



**UNIVERSIDAD DE JAÉN**  
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

---

**CREACIÓN DE UN SERVICIO CON  
REGISTRO PREMIUM/FREEMIUM  
QUE POSIBILITE LA  
CLASIFICACIÓN DE CONTENIDO  
MULTIMEDIA**

**Alumno: Álvaro Soriano Vilches**

**Tutor:** Prof. D. Ángel Inocencio Aguilera García  
**Depto.:** Departamento de Informática

**Septiembre, 2017**

## AGRADECIMIENTOS

Llegados a este punto, finalización del trabajo fin de grado y por consiguiente, finalización de mi Grado en Ingeniería Telemática, pienso que merece la pena hacer un inciso y ver como he llegado aquí.

Primeramente, mencionar a una persona, la cual, me enseñó a que no hay nada imposible de lograr si te lo propones con la suficiente fuerza y, que posiblemente, sin su apoyo en mi temprana etapa de mis estudios en la Universidad de Jaén me hubiera resultado, esta etapa hubiera resultado más dura: Jesús de la Casa Cardenas. Gracias por todo el apoyo recibido durante su asignatura.

Dar las gracias también a toda esa gente que me ha apoyado en este camino, amigos, familia y en especial, a mi novia Davinia, y a mi hermana María.

Por último y no por ello, menos importante. Dar las gracias a mi tutor Ángel, por el buen trato que me ha dado durante todo el tutelaje de este trabajo fin de grado, por todos sus consejos y por todo lo aprendido de él durante el transcurso del mismo.

## RESUMEN

La inclusión en el mercado de smartphones en la población es muy alta, éstos dispositivos nos proporcionan nuevas funcionalidades, tratándose de pequeños ordenadores y permitiéndonos interactuar con la web desde cualquier parte del mundo. Android se ha impuesto como el principal sistema operativo, en cuanto cuota de mercado, de la presente generación de teléfonos móviles. En este trabajo se ha desarrollado una aplicación híbrida que permite controlar nuestra biblioteca de medios y tenerla lo más centralizada posible, tanto en nuestro móvil como PC, tableta o cualquier otro dispositivo que sea capaz de ejecutar un navegador Web, llevando un control de acceso para cada usuario y ofreciendo un servicio añadido, llamado “Premium” por el cuál podremos acceder a la biblioteca de medios de otro usuario que haya dado su permiso para ser visualizada por los demás usuarios de la aplicación.

## ABSTRACT

The market inclusion of smartphones in the population is very high, these devices give us new features, being small computers and letting us interact with the web from any site of the world. Android has emerged as the main operating system, in terms of market share, of the current generation of mobile phones. This paper has developed an hybrid application which enables control to our media library and has it centralized, both in our mobile phone and PC, tablet or any other device which is able to launch a web browser, keeping track of the access for every user and offering another service, called “Premium”. A service that provide us access to the media library of another user that allow us to see it in our devices.

## Contenido

1.	Introducción.....	11
2.	Planificación del Proyecto.....	12
2.1.	Objetivos.....	12
2.1.1.	Objetivo general.....	12
2.1.2.	Objetivos Docentes.....	12
2.2.	Planificación del proyecto.....	13
2.2.1.	Diagrama de Gantt.....	13
2.2.1.1.	Diagrama de Gantt General.....	14
2.2.1.2.	Diagrama de Gantt de la realización del Backend.....	16
2.2.1.3.	Diagrama de Gantt de la realización del Frontend.....	17
3.	Materiales y métodos.....	19
3.1.	Antecedentes.....	19
3.1.1.	Plex.....	19
3.1.2.	Kodi.....	21
3.1.3.	Emby.....	23
3.1.4.	OSMC.....	24
3.1.5.	MediaPortal.....	25
3.1.6.	JRiver.....	26
3.1.7.	Stremio.....	27
3.1.8.	Usher.....	27
3.2.	Estado del Arte Tecnológico.....	29
3.2.1.	Estudio Preliminar.....	29
3.2.1.1.	Java.....	29
3.2.1.2.	Cordova.....	30
3.2.1.3.	PHP.....	30
3.2.1.4.	JavaScript.....	31

3.2.1.5.	JSON .....	31
3.2.1.6.	CodeIgniter.....	31
3.2.1.7.	Ionic .....	32
3.2.1.8.	MySQL.....	32
3.2.1.9.	PostgreSQL .....	32
3.2.2.	Elección de materiales y métodos.....	33
3.3.	Estado del arte legislativo .....	33
3.4.	Análisis del Sistema.....	34
3.4.1.	Definición del sistema .....	34
3.4.2.	Definición de requisitos funcionales .....	37
3.4.2.1.	RSF-01. Registro de una nueva cuenta. ....	37
3.4.2.2.	RSF-02. Acceso Cliente .....	38
3.4.2.3.	RSF-03. Acceso Administrador. ....	38
3.4.2.4.	RSF-04. Recuperar contraseña.....	39
3.4.2.5.	RSF-05. Consultar datos de acceso.....	39
3.4.2.6.	RSF-06 – Modificar datos de Acceso .....	40
3.4.2.7.	RSF-07 – Cerrar Sesión .....	41
3.4.2.8.	RSF-08 – Mirar Alertas.....	41
3.4.2.9.	RSF-09 – Búsqueda por Nombre .....	42
3.4.2.10.	RSF-10 - Reproducción .....	43
3.4.2.11.	RSF-11 – Subida de Ficheros.....	43
3.4.3.	Definición de requisitos no funcionales .....	44
3.4.3.1.	RSNF-01. Navegador.....	44
3.4.3.2.	RSNF-02. Idioma. ....	45
3.4.3.3.	RSNF-03. Utilización de la interfaz.....	45
3.4.3.4.	RSNF-04. Comprobación de campos obligatorios. ....	46
3.4.3.5.	RSNF-05. Unicidad en usuarios registrados. ....	46

3.4.3.6. RSNF-06. Uso de Validadores.....	47
3.4.3.7. RSNF-07. Seguridad. ....	48
3.5. Diseño del Sistema.....	49
3.5.1. Descripción de la Arquitectura del sistema y justificación de las tecnologías utilizadas .....	49
3.5.2. Diseño de la lógica de negocio .....	50
3.5.2.1. Casos de uso.....	50
3.5.2.1.1. Actor.....	50
3.5.2.1.2. Caso de Uso.....	51
3.5.2.1.3. Relaciones .....	51
3.5.2.1.3.1. Asociación .....	51
3.5.2.1.3.2. Dependencia o Instanciación .....	51
3.5.2.1.3.3. Generalización .....	51
3.5.2.1.4. Casos de uso de YouSync .....	52
3.5.2.1.4.1. Casos de uso del modulo del Usuario.....	52
3.5.2.1.4.1.1. Crear una cuenta.....	52
3.5.2.1.4.1.2. Acceder a la aplicación .....	54
3.5.2.1.4.1.3. Recuperar la contraseña .....	56
3.5.2.1.4.1.4. Consultar datos del usuario. ....	58
3.5.2.1.4.5. Modificar datos de acceso de un usuario registrado. ....	60
3.5.2.1.4.6. Cerrar la Sesión.....	62
3.5.2.1.4.2. Casos de uso del modulo de Administrador .....	64
3.5.2.1.4.2.1. Acceder a la aplicación. ....	64
3.5.2.1.4.2.2. Recuperar la contraseña. ....	66
3.5.2.1.4.2.3. Consultar datos del administrador.....	68
3.5.2.1.4.2.4. Modificar datos de acceso del administrador.....	70
3.5.2.1.4.2.5. Mirar alertas. ....	72

3.5.2.1.4.2.6.	Alerta Roja.....	74
3.5.2.1.4.2.7.	Alerta Verde.....	76
3.5.2.1.4.2.8.	Cerrar la Sesión.....	78
3.5.2.1.4.3.	Casos de uso del modulo del buscador.....	80
3.5.2.1.4.3.1.	Búsqueda por nombre.....	80
3.5.2.1.4.3.2.	Selección.....	82
3.5.2.1.4.3.3.	Reproducción de Archivos.....	84
3.5.3.	Diseño de la lógica de datos.....	86
3.5.3.1.	Modelo Entidad-Relación.....	86
4.	Conclusiones y Líneas Futuras.....	87
5.	Anexo 1. Manual Usuario.....	89
6.	Anexo 2. Instalación de Ionic.....	91
6.1.	Instalación de Node.js.....	91
6.2.	Instalación de la consola de comandos de Ionic.....	92
6.3.	Instalación de Bower.....	92
6.4.	Instalación de Java y Android en Windows.....	92
6.5.	Creando un proyecto en blanco y testeándolo.....	93
7.	Bibliografía.....	94

## Ilustraciones

Ilustración 1. Diagrama de Gantt General .....	15
Ilustración 2. Diagrama de Gantt del Backend .....	17
Ilustración 3. Diagrama de Gantt del Frontend .....	18
Ilustración 4. Vista general del ecosistema “Plex” .....	19
Ilustración 5. Plex en el lado de Servidor desde una Raspberry pi.....	20
Ilustración 6. Plex en el lado del cliente visto desde una Xbox One.....	21
Ilustración 7. Pantalla principal de XBMC corriendo en uan Xbox.....	22
Ilustración 8. Pantalla principal de Kodi .....	23
Ilustración 9. Emby instalado en Kodi .....	24
Ilustración 10. Pantalla Principal de OSMC.....	25
Ilustración 11. Pantalla principal de MediaPortal.....	26
Ilustración 12. Pantalla principal de JRiver .....	26
Ilustración 13. Pantalla principal de Stremio.....	27
Ilustración 14. Pantalla principal de Usher.....	28
Ilustración 15. Definición de Requisito .....	34
Ilustración 16. Diseño del caso de uso 1.....	53
Ilustración 17. Resultado Final del caso de uso 1.....	53
Ilustración 18. Diseño del caso de uso 2.....	55
Ilustración 19. Resultado Final del caso de uso 2.....	55
Ilustración 20. Diseño del caso de uso 3.....	57
Ilustración 21. Resultado Final del caso de uso 3.....	57
Ilustración 22. Diseño del caso de uso 4.....	59
Ilustración 23. Resultado Final del caso de uso 4.....	59
Ilustración 24. Diseño del caso de uso 5.....	61
Ilustración 25. Resultado Final del caso de uso 5.....	61
Ilustración 26. Diseño del caso de uso 6.....	63
Ilustración 27. Resultado Final del caso de uso 6.....	63
Ilustración 28. Diseño del caso de uso 7.....	65
Ilustración 29. Resultado Final del caso de uso 7.....	65
Ilustración 30. Diseño del caso de uso 8.....	67
Ilustración 31. Resultado Final del caso de uso 8.....	67
Ilustración 32. Diseño del caso de uso 9.....	69



Ilustración 33. Resultado Final del caso de uso 9.....	69
Ilustración 34. Diseño del caso de uso 10.....	71
Ilustración 35. Resultado Final del caso de uso 10.....	71
Ilustración 36. Diseño del caso de uso 11.....	73
Ilustración 37. Resultado Final del caso de uso 11.....	73
Ilustración 38. Diseño del caso de uso 12.....	75
Ilustración 39. Resultado Final del caso de uso 12.....	75
Ilustración 40. Diseño del caso de uso 13.....	77
Ilustración 41. Resultado Final del caso de uso 13.....	77
Ilustración 42. Diseño del caso de uso 14.....	79
Ilustración 43. Resultado Final del caso de uso 14.....	79
Ilustración 44. Diseño del caso de uso 15.....	81
Ilustración 45. Resultado Final del caso de uso 15.....	81
Ilustración 46. Diseño del caso de uso 16.....	83
Ilustración 47. Resultado Final del caso de uso 16.....	83
Ilustración 48. Diseño del caso de uso 17.....	85
Ilustración 49. Esquema E/R .....	86

## Tablas

Tabla 1. Actividades del Trabajo Fin de Grado.....	14
Tabla 2. Actividades de la realización del Backend.....	16
Tabla 3. Actividades de la realización del Frontend.....	18
Tabla 4. Ejemplo de Tabla de Requisito de Software .....	37
Tabla 5. Requisitos de Software. RSF-01 – Registro de una Nueva Cuenta.....	37
Tabla 6. Requisitos de Software. RSF-02 – Acceso Cliente .....	38
Tabla 7. Requisitos de Software. RSF-03 – Acceso Administrador.....	39
Tabla 8. Requisitos de Software. RSF-04 – Recuperar Contraseña .....	39
Tabla 9. Requisitos de Software. RSF-05 -- Consultar datos de acceso.....	40
Tabla 10. Requisitos de Software. RSF-06 – Modificar datos de acceso.....	40
Tabla 11. Requisitos de Software. RSF-07 – Cerrar Sesión.....	41
Tabla 12. Requisitos de Software. RSF-08 – Mirar Alertas .....	42
Tabla 13. Requisitos de Software. RSF-09 -- Búsqueda por nombre.....	42
Tabla 14. Requisitos de Software. RSF-10 -- Reproducción.....	43
Tabla 15. Requisitos de Software. RSF-11 – Subida de Archivos .....	44
Tabla 16. Requisito de Software. RSNF-01 – Navegador.....	44
Tabla 17. Requisito de Software. RSNF-02 – Idioma.....	45
Tabla 18. Requisito de Software. RSNF-03 – Utilización de la interfaz.....	46
Tabla 19. Requisito de Software. RSNF-04 – Comprobación de campos obligatorios .....	46
Tabla 20. Requisito de Software. RSNF-05 – Comprobación de que haya unicidad en usuarios registrados. ....	47
Tabla 21. Requisito de Software. RSNF-06 – Uso de Validadores.....	47
Tabla 22. Requisito de Software. RSNF-07 -- Seguridad.....	48
Tabla 23. Caso de Uso 1 .....	52
Tabla 24. Caso de Uso 2 .....	54
Tabla 25. Caso de Uso 3 .....	57
Tabla 26. Caso de uso 4.....	58
Tabla 27. Caso de uso 5 .....	60
Tabla 28. Caso de Uso 6.....	62
Tabla 29. Caso de Uso 7 .....	64
Tabla 30. Caso de Uso 8.....	67

Tabla 31. Caso de Uso 9 .....	68
Tabla 32. Caso de Uso 10 .....	70
Tabla 33. Caso de uso 11 .....	72
Tabla 34. Caso de Uso 12 .....	74
Tabla 35. Caso de Uso 13 .....	76
Tabla 36. Caso de Uso 14 .....	78
Tabla 37. Caso de Uso 15 .....	80
Tabla 38. Caso de Uso 16 .....	82
Tabla 39. Caso de Uso 17 .....	84

## 1. Introducción

En la actualidad, consumimos servicios como pueden ser “Spotify”, “Deezer”, “iTunes”, “Facebook”, “Netflix”, “Youtube”, “Lastfm”, etc. Dónde en cada uno de ellos, estamos compartiendo nuestros ficheros multimedia ya sean de audio, imagen o vídeo, o estamos usando archivos ya precargados en servidores de los cuáles reproducimos sus archivos via “Streaming”, pero éstos se suelen centrar, en contenidos comerciales o contenidos subidos por distintos usuarios. Sin embargo, ¿hay algún servicio que nos permita tanto tener una biblioteca online compartida con otros usuarios y que, además, nos deje elegir que contenidos compartimos y con qué usuarios lo compartimos? La respuesta, hasta ahora, es no. Por esta serie de motivos, he visto una gran oportunidad de realizar un TFG que inicie esta nueva senda, realizando en un principio una aplicación híbrida que sea, ante todo, funcional para más adelante poder ser continuado por otros TFG y llegar a poder competir con servicios de esta índole.

## 2. Planificación del Proyecto

### 2.1. Objetivos

#### 2.1.1. Objetivo general

El TFG surge de la necesidad de ofrecer a una posible alta cantidad de usuarios una nueva forma de organizar su biblioteca audiovisual y a la misma vez poder compartirla con otros usuarios.

El objetivo principal de este proyecto es dar a los clientes la posibilidad de tener toda su biblioteca audiovisual centralizada y que sea accesible desde cualquier parte del mundo mediante una conexión a Internet con cualquier plataforma, ya sea desde un PC con Windows/Linux/MacOs, o desde un smartphone o Tablet con Android/Ios/Windows Phone y que en la medida que nos sea posible no haya una diferenciación en el uso de la aplicación cuando se esté usando en un smartphone o Tablet frente a un PC.

Además, existirá una diferenciación entre dos servicios: el servicio freemium y el servicio Premium con la única diferencia de que siempre que se suba un contenido se marcará si éste contenido es Público o Privado, y solo los poseedores del servicio Premium podrán ver los contenidos que estén marcados como “Privados”.

#### 2.1.2. Objetivos Docentes

Al tener 12 créditos ECTS, el Trabajo Fin de Grado va a tener unos objetivos propios de ello como es la limitación de tiempo. El esfuerzo realizado en este Trabajo Fin de Grado nunca debería superar las 300 horas de trabajo ni tampoco estar por debajo de éstas mismas, presupuestándolo teniendo en consonancia esto mismo. Si realizáramos una estimación de tiempo en un cuatrimestre, nos saldría unas 20h/semana de dedicación para el Trabajo Fin de Grado.

Hay que tener en cuenta, además, que el Trabajo Fin de Grado se ha realizado en un entorno de desarrollo completamente nuevo para el alumno, y esto conlleva una complejidad adicional; Por lo tanto, hay que tener en cuenta las capacidades del alumno.

Si aunamos esto al propósito principal que se tendrá en cuenta a la hora de realizar la aplicación: Ser lo más accesible que se pueda, teniendo la menor cantidad de requisitos, llegamos al objetivo principal que tiene un Trabajo Fin de Grado; que es el de aprender y preparar al alumno para su inserción a la vida laboral.

## 2.2. Planificación del proyecto.

El objetivo principal de la planificación es conseguir garantizar las condiciones de entregables exigidos por la Universidad de Jaén y, además, ceñirnos al objetivo docente visto anteriormente de duración del mismo: 12 créditos ECTS lo que conlleva 300 horas de trabajo.

### 2.2.1. Diagrama de Gantt

El diagrama de Gantt es una de las técnicas de planificación más comunes. En él están todas las actividades o tareas en las que el proyecto está dividido, junto con la duración de cada una de ellas y los hitos a cumplir para garantizar la finalización del proyecto.

Este TFG tiene varias actividades dentro de las cuáles hay dos más grandes: La realización del FrontEnd y la realización del denominado Backend. Primero detallaré las actividades que conformarán el bloque, y subdividiré las actividades propias del Frontend y del Backend. Así mismo reflejaré en esta memoria el Diagrama de Gantt general y realizaré el específico para el desarrollo del FrontEnt y el desarrollo del BackEnd.

### 2.2.1.1. Diagrama de Gantt General

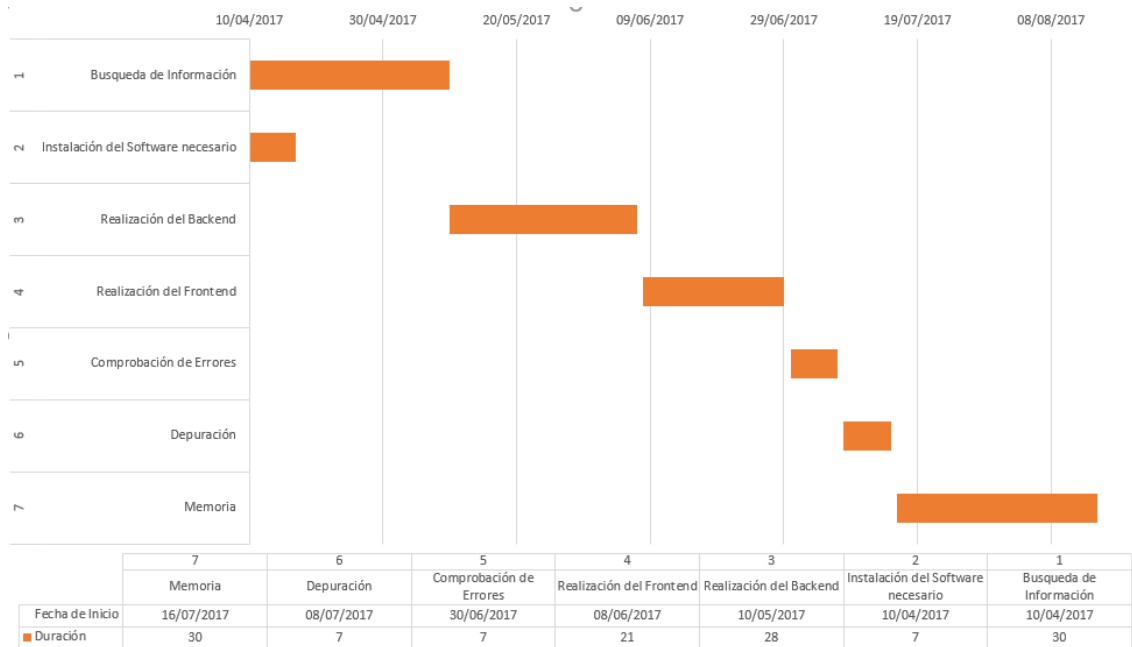
Primero tenemos que tener en cuenta la duración en horas que se prevee que se va a utilizar en el TFG durante el desarrollo del mismo. Son 300 horas de Trabajo Fin de Grado entre los 130 días entre los cuáles se realiza la estimación, que hace un total entre 2h y cuarto por día y 2 horas y veinte por día. Lo que nos da las 20h/semana descritas anteriormente.

En esta tabla podremos ver en qué actividades se han dividido este TFG:

Número Actividad	Actividad	Fecha de Inicio	Duración (días)	Fecha de Finalización
1	Búsqueda de información	10/04/2017	30	10/05/2017
2	Instalación del software necesario	10/04/2017	7	17/04/2017
3	Realización del Backend	10/05/2017	28	07/06/2017
4	Realización del Frontend	08/06/2017	21	29/06/2017
5	Comprobación de errores	30/06/2017	7	07/07/2017
6	Depuración	08/07/2017	7	15/07/2017
7	Memoria	16/07/2017	30	15/08/2017

Tabla 1. Actividades del Trabajo Fin de Grado

En la siguiente figura se muestra el gráfico Gantt con la planificación y duración de las tareas:



*Ilustración 1. Diagrama de Gantt General*

Cómo se puede observar en la anterior figura el TFG se divide en un total de 7 actividades habiendo solo dos que se realizan al mismo tiempo. En las demás para poder seguir con la siguiente actividad e ir cumpliendo los hitos deberemos haber realizado la anterior actividad.



2.2.1.2. Diagrama de Gantt de la realización del Backend

En la siguiente tabla podemos ver en que actividades se ha subdividido la realización del Backend:

Número Actividad	Actividad	Fecha de Inicio	Duración (días)	Fecha de Finalización
1	Instalación de MySQL	10/05/2017	1	11/05/2017
2	Modelo Entidad-Relación	12/05/2017	2	14/05/2017
3	Casos de uso de la aplicación	15/05/2017	2	17/05/2017
4	Creación y funcionamiento del primer usuario	18/05/2017	22	20/05/2017
5	Consultas, instancias a base de datos	21/05/2017	2	23/05/2017
6	Creación del Código	24/05/2017	11	04/06/2017
7	Depuración	05/06/2017	2	07/06/2017

Tabla 2. Actividades de la realización del Backend.

En la siguiente figura se muestra el gráfico Gantt con la planificación y duración de las tareas referentes a la realización del Backend:

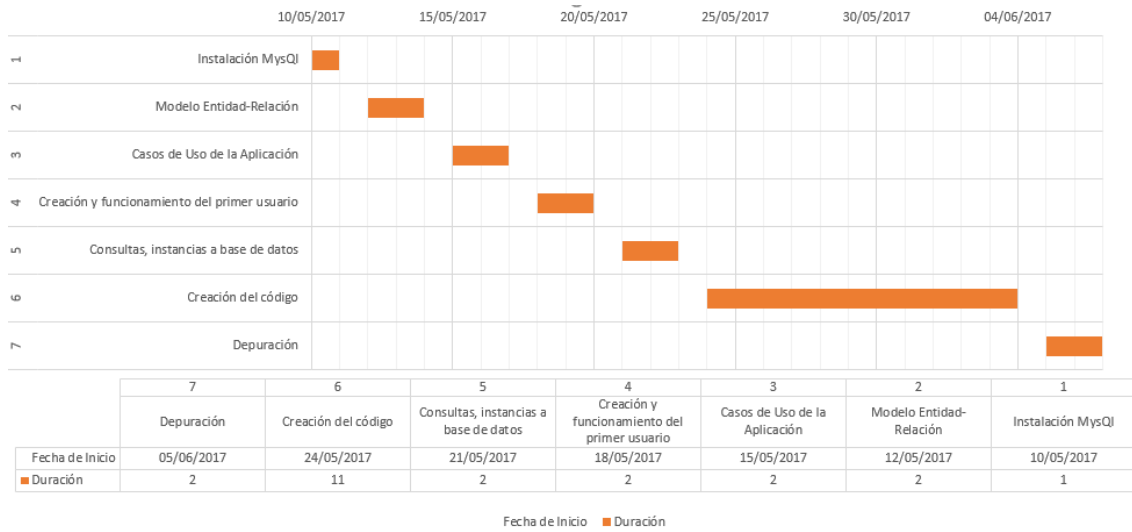


Ilustración 2. Diagrama de Gantt del Backend

### 2.2.1.3. Diagrama de Gantt de la realización del Frontend

En la siguiente tabla podemos ver en que actividades se ha subdividido la realización del Frontend:

Número de Actividad	Actividad	Fecha de Inicio	Duración (días)	Fecha de Finalización
1	Elección de Software	08/06/2017	3	11/06/2017
2	Programación del sitio Web	12/06/2017	3	15/06/2017
3	Depuración del sitio Web	17/06/2017	2	19/06/2017

4	Programación aplicación Móvil	20/06/2017	4	24/06/2017
5	Depuración aplicación móvil	25/06/2017	2	27/06/2017
6	Depuración FrontEnd	28/06/2017	1	29/06/2017

Tabla 3. Actividades de la realización del Frontend

En la siguiente figura se muestra el gráfico Gantt con la planificación y duración de las tareas referentes a la realización del Frontend:

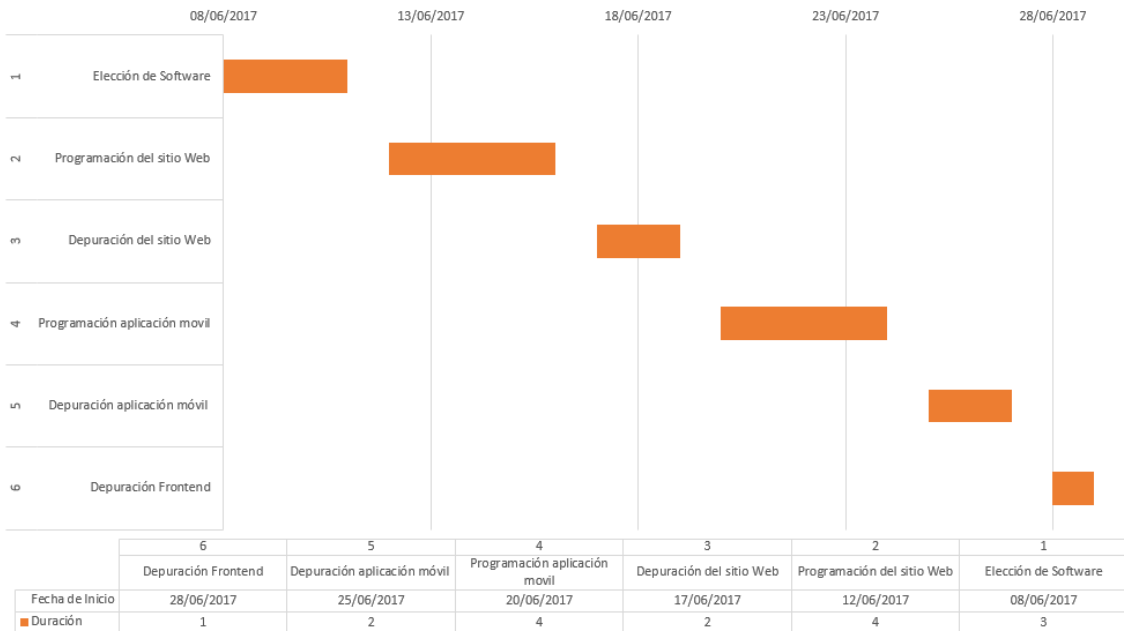


Ilustración 3. Diagrama de Gantt del Frontend

### 3. Materiales y métodos

En este capítulo de la memoria se estudiará el estado del arte tanto tecnológico como legislativo que hay en la actualidad y se detallará con la mayor profundidad posible el análisis realizado para la elaboración de nuestro sistema híbrido, señalando casos de uso del mismo, definiendo los requisitos funcionales y no funcionales, y por último desglosando el diseño de las distintas pantallas que recorrerá los usuarios.

#### 3.1. Antecedentes.

A pesar de que no hay en la actualidad, servicios o gestores de contenido multimedia, que garantice totalmente los servicios que vamos a ofrecer, si hay algún que otro programa/servicio que tiene alguna función similar y que es conveniente que analicemos en este apartado. [1] [3]

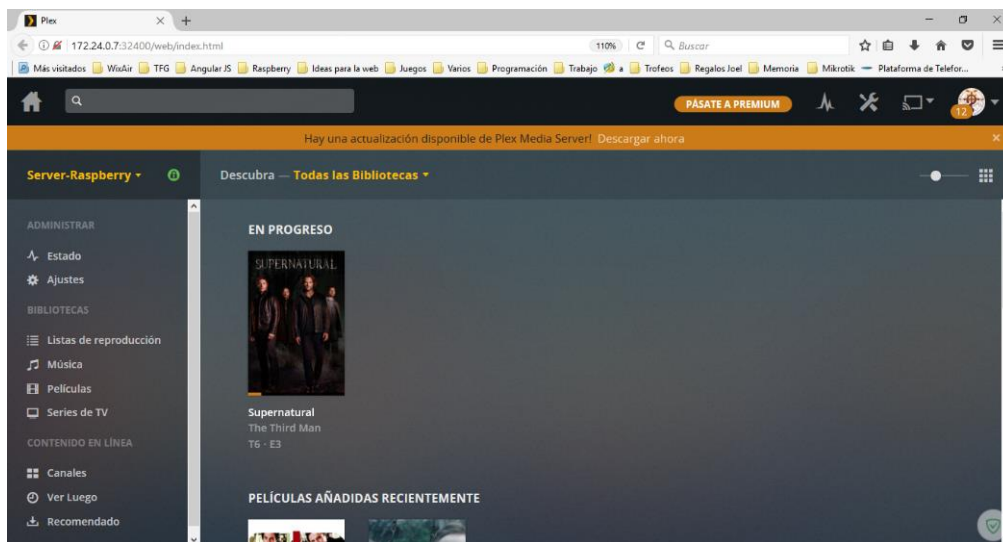
##### 3.1.1. Plex



Ilustración 4. Vista general del ecosistema "Plex"

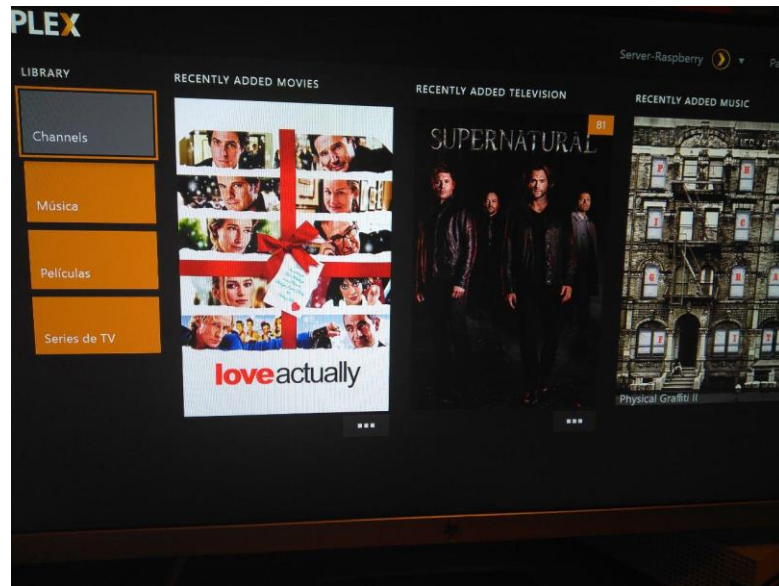
Plex es un gestor de contenidos multimedia que está basado en XBMC (Xbox Media Center) que se divide en dos partes. Tenemos el lado del Servidor y el lado del Cliente. Todo el contenido estaría almacenado en el servidor pudiendo almacenar tanto imágenes, audio y vídeos, permitiéndonos obtener información de éstos mismos a través de Internet si los archivos están nombrados correctamente como para que los servicios online sean capaces de ofrecernos los metadatos. Además, escaneará periódicamente todo el contenido almacenado en busca de nuevos cambios descargándose esta información también de manera periódica para los ficheros nuevos que obtengamos.

El servidor puede ser instalado en multitud de dispositivos. Tanto en PC con Windows, Linux o Macintosh, como en una Raspberry Pi, convirtiéndolo en un sistema muy versátil. En el cliente, el usuario debe ingresar su nombre de usuario y su contraseña y registrar su servidor. De esta manera podremos acceder desde una aplicación cliente a nuestros contenidos.



*Ilustración 5. Plex en el lado de Servidor desde una Raspberry pi*

Actualmente hay aplicaciones clientes para dispositivos de diversa índole (Smart TV, Videoconsolas, PCs, etcétera).



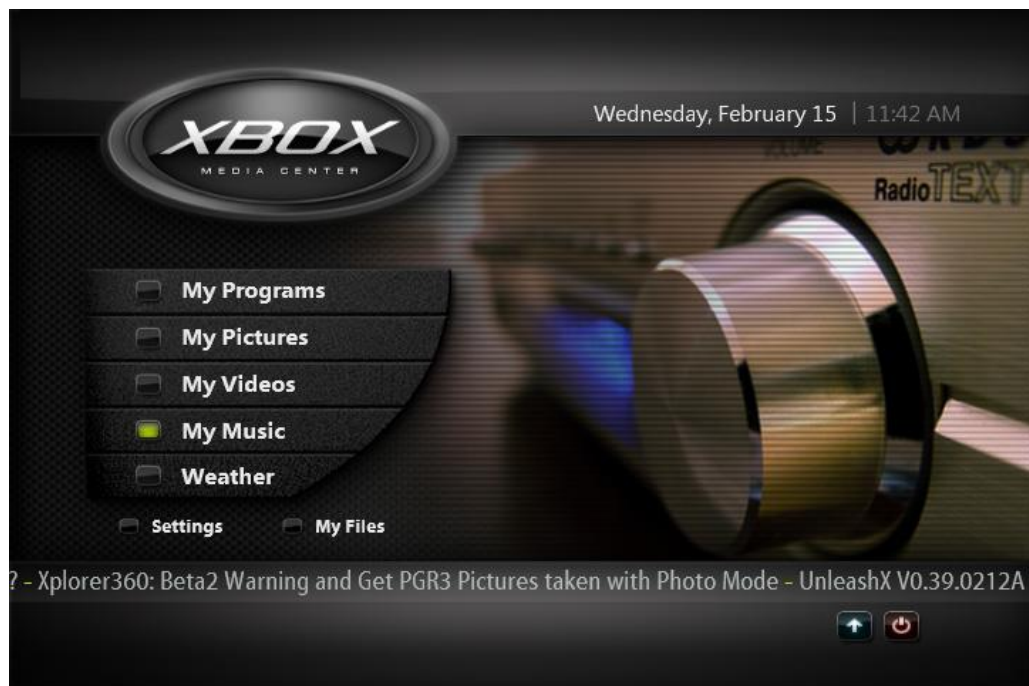
*Ilustración 6. Plex en el lado del cliente visto desde una Xbox One*

De esta manera habría un servidor que contendría una biblioteca multimedia en su almacenamiento interno y desde un cliente Plex que podría estar en la misma red local, o fuera de ésta, se haría uso de la susodicha biblioteca multimedia guardando el progreso de visualización de cada fichero, es decir, si desde un cliente Plex se ha reproducido un vídeo y se ha dejado en el instante 21m 12s, cuando abramos nuestra cuenta Plex desde otro cliente Plex podremos seguir reproduciendo ese vídeo en el instante que lo habíamos dejado con anterioridad. La única limitación que se podría tener es la velocidad de subida de la que disponga el usuario para ofrecer sus ficheros fuera de la red interna.

### 3.1.2. Kodi

Kodi es un denominado “media center” multiplataforma que nació en la “Scene” de la primera videoconsola de Microsoft, la llamada XBox aprovechando la potencia extra de la misma y a su propia arquitectura más parecida a la de un PC que la del resto de competidoras (Playstation 2 de Sony y GameCube de Nintendo), recibiendo su primer nombre “XBMP” (Xbox Media Player) que recibiría en los años 2002-2003. [2]

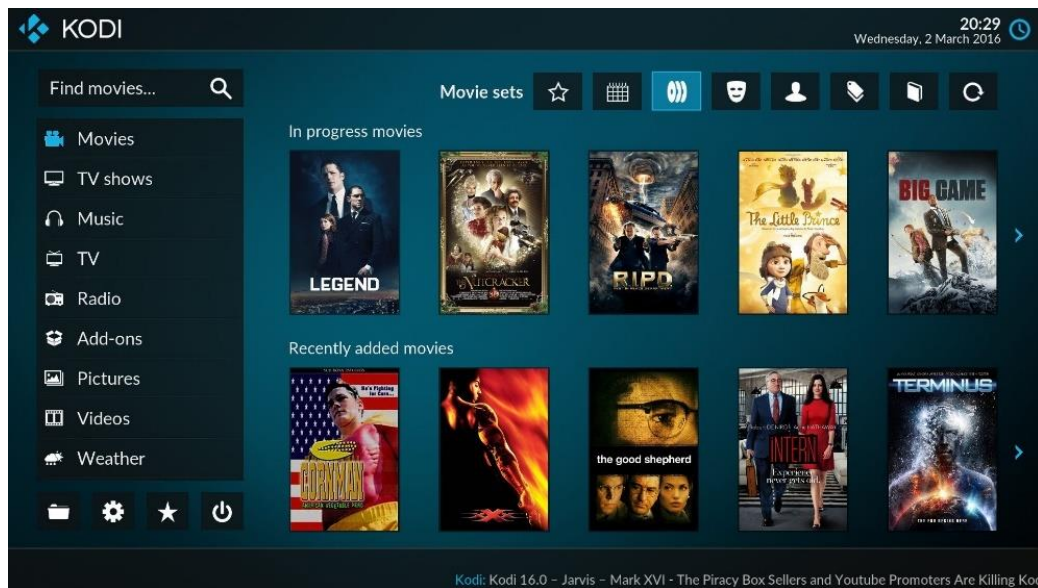
En los años 2004-2014 recibiría no solo su primera versión oficial, la 1.0.0 que fue lanzada el 29 de junio de 2004 sino que saldría del ecosistema de Microsoft y tendría un gran desarrollo, para así dar muchas más posibilidades mediante “Addons” cómo el soporte para iTunes y UPnP (Universal Plug and Play) en la versión final estable de XBMC 2.0.0 el 29 de Septiembre de 2006.



*Ilustración 7. Pantalla principal de XBMC corriendo en un Xbox*

A partir del año 2007 se empezó a realizar el port al sistema operativo Linux haciendo uso de OpenGL y SDL reemplazando a DirectX, que era lo que se utilizaba en la versión de Xbox del XBMC.

Ya en el año 2014, recibiría el nombre por el que hoy lo conocemos: Kodi, más exactamente: el 1 de agosto del 2014, habiéndose convertido en un gran reproductor de medios con un gran acceso a listas de contenido con el que poder ver tanto series y películas y con más de 900 addons.



*Ilustración 8. Pantalla principal de Kodi*

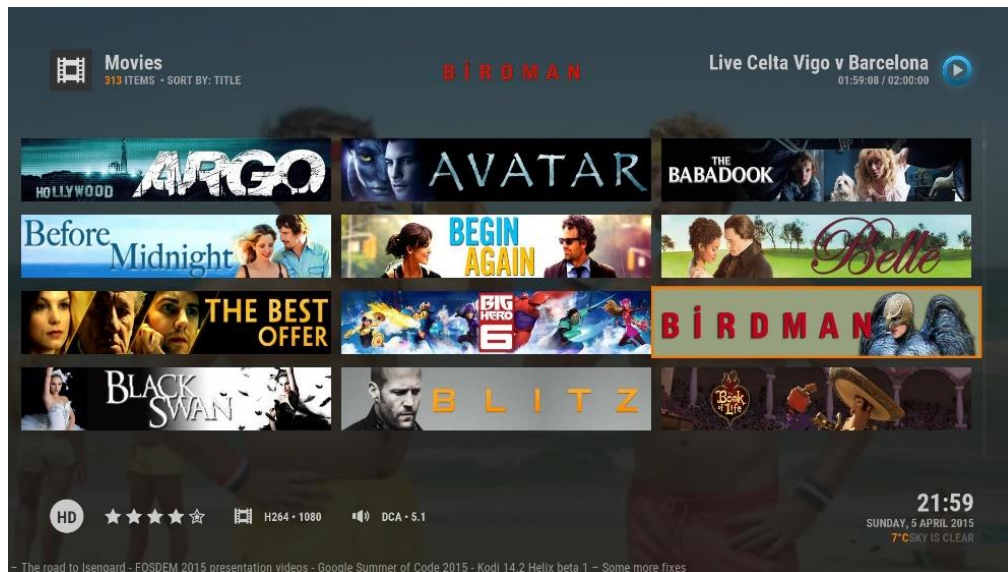
La principal diferencia que tendríamos con Plex, es que en Kodi no necesitamos ese ecosistema Cliente/Servidor para reproducir contenidos en Kodi: La misma máquina que contenga Kodi, será capaz de reproducir desde la misma a través de la reproducción de archivos contenidos en la misma máquina como a través de la reproducción de archivos mediante streaming a través de las ya citadas anteriormente: Listas de contenido que se pueden encontrar en la red.

### 3.1.3. Emby

Emby sería otro media center, pero con la principal diferencia con Plex, que tiene componentes de software propietario, de que es Open Source. También se basa en un sistema cliente-servidor incluyendo funcionalidades de TV en directo. Además, permite realizar streaming a dispositivos compatibles con DLNA (Digital Living Network Alliance), tiene control parental y herramientas para editar metadatos y buscar subtítulos. Además, se puede instalar Emby para Kodi, teniendo así la versatilidad de ambas aplicaciones en solo una. Es totalmente gratuito, pero con una suscripción de pago opcional desde 4,99\$ al mes tendremos características adicionales como la posibilidad de



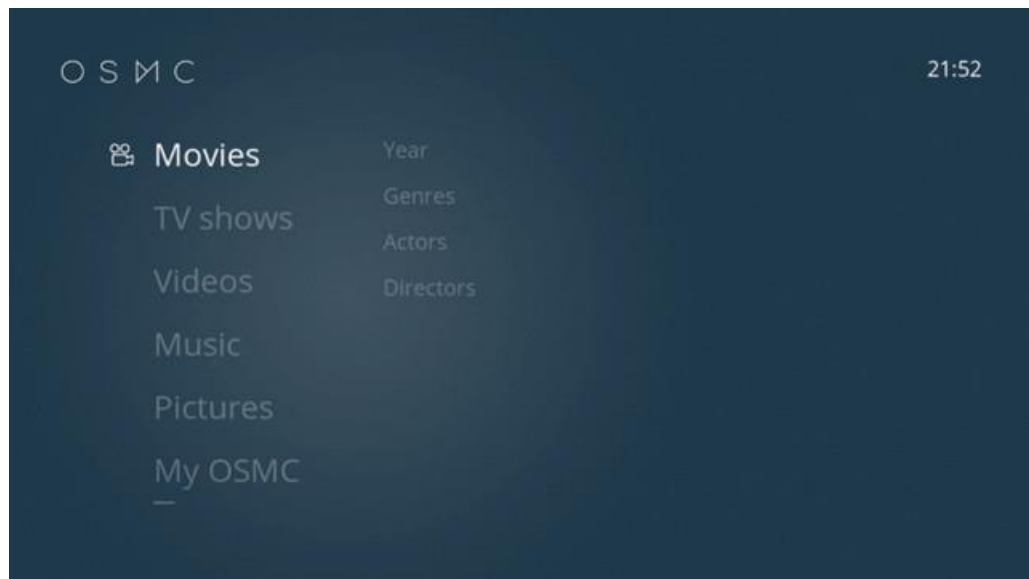
reproducir contenido de manera offline, backup de nuestra configuración del servidor, etc.



*Ilustración 9. Emby instalado en Kodi*

#### 3.1.4. OSMC

OSMC es la abreviatura de Open Source Media Center y es una distribución de Linux basada en Debian que nos trae Kodi a una variedad de dispositivos facilitándonos aún más su instalación y uso, pudiendo utilizar todos los complementos que tiene Kodi. Está disponible para los dispositivos Apple TV y Raspberry Pi siendo completamente gratuito.



*Ilustración 10. Pantalla Principal de OSMC*

### 3.1.5. MediaPortal

Es un fork de XBMC, antiguo nombre del media center Kodi y tiene como funcionalidad que la distingue de las demás la capacidad de grabación de vídeo digital (DVR), que permite ver, programar y grabar televisión en directo con facilidad, dando además soporte para tarjetas sintonizadoras de TV ofreciendo también la posibilidad de reproducir videos, música, etc.

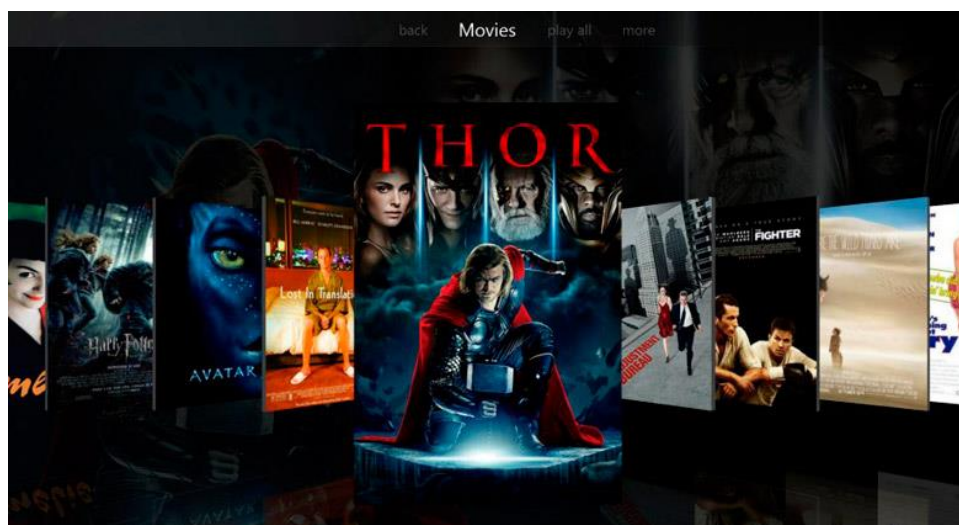
Tenemos la posibilidad de sintonizar emisoras de radio por Internet, capacidad de incluir plugins haciéndolo más versátil. Está solamente disponible para el Sistema Operativo de Microsoft: Windows, es completamente gratuito y actualmente no es tan popular como otras alternativas.



*Ilustración 11. Pantalla principal de MediaPortal*

### 3.1.6. JRiver

JRiver es un media center, y tiene como principal ventaja el control de la interfaz a través del ratón. Muchas de los media center que hemos visto anteriormente están pensados para ser utilizados en una televisión y por lo tanto su control a través de ratón es algo más tosco que en el caso de JRiver que está pensado para trabajar también mediante ratón. Está disponible para Windows, Linux y Mac OS X con un precio por plataforma de 49,99€ y un precio por su versión multiplataforma de 69,98€.



*Ilustración 12. Pantalla principal de JRiver*

### 3.1.7. Stremio

Stremio es un media center que trabaja de manera local disponible en Windows, Linux, Android, iOS y Mac soportando tanto la reproducción de archivos locales y de TV. En el apartado de streaming, éste también es capaz de realizar streaming de archivos multimedia mediante DLNA. Es completamente gratuito.

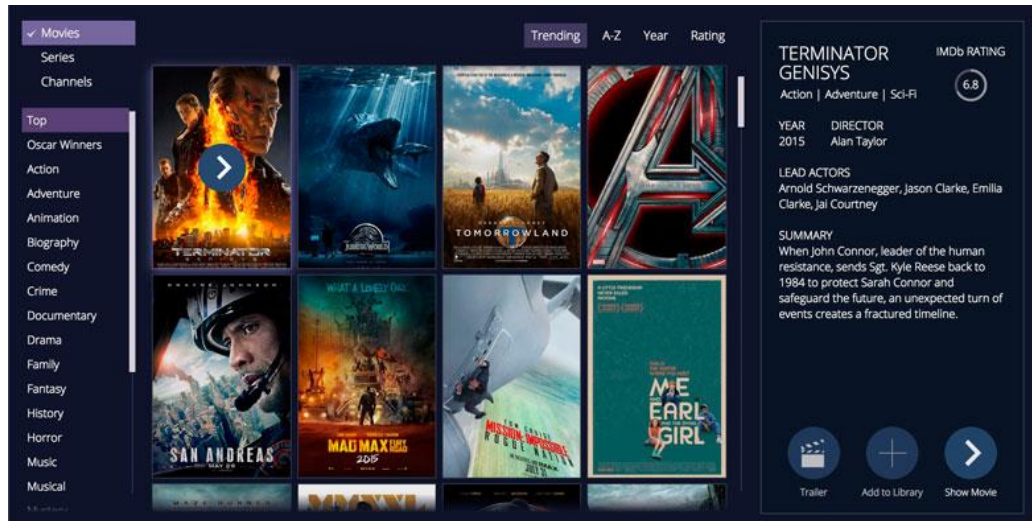


Ilustración 13. Pantalla principal de Stremio

### 3.1.8. Usher

Usher es un media center pensado para usarse en Mac, facilitando la administración de archivos en él mismo gestionando la biblioteca de iTunes. Está basado en el reproductor Quick Time, aplicación nativa del sistema operativo que llevan los Macs de Apple. Permite el streaming mediante DLNA, es completamente gratuito y solo está disponible para el sistema operativo de Apple [1]



*Ilustración 14. Pantalla principal de Usher*

Habiendo visto un poco todas las posibles opciones que hay para gestionar nuestros contenidos multimedia podríamos decir que nuestro principal competidor en el mercado sería Kodi y/o Plex que serían los sistemas con más fuerza ahora mismo en el mercado. Se aportará con respecto a estos sistemas la posibilidad de poder subir tus propios contenidos y que éstos sean accesibles por otros usuarios premium de la plataforma

## 3.2. Estado del Arte Tecnológico

A continuación, vamos a ver, en primer lugar, las características que priman sobre el resto de las demás a la hora de elegir para este TFG una tecnología, haremos un breve viaje por las distintas tecnologías que se han visto y por último se justificará la elección de una o varias tecnologías.

### 3.2.1. Estudio Preliminar

En el estudio preliminar se han revisado las siguientes tecnologías: Java, PHP, cordova, phonegap, HTML5, js, JSON, Ionic, Codeigniter y los siguientes gestores de base de datos dentro de las bases de datos relacionales: MySQL, PostgreSQL, Oracle. Para la elección de la tecnología a usar en este TFG me he basado en los siguientes puntos:

- **Versatilidad:** Sea funcional para la mayoría de las plataformas (PC, Android, iOS, etc)
- **Escalabilidad:** Dicha tecnología nos permita mejorar nuestro producto en el futuro con la mayor facilidad posible.
- **Depurador:** Su depurador tenía que ser potente debido a la cantidad de pruebas que se realizaría en el código para asegurarnos de que no existieran bugs en la versión comercial del mismo.
- **Utilidad:** Mirar otras opciones como pueden ser la confluencia de varios lenguajes de programación, facilidad de ver los cambios realizados en la aplicación a tiempo real, etc.

#### 3.2.1.1. Java

Java es un lenguaje de programación concurrente y orientado a objetos que apareció por primera vez en el año 1995. Este lenguaje de programación nos permite escribir el código y ejecutarlo en cualquier dispositivo gracias a su máquina virtual Java (JVM) sin tener que preocuparnos por la arquitectura de la computadora subyacente. Es

un lenguaje muy versátil, con una gran escalabilidad, pero en la actualidad existen opciones más potentes y orientadas a la multiplataforma y/o aplicaciones web. Por lo tanto, quedó descartada en pro de otras tecnologías.

#### *3.2.1.2. Cordova*

Apache Cordova, anteriormente conocido como “Phonegap” es un entorno de aplicaciones móviles que se ha popularizado en los últimos años. Permite construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5 y Javascript en vez de utilizar las APIs de cada plataforma (Android, iOS o Windows Phone). Nos permite construir aplicaciones híbridas; no son aplicaciones nativas, ya que toda la representación gráfica la realiza a través de vistas web en vez de utilizar los frameworks nativos, pero tampoco son aplicaciones Web, estando empaquetadas como aplicaciones y teniendo acceso a las APIs nativas del dispositivo.

#### *3.2.1.3. PHP*

PHP es el acrónimo de Hypertext Preprocessor. Apareció por primera vez en el año 1995 y es un lenguaje de programación diseñado para el desarrollo web de contenido dinámico y que puede ser incrustado en HTML. El código es interpretado por un servidor web con un módulo procesador de PHP, el cual genera la página web resultante. Permite la conexión a diferentes tipos de servidores de bases de datos tanto SQL como NoSQL, tales como MySQL, PostgreSQL, ODBC, MongoDB, etc. También puede ser ejecutado en la mayoría de sistemas operativos. Destacar, por último, que es un lenguaje libre, y esto lo hace una alternativa de fácil acceso para todos.

#### 3.2.1.4. *JavaScript*

JavaScript, abreviado JS, es un lenguaje de programación interpretado, que apareció por primera vez en el año 1995, dialecto del estándar ECMAScript. Se utiliza principalmente en el lado del cliente realizando mejoras en la interfaz de usuario y páginas web dinámicas. Actualmente, todos los navegadores modernos soportan completamente ECMAScript lo cual le da una grandísima versatilidad a este lenguaje de programación.

#### 3.2.1.5. *JSON*

JSON (JavaScript Object Notation) es un formato para el intercambio de datos. Nació como una alternativa a XML. Básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. Una de sus mayores ventajas, es que puede ser leído por cualquier lenguaje de programación y, por lo tanto, puede ser usado para el intercambio de información para distintas tecnologías.

#### 3.2.1.6. *CodeIgniter*

CodeIgniter es un poderoso Framework de PHP construida para desarrolladores que necesitan una herramienta simple y elegante para crear aplicaciones web llenas de funcionalidades. Es capaz de trabajar con la mayoría de entornos o servidores, incluso en sistemas de alojamiento compartido, dónde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración. Además, el núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.



#### 3.2.1.7. *Ionic*

Ionic es una completa SDK de código abierto para el desarrollo de aplicaciones híbridas. Está desarrollado sobre AngularJS y Apache Cordova. Ionic nos proporciona herramientas y servicios para desarrollar aplicaciones híbridas usando tecnologías web como CSS, HTML5 y SASS. Usa distintas tecnologías que garantizan que las aplicaciones desarrolladas con ionic sean rápidas y escalables. Además destacar que tiene la ausencia de la tecnología JQuery y tiene una potente interfaz de línea de comandos desde donde se compilará y creará las aplicaciones.

#### 3.2.1.8. *MySQL*

MySQL es un sistema gestor de base de datos relacional multihilo y multiusuario, es decir, que se pueden tener abiertas varias instancias de una misma base de datos y hacer cambios en una base de datos por varios usuarios al mismo tiempo. Nació en el año 1995 y en la actualidad es la base de datos de código abierto más popular del mundo, aunque también se puede comercializar una licencia comercial si una empresa así lo necesita, y ésta sería dada por Oracle Corporation. Está escrita en C/C++. Es muy utilizado en aplicaciones web, como Wordpress, Joomla y suele verse muy ligado a PHP. MySQL es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

#### 3.2.1.9. *PostgreSQL*

PostgreSQL es un sistema gestor de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad. Un fallo en uno de los procesos no afectará al resto y el sistema seguirá funcionando. Es capaz de hacer copias de seguridad en caliente (Online/hot Backus) y tiene acceso encriptado via SSL.

### 3.2.2. Elección de materiales y métodos

Las tecnologías usadas en éste TFG han sido las siguientes: Ionic, AngularJS, HTML, CSS, JavaScript, JSON, el framework CodeIgniter y el sistema gestor de base de datos MySQL, debido a que son tecnologías que, aun siendo potentes por separado, en su conjunto crean una herramienta mucho más sólida, versátil y escalable. Un ejemplo rápido explicará este razonamiento por sí solo.

Pongamos que queremos acceder a unos contenidos de una de las tablas de nuestra base de datos y mostrarlas de manera dinámica en nuestra aplicación móvil. En nuestro servidor tendremos montado CodeIgniter que no solo hará la petición al SGBD, sino que lo codificará en JSON permitiendo que desde AngularJS que es uno de los lenguajes de programación incluidos en Ionic podamos mostrar los datos recibidos dinámicamente y trabajar con ellos desde nuestra aplicación móvil añadiéndole funcionalidades a través de JavaScript.

### 3.3. Estado del arte legislativo

Es de gran importancia conocer la legislación vigente en los países en los que la aplicación vaya a ser distribuida con el objetivo de evitar problemas de violación de leyes. Por motivos de desconocimiento de las leyes de otros países, la aplicación inicialmente solo se distribuiría en España.

Al tratarse de una aplicación que tendrá almacenados datos e información de una gran cantidad de usuarios, los usuarios que se registren en nuestra aplicación deberán ser informados conforme a lo previsto en la Ley Orgánica 15/1999 de 13 de diciembre de Protección de datos de carácter personal, concretamente en el apartado 6.1: El tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco del afectado, salvo que la ley disponga otra cosa.

Además, al ser una aplicación en el que los usuarios podrán subir su biblioteca multimedia, se le avisará que debido a la Ley de Propiedad Intelectual se podría proceder a retirar contenido subido a la plataforma que no haya sido consentido por el autor del susodicho.

Para evitar que se den problemas por desconocimiento de las partes, se informaría sobre ello en el espacio en Google Play donde se distribuiría la aplicación.

### 3.4. Análisis del Sistema

En este capítulo se van a dar a conocer el proceso realizado para describir los objetivos que se han descrito en el apartado 2.1 del trabajo fin de grado. Además, se propondrá la solución, con las limitaciones que poseen los dispositivos disponibles siempre presente, y por último se detallará el proceso de diseño de la aplicación.

#### 3.4.1. Definición del sistema

En Ingeniería del Software tenemos varios tipos de requisitos, en este capítulo vamos a detallar los requisitos funcionales y no funcionales de los que va a constar la aplicación. Se puede definir un requisito como toda *condición o capacidad necesaria para el usuario para que éste sea capaz de resolver un problema o conseguir un objetivo determinado*. [4] [5] [6]

En todo requisito funcional y no funcional vamos a tener una serie de entradas que entrará en nuestro sistema y dará una cierta salida o salidas.

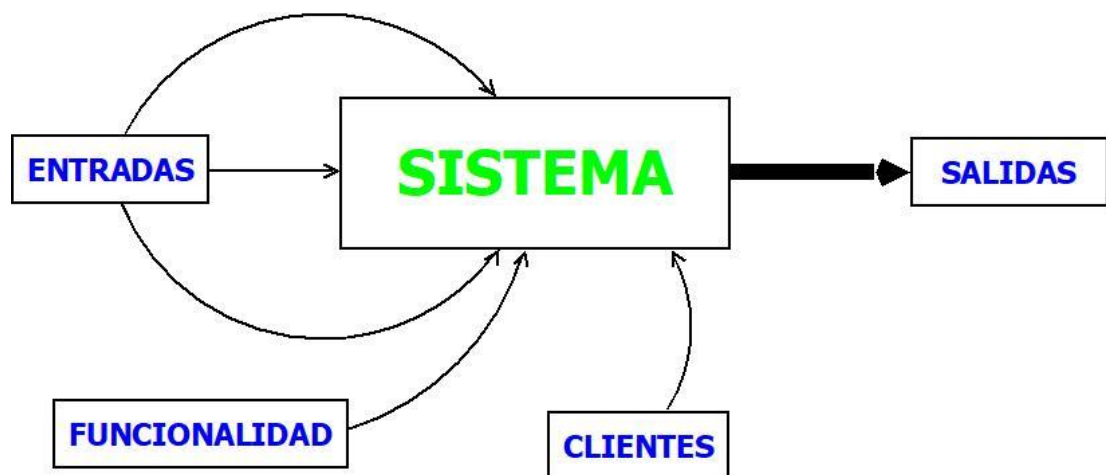


Ilustración 15. Definición de Requisito

Los requisitos pueden dividirse en:

- **Requisitos de usuario:** Se pueden definir como frases en lenguaje natural o diagramas de los servicios que se espera que va a ofrecer el sistema y las restricciones que tendrá el mismo
- **Requisitos de sistema:** Son una descripción más detallada de cada uno de los servicios que se van a proporcionar y de sus restricciones. A su vez, los requisitos de sistema pueden dividirse en requisitos funcionales, no funcionales y de dominio.
  - **Requisitos funcionales:** Estos definen lo que debe hacer el sistema o los servicios que debe proporcionar el sistema.
  - **Requisitos No funcionales:** Estos definen las restricciones de los servicios del sistema o las funciones que ofrece el sistema.
  - **Requisitos de dominio:** Estos reflejan características del dominio de la aplicación. Por ejemplo, el tipo de datos con el que la aplicación trabajará.

Las tablas que se van a utilizar en este apartado van a contar con los siguientes campos:

- **Identificador:** Permitirá reconocer cada requisito de manera clara.

La nomenclatura a seguir será: RSF-XX para los requisitos funcionales y RSNF-XX para los requisitos no funcionales donde X se corresponderá con un dígito numérico empezando por 01.

- **Nombre:** Describirá de manera clara y concisa el objetivo del requisito.
- **Descripción:** Detallará en que consiste el requisito de manera concisa.
- **Fuente:** Especificará de dónde proviene el requisito.

- **Necesidad:** Especificará la importancia de cada requisito en el proyecto y puede tomar los siguientes valores: importante, deseable u opcional.
- **Prioridad:** Especificará la urgencia con la que debe cumplirse el requisito. Este campo tendrá los siguientes valores: 1,2,3,4 y 5 siendo el 5 la prioridad más alta del requisito en la aplicación y el 1 la prioridad más baja en la aplicación del requisito.
- **Claridad:** Especificará si el requisito se entiende por si solo o no si fuera el caso. Puede tomar los siguientes valores: 1,2,3,4 y 5 siendo el 5 el valor más alto indicando que el requisito se entiende por si solo.
- **Estabilidad:** Especificará si el requisito variará a lo largo de la vida del proyecto o no. Los posibles valores de este campo serán: estable o inestable
- **Verificabilidad:** Especificará el cumplimiento del requisito. Se trataría como una comprobación del mismo. Puede tomar los siguientes valores: 1,2,3,4 y 5 siendo el 5 el valor más alto.
- **Relaciones:** Especificará de que requisitos de Software o conjunto de requisitos de Software son de los que proviene el requisito.

A continuación, se mostrará un ejemplo de la tabla a utilizar y se detallarán los requisitos funcionales y los requisitos no funcionales de los que va a constar la aplicación.

RSF-XX	
Nombre	
Descripción	

<b>Fuente</b>
<b>Prioridad</b>
<b>Estabilidad</b>
<b>Relaciones</b>
<b>Necesidad</b>
<b>Claridad</b>
<b>Verificabilidad</b>

Tabla 4. Ejemplo de Tabla de Requisito de Software

3.4.2. Definición de requisitos funcionales

3.4.2.1. RSF-01. Registro de una nueva cuenta.

<b>RSF-01</b>	
<b>Nombre</b>	Registro de una nueva Cuenta
<b>Descripción</b>	El sistema deberá tener la capacidad de registrar nuevos usuarios en la aplicación. Este registro se realizará mediante un formulario.
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 5. Requisitos de Software. RSF-01 – Registro de una Nueva Cuenta.

3.4.2.2. *RSF-02. Acceso Cliente*

RSF-02	
<b>Nombre</b>	Acceso Cliente
<b>Descripción</b>	La aplicación constará de un login desde el cual el usuario rellenará un pequeño formulario para validar su login.
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

*Tabla 6. Requisitos de Software. RSF-02 – Acceso Cliente*

3.4.2.3. *RSF-03. Acceso Administrador.*

RSF-03	
<b>Nombre</b>	Acceso Administrador
<b>Descripción</b>	La aplicación constará de un login especial para el administrador de la aplicación.
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5

<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 7. Requisitos de Software. RSF-03 – Acceso Administrador

3.4.2.4. RSF-04. Recuperar contraseña.

RSF-04	
<b>Nombre</b>	Recuperar Contraseña
<b>Descripción</b>	La aplicación nos permitirá recuperar nuestra contraseña en caso de perderla, seamos usuario normal o administrador.
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	RSF-02                      RSF-03
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 8. Requisitos de Software. RSF-04 – Recuperar Contraseña

3.4.2.5. RSF-05. Consultar datos de acceso

RSF-05	
<b>Nombre</b>	Consultar datos de acceso
<b>Descripción</b>	La aplicación nos permitirá consultar nuestros datos de acceso, seamos usuario normal o administrador
<b>Fuente</b>	Cliente



<b>Prioridad</b>	5	
<b>Estabilidad</b>	Estable	
<b>Relaciones</b>	RSF-02	RSF-03
<b>Necesidad</b>	5	
<b>Claridad</b>	5	
<b>Verificabilidad</b>	5	

Tabla 9. Requisitos de Software. RSF-05 -- Consultar datos de acceso

#### 3.4.2.6. RSF-06 – Modificar datos de Acceso

<b>RSF-06</b>		
<b>Nombre</b>	Modificar datos de Acceso	
<b>Descripción</b>	La aplicación nos permitirá modificar nuestros datos de acceso, seamos usuario normal o administrador	
<b>Fuente</b>	Cliente	
<b>Prioridad</b>	5	
<b>Estabilidad</b>	Estable	
<b>Relaciones</b>	RSF-02	RSF-03
<b>Necesidad</b>	5	
<b>Claridad</b>	5	
<b>Verificabilidad</b>	5	

Tabla 10. Requisitos de Software. RSF-06 – Modificar datos de acceso

3.4.2.7. *RSF-07 – Cerrar Sesión*

RSF-07	
<b>Nombre</b>	Cerrar Sesión
<b>Descripción</b>	La aplicación nos permitirá cerrar nuestra sesión, seamos usuario normal o administrador.
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	5
<b>Relaciones</b>	RSF-02                      RSF-03
<b>Necesidad</b>	3
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

*Tabla 11. Requisitos de Software. RSF-07 – Cerrar Sesión*

3.4.2.8. *RSF-08 – Mirar Alertas.*

RSF-08	
<b>Nombre</b>	Mirar Alertas
<b>Descripción</b>	La aplicación nos permitirá ver las posibles violaciones de copyright existentes en el sistema, solo siendo administrador.
<b>Fuente</b>	Jefe de Proyecto
<b>Prioridad</b>	2
<b>Estabilidad</b>	Inestable
<b>Relaciones</b>	

<b>Necesidad</b>	3
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 12. Requisitos de Software. RSF-08 – Mirar Alertas

3.4.2.9. RSF-09 – Búsqueda por Nombre

<b>RSF-09</b>	
<b>Nombre</b>	Búsqueda por Nombre
<b>Descripción</b>	La aplicación nos permitirá buscar nuestros ficheros por nombre
<b>Fuente</b>	Cliente
<b>Prioridad</b>	5
<b>Estabilidad</b>	Inestable
<b>Relaciones</b>	
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 13. Requisitos de Software. RSF-09 -- Búsqueda por nombre

3.4.2.10. RSF-10 - Reproducción

RSF-10	
Nombre	Reproducción
Descripción	La aplicación nos permitirá reproducir el fichero seleccionado
Fuente	Cliente
Prioridad	5
Estabilidad	Estable
Relaciones	
Necesidad	5
Claridad	5
Verificabilidad	5

Tabla 14. Requisitos de Software. RSF-10 -- Reproducción

3.4.2.11. RSF-11 – Subida de Ficheros

RSF-10	
Nombre	Subida de Ficheros
Descripción	La aplicación nos permitirá subir a la aplicación el fichero seleccionado, permitiendo elegir el contenido como Publico (accesible por usuarios Premium también) o como Privado, solo accesible por el usuario que lo suba.
Fuente	Cliente
Prioridad	5
Estabilidad	Estable
Relaciones	

<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 15. Requisitos de Software. RSF-11 – Subida de Archivos

### 3.4.3. Definición de requisitos no funcionales

#### 3.4.3.1. RSNF-01. Navegador.

<b>RSNF-01</b>	
<b>Nombre</b>	Navegador
<b>Descripción</b>	La aplicación deberá ser compatible con los navegadores web más comunes de ordenador (Mozilla Firefox, Google Chrome, Opera y Safari) y el navegador web integrado en el iPhone (Safari).
<b>Fuente</b>	Jefe de Proyecto
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 16. Requisito de Software. RSNF-01 – Navegador

3.4.3.2. RSNF-02. Idioma.

RSNF-02	
Nombre	Idioma
Descripción	El idioma de la aplicación deberá ser Castellano, ya que el principal mercado para el que está pensado será el español
Fuente	Cliente
Prioridad	5
Estabilidad	Estable
Relaciones	Ninguna
Necesidad	5
Claridad	5
Verificabilidad	5

Tabla 17. Requisito de Software. RSNF-02 – Idioma

3.4.3.3. RSNF-03. Utilización de la interfaz.

RSNF-03	
Nombre	Utilización de la interfaz
Descripción	La interfaz deberá ser fácil de usar e intuitiva.
Fuente	Cliente
Prioridad	5
Estabilidad	Estable
Relaciones	Ninguna
Necesidad	5
Claridad	5

**Verificabilidad** 5

Tabla 18. Requisito de Software. RSNF-03 – Utilización de la interfaz

3.4.3.4. RSNF-04. Comprobación de campos obligatorios.

RSNF-04	
<b>Nombre</b>	Comprobación de campos obligatorios
<b>Descripción</b>	Se deberá comprobar que están rellenos todos los campos obligatorios de todos los formularios.
<b>Fuente</b>	Jefe de Proyecto
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 19. Requisito de Software. RSNF-04 – Comprobación de campos obligatorios

3.4.3.5. RSNF-05. Unicidad en usuarios registrados.

RSNF-05	
<b>Nombre</b>	Comprobación de que haya unicidad en usuarios registrados
<b>Descripción</b>	No se podrá registrar dos usuarios que tengan el mismo correo electrónico.
<b>Fuente</b>	Jefe de Proyecto
<b>Prioridad</b>	5

<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 20. Requisito de Software. RSNF-05 – Comprobación de que haya unicidad en usuarios registrados.

#### 3.4.3.6. RSNF-06. Uso de Validadores.

<b>RSNF-06</b>	
<b>Nombre</b>	Uso de Validadores
<b>Descripción</b>	Comprobar que no se utilizan caracteres no permitidos en los formularios: \,/,;*,&,”,<,>, .
<b>Fuente</b>	Jefe de Proyecto
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 21. Requisito de Software. RSNF-06 – Uso de Validadores



3.4.3.7. RSNF-07. Seguridad.

RSNF-07	
<b>Nombre</b>	Seguridad
<b>Descripción</b>	Los datos de cada usuario se almacenarán cifrados en la base de datos
<b>Fuente</b>	Cliente y Jefe de Proyecto
<b>Prioridad</b>	5
<b>Estabilidad</b>	Estable
<b>Relaciones</b>	Ninguna
<b>Necesidad</b>	5
<b>Claridad</b>	5
<b>Verificabilidad</b>	5

Tabla 22. Requisito de Software. RSNF-07 -- Seguridad

### 3.5. Diseño del Sistema

El diseño del sistema en Ingeniería de Software, puede ser dividido en varios apartados, detallando en primer lugar el tipo de arquitectura a usar en el sistema, seguido por la lógica de negocio donde se establecen todas las reglas que deben cumplirse y finalizando con la lógica de datos donde residen los datos y donde se accede a los mismos. Suele estar formada por un gestor base de datos que es el que realiza el almacenamiento de datos.

#### 3.5.1. Descripción de la Arquitectura del sistema y justificación de las tecnologías utilizadas

Aquí se analizará la arquitectura que debe tomar la aplicación. En nuestro caso se va a hacer uso de una Arquitectura Cliente-Servidor. Se puede definir la Arquitectura Cliente/Servidor como una arquitectura que separa el procesamiento entre clientes y servidores dentro de una misma red. En una Arquitectura Cliente/Servidor podemos diferenciar 3 componentes: [8] [9] [10] [11]

- **Cliente:** Los equipos que solicitan datos o servicios al Servidor. Al software que hay en la parte de cliente se le conoce como Front-End y ofrecen una interfaz gráfica amigable al usuario final.
- **Servidor:** El equipo donde reside la base de datos y que responde a las solicitudes del cliente. Al software que hay en la parte del servidor se le conoce como Back-End. Un servidor puede aceptar un gran número de conexiones simultáneas de clientes y no interactúa directamente con los usuarios finales.
- **Red:** El medio que transporta estas peticiones y respuestas.

Este modelo de Arquitectura tiene las siguientes ventajas, por las cuales se ha elegido para realizar la aplicación:

- Nos permite poder utilizar entornos heterogéneos, es decir, la plataforma no tiene porque ser la misma. El cliente puede estar en un PC, en un móvil Android, etc.
- Tiene una gran escalabilidad. Al incrementar los recursos empleados el rendimiento del sistema aumenta proporcionalmente, por ejemplo,

tenemos la capacidad de actualizar el sistema informático si éste en algún momento se viera congestionado.

- Nos ofrece portabilidad. Podemos, por ejemplo, migrar el servidor a un equipo más potente sin mucho esfuerzo.

Al ser una aplicación que se realiza pensando en que pasado un tiempo, puede tener un gran número de usuarios, un gran número de ficheros almacenados en el servidor y con la consiguiente sobrecarga de uso de recursos, era primordial usar una arquitectura que nos permitiera poder aumentar los recursos de la misma sin verse afectado el servicio final.

### 3.5.2. Diseño de la lógica de negocio

#### 3.5.2.1. Casos de uso

Los casos de uso representan la forma en como un Actor se relaciona con el sistema en desarrollo, cómo los elementos interactúan, de que forma interactúan entre ellos y de que tipo son. Un diagrama de casos de uso tiene los siguientes elementos: [7]

- Actor
- Casos de Uso
- Relaciones de Uso, Herencia y Comunicación

##### 3.5.2.1.1. Actor

Un actor es un rol que un usuario tiene con respecto al sistema que se va a desarrollar. Se dice que es un rol porque un actor no necesariamente tiene que representar a una persona en particular, sino una acción que se realiza frente al sistema.

En este caso vamos a tener los siguientes actores:

- **Usuario:** será toda persona que ejecuta la aplicación.
- **Usuario registrado:** Toda aquella persona que se ha registrado previamente creando una cuenta y que accede a la aplicación mediante su correo electrónico y contraseña.
- **Administrador:** Aquel encargado de eliminar contenido que no cumpla las normas de la aplicación de cualquier usuario.

#### 3.5.2.1.2. Caso de Uso

Un caso de uso es una operación o tarea que se ejecuta tras una orden de un agente externo, ya sea desde la petición de un actor o desde la invocación a través de otro caso de uso. Más adelante, vamos a ver los distintos casos de uso que va a tener nuestra aplicación.

#### 3.5.2.1.3. Relaciones

Por último, como último elemento, pero no menos importante, tenemos las relaciones. Aquí tenemos distintos tipos de relaciones que vamos a detallar a continuación.

##### 3.5.2.1.3.1. Asociación

Esta sería el tipo de relación más básica que tenemos en un caso de uso e indica la invocación desde un caso de uso o desde un actor a otra operación. Se denota con una flecha simple.

##### 3.5.2.1.3.2. Dependencia o Instanciación

Esta relación se da cuando una clase depende de otra, se instancia, se crea. Se denota con una flecha punteada.

##### 3.5.2.1.3.3. Generalización

Esta relación es de las más usadas y tiene dos funciones dependiendo de su estereotipo. Su primera función es la de **Uso** (<<uses>>) o de **Herencia** (<<extends>>). Esta relación se usa exclusivamente para casos de uso y no para actores.

- **Extends:** Se utiliza cuando un caso de uso es similar a otro. Tiene características similares.
- **Uses:** Se utiliza cuando hay un conjunto de características que son similares en más de un caso de uso.

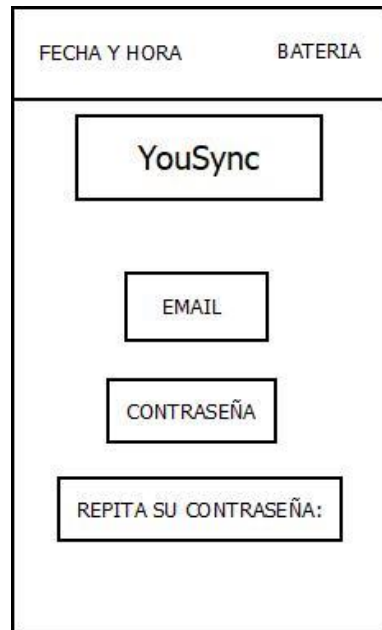
### 3.5.2.1.4. Casos de uso de YouSync

#### 3.5.2.1.4.1. Casos de uso del modulo del Usuario

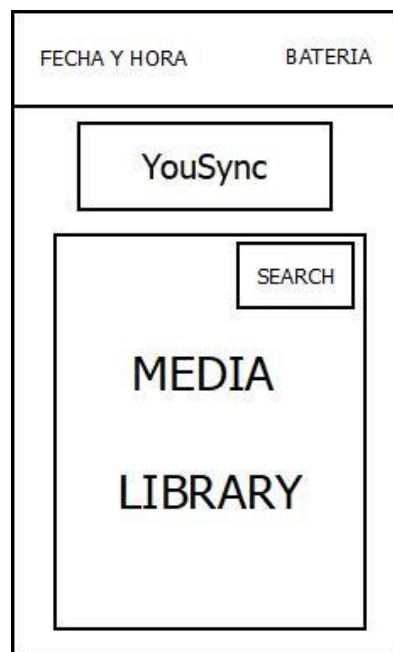
##### 3.5.2.1.4.1.1. *Crear una cuenta*

<b>Caso de uso 1</b>	<b>Crear una cuenta</b>
<b>Resumen de la Funcionalidad</b>	Permite al usuario crear una cuenta nueva
<b>Casos de uso relacionados</b>	
<b>Actores</b>	Usuario
<b>Precondición</b>	No puede existir una Cuenta con otro correo electrónico igual
<b>Postcondición</b>	Se creará la cuenta y se le redirigirá al usuario a la pantalla principal de la aplicación
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. La aplicación muestra la pantalla principal</li><li>2. El usuario selecciona la opción “Registro”</li><li>3. El sistema mostrará la pantalla de “Registro” con el formulario para ello</li><li>4. El usuario introduce el nombre, correo electrónico, contraseña, repite la contraseña y, por último, pulsa “Registrarse”</li><li>5. El sistema crea la cuenta nueva y registra los datos del usuario en la base de datos y redirige al usuario a la página “Multimedia” con los ficheros del usuario</li></ol>
<b>Extensiones</b>	<ol style="list-style-type: none"><li>2a. El usuario elige salir de la aplicación<ol style="list-style-type: none"><li>2a1. Se finaliza el caso de uso</li></ol></li><li>4a. El usuario introduce alguno de los datos de forma incorrecta.<ol style="list-style-type: none"><li>4a1. El sistema muestra un mensaje de error.</li></ol></li></ol>

Tabla 23. Caso de Uso 1



*Ilustración 16. Diseño del caso de uso 1*

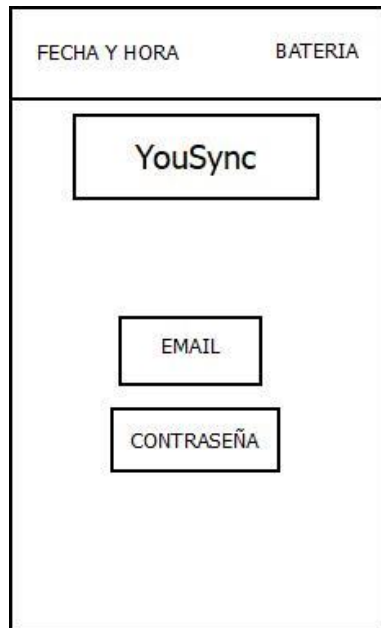


*Ilustración 17. Resultado Final del caso de uso 1*

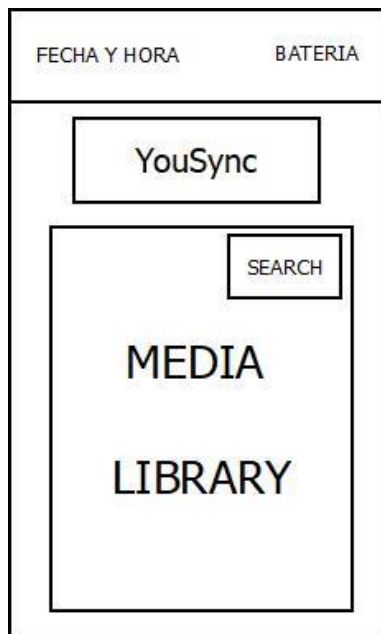
3.5.2.1.4.1.2. Acceder a la aplicación

Caso de uso 2	Acceder a la aplicación
Resumen de la Funcionalidad	Permite al usuario acceder a la aplicación
Casos de uso relacionados	Crear una cuenta
Actores	Usuario registrado
Precondición	El usuario debe haberse creado una cuenta anteriormente y haber accedido con sus datos
Postcondición	Se mostrará la pantalla “Multimedia” del usuario registrado
Escenario de éxito	<ol style="list-style-type: none"><li>1. El sistema muestra la pantalla principal</li><li>2. El usuario registrado introduce sus datos: correo electrónico y contraseña y pulsa en “Login”</li><li>3. El sistema comprueba que los datos son correctos y redirige al usuario a la pantalla “Multimedia” del usuario registrado</li></ol>
Extensiones	<p>2a. El usuario introduce alguno de los datos de forma incorrecta o deja un dato sin rellenar.</p> <p>2a1. El sistema muestra un mensaje de error.</p> <p>3a. El sistema no detecta el conjunto de datos de correo electrónico y contraseña del usuario en la base de datos.</p> <p>3a1. El sistema muestra un mensaje de error diciendo que no hay un usuario registrado con ese correo electrónico y/o contraseña</p>

Tabla 24. Caso de Uso 2



*Ilustración 18. Diseño del caso de uso 2*



*Ilustración 19. Resultado Final del caso de uso 2*



3.5.2.1.4.1.3. *Recuperar la contraseña*

Caso de uso 3	Recuperar la contraseña
Resumen de la Funcionalidad	Permite al usuario recuperar su contraseña en caso de haberla olvidado
Casos de uso relacionados	
Actores	Usuario
Precondición	El usuario debe haberse creado una cuenta anteriormente
Postcondición	El sistema enviará al correo electrónico del usuario una nueva contraseña para acceder a la aplicación
Escenario de éxito	<ol style="list-style-type: none"><li>1. El sistema muestra la pantalla principal</li><li>2. El usuario selecciona la opción “Recuperar mi contraseña”</li><li>3. El sistema muestra un formulario para introducir el correo electrónico del usuario donde se enviará la nueva contraseña</li><li>4. El usuario introduce su correo electrónico y pulsa “Recuperar Contraseña”</li><li>5. El sistema comprueba que tiene en la base de datos ese correo electrónico y dirige al usuario a la pantalla “Login” del sistema.</li></ol>
Extensiones	<ol style="list-style-type: none"><li>3a. El usuario introduce una dirección de correo electrónico errónea<ol style="list-style-type: none"><li>3a1. El sistema muestra un mensaje de error.</li></ol></li><li>4a. El sistema no puede acceder a la base de datos en ese momento.<ol style="list-style-type: none"><li>4a1. El sistema muestra un mensaje de error diciendo que no se puede recuperar la contraseña en ese instante</li></ol></li></ol>

Tabla 25. Caso de Uso 3



Ilustración 20. Diseño del caso de uso 3

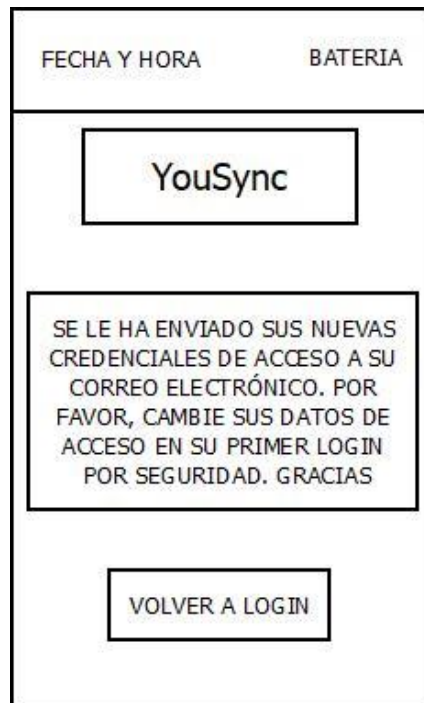


Ilustración 21. Resultado Final del caso de uso 3

3.5.2.1.4.1.4. Consultar datos del usuario.

Caso de uso 4	Consultar datos del usuario
Resumen de la Funcionalidad	Permite al usuario consultar sus datos de acceso al sistema
Casos de uso relacionados	
Actores	Usuario registrado
Precondición	El usuario debe haberse creado una cuenta anteriormente y haber accedido al sistema con sus datos de acceso
Postcondición	El sistema muestra los datos de acceso del usuario
Escenario de éxito	<ol style="list-style-type: none"><li>1. El usuario registrado selecciona “Mi perfil”</li><li>2. El sistema muestra los datos de acceso</li></ol>
Extensiones	<ol style="list-style-type: none"><li>2a. El sistema no puede acceder a la base de datos en ese momento.<ol style="list-style-type: none"><li>2a1. El sistema muestra un mensaje de error diciendo que no se puede recuperar los datos de acceso del usuario registrado en ese instante</li></ol></li></ol>

Tabla 26. Caso de uso 4

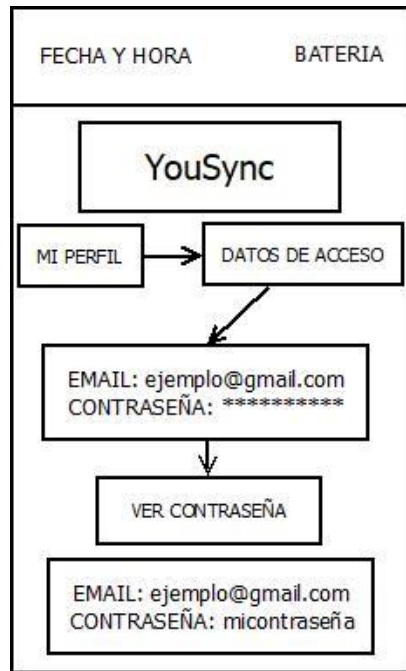


Ilustración 22. Diseño del caso de uso 4



Ilustración 23. Resultado Final del caso de uso 4

3.5.2.1.4.5. *Modificar datos de acceso de un usuario registrado.*

<b>Caso de uso 5                      Modificar datos de acceso de un usuario registrado</b>	
<b>Resumen de la Funcionalidad</b>	Permite al usuario modificar sus datos de acceso al sistema
<b>Casos de uso relacionados</b>	Consultar datos del usuario
<b>Actores</b>	Usuario registrado
<b>Precondición</b>	El usuario debe consultar los datos de acceso
<b>Postcondición</b>	El sistema actualiza los datos de acceso nuevos del usuario registrado
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. El usuario registrado selecciona modifica uno de los datos de acceso: Correo electrónico y/o contraseña y pulsa en “Modificar datos de acceso”</li><li>2. El sistema actualiza los datos de acceso del usuario registrado y redirige al usuario registrado a la página “Login” para que ingrese con los nuevos datos de acceso.</li></ol>
<b>Extensiones</b>	<ol style="list-style-type: none"><li>1a. El usuario registrado introduce algún dato de forma errónea.<ol style="list-style-type: none"><li>1a1. El sistema muestra un mensaje de error diciendo que hay un dato introducido de forma errónea</li></ol></li><li>2a. El sistema no puede conectar con la base de datos y actualizar los datos de acceso del usuario.<ol style="list-style-type: none"><li>2a1. El sistema muestra un mensaje indicando que no se pueden modificar los datos de acceso del usuario registrado en ese instante</li></ol></li></ol>

Tabla 27. *Caso de uso 5*

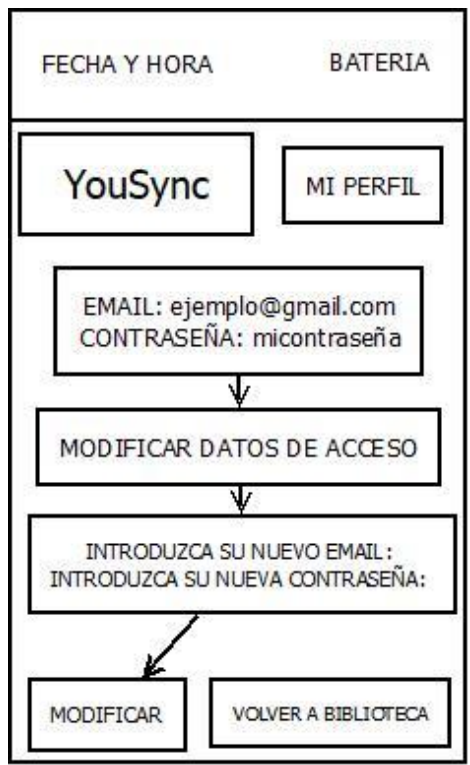


Ilustración 24. Diseño del caso de uso 5



Ilustración 25. Resultado Final del caso de uso 5

#### 3.5.2.1.4.6. Cerrar la Sesión.

Caso de uso 6	Cerrar la Sesión
Resumen de la Funcionalidad	Permite al usuario cerrar su sesión
Casos de uso relacionados	Acceder a la aplicación
Actores	Usuario registrado
Precondición	El usuario debe haberse creado una cuenta y haber accedido con sus datos de acceso.
Postcondición	El sistema cerrará la sesión y el usuario deberá introducir de nuevo sus datos de acceso para acceder a la aplicación
Escenario de éxito	<ol style="list-style-type: none"><li>1. El usuario registrado selecciona “Logout”</li><li>2. El sistema cierra la sesión y redirige al usuario a la pantalla de “Login”.</li></ol>
Extensiones	

Tabla 28. Caso de Uso 6

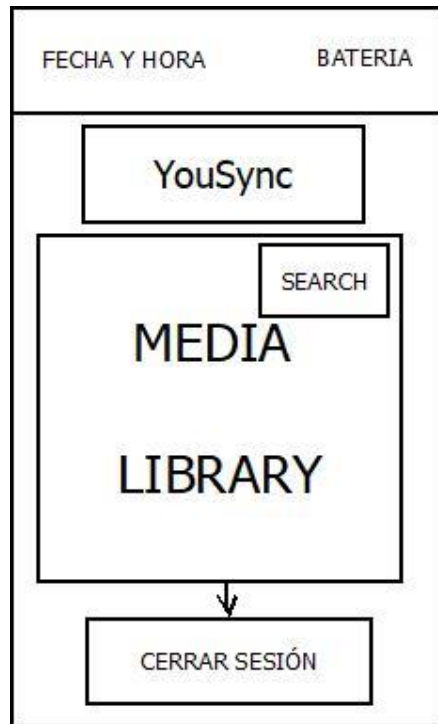


Ilustración 26. Diseño del caso de uso 6



Ilustración 27. Resultado Final del caso de uso 6

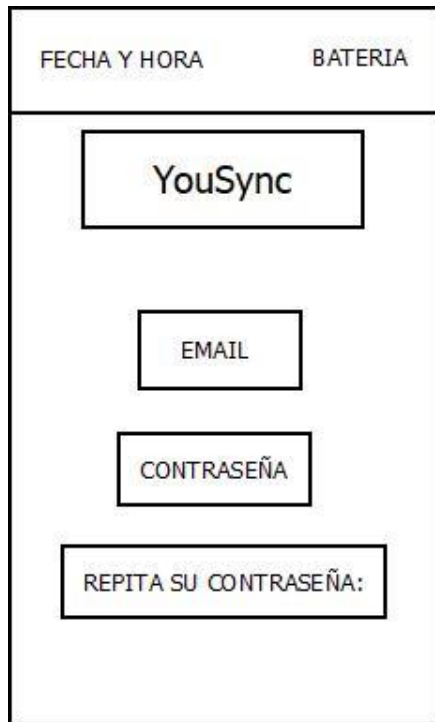


### 3.5.2.1.4.2. Casos de uso del modulo de Administrador

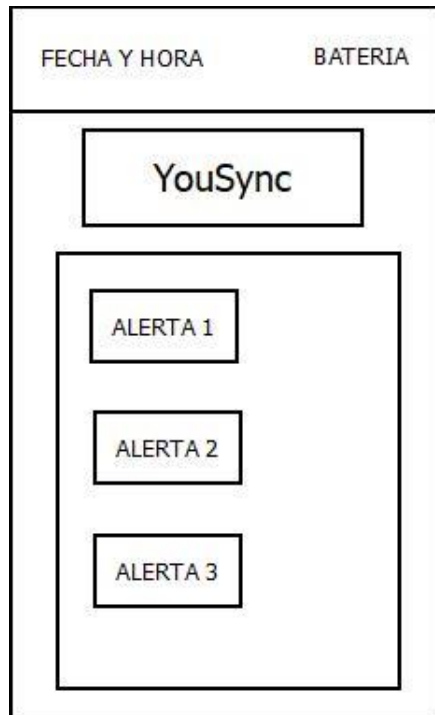
#### 3.5.2.1.4.2.1. Acceder a la aplicación.

Caso de uso 7	Acceder a la aplicación
Resumen de la Funcionalidad	Permite al administrador acceder a la aplicación
Casos de uso relacionados	
Actores	Administrador
Precondición	Debe habersele proporcionado un nombre de usuario y una contraseña a la persona que vaya a administrar la aplicación
Postcondición	Se mostrará la pantalla “Alertas” del administrador
Escenario de éxito	<ol style="list-style-type: none"><li>1. El sistema muestra la pantalla principal</li><li>2. El Administrador introduce sus datos: correo electrónico y contraseña y pulsa en “Login”</li><li>3. El sistema comprueba que los datos son correctos y redirige al usuario a la pantalla “Alertas” del Administrador</li></ol>
Extensiones	<p>2a. El Administrador introduce alguno de los datos de forma incorrecta o deja un dato sin rellenar.</p> <p>2a1. El sistema muestra un mensaje de error.</p> <p>3a. El sistema no detecta el conjunto de datos de correo electrónico y contraseña del Administrador en la base de datos.</p> <p>3a1. El sistema muestra un mensaje de error diciendo que no hay un usuario registrado con ese correo electrónico y/o contraseña</p>

Tabla 29. Caso de Uso 7



*Ilustración 28. Diseño del caso de uso 7*



*Ilustración 29. Resultado Final del caso de uso 7*

3.5.2.1.4.2.2. *Recuperar la contraseña.*

Caso de uso 8	Recuperar la contraseña
Resumen de la Funcionalidad	Permite al Administrador recuperar su contraseña en caso de haberla olvidado
Casos de uso relacionados	
Actores	Administrador
Precondición	
Postcondición	El sistema enviará al correo electrónico del administrador una nueva contraseña para acceder a la aplicación
Escenario de éxito	<ol style="list-style-type: none"><li>1. El sistema muestra la pantalla principal</li><li>2. El administrador selecciona la opción “Recuperar mi contraseña”</li><li>3. El sistema muestra un formulario para introducir el correo electrónico del administrador donde se enviará la nueva contraseña</li><li>4. El administrador introduce su correo electrónico y pulsa “Recuperar Contraseña”</li><li>5. El sistema comprueba que tiene en la base de datos ese correo electrónico y redirige al administrador a la pantalla “Login” del sistema.</li></ol>
Extensiones	<p>3a. El administrador introduce una dirección de correo electrónico errónea</p> <p>3a1. El sistema muestra un mensaje de error.</p> <p>4a. El sistema no puede acceder a la base de datos en ese momento.</p> <p>4a1. El sistema muestra un mensaje de error diciendo que no se puede recuperar la contraseña en ese instante.</p>

Tabla 30. Caso de Uso 8

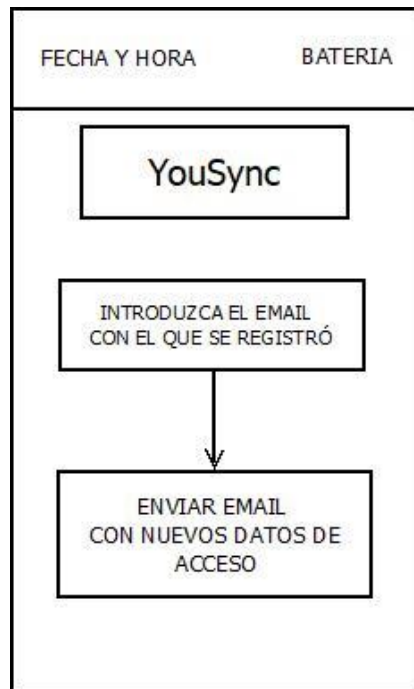


Ilustración 30. Diseño del caso de uso 8

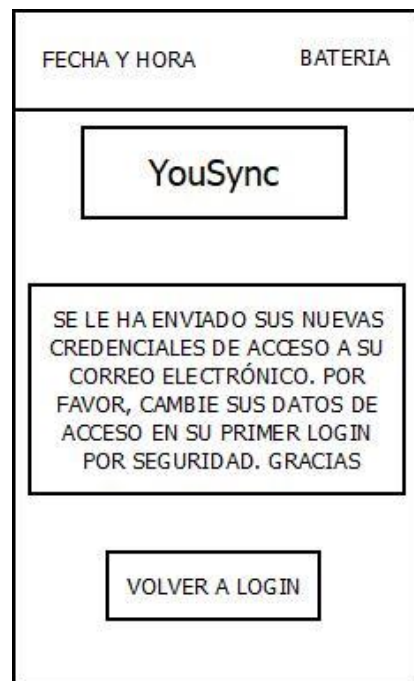


Ilustración 31. Resultado Final del caso de uso 8

3.5.2.1.4.2.3. Consultar datos del administrador.

Caso de uso 9	Consultar datos del administrador
Resumen de la Funcionalidad	Permite al administrador consultar sus datos de acceso al sistema
Casos de uso relacionados	
Actores	Administrador
Precondición	El administrador debe haber accedido al sistema con sus datos de acceso
Postcondición	El sistema muestra los datos de acceso del usuario
Escenario de éxito	<ol style="list-style-type: none"><li>1. El administrador selecciona “Mi perfil”</li><li>2. El sistema muestra los datos de acceso</li></ol>
Extensiones	<ol style="list-style-type: none"><li>2a. El sistema no puede acceder a la base de datos en ese momento.<ol style="list-style-type: none"><li>2a1. El sistema muestra un mensaje de error diciendo que no se puede recuperar los datos de acceso del usuario registrado en ese instante</li></ol></li></ol>

Tabla 31. Caso de Uso 9

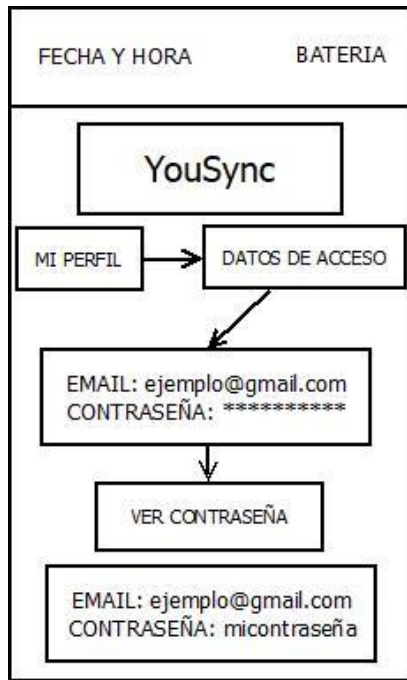


Ilustración 32. Diseño del caso de uso 9



Ilustración 33. Resultado Final del caso de uso 9

3.5.2.1.4.2.4. *Modificar datos de acceso del administrador.*

<b>Caso de uso 10                      Modificar datos de acceso del administrador</b>	
<b>Resumen de la Funcionalidad</b>	Permite al administrador modificar sus datos de acceso al sistema
<b>Casos de uso relacionados</b>	Consultar datos del administrador
<b>Actores</b>	Administrador
<b>Precondición</b>	El administrador debe consultar los datos de acceso
<b>Postcondición</b>	El sistema actualiza los datos de acceso nuevos del administrador
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. El administrador selecciona modifica uno de los datos de acceso: Correo electrónico y/o contraseña y pulsa en “Modificar datos de acceso”</li><li>2. El sistema actualiza los datos de acceso del administrador y redirige al administrador a la página “Login” para que ingrese con los nuevos datos de acceso.</li></ol>
<b>Extensiones</b>	<ol style="list-style-type: none"><li>1a. El administrador introduce algún dato de forma errónea.<ol style="list-style-type: none"><li>1a1. El sistema muestra un mensaje de error diciendo que hay un dato introducido de forma errónea.</li></ol></li><li>2a. El sistema no puede conectar con la base de datos y actualizar los datos de acceso del usuario.<ol style="list-style-type: none"><li>2a1. El sistema muestra un mensaje indicando que no se pueden modificar los datos de acceso del usuario registrado en ese instante</li></ol></li></ol>

*Tabla 32. Caso de Uso 10*

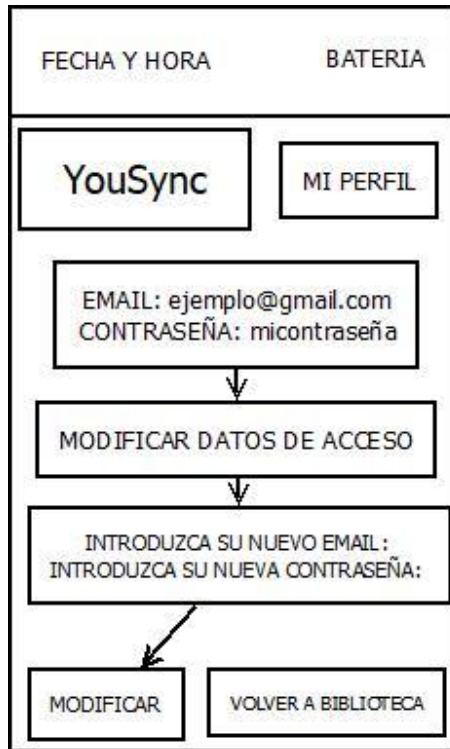


Ilustración 34. Diseño del caso de uso 10



Ilustración 35. Resultado Final del caso de uso 10



3.5.2.1.4.2.5. *Mirar alertas.*

<b>Caso de uso 11</b>	<b>Mirar alertas</b>
<b>Resumen de la Funcionalidad</b>	Permite al administrador mirar las alertas pendientes
<b>Casos de uso relacionados</b>	Acceder a la aplicación
<b>Actores</b>	Administrador
<b>Precondición</b>	El administrador debe haber accedido con sus datos de acceso.
<b>Postcondición</b>	El sistema accederá a las alertas que haya en ese momento
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. El administrador selecciona “Alertas”</li><li>2. El sistema muestra la pantalla “Alertar”.</li></ol>
<b>Extensiones</b>	<ol style="list-style-type: none"><li>2a. El sistema no puede obtener las alertas de la base de datos<ol style="list-style-type: none"><li>2a1. El sistema indica al administrador que no ha sido posible acceder a las alertas en ese instante</li></ol></li></ol>

*Tabla 33. Caso de uso 11*

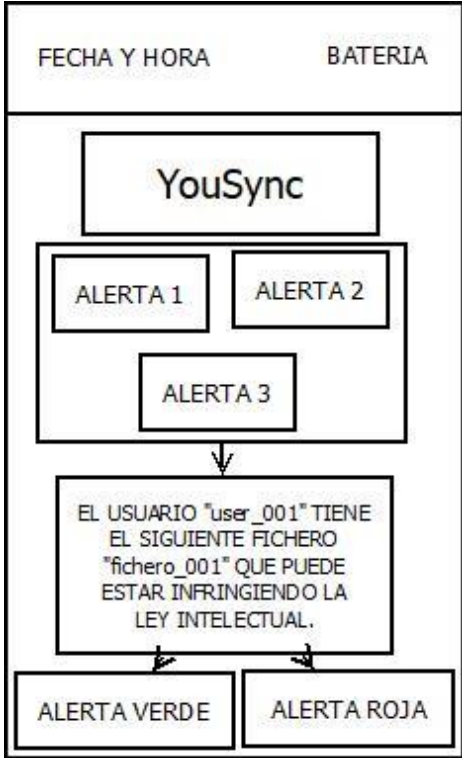


Ilustración 36. Diseño del caso de uso 11

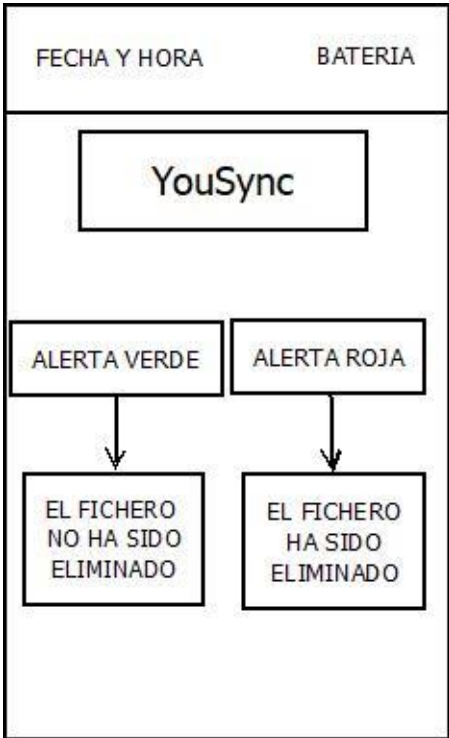


Ilustración 37. Resultado Final del caso de uso 11

3.5.2.1.4.2.6. *Alerta Roja.*

<b>Caso de uso</b>	<b>Alerta Roja</b>
<b>12</b>	
<b>Resumen de la Funcionalidad</b>	Permite al administrador verificar un contenido como inapropiado
<b>Casos de uso relacionados</b>	Mirar Alertas
<b>Actores</b>	Administrador
<b>Precondición</b>	El administrador debe haber accedido con sus datos de acceso.
<b>Postcondición</b>	El sistema borrará el contenido asociado a esa alerta
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. El administrador selecciona una alerta</li><li>2. El sistema muestra la alerta.</li><li>3. El administrador pulsa sobre “Alerta Roja”</li><li>4. El sistema indica que se ha borrado el contenido de manera correcta y redirige al administrador a la pantalla de “Alertas”</li></ol>
<b>Extensiones</b>	<ol style="list-style-type: none"><li>2a. El sistema no puede obtener las alertas de la base de datos<ol style="list-style-type: none"><li>2a1. El sistema indica al administrador que no ha sido posible acceder a esa alerta en ese instante</li></ol></li><li>3a. El sistema no puede acceder a la base de datos<ol style="list-style-type: none"><li>3a1. El sistema nos indica que no se puede borrar el contenido en ese momento.</li></ol></li></ol>

*Tabla 34. Caso de Uso 12*

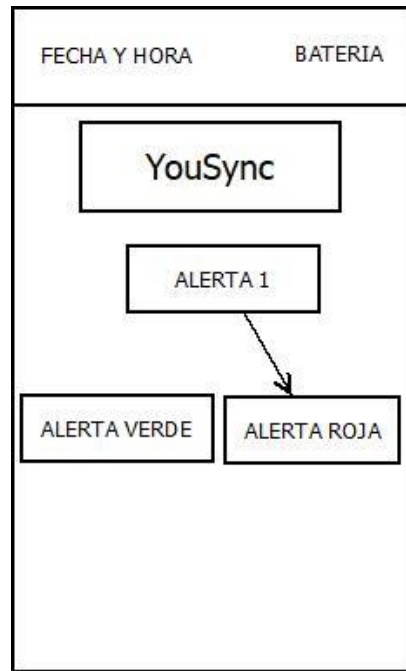


Ilustración 38. Diseño del caso de uso 12

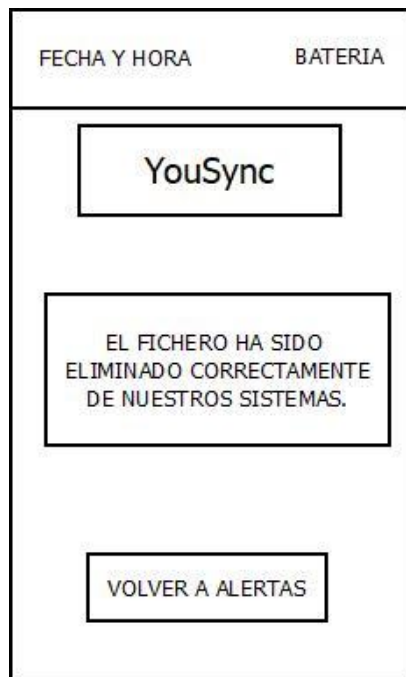


Ilustración 39. Resultado Final del caso de uso 12

3.5.2.1.4.2.7. Alerta Verde.

Caso de uso 13	Alerta Verde
Resumen de la Funcionalidad	Permite al administrador verificar un contenido como apropiado
Casos de uso relacionados	Mirar Alertas
Actores	Administrador
Precondición	El administrador debe haber accedido con sus datos de acceso.
Postcondición	El sistema borrará la alerta, pero no el contenido asociado a esa alerta
Escenario de éxito	<ol style="list-style-type: none"><li>1. El administrador selecciona una alerta</li><li>2. El sistema muestra la alerta.</li><li>3. El administrador pulsa sobre “Alerta Verde”</li><li>4. El sistema indica que se ha borrado la alerta del sistema y redirige al administrador a la pantalla de “Alertas”</li></ol>
Extensiones	<ol style="list-style-type: none"><li>2a. El sistema no puede obtener las alertas de la base de datos<ol style="list-style-type: none"><li>2a1. El sistema indica al administrador que no ha sido posible acceder a esa alerta en ese instante</li></ol></li><li>3<sup>a</sup>. El sistema no puede acceder a la base de datos<ol style="list-style-type: none"><li>3<sup>a</sup>1. El sistema nos indica que no se puede la alerta en ese momento.</li></ol></li></ol>

Tabla 35. Caso de Uso 13

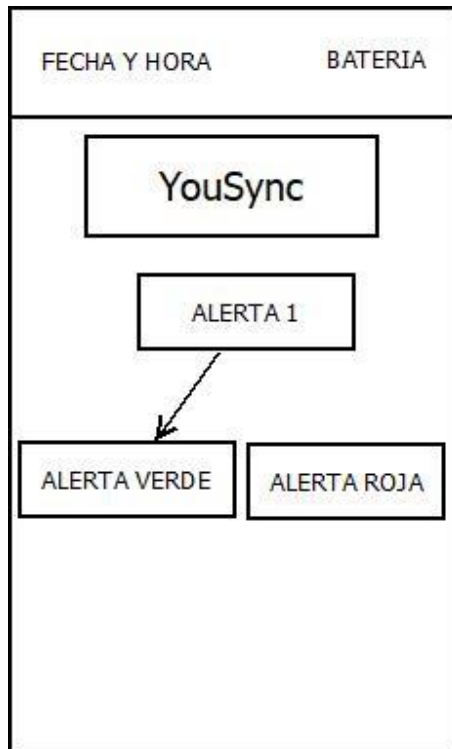


Ilustración 40. Diseño del caso de uso 13

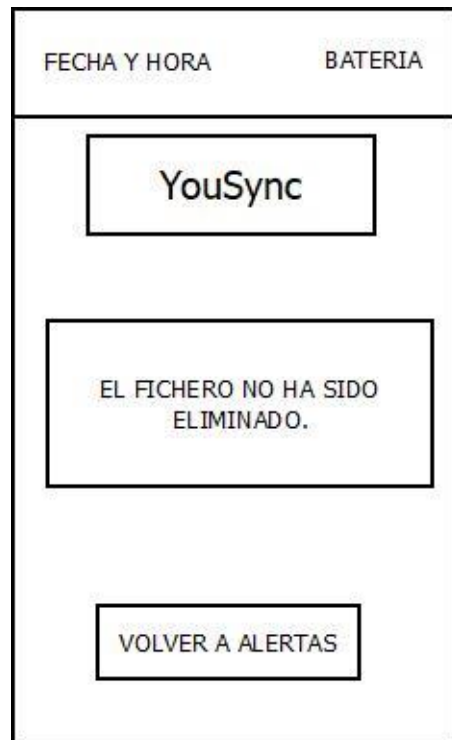


Ilustración 41. Resultado Final del caso de uso 13

3.5.2.1.4.2.8. Cerrar la Sesión.

<b>Caso de uso 14</b>	<b>Cerrar la Sesión</b>
<b>Resumen de la Funcionalidad</b>	Permite al administrador cerrar su sesión
<b>Casos de uso relacionados</b>	Acceder a la aplicación
<b>Actores</b>	Administrador
<b>Precondición</b>	El administrador debe haber accedido con sus datos de acceso.
<b>Postcondición</b>	El sistema cerrará la sesión y el administrador deberá introducir de nuevo sus datos de acceso para acceder a la aplicación
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>5. El administrador selecciona “Logout”</li><li>6. El sistema cierra la sesión y redirige al administrador a la pantalla de “Login”.</li></ol>
<b>Extensiones</b>	

Tabla 36. Caso de Uso 14

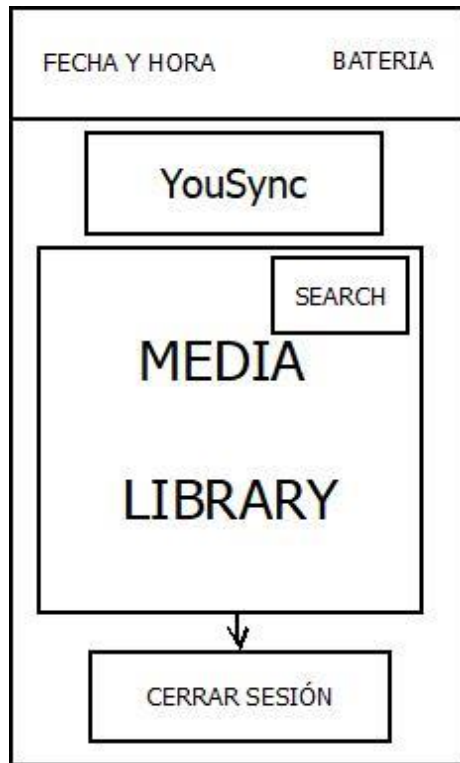


Ilustración 42. Diseño del caso de uso 14



Ilustración 43. Resultado Final del caso de uso 14



3.5.2.1.4.3. Casos de uso del modulo del buscador.

3.5.2.1.4.3.1. *Búsqueda por nombre.*

<b>Caso de uso 15</b>	<b>Búsqueda por nombre</b>
<b>Resumen de la Funcionalidad</b>	Permite al usuario buscar un archivo por su nombre
<b>Casos de uso relacionados</b>	Acceder a la aplicación
<b>Actores</b>	Usuario Registrado
<b>Precondición</b>	El usuario registrado debe haber accedido con sus datos de acceso.
<b>Postcondición</b>	El sistema mostrará una lista con resultados obtenidos a tiempo real
<b>Escenario de éxito</b>	<ol style="list-style-type: none"><li>1. El usuario registrado debe escribir en el cuadro de búsqueda</li><li>2. El sistema muestra los resultados a tiempo real</li></ol>
<b>Extensiones</b>	

*Tabla 37. Caso de Uso 15*

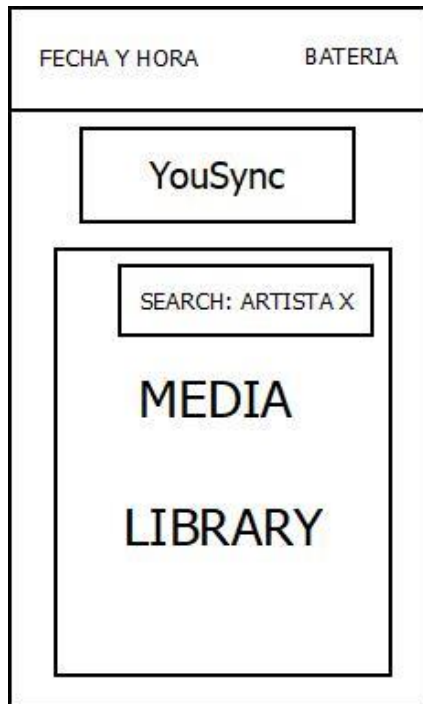


Ilustración 44. Diseño del caso de uso 15



Ilustración 45. Resultado Final del caso de uso 15

3.5.2.1.4.3.2. Selección

Caso de uso 16	Selección
Resumen de la Funcionalidad	Permite al usuario seleccionar un tipo de archivo
Casos de uso relacionados	Acceder a la aplicación
Actores	Usuario Registrado
Precondición	El usuario registrado debe haber accedido con sus datos de acceso.
Postcondición	El sistema mostrará una lista con resultados obtenidos
Escenario de éxito	<ol style="list-style-type: none"><li>1. El usuario registrado pulsará en “Tipo de contenido”</li><li>2. El sistema muestra 3 opciones: Música, Video e Imágenes.</li></ol>
Extensiones	<ol style="list-style-type: none"><li>2a. El sistema no puede acceder a la base de datos del usuario.<ol style="list-style-type: none"><li>2a1. El sistema muestra un mensaje de error indicando que en ese momento no puede mostrar el contenido del usuario.</li></ol></li></ol>

Tabla 38. Caso de Uso 16



Ilustración 46. Diseño del caso de uso 16



Ilustración 47. Resultado Final del caso de uso 16

3.5.2.1.4.3.3. Reproducción de Archivos.

Caso de uso 17	Reproducción de Archivos
Resumen de la Funcionalidad	Permite al usuario registrado reproducir un archivo
Casos de uso relacionados	Acceder a la aplicación
Actores	Usuario Registrado
Precondición	El usuario registrado debe haber accedido con sus datos de acceso y/o haber seleccionado el tipo de fichero que quiere mostrar
Postcondición	El sistema reproducirá el archivo
Escenario de éxito	<ol style="list-style-type: none"><li>1. El usuario registrado pulsará en el archivo deseado</li><li>2. El sistema abrirá un reproductor y reproducirá el archivo seleccionado por el usuario</li></ol>
Extensiones	

Tabla 39. Caso de Uso 17

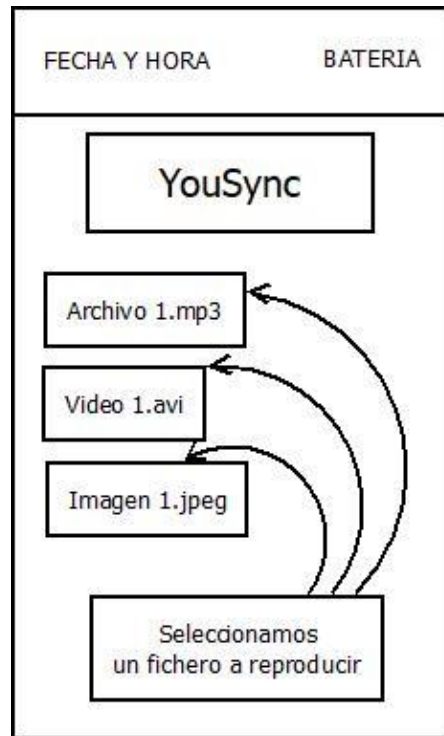


Ilustración 48. Diseño del caso de uso 17

### 3.5.3. Diseño de la lógica de datos

En este sub-apartado vamos a ver el modelo Entidad-Relación que se ha seguido para la base de datos de la aplicación. [12]

#### 3.5.3.1. Modelo Entidad-Relación

Para este Trabajo Fin de Grado vamos a tener 4 entidades: “Usuarios”, “Administradores”, “Multimedia” y “Alertas” y el esquema E/R sería el siguiente:

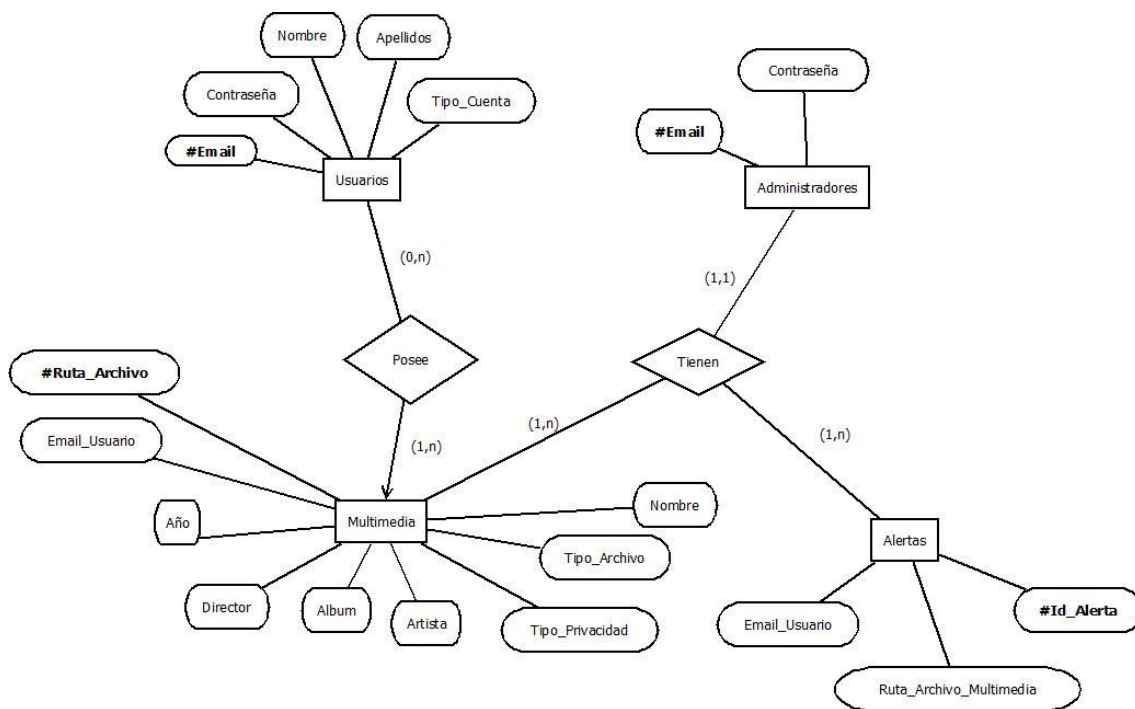


Ilustración 49. Esquema E/R

Las relaciones planas de nuestras entidades serían las siguientes:

Usuarios (Email, Contraseña, Nombre, Apellidos, Tipo\_Cuenta).

Administradores (Email, Contraseña).

Multimedia (Ruta\_archivo, Email\_Usuario, Nombre, Tipo\_Archivo, Tipo\_Privacidad, Artista, Album, Director, Año)

Alertas (Email Usuario, Ruta Archivo Multimedia, ID Alerta).

Posee (Email Usuario, Ruta Archivo Multimedia)

Tienen (Email Administradores, Email Usuarios, Ruta Archivo Multimedia, ID Alerta Alertas).

#### 4. Conclusiones y Líneas Futuras.

El desarrollo de esta Trabajo Fin de Grado no ha sido nada fácil ya que a lo largo de su ejecución me he encontrado con varios retos que he tenido que superar y tomar distintas decisiones a lo largo de su desarrollo. En un primer instante, lo único que tenía claro, era que tipo de servicio quería, un servicio que fuera capaz de manejar bibliotecas de contenido de distintos usuarios y que existiera alguien con un rol de administrador que pudiera gestionar el contenido que pudiera ser ilegal en la actual ley.

Cuando me puse a trabajar, surgió la primera pregunta, ¿Sobre que plataforma voy a trabajar? Trabajo sobre Android, o bien trabajo sobre iOS. Pero quería que mi aplicación llegara al mayor número de dispositivos o bien que fuera capaz de llegar a un gran número de dispositivos. Por eso me puse a estudiar sobre las tecnologías que habían actualmente priorizando siempre que fuera posible migrar al menos, fácilmente, el código de una plataforma a otra. De esta manera descubrí el maravilloso mundo de las aplicaciones híbridas y dentro de ellas, la tecnología Ionic que me daba la solución que yo buscaba. Realizar un solo código para una plataforma y con solo unos pocos cambios o nulos cambios en algunos casos tener el código resultante para otra plataforma.

Una vez elegida la “plataforma” sobre la que iba a trabajar, había que elegir la arquitectura con la que íbamos a trabajar. En este caso al ser un conjunto de usuarios que van a hacer uso de un servicio me fue algo más fácil que la elección de la plataforma en la que trabajar, y terminé eligiendo la Arquitectura Cliente/Servidor.

El transcurso de estos meses en los que he ido desarrollando el Trabajo Fin de Grado, han sido muy satisfactorios, aun habiendo dificultades, aun teniendo que aprender tecnologías totalmente nuevas para mí, que nunca antes había visto para las que me he tenido que preparar, cada vez que superaba un obstáculo, una dificultad más me sentía más reconfortado y con más ganas de seguir avanzando, de llegar al final del camino y de este Trabajo Fin de Grado y de ver el estado final del propio.

Finalmente, uno de los propósitos del Trabajo Fin de Grado es el de preparar al alumno para el mercado laboral. Yo puedo decir que me ha ayudado a no tener miedo a aprender un lenguaje nuevo, formarme en una nueva tecnología y a enfrentarme a problemas completamente nuevos para mí.



Uno de mis principales objetivos con este Trabajo Fin de Grado era el de convertirlo en un Trabajo Fin de Grado en el que mis compañeros de años posteriores trabajaran y mejoraran él mismo. Para unas líneas futuras se podrían agregar los siguientes servicios:

- Los mismos usuarios a través de la aplicación podrían marcar contenidos como inapropiados (si hay algún contenido que les moleste o vaya en contra de sus propios derechos o privacidad) para que le saltara alarmas al administrador y éste pudiera mantener estos contenidos más en orden.
- Podría integrarse el servicio a otros servicios como por ejemplo Spotify, Youtube y mientras que se están subiendo, o gestionando la biblioteca de medios de un usuario poder estar disfrutando de contenidos audiovisuales.
- Además, cabría la posibilidad de crear una pequeña red social entre usuarios con mismos gustos musicales, mismos gustos de cine, etc.
- Inclusive, cabría la posibilidad de crear una plataforma que, a través de publicidad, o de algún tipo de plan de negocio tuviera en su haber sistemas de streaming aunados como son Netflix, HBO, Lastfm, Spotify, etc. Y tener en una misma plataforma todos los más famosos servicios del momento.

Cómo se puede ver, se puede agrandar la plataforma por si misma, ya que con una buena publicidad, se aumentaría el número de usuarios y con ella el número de contenido audiovisual, teniendo cada vez más una mayor biblioteca de medios general del servicio, como también unirse a otras plataformas ya existentes sumando sus ventajas con las de nuestra plataforma.

## 5. Anexo 1. Manual Usuario

En este apartado se va a realizar el manual de usuario de la aplicación. Este se hará sin seguir ningún diseño actual, por si en el futuro se cambia el diseño de la aplicación, se modifica la posición de botones, etc, siguiera sirviendo este manual. Ya que se trataría de una aplicación/servicio que iría creciendo y formándose en conjunto al crecimiento de personas que accedan al mismo.

En primera instancia tendríamos la página principal, con el logo de la aplicación a la que he llamado YouSync y una pantalla de bienvenida que mostrara dos opciones. La opción de loguearse, y la opción de registrarse si no se tuviera cuenta en el servicio.

Si le damos a loguearse, iremos a una nueva ventana, con un formulario que deberá ser rellenado con el email y la contraseña con la que nos registramos. Si se meten bien los datos, la aplicación nos mostraría un mensaje indicándonos que se ha introducido unos datos que existen en la base de datos de usuarios y nos llevaría directamente a la biblioteca de medios del usuario.

Si le damos a registrarse, nos dirigiremos a una nueva ventana, con un formulario que se deberá rellenar con el email que queremos usar para nuestro registro, con la contraseña que queremos tener, y un campo para repetir la contraseña y que el sistema se asegure que hemos introducido correctamente la contraseña que queremos. Si todo va bien, el sistema nos dejará un mensaje indicando que se ha introducido correctamente los datos y podremos ya loguearnos.

Una vez logueados, tendremos a nuestra disposición un cajón de búsqueda en el que una vez que vayamos introduciendo caracteres nos mostrará en tiempo real los ficheros que contengan el conjunto de caracteres que hayamos introducido en ese momento, permitiéndonos pulsar sobre un fichero y reproducirlo.

Tendremos también un selector de tipo de archivo. En principio se hará una selección por tres tipos diferentes de archivo: “Video, Imagen y Audio”, mostrándonos una vez que pulsemos sobre uno de estas tres opciones todos los ficheros que el usuario tenga de ese tipo de archivo.

Habrà un menú o submenú en el que podamos ver nuestros datos de acceso y en el cuál podamos modificar los mismos.

Además, tendremos un botón que nos permitirá subir contenido al servidor. Se nos abrirá una ventana emergente para explorar los directorios de nuestro equipo o móvil y así buscar y elegir el archivo a subir. Si el archivo es compatible con el sistema, lo subirá y nos mostrará un mensaje confirmándonos que se ha enviado correctamente y sino se elige un archivo compatible con el sistema, nos mostrará un mensaje con un error.

En la parte de administrador. Actualmente cuando se hace login, si el sistema detecta que se está logueando un administrador, no nos aparecerá una nueva ventana con la biblioteca de medios del usuario, sino que se abrirá una ventana en la que se muestren una serie de Alertas acerca de contenido inapropiado o ilegal que tendrá el administrador que corroborar si efectivamente esas alertas, puestas por otro administrador verdaderamente son por algún contenido inapropiado o ilegal y marcarlas como alerta roja borrando de esta manera el fichero del sistema. Además tendrá una opción adicional que no tiene el usuario normal, y aparecerá en un menú o submenú y será la de explorar toda la biblioteca de medios del servidor, pudiendo marcar cualquier archivo de la biblioteca como una posible alerta, para que sea investigada y corroborada por otro administrador.

## 6. Anexo 2. Instalación de Ionic

Al ser una tecnología bastante actual, se ha pensado en hacer un pequeño manual para la instalación de Ionic tanto para Mac como para Windows.

### 6.1. Instalación de Node.js

En primer lugar, deberemos instalar “Node.js”. La instalación en el sistema MAC la podemos realizar de dos formas diferentes:

#### 1. HomeBrew

Nos dirigiremos a la página <http://brew.sh> y seguimos las instrucciones en la página web de Homebrew bajo la sección de “Install Homebrew”.

Luego, abrimos una terminal y escribimos:

```
$ brew install node --without-npm
```

```
$ echo prefix=/.node >> /.npmrc
```

```
$ curl -L https://www.npmjs.com/install.sh | sh
```

Después de haber realizado esos comandos en la terminal editamos el fichero “.bash\_profile” y nos aseguramos de añadir la variable “PATH environment” con el siguiente comando:

```
$ export PATH="HOME/.node/bin:$PATH"
```

#### 2. Mediante el instalador de Node de Mac OS X

Si por alguna razón no se quiere usar homebrew se puede instalar usando su instalador. Nos dirigimos a la siguiente página:

<https://nodejs.org/download/>

Y solamente sería seguirlos pasos que se indican en el instalador.

Para la instalación en el sistema de Microsoft solo tenemos la opción de realizarla mediante el instalador de Node dirigiéndonos a la siguiente dirección:

<https://nodejs.org/download/>

## 6.2. Instalación de la consola de comandos de Ionic

Ahora procederemos a instalar la consola de comandos de Ionic, al estar basado en Cordova debemos instalarla.

```
$ npm -g install cordova
```

```
$ npm -g install ionic
```

Nos aseguramos de que haya funcionado correctamente con el siguiente comando:

```
$ ionic info
```

Debería salir algo parecido a lo siguiente:

```
Ñkjasdñljkjasñlkdjfnaskjd  fñajsdñfkja  sñlfdpj  asñlkdjf  ñaskj  dfñlkjañslkdjf  
ñaskjdñfkjasñkdjf añslkdj fñaskj dfñakjsñdfkl jañslkje fdñaspjkd
```

## 6.3. Instalación de Bower

Ahora procederemos a instalar Bower. Bower es un gestor de paquetes del sistema que se usa para instalar librerías de javascript de terceros en las aplicaciones.

```
$ npm install -g bower
```

Con esto ya tendríamos instalado el gestor de paquetes

## 6.4. Instalación de Java y Android en Windows

En el sistema operativo de Microsoft deberemos instalar la versión de desarrollo de Java “Java JDK” dirigiéndonos a la siguiente dirección:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Una vez que hemos instalado la versión JDK de Java tenemos que crear la variable del sistema para “JAVA\_HOME” que apunte al directorio raíz de donde hemos instalado la versión JDK de Java. Si hemos instalado Java en la siguiente dirección “C:\Program Files\Java\jdk” tendremos que apuntar JAVA\_HOME a este directorio. Después de esto, añadiríamos el subdirectorio bin al JAVA\_HOME.

A continuación tendríamos que instalar la versión de desarrollode Android “Android SDK”. El Android SDK nos proviene las librerías de la API y herramientas de desarrollo necesarias para compilar, testear y debugar aplicaciones para Android.

Cordova requiere que la variable del sistema ANDROID\_HOME este realizada. Éste debería apuntar al directorio sdk perteneciente al directorio raíz donde esté instalado el Android SDK. Por último incluir a la variable del sistema ANDROID\_HOME los subdirectorios "tools" y “platform-tools”

#### 6.5. Creando un proyecto en blanco y testeándolo

Para crear un proyecto en blanco en ionic ponemos el siguiente comando en la terminal, ya sea de Windows o de Mac una vez habiendo instalado todo lo anterior en este manual:

```
$ ionic start ejemplo blank --type ionic1
```

Si queremos ver si se ha creado correctamente, podemos probarlo de la siguiente manera:

```
$ cd ejemplo $$ ls
```

Si aquí vemos un conjunto de carpetas es que todo ha ido correctamente.

Una vez que tenemos instalado nuestro proyecto en blanco podremos ejecutarlo en ios o Android deberemos añadir las plataformas al proyecto ejecutando los siguientes comandos respectivamente para cada plataforma:

```
$ ionic cordova platform add ios
```

```
$ ionic cordova platform add Android
```

Y por último, para poder ejecutarlo en un emulador ejecutaríamos los siguientes comandos respectivamente para cada plataforma:

```
$ ionic cordova build ios
```

```
$ ionic cordova emulate ios
```

```
$ ionic cordova build Android
```

```
$ ionic cordova emulate Android
```

## 7. Bibliografía

Udemy. – Curso de Ionic

Apuntes tomados durante el Grado de Ingeniería Telemática.

[1] <https://naseros.com/2014/05/20/plex-un-mediacentro-o-gestor-de-contenidos-multimedia-para-tu-nas/>

[2] <http://mundokodi.com/la-historia-de-kodi/#gs.tWrhAYU>

[3] <http://www.neonez.com/mejores-alternativas-kodi/>

[4] <https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>

[5] <http://ocw.uc3m.es/ingenieria-informatica/principios-de-ingenieria-informatica/introduccion-a-la-ingenieria-del-software>

[6] [http://computingcareers.acm.org/?page\\_id=12](http://computingcareers.acm.org/?page_id=12)

[7] <https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>

[8] <https://www.programacion.com.py/varios/arquitectura-cliente-servidor>

[9] <http://www.clubinvestigacioncr.com/docs/informe23.pdf>

[10] <https://radiosyculturalibre.com.ar/biblioteca/REDES/linuxito%20-%20El%20Modelo%20Cliente-Servidor.pdf>

[11] [http://www.it.uc3m.es/mcftp/docencia/si/material/1\\_cli-ser\\_mcfp.pdf](http://www.it.uc3m.es/mcftp/docencia/si/material/1_cli-ser_mcfp.pdf)

[12] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-entidad-relaci%C3%B3n>