



## TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

# drawercloud

---

Aplicación web orientada al almacenamiento de archivos en la nube

### Autor

José Manuel Rejón Santiago (alumno)

### Director

Prof. Dr. José María Guirao Miras (tutor)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, Septiembre de 2017









# drawercloud

---

Aplicación web orientada al almacenamiento de archivos en la nube.

**Autor**

José Manuel Rejón Santiago (alumno)

**Director**

Prof. Dr. José María Guirao Miras

## **drawercloud: aplicación web orientada al almacenamiento de archivos en la nube**

José Manuel Rejón Santiago (alumno)

**Palabras clave:** aplicación web, cloud, pc, tablet, smartphone, servidor web

### **Resumen**

**drawercloud** es una aplicación web para el alojamiento de archivos en la nube (cloud). Los usuarios podrán realizar diversas tareas sobre sus archivos, como por ejemplo: Añadir/Eliminar archivos en la nube, reproducir archivos multimedia, organizar archivos en directorios, compartir archivos, descargar en un dispositivo (pc, tablet, smartphone)... De este modo, conseguiremos tener nuestros archivos almacenados en un servidor web, ahorrando espacio en nuestros dispositivos y con la seguridad de tener nuestro material a salvo en otro lugar.

# **drawercloud: aplicación web orientada al almacenamiento de archivos en la nube**

José Manuel Rejón Santiago (student)

**Keywords:** web application, cloud, pc, tablet, smartphone, web server

## **Abstract**

drawecloud is a web application for file hosting in the cloud. The users can work with their files, for example: add/delete files in the cloud, play multimedia files, organize the files in different directories, share files, download to devices (pc, tablet, smartphone)... With this, we'll achieve the task of having all our files stored in a web server, saving space in our devices and with the security of having everything safe in another place.





---

Yo, **José Manuel Rejón Santiago**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 20077299R, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: José Manuel Rejón Santiago

Granada a 12 de Septiembre de 2017.



---

D. **Prof. Dr. José María Guirao Miras (tutor)**, Profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *drawercloud, aplicación web orientada al almacenamiento de archivos en la nube*, ha sido realizado bajo su supervisión por **José Manuel Rejón Santiago (alumno)**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 12 de Septiembre de 2017.

**El tutor:**

**Prof. Dr. José María Guirao Miras**



# Agradecimientos

En primer lugar quiero mostrar mi agradecimiento a mi tutor José María Guirao Miras por darme la oportunidad de realizar este proyecto y toda la ayuda y consejos recibidos.

También agradecer a mi familia y amigos/as que han estado apoyándome durante todo el desarrollo del proyecto como a lo largo de estos años de carrera.

Por último y no menos importante a mis compañeros durante esta experiencia, un gran apoyo siempre tanto dentro como fuera de clase.



# Índice general

<b>1. Introducción</b>	<b>17</b>
1.1. Motivación . . . . .	18
<b>2. Análisis</b>	<b>19</b>
2.1. Estudio de mercado . . . . .	19
2.1.1. Dropbox [13] . . . . .	19
2.1.2. Google Drive [14] . . . . .	19
2.1.3. Mega [15] . . . . .	20
2.1.4. OneDrive [16] . . . . .	20
2.2. Objetivos del sistema . . . . .	21
2.3. Requisitos funcionales del sistema . . . . .	21
2.4. Requisitos no funcionales del sistema . . . . .	23
2.5. Requisitos de información . . . . .	24
2.6. Casos de uso . . . . .	24
2.6.1. Descripción de los actores . . . . .	24
2.6.2. Diagramas del modelo de casos de uso . . . . .	25
2.6.3. Descripción de los casos de uso . . . . .	27
2.6.4. Diagrama de clases del diseño . . . . .	41
<b>3. Diseño</b>	<b>43</b>
3.1. Backend . . . . .	43
3.1.1. Patrón Modelo-Vista-Controlador (MVC) . . . . .	43
3.1.2. Django . . . . .	44
3.1.3. Sistema gestor de base de datos . . . . .	45
3.2. Frontend . . . . .	45
3.2.1. jQuery . . . . .	45
3.2.2. AJAX . . . . .	46
3.2.3. Twitter Bootstrap . . . . .	46
<b>4. Implementación</b>	<b>47</b>
4.1. Entorno de trabajo . . . . .	47
4.2. Estructura del proyecto . . . . .	48
4.2.1. Fichero de configuración settings.py . . . . .	48

4.2.2.	Fichero de configuración urls.py . . . . .	49
4.2.3.	Fichero views.py . . . . .	49
4.2.4.	Fichero models.py . . . . .	49
4.2.5.	Fichero forms.py . . . . .	50
4.2.6.	Fichero urls.py . . . . .	50
4.2.7.	Directorio templates . . . . .	51
4.2.8.	Directorio static . . . . .	51
<b>5.</b>	<b>Pruebas</b>	<b>53</b>
5.1.	Resultados de las pruebas con usuarios . . . . .	53
<b>6.</b>	<b>Conclusiones y trabajos futuros</b>	<b>55</b>
6.1.	Conclusiones . . . . .	55
6.1.1.	Conocimientos adquiridos . . . . .	55
6.2.	Trabajos futuros . . . . .	56
<b>7.</b>	<b>Manual de usuario</b>	<b>57</b>
7.1.	Acceso a la aplicación . . . . .	57
7.2.	Página principal - Documentos . . . . .	58
7.2.1.	Opciones para archivos . . . . .	60
7.2.2.	Opciones para directorios . . . . .	60



# Índice de figuras

2.1. Diagrama de casos de uso. Gestión de usuarios. . . . .	25
2.2. Diagrama de casos de uso. Gestión de archivos. . . . .	26
2.3. Diagrama de casos de uso. Gestión de archivos. . . . .	26
2.4. Diagrama de casos de uso. Gestión de archivos. . . . .	27
2.5. Diagrama de clases del diseño. . . . .	42
3.1. Diagrama Model-View-Controller. . . . .	44
3.2. Diagrama Model-View-Template. . . . .	45
7.1. Acceso a la aplicación. Página de login . . . . .	57
7.2. Acceso a la aplicación. Página de registro . . . . .	58
7.3. Botones disponibles para cambiar la vista . . . . .	58
7.4. Documentos. Aspecto de la página principal . . . . .	59
7.5. Documentos. Aspecto de la página principal . . . . .	59
7.6. Documentos. Opciones de la página Documentos . . . . .	60



# Índice de cuadros

2.1. Curso normal de CU-1. Alta de usuarios. . . . .	27
2.2. Curso alterno de CU-1. Alta de usuarios. . . . .	27
2.3. Curso normal de CU-2. Baja de usuarios. . . . .	28
2.4. Curso normal de CU-3. Identificación de usuarios. . . . .	28
2.5. Curso alterno de CU-3. Identificación de usuarios. . . . .	28
2.6. Curso normal de CU-4. Subir archivo. . . . .	29
2.7. Curso alterno de CU-4. Subir archivo. . . . .	29
2.8. Curso normal de CU-5. Visualizar archivo. . . . .	30
2.9. Curso alterno de CU-5. Visualizar archivo. . . . .	30
2.10. Curso normal de CU-6. Eliminar archivo. . . . .	30
2.11. Curso normal de CU-7. Descargar archivo. . . . .	31
2.12. Curso normal de CU-8. Compartir archivo. . . . .	31
2.13. Curso alterno de CU-8. Compartir archivo. . . . .	31
2.14. Curso normal de CU-9. Dejar de compartir archivo. . . . .	32
2.15. Curso normal de CU-10. Cambiar nombre de un archivo. . . . .	32
2.16. Curso alterno de CU-10. Cambiar nombre de un archivo. . . . .	32
2.17. Curso normal de CU-11. Copiar archivo. . . . .	33
2.18. Curso alterno de CU-11. Copiar archivo. . . . .	33
2.19. Curso normal de CU-12. Mover archivo. . . . .	33
2.20. Curso alterno de CU-12. Mover archivo. . . . .	34
2.21. Curso normal de CU-13. Añadir archivo a favoritos. . . . .	34
2.22. Curso normal de CU-14. Eliminar archivo de favoritos. . . . .	34
2.23. Curso normal de CU-15. Añadir archivo a mi nube. . . . .	35
2.24. Curso normal de CU-16. Añadir directorio. . . . .	35
2.25. Curso alterno de CU-16. Añadir directorio. . . . .	35
2.26. Curso normal de CU-17. Eliminar directorio. . . . .	36
2.27. Curso normal de CU-18. Visualizar directorio. . . . .	36
2.28. Curso normal de CU-19. Cambiar nombre de un directorio. . . . .	37
2.29. Curso alterno de CU-19. Cambiar nombre de un directorio. . . . .	37
2.30. Curso normal de CU-20. Copiar directorio. . . . .	37
2.31. Curso alterno de CU-20. Copiar directorio. . . . .	37
2.32. Curso normal de CU-21. Mover directorio. . . . .	38
2.33. Curso alterno de CU-21. Mover directorio. . . . .	38

2.34. Curso normal de CU-22. Añadir grupo de trabajo. . . . .	39
2.35. Curso normal de CU-23. Salir del grupo de trabajo. . . . .	39
2.36. Curso alterno de CU-23. Salir del grupo de trabajo. . . . .	39
2.37. Curso normal de CU-24. Ver grupo de trabajo. . . . .	40
2.38. Curso normal de CU-25. Añadir participante al grupo de tra- bajo. . . . .	40
2.39. Curso alterno de CU-25. Añadir participante al grupo de tra- bajo. . . . .	40
2.40. Curso normal de CU-26. Ver participantes del grupo de trabajo.	41

# Capítulo 1

## Introducción

El objetivo de este proyecto es el desarrollo de una aplicación web que proporcione al usuario la posibilidad de almacenar sus archivos en la nube. La aplicación web cuenta con opciones como el almacenamiento en un lugar seguro, la organización por carpetas, la posibilidad de crear grupos de trabajo donde compartir los archivos con compañeros, entre otras más.

Para el desarrollo de la web usaremos **Django** [1], que es un framework para aplicaciones web gratuito y open source, escrito en **Python** [2]. La aplicación web contará con un **responsive design** para ser adaptada a otros dispositivos (como smartphones o tablets). En el lado del servidor, se usará la tecnología proporcionada por Amazon Web Services o Google.

Para el aspecto gráfico usamos la tecnología que nos ofrece **Twitter Bootstrap** [3], un framework **HTML** [4], **CSS** [5] y **JavaScript** [6] para lograr un diseño web adaptable (responsive design) entre los distintos dispositivos que puedan hacer uso de la aplicación web. **Ajax** [7] y **jQuery** [8] serán las herramientas elegidas para ofrecer una mejor interacción entre el usuario y la aplicación web.

La base de datos será gestionada con **MongoDB** [9], que es una base de datos orientada a documentos estilo JSON (incluye información del tipo de dato).

drawercloud hace uso de **EC2** [10] (Amazon Virtual Servers), que nos permite alojar la web gratuitamente durante un año y poder acceder a ella desde cualquier dispositivo que se conecte a la red. Como servidor web usamos **NGINX** [11] y **Gunicorn** [12] que es un WSGI de Python.

Este proyecto está bajo la licencia de software libre, lo que significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar,

modificar y mejorar el software.

## 1.1. Motivación

Unos de los problemas que surgen en el uso de nuestros dispositivos es el almacenamiento. Es cierto que el precio del **gigabyte** es cada vez más barato e incluso más rápido (como por ejemplo en los discos duros SSD, que están dotados de un gran incremento de velocidad, además de consumir menos energía) pero, aunque en un ordenador el almacenamiento puede no ser un problema, si que lo es si hablamos de dispositivos móviles como smartphones o tablets. A día de hoy, encontramos dispositivos que no permiten la insercción de tarjetas de almacenamiento flash, lo cual nos obliga a conformarnos con el espacio que el fabricante pone a nuestra disposición (o a comprar el modelo con más capacidad, pero obviamente más caro). **drawercloud** viene a poner solución a este problema, ofreciendo un espacio personal en un servidor web dedicado para dicha aplicación, de modo que nuestros datos permanecen en tal servidor sin ocupar memoria en nuestros dispositivos.

¿Significa esto que solo es una solución para dispositivos que cuentan con un espacio muy limitado? No, ya que otro problema que puede surgirnos con mucha facilidad es la pérdida de información por diversas causas (disco duro roto, pérdida del dispositivo, virus, etc) y para ello este proyecto también ofrece una solución a dicho problema. Podemos almacenar nuestros archivos tanto en el disco duro propio del dispositivo en cuestión, como en el espacio ofrecido en el servidor web, de modo que si nos vemos afectados por alguno de los motivos mencionados, o bien por otros que afectan a la desaparición de nuestra información, tengamos una copia en otro lugar y no perdamos nuestros archivos.

Otro caso en el que esta aplicación resulta interesante es que podemos disponer de nuestros datos en cualquier parte. A modo de ejemplo, podemos pensar en un archivo que estamos modificando en casa y en nuestro puesto de trabajo. Almacenando dicho archivo en la nube nos tenemos que olvidar de mover la información a través de, por ejemplo, un pen drive o de tener una copia en casa y otra en la oficina y andar actualizando ambas cada vez que queramos trabajar. Simplemente entraremos con nuestra cuenta a **drawercloud** desde cualquier dispositivo y tendremos acceso al archivo en cuestión. Además, proporciona un método para compartir archivos y crear grupos de trabajo, de modo que si sobre ese archivo están trabajando, por ejemplo, 3 personas más, podrán ver las modificaciones que hemos realizado directamente sobre el archivo alojado en el servidor web.

## Capítulo 2

# Análisis

### 2.1. Estudio de mercado

Antes de comenzar el desarrollo del proyecto es recomendable realizar un estudio de mercado. Dicho estudio nos ofrecerá una visión de las virtudes y carencias de los productos que actualmente ofrecen un servicio similar y así, saber en que punto nos encontraremos respecto a ellos. Ésto nos dará la ventaja de poder realzar los puntos fuertes de nuestro proyecto frente a la competencia. Veamos algunos ejemplos de aplicaciones que ofrecen el mismo servicio:

#### 2.1.1. Dropbox [13]

Dropbox es la aplicación para almacenar archivos en la nube posiblemente más conocida y usada, siendo ésta una de las primeras en ofrecernos este tipo de servicios. En su versión gratuita, los usuarios de esta aplicación disponen de 2GB, pero se puede ampliar hasta 16GB realizando algunas sencillas tareas (como subir fotos desde el móvil o recomendar a amigos). Por 9,99 €/mes se dispondrá de 1TB de espacio disponible. **Equipo** y **Paper** son dos opciones que ofrece Dropbox para trabajar en equipo, siendo éstas dos puntos fuertes de este servicio. También su aplicación de escritorio es un punto muy a favor, dando la posibilidad al usuario de sincronizar sus datos más rápidamente (aunque puede ralentizar el arranque del pc mientras sincroniza).

#### 2.1.2. Google Drive [14]

Como era de esperar, no podía faltar Google en esta lista. En su versión gratuita ofrece 15GB de almacenamiento (sin necesidad de realizar tareas

para ampliar como hace dropbox). Para ampliar el espacio, Google Drive ofrece dos posibilidades:

1. 100GB por 1,99 €/mes
2. 1TB por 9,99 €/mes

Sin duda, el punto fuerte de Google Drive es la posibilidad de crear archivos (como hojas de texto, hojas de cálculo, formularios, ...) que pueden ser editados por otras personas (si le damos el permiso de compartir) y así trabajar en grupo sobre el mismo documento en la nube, almacenando los cambios automáticamente.

### 2.1.3. Mega [15]

Aunque es más conocido por la piratería, Mega es otra de las soluciones para alojar archivos en la nube. Ofrece hasta 50GB de forma gratuita y en caso de querer más espacio, se pueden usar estas alternativas:

1. 500GB por 9,99 €/mes
2. 2TB por 19,99 €/mes
3. 4TB por 29,99 €/mes

Además de la enorme cantidad de espacio, el alto ancho de banda que ofrece sería un punto a favor de Mega.

### 2.1.4. OneDrive [16]

OneDrive es la solución para almacenamiento en la nube propuesta por Microsoft. Viene preinstalado en Windows 10 y eso permite que los documentos y las fotografías se guarden automáticamente en OneDrive. Ofrece también un apartado de empresas para que un grupo de trabajo pueda colaborar sobre los mismos archivos en tiempo real (permite colaborar con Word, Excel, PowerPoint, OneNote). La versión gratuita de este servicio se ha reducido a 5GB (siendo antes 15GB). Las soluciones de pago son las siguientes:

1. 50GB por 2,00 €/mes
2. 1TB por 7,00 €/mes (incluye Office 365 y la opción de instalar en un PC o Mac y en una tablet y un smartphone)
3. 5TB (1000 GB por usuario) por 10,00 €/mes (incluye Office 365 y la opción de instalar en 5 PC o Mac y en 5 tablets y 5 smartphones)



## 2.2. Objetivos del sistema

El objetivo principal del sistema es poner a disposición del usuario un espacio de almacenamiento en la nube. En dicho espacio, el usuario tendrá la posibilidad de almacenar información de forma organizada, compartir los archivos deseados y tener accesibilidad a éstos desde cualquier plataforma con un navegador web.

Con el fin de ofrecer un sistema fiable a los clientes, este programa deberá cumplir algunas características:

- Disponibilidad: los datos del usuario siempre estarán disponibles.
- Seguridad: los datos deberán estar alojados en un lugar protegido de posibles amenazas.
- Accesibilidad: los datos del usuario siempre estarán disponibles para él y para quien tenga permisos sobre ellos.

Visto el objetivo principal, se listan los objetivos específicos a realizar para la realización de esta aplicación web:

- **OBJ-1:** El sistema deberá almacenar los archivos relativos a cada usuario en su espacio disponible.
- **OBJ-2:** El sistema deberá permitir el acceso a la información solo a los usuarios con los permisos necesarios.
- **OBJ-3:** El sistema deberá ser accesible desde todos los dispositivos dotados de navegador web (ordenador, smartphone, tablet, etc).

## 2.3. Requisitos funcionales del sistema

En esta sección indicamos qué debe hacer el sistema:

- **RF-1: Gestión de usuarios.** En el sistema realizaremos altas, bajas y modificaciones de los usuarios.
  - **RF-1.1: Alta usuario.** El sistema permite realizar el alta de un usuario.
  - **RF-1.2: Baja usuario.** El sistema permite realizar la baja de un usuario.
  - **RF-1.3: Modificación usuario.** El sistema permite realizar una modificación de un usuario.
  - **RF-1.4: Consulta usuario.** El sistema permite realizar una consulta de un usuario en concreto.

■ **RF-2: Gestión de archivos.**

- **RF-2.1: Añadir archivo.** Junto al archivo se añadirá una serie de datos como nombre, fecha de subida, propietario, etc.
- **RF-2.2: Eliminar archivo.** Se eliminará el archivo y toda la información relativa a él.
- **RF-2.3: Visualizar archivo.** Se podrá ver el contenido del archivo que sea posible (como reproducir canciones o ver una imagen).
- **RF-2.4: Descargar archivo.** Descarga el archivo en el dispositivo usado.
- **RF-2.5: Compartir archivo.** Da acceso al archivo (en modo lectura) a otro usuario.
- **RF-2.6: Dejar de compartir archivo.** Retira el acceso al archivo a otro usuario.
- **RF-2.7: Cambiar nombre del archivo.** Permite cambiar el nombre del archivo.
- **RF-2.8: Copiar archivo.** Se creará una copia del archivo.
- **RF-2.9: Mover archivo.** Se moverá el archivo al directorio de destino seleccionado.
- **RF-2.10: Añadir archivo a favoritos.** El archivo pasará a la lista de favoritos.
- **RF-2.10: Eliminar archivo de favoritos.** El archivo dejará la lista de favoritos.
- **RF-2.11: Añadir archivo a mi nube.** El archivo compartido (en modo lectura) pasará a ser un archivo propiedad del usuario.

■ **RF-3: Gestión de directorios.**

- **RF-3.1: Añadir directorio.** Junto al directorio se añadirá una serie de datos como nombre, contenido, propietario y directorio padre.
- **RF-3.2: Eliminar directorio.** Se eliminará el directorio y todos los directorios y archivos que contenga.
- **RF-3.3: Visualizar directorio.** Acceder al contenido del directorio.
- **RF-3.4: Cambiar nombre directorio.** Permite cambiar el nombre del directorio.
- **RF-3.5: Copiar directorio.** Se creará una copia del directorio y todo su contenido.

- **RF-3.6: Mover directorio.** Se moverá el directorio y todo su contenido al directorio de destino seleccionado.
- **RF-4: Gestión de grupos de trabajo.**
  - **RF-4.1: Añadir grupo de trabajo.** Se añade un grupo con un nombre.
  - **RF-4.2: Ver grupo de trabajo.** Visualiza el contenido del grupo de trabajo.
  - **RF-4.3: Salir del grupo de trabajo.** Un usuario sale del grupo de trabajo. Cuando el grupo de trabajo queda sin usuarios, se elimina toda la información relativa a él.
  - **RF-4.4: Añadir participante al grupo de trabajo.** Se añade un participante al grupo con todos los permisos (subir archivo, eliminar archivo, crear carpeta, añadir participante, etc).
  - **RF-4.5: Ver participantes del grupo de trabajo.** Se muestran todos los participantes del grupo.

## 2.4. Requisitos no funcionales del sistema

En esta sección se incluyen restricciones que afectarán a los requisitos funcionales:

- **RNF-1: Seguridad de la información.** Ante un posible fallo del sistema, la información previamente almacenada no debería perderse. En principio esto se garantiza, ya que el sistema y la base de datos son entidades relacionadas pero separadas. Sería buena práctica tener en cuenta la posibilidad de realizar copias de seguridad periódicamente o alguna otra alternativa que permita replicar los datos.
- **RNF-2: Concurrencia.** Tanto administradores como usuarios podrán acceder simultáneamente al sistema, introduciendo información, consultándola o eliminándola.
- **RNF-3: Autenticación.** Es necesario controlar el acceso de los usuarios, de manera que cada uno pueda realizar sólo las tareas sobre su espacio.
- **RNF-4: Legales.** Se deben tener en cuenta ciertos criterios de responsabilidad sobre los datos publicados, como que los datos de un usuario no sean accesibles por otro.
- **RNF-5: Rendimiento.** El sistema debe tener un tiempo de respuesta y aprovechamiento de los recursos óptimo.

## 2.5. Requisitos de información

En esta sección se indica la información que es necesaria almacenar en el sistema:

- **RI-1: Usuarios.** Datos sobre las personas que usarán la aplicación.
  - **Contenido:** nombre de usuario, contraseña, dirección de correo electrónico.
  - **Requisitos asociados:** RF-1, RNF-1, RNF-3.
- **RI-2: Archivos.** Datos sobre los archivos que se alojarán en la aplicación.
  - **Contenido:** archivo, nombre del archivo, tipo del archivo, fecha de subida, propietario, tamaño del archivo, booleano para indicar si es favorito.
  - **Requisitos asociados:** RF-2, RNF-1, RNF-3, RNF-4, RNF-5.
- **RI-3: Directorios.** Datos sobre los directorios que se alojarán en la aplicación.
  - **Contenido:** nombre del directorio, contenido del directorio, propietario.
  - **Requisitos asociados:** RF-3, RNF-1, RNF-3, RNF-4, RNF-5.
- **RI-4: Grupos de trabajo.** Datos sobre los grupos de trabajo de la aplicación.
  - **Contenido:** nombre del grupo de trabajo, usuarios que pertenecen al grupo de trabajo, directorios que pertenecen al grupo de trabajo, archivos que pertenecen al grupo de trabajo.
  - **Requisitos asociados:** RF-4, RNF-1, RNF-4, RNF-5.

## 2.6. Casos de uso

### 2.6.1. Descripción de los actores

- **DA-1: Usuario.**
  - **Descripción:** Usuario que accede al sistema para gestionar su espacio personal o sus grupos de trabajo.
  - **Características:** Conocimientos básicos sobre el manejo de algún dispositivo con navegador web.

- **Relaciones:** La relación con otros usuarios del sistema se da, o bien compartiendo un archivo directamente con el usuario, o bien perteneciendo al mismo grupo de trabajo.
- **Referencias:** Puede consultar la base de datos de sus archivos, directorios, grupos de trabajo, información asociada a su perfil y la ayuda.

### 2.6.2. Diagramas del modelo de casos de uso

Los diagramas de casos de uso representan como el usuario se relaciona con el sistema para usar sus funciones. A continuación vemos dichos diagramas:

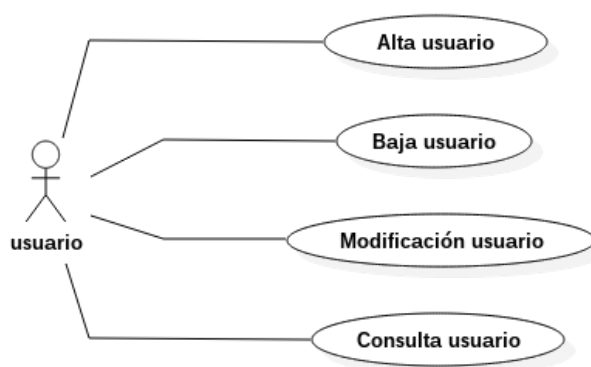


Figura 2.1: Diagrama de casos de uso. Gestión de usuarios.

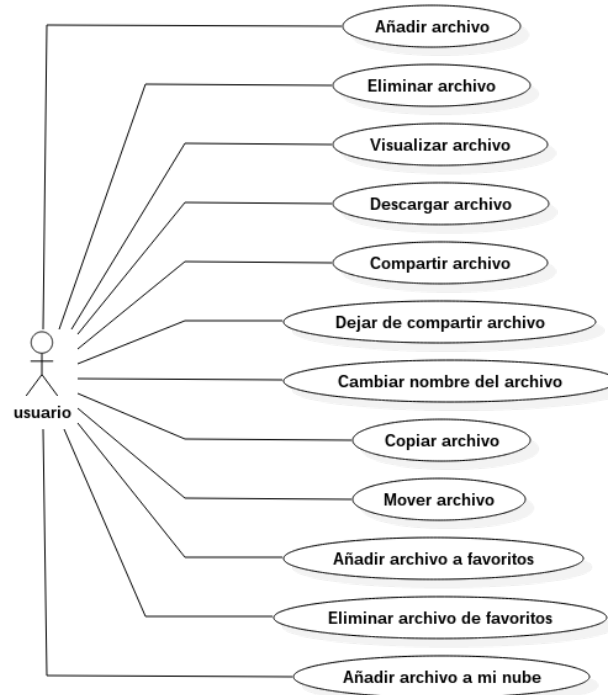


Figura 2.2: Diagrama de casos de uso. Gestión de archivos.

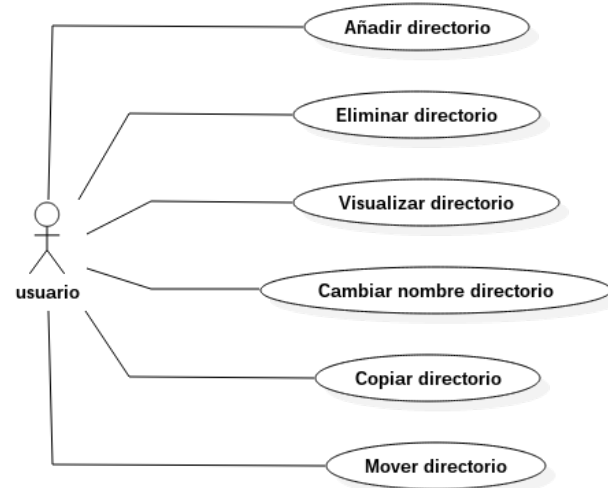


Figura 2.3: Diagrama de casos de uso. Gestión de directorios.

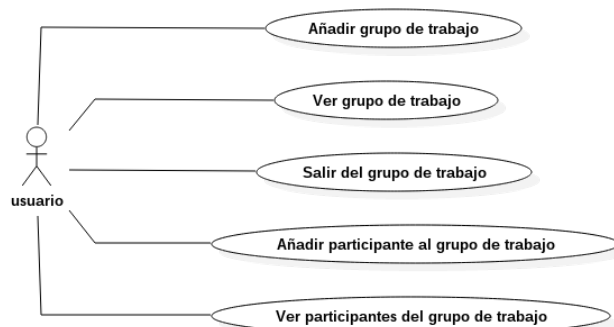


Figura 2.4: Diagrama de casos de uso. Gestión de archivos.

### 2.6.3. Descripción de los casos de uso

A continuación se exponen los casos de uso principales del sistema, correspondientes a las acciones básicas del usuario con la plataforma, definiendo el comportamiento normal y alternativo de cada acción.

#### ■ CU-1: Alta de usuarios.

- **Descripción:** el usuario se registra con sus datos en el sistema.
- **Precondición:** el username no se encuentra registrado en el sistema.
- **Postcondición:** el usuario queda registrado en el sistema con un identificador único.

Curso normal			
Actor		Sistema	
1	Usuario: inicia la acción de registro en el sistema.		
		2a	Dar de alta al usuario en el sistema.
		3	El usuario queda dado de alta en el sistema y puede empezar a usar la aplicación.

Cuadro 2.1: Curso normal de CU-1. Alta de usuarios.

Curso alterno	
2b	Si ya existe el usuario o se da algún otro tipo de error, se notifica al usuario y no se produce el registro.

Cuadro 2.2: Curso alterno de CU-1. Alta de usuarios.

■ **CU-2: Baja de usuarios.**

- **Descripción:** el usuario se da de baja en el sistema.
- **Precondición:** el usuario ha iniciado sesión en el sistema.
- **Postcondición:** el usuario queda eliminado en el sistema.

Curso normal			
Actor		Sistema	
1	Usuario: inicia la acción de baja en el sistema.		
		2	Elimina todos los archivos/-directorios/grupos de trabajo del usuario y los datos relativos a éste.
		3	El usuario queda dado de baja en el sistema y no puede usar la aplicación.

Cuadro 2.3: Curso normal de CU-2. Baja de usuarios.

■ **CU-3: Identificación de usuarios.**

- **Descripción:** el usuario accede al sistema con su username y contraseña.
- **Precondición:** el usuario debe estar registrado.
- **Postcondición:** el usuario inicia sesión en el sistema.

Curso normal			
Actor		Sistema	
1	Usuario: inicia sesión.		
		2a	Comenzar la sesión para el usuario.
		3	El usuario puede empezar a usar la aplicación.

Cuadro 2.4: Curso normal de CU-3. Identificación de usuarios.

Curso alternativo	
2b	Si el usuario no coincide o la contraseña es incorrecta, se notifica al usuario y no se inicia sesión.

Cuadro 2.5: Curso alternativo de CU-3. Identificación de usuarios.



■ **CU-4: Subir archivo.**

- **Descripción:** el usuario añade un archivo a su espacio.
- **Precondición:** el usuario ha iniciado sesión en el sistema.
- **Postcondición:** el archivo se añade a la cuenta del usuario.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para subir.		
		2a	El archivo se almacena en la cuenta del usuario y en el directorio desde el que el usuario se encuentre.
		3a	El archivo queda vinculado al usuario.

Cuadro 2.6: Curso normal de CU-4. Subir archivo.

Curso alterno	
2b	Si en la subida del archivo se produce un error, se notifica al usuario y no se continua.
3b	Si se trata de un archivo multimedia (música, imagen, vídeo) se añadirá a uno de los apartados disponibles en la sección Multimedia (Música, Imágenes, Vídeo).

Cuadro 2.7: Curso alterno de CU-4. Subir archivo.

■ **CU-5: Visualizar archivo.**

- **Descripción:** el usuario selecciona un archivo para mostrar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario tiene permiso de lectura sobre el archivo.
- **Postcondición:** se muestra el archivo según el formato.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para mostrar.		
		2a	El archivo seleccionado se muestra según el tipo de contenido del archivo (reproducir si es canción, mostrar si es imagen, ...).

Cuadro 2.8: Curso normal de CU-5. Visualizar archivo.

Curso alterno	
2b	Si el archivo no se puede mostrar o reproducir, automáticamente dará la opción de Descargar.

Cuadro 2.9: Curso alterno de CU-5. Visualizar archivo.

#### ■ CU-6: Eliminar archivo.

- **Descripción:** el usuario selecciona un archivo para eliminar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo.
- **Postcondición:** el archivo queda eliminado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para eliminar.		
		2	Se eliminan el archivo y todos sus datos almacenados en la base de datos.

Cuadro 2.10: Curso normal de CU-6. Eliminar archivo.

#### ■ CU-7: Descargar archivo.

- **Descripción:** el usuario selecciona un archivo para descargar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario tiene permiso de lectura sobre el archivo.
- **Postcondición:** se descarga el archivo en el dispositivo usado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para descargar.		
		2	El archivo seleccionado se descarga en el dispositivo usado.

Cuadro 2.11: Curso normal de CU-7. Descargar archivo.

■ **CU-8: Compartir archivo.**

- **Descripción:** el usuario selecciona un archivo y otro usuario para compartir.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo.
- **Postcondición:** el otro usuario tiene permiso de lectura sobre el archivo.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para compartir.		
2a	Usuario: selecciona un usuario para compartir.		
		3	El archivo seleccionado se comparte con el usuario elegido y dicho usuario recibe permiso de lectura sobre el archivo en cuestión.

Cuadro 2.12: Curso normal de CU-8. Compartir archivo.

Curso alterno	
2b	Si el usuario intenta compartir consigo mismo, recibe un mensaje de error y no se continua con el proceso.
2c	Si el usuario intenta compartir con un usuario que no existe, recibe un mensaje de error y no se continua con el proceso.

Cuadro 2.13: Curso alterno de CU-8. Compartir archivo.

■ **CU-9: Dejar de compartir archivo.**

- **Descripción:** el usuario selecciona un archivo para dejar de compartir.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El archivo debe estar compartido.
- **Postcondición:** el usuario destino en la relación de compartición deja de tener permiso de lectura sobre el archivo.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para dejar de compartir.		
		2	Se elimina la relación de compartición y el usuario destino deja de tener permiso de lectura sobre el archivo.

Cuadro 2.14: Curso normal de CU-9. Dejar de compartir archivo.

▪ **CU-10: Cambiar nombre de un archivo.**

- **Descripción:** el usuario selecciona un archivo para cambiar el nombre.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo.
- **Postcondición:** el nombre del archivo queda cambiado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para cambiar nombre.		
2	Usuario: inserta un nuevo nombre para el archivo.		
		3a	Se cambia el nombre del archivo por el nuevo introducido.

Cuadro 2.15: Curso normal de CU-10. Cambiar nombre de un archivo.

Curso alternativo	
3b	Si el nuevo nombre ya existe en el directorio, se añade al principio del nombre la palabra «copia».

Cuadro 2.16: Curso alternativo de CU-10. Cambiar nombre de un archivo.

■ **CU-11: Copiar archivo.**

- **Descripción:** el usuario selecciona un archivo para copiar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo.
- **Postcondición:** se crea una copia del archivo en el directorio seleccionado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para copiar.		
2	Usuario: selecciona un directorio para copiar el archivo.		
		3a	Se copia el archivo en el directorio introducido.

Cuadro 2.17: Curso normal de CU-11. Copiar archivo.

Curso alterno	
3b	Si el nombre del archivo copiado existe en el directorio destino, se añade al principio del nombre la palabra «copia».

Cuadro 2.18: Curso alterno de CU-11. Copiar archivo.

■ **CU-12: Mover archivo.**

- **Descripción:** el usuario selecciona un archivo para mover.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo.
- **Postcondición:** se desplaza el archivo al directorio seleccionado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para mover.		
2	Usuario: selecciona un directorio para mover el archivo.		
		3a	Se mueve el archivo al directorio introducido.

Cuadro 2.19: Curso normal de CU-12. Mover archivo.

Curso alterno	
3b	Si el nombre del archivo movido existe en el directorio destino, se añade al principio del nombre la palabra «copia».

Cuadro 2.20: Curso alterno de CU-12. Mover archivo.

■ **CU-13: Añadir archivo a favoritos.**

- **Descripción:** el usuario selecciona un archivo para añadir a favoritos.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo. El archivo no está marcado como favorito.
- **Postcondición:** el archivo queda marcado como favorito.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para añadir a favoritos.		
		2	Se añade al archivo la condición de favorito.

Cuadro 2.21: Curso normal de CU-13. Añadir archivo a favoritos.

■ **CU-14: Eliminar archivo de favoritos.**

- **Descripción:** el usuario selecciona un archivo para eliminar de favoritos.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del archivo. El archivo está marcado como favorito.
- **Postcondición:** el archivo queda eliminado como favorito.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para eliminar de favoritos.		
		2	Se elimina del archivo la condición de favorito.

Cuadro 2.22: Curso normal de CU-14. Eliminar archivo de favoritos.

■ **CU-15: Añadir archivo a mi nube.**

- **Descripción:** el usuario selecciona un archivo para añadir a su cuenta.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario está recibiendo el archivo como compartido.
- **Postcondición:** el usuario tiene su copia del archivo y es propietario de ella.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un archivo para añadir a su cuenta.		
		2	Se crea una copia en la cuenta del usuario y se convierte en propietario de ella.

Cuadro 2.23: Curso normal de CU-15. Añadir archivo a mi nube.

#### ■ CU-16: Añadir directorio.

- **Descripción:** el usuario añade un directorio.
- **Precondición:** el usuario ha iniciado sesión en el sistema.
- **Postcondición:** el directorio se añade a la cuenta del usuario.

Curso normal			
Actor		Sistema	
1	Usuario: crea un nuevo directorio añadiendo un nombre.		
		2a	Se crea el nuevo directorio.

Cuadro 2.24: Curso normal de CU-16. Añadir directorio.

Curso alterno	
2b	Si el nombre del nuevo directorio ya existe en el directorio actual, se añade al principio del nombre la palabra «copia».

Cuadro 2.25: Curso alterno de CU-16. Añadir directorio.

#### ■ CU-17: Eliminar directorio.

- **Descripción:** el usuario selecciona un directorio para eliminar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del directorio.

- **Postcondición:** el directorio queda eliminado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un directorio para eliminar.		
		2	Se eliminan el directorio y todo su contenido (directorios y archivos contenidos en éste).

Cuadro 2.26: Curso normal de CU-17. Eliminar directorio.

■ **CU-18: Visualizar directorio.**

- **Descripción:** el usuario selecciona un directorio para visualizar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del directorio.
- **Postcondición:** el usuario accede al interior del directorio.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un directorio para visualizar.		
		2	Se muestran los archivos y/o directorios contenidos en el directorio seleccionado.

Cuadro 2.27: Curso normal de CU-18. Visualizar directorio.

■ **CU-19: Cambiar nombre de un directorio.**

- **Descripción:** el usuario selecciona un directorio para cambiar el nombre.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del directorio.
- **Postcondición:** el nombre del directorio queda cambiado.



Curso normal			
Actor		Sistema	
1	Usuario: selecciona un directorio para cambiar nombre.		
2	Usuario: inserta un nuevo nombre para el directorio.		
		3a	Se cambia el nombre del directorio por el nuevo introducido.

Cuadro 2.28: Curso normal de CU-19. Cambiar nombre de un directorio.

Curso alterno	
3b	Si el nuevo nombre ya existe en el directorio actual, se añade al principio del nombre la palabra «copia».

Cuadro 2.29: Curso alterno de CU-19. Cambiar nombre de un directorio.

■ **CU-20: Copiar directorio.**

- **Descripción:** el usuario selecciona un directorio para copiar.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del directorio.
- **Postcondición:** se crea una copia del directorio y todo su contenido en el directorio seleccionado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un directorio para copiar.		
2	Usuario: selecciona un directorio destino para copiar el directorio.		
		3a	Se copia el directorio y todo su contenido en el directorio destino introducido.

Cuadro 2.30: Curso normal de CU-20. Copiar directorio.

Curso alterno	
3b	Si el nombre del directorio copiado existe en el directorio destino, se añade al principio del nombre la palabra «copia».

Cuadro 2.31: Curso alterno de CU-20. Copiar directorio.

■ **CU-21: Mover directorio.**

- **Descripción:** el usuario selecciona un directorio para mover.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario es el propietario del directorio.
- **Postcondición:** se desplaza el directorio y todo su contenido al directorio seleccionado.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un directorio para mover.		
2	Usuario: selecciona un directorio destino para mover el directorio.		
		3a	Se mueve el directorio seleccionado y todo su contenido al directorio destino introducido.

Cuadro 2.32: Curso normal de CU-21. Mover directorio.

Curso alterno	
3b	Si el nombre del directorio movido existe en el directorio destino, se añade al principio del nombre la palabra «copia».

Cuadro 2.33: Curso alterno de CU-21. Mover directorio.

■ **CU-22: Añadir grupo de trabajo.**

- **Descripción:** el usuario crea un nuevo grupo de trabajo.
- **Precondición:** el usuario ha iniciado sesión en el sistema.
- **Postcondición:** el grupo de trabajo queda creado y el usuario se convierte en participante de éste.

Curso normal			
Actor		Sistema	
1	Usuario: crea un grupo de trabajo dándole un nombre.		
		2	Se crea el nuevo grupo de trabajo.
		3	El usuario se añade al grupo como participante.

Cuadro 2.34: Curso normal de CU-22. Añadir grupo de trabajo.

■ **CU-23: Salir del grupo de trabajo.**

- **Descripción:** el usuario sale del grupo de trabajo.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario pertenece al grupo de trabajo.
- **Postcondición:** el usuario pierde el acceso al grupo.

Curso normal			
Actor		Sistema	
1	Usuario: sale del grupo de trabajo.		
		2	El usuario queda eliminado del grupo de trabajo.
		3a	Se actualiza el estado del grupo de trabajo.

Cuadro 2.35: Curso normal de CU-23. Salir del grupo de trabajo.

Curso alterno	
3b	Si el usuario en salir del grupo de trabajo era el último, entonces el grupo se eliminará automáticamente junto a todo su contenido.

Cuadro 2.36: Curso alterno de CU-23. Salir del grupo de trabajo.

■ **CU-24: Ver grupo de trabajo.**

- **Descripción:** el usuario accede a un grupo de trabajo.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario pertenece al grupo de trabajo.
- **Postcondición:** el usuario accede al contenido del grupo de trabajo.

Curso normal			
Actor		Sistema	
1	Usuario: selecciona un grupo de trabajo para visualizar.		
		2	Se muestran los archivos y/o directorios contenidos en el grupo de trabajo seleccionado.

Cuadro 2.37: Curso normal de CU-24. Ver grupo de trabajo.

■ **CU-25: Añadir participante al grupo de trabajo.**

- **Descripción:** el usuario añade a otro usuario al grupo de trabajo.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario pertenece al grupo de trabajo.
- **Postcondición:** el usuario añadido es participante del grupo de trabajo.

Curso normal			
Actor		Sistema	
1a	Usuario: elige un usuario para añadir al grupo de trabajo.		
		2	El usuario añadido se convierte en participante del grupo de trabajo.

Cuadro 2.38: Curso normal de CU-25. Añadir participante al grupo de trabajo.

Curso alterno	
1b	Si el usuario intenta añadir al grupo de trabajo a sí mismo, recibe un mensaje de error y no se continua con el proceso.
1c	Si el usuario intenta añadir al grupo de trabajo a un usuario que no existe, recibe un mensaje de error y no se continua con el proceso.

Cuadro 2.39: Curso alterno de CU-25. Añadir participante al grupo de trabajo.

■ **CU-26: Ver participantes del grupo de trabajo.**

- **Descripción:** el usuario ve quienes son los participantes del grupo de trabajo.
- **Precondición:** el usuario ha iniciado sesión en el sistema. El usuario pertenece al grupo de trabajo.

Curso normal			
Actor		Sistema	
1	Usuario: elige la opción de ver los participantes del grupo.		
		2	Se muestran los participantes del grupo de trabajo.

Cuadro 2.40: Curso normal de CU-26. Ver participantes del grupo de trabajo.

#### 2.6.4. Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases e interfaces software y las relaciones entre éstas en una aplicación. Puede contener los siguiente elementos:

- Clases con sus atributos y sus operaciones.
- Interfaces con sus operaciones y constantes.
- Relaciones entre Clase/Clase, Clase/Interface o Interface/Interface.
- Información sobre el tipo de los atributos y parámetros.
- Navegabilidad de las asociaciones.

A continuación podemos ver el diagrama de clases del diseño propuesto:

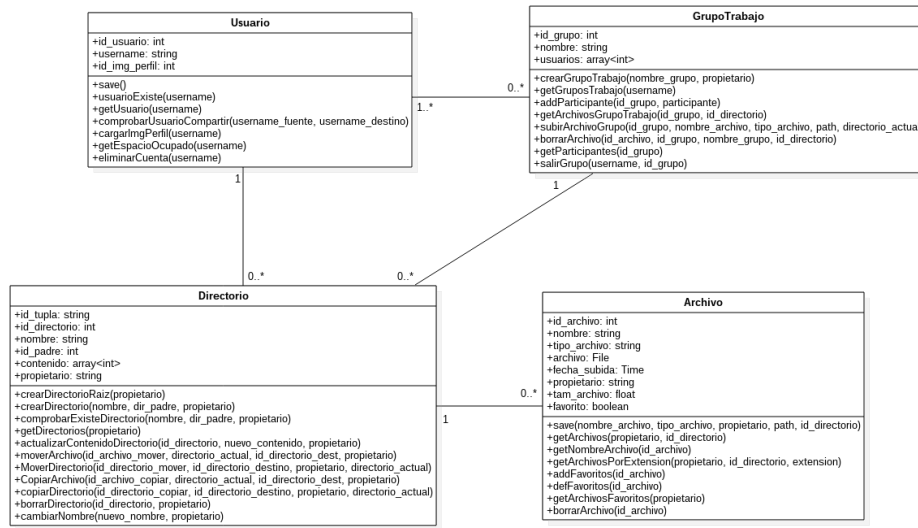


Figura 2.5: Diagrama de clases del diseño.

## Capítulo 3

# Diseño

En este apartado se comentarán los aspectos más importantes para el desarrollo de la aplicación. Para estudiar este apartado vamos a dividir el contenido en dos grupos: servidor (backend) y cliente (frontend).

### 3.1. Backend

El **backend** se encarga de recibir los datos desde el frontend y procesar dichos datos.

#### 3.1.1. Patrón Modelo-Vista-Controlador (MVC)

Para el desarrollo de la aplicación se plantea el uso del patrón de arquitectura de software MVC, el cual se basa en separar los datos de la lógica de la aplicación y su interfaz. Ésto facilita el desarrollo y el posterior mantenimiento de la aplicación.

- **Modelo** (Model): **información** con la que trabaja el sistema. Se encarga de gestionar los accesos a dicha información, tanto consultas como actualizaciones. Se corresponde con la base de datos.
- **Vista** (View): se refiere a los datos que se van a mostrar y como mostrarlos. La vista será el **frontend** de nuestra aplicación.
- **Controlador** (Controller): responde a **eventos** (generalmente los proporcionará el usuario) y llama al modelo cuando se realiza alguna solicitud sobre la información. También podrá enviar solicitudes a la vista si se solicita un cambio en la forma que mostrar la información.

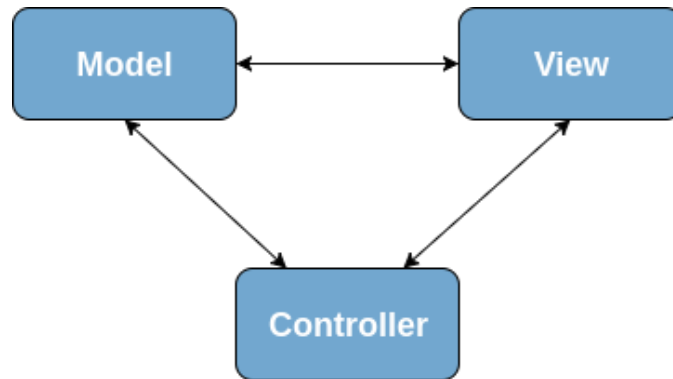


Figura 3.1: Diagrama Model-View-Controller.

### 3.1.2. Django

La decisión de usar el framework **Django** viene dada por su uso de la arquitectura MVC y su disponibilidad para trabajar con Python. Debido a que el controlador es manejado por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV (Model- View - Template) [17].

- **Modelo** (Model): **información** con la que trabaja el sistema. Se encarga de gestionar los accesos a dicha información, tanto consultas como actualizaciones. Se corresponde con la base de datos.
- **Vista** (View): esta capa contiene la **lógica** que accede al modelo y la delega a la plantilla apropiada: actúa como puente entre el modelos y las plantillas.
- **Plantilla** (Template): se relaciona con la **presentación** de la información: como algunas cosas son mostradas sobre una página web u otro tipo de documento.



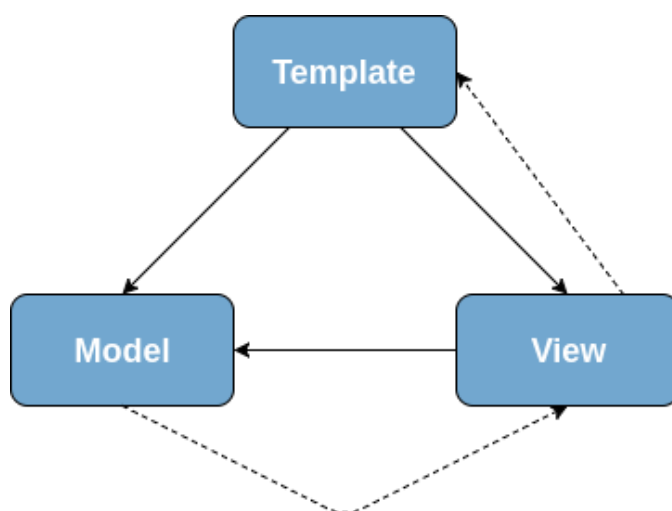


Figura 3.2: Diagrama Model-View-Template.

Django nos permite crear sitios web complejos ofreciendo diversos componentes que podemos usar para el desarrollo de nuestra aplicación.

### 3.1.3. Sistema gestor de base de datos

El sistema gestor de base de datos es el encargado de proporcionar la capa de acceso a la información de la base de datos. Haciendo uso del ORM de Django, utilizaremos un conector para la base de datos **MongoDB**.

## 3.2. Frontend

El **frontend** se refiere a la parte del sistema que se encarga de interactuar con los usuarios y mandar las peticiones al backend. Para interactuar con los usuarios hacemos uso de tecnologías web como **HTML**, **CSS** y **Javascript**. Este sistema trabaja todo el tiempo recibiendo peticiones de usuarios, por lo tanto usando las tecnologías mencionadas y añadiendo otras, como **jQuery**, haremos de nuestra interfaz un entorno intuitivo para los usuarios.

### 3.2.1. jQuery

**jQuery** es una librería de Javascript la cual ofrece la posibilidad de manipular los documentos HTML y manejar eventos, dotando a la web de una mayor dinamicidad. Es compatible con la mayoría de los navegadores actuales, tanto en pc como en dispositivos móviles.

### 3.2.2. AJAX

Por otro lado **AJAX** nos brinda la oportunidad de cargar datos desde el backend sin la necesidad de recargar toda la página del navegador.

### 3.2.3. Twitter Bootstrap

Otra tecnología usada de cara a la interacción con los usuarios de nuestra aplicación es la que nos aporta **Twitter Bootstrap**. Twitter Bootstrap es un framework CSS que nos facilita la tarea del diseño de la aplicación. Con esta tecnología podremos crear diseños muy elegantes y limpios, además de dotar a la aplicación web de un diseño adaptativo (responsive design) y así poder disfrutar de nuestra aplicación sin problemas en cualquier dispositivo y tamaño de pantalla.

## Capítulo 4

# Implementación

Tras haber visto la parte del diseño toca trabajar la implementación. En este apartado profundizaremos sobre el entorno de trabajo usado y la estructura de la aplicación.

### 4.1. Entorno de trabajo

Django nos da la posibilidad de trabajar con un “entorno virtual”. Estos entornos son creados con el objetivo de aislar recursos, como son las librerías y el entorno de ejecución, del sistema principal o de otros entornos virtuales. Esto quiere decir que podemos instalar distintas versiones de una librería sin que se provoquen conflictos.

Para usar el entorno de trabajo hacemos uso de la herramienta **virtualenv**. Los comandos para crear un entorno y activarlo son los siguientes:

- **Instalar virtualenv:** `sudo pip install virtualenv`
- **Crear entorno de trabajo:** `virtualenv drawercloud_env`
- **Activar el entorno de trabajo:** `source drawercloud_env/bin/activate`

Como podemos ver, para la instalación de virtualenv se ha usado **pip**. pip es la herramienta recomendada para instalar y administrar los paquetes de python. Los comandos que se usarán con más frecuencia son los que proceden:

- **Instalar un paquete:** `pip install <nombre_paquete>`
- **Desinstalar un paquete:** `pip uninstall <nombre_paquete>`
- **Actualizar un paquete:** `pip install <nombre_paquete>--upgrade`
- **Listar los paquetes instalados:** `pip freeze`

**Importante:** Todos los comandos se deberán usar con el entorno de trabajo activado.

## 4.2. Estructura del proyecto

A continuación se mostrarán las partes en las que se compone la estructura del proyecto.

### 4.2.1. Fichero de configuración `settings.py`

El fichero **`settings.py`** es el encargado de controlar la configuración del proyecto en Django. A continuación mencionaremos las opciones más destacables de este fichero según este proyecto:

- **DEBUG:** cuando la variable `DEBUG` se encuentre igualada a **`True`** querrá decir que el sistema se ejecuta en modo "depuración", el cual nos proporcióna los detalles de un error cuando éste se produzca. Cuando la aplicación se ejecute en "producción", a la variable le asignaremos el valor **`False`** por motivos de seguridad.
- **SECRET\_KEY:** es la llave secreta de nuestro proyecto y sirve para encriptar la información dentro de la base de datos.
- **INSTALLED\_APPS:** es una lista en la que se indica las aplicaciones habilitadas para este proyecto. Por ejemplo, para usar el administrador de Django añadimos la sentencia **`'django.contrib.admin'`**.
- **MIDDLEWARE:** es un framework que se usa como enlace en el proceso de solicitud/respuesta.
- **STATICFILES\_DIRS:** se indica la ruta a la carpeta **`static`** de nuestro proyecto. Dicha carpeta es la que contiene archivos relacionados con el frontend, como pueden ser los archivos CSS o los archivos Javascript.
- **`connect(database_name)`:** esta función se ocupa de conectar con la base de datos indicada en el parámetro `database_name`. **`connect`** es una función de MongoDB, para usarla haremos las siguientes importaciones:
  - `from pymongo import MongoClient`
  - `from mongoengine import connect`
- **REGISTRATION\_OPEN:** esta variable se igualará a **`True`** para permitir el registro de usuarios.
- **LOGIN\_REDIRECT\_URL:** indicamos la ruta a la que debe redirigirnos la aplicación una vez se ha realizado el log in con éxito.

- **LOGIN\_URL**: con esta variable decimos al sistema donde debemos aparecer en caso de no haber realizado el log in, o bien si se intenta acceder a páginas que requieren haberse logueado previamente.

Para obtener más información acerca del fichero de configuración settings.py de Django podemos visitar el enlace de la cita [18].

#### 4.2.2. Fichero de configuración urls.py

En este fichero se indican las **URLs** que se van a usar en la aplicación. Tendremos dos ficheros urls.py. El que ahora comentamos se encontrará en la misma carpeta que el fichero settings.py mientras que el otro se encontrará en la carpeta del proyecto, la cual contiene los archivos como views.py, la carpeta static o la carpeta templates. En este apartado nos referimos al archivo urls.py almacenado junto a settings.py. En éste indicaremos las rutas a la página de administración, la ruta al resto de urls del proyecto y la ruta necesaria para el registro de usuarios mediante django-registration-redux:

```
1  urlpatterns = [  
2      url(r'^admin/', admin.site.urls),  
3      url(r'^proyecto/', include('proyecto.urls')),  
4      url(r'^accounts/', include('registration.backends.simple.urls')),  
5  ]
```

#### 4.2.3. Fichero views.py

El archivo **views.py** contiene las funciones que recibirán peticiones web y responderán a dichas peticiones también con un resultado web. Las respuestas prodrán ser un código HTML, una imagen, reproducir una canción, etc. El propio archivo contiene cualquier lógica que sea necesaria para poder devolver una respuesta. A modo de ejemplo vemos una vista simple como es la página principal de la sección "Multimedia" de nuestra aplicación.

```
1  @login_required(login_url='/accounts/login/') #Requiere estar logueado para usar  
   la funcion  
2  def multimedia(request):  
3      return render(request, 'multimedia.html', {'pagina_actual': 'Multimedia'}) #  
       Renderiza el fichero HTML multimedia.html
```

#### 4.2.4. Fichero models.py

Recordamos que estamos usando una base de datos MongoDB. MongoDB es una base de datos orientada a documentos. Esto quiere decir que

los datos no se guardan en registros, sino que se guardan en los denominados documentos. Éstos se almacenan en BSON, que nos es más que una representación binaria de JSON.

El fichero encargado para la gestión de documentos es el fichero **models.py**. En dicho fichero detallamos la estructura de los documentos que vayamos a crear. Por ejemplo, para el documento **Archivo** (usado para almacenar los archivos que se suban a drawercloud) tenemos la siguiente disposición:

```
1 class Archivo(Document):
2     id_archivo = DecimalField()
3     nombre = StringField()
4     tipo_archivo = StringField()
5     archivo = FileField()
6     fecha_subida = StringField()
7     propietario = StringField()
8     tam_archivo = DecimalField()
9     favorito = BooleanField()
```

#### 4.2.5. Fichero forms.py

En el archivo **forms.py** escribiremos toda la funcionalidad necesaria para gestionar los documentos de la base de datos. Las acciones tales como crear un usuario, almacenar un archivo, añadir un usuario a un grupo de trabajo... vendrán descritas en este fichero. Por ejemplo, la función para guardar un nuevo usuario quedaría de la siguiente manera:

```
1 def save(self, _username):
2     u = Usuario()
3
4     u.username = _username
5     u.id_username = Usuario.objects.count() + 1
6     u.img_perfil = -1
7
8     u.save()
9
10    return u
```

#### 4.2.6. Fichero urls.py

Nos encontramos aquí con el otro fichero **urls.py**. Dicho fichero contiene todas las urls de todas las funciones que se encuentran en el fichero views.py. Por ejemplo, para la función que se ocupa de descargar un archivo tendremos algo tal que así:

---

```
1 url(r'^descargarArchivo/$', views.descargarArchivo, name='descargarArchivo'),
```

---

Analizando la sentencia, tenemos que el primer argumento es la url que tendrá en la web. El segundo argumento indica el nombre de la función en el fichero `views.py`. Por último, el tercer argumento indica el nombre con el cual nos podremos referir a la función cuando queramos invocarla, por ejemplo, desde un fichero HTML o Javascript.

#### 4.2.7. Directorio templates

Este directorio contiene todos los archivos **HTML** que forman la vista de la aplicación.

#### 4.2.8. Directorio static

Este directorio contiene todos los archivos **CSS, Javascript, imágenes, ...** que complementan a los archivos HTML almacenados en el directorio templates y que juntos forman el frontend.





## Capítulo 5

# Pruebas

Se han realizado distintas pruebas para estudiar el comportamiento de la aplicación web (dichas pruebas se realizaron con la última versión de la aplicación). Para realizar las pruebas se ha usado la ayuda de un grupo de personas, las cuales han estado interactuando con la aplicación durante 10 minutos. Los resultados han sido los siguientes:

### 5.1. Resultados de las pruebas con usuarios



## Capítulo 6

# Conclusiones y trabajos futuros

### 6.1. Conclusiones

Tras la finalización del proyecto se ha obtenido un prototipo funcional de la aplicación web, que cumple con casi todas las expectativas y los objetivos marcados desde el principio, además de una completa memoria del proyecto.

Gracias a las pruebas realizadas con usuarios podemos determinar que.....

En definitiva tenemos que **drawercloud** cumple con los requisitos para efectuar su función principal de alojar archivos en la nube, a excepción del espacio disponible de almacenamiento actual que es insuficiente para una aplicación de estas características. Esto se debe a que el espacio que nos ofrece Amazon es solamente de 2GB, debiéndose esto a que se trata de la versión gratuita. No obstante esto tendría solución comprando más espacio de almacenamiento y ofreciendo a los clientes distintas versiones, las cuales serían de pago para grandes cantidades de espacio y una gratuita que ofrecería menos capacidad.

#### 6.1.1. Conocimientos adquiridos

Gracias al análisis y al desarrollo de este proyecto se han adquirido conocimientos en diversos ámbitos del desarrollo de software, en su mayoría en el sector web. Entre los conocimientos adquiridos podemos nombrar:

- **Python:**
- **Django:**
- **Arquitectura MVC:**

- Tecnologías web:
- Pruebas con usuarios:
- Desarrollo de documentación:

## 6.2. Trabajos futuros

## Capítulo 7

# Manual de usuario

### 7.1. Acceso a la aplicación

La primera página que se carga de nuestra aplicación web es la que se refiere al login. En la imagen 7.1 podemos consultar el aspecto de dicha página. Observamos dos entradas de texto, una para insertar el nombre de usuario y otra para la contraseña.

Para usuarios que no estén registrados se presenta la opción de **Crear una nueva cuenta**, en la cual debemos rellenar los datos requeridos: nombre de usuario, dirección de correo electrónico, contraseña y confirmar contraseña. En la imagen 7.2



Figura 7.1: Acceso a la aplicación. Página de login

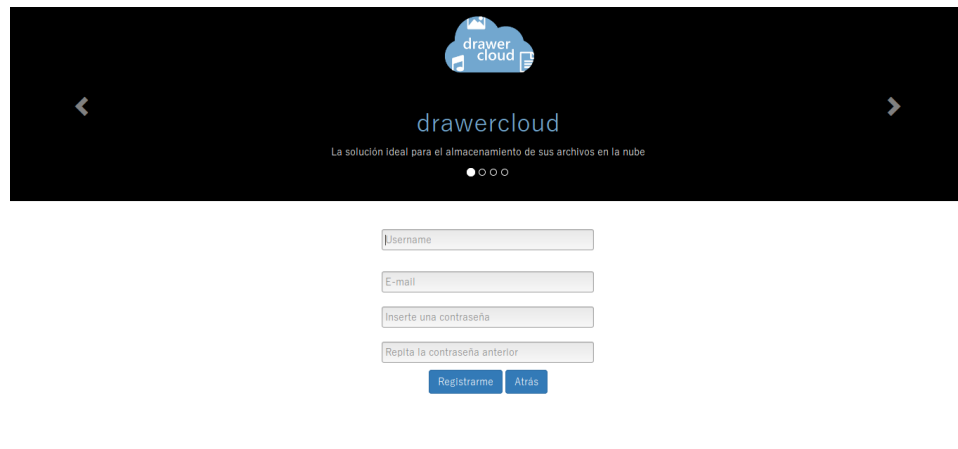


Figura 7.2: Acceso a la aplicación. Página de registro

## 7.2. Página principal - Documentos

La página principal es la sección **Documentos** 7.4. En esta página nos encontraremos con una estructura de directorios y archivos. La vista en que se muestran los directorios y archivos se podrá cambiar haciendo uso de los botones de la imagen 7.3, estando disponibles dos tipos de vista: lista 7.4 o iconos 7.5.

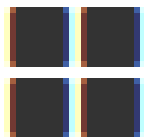
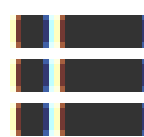
(a) Cambiar a la vista **iconos**(b) Cambiar a la vista **lista**

Figura 7.3: Botones disponibles para cambiar la vista

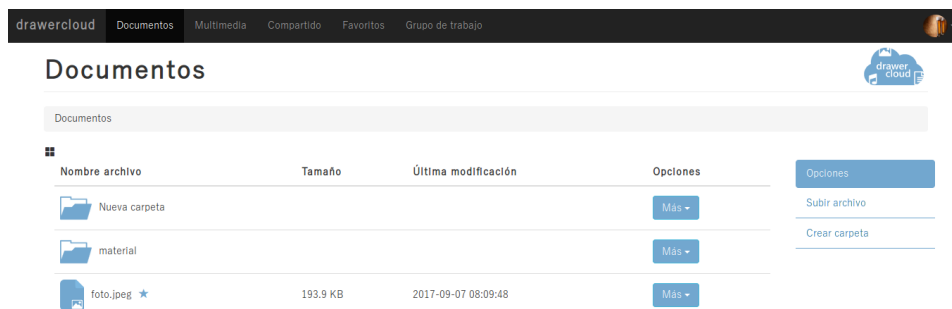


Figura 7.4: Documentos. Aspecto de la página principal



Figura 7.5: Documentos. Aspecto de la página principal

En **Documentos** vamos a poder almacenar los archivos que deseemos, pudiendo organizar tales archivos en directorios. Para **crear directorios** o **subir archivos** hacemos uso de las opciones que aparecen a la derecha de la página documentos 7.6.



Figura 7.6: Documentos. Opciones de la página Documentos

Los archivos y directorios disponen de **funcionalidades** propias, como la posibilidad de descargar un archivo, copiar un directorio, etc. Para acceder a estas opciones podemos elegir entre hacer **click derecho** sobre los directorios/archivos o seleccionar el botón **Más** disponible para la vista lista.

#### 7.2.1. Opciones para archivos

- Descargar:
- Compartir:
- Cambiar nombre:
- Copiar en:
- Mover a:
- Añadir a favoritos:
- Eliminar:

#### 7.2.2. Opciones para directorios

- Cambiar nombre:
- Copiar en:
- Mover a:
- Eliminar:







# Bibliografía

- [1] “Web oficial de django <https://www.djangoproject.com/>,” consultado en Marzo de 2017.
- [2] “Web oficial de python <https://www.python.org/>,” consultado en Marzo de 2017.
- [3] “Web oficial de bootstrap <http://getbootstrap.com/>,” consultado en Marzo de 2017.
- [4] “Web oficial de w3schools <https://www.w3schools.com/html/>,” consultado en Marzo de 2017.
- [5] “Web oficial de w3schools <https://www.w3schools.com/css/>,” consultado en Marzo de 2017.
- [6] “Web oficial de javascript <https://www.javascript.com/>,” consultado en Marzo de 2017.
- [7] “Web oficial de ajax <https://learn.jquery.com/ajax/>,” consultado en Marzo de 2017.
- [8] “Web oficial de jquery <https://jquery.com/>,” consultado en Marzo de 2017.
- [9] “Web oficial de mongodb <https://www.mongodb.com/>,” consultado en Marzo de 2017.
- [10] “Web oficial de amazon aws [https://aws.amazon.com/es/ec2/?sc\\_channel=PS&sc\\_campaign=acquisition\\_ES&sc\\_publisher=google&sc\\_medium=english\\_ec2\\_b&sc\\_content=ec2\\_e&sc\\_detail=ec2%20amazon&sc\\_category=ec2&sc\\_segment=176355462675&sc\\_matchtype=e&sc\\_country=ES&s\\_kwid=AL!4422!3!176355462675!e!!g!!ec2%20amazon&ef\\_id=V01R4AAABFRIjhue:20170718144015:s](https://aws.amazon.com/es/ec2/?sc_channel=PS&sc_campaign=acquisition_ES&sc_publisher=google&sc_medium=english_ec2_b&sc_content=ec2_e&sc_detail=ec2%20amazon&sc_category=ec2&sc_segment=176355462675&sc_matchtype=e&sc_country=ES&s_kwid=AL!4422!3!176355462675!e!!g!!ec2%20amazon&ef_id=V01R4AAABFRIjhue:20170718144015:s),” consultado en Julio de 2017.
- [11] “Web oficial de nginx <https://www.nginx.com/>,” consultado en Julio de 2017.

- [12] “Web oficial de gunicorn <http://gunicorn.org/>,” consultado en Julio de 2017.
- [13] “Web oficial de dropbox <https://www.dropbox.com/>,” consultado en Marzo de 2017.
- [14] “Web oficial de google [https://www.google.com/intl/es\\_ALL/drive/](https://www.google.com/intl/es_ALL/drive/),” consultado en Marzo de 2017.
- [15] “Web oficial de mega <https://mega.nz/>,” consultado en Marzo de 2017.
- [16] “Web oficial de onedrive <https://onedrive.live.com/about/es-419/>,” consultado en Marzo de 2017.
- [17] “Web librosweb [http://librosweb.es/libro/django\\_1\\_0/capitulo\\_5/el\\_patron\\_de\\_diseno\\_mtv.html](http://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html),” consultado en Septiembre de 2017.
- [18] “Web django - settings <https://docs.djangoproject.com/en/1.11/topics/settings/>,” consultado en Septiembre de 2017.