



Grupo 01



13 DE MAYO, 2019

Proyecto TAD 2019

Índice

[**1.** **VISIÓN DE LA APLICACIÓN** 2](#_Toc8249274)

[1.1. Definición 2](#_Toc8249275)

[1.2. Objetivos 2](#_Toc8249276)

[1.3. Restricciones Funcionales 3](#_Toc8249277)

[1.4. Esbozo de las interfaces de usuario 4](#_Toc8249278)

[1.4.1. Login 4](#_Toc8249279)

[1.4.2. Registro 5](#_Toc8249280)

[1.4.3. Suscripciones 6](#_Toc8249281)

[1.4.4. Buscador 6](#_Toc8249282)

[1.4.5. Ver Video 7](#_Toc8249283)

[1.4.6. Subir Vídeo 7](#_Toc8249284)

[1.4.7. Subir Miniatura 8](#_Toc8249285)

[1.4.8. Mi Canal 8](#_Toc8249286)

[1.4.9. Administrar Perfil 9](#_Toc8249287)

[1.5. Esquema de navegación 9](#_Toc8249288)

[**2.** **DISEÑO DE LA APLICACIÓN** 10](#_Toc8249289)

[2.1. Diagrama de casos de uso 10](#_Toc8249290)

[2.2. Componentes 11](#_Toc8249291)

[2.3. Clases de la aplicación 14](#_Toc8249292)

[2.4. Estructura física de la base de datos 15](#_Toc8249293)

[**3.** **IMPLEMENTACIÓN** 17](#_Toc8249294)

[3.1. Detalles de la implementación 17](#_Toc8249295)

[**4.** **MANUAL DE USUARIO** 20](#_Toc8249296)

[4.1. Inicio de sesión y registro 20](#_Toc8249297)

[4.2. Suscripciones 21](#_Toc8249298)

[4.3. Ver vídeo 21](#_Toc8249299)

[4.4. Subir vídeo 21](#_Toc8249300)

[4.5. Buscar un vídeo 21](#_Toc8249301)

[4.6. Ir a un canal 21](#_Toc8249302)

[4.7. Administrar canal 21](#_Toc8249303)

[4.8. Cerrar sesión 21](#_Toc8249304)

# **VISIÓN DE LA APLICACIÓN**

## Definición

El objetivo de este proyecto es desarrollar una plataforma que permita a sus usuarios ver, interactuar y gestionar varios aspectos de los vídeos que se suban a la página web. Para ello, una vez que el usuario se registre, este podrá buscar vídeos de temas que le interese e incluso subir sus propios vídeos a la plataforma para que los vean el resto de los usuarios.

Una vez el usuario encuentre un vídeo que quiera ver gracias a su título, miniatura y visitas, este podrá indicar qué le ha parecido el vídeo mediante los botones de like y dislike habilitados para ello y dejando un comentario en la sección de comentarios del vídeo. El usuario también podrá suscribirse a diversos usuarios, para así no perderse nunca los vídeos que suban a la plataforma y además acceder al canal de distintos usuarios para ver todos sus vídeos subidos.

De la misma forma, el usuario también podrá gestionar su canal, teniendo la opción de cambiar su foto de perfil, cambiar la miniatura de un vídeo o borrando uno de sus vídeos.

## Objetivos

La aplicación a desarrollar deberá cumplir con los siguientes objetivos:

|  |  |  |
| --- | --- | --- |
| **Código** | **Objetivo** | **Descripción** |
| **OBJ-01** | Login | El sistema permitirá a los usuarios iniciar sesión para acceder a las funcionalidades básicas de la aplicación, y, a los administradores, acceder a funcionalidades avanzadas. |
| **OBJ-02** | Registro | El sistema permitirá a los usuarios registrarse para poder acceder a la plataforma. |
| **OBJ-03** | Ver vídeos | El sistema permitirá a los usuarios registrados ver vídeos y se les mostrará información básica del vídeo como el número de visitas, su fecha de publicación y la cantidad de likes y dislikes que tiene. Además, los usuarios registrados tendrán un control básico sobre el vídeo como ir hacia delante, hacia atrás o mostrar el vídeo en pantalla completa. Finalmente, también se tendrá un control sobre el volumen del vídeo |
| **OBJ-04** | Escribir Comentarios | Los usuarios registrados podrán escribir un comentario en los vídeos de la plataforma. Los comentarios irán acompañados del nombre del usuario y de su fecha de publicación. |
| **OBJ-05** | Puntuar vídeos | Los usuarios registrados podrán interactuar con el vídeo que están viendo, decidiendo si pulsar el botón de like o dislike |
| **OBJ-06** | Suscripciones | Los usuarios registrados podrán suscribirse a otros usuarios y tendrán una opción llamada Suscripciones en la que podrán ver los últimos vídeos de los usuarios a los que esté suscrito. |
| **OBJ-07** | Subir Vídeo | Los usuarios registrados podrán subir vídeos a la plataforma, y el vídeo deberá ir acompañado de una miniatura para llamar la atención de los usuarios y así quieran ver el vídeo. Los vídeos además tendrán un título, para que así sea fácilmente encontrado mediante un buscador. |
| **OBJ-08** | Canal | Los usuarios registrados tendrán una página única para ellos, en la que se podrán visualizar todos sus vídeos subidos a la plataforma. |
| **OBJ-09** | Administración de cuenta | Los usuarios registrados podrán, desde una opción habilitada para ello, cambiar su información básica o su contraseña o su foto de perfil. Además, podrán visualizar el rendimiento de su canal según unas gráficas con el crecimiento de likes, visitas y seguidores de los últimos 6 meses |
| **OBJ-10** | Buscador | Los usuarios registrados podrán buscar vídeos a partir del título del vídeo que quieran ver. |

## Restricciones Funcionales

Se aplicarán las siguientes restricciones:

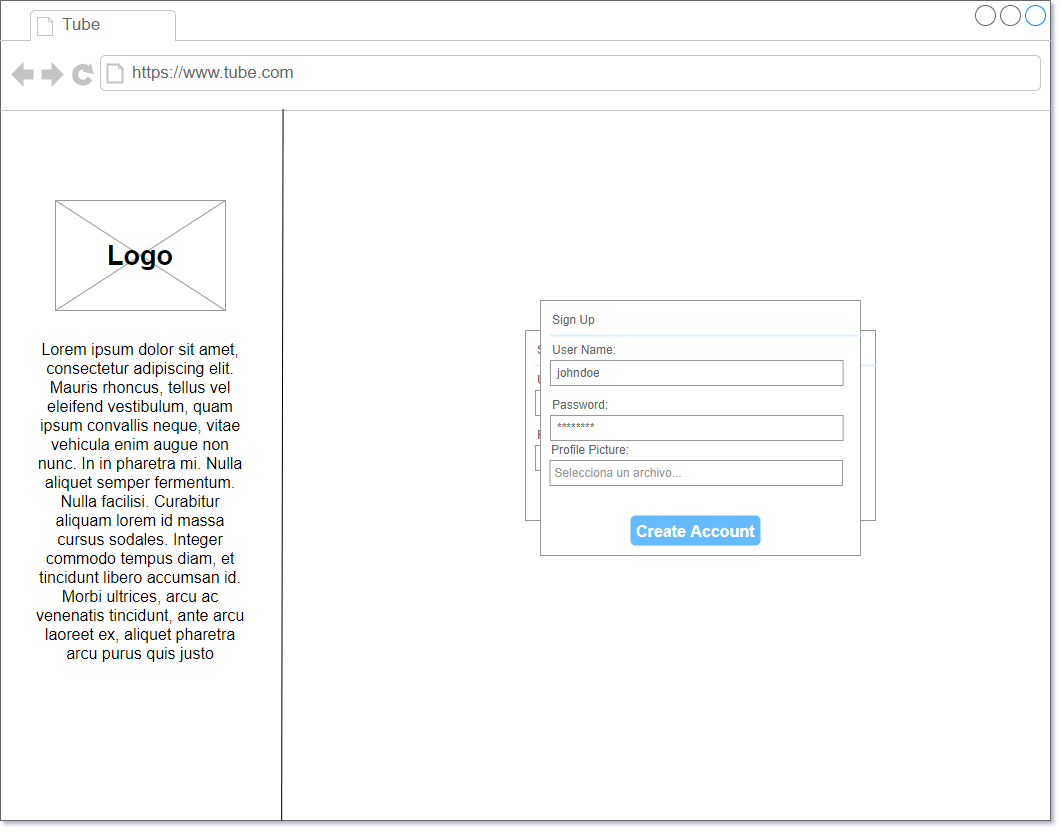
* **Vídeos:**
  + Solamente se admitirá la subida de archivos de vídeos y además estos deberán estar en formato .mp4.
  + El título del vídeo será el nombre del archivo de vídeo sin la extensión.
  + No se podrán modificar los datos de un vídeo una vez subido.
* **Miniatura de vídeo:**
  + Si no se sube una miniatura al subir un vídeo, se le aplicará una miniatura por defecto.
  + Solamente se admitirá la subida de archivos de imagen y estos deberán estar en un formato .png
* **Suscripciones:**
  + Un usuario podrá suscribirse a su propio canal.
* **Likes y dislikes:**
  + Un usuario no podrá dar más de un like o dislike a un vídeo.
  + Si un usuario cambia de like a dislike o viceversa, primero se le quitará del contador de donde estaba antes y se le aplicará hacia el contador del botón que pulsó. Es decir, si un usuario ha dado antes a like a un vídeo y ahora pulsa el botón de dislike, se quitará el like del contador de likes y se le agregará al contador de dislikes
* **Comentarios:**
  + Un usuario no podrá borrar su comentario una vez publicado.
* **Administradores:**
  + Obtendrán sus credenciales directamente desde la base de datos y no podrán participar en la plataforma como usuarios normales.

## Esbozo de las interfaces de usuario

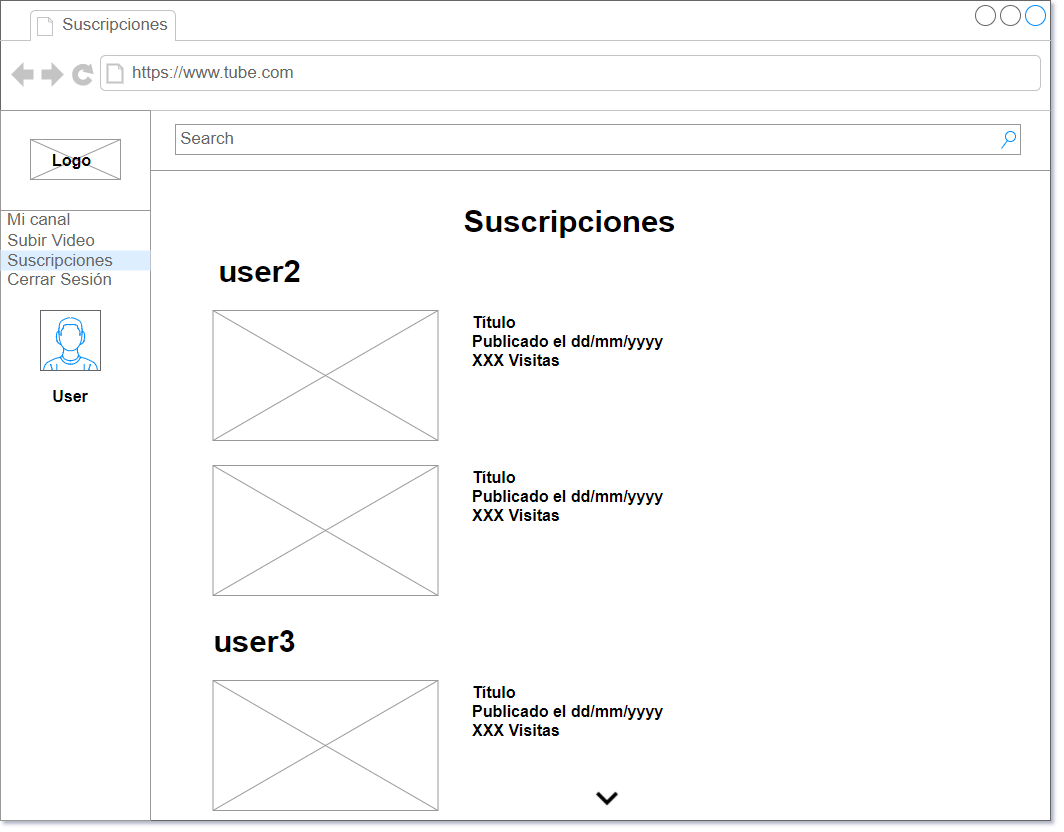
A continuación, se muestran los esbozos realizados para la aplicación y sobre los cuales se basará la implementación de la aplicación:

## Login

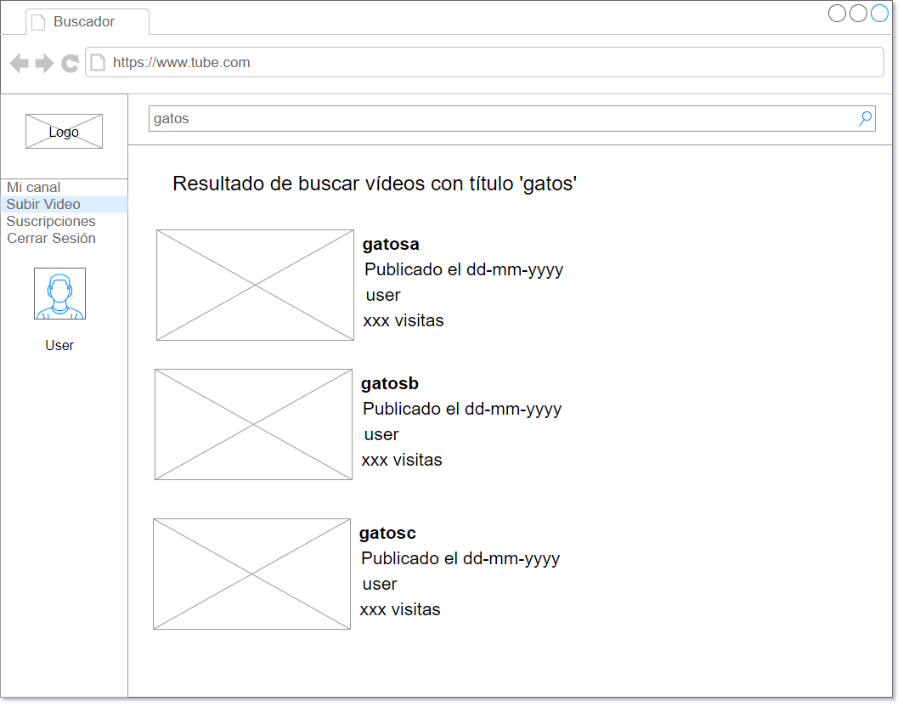
## Registro



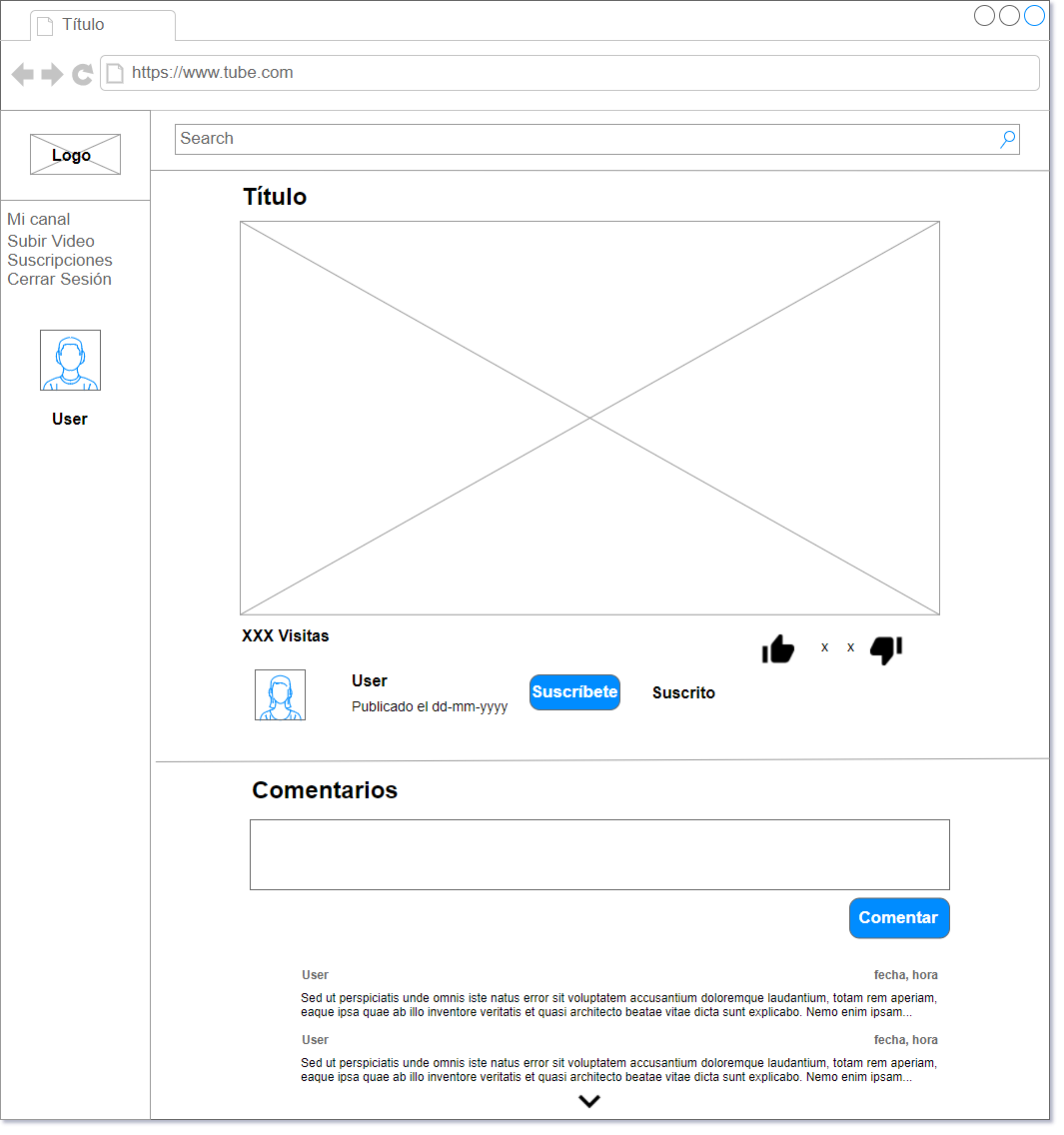
## Suscripciones



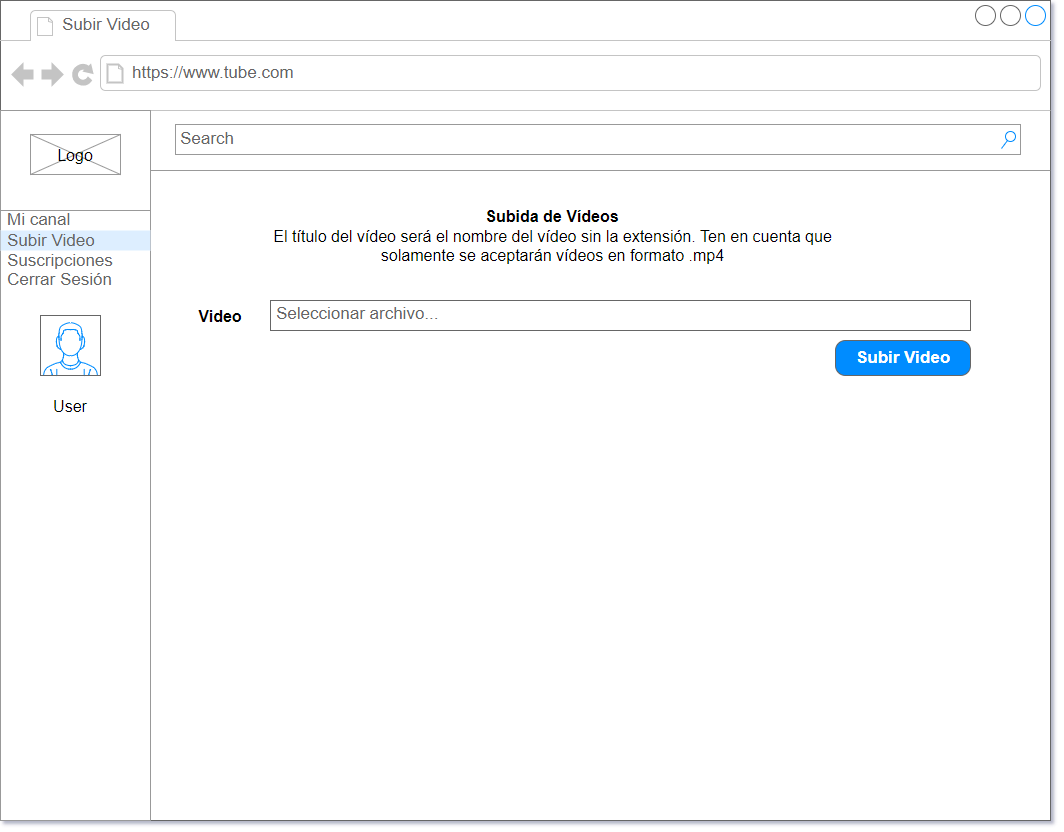
## Buscador



## Ver Video



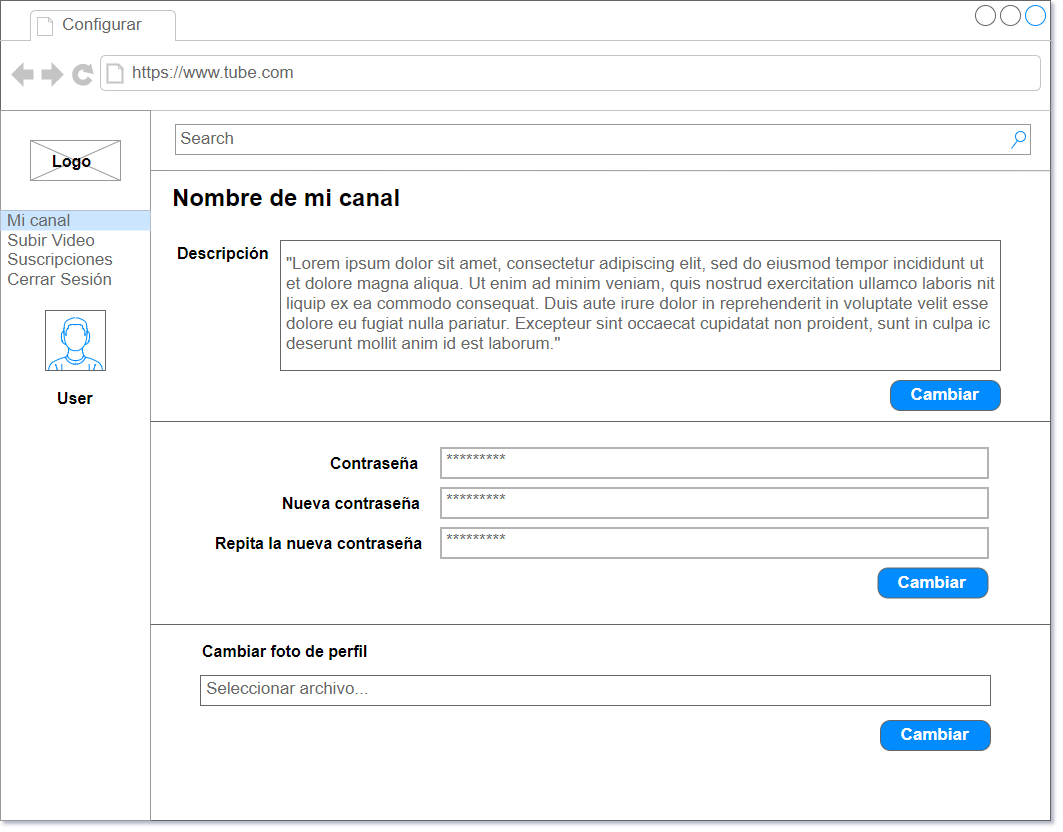
## Subir Vídeo



## Subir Miniatura

## Mi Canal

## Administrar Perfil



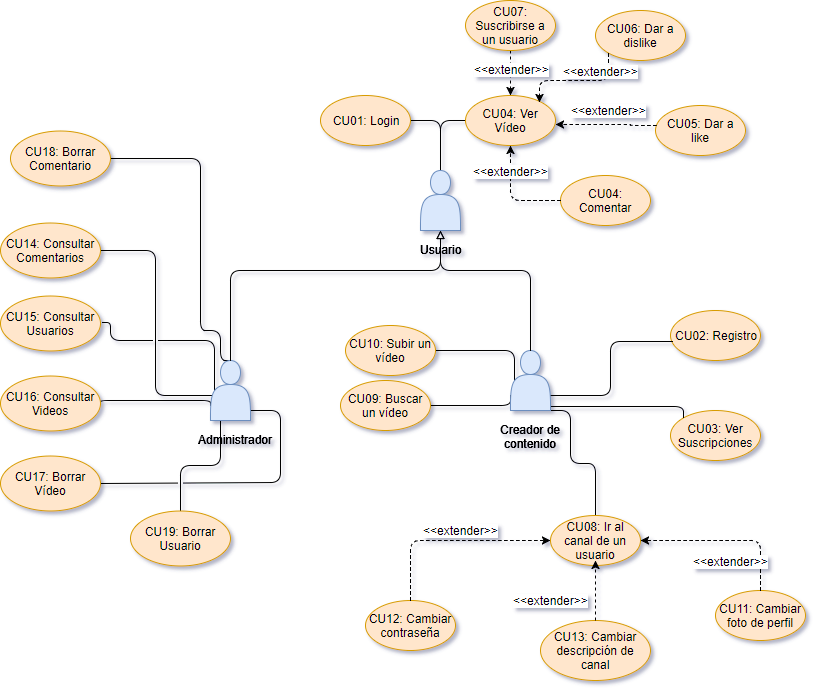
## Esquema de navegación

Para facilitar la lectura del esquema, es importante mencionar:

* Subir video, Suscripciones, Mi canal y Buscador son accesibles desde cualquier vista de usuario, excepto Login/Registro y Cerrar Sesión.
* Administrar Canal solamente es accesible desde Mi canal.
* Cerrar Sesión es accesible desde cualquier vista excepto Login/Registro.
* Gestión de usuarios, vídeos y comentarios es una vista exclusiva de administradores.

## **DISEÑO DE LA APLICACIÓN**

## Diagrama de casos de uso

Los casos de uso contemplados de la aplicación se recogen en el siguiente diagrama:

## Componentes

Los componentes de la aplicación pueden dividirse en dos grupos según el patrón Modelo-Vista-Controlador utilizado en el desarrollo del proyecto:

* **Componentes web:** Corresponde a los elementos que interactúan con el usuario en el navegador y que se implementarán mediante UIs de Vaadin. Siguiendo la metodología utilizada en Vaadin, todas las vistas estarán en una única página que irá invocando a los distintos elementos según se vayan pulsando botones, además la navegación se llevará a cabo mediante una barra de menú lateral y superior que estarán presentes en todo momento que mostrará al usuario todas las opciones disponibles.
* **Componentes de la lógica del negocio:** Implementarán la funcionalidad de la aplicación. Los elementos controladores se encargarán de la comunicación de las vistas con la capa de acceso a datos mediante el uso de clases que representan los documentos en la base de datos y una clase específica para poder manipular la base de datos.

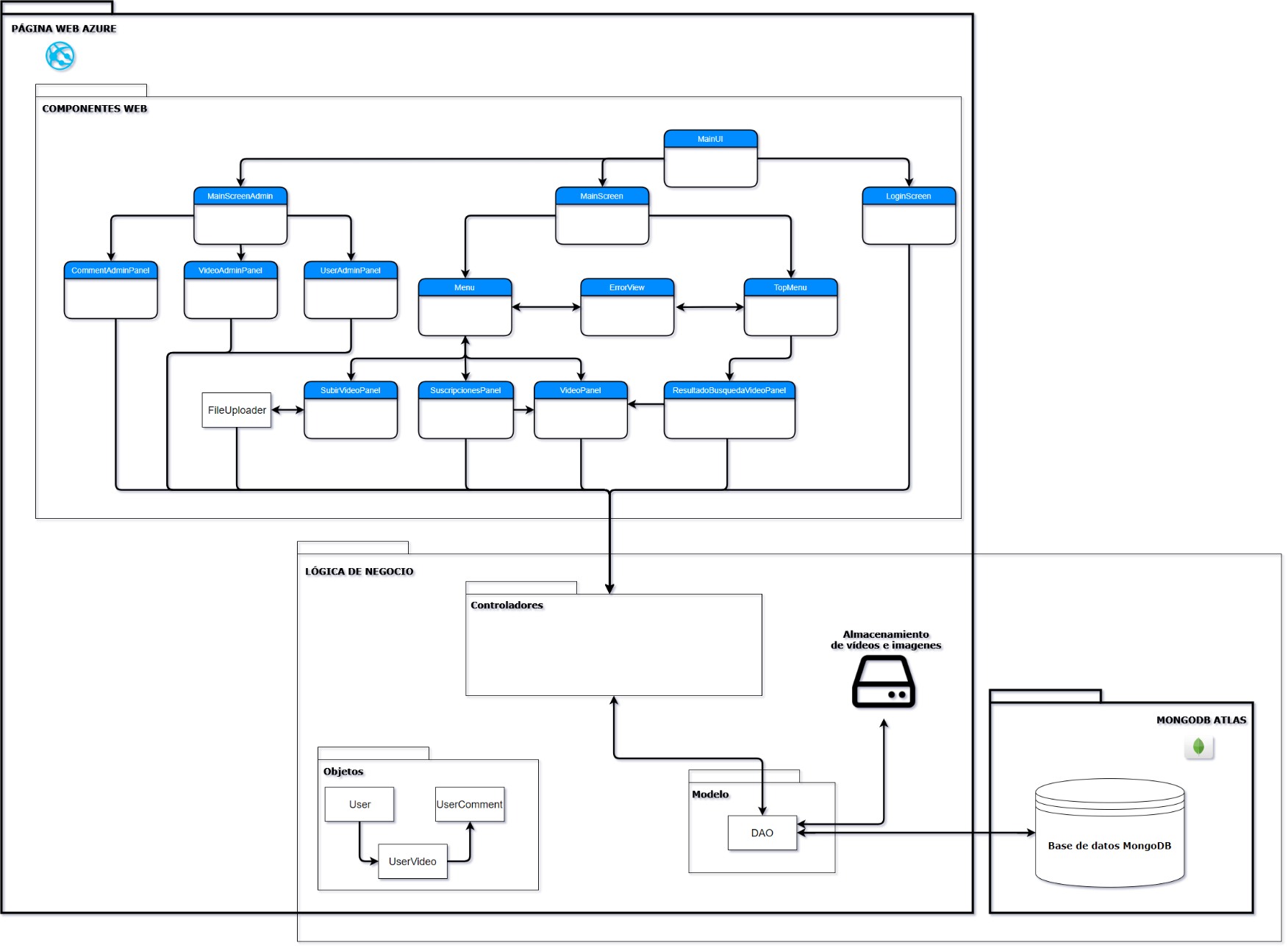
La descripción de los componentes web es la siguiente:

|  |  |
| --- | --- |
| **COMPONENTE** | **FUNCIONALIDAD** |
| MainUI | Página única que contiene todas las vistas Vaadin |
| LoginScreen | Vista para que los usuarios se identifiquen y registren |
| MainScreen | Vista que contiene las funcionalidades principales de la aplicación para los usuarios. |
| MainScreenAdmin | Vista que contiene las funcionalidades principales de la aplicación para los administradores. |
| Menu | Vista que contiene la barra de navegación lateral. Desde aquí se accede a la mayoría de las funcionalidades de la aplicación |
| TopMenu | Vista que contiene la barra de navegación superior encargada de la búsqueda de vídeos |
| ErrorView | Vista que muestra un mensaje de error si fuera necesario |
| SubirVideosPanel | Vista que contiene las funcionalidades relativas a subir vídeos y miniaturas de vídeos |
| VideoPanel | Vista que contiene las funcionalidades relativas a ver vídeo y su interacción con el mismo |
| ResultadoBusquedaVideosPanel | Vista que contiene los resultados de la búsqueda de vídeos por su título |
| SuscripcionesPanel | Vista principal de bienvenida a la aplicación para los usuarios registrados |
| CommentAdminPanel | Vista que contiene todos los comentarios publicados por los usuarios y desde donde se gestionan. |
| UserAdminPanel | Vista que contiene todos los usuarios de la plataforma y desde donde se gestionan. |
| VideoAdminPanel | Vista que contiene todos los vídeos de la plataforma y desde donde se gestionan. |
| FileUploader | Clase que creará un formulario para la subida de archivos y comprobará que los archivos subidos cumplen con las restricciones impuestas |

El desglose de componentes de lógica de negocio es el siguiente:

|  |  |  |
| --- | --- | --- |
| **PAQUETE** | **COMPONENTE** | **FUNCIONALIDAD** |
| Controllers | Controladores | Se encargan de comunicar la vista con el acceso a datos, así como de realizar las operaciones sobre los datos que sean necesarias. Será necesario un controlador por cada tipo de documento en la base de datos |
| Objects | User | Clase que representa un documento usuario de la base de datos con el que representar, en Java, la información que este contiene |
| UserComment | Clase que representa un documento comentario de base de datos con el que representar, en Java, la información que este contiene |
| UserVideo | Clase que representa un documento vídeo de la base de datos con el que representar, en Java, la información que este contiene |
| Model | DAO | Se encarga de realizar el acceso a datos, tomando parámetros desde el controlador y devolviendo instancias de los documentos |
| Base de datos | Base de datos | Base de datos MongoDB que se encarga de la persistencia de la información de la aplicación |
| Almacenamiento Físico | Almacenamiento | Almacenamiento que se encargará de guardar los archivos de vídeo e imágen que se suban a la plataforma |

Finalmente, la relación entre los distintos componentes se representa en el siguiente diagrama:



Como se puede ver la aplicación estará dividida en 2 servicios, la propia plataforma, compuesta por los componentes web y parte de la lógica, estará hosteada en Microsoft Azure y la base de datos en MongoDB Atlas.

También se puede observar que los componentes web solamente tienen acceso a los controladores de la aplicación, de la misma forma que el controlador sirve como intermediario entre las vistas y la clase DAO encargada de la comunicación con la base de datos localizada en MongoDB Atlas.

Cabe destacar que la clase FileUploader está localizada en los componentes web ya que, si existe algún fallo a la hora de la subida, debido a que la clase tiene que tener un método específico para gestionar el archivo subido, resulta más fácil ver rápidamente si el fichero es correcto en la vista y si estuviese localizado en el modelo, resultaría extraño que el modelo muestre mensajes de error al usuario.

## Clases de la aplicación

Siguiendo el patrón modelo-vista-controlador, se ha planteado el esquema anterior como una aplicación Java, y se han implementado las siguientes clases divididas por paquetes según su funcionalidad:

|  |  |  |
| --- | --- | --- |
| **PAQUETE** | **CLASE** | **FUNCIONALIDAD** |
| Objects | User | Implementan los objetos extraídos de la base de datos. Todas estas clases tienen métodos getter y setter pero no implementan ninguna lógica de negocio. |
| UserVideo |
| UserComment |
| FileUploader | Encargado de crear los formularios para subir archivos y comprobar que los archivos subidos son correctos |
| Controllers | GeneralController | Implementa la lógica de negocio que no está relacionada con ninguno de los demás controladores |
| SuscripcionesController | Implementa la lógica de negocio relacionada con las suscripciones |
| UsuarioController | Implementa la lógica de negocio relacionada con los usuarios |
| VideoController | Implementa la lógica de negocio relacionada con los vídeos |
| Modelo | DAO | Implementa la capa de acceso a datos mediante conexiones a una base de datos MongoDB. Sus métodos realizan consultas mediante parámetros y devuelven distintos tipos de parámetros |
| Views | ErrorView | Implementa la vista de error |
| LoginScreen | Implementa la pantalla de login y registro |
| Menu | Implementa el menú lateral |
| MainScreenAdmin | Rellena la pantalla del administrador con distintas opciones |
| MainScreen | Rellena la pantalla del usuario con distintos elementos según se pulsen botones |
| MainUI | Implementa la vista principal que arranca la aplicación y llama a las demás vistas |
| View.Panels | ResultadosBusquedaVideosPanel | Genera el panel que muestra los resultados de buscan vídeos por su título |
| SubirVideoPanel | Genera el panel donde se gestiona la subida de vídeos a la plataforma |
| SuscripcionalesPanel | Genera el panel donde se ven los vídeos subidos de las suscripciones del usuario |
| VideoPanel | Genera el panel donde un se ve y se interactúa con un vídeo |
| UserAdminPanel | Genera un panel con una tabla donde se gestionar todos los usuarios de la plataforma |
| VideoAdminPanel | Genera un panel con una tabla donde se gestionar todos los vídeos subidos a la plataforma |
| CommentAdminPanel | Genera un panel con una tabla donde se gestionar todos los comentarios de la plataforma |

## Estructura física de la base de datos

La base de datos tiene la siguiente estructura física:

{

**\_id:** ObjectId(“5cc4651c1c9d440000ae1589”),

**username:** “usuario”,

**password:** “contraseña”,

**type**: “user”,

**email:** “example@mail.com”,

**suscriptores:** 1,

**suscripciones:** [

“usuario2”,

“usuario3”

],

**videos:** [

{

**title:** “titulo”,

**date:** 2019-05-05T18:20:22.829+00:00,

**likes:** [

“usuario2”

],

**dislikes:** [

“usuario3”

],

**views:** 100

**comments:** [

{

**date:** 2019-04-28T16:00:00.000+00:00,

**username:** “usuario2”,

**comment:** “esto es un comentario”

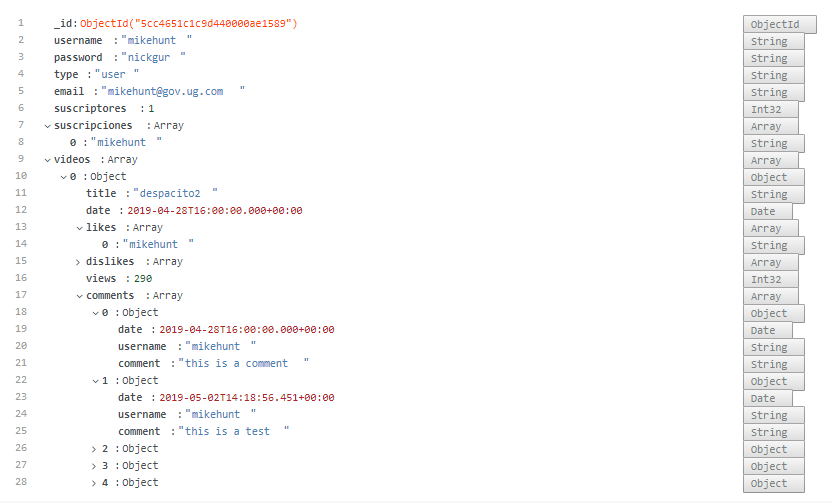
}

]

}

]

}

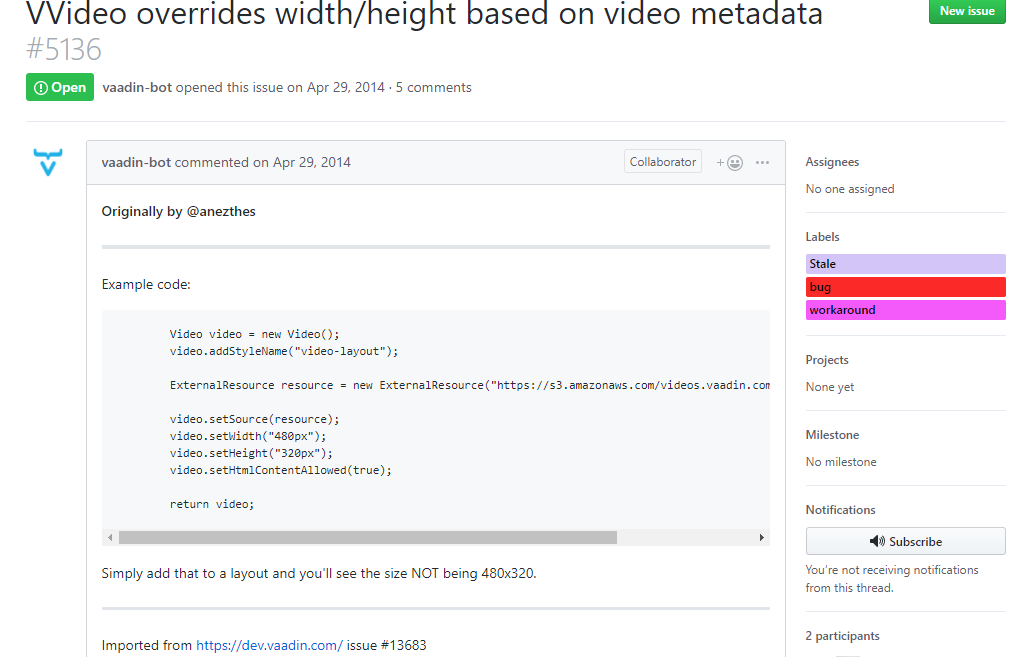
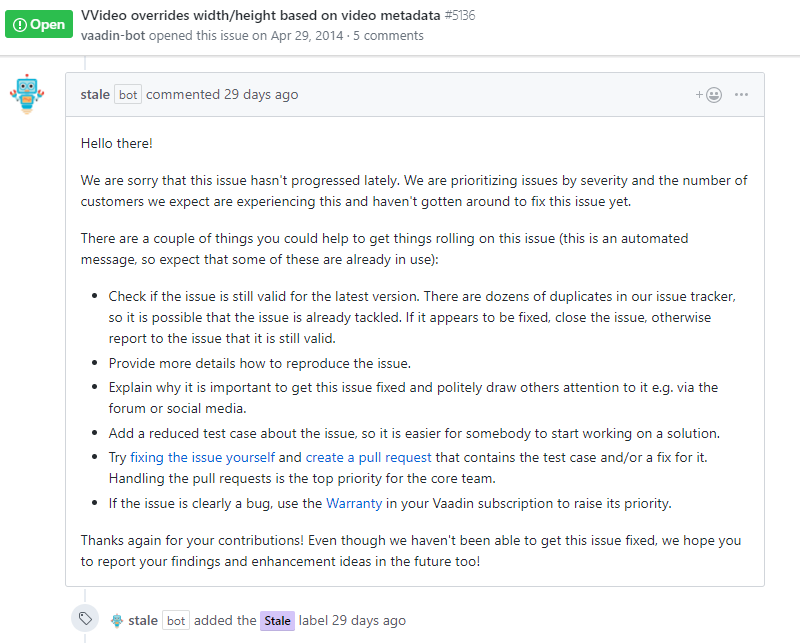
Finalmente, un ejemplo de un documento en la base de datos es el siguiente:

# **IMPLEMENTACIÓN**

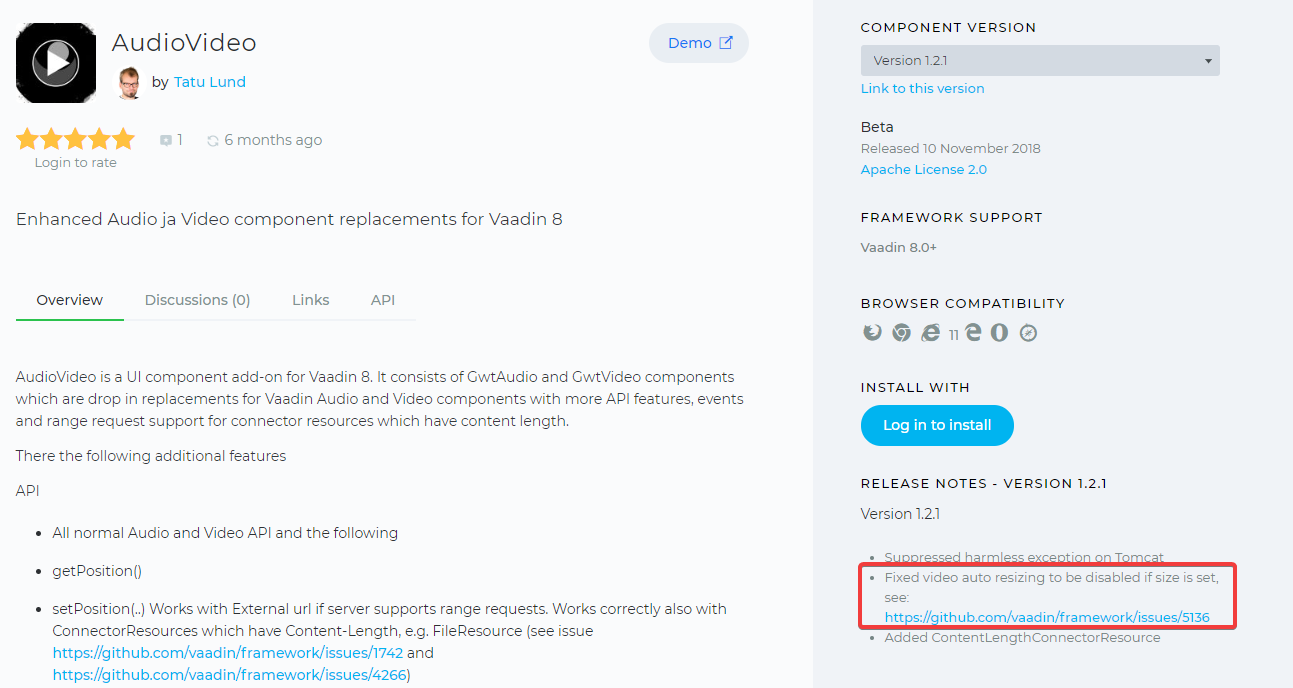
## Detalles de la implementación

## Problemas con el componente Video

Durante el desarrollo del proyecto, se observó que los vídeos se redimensionaban a su tamaño original, incluso cuando se especificaba su tamaño en el código. Investigando esto, se encontró la siguiente issue en Github que se puede acceder [haciendo click aquí:](https://github.com/vaadin/framework/issues/5136)



Como se puede observar, el fallo encontrado en el componente de vídeo ha estado presente en Vaadin desde 2014 y según el último comentario en la issue, no parece que se resuelva pronto. Para hacer frente a este problema, se ha investigado un poco más y se ha encontrado un componente de Vaadin llamado [AudioVideo](https://vaadin.com/directory/component/audiovideo) creado por un usuario que resuelve este problema:

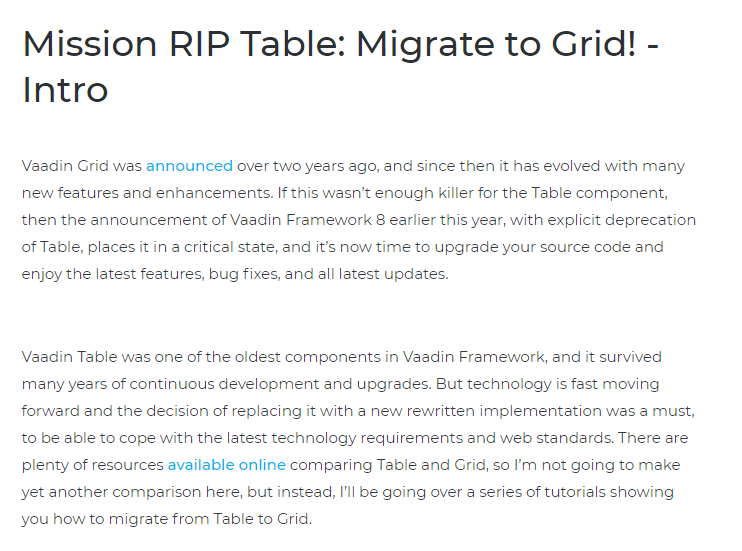


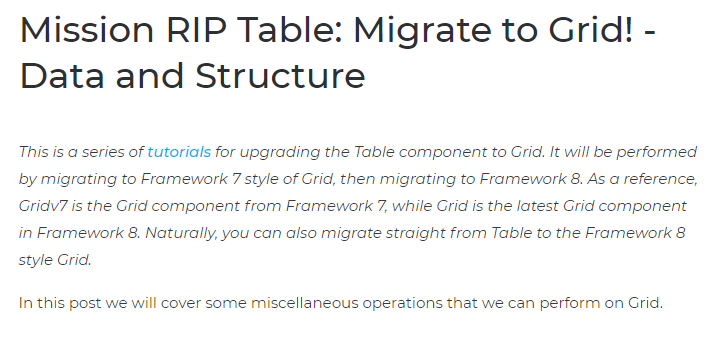
Gracias a este componente, se ha podido arreglar correctamente el fallo en el componente de vídeo en Vaadin.

## Actualización a una versión de Vaadin superior

Debido que para seguir el desarrollo del proyecto se ha tenido que utilizar el componente AudioVideo, hemos tenido que actualizar la versión de Vaadin del proyecto a una soportada por el componente, ya que requiere de Vaadin 8.0 o superior. Concretamente se ha tenido que utilizar **Vaadin 8.7.2** y debido a este cambio de versión, se han tenido que modificar algunas partes del código ya que estaba basado en la teoría de la asignatura que esta usa Vaadin 7.7.7 y daba algunos errores menores, tales como incorrectas importaciones de clases o de constructores de clases.

## Ausencia del componente Table en Vaadin 8+

Finalmente, a partir de Vaadin 8.0 se quitó el componente Table de Vaadin tal y como se explica en [la siguiente publicación en el blog de Vaadin](https://vaadin.com/blog/mission-rip-table-migrate-to-grid-data-and-structure):



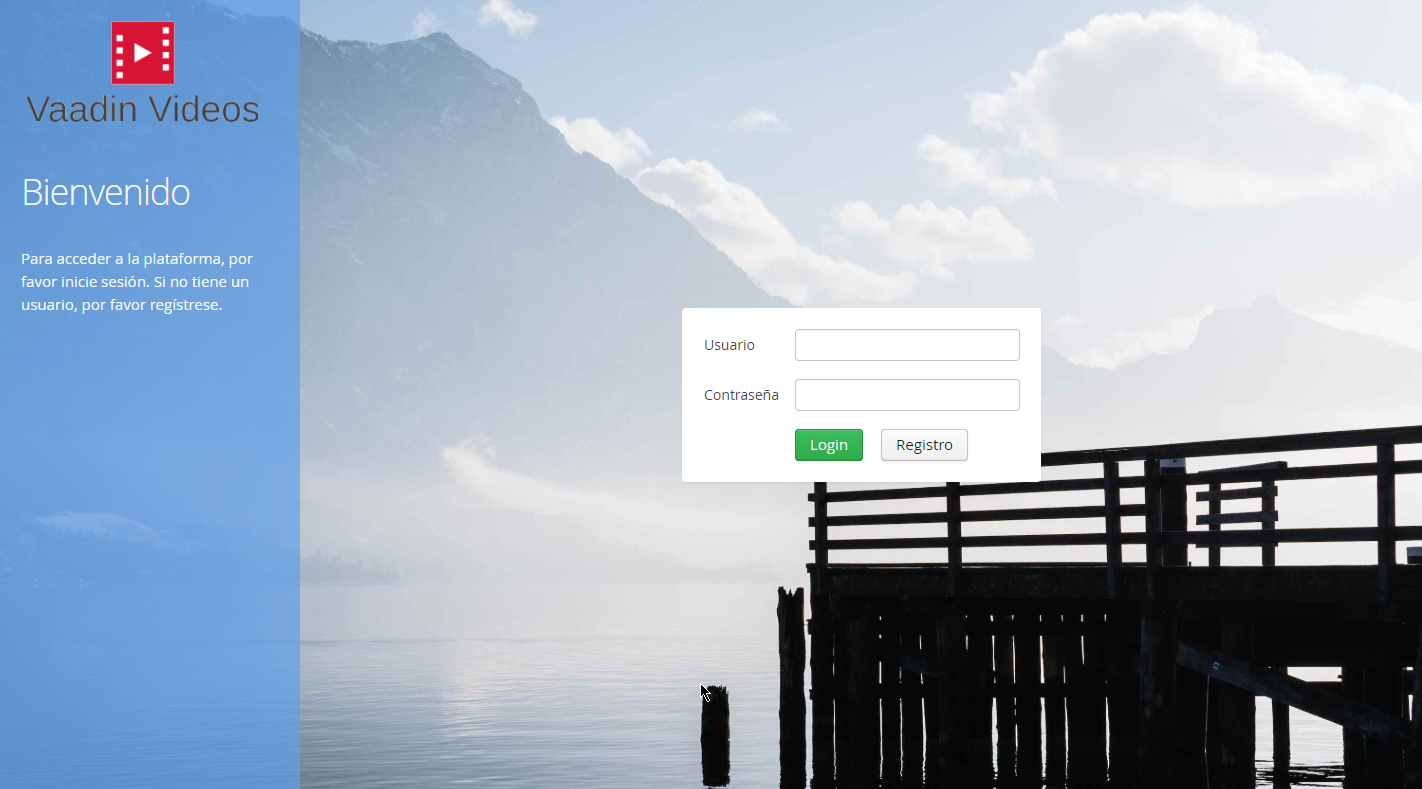
Así que debido a esto, los componentes del grupos tuvimos que aprender y estudiar cómo utilizar el componente Grid como una tabla según esta publicación y la documentación del componente Grid en Vaadin 8.0

Una vez solucionados estos problemas, se ha continuado con el desarrollo del proyecto.

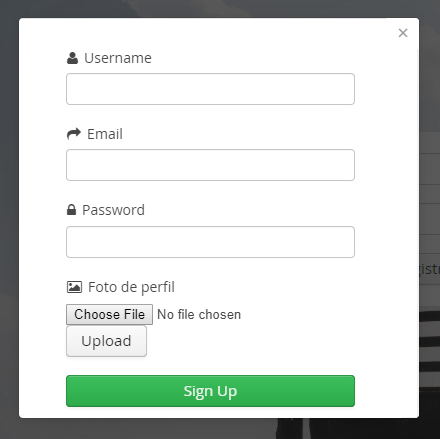
# **MANUAL DE USUARIO**

## Inicio de sesión y registro

Cuando accedemos a la página principal de la plataforma localizada en <https://proyectotad2019grupo1.azurewebsites.net>, nos encontramos con el siguiente contenido:



Si queremos acceder a la plataforma y estamos registrados, solamente tenemos que introducir nuestros datos en los campos “Usuario” y “Contraseña”. Si por el contrario no estamos registrados, hacemos click en el botón “Registro” y rellenamos los siguientes campos:



**OJO:** Solamente se aceptan imágenes en formato .png

## Suscripciones

## Ver vídeo

## Subir vídeo

## Buscar un vídeo

## Ir a un canal

## Administrar canal

## Cerrar sesión