

Diseño y implementación de un sistema web para la gestión de elementos multimedia

Lluís Falcó García

Resumen— Este proyecto trata de la creación de un sistema web capaz de gestionar y organizar contenidos multimedia. Actualmente, el volumen de elementos multimedia que podemos recabar o generar puede ser de un calibre inmenso. Debido a esta cantidad de archivos, es fácil perderse en aquel sistema que los almacena, olvidarse de estos o simplemente no llegar a conocer de su existencia. Para tratar de dar solución a este problema, el proyecto quiere dar como resultado un sistema que permita organizar nuestros diferentes tipos de elementos (ya sean, imágenes, videos, audios o documentos de texto) con la única ayuda de un navegador. Esto nos permitirá tener acceso al sistema desde prácticamente cualquier dispositivo conectado a internet, visualizar y organizar los elementos de una forma intuitiva a gusto del propio usuario y poder compartir estos con otros usuarios mediante la introducción de su correo electrónico.

Palabras clave— Gestión, Organización, Multimedia, Red Social, Diseño, Desarrollo, Web, C#, Javascript, AngularJS, MySql, Html, Css, Aplicación web

Abstract— This project relates to the creation of a web system capable of managing and organizing multimedia content. Currently, the volume of multimedia elements we are able to collect or generate can be of immense caliber. Due to the amount of archives, it is easy to get lost in the system which stores them, forget about them or simply never find out about them. As a solution to this problem, this project aims to provide a system that allows us to organize our different types of elements (images, videos, audio or text documents) with just the help of a browser. This will allow us to access the system from practically any device connected to the internet, visualize and organize the elements in an intuitive form to the user's liking and share these elements with other users via email.

Index Terms— management, organization, multimedia, social network, design, development, web, C#, Javascript, AngularJS, MySql, HTML, CSS, Web application



1 INTRODUCCIÓN

HOY en día, podemos recabar o generar una cantidad inmensa de contenidos multimedia. Esta abrumadora cantidad de elementos hace que en ocasiones, sin la organización adecuada, perdamos, olvidemos o no extraigamos todo el potencial que estos son capaces de ofrecer.

Este problema fue el causante del inicio del proyecto. Una gran cantidad de elementos sin apenas organización y dispersos en un sistema de archivos que provocaba que en gran medida se olvidaran.

Estos elementos, archivos de texto, videos, imágenes, audios que formaban parte, en gran medida de cursos, asignaturas, recuerdos, libros que caían en el olvido junto con el potencial que ofrecían.

Este proyecto pretende diseñar y desarrollar un sistema web, que aproveche el gran potencial que ofrece Internet, para llevar a cabo tareas que nos ayuden a organizar este gran volumen de elementos multimedia y que permita a las personas conectar entre ellas de una forma rápida e intuitiva, compartiendo esos contenidos con aquellos que deseen, exponiéndolos a la opinión pública o de un grupo de usuarios concreto para que estos puedan puntuar, opinar y aprender de éstos.

El segundo capítulo Estado del arte, nos expondrá los conceptos e ideas más relevantes y analizará los sistemas existentes. El tercer capítulo corresponde a los objetivos que se proponen para el proyecto. El cuarto, Requisitos del sistema, nos lista los requisitos y las restricciones. El quinto, Metodología, expone la metodología que se utiliza a lo largo del desarrollo del proyecto. El sexto punto, nos describe las tareas realizadas. El séptimo punto expone los resultados obtenidos. El octavo, presenta las conclusiones extraídas de los resultados del proyecto. El noveno punto expone posibles futuras implementaciones del sistema. En el décimo capítulo se exponen los agradecimientos y por último, en el decimoprimer punto, la bibliografía utilizada para la realización del proyecto.

-
- E-mail de contacte: lluisfalconeti@gmail.com
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Fernando Luís Vilarinho Freire (CVC) y Daniel Alexander Norton
 - Curs 2015/16

2 ESTADO DEL ARTE

A día de hoy, con la proliferación de Internet, podemos encontrar diversos programas o portales web que nos ofrecen la posibilidad de visualizar o reproducir archivos online.

En su mayoría, estos sistemas, están especializados en un tipo de archivo, ya sea imagen, video, audio o texto y ofrecen la posibilidad de añadir un componente social a estos mediante un sistema de puntuación o de comentarios asociados a estos elementos. Un pequeño extracto de los más conocidos de este tipo de sistemas son:

- *Youtube* [1]. El portal por excelencia que permite visualizar videos online. También permite el registro de usuarios que pueden subir sus propios videos, comentar, puntuar o generar sus propias listas.
- *Google images* [2]. Una especialización del buscador Google para imágenes.
- *Flickr* [3]. Sitio web que permite almacenar, ordenar, buscar, vender y compartir fotografías online.
- *Instagram* [4]. Red social para la subir fotos y videos en las que se permite aplicar efectos o filtros a estos y compartirlos.
- *Spotify* [5]. Aplicación social que se emplea para la reproducción de música. Permite la creación de listas a usuarios.
- *Coursera* [6]. Plataforma de educación virtual que permite a los usuarios realizar cursos, tanto gratuitos como de pago sobre temática muy variada.
- *Google Drive* [7]. Servicio de alojamiento de archivos.
- *Dropbox* [8]. Servicio de alojamiento de archivos online.

Este listado es un pequeño extracto de algunos de los más conocidos sistemas con similitudes al propuesto. Se puede observar que en muchos casos, estos sistemas aparte de servir estos tipos de archivos online, también implementan módulos que permiten la interacción social entre los usuarios. También observamos cómo se permite a los usuarios gestionar estos archivos, permitiéndoles crear listados que aportan más valor.

En estos sistemas podemos diferenciar aquellos que se dedican exclusivamente a la visualización o gestión de un tipo de elemento (*Youtube*, *Instagram*, *Spotify*, *Soundcloud*,...) y a otros que se dedican a su almacenamiento (*Google drive*, *Dropbox*,...). Estos últimos permiten el almacenamiento de documentos y en ocasiones su reproducción online, pero están enfocados al almacenamiento en un sistema de ficheros online.

Una posible forma de innovación en estos tipos de sistemas sería la de unir estas dos grandes funcionalidades:

1. Permitir registrar elementos de tal forma que podamos organizarlos como si de un sistema de ficheros se tratara.
2. Permitir visualizar estos elementos sin la necesidad de disponer de ellos localmente.

De esta forma se podrían mezclar todo tipo de archivos en el mismo espacio lógico. Una colección no estaría restringida a ser de un formato específico, y no deberíamos acudir a otro sistema para buscar otro tipo de archivos. Las temáticas de estas colecciones podrían almacenar archivos de texto, imágenes, grabaciones de audio, videos y permitiría obtener y acceder a grandes cantidades de información de temáticas muy variadas.

3 OBJETIVOS

El objetivo propuesto para este proyecto es el de crear un sistema informático que permita a los usuarios la gestión de elementos multimedia online.

1. Estos documentos pueden ser 4 tipos diferentes: Audio, Video, Imagen o Texto y mediante las direcciones URL de estos se podrán registrar en el sistema.
2. Cada uno de los elementos debe de contar con un espacio en el cual se podrán anotar las opiniones, descripciones u otro tipo de valoraciones del resto de usuarios y/o valorar el elemento mediante un sistema de puntuación.
3. Para poder registrar y/o gestionar los elementos existentes o nuevos, los usuarios deberán identificarse de una forma rápida y sencilla.
4. Con tal de permitir el acceso a este sistema desde cualquier dispositivo con conexión a internet y facilitar así su utilización, sin instalaciones ni otros procesos costosos, este sistema debe ser capaz de funcionar en la mayoría de los navegadores actuales tales como *google chrome*, *firefox*, *safari*, *opera*, etc.
5. Este también debe permitir al usuario crear sus propias colecciones de elementos, pudiendo así organizarlos a su gusto y gestionarlas de una forma rápida, sencilla e intuitiva.
6. Tanto colecciones como elementos, se deben poder compartir con otros usuarios permitiendo a estos, visualizar los elementos compartidos aun siendo privados.

4 METODOLOGÍA

En la Ingeniería del software, existen una gran variedad de métodos, que en gran medida nos aseguran llegar al objetivo deseado de una forma exitosa.

Para llevar a cabo el proyecto, y tras tener en cuenta las necesidades de este, se ha utilizado la metodología ágil de desarrollo del software conocido como *Modelo en espiral* [9] [10] (Figura 1). Esta metodología contempla los posibles cambios del proyecto y permite así adaptar de una manera fácil este a los nuevos requisitos o soluciones que surjan a lo largo de su desarrollo. De esta forma, también asegura en gran medida que los módulos más básicos del sistema serán funcionales.

El modelo de ciclo de vida del software está compuesto por una serie iteraciones en las que se desarrollan un conjunto

ordenado de actividades.

Este modelo en espiral está formado por 4 actividades:

- **Determinar los objetivos.** Obtener requisitos especificaciones, detalles y restricciones del módulo a implementar.
- **Análisis de riesgo.** Identificar riesgos y evaluar alternativas antes de comenzar a desarrollar.
- **Desarrollo y test.** Implementar soluciones y realizar testing.
- **Planificación.** Revisar el resultado obtenido y determinar las tareas, tiempo y recursos necesarios para la siguiente iteración.

Cada iteración permite mejorar el estado de las tareas realizadas anteriormente generando así nuevas versiones del sistema. En caso de bloqueo o falta de recursos, se puede optar por alternativas más sencillas, que no comprometen la evolución de este.



Figura 1: Espiral de boehm

4.1 Planificación

Para llevar a cabo el proyecto, se genera una planificación inicial.

El proyecto se divide en 8 apartados:

- **Definición objetivos y captura de requisitos.** En esta etapa se define el alcance deseado para el sistema y sus requisitos.
- **Puesta en marcha y configuración del servidor de pruebas.** Se procede a la configuración del servidor destinado al testing del sistema.
- **Diseño general de la aplicación web.** Se procede al diseño y maquetación general de la página web.
- **Implementación de las funcionalidades primarias** del sistema. Implementación de las funcionalidades más básicas o críticas indispensables para el sistema.
- **Implementación de las funcionalidades secundarias.** Implementación de las funcionalidades que se consideran menos prioritarias.
- **Mejora de los estilos, acabados y experiencia de usuario.** Mejora estética de la página web, optimización de los procesos e interacciones.
- **Redacción de los informes finales.**
- **Defensa del proyecto.** Contiene el conjunto de actividades destinadas a preparar la defensa del proyecto.

Tras la definición de los diferentes apartados, se genera

una planificación temporal aproximada que contiene una serie de tareas que nos permitirá poder llevar a cabo el proyecto (Apéndice 1).

4.2 Entorno de pruebas

Para poder llevar a cabo pruebas exhaustivas sobre las diferentes implementaciones realizadas, se configura un servidor de pruebas.

Este cuenta con las últimas versiones desarrolladas del sistema y proporciona una versión *limpia* que evita las posibles incoherencias existentes en el mismo entorno de desarrollo.

Con la finalidad también, de poder hacer un seguimiento y valoración del desarrollo del proyecto por parte de los stakeholders, se permite el acceso a este servidor de pruebas a través de la URL "tfg.lluisfalco.com".

4.3 Seguimiento de tareas

Para llevar a cabo el seguimiento de las tareas realizadas, realizando o por realizar, se utilizó la plataforma web Trello [10]. En esta, se abrió un tablero en el cual, se registrarían estas tareas.

El tablero, está compuesto por 6 apartados:

- **Referencias.** En este listado se registraban los documentos, libros, páginas web y otros elementos consultados o de ayuda para la elaboración del proyecto.
- **Ideas.** Este listado registraba las posibles nuevas funcionalidades del sistema, nuevos objetivos, ideas innovadoras.
- **ToDo.** Este apartado registraba las tareas que se tenían que llevar a cabo para el desarrollo del proyecto.
- **Bugs.** Registro de nuevos bugs encontrados en el sistema, junto a una breve descripción de estos.
- **In Progress.** Contiene las tareas que se están realizando actualmente.
- **Done.** Este listado es el que contiene todas las tareas realizadas con éxito.

Las tareas están también organizadas por un sistema de colores que indica la urgencia de estas por realizarse. Estos colores son el rojo para tareas urgentes o muy prioritarias, naranja para tareas prioritarias y amarillo para las poco prioritarias. También existen otros colores como el azul, el púrpura o multicolor, estos se utilizan para indicar que contienen links a la documentación del proyecto, documentos o libros de ayuda que hacen referencia a la temática especificada por el título y información extremadamente importante y útil que se ha redactado como resultado de investigaciones.

Semanalmente, se realizan reuniones con el tutor Daniel Alexander Norton para comprobar el correcto desarrollo del proyecto, discutir nuevas funcionalidades, mejorar procesos e intercambiar opiniones. Daniel, gracias a su amplia experiencia profesional también actuó como un cliente.

5 REQUISITOS

Los requisitos del sistema definen el comportamiento deseado del sistema. A continuación, se listan y explican los principales requisitos planificados para el sistema.

5.1 Requisitos funcionales

Los requisitos funcionales especifican todo aquello que nuestro sistema debe ser capaz de realizar.

Requisitos para usuarios anónimos (Apéndice 3, Figura 14):

1. Visualizar elementos públicos
2. Debe permitir comentar los elementos públicos de forma anónima.
3. Se debe poder registrar nuevos usuarios mediante la introducción de un correo, un nombre y una contraseña.
4. Identificarse especificando el correo y contraseña previamente registrados.

Requisitos para los usuarios identificados (Figura 2):

5. Deben poder visualizar sus elementos y colecciones privadas y las compartidas con él.
6. Podrán puntuar y opinar los elementos y colecciones visibles por ellos de forma no anónima.
7. Se debe poder crear colecciones vacías o con unos elementos especificados.
8. Se debe permitir editar los elementos de las colecciones, junto a sus nombres, descripciones y opciones de privacidad.
9. Se debe poder compartir un elemento con otros usuarios especificando su correo electrónico.
10. Debe permitir al usuario, registrar nuevos elementos mediante la introducción de su dirección URL y un nombre.
11. Debe permitir añadir elementos mediante un buscador interno que muestre elementos de otros sistemas como *Youtube*, *Google*, *Vimeo*, *Flickr*.

5.2 Requisitos no funcionales

Restricciones en el desarrollo:

1. Se dispone de 300 horas para el desarrollo del proyecto hasta Julio.
2. El sistema debe estar disponible las 24h del día.
3. El sistema debe poder ser capaz de llevar a cabo sus funcionalidades en pocas acciones

6 DESARROLLO

El proyecto se ha desarrollado en las etapas de diseño, elección de tecnologías, selección de herramientas e implementación

6.1 Diseño

En esta etapa, una vez definidos objetivos y requisitos del proyecto, se procede al diseño del sistema que esta dividido en el diseño de la interfaz web y el de la base de datos.

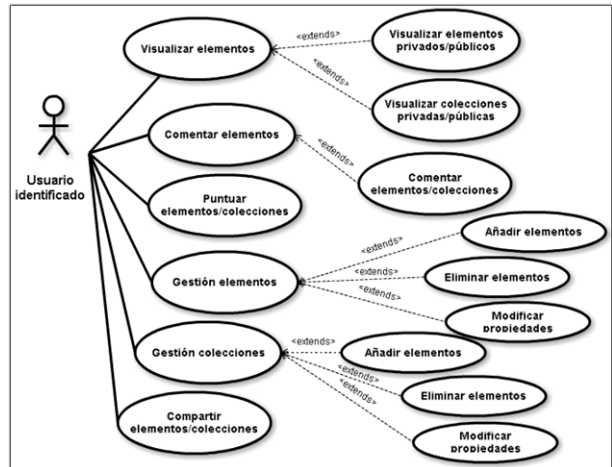


Figura 2: Diagrama de casos de uso para usuarios identificados

6.1.1 Interfaz web

Con el objetivo de simplificar la utilización del sistema, se intenta diseñar las pantallas uniformemente, intentando que las diferencias entre estas sean mínimas. Para ello, se definen cuatro zonas.

- El visor de elementos. Este apartado, nos permitirá visualizar los elementos seleccionados de tal forma que podamos seguir navegando por la página sin necesidad de cerrar esta sección.
- El menú. El menú permitirá a los usuarios moverse entre las diferentes secciones del sistema.
- La zona que lista los elementos visibles (Apéndice 2, Figura 8). En este apartado, se muestran los elementos o colecciones visibles de la sección en la que se encuentre el usuario.
- La zona que muestra las descripciones, puntuaciones y comentarios de los elementos (Apéndice 2, Figuras 9 y 12).

Estas cuatro zonas están compuestas por un seguido de vistas independientes que, siguiendo la arquitectura Modelo-Vista-Controlador (MVC), están controladas por sus respectivos controladores. Estas vistas están diseñadas para poder ser recicladas, de tal forma que faciliten el mantenimiento y simplifiquen la interacción con el sistema (Apéndice 2). La zona que muestra el listado de elementos, es la única que puede ser controlada por más de un controlador y dependiendo de la sección de la página en la que nos encontremos, los controladores irán variando y tomando el control de esta vista.

También se ha priorizado que las acciones requieran el mínimo de clicks posibles y que estas no obliguen a desplazarse a otras secciones para llevarlas a cabo.

6.1.2 Base de datos

Para poder llevar a cabo el proyecto, se ha diseñado una base de datos capaz de almacenar los datos suficientes para cumplir con los objetivos especificados y con otras propiedades que permitan añadir nuevas funcionalidades al sistema.

La base de datos consta de 8 tablas (Figura 3):

- **Comentarios.** Esta tabla almacena los comentarios

realizados en los diferentes elementos del sistema.

- **Compartidos.** Almacena las comparticiones de elementos entre usuarios del sistema.
- **Listas.** Almacena todas las colecciones existentes.
- **Listas_items.** Permite relacionar los elementos con las colecciones a las que pertenecen.
- **Login_usuarios.** Contiene los datos necesarios para permitir hacer login a los usuarios.
- **Media_items.** En esta tabla se almacenan todos los elementos registrados en el sistema.
- **Puntuación.** Almacena todas las puntuaciones recibidas de los diferentes elementos.
- **Usuarios.** Contiene los detalles del perfil del usuario.

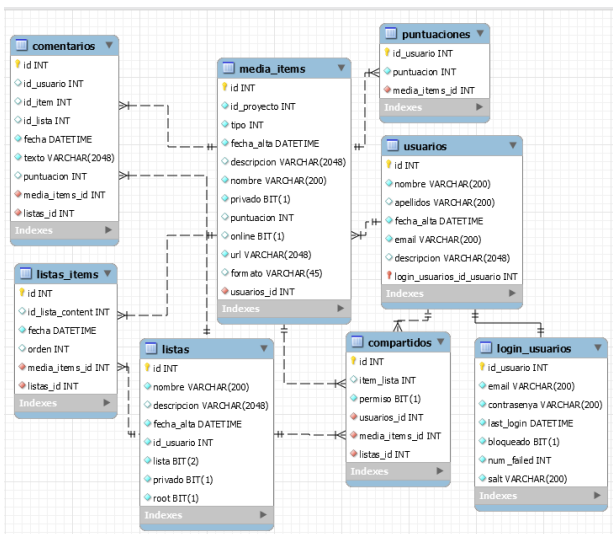


Figura 3: Diagrama relacional de la base de datos

6.2 Tecnologías

Para conseguir un sistema flexible y fácilmente adaptable, se ha separado el sistema en dos bloques, el backend y el frontend.

6.2.1 Frontend

Para poder crear una aplicación web dinámica, el apartado frontend se ha creado utilizando las tecnologías HTML5, CSS3 y Javascript.

Para el desarrollo también se ha utilizado el framework MVC AngularJS de Javascript, para el control de las vistas, el framework Twitter Bootstrap. Dos de las librerías más

importantes utilizadas en este apartado son los de Masonry [11], que es una librería javascript que permite mostrar elementos en forma de cuadrícula y la directiva de Angular angular-drag-and-drop-lists, como base para poder crear el modulo que permite gestionar los elementos de forma fácil e intuitiva.

6.2.2 Backend

En el backend del sistema, se ha optado por desarrollar una API REST, utilizando C# como lenguaje de programación. Este recibirá peticiones Http formuladas por el frontend o cliente y actuará en consecuencia dependiendo de qué tipo de petición se lleve a cabo.

También se han utilizado frameworks como entity framework [14] [15] o Linq [14] [15] con tal de conectar esta API a la base de datos. Con tal de mantener la persistencia de los datos, se ha instalado una base de datos MySQL.

Este tipo de arquitectura permite al sistema expandirse a otro tipo de tecnologías como Android o IOs utilizando el mismo backend para todos.

La comunicación entre ambos sistemas se realiza mediante peticiones Http a las que reciben una respuesta por parte del servidor. En la figura 4, podemos ver como el usuario realiza una acción, el cliente la verifica y hace la petición al servidor que vuelve a verificar los datos. Actúa de forma consecuyente y persiste la información necesaria, tras esto contesta al cliente si se ha podido realizar o no esa acción. La siguiente figura (Figura 5) muestra en un diagrama las tecnologías utilizadas.

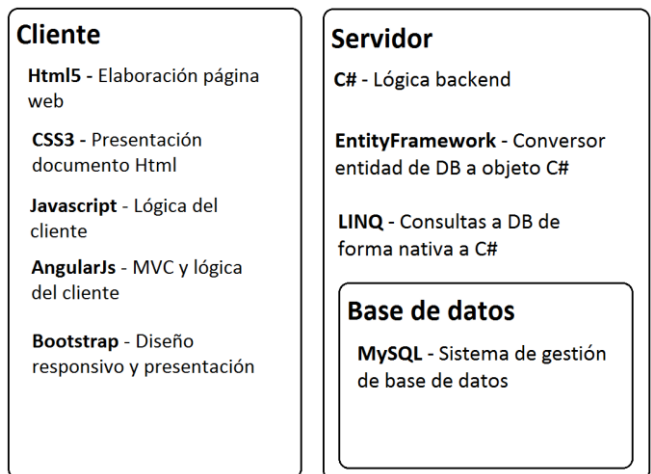


Figura 5: Uso de las tecnologías

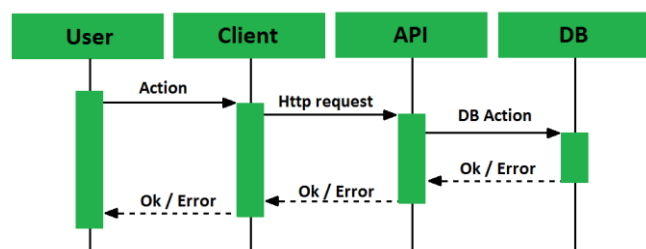


Figura 4: Diagrama de secuencia comunicación cliente-servidor

6.3 Herramientas

Para desarrollar el proyecto se han utilizado las siguientes herramientas:

- *Balsamiq Mockups.* Herramienta utilizada para la realización de los diseños iniciales.
- *Visual Studio 2013.* Entorno de desarrollo
- *MySQL Workbench.* Herramienta de diseño, creación y mantenimiento para el sistema de base de datos MySQL
- *Internet Information Services (IIS).* Conjunto de servicios de Windows que permite convertir

un PC en un servidor web. Utilizado para la configuración del servidor de pruebas y desarrollo.

6.4 Implementación

La implementación del sistema se ha dividido en dos fases. La primera, implementación del núcleo que tiene como objetivo desarrollar las funcionalidades más prioritarias y la implementación del módulo complementario que pretende desarrollar las funcionalidades menos críticas y mejorar el acabado de la interfaz web.

6.4.1 Implementación del núcleo

En esta fase, se empieza el desarrollo del sistema. Se pretende llevar a cabo la funcionalidad básica o crítica de este. En este caso, el sistema desarrollado por este proyecto es una aplicación web que debe ser accesible desde un navegador web a cualquier usuario que quiera entrar en ella. Esta fase contiene varios módulos a implementar.

6.4.1.1 Maquetación general de la web

Una vez se ha decidido el diseño de la página web, se procede a la implementación de este diseño. Esta tarea, da como resultado una página web con todas las secciones y apartados necesarios para el sistema pero sin funcionalidad alguna (Apéndice 3, Figura 13).

A continuación, detallaremos el funcionamiento de las pantallas desarrolladas y las posibles interacciones del usuario con estas. En la pantalla inicial se nos muestra la sección *Home* de la página web que permite visualizar los elementos públicos registrados en el sistema. Estos se muestran en forma de cuadrícula donde cada elemento ocupa una sección que contiene su nombre, su puntuación, parte de la descripción y permite visualizar el contenido en un espacio reducido. También contienen 2 botones, el primero de los cuales te abre un pop-up que dependiendo de los permisos que tenga el usuario te permitirá editar el elemento, añadirlo o eliminarlo de tus colecciones y compartirlo con otro usuario mediante una serie de pequeños formularios. El segundo botón, nos da acceso a la sección que nos permite visualizar más detalladamente el elemento, su descripción, puntuación o los comentarios asociados a este.

A la izquierda de esta pantalla reservamos un espacio que nos permitirá visualizar, en caso de estar identificado, los contenidos del usuario y que mediante un botón que abre un pop-up nos permitirá registrar nuevos elementos. Para ello deberemos especificar la dirección URL, un nombre y un tipo de archivo para poder registrarlo. También nos dará la opción de especificar una descripción, una configuración de privacidad y de añadirlo inmediatamente a una colección específica.

En la parte superior observamos la barra del menú que nos permite hacer una búsqueda de los elementos que nos muestra la cuadrícula. Si el usuario no se encuentra identificado nos muestra un botón que nos abrirá el pop-up con dos formularios diferentes, uno de ellos nos dará la opción de identificarnos mediante la introducción de un correo y una contraseña y el otro nos permite registrarnos en el sis-

tema pidiendo adicionalmente un nombre de usuario, también podremos cerrar esta pantalla mediante el botón *Close* que aparece en la parte inferior. En caso de estar identificado en la barra superior se muestran 3 secciones adicionales cada una estas secciones nos muestra un tipo diferente de elementos. La primera es la sección *Shared*, esta nos muestra todos los elementos o colecciones que se han compartido con el usuario identificado. La segunda sección *My Contents* muestra todos los elementos propios del usuario, sin distinción de las colecciones a las que pertenezcan. Por último encontramos la sección *My collections*, esta nos permite visualizar todas las colecciones del usuario identificado y un botón situado en la parte superior abrirá un nuevo pop-up que nos dará la opción de añadir una nueva colección mediante la introducción de un nombre.

Todas estas secciones explicadas anteriormente se muestran en el mismo formato de tipo cuadrícula permitiendo así reciclar la vista principal y ofreciendo diferentes funcionalidades gracias a los diferentes controladores de las vistas.

6.4.1.2 Creación de la base de datos

La siguiente tarea en la implementación del sistema es la creación de la base de datos. En esta se crean todas las tablas y campos necesarios definidos en el diseño de la base de datos y para ello, se utiliza la herramienta MySQL Workbench. Primero creamos una base de datos vacía y seguidamente utilizando como referencia el diseño de la base de datos, procedemos a crear todas las tablas con sus respectivas propiedades.

6.4.1.3 Implementación módulo gestión de elementos

Esta actividad pretende implementar la funcionalidad más básica del sistema la de adición, eliminación, visualización y edición de elementos multimedia. Para ello, se implementa el controlador *MediaItemController*. Esta clase de tipo controlador contiene un seguido de funciones que mediante peticiones Http pueden ser ejecutadas, que llevan a cabo las respectivas comprobaciones de seguridad y requisitos y que se encargan de gestionar los elementos multimedia.

Una vez implementadas estas funciones en el backend del sistema, se procede a implementar la lógica en frontend. Para ello acudimos a los controladores de las vistas que necesiten estas opciones donde implementamos la lógica que nos permite crear los objetos JSON necesarios y/o hacer la petición adecuada a la API.

Un ejemplo lo encontramos en la función *insertItem(Objeto)* que recibe como parámetro un objeto JSON elaborado en el cliente y que transforma a otro objeto de tipo *MediaItem*. Por último en caso de ser todo correcto lo guarda en la base de datos junto a los datos del usuario al que pertenece. Otro ejemplo lo encontramos en la función *deleteItem(id)* a la que se puede acceder a través de peticiones de tipo *delete* a la URL <http://luisfalco.no-ip.biz:8080/MediaItem/{id}>, esta recibe como parámetros el identificador del elemento a eliminar y seguidamente comprueba que es de la propiedad del usuario que ha hecho la petición, entonces busca el elemento, y lo elimina por completo de sus colecciones y del

sistema.

6.4.1.4 Visualización de los elementos

Esta actividad, comprende las tareas necesarias para recoger y mostrar los elementos previamente registrados. Esta extracción de contenidos debe poder diferenciarse por categorías, como por ejemplo, extraer todos los elementos públicos, los elementos compartidos con el usuario o los elementos del usuario identificado, etc. Para ello recurrimos de nuevo a la clase controladora *MediaItemController del backend* donde se implementan las funciones necesarias para extraer los elementos de la base de datos. Cada posible modalidad de estas peticiones requiere una función diferente que sea capaz de comprobar los requisitos y la seguridad necesarios, que pueda extraer la información necesaria y la devuelva al cliente. Una vez implementadas estas funciones procedemos al frontend del sistema implementando la nueva lógica que permitirá al cliente realizar las peticiones apropiadas en el formato correcto. Un ejemplo lo encontramos en la función *getPublic()*, que se llama a través de las peticiones dirigidas a la URL <http://lluís.falco.no-ip.biz/MediaItem/public> que recaba los datos de los elementos públicos y los devuelve al controlador de la vista. Por último, mediante la directiva de angular ng-repeat, estos datos quedan reflejados en la vista.

6.4.1.5 Implementación del módulo que permite comentar, puntuar y editar los elementos

Este módulo permite a los usuarios ver en detalle el elemento seleccionado y también permite a los usuarios registrados puntuar los elementos y comentarlos. Para la implementación del módulo de puntuaciones y de las ediciones de los elementos, seguimos utilizando la clase controladora del backend *MediaItemController* donde definimos la función que permiten a los usuarios puntuar los elementos *setRate(idItem, puntuacion)*. Esta función recibe como parámetros el identificador y la puntuación que se le ha dado a un elemento, calcula la media con todas las puntuaciones existentes del elemento y la guarda. Como esta funcionalidad solo existe en la pantalla que muestra los detalles de los diferentes elementos, acudimos a su controlador donde implementamos la petición PUT a la URL <http://lluísfalco.no-ip.biz:8080/MediaItem/rate/{itemId}/puntuacion> que permite el registro de las puntuaciones.

Con tal de gestionar los comentarios de los diferentes elementos o colecciones, implementamos la nueva clase *CommentsController* que contiene todos los métodos necesarios para gestionar los comentarios. Este controlador solo contiene 5 funciones que permiten tomar los comentarios, añadir nuevos, editarlos o eliminarlo y una vez implementadas, acudimos al frontend del sistema al controlador de la vista que permite comentar los elementos e implementamos las diferentes peticiones para los comentarios.

6.4.1.6 Implementación del módulo de gestión de usuarios

Este módulo permite al sistema el registro e identificación de los usuarios para así poder acceder a su espacio. Para ello, implementamos la nueva clase *AccountController*

que tiene todos los métodos necesarios para el registro, login y logoff de los usuarios. Esta nueva clase, contiene 3 funciones, *login()* es la encargada de permitir a los usuarios previamente registrados identificarse y para ello recibe un objeto JSON que contiene el usuario y contraseña introducidos y comprueba que el hash generado por esta contraseña y el salt del usuario, coincide con el hash correspondiente a la contraseña de la base de datos. La función *register()* se encarga de registrar nuevos usuarios en el sistema, esta recibe como parámetros el formulario de registro en forma de objeto JSON, comprueba que no existan otros usuarios con el mismo correo ya registrados, hace el hash de la contraseña y genera el salt para guardarlo en la base de datos. Por último la función *logout()* cierra la sesión del usuario que ha formulado la petición. Para controlar el estado de las sesiones en el backend del sistema utilizamos la clase de C# *FormsAuthentication* [13], esta permite autenticar a los usuarios y mantener un token de identificación en una cookie.

Para controlar el estado de las sesiones y su información en el frontend del sistema, utilizamos un servicio de AngularJS [12]. Un servicio es un objeto javascript que mantiene los datos en todo el ciclo de vida de la aplicación y que contiene información que puede ser consultada por los diferentes controladores. Un ejemplo lo podemos ver al intentar hacer un login, al rellenar el formulario y hacer click se hace la petición Http a la URL <http://lluísfalco.no-ip.biz:8080/Usuario/log> si la petición y la información proporcionada es correcta, el backend responderá con la información del usuario y seguidamente esta se almacenara en el servicio de angular para poder utilizar la información de este.

6.4.1.7 Implementación del módulo de creación y gestión de colecciones

Este módulo permite a los usuarios crear sus propias colecciones de elementos organizándolos así a su gusto. Una vez se ha creado una colección, se puede acceder a su descripción, nombre, puntuación o módulo de comentarios y se permite también añadir los elementos existentes a las colecciones existentes. Para ello implementamos la nueva clase "ListaController" que implementa todos los métodos necesarios para la gestión de colecciones y a los que podemos acceder mediante peticiones Http.

6.4.2 Implementación del módulo complementario

En esta fase del desarrollo, cuando las funcionalidades básicas ya han sido implementadas y probadas, se procede a realizar las tareas con funcionalidades menos críticas, mejorar los procesos del sistema y optimizar las interacciones con los usuarios.

6.4.2.1 Implementación del módulo de comparticiones

Este módulo, permite a los usuarios, compartir elementos o colecciones con otros usuarios mediante la introducción del email de registro del usuario destino. El usuario puede escoger conceder permisos de administración sobre el elemento o no al usuario destino.

Para implementar este módulo, utilizamos las clases *CollectionController* y *MediaItemController* ya existentes donde se implementan los métodos necesarios para realizar comparaciones de colecciones y elementos sueltos. Cada una dispone de 2 métodos, uno para compartir y otro para revocar la compartición. Estas funciones solo pueden ser accedidas por los usuarios propietarios de los elementos o colecciones, de tal forma que solo el propietario es capaz de compartir sus elementos.

Una vez implementados los métodos necesarios en el backend, nos dirigimos al frontend al controlador de la vista que permite la configuración de los elementos o colecciones y en este implementamos las pertinentes peticiones Http post o delete para conceder o revocar comparticiones.

6.4.2.2 Mejora estilos, diseño, procesos y experiencia de usuario

Esta fase contiene las tareas dedicadas a la mejora de la experiencia de usuario. Una vez implementadas y probadas las diferentes funcionalidades se mejoran los estilos de la página, la interacción y los procesos de esta.

La tarea principal de esta fase corresponde a la implementación de un árbol en forma de directorios (Apéndice 4, Figura 15) que permite al usuario una interacción más fácil e intuitiva gracias al sistema drag & drop de elementos o colecciones.

Para llevar a cabo ese sistema de organización más rápido e intuitivo se utiliza la directiva de angular “drag-and-drop-lists”. Esta librería ofrece la base de esta nueva funcionalidad permitiendo a los elementos mostrarse en forma de listas o árbol de directorios y permitiendo el drag and drop de los elementos o colecciones.

Para poder utilizar esta funcionalidad se tiene que adaptar esta librería mediante una serie de modificaciones que permitan a los elementos guardar sus nuevos estados y evitar mal funcionamiento del sistema.

Este proceso da como resultado un árbol de directorios con la capacidad de ser ordenador o redistribuido mediante arrastres de elementos a sus nuevos contenidos. También permite que mediante la acción de mantener una tecla del teclado pulsada como Ctrl o Shift mientras se realiza el arrastre o click, esta produzca como resultado, copias o agregue los elementos seleccionados al visor.

La colección de nombre *CAOS collection* es la colección base de los elementos que no han sido ordenados o clasificados todavía en el sistema. Esto es así ya que Daniel, en el rol de cliente y profesional en video DJ aconsejo implementar, siendo la práctica habitual en un gran porcentaje de los profesionales del sector.

6.4.2.3 Implementación del módulo de búsqueda interno

En este módulo se pretende, implementar un buscador que permita añadir nuevos elementos de una forma fácil y rápida. Este mostraría resultados de los diferentes sistemas existentes tales como Youtube, Google images, Flickr, Vimeo,...

Este módulo se ha aplazado para futuras líneas a causa de los diferentes problemas e incidencias que surgieron con las tareas anteriores.

7 RESULTADOS

El resultado obtenido por el proyecto ha sido un sistema capaz de gestionar diferentes tipos de elementos multimedia y organizarlos a gusto del usuario (Apéndice 4). Estos elementos que pueden ser de video, audio, imagen o documento pdf, se gestionan de forma heterogénea e unificada y esto permite al sistema tratarlos de la misma forma facilitando así su organización y aportando el mayor de los valores al sistema.

Se han cumplido en gran medida los objetivos propuestos al inicio del proyecto, pero el margen de mejora que ofrece este sistema es aún muy amplio.

A continuación mostramos la tabla de objetivos alcanzados

Tabla 1: Tabla de objetivos alcanzados

O.1	Cumplido en gran medida, pero no se ha logrado completar al 100% por la falta de gestión de elementos con extensiones comunes como Word o la elaboración de un editor de texto.
O.2	Cumplido al 100% permitiendo a los usuarios, describir, puntuar y opinar sobre los diferentes elementos o colecciones.
O.3	Cumplido al 100% con la creación del módulo de gestión de usuarios.
O.4	Cumplido también en gran medida, pero no podemos afirmar que el sistema sea 100% funcional en todos los navegadores por la diferencia en sus comportamientos.
O.5	Cumplido al 100% con la creación del módulo de gestión de colecciones
O.6	Cumplido al 100% con la creación del módulo que permite comparticiones

La planificación inicial generada en el proyecto, fue seguida de una forma correcta y con precisión hasta el último tramo del desarrollo del proyecto, donde se tuvo que volver a plantear las últimas tareas a causa de la extensión de otras.

Respecto a los requisitos formulados inicialmente para el proyecto, se han cumplido todos a excepción del buscador interno que ofrecía la posibilidad de añadir nuevos elementos encontrados en otros sistemas, que se aplazó para líneas futuras a causa de la extensión de algunas de las tareas más críticas. La tarea de intentar mejorar el diseño, los procesos y la experiencia de usuario se ha extendido más de lo planeado inicialmente y los resultados obtenidos no acaban de reflejar la inversión de tiempo que se ha dedicado a ella y por ello se decidió alargar más esta aplazando así la siguiente y última actividad en el desarrollo, el buscador interno. Finalmente, esta tarea ha dado frutos con la generación del árbol de directorios y la mejora en los estilos y diseños de la página web.

8 CONCLUSIONES

Podemos observar que el proyecto ha dado como resultado un prototipo del sistema que a pesar de disponer de la gran mayoría de las funcionalidades requeridas, aún precisaría

de un poco más de trabajo para poder presentarlo como una versión estable. Una vez realizadas estas tareas, este podría ser un producto comercializable. Tras el desarrollo de este proyecto, se continuará trabajando con el sistema y se empezará a utilizar de forma directa para poder así descubrir e implementar nuevas funcionalidades o soluciones. A lo largo de este trabajo, han surgido una gran cantidad de situaciones y problemas nuevos que poco a poco se han podido ir solucionando, algunos de ellos ya se tenían en cuenta en la planificación inicial, dejando así un margen de tiempo para poder superarlos y otros sin embargo no estaban planificadas tales como bugs en algunos frameworks utilizados o el incorrecto diseño del sistema que no contemplaba en algunos casos el nivel de detalle que era necesario. En algunas ocasiones estos problemas obligaron al proyecto a sufrir diversas modificaciones en el diseño y ayudaron a que el análisis de las posibles soluciones o alternativas se llevaran a cabo de una forma más exhaustiva. Como reflexión personal, estoy muy satisfecho con la planificación inicial, ya que pese a mi corta experiencia con proyectos de tal envergadura, se aproximó mucho a la realidad hasta las últimas semanas de proyecto y me ha permitido adquirir nuevas habilidades en el desarrollo. De una forma indirecta, el trabajo me ayudó adentrarme en el mundo de los servidores, de su configuración, de los dominios y me animo a experimentar de una forma más activa con nuevos proyectos.

9 LÍNEAS FUTURAS

En este apartado, se explican los módulos o funcionalidades que no se han podido implementar en este proyecto por falta de tiempo:

- Implementación de un módulo que permita la subida de archivos y registro de estas en el sistema automáticamente. Este, mediante la cuenta facilitada por el usuario, subiría en archivo vía API a una de las principales plataformas y registraría el elemento en el sistema.
- Implementación de una edición de texto online, que permita la creación de documentos online editables.
- Generación de URL's asociadas a un elemento o colección, que permitirían acceder al elemento sin la necesidad de buscarlo por el sistema.
- Adición de proyectos y equipos al sistema, de tal forma que un equipo definido por un número específico de usuarios, podrán tener acceso a varios colecciones sin necesidad de conceder permisos.

10 AGRADECIMIENTOS

A todos aquellos que me ayudaron en la captura de requisitos, con grandes y innovadoras ideas para el sistema. A mi familia, por escuchar atentamente las nuevas ideas y opinar sobre ellas.

A mis tutores, Fernando Luis Vilariño y Daniel Alexander Norton por guiarme a lo largo del proyecto, por sus ideas opiniones e consejos que me ayudaron a aclarar el camino que debía seguir, por el valioso tiempo que me dedicaron

y por motivarme a seguir con él.

11 BIBLIOGRAFÍA

- [1] Definición de youtube [Online]. Disponible: <https://es.wikipedia.org/wiki/YouTube>
- [2] Definición de Google Imagenes [Online]. Disponible: https://es.wikipedia.org/wiki/Google_Im%C3%A1genes
- [3] Definición de Flickr [Online]. Disponible: <https://es.wikipedia.org/wiki/Flickr>
- [4] Definición de instagram [Online]. Disponible: <https://es.wikipedia.org/wiki/Instagram>
- [5] Definición de Spotify [Online]. Disponible: <https://es.wikipedia.org/wiki/Spotify>
- [6] Definición de Coursera [Online]. Disponible: <https://es.wikipedia.org/wiki/Coursera>
- [7] Definición de Google Drive [Online]. Disponible: https://es.wikipedia.org/wiki/Google_Drive
- [8] Definición de Dropbox [Online]. Disponible: <https://es.wikipedia.org/wiki/Dropbox>
- [9] Wikipedia, Desarrollo en espiral [Online]. Disponible: https://es.wikipedia.org/wiki/Desarrollo_en_espiral
- [10] Ingeniería del Software [Online]. Disponible: <http://software1nathalygri-jalva.blogspot.com.es/2012/10/modelo-espiral.html>
- [11] Masonry web oficial [Online]. Disponible <http://masonry.desandro.com/>
- [12] ¿Que es un servicio?, 2015 [Online] . Disponible: <http://java-white-box.blogspot.com.es/2015/06/angularjs-que-es-un-servicio-que-es-un.html>
- [13] FormsAuthentication (Clase). [Online], Disponible: [https://msdn.microsoft.com/es-es/library/system.web.security.formsauthentication\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.web.security.formsauthentication(v=vs.110).aspx)
- [14] Información general de Entity Framework [Online]. Disponible: [https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)
- [15] Joe Mayo, C# Succinctly, Syncfusion 2015

APÉNDICE

A1. DIAGRAMA DE GANT

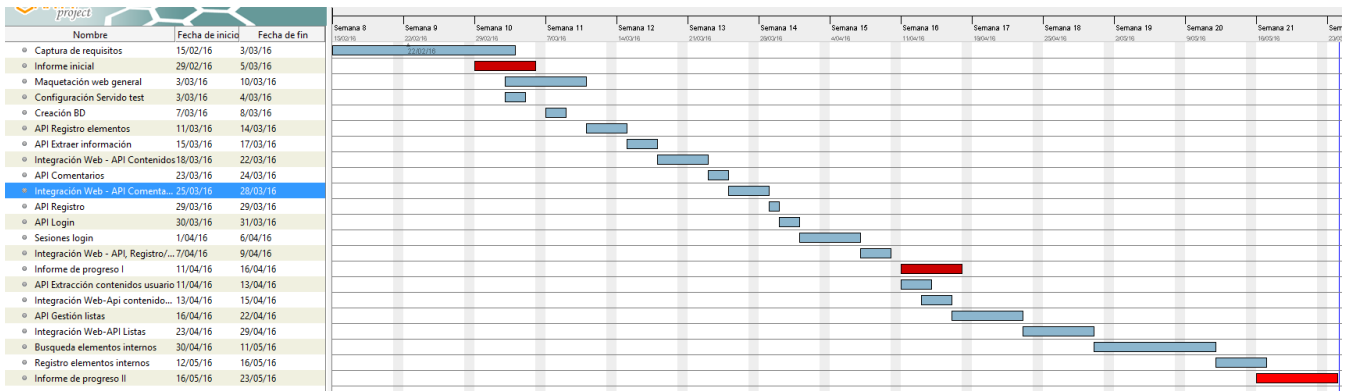


Figura 6: Diagrama de gant de la planificación

A2. DISEÑO INICIAL

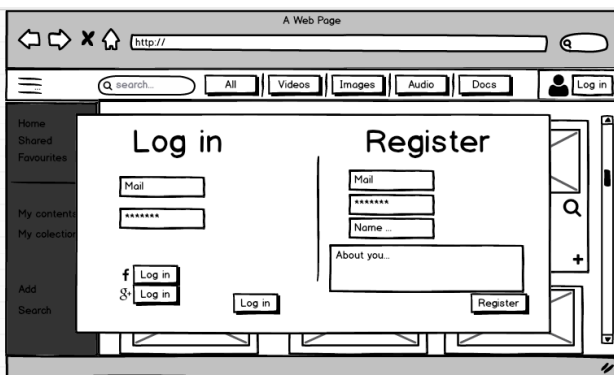


Figura 7: Diseño inicial login

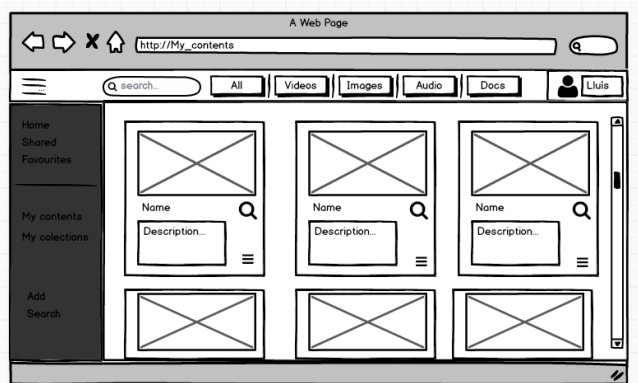


Figura 8: Diseño inicial visor de elementos

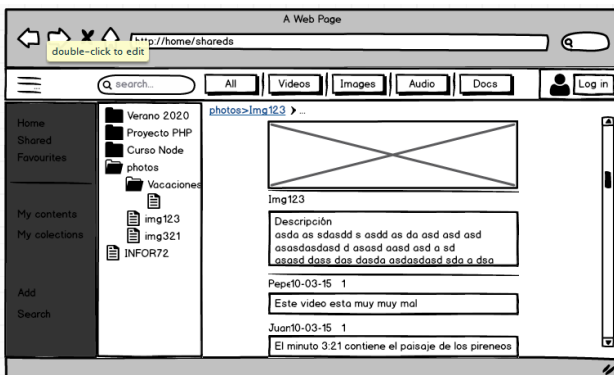


Figura 9: Diseño inicial, visor de descripción, puntuación y comentarios

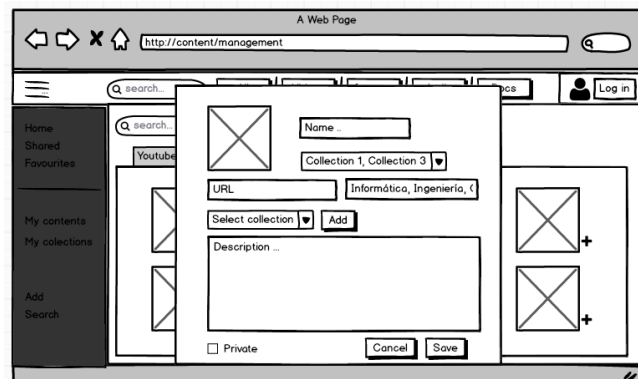


Figura 10: Diseño inicial, adición de contenido

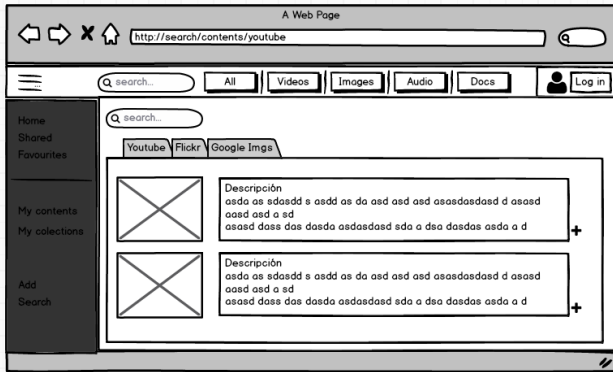


Figura 11: Diseño inicial, buscador interno

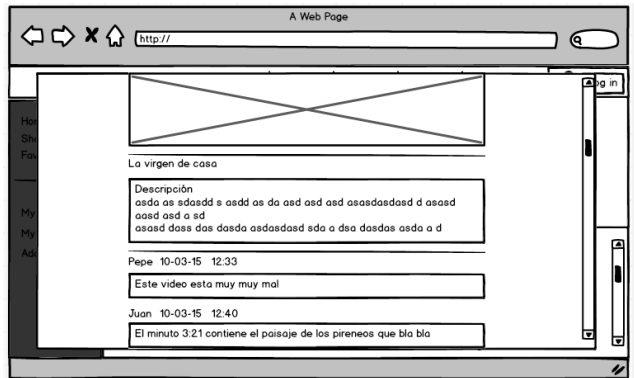


Figura 12: Diseño inicial, pop-up visor de descripción, puntuación y comentarios

A3. MAQUETACIÓN INICIAL Y DIAGRAMA DE CASOS DE USO

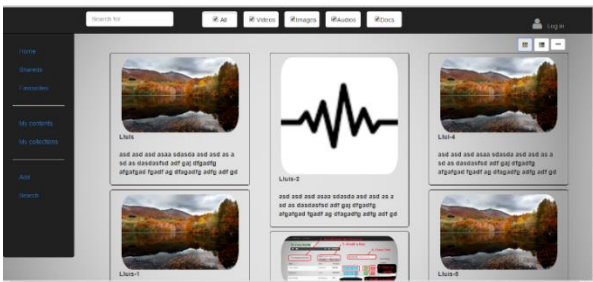


Figura 13: Maquetación inicial interfaz web

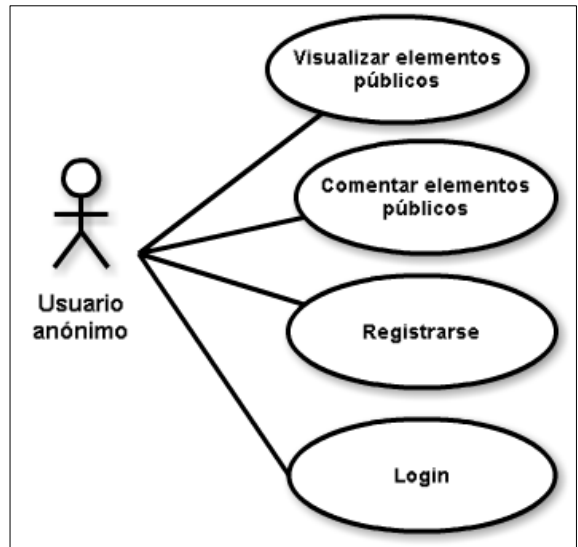


Figura 14: Diagrama de casos de uso para usuarios no identificados

A4. ÁRBOL DE DIRECTORIOS

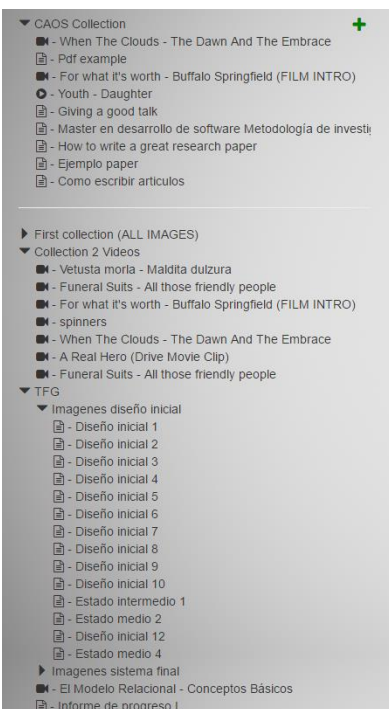


Figura 15: Árbol de directorios drag & drop

A5. SISTEMA FINAL

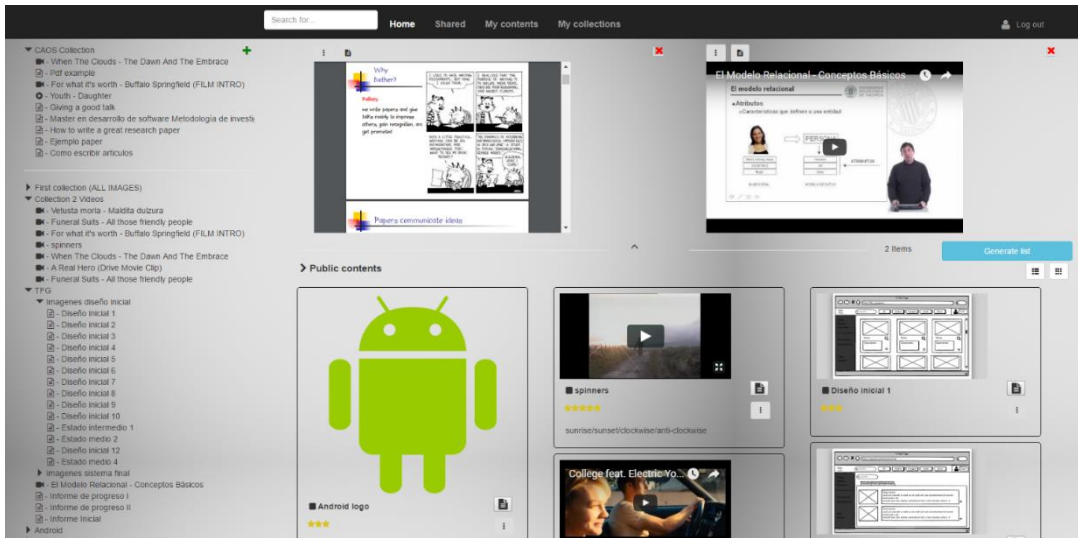


Figura 16: Visor de elementos final

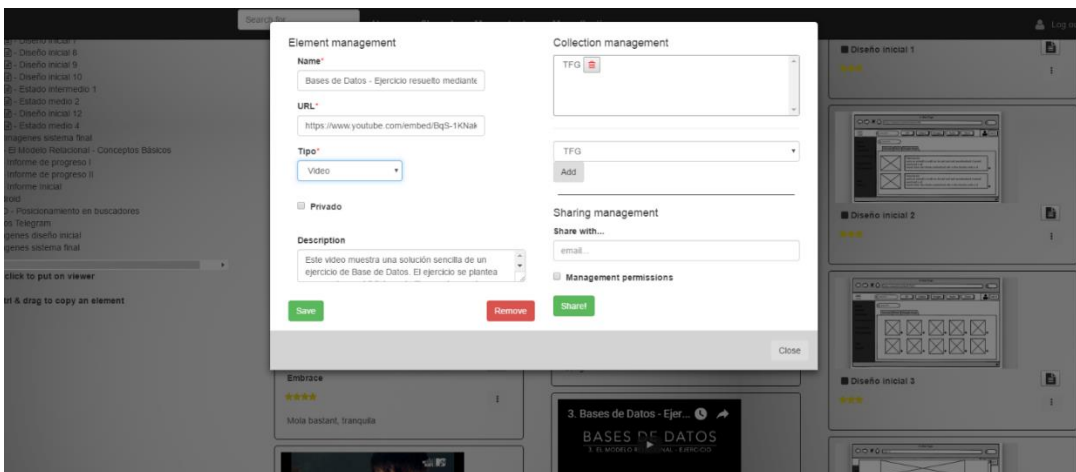


Figura 17: Gestión de contenidos o colecciones



Figura 18: Pantalla que permite la visualización de la descripción, puntuación y comentarios

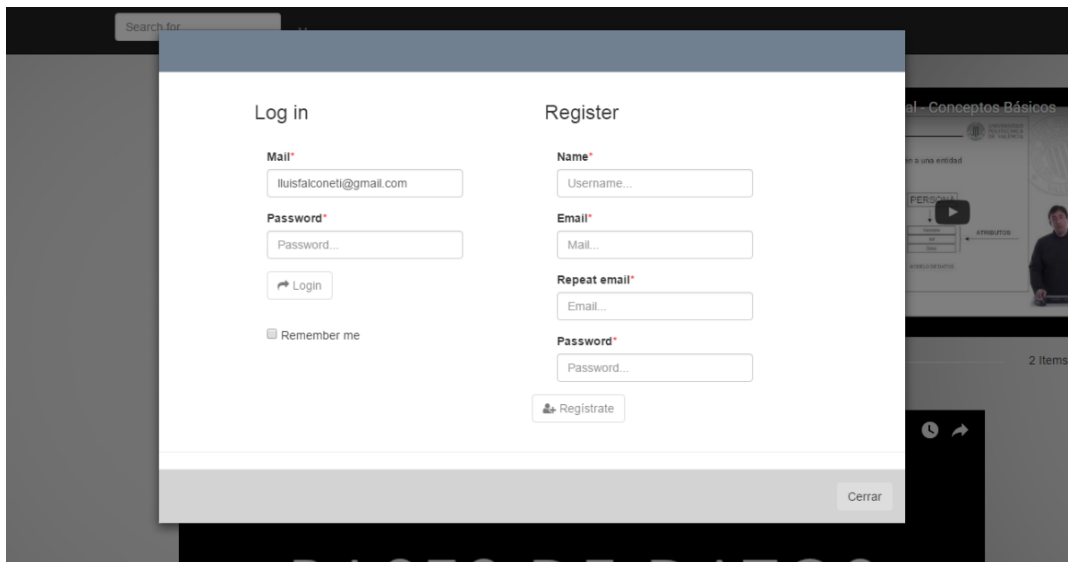


Figura 19: Pantalla de login o registro de usuarios