que, si se llama al procesamiento de unión con solo un fichero, el resultante tiene tantas páginas como el inicial.	Ninguno.
<b>test_merge_more_than_one_file</b>	Comprueba que, si se unen dos ficheros, el número de páginas resultantes es la suma del número de páginas de ambos ficheros.	Ninguno.
<b>test_merge_same_file_two_times</b>	Comprueba que, si se une dos veces el mismo fichero, junto con otro no existiría ningún problema.	Ninguno.

**CropFileProcessorTest**

Nombre método	Descripción	Defectos encontrados
<b>test_crop_no_change</b>	Comprueba que, si se selecciona todo el tamaño del documento para recortar la <i>mediabox</i> permanece invariable.	Ninguno.
<b>test_crop_change</b>	Comprueba que, si el área de recorte es menor, las transformaciones se aplican correctamente.	Ninguno.

**DivideFileProcessorTest**

Nombre método	Descripción	Defectos encontrados
<b>test_divide_file</b>	Comprueba que, al dividir un fichero se genera un zip con tantos ficheros como páginas tenía el inicial.	Ninguno.

<b>OrderPagesFileProcessorTest</b>		
<b>Nombre método</b>	<b>Descripción</b>	<b>Defectos encontrados</b>
<b>test_empty_string</b>	Comprueba que si la función recibe un string vacío retorna una lista con un único elemento, el 0.	Ninguno.
<b>test_white_list_string</b>	Comprueba que si la función recibe un string de espacios en blanco retorna una lista con un único elemento, el 0.	Ninguno.
<b>test_comma_separated_numbers_list_string</b>	Comprueba que si la función recibe un string de números separados por comas retorna una lista de enteros con los valores recibidos.	Ninguno.
<b>test_hyphen_separated_numbers_list_string</b>	Comprueba que si la función recibe un string de números separados por guiones retorna una lista de enteros con todos los valores que se encuentran entre los dos proporcionados.	Ninguno.
<b>multiple_range_list_string</b>	Comprueba que, si la función recibe un string de rangos separados por comas, retorna una lista de los enteros que se encuentran en alguno de los rangos.	Ninguno.
<b>mixed_range_and_list_string</b>	Comprueba que, si la función recibe un string conformado por todas las combinaciones posibles, retorna una lista de enteros correcta.	Ninguno.

## 8.2 Pruebas de usabilidad

En este apartado se mostrarán los resultados de las pruebas de usabilidad realizadas a los usuarios del sistema. A partir de estos se extraerán conclusiones y se detectarán posibles puntos de mejora.

## 8.2.1 Resultados de los cuestionarios de evaluación

A continuación, se recogen los resultados de los cuestionarios cumplimentados por los usuarios, marcando en **negrita** la opción u opciones seleccionadas.

### 8.2.1.1 Usuario 1

#### 8.2.1.1.1 Cuestionario general

<b>Edad</b>
22
<b>Profesión</b>
Estudiante
<b>¿Con qué frecuencia utiliza un ordenador o dispositivo similar?</b>
<ul style="list-style-type: none"> <li>• <b>Todos los días.</b></li> <li>• Más de una vez por semana.</li> <li>• Raras veces.</li> </ul>
<b>¿Cómo calificaría su nivel informático?</b>
<ul style="list-style-type: none"> <li>• Bajo.</li> <li>• <b>Medio.</b></li> <li>• Alto.</li> </ul>
<b>Cuando lo hace, ¿Para qué emplea un ordenador? (Puede seleccionar varias opciones)</b>
<ul style="list-style-type: none"> <li>• Navegar.</li> <li>• Consultar el correo electrónico.</li> <li>• Ocio (jugar a videojuegos, ver películas o vídeos...).</li> <li>• Es mi herramienta de trabajo principal.</li> <li>• Hago uso de aplicaciones de ofimática.</li> </ul>
<b>¿Sabe qué es un documento PDF?</b>
<ul style="list-style-type: none"> <li>• Sí, los uso en mi día a día.</li> <li>• <b>Sí, y los uso de vez en cuando.</b></li> <li>• Sé lo que es, pero nunca he usado uno.</li> <li>• No.</li> </ul>
<b>¿Alguna vez ha tratado de manipular ficheros PDF?</b>
<ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• No.</li> </ul>
<b>¿De entre estos aspectos, cuál valoraría más en una aplicación?</b>
<ul style="list-style-type: none"> <li>• Que sea rápida.</li> <li>• <b>Que se comprendan rápido todas sus funcionalidades.</b></li> </ul>

- Que tenga gran número de funcionalidades.

#### 8.2.1.1.2 Cuestionario sobre las tareas realizadas

##### ¿Ha conseguido realizar todas las tareas?

- **Sí, con facilidad.**
- Sí, pero con complicaciones.
- No.

##### Si ha tenido complicaciones o no ha podido completarlas indique aquí el motivo

##### ¿Considera que la aplicación es rápida?

- **Sí.**
- Sí al navegar, pero no los tiempos de espera a la hora de procesar documentos.
- No

##### ¿Se ajusta la forma en la que hay que desarrollar cada tarea a lo que pensaba que tendría que hacer?

- Sí, en todas.
- **En la mayoría.**
- Solo en algunas.
- En ninguna.

##### ¿Es capaz de navegar rápidamente por las distintas pantallas?

- **Sí**
- No del todo, pero sí habitualmente.
- No, la navegación no es intuitiva.

##### Conoce en qué página se encuentra

- **En todo momento.**
- Algunas veces.
- Nunca.

##### ¿Le resulta fácil el uso del sistema en conjunto?

- **Sí.**
- No.

##### ¿Considera que los tamaños de letra son correctos?

- **Sí, en toda la aplicación.**
- En algunas pantallas.
- No.

##### ¿Considera que la combinación de colores empleada es correcta?

- **Sí.**



<ul style="list-style-type: none"> <li>No.</li> </ul>
<b>¿Considera el diseño en general atractivo?</b>
<ul style="list-style-type: none"> <li>Sí.</li> <li>No.</li> </ul>
<b>¿Cumple el sistema con su propósito?</b>
<ul style="list-style-type: none"> <li>Sí.</li> <li>No.</li> </ul>
<b>Observaciones. Indique en este cuadro de texto aquellos aspectos que le han resultado extraños, sugerencias de mejora, funcionalidades que considera interesante añadir...</b>
Resulta extraño el proceso de recorte de un fichero. Se toma como referencia la primera página cuando no tiene por qué coincidir con el resto del documento.

### 8.2.1.2 Usuario 2

#### 8.2.1.2.1 Cuestionario general

<b>Edad</b>
33
<b>Profesión</b>
Desarrollador de Software
<b>¿Con qué frecuencia utiliza un ordenador o dispositivo similar?</b>
<ul style="list-style-type: none"> <li>Todos los días.</li> <li>Más de una vez por semana.</li> <li>Raras veces.</li> </ul>
<b>¿Cómo calificaría su nivel informático?</b>
<ul style="list-style-type: none"> <li>Bajo.</li> <li>Medio.</li> <li>Alto.</li> </ul>
<b>Cuando lo hace, ¿Para qué emplea un ordenador? (Puede seleccionar varias opciones)</b>
<ul style="list-style-type: none"> <li>Navegar.</li> <li>Consultar el correo electrónico.</li> <li>Ocio (jugar a videojuegos, ver películas o vídeos...).</li> <li>Es mi herramienta de trabajo principal.</li> <li>Hago uso de aplicaciones de ofimática.</li> </ul>
<b>¿Sabe qué es un documento PDF?</b>
<ul style="list-style-type: none"> <li>Sí, los uso en mi día a día.</li> </ul>

<ul style="list-style-type: none"> <li>• Sí, y los uso de vez en cuando.</li> <li>• Sé lo que es, pero nunca he usado uno.</li> <li>• No.</li> </ul>
<b>¿Alguna vez ha tratado de manipular ficheros PDF?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• No.</li> </ul>
<b>¿De entre estos aspectos, cuál valoraría más en una aplicación?</b>
<ul style="list-style-type: none"> <li>• Que sea rápida.</li> <li>• <b>Que se comprendan rápido todas sus funcionalidades.</b></li> <li>• Que tenga gran número de funcionalidades.</li> </ul>

#### 8.2.1.2.2 Cuestionario sobre las tareas realizadas

<b>¿Ha conseguido realizar todas las tareas?</b>
<ul style="list-style-type: none"> <li>• <b>Sí, con facilidad.</b></li> <li>• Sí, pero con complicaciones.</li> <li>• No.</li> </ul>
<b>Si ha tenido complicaciones o no ha podido completarlas indique aquí el motivo</b>
<b>¿Considera que la aplicación es rápida?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• <b>Sí al navegar, pero no los tiempos de espera a la hora de procesar documentos.</b></li> <li>• No</li> </ul>
<b>¿Se ajusta la forma en la que hay que desarrollar cada tarea a lo que pensaba que tendría que hacer?</b>
<ul style="list-style-type: none"> <li>• <b>Sí, en todas.</b></li> <li>• En la mayoría.</li> <li>• Solo en algunas.</li> <li>• En ninguna.</li> </ul>
<b>¿Es capaz de navegar rápidamente por las distintas pantallas?</b>
<ul style="list-style-type: none"> <li>• <b>Sí</b></li> <li>• No del todo, pero sí habitualmente.</li> <li>• No, la navegación no es intuitiva.</li> </ul>
<b>Conoce en qué página se encuentra</b>
<ul style="list-style-type: none"> <li>• <b>En todo momento.</b></li> <li>• Algunas veces.</li> <li>• Nunca.</li> </ul>

<b>¿Le resulta fácil el uso del sistema en conjunto?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• No.</li> </ul>
<b>¿Considera que los tamaños de letra son correctos?</b>
<ul style="list-style-type: none"> <li>• <b>Sí, en toda la aplicación.</b></li> <li>• En algunas pantallas.</li> <li>• No.</li> </ul>
<b>¿Considera que la combinación de colores empleada es correcta?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• No.</li> </ul>
<b>¿Considera el diseño en general atractivo?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• No.</li> </ul>
<b>¿Cumple el sistema con su propósito?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• No.</li> </ul>
<b>Observaciones. Indique en este cuadro de texto aquellos aspectos que le han resultado extraños, sugerencias de mejora, funcionalidades que considera interesante añadir...</b>
<ul style="list-style-type: none"> <li>• El indicador “?” situado junto al tamaño total de los ficheros parece que aparece por error, cuando se usa para mostrar un tooltip.</li> <li>• Al calcar borrar en el historial se pide confirmación, aunque no haya ficheros seleccionados.</li> <li>• La opción de cierre de sesión está un poco oculta.</li> <li>• Los elementos Acciones e Historial aparecen a pesar de que no se tenga sesión iniciada.</li> <li>• Si calco en “Ordenar” justo después de seleccionar un fichero se produce un error.</li> </ul>

### 8.2.1.3 Usuario 3

#### 8.2.1.3.1 Cuestionario general

<b>Edad</b>
55
<b>Profesión</b>
Profesora
<b>¿Con qué frecuencia utiliza un ordenador o dispositivo similar?</b>
<ul style="list-style-type: none"> <li>• <b>Todos los días.</b></li> <li>• Más de una vez por semana.</li> </ul>

<ul style="list-style-type: none"> <li>• Raras veces.</li> </ul>
<b>¿Cómo calificaría su nivel informático?</b>
<ul style="list-style-type: none"> <li>• Bajo.</li> <li>• <b>Medio.</b></li> <li>• Alto.</li> </ul>
<b>Cuando lo hace, ¿Para qué emplea un ordenador? (Puede seleccionar varias opciones)</b>
<ul style="list-style-type: none"> <li>• <b>Navegar.</b></li> <li>• <b>Consultar el correo electrónico.</b></li> <li>• <b>Ocio (jugar a videojuegos, ver películas o vídeos...).</b></li> <li>• Es mi herramienta de trabajo principal.</li> <li>• Hago uso de aplicaciones de ofimática.</li> </ul>
<b>¿Sabe qué es un documento PDF?</b>
<ul style="list-style-type: none"> <li>• Sí, los uso en mi día a día.</li> <li>• <b>Sí, y los uso de vez en cuando.</b></li> <li>• Sé lo que es, pero nunca he usado uno.</li> <li>• No.</li> </ul>
<b>¿Alguna vez ha tratado de manipular ficheros PDF?</b>
<ul style="list-style-type: none"> <li>• Sí.</li> <li>• <b>No.</b></li> </ul>
<b>¿De entre estos aspectos, cuál valoraría más en una aplicación?</b>
<ul style="list-style-type: none"> <li>• <b>Que sea rápida.</b></li> <li>• Que se comprendan rápido todas sus funcionalidades.</li> <li>• Que tenga gran número de funcionalidades.</li> </ul>

#### 8.2.1.3.2 Cuestionario sobre las tareas realizadas

<b>¿Ha conseguido realizar todas las tareas?</b>
<ul style="list-style-type: none"> <li>• <b>Sí, con facilidad.</b></li> <li>• Sí, pero con complicaciones.</li> <li>• No.</li> </ul>
<b>Si ha tenido complicaciones o no ha podido completarlas indique aquí el motivo</b>
<b>¿Considera que la aplicación es rápida?</b>
<ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• Sí al navegar, pero no los tiempos de espera a la hora de procesar documentos.</li> <li>• No</li> </ul>

<p><b>¿Se ajusta la forma en la que hay que desarrollar cada tarea a lo que pensaba que tendría que hacer?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí, en todas.</b></li> <li>• En la mayoría.</li> <li>• Solo en algunas.</li> <li>• En ninguna.</li> </ul>
<p><b>¿Es capaz de navegar rápidamente por las distintas pantallas?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí</b></li> <li>• No del todo, pero sí habitualmente.</li> <li>• No, la navegación no es intuitiva.</li> </ul>
<p><b>Conoce en qué página se encuentra</b></p> <ul style="list-style-type: none"> <li>• <b>En todo momento.</b></li> <li>• Algunas veces.</li> <li>• Nunca.</li> </ul>
<p><b>¿Le resulta fácil el uso del sistema en conjunto?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• No.</li> </ul>
<p><b>¿Considera que los tamaños de letra son correctos?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí, en toda la aplicación.</b></li> <li>• En algunas pantallas.</li> <li>• No.</li> </ul>
<p><b>¿Considera que la combinación de colores empleada es correcta?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• No.</li> </ul>
<p><b>¿Considera el diseño en general atractivo?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• No.</li> </ul>
<p><b>¿Cumple el sistema con su propósito?</b></p> <ul style="list-style-type: none"> <li>• <b>Sí.</b></li> <li>• No.</li> </ul>
<p><b>Observaciones. Indique en este cuadro de texto aquellos aspectos que le han resultado extraños, sugerencias de mejora, funcionalidades que considera interesante añadir...</b></p> <p>El lugar en que se selecciona el fichero tiene el texto en inglés, cuando el resto de la aplicación está en español.</p>

No parece que se pueda calcar en las filas del historial.

Todos los documentos tienen que estar en la misma carpeta.

## 8.2.2 Análisis de los resultados de los cuestionarios

A pesar de que el número de usuarios encuestados es bajo, a partir de las contestaciones se pueden extraer algunas conclusiones:

- El sistema parece fácil de aprender. Ninguno de los usuarios tuvo problemas relevantes a la hora de completar sus tareas. Tan solo se han detectado algunos comportamientos extraños, que no tienen gran relevancia en la funcionalidad general de la aplicación, y una vez detectados no presentarían problemas en un futuro.
- En cuanto a aspectos como el diseño del interfaz, tamaño de letras y combinación de colores, a los usuarios les parecen correctos. Además, todos indican que saben en la página en la que se encuentran en todo momento, facilitando la navegación por la aplicación. El resumen es que, en general, la satisfacción con el sistema es elevada.
- Uno de los usuarios detecta que la aplicación “no es rápida” en el procesamiento de ficheros. Esto puede deberse a distintos factores, pero en mediciones realizadas se ha detectado que, con una conexión a internet lenta, pueden producirse retrasos vinculados a el tiempo de subida del fichero al servidor, que sería mayor que el tiempo de procesamiento del fichero en sí.

A continuación, se indica como se han tratado las distintas observaciones indicadas por los usuarios:

- **Resulta extraño el proceso de recorte de un fichero.** Se considera que es cierto y que no es del todo intuitivo y usable, ya que puede parecer que se recorta únicamente una página o cometer errores en el recorte. Se tratará este tema como una posible ampliación, indicando el número de página que se desea tomar como referencia.
- **El indicador “?” situado junto al tamaño total de los ficheros parece que aparece por error, cuando se usa para mostrar un tooltip.** Como es un cambio simple y se piensa que puede parecer lo que se indica en el comentario, se decide modificar el icono, cambiándolo por un icono de información, quedando esto tal y como se indica:

**Tamaño total: 24.5 MB (?)**

**Figura 8.1 Indicador de tamaño inicial**

**Tamaño total: 24.5 MB** ⓘ

**Figura 8.2 Indicador de tamaño tras el cambio**

- **Al calcar borrar en el historial se pide confirmación, aunque no haya ficheros seleccionados:** Si bien es cierto que esto ocurre, creo que es algo inocuo, ya que no afecta en nada más que tener que cerrar un diálogo de confirmación y realmente no se borra nada, por lo que se deja tal y como está.
- **La opción de cierre de sesión está un poco oculta:** No resulta un problema relevante, ya que si el usuario no cierra sesión manualmente se cerrará automáticamente con el paso del tiempo, cuando caduque el token JWT vinculado a la sesión.
- **Los elementos Acciones e Historial aparecen a pesar de que no se tenga sesión iniciada:** Se están mostrando de forma consciente para que un usuario que llegue al sistema por primera vez tenga una visión de las acciones que puede realizar, sin necesidad de estar registrado. Además, se controla que, si se intenta acceder a una de ellas, sí que se tenga que iniciar sesión.
- **Si calco en “ordenar” justo después de seleccionar un fichero se produce un error.** Se decide controlar este comportamiento, manteniendo el botón de ordenar desactivado hasta que se valide el fichero y se haya cargado el orden inicial de las páginas.
- **El lugar en que se selecciona el fichero tiene el texto en inglés, cuando el resto de la aplicación está en español:** Aparece *Browse* en lugar de *Explorar* o algo similar en el selector de ficheros. De nuevo se considera que no tiene mayor importancia y se deja tal y como está.
- **No parece que se pueda calcar en las filas del historial.** Considero que sí lo parece, ya que al pasar por encima el ratón aparece el puntero, de tal forma que da a entender que se puede calcar. Además, una vez detectado que se puede realizar no traería ningún problema.
- **Todos los documentos tienen que estar en la misma carpeta.** Esto tiene que ver con el selector de ficheros del sistema operativo más que con la aplicación en sí, que obliga a que todos los ficheros estén en el mismo directorio. No obstante, se registrará como una posible ampliación una alternativa para añadir los documentos en distintas subidas, permitiendo así la carga de elementos de distintas carpetas.

## Capítulo 9. Manuales del Sistema

En este capítulo se recogen manuales que podrían ser interesantes para comprender el sistema desde distintos puntos de vista, ya sea para ejecutarlo en local, realizar ampliaciones sobre él o simplemente usarlo.

### 9.1 Manual de Instalación

Al tratarse de una aplicación desplegada en un servidor y accesible desde cualquier navegador, no es necesario ningún tipo de instalación. No obstante, por si se desease por algún motivo emplear la aplicación en un entorno local, en el *Manual de Ejecución y Despliegue* se indicará como poner este entorno en funcionamiento.

Para usar la aplicación, únicamente habrá que acceder a la siguiente dirección: <https://pdf-toolbox.herokuapp.com/>

### 9.2 Manual de Ejecución y Despliegue

Este manual recogerá los pasos necesarios para ejecutar los subsistemas en un entorno local. Además, se documentarán los pasos que se siguieron para desplegar estos en un servidor de Heroku. Se parte de la base de que ya se dispone del código.

#### 9.2.1 Instalación de Python y Node.js

Lo primero que habría que hacer es instalar, si no se tiene instalado, tanto Python como Node.js. La aplicación está desarrollada con la versión 3.8.1 de Python, por lo que sería interesante instalar esta. De Node se puede instalar la 12.5.0, cercana a la versión sobre la que se hizo el desarrollo.

Desde <https://python.org/downloads/> se puede buscar por versión y descargar. Existen muchas opciones para descargarla. En este manual se escoge **Windows x86-64 executable installer**.

En el instalador se seleccionará **Install Now**, marcando anteriormente la opción **Add Python 3.8 to PATH**. En principio no sería necesario ningún cambio más, dejando que se ejecute el instalador hasta el final.

En el caso de Node.js se puede acceder a las Descargas en <https://nodejs.org/es/download/releases>, seleccionando la versión deseada y dentro de ella uno de los instaladores. En este ejemplo para tener el instalador de Windows la opción **node-v12-5-0-x64-msi** para un equipo de 64 bits.

Se ejecuta el instalador descargado, dejando en principio todas las opciones por defecto y esperando a que finalice el instalador.



## 9.2.2 Puesta en marcha del backend

Con una terminal, PowerShell, por ejemplo, nos situamos en la raíz del proyecto, donde se encuentra el fichero `manage.py`.

Si hubiese en este directorio una carpeta **venv** se recomienda borrarla y generar una nueva, siempre y cuando sea la primera puesta en marcha. Una vez que se tenga montada en local la estructura de directorios no se deberá borrar más.

Una vez borrada o si no estuviese presente, se creará un entorno virtual, con el comando **python -m venv venv**.

Esto generará la carpeta `venv`.

Entramos en esta carpeta y accedemos a sus Scripts con:

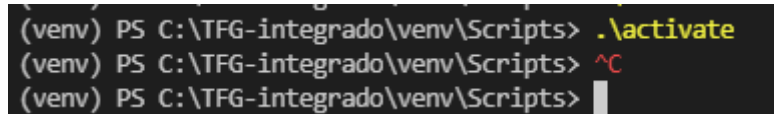
```
cd .\venv\Scripts\
```

Ejecutamos el script de activación:

```
.\activate
```

En este momento puede ser que se indique que en el sistema esta deshabilitada la ejecución de scripts. En la web [49] se indica como resolver este problema.

Una vez que se ejecute correctamente el comando aparecerá en la consola antes de la ruta (**venv**), tal que así



```
(venv) PS C:\TFG-integrado\venv\Scripts> .\activate
(venv) PS C:\TFG-integrado\venv\Scripts> ^C
(venv) PS C:\TFG-integrado\venv\Scripts> |
```

Figura 9.1 Muestra de activación del entorno virtual `venv`

Antes de continuar, habría que actualizar la versión de `pip`, ya que, si no, los siguientes pasos podrían fallar. Para ello, se debe ejecutar **python -m pip install --upgrade pip**.

En este momento ya se pueden instalar los paquetes de Python. Para ello ejecutamos, **desde la raíz del proyecto**:

```
pip install -r requirements.txt.
```

Este fichero **requirements.txt** indica los paquetes que se deben instalar para el correcto funcionamiento del sistema.

Cuando estén instalados ya se podrá arrancar el servidor Django, pero antes habrá que adaptar la configuración para que el sistema pueda funcionar en local.

Para ello, en el fichero **settings.py** se debería dejar tal que así la configuración de base de datos, teniendo sin comentar únicamente la opción de `sqlite`:

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

#DATABASES={
#    'default':{
#        'ENGINE': 'django.db.backends.postgresql_psycopg2',
#        'NAME': 'pdfertools',
#        'USER': 'postgres',
#        'PASSWORD': '',
#        'HOST': '127.0.0.1',
#        'PORT': '5432',
#    }
#}

#DATABASES = {
#    'default': dj_database_url.config(
#        default=config('DATABASE_URL')
#    )
#}

```

Figura 9.2 Distintas configuraciones de base de datos

De esta forma, se empleará la base de datos sqlite en local. Si se desea emplear una PostgreSQL, se dejaría sin comentar la segunda opción y se rellenarían los campos con los datos de conexión de esta. No obstante, implicaría tener que instalar dicha base de datos, por lo que para realizar una primera prueba de ejecución se recomienda la opción mostrada en la imagen.

Antes de arrancar el servidor, para asegurarse de que la base de datos está actualizada, se deberían ejecutar los siguientes comandos (en el directorio en que se encuentra manage.py):

1. **python .\manage.py makemigrations**
2. **python .\manage.py migrate**

En teoría con esto todo estaría preparado. Con el comando **python manage.py runserver** se arrancaría el servidor de Django y nos indicará algo tal que así:

```

System check identified no issues (0 silenced).
June 03, 2021 - 17:51:56
Django version 3.1.4, using settings 'tfg.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Figura 9.3 Indicador de servidor Django arrancado

Si accedemos a esa URL en realidad no se encontrará nada porque está preparado para no mostrar información por configuración. Se mostrará una página tal que así:

# Not Found

The requested resource was not found on this server.

## Figura 9.4 Página de error del servidor

Para comprobar el funcionamiento habrá que instalar el *frontend* y usar todo el sistema desde la interfaz que proporciona este.

Se recomienda no cerrar esta terminal para poder probar todo el sistema en conjunto.

Por último, hay que mencionar que siempre que se desee poner en marcha el sistema, habrá que tener activado el entorno virtual, que se activa tal y como se explicó anteriormente en este mismo apartado.

### 9.2.3 Puesta en marcha del frontend

Con una terminal, PowerShell, por ejemplo, nos situamos en la raíz del proyecto, donde se encuentra el fichero **package.json**.

Si hubiese una carpeta `node_modules` y fuese la primera puesta en marcha se debería borrar para proseguir.

Para instalar todas las dependencias, desde esta ubicación ejecutamos el comando **npm install**. Puede tardar un tiempo en ejecutarse.

Al igual que el *backend*, hay que configurarlo, si aún no lo estuviese, para que funcione en local. En este caso habrá que hacer los siguientes cambios.

Asegurarse que en el fichero **store.js** la configuración está tal que así, asociando el valor `URL_BASE_LOCAL` en **obtainJWT** y **refreshJWT**.

```

Vue.use(Vuex, axios)
const URL_BASE_LOCAL = 'http://localhost:8000/'
const URL_BASE_PROD = 'https://tfg-pdftools-backend.herokuapp.com/'

// Manejo de estado temporal para el login
export default new Vuex.Store({
  state: {
    jwt: localStorage.getItem('t') || '',
    endPoints: {
      obtainJWT: URL_BASE_LOCAL + 'auth/obtain_token/',
      refreshJWT: URL_BASE_LOCAL + '/auth/refresh_token/'
    },
    authStatus: ''
  }
})

```

Figura 9.5 Configuración de URL de autenticación

De forma similar, en el fichero `main.js`, `axios.default.baseURL` tiene que tener asociada la ruta local, como se muestra en la siguiente imagen.

```

Vue.config.productionTip = false

// axios config
axios.defaults.baseURL = 'http://localhost:8000/api/v1.0/'
// axios.defaults.baseURL = 'https://tfg-pdftools-backend.herokuapp.com/api/v1.0/'
axios.defaults.xsrfHeaderName = 'X-CSRFToken'
axios.defaults.xsrfCookieName = 'csrftoken'

```

Figura 9.6 Configuración de URL genérica

Tras esto, desde la raíz de la aplicación se ejecuta `npm run dev`.

Cuando acabe de ejecutarse este comando, si no hemos cerrado la terminal del *backend*, tendríamos desplegados en local ambos subsistemas, y se podría utilizar la aplicación accediendo a la dirección indicada:

```

DONE Compiled successfully in 43039ms

I Your application is running here: http://localhost:8000

```

Figura 9.7 Indicador servidor Frontend

## 9.2.4 Despliegue de la aplicación

Ambos subsistemas han sido desplegados en Heroku de una forma simple empleando su dashboard.

Para el *frontend*, simplemente se ha creado una nueva app de Heroku en la región europea:

**Figura 9.8 Creación de nueva app (servidor) en Heroku**

Una vez creada, se ha elegido el método de despliegue integrado con GitHub, y se ha enlazado con el repositorio en el que se encuentra el código:

**Figura 9.9 Configuraciones de despliegue de Heroku**

Por último, se han activado los despliegues automáticos, para que cada vez que se suba código a Github, se haga un nuevo despliegue.

Para el *backend*, se ha realizado un proceso muy similar, pero algo más complejo. Se han seguido los pasos indicados en el tutorial [50].

## 9.3 Manual de Usuario

En este manual se recogerán las funcionalidades del sistema de una forma básica, de tal forma que un usuario que lo lea sea capaz de ejecutarlas todas sin mayores problemas.

### 9.3.1 Barra de navegación

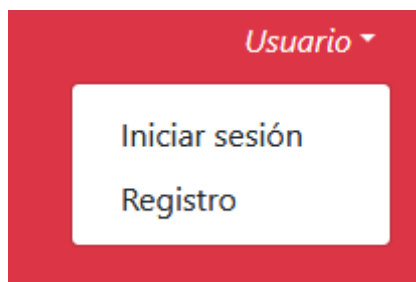
La barra de navegación es un componente importante que permitirá moverse por la aplicación.



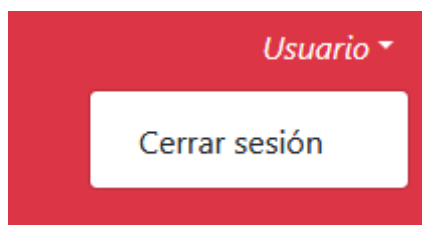
**Figura 9.10 Barra de navegación de la aplicación**

Sus contenidos son los siguientes:

- **Enlace con el texto PDFTools:** Permite acceder a la página de inicio del sistema.
- **Acciones:** Despliega un menú con enlaces a las acciones que puede realizar un usuario sobre ficheros PDF. Si el usuario no está en sesión, estos enlaces redirigirán a la pantalla de inicio de sesión.
- **Historial:** Permite acceder al historial de ficheros generados. Si el usuario no está en sesión redirigirá a la pantalla de inicio de sesión.
- **Usuario:** Contiene un menú desplegable con acciones dependientes de si el usuario está o no con sesión iniciada.



**Figura 9.11 Opciones de un usuario sin sesión iniciada**



**Figura 9.12 Opciones de usuario con sesión iniciada**

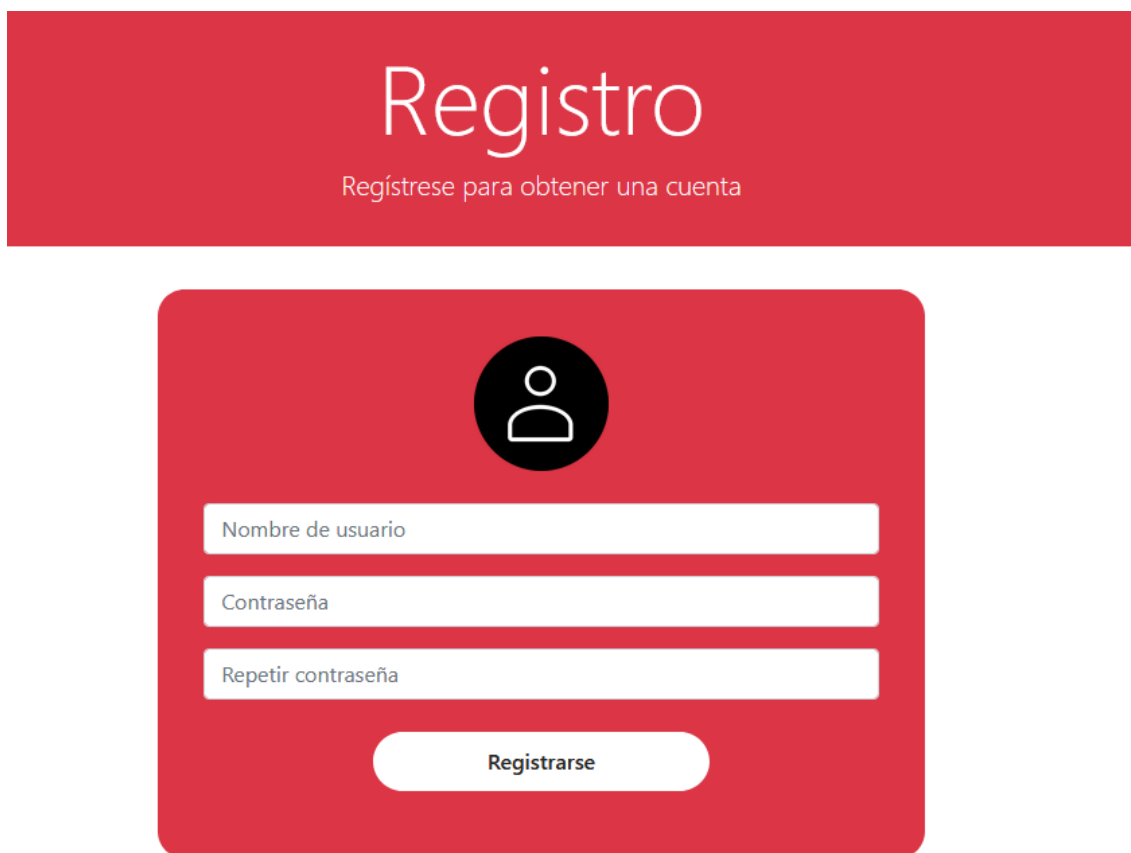
### 9.3.2 Registro

En esta pantalla el usuario podrá registrarse en el sistema. Para ello deberá proporcionar un nombre de usuario y una contraseña, repitiendo esta última para su confirmación.

Los valores del formulario tienen las siguientes restricciones:

- No puede haber ningún campo vacío.

- El nombre de usuario no puede estar en uso en el sistema.
- La contraseña debe tener al menos 8 caracteres.
- Las contraseñas deben coincidir.



Registro

Regístrese para obtener una cuenta

Nombre de usuario

Contraseña

Repetir contraseña

Registrarse

The image shows a registration form on a red background. At the top, the word 'Registro' is written in large white font, with the subtitle 'Regístrese para obtener una cuenta' below it. In the center, there is a white rounded rectangle containing a black circular icon of a person. Below the icon are three white input fields with labels: 'Nombre de usuario', 'Contraseña', and 'Repetir contraseña'. At the bottom of the white rectangle is a white rounded button with the text 'Registrarse'.

**Figura 9.13 Campos del componente de registro de usuario**

En caso de que los datos se rellenen de forma satisfactoria se redirigirá al usuario a la pantalla de inicio de sesión.

### 9.3.3 Inicio de sesión


A través de esta pantalla es como el usuario podrá iniciar una sesión en el sistema, requisito indispensable para consultar su historial de acciones y poder manipular ficheros.

Únicamente tendrá que introducir su nombre de usuario y contraseña de forma correcta. Si la combinación no es correcta, se le mostrará un aviso indicándolo. Si fuese correcta se le redirigirá a la página de inicio.

Además, desde la barra de navegación podrá acceder desde aquí a la creación de una nueva cuenta.

# Iniciar sesión

Inicie sesión para empezar a usar la aplicación



**Figura 9.14 Campos del componente de inicio de sesión**

### 9.3.4 Página de inicio

Esta página permite al usuario acceder a todas las opciones de procesamiento disponibles.





**Figura 9.15 Opciones de la página de inicio**

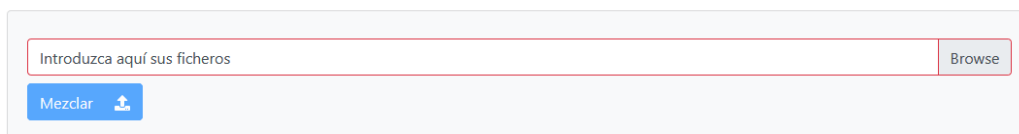
Cada una de las opciones que se muestran en la imagen anterior tiene un enlace a la página correspondiente.

### 9.3.5 Mezclar/Unir documentos

En esta página el usuario podrá unir múltiples documentos en un único fichero. Se han establecido limitaciones para evitar sobrecargar el sistema de ficheros. Estas limitaciones son:

- Se pueden unir entre 2 y 5 ficheros.
- El tamaño de cada fichero no podrá superar los 10MB. (Esta limitación afecta a todas las pantallas).

Para unir ficheros el usuario tendrá que hacer click en el componente de selección de ficheros:



**Figura 9.16 Componente selector de ficheros**

Mientras se muestre con el borde en rojo, no se podrá realizar ninguna operación. Si los ficheros son PDF válidos se pondrá este borde de color verde.

Se abrirá un selector de ficheros para seleccionar elementos que tenga el usuario en su equipo:

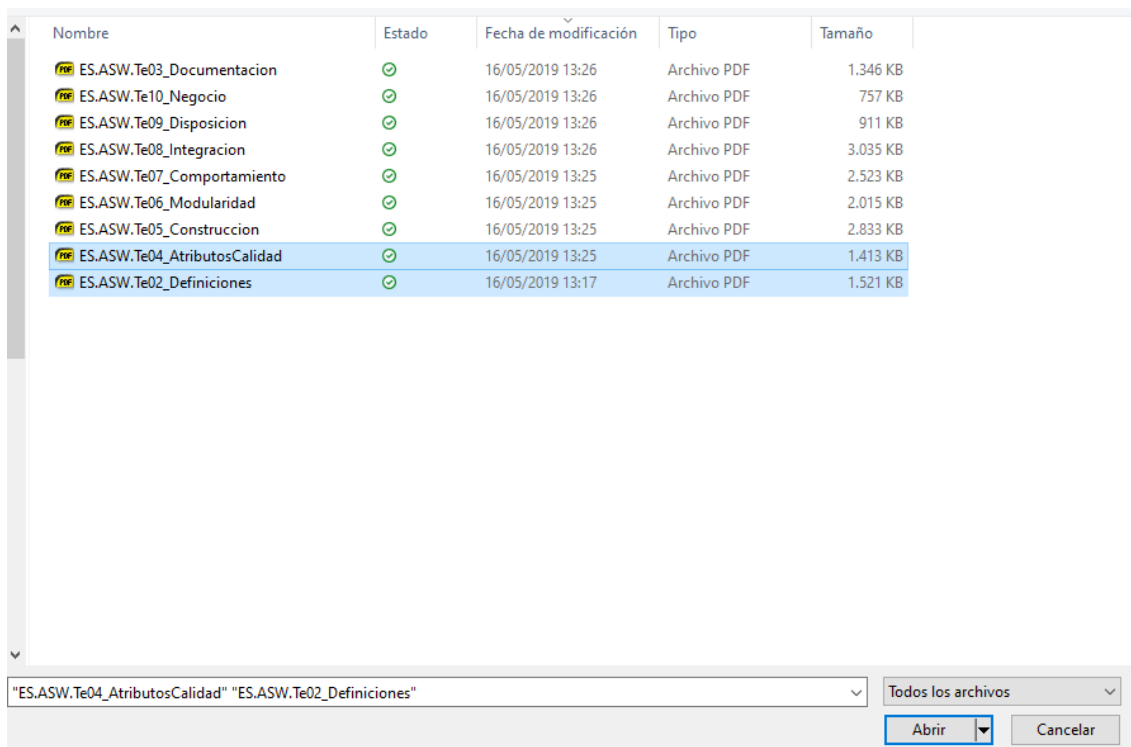


Figura 9.17 Selector de ficheros del sistema operativo

Una vez seleccionados, se mostrará en la pantalla, además del borde verde alrededor del componente de selección indicando que se pueden unir esos documentos, tarjetas de previsualización de cada uno de los ficheros, mostrando la primera página y el nombre del documento.



Figura 9.18 Página de unión de PDF una vez seleccionados archivos válidos

Para modificar el orden, basta con calcar sobre la previsualización de uno de los documentos y arrastrarlo a la posición que se desea que ocupe.

Una vez que se tenga el orden deseado se puede hacer click en el botón Mezclar. Este botón permanecerá desactivado hasta el momento en que se hayan seleccionado ficheros válidos.

Cuando se inicia la acción, se mostrará una capa de espera sobre la página:



**Figura 9.19** Capa de espera

Al acabar el proceso, la aplicación redirigirá al usuario a la pantalla de descarga, desde la que podrá obtener el fichero generado a través del botón “Descargar el resultado”:



**Figura 9.20** Componente de descarga de ficheros

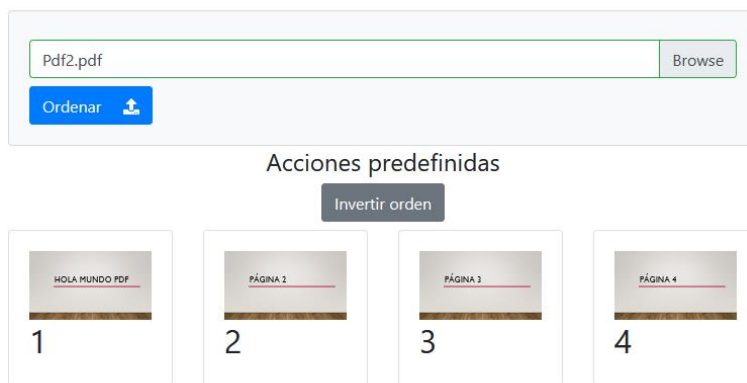
Esta pantalla será común a todos los procesos.

### 9.3.6 Ordenar

Los primeros pasos que hay que hacer son prácticamente idénticos a los realizados en la unión de documentos. Al principio, aparecerá únicamente un selector (en este caso permite un único fichero).

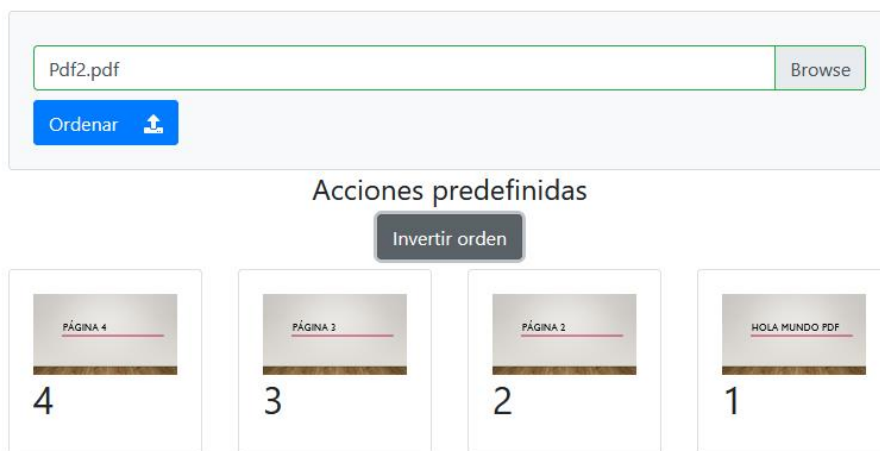
De nuevo, mientras no haya documento o este no sea válido el borde del selector será rojo y el botón de acción estará desactivado.

Una vez subido un documento, se obtendrá el número de páginas de este y se mostrarán en orden, además de una acción predefinida que permitirá invertir el orden del documento:



**Figura 9.21 Presentación de las páginas una vez seleccionado un fichero**

Si empleamos la opción Invertir orden, la última página pasa a ser la primera y la primera la última. A continuación, se muestra la pantalla tras haber invertido el orden:



**Figura 9.22 Presentación de las páginas al invertirlas**

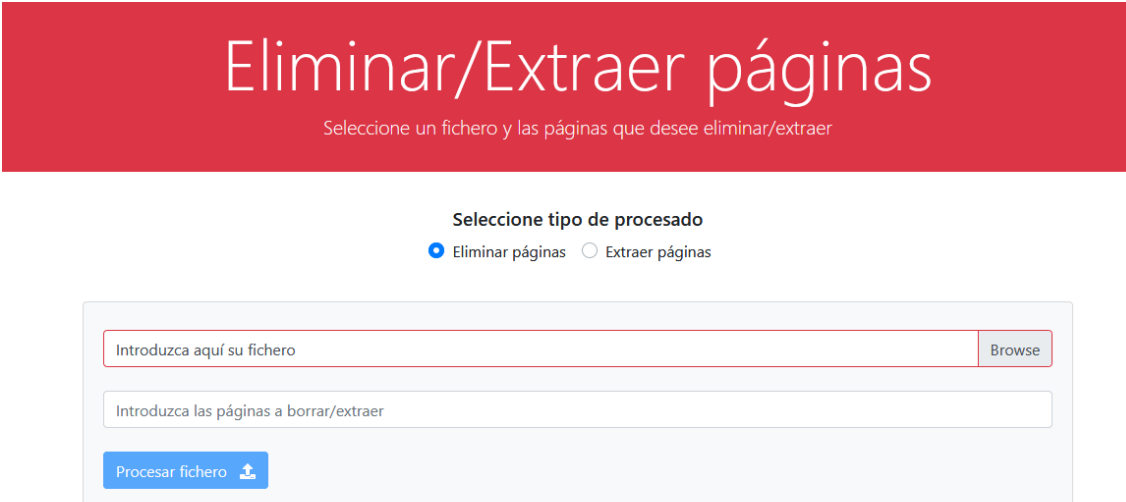
Para que se puedan seguir haciendo cambios, este invertir orden no llama al procesamiento del fichero directamente, para ello habría que calcar en el botón ordenar.

Si lo que se desea es alterar manualmente el orden de una página el usuario tan solo tendrá que arrastrar esa página a la posición que quiere que ocupe.

Una vez que el orden sea el que se quería conseguir, calcando el botón ordenar se iniciará el proceso, siendo lo que ocurre análogo a lo que sucedía al iniciar el proceso de unión de ficheros. Puede consultarse en la parte final de ese apartado.

### 9.3.7 Eliminar extraer páginas

Para esta acción aparece inicialmente más información en pantalla que para las anteriores:



Eliminar/Extraer páginas


Seleccione un fichero y las páginas que desee eliminar/extraer

Seleccione tipo de procesado

Eliminar páginas  Extraer páginas

Introduzca aquí su fichero Browse

Introduzca las páginas a borrar/extraer

Procesar fichero 

**Figura 9.23 Aspecto inicial de la página de eliminación/extracción de páginas**

En el radio botón, se podrá elegir un tipo de procesado que se desea:

- **Eliminar páginas:** Si selecciona esta opción el PDF resultante tendrá todas las páginas que tenía inicialmente exceptuando las indicadas.
- **Extraer páginas:** Si selecciona esta opción el PDF resultante tendrá únicamente las páginas seleccionadas.

Además, existe un componente de texto para seleccionar las páginas que se deseen extraer/eliminar. Este texto deberá rellenarse una vez que se haya seleccionado un fichero, ya que se reinicia cada vez que se indica uno nuevo.

Si se selecciona un PDF la apariencia es la siguiente:


**Seleccione tipo de procesado**

Eliminar páginas     Extraer páginas


Pdf2.pdf Browse

Introduzca las páginas a borrar/extraer


Procesar fichero ⬆




1



2



3



4

**Figura 9.24 Aspecto de la página de eliminación/extracción de páginas una vez seleccionado un fichero**

La página presenta dos formas de seleccionar las páginas:

- A través del componente de texto que se indicó anteriormente.
- A partir de un checkbox que tiene cada página y que al marcarlo hará que quede seleccionada.

Estas dos formas de selección están sincronizadas. Si se añade una página o grupo de páginas a través del componente de texto se marcarán los checkbox y viceversa.

El componente de texto permitirá definir las páginas que procesar de la siguiente forma:

- **Páginas separadas por comas:** 1,3,5,10...
- **Grupos de páginas consecutivas:** 1-6 (Selecciona todas las páginas de la 1 a la 6).
- **Varios grupos de páginas consecutivas:** 1-6,10-20 (Selecciona todas las páginas de la 1 a la 6 y todas las que se encuentren entre la 10 y la 20).
- **Todo lo anterior unido:** 1,3-6,8,12-16.

Al seleccionar las páginas desde los checkbox, los rangos se generarán automáticamente en el componente de texto.

Una vez que se tenga seleccionado el tipo de procesamiento (extraer/eliminar), las páginas y se inicie el proceso, el comportamiento será el mismo que en la unión de ficheros. Se puede consultar en la parte final de dicho apartado.

### 9.3.8 Recortar

La apariencia inicial será muy similar a las de las otras páginas de procesamiento, permitiendo de primeras únicamente seleccionar un fichero.

Una vez subido el archivo, se mostrará una previsualización de la primera página con una rejilla superpuesta, que será la que permita definir el área de recorte:



**Figura 9.25** Componente de recorte de una página

El usuario podrá tanto desplazar la rejilla como aumentar o disminuir su tamaño desde cualquiera de sus bordes. Todas las páginas pasarán a tener el área definida sobre esta previsualización.

De nuevo, una vez que se define el área de recorte y se inicia el proceso, el comportamiento será el mismo que en la unión de ficheros, pudiendo consultarse en este apartado cómo se descarga el fichero generado.

### 9.3.9 Dividir

Desde esta página el usuario podrá dividir su documento en tantos PDFs como páginas tenga el original.

Es una de las páginas más simples y su apariencia es prácticamente igual que la de la pantalla de recorte de documentos, con la diferencia de que la imagen de previsualización es estática, y se muestra para que el usuario pueda comprobar que el fichero es efectivamente el que se deseaba seleccionar. Al seleccionar un documento la página queda tal que así:



**Figura 9.26** Página de división de ficheros una vez seleccionado un documento

Al calcar en el botón de “Dividir” comenzará el procesamiento. Cuando termina, la aplicación redirecciona a la página de descarga. El fichero resultante será un zip cuyo contenido son tantos PDF como páginas tenía el original.

### 9.3.10 Historial

En esta página se encontrará un registro de los ficheros que ha generado el usuario con el uso del sistema, como se ve a continuación:



**Figura 9.27** Historial de ficheros



Un usuario con gran número de ficheros tendrá más de una página de resultados, pudiendo avanzar entre ellos haciendo uso de la paginación que se encuentra en la parte inferior de la tabla.

La tabla no es únicamente de consulta. Sobre ella se pueden realizar una serie de acciones.

Haciendo doble click sobre un fichero este se descargará. Por tanto, se tendrá acceso a los documentos no solo en el momento en que se generan, también se podrán conseguir en cualquier momento desde este historial.



**Figura 9.28 Descarga de un fichero almacenado**

Se pueden seleccionar documentos para borrar. Esto puede ser interesante una vez que ya no se necesiten, ya que existe una limitación para que un usuario, una vez superados los 100MB de almacenamiento, no pueda realizar más acciones.

Para ello, habrá que seleccionarlos sobre la tabla y calcar en el botón de borrado. Se solicitará confirmación antes de proceder con este.

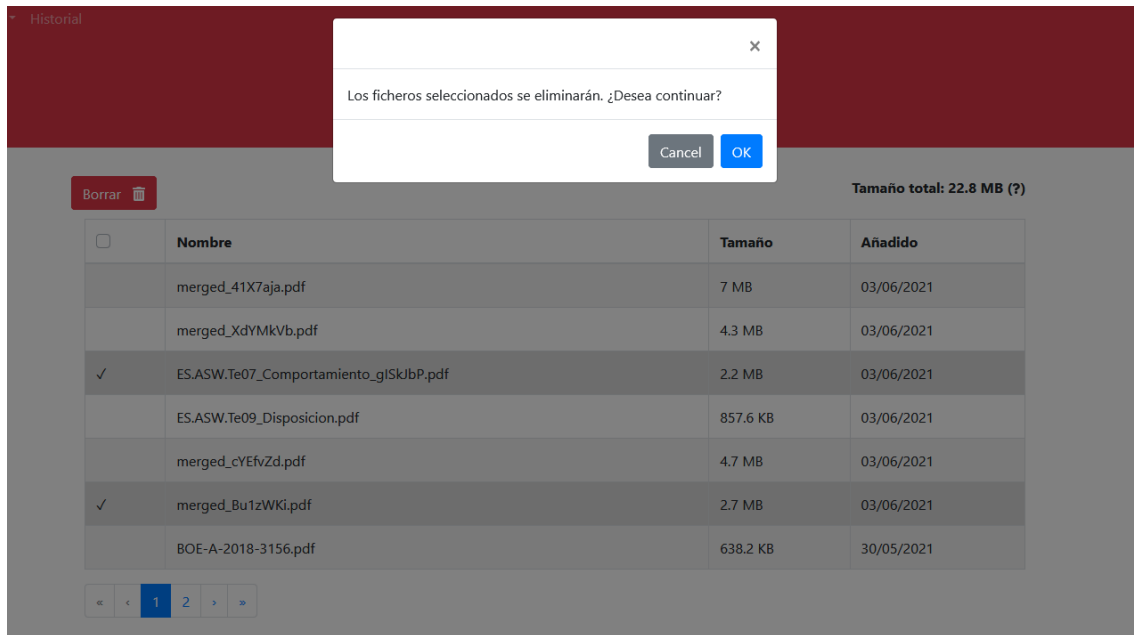


Figura 9.29 Confirmación de la eliminación de ficheros

Para evitar borrados no deseados se podrán marcar a la vez solo ficheros de la página en la que se encuentre actualmente el usuario, de tal forma que, en ningún momento pierda de vista qué elementos son los que tiene seleccionados.

Por último, existe la opción de seleccionar/deseleccionar todos los elementos de una página para su borrado, a través del checkbox situado en la primera columna de la cabecera de la tabla:

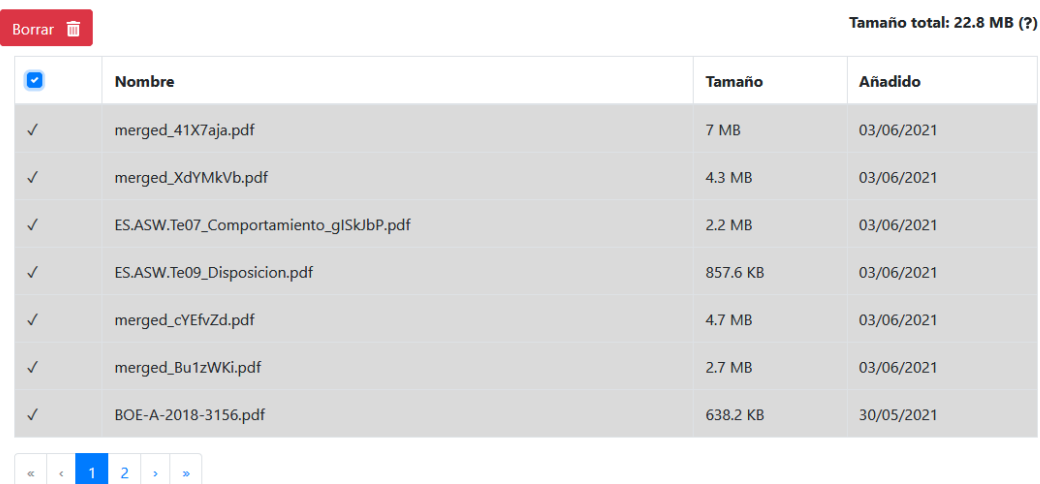


Figura 9.30 Selección de todos los elementos de una página

## 9.4 Manual del Programador

En este apartado se recogerá información que puede ser de utilidad para una persona que desee continuar desarrollando el sistema para añadir nuevas funcionalidades o simplemente conocer cómo funciona. Se detallarán las librerías y *framework* más relevantes, dónde encontrar información más detallada sobre estas y cuál es la estructura de carpetas del proyecto. Se presenta además un ejemplo del flujo necesario para añadir, por ejemplo, una nueva funcionalidad de manipulación.

En este manual se asume que se tiene a disposición el código fuente de la aplicación y que se tiene el conocimiento para arrancar el sistema en un entorno local. Esta información se recoge en el *Manual de Ejecución y Despliegue*.

### 9.4.1 Herramientas de gestión de dependencias/paquetes

Para gestionar las dependencias del proyecto se han de usar herramientas de este tipo. Como en el caso de este sistema existen dos sistemas diferenciados, cada uno empleará uno distinto. Son estos:

- **PIP:** Es el gestor de paquetes más conocido de Python. De hecho, ya viene instalado con la versión empleada para el desarrollo de la aplicación. Permite la instalación, desinstalación, listado de paquetes existentes y otras funciones. Una explicación básica puede encontrarse en [51]. Para obtener información más detallada se puede consultar [52].
- **npm:** Se trata del gestor de paquetes que viene por defecto con Node.js [53] y el más conocido junto con Yarn. Proporciona las mismas funcionalidades que PIP y su documentación se encuentra en [54].

### 9.4.2 Librerías y frameworks más importantes

Obviamente, en el proyecto se han utilizado múltiples librerías, algunas para solventar un problema concreto y otras que tienen importancia a nivel global. Estas se detallan a continuación:

- **DRF (Django Rest Framework):** Se trata de un *framework* construido sobre Django, que permite la creación de APIs REST de forma rápida y flexible. Sigue un patrón muy similar al MVC, en el que se distinguen claramente Viewsets (similares a controladores), Serializers (encargados de realizar transformaciones de datos para el guardado y retorno de respuestas) y Models (que definen la estructura de los elementos en la base de datos). Información detallada, guías de iniciación y ejemplos se pueden encontrar en [26].
- **PyPDF2:** Es una librería de Python que permite la manipulación de ficheros PDF y, por tanto, la base empleada para cualquier procesamiento que se ha realizado en la aplicación. La base son las siguientes clases:
  - **PdfFileReader:** Permite obtener la información de un fichero.

- **PdfFileMerger:** Permite unir documentos.
- **PageObject:** Representa una página de un PDF.
- **PdfFileWriter:** Permite la creación de un nuevo documento.

En [55] existe una documentación simple pero muy clara.

- **django-storages[azure]:** Se trata de la herramienta empleada para configurar la aplicación de tal forma que emplee como sistema de ficheros un contenedor de documentos blob de Azure [28]. Permite realizar este enlace de una forma simple, manipulando (en su forma más básica) únicamente documentos de configuración. Se puede encontrar información en [56].
- **Vue.js:** Se trata de un *framework* de Javascript orientado a componentes, similar a Angular y React en los objetivos, para la creación de aplicaciones SPA. Es por tanto la base de cómo se ha estructurado el *frontend* en la aplicación. En [9] se pueden obtener guías de iniciación, de estilos, información sobre el proyecto...
- **Vuex:** Se trata tanto de un patrón de gestión de estados como una librería que lo aplica. Está pensado para trabajar en aplicaciones realizadas con Vue.js y permite tener un punto de almacenamiento centralizado y accesible desde toda la aplicación. Esta información y otra como guías de uso y conceptos principales se puede obtener en [19]. Se emplea en el sistema para gestionar la sesión de los usuarios.
- **Vue Router:** Es otro elemento del ecosistema de librerías de Vue. Su función principal es la gestión de rutas de la SPA para simular la navegación entre distintas páginas. Además de gestionar la navegación de una página a otra, aporta funcionalidades como los *guards*. Estos permiten definir funciones que se ejecutan antes o después de acceder a una página concreta. Sus funcionalidades, guías de inicio y demás se encuentran en [27].
- **Axios:** Se trata de una librería de JavaScript para realizar peticiones HTTP de forma simple. Permite, además, configurar elementos por defecto como cabeceras, cookies, una URL base para todas las peticiones que solo haya que extender en cada caso concreto... La comunicación entre *frontend* y *backend* de la aplicación se hace a través de esta librería. En su página de Github [21] puede encontrarse toda la información necesaria.
- **BootstrapVue:** Se trata de una integración de la conocida librería Bootstrap en Vue. Aporta un gran número de componentes con funcionalidad, ahorrándole al desarrollador la creación de estos. La documentación de todos los componentes y el proyecto en sí se encuentra en [13].

### 9.4.3 Estructura de directorios del proyecto

En este apartado se comentará brevemente la forma de distribuir los archivos en las carpetas del proyecto para ambos subsistemas.

#### 9.4.3.1 Estructura de directorios del frontend

Partiendo de la raíz del proyecto se encuentran los siguientes ficheros o directorios:

- **build:** En esta carpeta se encuentran los ficheros que utiliza **webpack** [57] para empaquetar la aplicación. En principio, no debería ser necesario manipularlos de forma manual.
- **config:** Ficheros de configuración del entorno. Establecen las configuraciones para los entornos de desarrollo y producción.
- **dist:** Aquí se almacenan los ficheros de despliegue que se generan al construir la aplicación con el comando `npm run build`.
- **docs:** Contiene la documentación de los métodos de los ficheros de la aplicación.
- **node\_modules:** Aquí se almacenará el código de los paquetes que se han instalado a través de `npm install`. Si aún no se ha ejecutado este comando, no estará presente.
- **src:** Contiene el código fuente de la aplicación. Se detallan a continuación sus subdirectorios y ficheros:
  - **assets:** Elementos multimedia utilizados como imágenes, vídeos...
  - **components:** Componentes Web de la aplicación. Elementos reutilizables en distintas pantallas.
  - **mixins:** Fragmentos de código que son utilizados en distintas vistas y que se extraen a ficheros `mixins` para no duplicar código. Es el caso de la gestión de capas de espera, sistema de alertas...
  - **plugins:** Elementos que se aplican en todo el ámbito de la aplicación. En este caso contiene el fichero de internacionalización.
  - **routing:** Aquí se almacenan los ficheros que gestionan el ruteo de la aplicación.
  - **services:** Ficheros que contienen, en forma de métodos, las peticiones HTTP que se hacen al backend.
  - **views:** Almacena las vistas, que contienen una plantilla de código HTML, código JavaScript y CSS. Básicamente una vista es una pantalla de la aplicación.
  - **vuex:** En su interior se encuentra el fichero que gestiona el store, almacén centralizado de la información de sesión del usuario.
  - **App.vue:** Sería la vista principal del sistema. Define la estructura de toda la aplicación. Las vistas, además del componente **Header** se incrustarán en la estructura aquí definida.
  - **main.js:** Realiza todos los `imports` necesarios a nivel global del sistema y establece las configuraciones iniciales. Lanza la instancia de Vue (representada por el componente **App.vue**).
- **index.html:** Define la estructura HTML de la aplicación, dejando un `div` con `id="app"` en el que se incrustará la aplicación de Vue.
- **package.json:** Contiene información del proyecto, las dependencias necesarias para el correcto funcionamiento de la aplicación y scripts `npm` ejecutables.
- **package-lock.json:** Derivado del anterior, se genera automáticamente.
- **server.js:** Fichero que crea un servidor de `express` simple para la aplicación, de manera que pueda ser desplegada.

El resto de los ficheros o directorios son elementos que rara vez habrá que modificar y no se considera importante conocer su contenido.

#### 9.4.3.2 Estructura de directorios del backend

A continuación, se detallan los directorios y ficheros que se considera que contienen información importante que un programador debe conocer, partiendo de la raíz del proyecto:

- **pdftools:** Contiene los ficheros de código fuente de la aplicación de manipulación de documentos PDF, documentación... Se desglosa a continuación el contenido de este directorio:
  - **docs:** Documentación de las distintas clases, métodos y funciones del proyecto.
  - **migrations:** Contiene las migraciones que han sido realizadas en el proyecto a lo largo del tiempo de desarrollo. Cambios en los modelos que provocan que haya que actualizar la base de datos.
  - **\_\_init\_\_.py:** Identificador de un paquete Python [58].
  - **admin.py:** Está pensado para el entorno de administrador de sistema, no se emplea por el momento en la aplicación.
  - **apps.py:** Configuración de la aplicación.
  - **fileprocessor.py:** Contiene los distintos procesadores de documentos, uno para cada tipo de procesamiento.
  - **models.py:** En su interior se encuentran los modelos o entidades que emplea la aplicación.
  - **obfuscation.py:** Funciones de utilidad para ofuscar las rutas en las que se almacenarán los ficheros.
  - **serializers.py:** Serializadores que transforman la estructura de los datos según sea necesario.
  - **tests.py:** Pruebas unitarias de la aplicación.
  - **utils.py:** Funciones de utilidad general.
  - **viewsets.py:** Contiene los viewsets de la aplicación.
- **tfg:** Ficheros de código del proyecto en general. Un proyecto Django (**tfg**), puede tener varias aplicaciones. En este caso, solo tiene una (**pdftools**). A continuación, se indica su contenido:
  - **\_\_init.py\_\_:** Tiene exactamente el mismo cometido que el fichero de mismo nombre del directorio **pdftools**.
  - **custom\_azure.py:** Contiene una clase que establece la configuración de guardado de documentos en el contenedor de azure.
  - **routers.py:** En él se asocian las url a Viewsets.
  - **settings.py:** Configuración del proyecto en general.
  - **urls.py:** Configuración de URLs del sistema.
- **venv:** Contiene los ficheros asociados al entorno virtual. Dentro hay scripts que permiten activar este entorno. Aparecerá una vez creado el entorno virtual.
- **.gitignore:** Incluye carpetas o extensiones que no deben ser almacenadas en git.
- **db.sqlite:** Fichero de la base de datos SQLite que puede ser utilizada en desarrollo.
- **manage.py:** Permite realizar múltiples tareas sobre el proyecto. Arrancar el servidor (*python manage.py runserver*), crear migraciones (*python manage.py makemigrations*), y aplicar estas migraciones (*python manage.py migrate*) son quizás las más importantes.
- **Procfile:** Información para el servidor de Heroku en el que se despliega la aplicación.

- **requirements.txt:** Recoge todos los paquetes empleados en el proyecto junto a su versión. Sirve para instalarlos fácilmente en un nuevo entorno.

#### 9.4.4 Ejemplo de creación de una nueva funcionalidad

En este apartado se recogerán los pasos necesarios para crear una nueva funcionalidad, suponiendo que esta es una manipulación de un PDF para realizar una operación que no se encuentra implementada. Se indican los ficheros que habría que crear o modificar.

##### 9.4.4.1 Cambios en el backend

1. Añadir un nuevo método al viewset **ManipulateFilesViewSet** que represente esta acción. La estructura del método será similar a la del resto (**removePages,merge...**). Llamará a un procesador (que se detallará en el apartado 2) y guardará los datos empleando el **PDFFileSerializer** ya existente.
2. Añadir un nuevo procesador que herede de **FileProcessor** en el que se redefina el método **customProcessing**.

##### 9.4.4.2 Cambios en el frontend

1. Habría que crear una nueva vista para esta operación, similar a Merge.vue, Order.vue... En ella se incluiría la estructura HTML, el código JavaScript y los estilos análogamente a lo que se hace en las mencionadas. Se podrían crear opcionalmente componentes si se detecta que algunos de ellos podrían ser reutilizables. Se podrá hacer uso de los mixins existentes para funcionalidades “repetidas”.
2. Añadir un nuevo método en el fichero manipulateFiles.js, dentro de la carpeta **services** que envíe la petición al backend. (El último elemento de la ruta deberá coincidir con el nombre del método del viewset creado en back). Por ejemplo, para la unión, la ruta acaba con **/manipulate/merge/**. El “merge” es lo que establece el método a aplicar en el viewset.
3. Asociar a esta vista una entrada en el array **routes** del fichero **router.js**
4. Crear enlaces en la barra de navegación y pantalla principal que redirijan a esta nueva vista.

# Capítulo 10. Conclusiones y ampliaciones

## 10.1 Conclusiones

Una vez acabado el proyecto se puede concluir que ha sido llevado a cabo de manera satisfactoria, habiendo sido ya probado por una serie de usuarios que han conseguido manipular sus documentos PDF de la manera que esperaban y según lo definido en los objetivos del proyecto.

Pese a tener el sistema desplegado en servidores en la nube con capacidad de almacenamiento y procesamiento no muy elevada (debido a que son opciones de bajo costo), el sistema responde bien, ofreciendo unos tiempos de espera aceptables. A no ser que se tenga una conexión a internet muy rápida, la mayor parte del tiempo que espera un usuario es debido a lo que tarda en enviarse a través de la red el documento, no tiempo real de procesamiento.

La selección de tecnologías del proyecto trajo consigo problemas en las primeras etapas del desarrollo, que hicieron que los tiempos se dilatasen, pero una vez terminado el proyecto, supone una gran satisfacción haberlos usado, ya que considero que aporta mucho en cuanto a mi formación. Gracias a esto, me he demostrado a mí mismo que soy capaz de afrontar un proyecto de cierta dimensión empleando herramientas que no había usado nunca, haciéndome ver que, teniendo unos conocimientos sólidos de base, las tecnologías son un apoyo, pero no la parte primordial.

En términos de conocimiento adquirido, cabe destacar también la experiencia de realizar un sistema pasando por todas las fases del ciclo de vida de un proyecto software, en las que una decisión en una fase condiciona las sucesivas.

También es la primera vez que realizo un proyecto con una clara separación entre subsistemas *backend* y *frontend*, diferenciados tanto en la etapa de desarrollo, como en la etapa de despliegue, teniendo una máquina dedicada cada uno de estos subsistemas.

En definitiva, la sensación final es muy buena. Obviamente, ha habido decisiones en el proyecto que no han sido las mejores: malas estimaciones, optimizaciones tempranas que hubo que desechar posteriormente... Pero todas estas “equivocaciones” suponen un aprendizaje. Este aprendizaje junto con la aplicación resultante hace que me sienta especialmente orgulloso tanto del producto como del proceso.

## 10.2 Ampliaciones

En este apartado se recogen funcionalidades que pueden implementarse en el sistema para mejorar el comportamiento general de la aplicación. No han sido implementadas a priori porque se salían del alcance del proyecto y el tiempo de realización de este aumentaría considerablemente.



### 10.2.1 Añadir más acciones a la aplicación

Como se comentó al principio de la memoria, existen gran cantidad de operaciones que se podrían realizar sobre ficheros PDF. No obstante, el tiempo que se le debe dedicar a un proyecto de estas características hace que se pueda implementar únicamente una muestra de las más utilizadas, dejando un diseño que permita añadir nuevas operaciones de forma simple.

Algunas de estas operaciones que se podrían añadir son: proteger/desproteger ficheros, añadir marcas de agua a las páginas del archivo, comprimir el documento o permitir rotarlo.

### 10.2.2 Encadenamiento de acciones

Actualmente, cuando se realiza cualquier acción sobre un documento, este puede ser descargado, tanto una vez que acaba el procesamiento como desde el historial de documentos.

Si un usuario quisiese realizar dos operaciones consecutivas sobre un documento, por ejemplo, extraer una serie de páginas y posteriormente ordenarlo, tendría que extraer las páginas, descargar el documento y volver a subirlo en la pantalla de ordenación.

Sería interesante proporcionar una opción para que una vez se obtiene el resultado del fichero, se pudiese continuar usándolo para otras manipulaciones sin necesidad de descargarlo.

### 10.2.3 Emplear ficheros del historial para realizar acciones

Esta ampliación sería similar a la anterior. Consistiría en, sin necesidad de descargar los ficheros, poder acceder a manipularlos desde el historial de operaciones, abriendo la página de manipulación correspondiente con el fichero o los ficheros precargados.

### 10.2.4 Permitir la selección de documentos de distintas carpetas

Esta ampliación viene derivada de los resultados obtenidos en las pruebas de usabilidad, donde se me indicó que, en el caso de querer unir ficheros, el selector solo permitía añadir documentos que se encontrasen en una misma carpeta.

Para conseguir este funcionamiento habría que cambiar la forma en que se van añadiendo los archivos, no obligando al usuario a añadir todos a la vez y permitiendo que vaya seleccionándolos de forma secuencial, ya que el selector de ficheros no permite escoger elementos de más de una carpeta al mismo tiempo.

### 10.2.5 Permitir la selección de la página de referencia para el recorte de documentos

De nuevo, es una ampliación derivada de las pruebas de usabilidad. En ellas, un usuario me indicó que podría ser confuso el recorte de páginas, ya que muchas veces, la página que se toma como referencia, que es la primera, no tiene la misma estructura que el resto. Como

consecuencia, el usuario podrá recortar zonas que no desea, teniendo que realizar varios intentos para conseguir el resultado esperado.

Como solución a esta problemática, se podría añadir una opción para que el usuario indique cual es la página que desea tomar como referencia para el recorte y mostrar la previsualización de dicha página en lugar de la primera.

## 10.2.6 Mejorar las opciones de usuarios

Actualmente los usuarios únicamente pueden registrarse con un usuario y una contraseña, pero no pueden realizar ningún cambio en su perfil.

Existe un *dashboard* de administración, generado por Django, en el que el administrador del sistema podría cambiar los datos de todos los usuarios, e incluso podría dar acceso a estos puntualmente para que cambien sus propios datos, como la contraseña, en caso de ser necesario.

No obstante, sería interesante permitir a los usuarios un mecanismo para tareas como gestionar su perfil o recuperar su contraseña sin necesidad de tener que ponerse en contacto con el administrador del sistema.

## 10.2.7 Internacionalización de la aplicación

Por sus características se trata de una aplicación que, en potencia, podría ser usada por usuarios de múltiples nacionalidades. Por tanto, se considera que sería interesante internacionalizarla con el objetivo de tener un número de usuarios potenciales mayor.

Un gran paso para realizar esta ampliación ya está realizado, y es que todos los textos del *frontend* están parametrizados haciendo uso de un plugin de internacionalización de Vue. Con esto, lo que se debería realizar es únicamente la adición de los textos en otros idiomas y permitir al usuario cambiar al idioma que desee.

# Capítulo 11. Planificación del Proyecto y Presupuesto finales

En este apartado se revisará la planificación temporal inicial del proyecto, adaptando el tiempo de realización de las tareas, añadiendo algunas que no se habían tenido en cuenta y otros aspectos de este tipo. También se actualizará el presupuesto en base a esta nueva planificación.

## 11.1 Planificación final

La duración final del proyecto ha sido de 372 horas, iniciándose el mismo día indicado en la planificación inicial, el 15 de diciembre de 2020, y finalizando el 7 de junio de 2021.

La planificación ha sido realizada empleando la herramienta Microsoft Project y se encuentra en el archivo adjunto PlanTFGFinal.mpp, dentro de la carpeta adjunta Planificación y Presupuesto.

Se ha ajustado el horario de trabajo mientras que se desarrollaba la aplicación puesto que se detectó durante la fase de implementación un retraso respecto a la planificación inicial en cuanto a tiempos.

El nuevo calendario pasa a ser:

- Días de trabajo: lunes a viernes.
- Horario: 17:00 a 20:00.

Los grupos de tareas son los mismos que para la planificación inicial. No obstante, han sufrido variaciones en cuanto a los tiempos y a las tareas que los conforman, ya que se han añadido elementos nuevos que no habían sido detectados o no estaban definidos al principio. Además, algunas tareas incluidas en la planificación inicial fueron eliminadas, por considerar que no tenían especial relevancia en el proyecto.

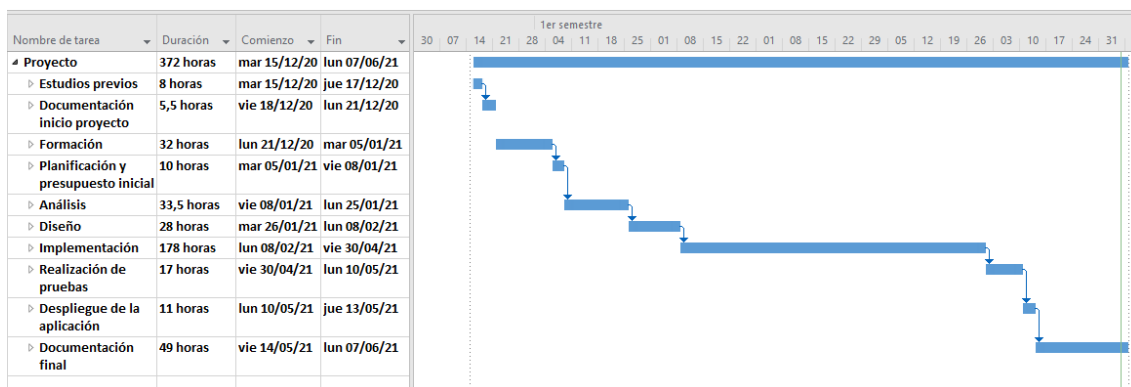
La tabla presentada a continuación incluye la misma información que la realizada durante la planificación inicial, ajustada a lo que realmente se tardó en ejecutar cada fase.

Nombre de tarea	Duración	Comienzo	Fin
<b>Estudios previos</b>	<b>8 horas</b>	<b>mar 15/12/20</b>	<b>jue 17/12/20</b>
<b>Documentación inicio proyecto</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>lun 21/12/20</b>
<b>Formación</b>	<b>32 horas</b>	<b>lun 21/12/20</b>	<b>mar 05/01/21</b>
<b>Planificación y presupuesto inicial</b>	<b>10 horas</b>	<b>mar 05/01/21</b>	<b>vie 08/01/21</b>
<b>Análisis</b>	<b>33,5 horas</b>	<b>vie 08/01/21</b>	<b>lun 25/01/21</b>

<b>Diseño</b>	<b>28 horas</b>	<b>mar 26/01/21</b>	<b>lun 08/02/21</b>
<b>Implementación</b>	<b>178 horas</b>	<b>lun 08/02/21</b>	<b>vie 30/04/21</b>
<b>Realización de pruebas</b>	<b>17 horas</b>	<b>vie 30/04/21</b>	<b>lun 10/05/21</b>
<b>Despliegue de la aplicación</b>	<b>11 horas</b>	<b>lun 10/05/21</b>	<b>jue 13/05/21</b>
<b>Documentación final</b>	<b>49 horas</b>	<b>vie 14/05/21</b>	<b>lun 07/06/21</b>

Se puede observar que el principal desajuste en cuanto a tiempos empleados es el de la implementación. Esto se debe a que, como comenté en otros apartados, el principio del desarrollo fue más lento de lo esperado como consecuencia de la falta de experiencia con las tecnologías empleadas.

Se presenta en la siguiente imagen este desglose de trabajo junto a un diagrama de Gantt:



**Figura 11.1 Actividades y diagrama de Gantt de alto nivel de la planificación final**

Se muestra ahora el desglose de las tareas de cada grupo. La explicación de la naturaleza de las incluidas en cada agrupación se puede encontrar en la *Planificación inicial*. En estos apartados me limitaré a recoger el listado de tareas.

### 11.1.1 Estudios previos

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.1</b>	<b>Estudios previos</b>	<b>8 horas</b>	<b>mar 15/12/20</b>	<b>jue 17/12/20</b>
1.1.1	Estudio de los objetivos del proyecto	2 horas	mar 15/12/20	mar 15/12/20
1.1.2	Búsqueda de información sobre librerías para modificación de ficheros PDF	2 horas	mié 16/12/20	mié 16/12/20

1.1.3	Realización de pruebas con diferentes librerías	4 horas	mié 16/12/20	jue 17/12/20
-------	---	---------	-----------------	--------------

### 11.1.2 Documentación inicio proyecto

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.2</b>	<b>Documentación inicio proyecto</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>lun 21/12/20</b>
<b>1.2.1</b>	<b>Introducción</b>	<b>5,5 horas</b>	<b>vie 18/12/20</b>	<b>lun 21/12/20</b>
1.2.1.1	Justificación	0,5 horas	vie 18/12/20	vie 18/12/20
1.2.1.2	Objetivo	0,5 horas	vie 18/12/20	vie 18/12/20
<b>1.2.1.3</b>	<b>Estudio situación actual</b>	<b>4,5 horas</b>	<b>vie 18/12/20</b>	<b>lun 21/12/20</b>
1.2.1.3.1	Estudio previo	1,5 horas	vie 18/12/20	vie 18/12/20
1.2.1.3.2	Evaluación de alternativas	3 horas	vie 18/12/20	lun 21/12/20

### 11.1.3 Formación

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>1.3</b>	<b>Formación</b>	<b>32 horas</b>	<b>lun 21/12/20</b>	<b>mar 05/01/21</b>	<b>12</b>
1.3.1	Formación en Vue	12 horas	lun 21/12/20	vie 25/12/20	
1.3.2	Formación en Django	12 horas	vie 25/12/20	jue 31/12/20	14
1.3.3	Formación Integración Vue+Django	8 horas	jue 31/12/20	mar 05/01/21	14;15

### 11.1.4 Planificación y presupuesto inicial

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.4</b>	<b>Planificación y presupuesto inicial</b>	<b>10 horas</b>	<b>mar 05/01/21</b>	<b>vie 08/01/21</b>
1.4.1	Realización del WBS inicial	3 horas	mar 05/01/21	mié 06/01/21
1.4.2	Estimar actividades del WBS	2 horas	mié 06/01/21	jue 07/01/21
1.4.3	Presupuesto inicial	5 horas	jue 07/01/21	vie 08/01/21

### 11.1.5 Análisis

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.5</b>	<b>Análisis</b>	<b>33,5 horas</b>	<b>vie 08/01/21</b>	<b>lun 25/01/21</b>
1.5.1	Lectura en detalle de la información aportada por el tutor	1 hora	vie 08/01/21	lun 11/01/21
1.5.2	Establecimiento del alcance del sistema	1 hora	lun 11/01/21	lun 11/01/21
<b>1.5.3</b>	<b>Análisis de requisitos</b>	<b>8 horas</b>	<b>lun 11/01/21</b>	<b>jue 14/01/21</b>
1.5.3.1	Extracción de requisitos	6 horas	lun 11/01/21	mié 13/01/21
1.5.3.2	Definición de requisitos	2 horas	mié 13/01/21	jue 14/01/21
1.5.4	Identificación de actores del sistema	1 hora	jue 14/01/21	jue 14/01/21
1.5.5	Análisis de casos de uso	5 horas	jue 14/01/21	lun 18/01/21
1.5.6	Identificación de los subsistemas	3 horas	lun 18/01/21	mar 19/01/21

<b>1.5.7</b>	<b>Diagrama de clases preliminar del Análisis</b>	<b>9 horas</b>	<b>mar 19/01/21</b>	<b>vie 22/01/21</b>
1.5.7.1	Diagrama Subsistema Backend	4 horas	mar 19/01/21	mié 20/01/21
1.5.7.2	Diagrama subsistema Frontend	5 horas	mié 20/01/21	vie 22/01/21
1.5.8	Análisis de interfaces de usuario	5 horas	vie 22/01/21	lun 25/01/21
1.5.9	Análisis del plan de pruebas	0,5 horas	lun 25/01/21	lun 25/01/21

### 11.1.6 Diseño

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.6</b>	<b>Diseño</b>	<b>28 horas</b>	<b>mar 26/01/21</b>	<b>lun 08/02/21</b>
<b>1.6.1</b>	<b>Definición de la arquitectura del sistema</b>	<b>4 horas</b>	<b>mar 26/01/21</b>	<b>mié 27/01/21</b>
<b>1.6.1.1</b>	<b>Realización de diagramas</b>	<b>4 horas</b>	<b>mar 26/01/21</b>	<b>mié 27/01/21</b>
1.6.1.1.1	Diagrama de paquetes	2 horas	mar 26/01/21	mar 26/01/21
1.6.1.1.2	Diagrama de despliegue	2 horas	mar 26/01/21	mié 27/01/21
1.6.2	Diseño de clases	8 horas	mié 27/01/21	vie 29/01/21
1.6.3	Diseño de interfaces de usuario	4 horas	lun 01/02/21	mar 02/02/21
1.6.4	Diagramas de Interacción	2 horas	mar 02/02/21	mar 02/02/21
1.6.5	Diseño de la base de datos	2 horas	mié 03/02/21	mié 03/02/21
<b>1.6.6</b>	<b>Especificación técnica del plan de pruebas</b>	<b>8 horas</b>	<b>mié 03/02/21</b>	<b>lun 08/02/21</b>

1.6.6.1	Pruebas funcionales	6 horas	mié 03/02/21	vie 05/02/21
1.6.6.2	Pruebas de usabilidad	2 horas	vie 05/02/21	lun 08/02/21

### 11.1.7 Implementación

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.7</b>	<b>Implementación</b>	<b>178 horas</b>	<b>lun 08/02/21</b>	<b>vie 30/04/21</b>
1.7.1	Instalación de los entornos de trabajo	3 horas	lun 08/02/21	mar 09/02/21
<b>1.7.2</b>	<b>Inicialización</b>	<b>6 horas</b>	<b>mar 09/02/21</b>	<b>jue 11/02/21</b>
1.7.2.1	Inicialización del proyecto frontend	3 horas	mar 09/02/21	mié 10/02/21
1.7.2.2	Inicialización del proyecto backend	3 horas	mié 10/02/21	jue 11/02/21
1.7.3	Definición de modelos y serializadores backend	3 horas	jue 11/02/21	vie 12/02/21
1.7.4	Pantalla de inicio	4 horas	vie 12/02/21	lun 15/02/21
<b>1.7.5</b>	<b>Sistema de inicio de sesion</b>	<b>16 horas</b>	<b>lun 15/02/21</b>	<b>lun 22/02/21</b>
1.7.5.1	Backend Inicio de sesión	8 horas	lun 15/02/21	jue 18/02/21
1.7.5.2	Frontend inicio de Sesión	8 horas	jue 18/02/21	lun 22/02/21
<b>1.7.6</b>	<b>Sistema de registro</b>	<b>8 horas</b>	<b>mar 23/02/21</b>	<b>jue 25/02/21</b>
<b>1.7.7</b>	<b>Eliminación/Extracción de páginas</b>	<b>32 horas</b>	<b>jue 25/02/21</b>	<b>vie 12/03/21</b>
<b>1.7.8</b>	<b>Mezcla de documentos</b>	<b>14 horas</b>	<b>vie 12/03/21</b>	<b>jue 18/03/21</b>



1.7.8.1	Backend Mezcla de documentos	6 horas	vie 12/03/21	mar 16/03/21
1.7.8.2	Frontend mezcla de documentos	8 horas	mar 16/03/21	jue 18/03/21
<b>1.7.9</b>	<b>Ordenación de documentos</b>	<b>20 horas</b>	<b>vie 19/03/21</b>	<b>lun 29/03/21</b>
1.7.9.1	Backend ordenación de documentos	8 horas	vie 19/03/21	mar 23/03/21
1.7.9.2	Frontend ordenación de documentos	12 horas	mar 23/03/21	lun 29/03/21
<b>1.7.10</b>	<b>Recorte de documentos</b>	<b>34 horas</b>	<b>lun 29/03/21</b>	<b>mar 13/04/21</b>
1.7.10.1	Backend recorte de documentos	14 horas	lun 29/03/21	lun 05/04/21
1.7.10.2	Frontend recorte de documentos	20 horas	lun 05/04/21	mar 13/04/21
<b>1.7.11</b>	<b>Historial de operaciones</b>	<b>9 horas</b>	<b>mié 14/04/21</b>	<b>vie 16/04/21</b>
1.7.12	Sistema de ruteo y navegación	4 horas	lun 19/04/21	mar 20/04/21
<b>1.7.13</b>	<b>División de documentos</b>	<b>8 horas</b>	<b>mar 20/04/21</b>	<b>jue 22/04/21</b>
1.7.13.1	Frontend división de documentos	3 horas	mar 20/04/21	mié 21/04/21
1.7.13.2	Backend división de documentos	5 horas	mié 21/04/21	jue 22/04/21
1.7.14	Conectar con sistema de ficheros de Azure	4 horas	vie 23/04/21	lun 26/04/21
1.7.15	Migración de base de datos a PostgreSQL	2 horas	lun 26/04/21	lun 26/04/21
1.7.16	Introducir limitaciones de tamaños de fichero	3 horas	mar 27/04/21	mar 27/04/21
1.7.17	Corrección de errores detectados	8 horas	mié 28/04/21	vie 30/04/21

### 11.1.8 Realización de pruebas

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.8</b>	<b>Realización de pruebas</b>	<b>17 horas</b>	<b>vie 30/04/21</b>	<b>lun 10/05/21</b>
1.8.1	Pruebas unitarias	10 horas	vie 30/04/21	mié 05/05/21
1.8.2	Pruebas funcionales manuales	6 horas	jue 06/05/21	vie 07/05/21
1.8.3	Pruebas de usabilidad	1 hora	lun 10/05/21	lun 10/05/21

### 11.1.9 Despliegue de la aplicación

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.9</b>	<b>Despliegue de la aplicación</b>	<b>11 horas</b>	<b>lun 10/05/21</b>	<b>jue 13/05/21</b>
1.9.1	Estudio de alternativas y formación para el despliegue	5 horas	lun 10/05/21	mar 11/05/21
1.9.2	Despliegue frontend	3 horas	mié 12/05/21	mié 12/05/21
1.9.3	Despliegue backend	3 horas	jue 13/05/21	jue 13/05/21

### 11.1.10 Documentación final

Número de esquema	Nombre de tarea	Duración	Comienzo	Fin
<b>1.10</b>	<b>Documentación final</b>	<b>49 horas</b>	<b>vie 14/05/21</b>	<b>lun 07/06/21</b>
1.10.1	Detallar aspectos teóricos	3 horas	vie 14/05/21	vie 14/05/21
<b>1.10.2</b>	<b>Documentación sobre implementación</b>	<b>7 horas</b>	<b>lun 17/05/21</b>	<b>mié 19/05/21</b>
1.10.2.1	Estándares y normas seguidos	1 hora	lun 17/05/21	lun 17/05/21

1.10.2.2	Lenguajes utilizados	2 horas	lun 17/05/21	lun 17/05/21
1.10.2.3	Herramientas y programas empleados en el desarrollo	2 horas	mar 18/05/21	mar 18/05/21
1.10.2.4	Creación del sistema	2 horas	mar 18/05/21	mié 19/05/21
<b>1.10.3</b>	<b>Documentación sobre el desarrollo de las pruebas</b>	<b>7 horas</b>	<b>mié 19/05/21</b>	<b>vie 21/05/21</b>
1.10.3.1	Pruebas funcionales	4 horas	mié 19/05/21	jue 20/05/21
1.10.3.2	Pruebas de usabilidad	3 horas	jue 20/05/21	vie 21/05/21
<b>1.10.4</b>	<b>Manuales del sistema</b>	<b>10,5 horas</b>	<b>vie 21/05/21</b>	<b>jue 27/05/21</b>
1.10.4.1	Manual de instalación	0,5 horas	vie 21/05/21	vie 21/05/21
1.10.4.2	Manual de ejecución y despliegue	4 horas	vie 21/05/21	mar 25/05/21
1.10.4.3	Manual de usuario	3 horas	mar 25/05/21	mié 26/05/21
1.10.4.4	Manual del programador	3 horas	mié 26/05/21	jue 27/05/21
<b>1.10.5</b>	<b>Conclusiones y ampliaciones</b>	<b>5 horas</b>	<b>jue 27/05/21</b>	<b>vie 28/05/21</b>
1.10.5.1	Conclusiones	2 horas	jue 27/05/21	jue 27/05/21
1.10.5.2	Ampliaciones	3 horas	jue 27/05/21	vie 28/05/21
1.10.6	Referencias bibliográficas	2 horas	vie 28/05/21	lun 31/05/21
<b>1.10.7</b>	<b>Apéndices</b>	<b>4 horas</b>	<b>lun 31/05/21</b>	<b>mar 01/06/21</b>
1.10.7.1	Glosario	2 horas	lun 31/05/21	mar 01/06/21
1.10.7.2	Contenido entregado	1 hora	mar 01/06/21	mar 01/06/21

1.10.7.3	Código fuente	1 hora	mar 01/06/21	mar 01/06/21
1.10.8	Memoria del proyecto	2 horas	mar 01/06/21	mié 02/06/21
<b>1.10.9</b>	<b>Planificación y presupuesto final</b>	<b>8,5 horas</b>	<b>mié 02/06/21</b>	<b>lun 07/06/21</b>
1.10.9.1	Adaptación WBS	2,5 horas	mié 02/06/21	jue 03/06/21
1.10.9.2	Registro tiempos totales	1 hora	jue 03/06/21	jue 03/06/21
1.10.9.3	Presupuesto final	5 horas	jue 03/06/21	lun 07/06/21

## 11.2 Presupuesto de costes final

Tomando como referencia esta planificación, se presenta el presupuesto de costes final del proyecto. En este caso, no se incluirá el modelo de empresa, se asume que es el mismo que en el inicial.

### 11.2.1 Desarrollo del presupuesto de costes final

Las partidas del presupuesto serán las mismas que en el caso inicial:

- Partida 1: Puesta en marcha (Estudios y planificación inicial).
- Partida 2: Análisis y Diseño del sistema.
- Partida 3: Implementación, pruebas y despliegue.
- Partida 4: Documentación.

Se recuerda que el coste por hora sin beneficios es 30€/hora.

La estructura de las tablas será la misma que se explica en *Desarrollo del presupuesto de costes inicial*. Además, el IVA no está incluido en ninguno de los cálculos que se presentan a continuación.

## 11.2.1.1 Partida 1: Puesta en marcha (Estudios y planificación inicial)

<b>Partida 1: Puesta en marcha (Estudios y planificación inicial)</b>						
n1	n2	Descripción	Horas	Precio/Hora	Subtotal(n2)	Subtotal(n1)
1		<b>Estudios previos</b>				<b>240 €</b>
	1	Estudio de los objetivos del proyecto	2	30 €	60 €	
	2	Búsqueda de información de librerías para la modificación de ficheros pdf	2	30 €	60 €	
	3	Realización de pruebas con diferentes librerías	4	30 €	120 €	
2		<b>Formación</b>				<b>960 €</b>
	1	Formación en Vue	12	30 €	360 €	
	2	Formación en Django	12	30 €	360 €	
	3	Formación Integración Vue+Django	8	30 €	240 €	
3		<b>Planificación y presupuesto inicial</b>				<b>300 €</b>
	1	Realización del WBS inicial	3	30 €	90 €	
	2	Estimar actividades del WBS	2	30 €	60 €	
	3	Presupuesto inicial	5	30 €	150 €	
<b>Total</b>						<b>1.500 €</b>

## 11.2.1.2 Partida 2: Análisis y Diseño del Sistema

Partida 2: Análisis y Diseño del sistema										
n1	n2	n3	n4	Descripción	Horas	Precio/Hora	Subtotal(n4)	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1				<b>Análisis</b>						<b>1.005 €</b>
	1			Lectura en detalle de la información aportada por el tutor	1	30 €			<b>30 €</b>	
	2			Establecimiento del alcance del sistema	1	30 €			<b>30 €</b>	
	3			<b>Análisis de requisitos</b>					<b>240 €</b>	
		1		Extracción de requisitos	6	30 €		180 €		
		2		Definición de requisitos	2	30 €		60 €		
	4			Identificación de los actores del sistema	1	30 €			<b>30 €</b>	
	5			Análisis de casos de uso	5	30 €			<b>150 €</b>	
	6			Identificación de los subsistemas	3	30 €			<b>90 €</b>	
	7			<b>Diagrama de clases preliminar del Análisis</b>					<b>270 €</b>	
		1		Diagrama Subsistema Frontend	4	30 €		120 €		
		2		Diagrama Subsistema Backend	5	30 €		150 €		
	8			Análisis de interfaces de usuario	5	30 €			<b>150 €</b>	
	9			Análisis del plan de pruebas	0,5	30 €			<b>15 €</b>	
2				<b>Diseño</b>						<b>840 €</b>

1		<b>Definición de la arquitectura</b>					<b>120 €</b>
	1	<i>Realización de diagramas</i>					120 €
		1	Diagrama de paquetes	2	30 €	60 €	
		2	Diagrama de despliegue	2	30 €	60 €	
2			Diseño de clases	8	30 €		<b>240 €</b>
3			Diseño de interfaces de usuario	4	30 €		<b>120 €</b>
4			Diagramas de Interacción	2	30 €		<b>60 €</b>
5			Diseño de la base de datos	2	30 €		<b>60 €</b>
6			<b>Especificación técnica del plan de pruebas</b>				<b>240 €</b>
	1		Pruebas funcionales	6	30 €	180 €	
	2		Pruebas de usabilidad	2	30 €	60 €	

<b>Total</b>	<b>1.845 €</b>
--------------	----------------

11.2.1.3 Partida 3: Implementación, pruebas y despliegue

**Partida 3: Implementación, pruebas y despliegue**

n1	n2	n3	Descripción	Horas	Precio/Hora	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1			<b>Implementación</b>					<b>5.340 €</b>
	1		Instalación de los entornos de trabajo	3	30 €		<b>90 €</b>	
	2		<b>Inicialización</b>				<b>180 €</b>	
		1	Inicialización del proyecto frontend	3	30 €	90 €		
		2	Inicialización del proyecto backend	3	30 €	90 €		
	3		Definición de modelos y serializadores backend	3	30 €		<b>90 €</b>	
	4		Pantalla de inicio	4	30 €		<b>120 €</b>	
	5		<b>Sistema de inicio de sesión</b>				<b>480 €</b>	
		1	Backend Inicio de sesión	8	30 €	240 €		
		2	Frontend inicio de sesión	8	30 €	240 €		
	6		<b>Sistema de registro</b>				<b>240 €</b>	
		1	Frontend sistema de registro	3	30 €	90 €		
		2	Backend sistema de registro	5	30 €	150 €		
	7		<b>Eliminación /Extracción de páginas</b>				<b>960 €</b>	
		1	Backend Eliminación/Extracción de páginas	18	30 €	540 €		
		2	Frontend Eliminación/Extracción de páginas	14	30 €	420 €		
	8		<b>Mezcla de documentos</b>				<b>420 €</b>	



	1	Backend Mezcla de documentos	6	30 €	180 €	
	2	Frontend Mezcla de documentos	8	30 €	240 €	
9		<b>Ordenación de documentos</b>				<b>600 €</b>
	1	Backend ordenación de documentos	8	30 €	240 €	
	2	Frontend ordenación de documentos	12	30 €	360 €	
10		<b>Recorte de documentos</b>				<b>1.020 €</b>
	1	Backend recorte de documentos	14	30 €	420 €	
	2	Frontend recorte de documentos	20	30 €	600 €	
11		<b>Historial de operaciones</b>				<b>270 €</b>
	1	Backend historial de operaciones	3	30 €	90 €	
	2	Frontend historial de operaciones	6	30 €	180 €	
12		Sistema de ruteo y navegación	4	30 €		<b>120 €</b>
13		<b>División de documentos</b>				<b>240 €</b>
	1	Frontend división de documentos	3	30 €	90 €	
	2	Backend división de documentos	5	30 €	150 €	
14		Conectar con sistema de ficheros de Azure	4	30 €		<b>120 €</b>
15		Migración de base de datos a PostgreSQL	2	30 €		<b>60 €</b>
16		Introducir limitaciones de tamaños de fichero	3	30 €		<b>90 €</b>

	17	Corrección de errores detectados	8	30 €		<b>240 €</b>	
2		<b>Realización de pruebas</b>		30 €			<b>510 €</b>
	1	Pruebas unitarias	10	30 €		<b>300 €</b>	
	2	Pruebas funcionales manuales	6	30 €		<b>180 €</b>	
	3	Pruebas de usabilidad	1	30 €		<b>30 €</b>	
3		<b>Despliegue de la aplicación</b>					<b>330 €</b>
	1	Estudio de alternativas y formación para el despliegue	5	30 €		<b>150 €</b>	
	2	Despliegue frontend	3	30 €		<b>90 €</b>	
	3	Despliegue backend	3	30 €		<b>90 €</b>	

<b>Total</b>	<b>6.180 €</b>
--------------	----------------

11.2.1.4 Partida 4: Documentación

Partida 4: Documentación										
n1	n2	n3	n4	Descripción	Horas	Precio/Hora	Subtotal(n4)	Subtotal(n3)	Subtotal(n2)	Subtotal(n1)
1				<b>Documentación inicio proyecto</b>						<b>165 €</b>
	2			<b>Introducción</b>					<b>165 €</b>	
		1		Justificación	0,5	30 €		15 €		
		2		Objetivo	0,5	30 €		15 €		

		3	<i>Estudio situación actual</i>				135 €		
		1	Estudio previo	1,5	30 €	45 €			
		2	Evaluación de alternativas	3	30 €	90 €			
2			<b>Documentación final</b>						<b>1.470 €</b>
	1		Detallar aspectos teóricos	3	30 €				<b>90 €</b>
	2		<b>Documentación sobre implementación</b>						<b>210 €</b>
		1	Estándares y normas seguidos	1	30 €		30 €		
		2	Lenguajes utilizados	2	30 €		60 €		
		3	Herramientas y programas empleados en el desarrollo	2	30 €		60 €		
		4	Creación del sistema	2	30 €		60 €		
	3		<b>Documentación sobre el desarrollo de las pruebas</b>						<b>210 €</b>
		1	Pruebas funcionales	4	30 €		120 €		
		2	Pruebas de usabilidad	3	30 €		90 €		
	4		<b>Manuales del sistema</b>						<b>315 €</b>
		1	Manual de instalación	0,5	30 €		15 €		
		2	Manual de ejecución y despliegue	4	30 €		120 €		
		3	Manual de usuario	3	30 €		90 €		

	4	Manual del programador	3	30 €	90 €	
5		<b>Conclusiones y ampliaciones</b>				<b>150 €</b>
	1	Conclusiones	2	30 €	60 €	
	2	Ampliaciones	3	30 €	90 €	
6		Referencias bibliográficas	2	30 €		<b>60 €</b>
7		<b>Apéndices</b>				<b>120 €</b>
	1	Glosario	2	30 €	60 €	
	2	Contenido entregado	1	30 €	30 €	
	3	Código fuente	1	30 €	30 €	
8		Memoria del proyecto	2	30 €		<b>60 €</b>
9		<b>Planificación y presupuesto final</b>				<b>255 €</b>
	1	Adaptación WBS	2,5	30 €	75 €	
	2	Registro tiempos totales	1	30 €	30 €	
	3	Presupuesto final	5	30 €	150 €	

<b>Total</b>	<b>1.635 €</b>
--------------	----------------

## 11.2.1.5 Resumen presupuesto de costes final

El resumen del presupuesto, agrupado por partidas, es el siguiente:

Presupuesto de costes	
Partida	Importe
1. Puesta en marcha (Estudios y planificación inicial)	1.500 €
2. Análisis y Diseño del sistema	1.845 €
3. Implementación, pruebas y despliegue	6.180 €
4. Documentación	1.635 €
<b>TOTAL</b>	<b>11.160 €</b>

Se recuerda que en este presupuesto no estaría incluido el IVA.

Si lo comparamos con el presupuesto inicial, en el que se había obtenido un coste de **9600€**, el sobrecoste del proyecto ha sido de **1560€**. Esto se debe a los motivos comentados al comparar la planificación con la inicial, y se puede observar que este sobrecoste se debe en un gran porcentaje a la partida de **Implementación, pruebas y despliegue**, ya que solo en esta existe ya una diferencia de **1260 €**. Por lo tanto, el resto de las partidas se mantienen prácticamente iguales, con variaciones que en ningún caso superan los **500€**.

## Capítulo 12. Referencias Bibliográficas

- [1] Shaumik Daityari «Angular vs React vs Vue: Which Framework to Choose in 2021», 2021. [En línea]. Disponible en <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>. [Accedido: 7-jun-2021].
- [2] Andrés Abraham «Angular vs React vs Vue: ¿Cuál es la mejor opción?», 2019. [En línea]. Disponible en <https://medium.com/somoswigou/angular-vs-react-vs-vue-cu%C3%A1l-es-la-mejor-opci%C3%B3n-941a207951c7>. [Accedido: 7-jun-2021].
- [3] «Transferencia de Estado Representacional», 2021. [En línea]. Disponible en [https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional). [Accedido: 7-jun-2021].
- [4] Red Hat «¿Qué es una API de REST?». [En línea]. Disponible en <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Accedido: 7-jun-2021].
- [5] Universidad de Alicante «Modelo Vista Controlador (MVC)». [En línea]. Disponible en <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Accedido: 7-jun-2021].
- [6] «Modelo-vista-controlador», 2021. [En línea]. Disponible en <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>. [Accedido: 7-jun-2021].
- [7] «Single-page application», 2021. [En línea]. Disponible en [https://es.wikipedia.org/wiki/Single-page\\_application](https://es.wikipedia.org/wiki/Single-page_application). [Accedido: 7-jun-2021].
- [8] «Django (framework)», 2021. [En línea]. Disponible en [https://es.wikipedia.org/wiki/Django\\_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework)). [Accedido: 7-jun-2021].
- [9] «Vue guide». [En línea]. Disponible en <https://es.vuejs.org/v2/guide/>. [Accedido: 7-jun-2021].
- [10] «Vue.js», 2021. [En línea]. Disponible en <https://es.wikipedia.org/wiki/Vue.js>. [Accedido: 7-jun-2021].
- [11] «PyPDF2». [En línea]. Disponible en <https://pypi.org/project/PyPDF2/>. [Accedido: 7-jun-2021].
- [12] Matthew Stamy «Home page for the PyPDF2 project». [En línea]. Disponible en <https://mstamy2.github.io/PyPDF2/>. [Accedido: 7-jun-2021].
- [13] «BootstrapVue Docs». [En línea]. Disponible en <https://bootstrap-vue.org/docs>. [Accedido: 7-jun-2021].

- [14] «Django MVT». [En línea]. Disponible en <https://www.javatpoint.com/django-mvt>. [Accedido: 7-jun-2021].
- [15] «ViewSets». [En línea]. Disponible en <https://www.django-rest-framework.org/api-guide/viewsets/>. [Accedido: 7-jun-2021].
- [16] «Serializers». [En línea]. Disponible en <https://www.django-rest-framework.org/api-guide/serializers/>. [Accedido: 7-jun-2021].
- [17] «Models». [En línea]. Disponible en <https://docs.djangoproject.com/en/3.2/topics/db/models/>. [Accedido: 7-jun-2021].
- [18] Uday Hiwarale «The anatomy of Web Components», 2019. [En línea]. Disponible en <https://medium.com/jspoint/the-anatomy-of-web-components-d6afedb81b37/>. [Accedido: 7-jun-2021].
- [19] «Vuex». [En línea]. Disponible en <https://vuex.vuejs.org/>. [Accedido: 7-jun-2021].
- [20] «Making queries». [En línea]. Disponible en <https://docs.djangoproject.com/es/3.2/topics/db/queries/>. [Accedido: 7-jun-2021].
- [21] «axios Github». [En línea]. Disponible en <https://github.com/axios/axios>. [Accedido: 7-jun-2021].
- [22] «ModelViewSet». [En línea]. Disponible en [https://www.cdrf.co/3.1/rest\\_framework.viewsets/ModelViewSet.html](https://www.cdrf.co/3.1/rest_framework.viewsets/ModelViewSet.html). [Accedido: 7-jun-2021].
- [23] «ModelSerializer». [En línea]. Disponible en [https://www.cdrf.co/3.1/rest\\_framework.serializers/ModelSerializer.html](https://www.cdrf.co/3.1/rest_framework.serializers/ModelSerializer.html). [Accedido: 7-jun-2021].
- [24] «Django Authentication System». [En línea]. Disponible en <https://docs.djangoproject.com/en/3.2/ref/contrib/auth/>. [Accedido: 7-jun-2021].
- [25] «MockFlow». [En línea]. Disponible en <https://www.mockflow.com/>. [Accedido: 7-jun-2021].
- [26] «Django Rest Framework». [En línea]. Disponible en <https://www.django-rest-framework.org/>. [Accedido: 7-jun-2021].
- [27] «Vue Router». [En línea]. Disponible en <https://router.vuejs.org/>. [Accedido: 7-jun-2021].
- [28] «Azure Blob Storage». [En línea]. Disponible en <https://azure.microsoft.com/es-es/services/storage/blobs/>. [Accedido: 7-jun-2021].
- [29] «Heroku Dynos». [En línea]. Disponible en <https://www.heroku.com/dynos>. [Accedido: 7-jun-2021].

- [30] «SQLite», 2021. [En línea]. Disponible en <https://es.wikipedia.org/wiki/SQLite>. [Accedido: 7-jun-2021].
- [31] «PostgreSQL». [En línea]. Disponible en <https://www.postgresql.org/>. [Accedido: 7-jun-2021].
- [32] «Introducing JSON». [En línea]. Disponible en <https://www.json.org/json-en.html>. [Accedido: 7-jun-2021].
- [33] «ECMA-404 The JSON data interchange syntax», 2017. [En línea]. Disponible en <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>. [Accedido: 7-jun-2021].
- [34] «Vue Style Guide». [En línea]. Disponible en <https://vuejs.org/v2/style-guide/>. [Accedido: 7-jun-2021].
- [35] «Python», 2021. [En línea]. Disponible en <https://es.wikipedia.org/wiki/Python>. [Accedido: 7-jun-2021].
- [36] «TIOBE Index», 2021. [En línea]. Disponible en <https://www.tiobe.com/tiobe-index/>. [Accedido: 7-jun-2021].
- [37] «General Purpose Programming Language», 2021. [En línea]. Disponible en [https://en.wikipedia.org/wiki/General-purpose\\_programming\\_language](https://en.wikipedia.org/wiki/General-purpose_programming_language). [Accedido: 7-jun-2021].
- [38] «JavaScript», 2021. [En línea]. Disponible en <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Accedido: 7-jun-2021].
- [39] «HTML: Lenguaje de etiquetas de hipertexto», 2021. [En línea]. Disponible en <https://developer.mozilla.org/es/docs/Web/HTML>. [Accedido: 7-jun-2021].
- [40] «CSS», 2021. [En línea]. Disponible en <https://developer.mozilla.org/es/docs/Web/css>. [Accedido: 7-jun-2021].
- [41] «Visual Studio Code». [En línea]. Disponible en <https://code.visualstudio.com/>. [Accedido: 7-jun-2021].
- [42] Sean Wheeler «¿Qué es Powershell?», 2021. [En línea]. Disponible en <https://docs.microsoft.com/es-es/powershell/scripting/overview?view=powershell-7.1>. [Accedido: 7-jun-2021].
- [43] «Git». [En línea]. Disponible en <https://git-scm.com/>. [Accedido: 7-jun-2021].
- [44] «POSTMAN». [En línea]. Disponible en <https://www.postman.com/>. [Accedido: 7-jun-2021].
- [45] «Browser Market Share», 2021. [En línea]. Disponible en <https://netmarketshare.com/browser-market-share.aspx>. [Accedido: 7-jun-2021].



- [46] «Blob». [En línea]. Disponible en <https://developer.mozilla.org/es/docs/Web/API/Blob>. [Accedido: 7-jun-2021].
- [47] «SQLite on Heroku». [En línea]. Disponible en <https://devcenter.heroku.com/articles/sqlite3>. [Accedido: 7-jun-2021].
- [48] «documentation.js Github»,2021. [En línea]. Disponible en <https://github.com/documentationjs/documentation>. [Accedido: 7-jun-2021].
- [49] Alex Medina «Habilitar la ejecución de scripts para Powershell». [En línea]. Disponible en <https://www.alexmedina.net/habilitar-la-ejecucion-de-scripts-para-powershell/>. [Accedido: 7-jun-2021].
- [50] Sammar Abbas «Deploy Django Project on HEROKU». [En línea]. Disponible en <https://github.com/8sagh8/DjangoRestApi-part1-youtube-project/blob/main/README.md>. [Accedido: 7-jun-2021].
- [51] «Python PIP». [En línea]. Disponible en [https://www.w3schools.com/python/python\\_pip.asp](https://www.w3schools.com/python/python_pip.asp). [Accedido: 7-jun-2021].
- [52] «PIP Project Description». [En línea]. Disponible en <https://pypi.org/project/pip/>. [Accedido: 7-jun-2021].
- [53] «npm», 2020. [En línea]. Disponible en <https://es.wikipedia.org/wiki/Npm>. [Accedido: 7-jun-2021].
- [54] «npmDocs». [En línea]. Disponible en <https://docs.npmjs.com/>. [Accedido: 7-jun-2021].
- [55] «pyPDF2 Documentation». [En línea]. Disponible en <https://pythonhosted.org/PyPDF2/>. [Accedido: 7-jun-2021].
- [56] «Django Azure Storage». [En línea]. Disponible en <https://django-storages.readthedocs.io/en/latest/backends/azure.html>. [Accedido: 7-jun-2021].
- [57] «webpack». [En línea]. Disponible en <https://webpack.js.org/>. [Accedido: 7-jun-2021].
- [58] «Python Regular packages». [En línea]. Disponible en <https://docs.python.org/3/reference/import.html#regular-packages>. [Accedido: 7-jun-2021].

# Capítulo 13. Apéndices

## 13.1 Glosario

- **API Rest:** Interfaz de programación de aplicaciones que sigue el estilo de arquitectura REST.
- **Azure:** Servicio de computación en la nube propiedad de Microsoft.
- **Backend:** Servidor en una aplicación web, encargado de realizar el procesamiento de los datos que recibe desde el *frontend* a través de alguna interfaz.
- **Django:** Framework de Python para el desarrollo web.
- **Django Rest Framework:** Adaptación del *framework* Django para la construcción de APIs Rest.
- **Entorno virtual python:** Forma de gestionar paquetes de Python almacenándolos en espacios aislados.
- **Framework:** Marco de trabajo que establece una organización y estructura para facilitar el desarrollo de un producto software.
- **Frontend:** Dentro de una aplicación web es la parte que se muestra al usuario y con la que interactúa, enviando datos y recuperándolos del *backend*.
- **Git:** Sistema de control de versiones distribuido.
- **GitHub:** Plataforma que almacena proyectos que emplean el sistema de control de versiones Git.
- **Heroku:** Servicio de computación en la nube que permite desplegar aplicaciones de una forma simple.
- **HTTP:** Protocolo de transferencia de hipertexto. Define la sintaxis y semántica que emplean sistemas para comunicarse en internet.
- **Internacionalización:** Proceso de adaptar el software para que este sea usable en distintos lugares para usuarios que empleen distintos idiomas.
- **JavaScript:** Lenguaje de programación multiparadigma, dialecto de ECMAScript. Se trata de un lenguaje interpretado.
- **Node.js:** Entorno de ejecución para el lenguaje JavaScript.
- **Npm:** Sistema de gestión de paquetes integrado en Node.js.
- **ORM:** Object-Relational mapping. Técnica empleada para el mapeo de datos entre una aplicación orientada a objetos y una base de datos relacional.
- **PIP:** Sistema de gestión de paquetes de Python.
- **PyPDF2:** Librería de Python para la manipulación de ficheros PDF.
- **Python:** Lenguaje de programación de propósito general, multiparadigma e interpretado.
- **SGBD:** Iniciales de Sistema Gestor de Bases de Datos.
- **SPA:** Single Page Application. Aplicación web en la que todo el contenido estático es cargado al principio, encontrándose todo este contenido en una única página.
- **Stack tecnológico:** Conjunto de tecnologías y servicios empleados en la construcción de un producto software.

- **MVC:** Iniciales de Modelo Vista Controlador. Patrón arquitectónico centrado en la separación de responsabilidades.
- **Vue:** Framework para la realización de aplicaciones web de tipo SPA.
- **WBS:** *Work Breakdown Structure* en inglés o Estructura de Desglose del Trabajo en español. Se trata de la representación de la descomposición de tareas a realizar en un proyecto siguiendo una jerarquía.
- **Webpack:** Empaquetador de módulos JavaScript para permitir que se puedan usar correctamente en un navegador.

## 13.2 Contenido entregado en el archivo adjunto

Se recogen en la siguiente tabla los contenidos entregados junto a esta documentación en un documento zip.

Directorio	Contenido
<b>./ Directorio raíz</b>	Contiene los directorios Planificación y Presupuesto, Diagramas y PDFTools.
<b>./ Planificación y Presupuesto</b>	Incluye los archivos de la planificación final e inicial realizados con la herramienta Microsoft Project y los cálculos para los presupuestos, en documentos Excel.
<b>./Diagramas</b>	Contiene los distintos diagramas que se han presentado en esta documentación.
<b>./PDFTools</b>	Directorio que contiene las carpetas con el código de ambos subsistemas.
<b>./PDFTools/frontend</b>	Almacena el código fuente, documentación y ficheros de configuración del <i>frontend</i> de la aplicación. La explicación detallada de los contenidos se encuentra en el apartado <i>Estructura de directorios del frontend</i> .
<b>./PDFTools/backend</b>	Almacena el código fuente, documentación y ficheros de configuración del <i>backend</i> de la aplicación. La explicación detallada de los contenidos se encuentra en el apartado <i>Estructura de directorios del backend</i> .

## 13.3 Código fuente

Como se indica en el apartado anterior, el código fuente de ambos subsistemas se encuentra dentro del directorio PDFTools entregado en el fichero adjunto. En los apartados *Estructura de*

*directorios del frontend* y *Estructura de directorios del backend* se encuentra el detalle de los contenidos de cada uno de ellos.