

# MEMORIA DEL PROYECTO

## PLATAFORMA PARA LA DIFUSIÓN DE CONTENIDOS AUDIOVISUALES "MY-TV"

Trabajo de Fin de Grado  
INGENIERÍA INFORMÁTICA



Julio de 2021

### AUTOR

Sergio Gómez de la Torre

### TUTORES

Juan Francisco De Paz Santana

André Filipe Sales Mendes

Gabriel Villarrubia González



## Certificado de los tutores

D. Juan Francisco de Paz Santana, D. André Filipe Sales Mendes y D. Gabriel Villarrubia González, tutores del Departamento de informática y Automática de la Universidad de Salamanca (USAL).

HACEN CONSTAR:

Que el trabajo titulado "PLATAFORMA PARA LA DIFUSIÓN DE CONTENIDOS AUDIOVISUALES" , ha sido realizado por D. Sergio Gómez de la Torre, con el número de documento 70926310-Z y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado para la Titulación Grado en Ingeniería informática de la Universidad de Salamanca.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 5 de julio de 2021.

D. Juan Francisco de  
Paz Santana

D. André Filipe Sales  
Mendes

D. Gabriel Villarrubia  
González



## Resumen

Actualmente, la mayor parte del contenido audiovisual que conlleva una producción compleja, como son los contenidos de la mayoría de las grandes cadenas televisivas, se transmiten mayoritariamente de forma única a través de la televisión. Sin embargo, está claro, que hoy en día y cada vez más en el futuro, gran parte del contenido será consumido a través de internet.

La idea principal, por tanto, es adaptar esos contenidos que actualmente se emiten en la televisión, a internet, de forma que cualquiera desde un dispositivo móvil o fijo, que cuente solo con conexión WI-FI, pueda consumir ese contenido, sin necesidad de acceder a una toma de antena de televisión convencional o cable.

Teniendo en cuenta, que de aquí a poco tiempo, las cadenas televisivas, tendrán esta necesidad de desplazarse a internet para poder competir con las formas de entretenimiento que están surgiendo en este, y que son cada vez más sofisticadas, como YouTube o Twitch. Esta idea, permitiría que las cadenas lo hagan de una forma colectiva, y de esta manera, todo el contenido se pueda seguir disfrutando en único medio, y no en webs separadas como lo hacen algunos canales, que cuelgan su contenido un tiempo después de emitir. Así, los consumidores finales, pueden despreocuparse de tener que buscar en muchos sitios distintos, además de que podrán recibir notificaciones de las emisiones o recomendaciones de qué ver. También se podrían implementar métodos de seguimiento, que no están del todo desarrollados en la TV convencional pero sí en las redes sociales de internet.

Además de esto, si existiese este "sistema común" para la televisión en internet, adicionalmente a las grandes cadenas que ya existen, también se abriría la posibilidad de que usuarios independientes con menos recursos, puedan comenzar un proyecto en un medio donde los más conocidos generarán audiencia y tráfico, de forma que todos puedan beneficiarse.

Es por esto, que en este trabajo de fin de grado, se pretende desarrollar una plataforma digital, que, en definitiva, se pueda utilizar para la difusión de contenidos audiovisuales tanto en directo como en diferido. Es decir, es una página web en la que canales digitales transmiten su contenido, ya sea pregrabado o en directo, y este, se emite en tiempo real a través de la red.

La diferencia principal con la televisión es que se emite a través de internet, y que, además, esas transmisiones y contenidos adicionales pueden quedar almacenados en la plataforma para su posterior visualización, en forma de contenido a la carta.

Adicionalmente a la televisión, también se generará una interacción con los usuarios a través de los comentarios y de los "me gustas", lo que se pretende que sea una mejora la experiencia general. Así como también se añaden comodidades adicionales como sistema de seguimiento, recomendaciones, filtros de búsqueda, categorías, y un sistema de gestión de contenidos bastante completo, tanto para el usuario como para los administradores de la página.

Para realizar este proyecto, se ha utilizado una adaptación al proyecto del proceso unificado, así como la trabajada metodología Durán y Bernárdez para los requisitos que aparecen en algunos documentos y que son fundamentales para el proyecto.

**Palabras clave:** Televisión en internet, transmisión en directo, videos a la carta, usuarios, canales, administración, seguimiento, me gusta.

## Summary

Currently, most of the audiovisual content that involves a complex production, such as the content of most of the large television networks, is transmitted mostly in a unique way through television. However, it is clear that today and increasingly in the future, a large part of the content will be consumed through the internet.

The main idea, therefore, is to adapt those contents, which are currently being broadcasted on television, to the internet, so that anyone from a mobile or fixed device, who has only a WI-FI connection, can consume that content, without need to access a conventional television cable.

It is important to understand that, in a short time, television networks will have the need to move to the internet to compete with the new forms of entertainment that are emerging there, and that are increasingly sophisticated, such as YouTube or Twitch. This idea would allow the TV channels to do it in a collective way, and in this way, all the content can be enjoyed in a single medium, and not on separate websites as some channels do, which post their content days after broadcasting it. In this way, end consumers can forget about having to search on many different sites, in addition to receiving notifications of the broadcasts, as well as recommendations of what to watch. Monitoring methods could also be implemented, which are not fully developed in conventional TV, but in internet social networks.

With a common system like the one that is proposed, in addition to the large TV channels that already exist, it would also open the possibility that independent people with fewer resources can start a project in a medium where the best known will generate audience and traffic, so that everyone will be able to benefit.

That is why, this final degree project, is intended to develop a digital platform, which, ultimately, can be used for the broadcasting of audiovisual content both live and delayed. In other words, it is a web page in which digital channels transmit their content, either pre-recorded or live, and this is broadcast in real time through the network.

In addition to television, an interaction with users will also be generated through comments and "likes", which is intended to improve the overall experience. Also, additional amenities are added such as a tracking system, recommendations, search filters, categories, as well as a complete content management system, both for the user and for the administrators of the page.

To carry out this project, an adaptation to the project of the unified process has been used, as well as worked methodologies to obtain requirements.

**Keywords:** Internet television, live broadcast, video on demand, users, channels, administration, follow-up, like.



## Tabla de contenido

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>3</b>
2.1. Objetivos del sistema	3
2.2. Objetivos personales	4
<b>3. Estado del arte y viabilidad del proyecto</b>	<b>5</b>
3.1. Otras plataformas similares	5
3.1.1. Video en diferido gratuito	5
3.1.2. Video en diferido y directo de pago	6
3.1.3. Transmisión en directo	7
3.1.4. Conclusión	8
3.2. Viabilidad del proyecto	8
<b>4. Conceptos teóricos</b>	<b>9</b>
4.1. Transmisiones en vivo o directo	9
4.2. Transmisión de video diferida	9
<b>5. Técnicas y herramientas</b>	<b>11</b>
5.1. Aplicación web	11
5.1.1. Estándar web	11
5.1.2. Vue.js	11
5.1.3. JavaScript	12
5.1.4. Vuetify	12
5.1.5. Axios	12
5.1.6. Firebase	12
5.1.7. HTML	13
5.1.8. CSS	14
5.2. Servidor de streaming	14
5.2.1. Node.js	14
5.2.2. NPM	14
5.2.3. Nginx y RTMP	14
5.2.4. PM2	14
5.3. Herramientas de documentación y desarrollo	15
5.3.1. Draw.io	15
5.3.2. Microsoft Project	15
5.3.3. Visual studio code	15
5.3.4. EZEstimate	15



5.3.5. JSDoc.....	16
5.3.6. Vue Styleguidist .....	16
5.3.7. Adobe XD .....	16
5.3.8. Google Drive.....	16
5.4. Herramientas de prueba.....	16
5.4.1. OBS .....	16
5.4.2. VirtualBox.....	16
<b>6. Aspectos relevantes del desarrollo .....</b>	<b>17</b>
6.1. Metodología utilizada.....	17
6.2. Estimación de esfuerzo.....	18
6.3. Planificación temporal .....	19
6.4. Especificaciones de requisitos del software .....	21
6.4.1. Participantes.....	21
6.4.2. Objetivos .....	21
6.4.3. Requisitos de información .....	22
6.4.4. Requisitos Funcionales.....	22
6.4.5. Requisitos no Funcionales.....	23
6.4.6. Matriz de rastreabilidad.....	23
6.5. Análisis.....	23
6.5.1. Modelo del dominio .....	23
6.5.2. Paquetes y clases de análisis.....	25
6.5.3. Realización de los casos de uso, modelo de análisis .....	25
6.6. Diseño.....	25
6.6.1. Modelo de diseño.....	26
6.6.1.1. Patrones arquitectónicos y de diseño.....	26
6.6.1.2. Subsistemas de diseño .....	27
6.6.1.3. Clases de diseño.....	27
6.6.1.3.1. Aplicación web .....	27
6.6.1.3.2. Backend CDN.....	32
6.6.1.4. Vista arquitectónica .....	32
6.6.1.5. Realización de los casos de uso modelo de diseño.....	34
6.6.2. Diseño de la base de datos .....	34
6.6.3. Modelo de despliegue .....	35
6.7. Implementación .....	37
6.7.1. Aplicación web .....	37

6.7.2. Backend CDN .....	40
6.7.3. Servidor de video.....	40
6.8. Pruebas .....	41
6.9. Funcionalidades .....	42
6.9.1. Usuario anónimo .....	42
6.9.1.1. Barra de opciones anónimo .....	42
6.9.1.2. Autenticación de usuarios.....	42
6.9.1.3. Páginas de la aplicación anónimo.....	44
6.9.1.4. Menú lateral anónimo .....	48
6.9.2. Usuario autenticado .....	48
6.9.2.1. Barra de aplicación autenticado .....	48
6.9.2.2. Opciones de usuario autenticado .....	50
6.9.2.3. Seguimiento y me gusta .....	50
6.9.2.4. Menú lateral autenticado.....	51
6.9.3. Usuario canal.....	51
6.9.3.1. Barra aplicación canal .....	51
6.9.3.2. Configuración de canal.....	51
6.9.4. Usuario Administrador.....	55
6.9.4.1. Barra de aplicación administrador.....	55
6.9.4.2. Administración.....	55
6.9.4.3. Administrar comentarios.....	59
<b>7. Conclusiones.....</b>	<b>60</b>
<b>8. Dificultades y líneas futuras.....</b>	<b>62</b>
8.1. Problemas y soluciones .....	62
8.2. Mejoras para el futuro.....	63
<b>9. Referencias.....</b>	<b>64</b>

## Lista de figuras

Figura 1: YouTube	5
Figura 2: Cuatro	6
Figura 3: Prime Video	7
Figura 4: Twitch	7
Figura 5: Live Streaming	9
Figura 6: Video Streaming	10
Figura 7: Fases de un Ciclo	17
Figura 8: Proceso unificado	18
Figura 9: Cálculo y resultados de la estimación	19
Figura 10: Estructura	20
Figura 11: Horas finales	21
Figura 12: Diagrama de casos de uso en el paquete Gestión de Directos	22
Figura 13: Modelo del dominio	24
Figura 14: Diagrama de secuencia de caso de uso Editar Directo UC-020	25
Figura 15: Patrón MVVM	26
Figura 16: Diseño Subsistemas	27
Figura 17: Vista	28
Figura 18: ModeloVista	29
Figura 19: Auth	30
Figura 20: Entidades	30
Figura 21: Modelo	31
Figura 22: Modelo Firebase	32
Figura 23: Backend CDN	32
Figura 24: Arquitectura	33
Figura 25: Diagrama de secuencia de caso de uso Registrarse UC-001	34
Figura 26: Firebase	35
Figura 27: Despliegue	36
Figura 28: Video Prototipo	38
Figura 29: Video	39
Figura 30: Collection Update	39
Figura 31: Auth	39
Figura 32: Axios	40
Figura 33: RTMP	41
Figura 34: Barra de aplicación Anónimo	42
Figura 35: Inicio de sesión	42
Figura 36: Registrarse	43
Figura 37: Correo Verificación	43
Figura 38: Recuperación de contraseña	43
Figura 39: Formulario contraseña	44
Figura 40: Descubrir	44
Figura 41: Categorías	45
Figura 42: Categoría	45
Figura 43: Página del Video 1	46
Figura 44: Página del vídeo 2	46

Figura 45: Página del directo 1	47
Figura 46: Página del directo 2	47
Figura 47: Menú lateral de navegación Anónimo	48
Figura 48: Barra de aplicación usuario Autenticado	48
Figura 49: Desplegable Usuario	49
Figura 50: Perfil 1	49
Figura 51: Perfil 2	50
Figura 52: Seguir	50
Figura 53: Dejar de seguir	50
Figura 54: Menú lateral Autenticado	51
Figura 55: Barra de aplicación canal	51
Figura 56: Configuración 1	52
Figura 57: Configuración 2	53
Figura 58: Configuración 3	53
Figura 59: Configuración 4	54
Figura 60: Configuración 5	54
Figura 61: Barra de aplicación desplegable administrador	55
Figura 62: Administración 1	55
Figura 63: Administración de directos	56
Figura 64: Administración vídeos	56
Figura 65: Administración de series	57
Figura 66: Administrar categoría	57
Figura 67: Administración solicitudes	58
Figura 68: Administración de usuarios	58
Figura 69: Suspensión usuarios	58
Figura 70: Cambiar rol	59
Figura 71: Administrar comentario	59

## 1. Introducción

En esta memoria, se documentan los aspectos más relevantes sobre el TFG llamado "Plataforma para la difusión de contenidos audiovisuales". Este Trabajo de fin de grado, se presenta en el grado en ingeniería informática de la USAL .

El proyecto consta de varios participantes, en primer lugar, el realizador, que es Sergio Gómez de la Torre, y por otro lado, 3 tutores de la organización USAL (**Universidad de Salamanca**) que se han encargado de dirigir al realizador en el proyecto, y que son, André Sales Mendes, Gabriel Villarrubia González y Juan Francisco de Paz Santana.

Actualmente, la mayor parte del contenido audiovisual que conlleva una producción compleja, como son los contenidos de la mayoría de las grandes cadenas televisivas, se transmiten mayoritariamente de forma única a través de la televisión. Sin embargo, está claro, que hoy en día y cada vez más en el futuro, gran parte del contenido será consumido a través de internet.

Por tanto, se quieren adaptar los contenidos que se emiten en la televisión, a internet, de forma que cualquiera desde un dispositivo móvil o fijo, que cuente solo con conexión WI-FI, pueda consumir ese contenido. Adicionalmente a las grandes cadenas que ya existen, también se abriría la posibilidad de que personas con menos recursos puedan transmitir.

En este trabajo de fin de grado, se pretende desarrollar una plataforma digital, que, en definitiva, se pueda utilizar para la difusión de contenidos audiovisuales tanto en directo como en diferido. Es decir, es una **página web en la que canales digitales transmiten su contenido**, ya sea grabado o en directo, y este, se emite en tiempo real a través de la red. Adicionalmente a la televisión, también se generará una respuesta de los usuarios a través de los comentarios y de los "me gustas". Además de esto se añaden comodidades como sistema de seguimiento, recomendaciones, filtros de búsqueda, categorías, así como un sistema de gestión de contenidos bastante completo, tanto para el usuario como para los administradores de la página.

Para realizar este documento, se pretende utilizar una estructura que recoja los principales elementos del trabajo, que es la siguiente:

- Objetivos**
- Estado del arte y viabilidad del proyecto**
- Conceptos teóricos**
- Técnicas y herramientas**
- Aspectos relevantes del desarrollo**
  - Metodología**
  - Especificaciones de requisitos**
  - Análisis**
  - Diseño**
  - Implementación**
  - Pruebas**
- Conclusiones**
- Dificultades y líneas futuras**

Es importante, comentar que para presentar este trabajo de fin de grado, se ha incluido el código de los distintos apartados de la aplicación, así como posibles configuraciones de elementos

fundamentales. También se aportan todos los documentos técnicos que se han realizado durante el desarrollo, a parte de este. Estos documentos son los siguientes:

**Anexo I:** Se trata del plan de proyecto software. En él, se muestran los detalles acerca de la planificación del proyecto software, de modo que podamos comprender mejor qué es lo que viene por delante, mediante estimaciones de esfuerzo y planificaciones temporales.

**Anexo II:** Se trata de la especificación de requisitos del software. En este, se documenta la especificación de requisitos del software del sistema que se desarrolla en este trabajo de fin de grado, tanto funcionales como no funcionales.

**Anexo III:** Es el análisis del sistema software. Este se centra en la creación de una estructura más precisa para el sistema, de forma que se pueda dar un paso más, refinando y documentando los requisitos del software.

**Anexo IV:** Es el documento acerca del diseño del sistema software. En este, nos centramos en la descripción de la realización física de los casos de uso en el dominio de la solución en base al análisis de requisitos del dominio del problema.

**Anexo V:** Se trata de la documentación técnica del proyecto. En este documento, se pretende hacer comprensible el código y las configuraciones entregadas como parte fundamental de la práctica, de cara a cualquier persona que deba usarlo.

**Anexo VI:** Este es el manual de usuario. Este, pretende hacer comprensible el programa de cara a cualquier persona que deba o quiera utilizar, es decir, al usuario final, mediante el uso de imágenes y explicaciones detalladas.

## 2. Objetivos

En este apartado, se presentarán los que son los principales objetivos del proyecto, y por lo tanto del sistema propuesto. Para ello se muestran tanto los objetivos que se quieren cumplir para el sistema, como los objetivos que se consideran personales.

Con todo esto, se podría decir que el objetivo prioritario del proyecto es desarrollar una plataforma digital, que se pueda utilizar para la difusión de contenidos audiovisuales tanto en directo como en diferido. Es decir, es una aplicación web en la que canales digitales transmiten su contenido, ya sea grabado o en directo, y este, se emite en tiempo real a través de la red.

### 2.1. Objetivos del sistema

En base a lo anteriormente comentado, se pueden presentar una serie de objetivos que son los que se han de cumplir para completar el sistema:

- **Gestión de usuarios:** Hay que poder gestionar las cuentas de los diferentes usuarios, lo que permitirá controlarlos, así como gestionar sus datos de una forma eficiente. Esto se hará mediante el uso de roles y sistemas de administración
- **Gestión de Directos:** Se deben de poder gestionar y controlar los directos de los usuarios canales de la plataforma, mediante su modificación, iniciación y finalización, ya sea por estos mismos o por los moderadores.
- **Gestión de Vídeos:** Se tienen que poder gestionar los vídeos subidos por los usuarios de tipo canal a la plataforma. Además de hacer que puedan ser consumidos por el resto de los usuarios.
- **Gestión de Series:** Se debe gestionar la creación, eliminación y modificación de las series que pertenecen a los usuarios de tipo canal en la plataforma. Pudiendo ser moderadas por los administradores.
- **Gestión de Categorías:** Se deben gestionar las categorías de las que dispondrá la plataforma para generar una mejor experiencia a la hora de encontrar contenidos. Estas categorías serán añadidas por los Administradores de la plataforma, en función de su necesidad.
- **Gestión de Eventos:** Se deben gestionar los eventos, que serán establecidos por los usuarios canales y que permitirán al resto de usuarios informarse acerca de la actividad de estos.
- **Gestión de Solicitudes:** Se deben gestionar las solicitudes procedentes de los usuarios no canales, para convertirse en canales y así poder transmitir. De modo que tendrán que existir opciones para solicitar, aceptar o rechazar.
- **Gestión de Comentarios:** Se ha de realizar la gestión de los comentarios, que serán dispuestos por los usuarios en los videos, y que tendrán que poder modificarse, ya sea por sus creadores o por los administradores.
- **Gestión de Seguimiento:** Se debe gestionar el sistema de seguimiento de los usuarios a los canales, lo que facilitará la tarea de encontrar de nuevo sus contenidos.
- **Gestión de Me gusta:** Se debe gestionar el sistema de me gustas en la plataforma, lo que requiere que un usuario pueda indicar que un contenido le gusta o le deja de gustar.
- **Gestión de Estadísticas:** Se deben manejar los datos de las actividades de los usuarios en la plataforma, para poder generar estadísticas simples sobre el flujo de acceso a los contenidos de los canales de la plataforma.

- **Gestión de Alertas:** Se debe de poder mostrar alertas de diferentes tipos a los usuarios en las ocasiones en la que se requiera.
- **Gestión de Notificaciones:** Debe ser posible generar notificaciones de diferentes tipos, que se enviarán a los usuarios en las ocasiones en las que sea necesario.

## 2.2. Objetivos personales

Después de los objetivos relacionados con el sistema, se pueden extraer algunos objetivos que son personales, y que son los principales motivos para la realización del proyecto.

En primer lugar, se podría decir que un gran objetivo personal es la **finalización del grado** en ingeniería informática, para el que es imprescindible la realización y entrega del TFG.

Por otro lado, el objetivo más importante, entre los personales, es, la puesta en común de todos los conocimientos adquiridos a lo largo de los años, para poder realizar un **proyecto completo**, desde la idea hasta la obtención de un producto funcional. De hecho, se trata de la primera vez en la que el alumno crea un proyecto en su totalidad, ya que previamente se habían realizado prácticas que comprendían distintos apartados, pero no se han puesto en común, lo que es, personalmente, un reto.

Otro objetivo relevante, y que se puede relacionar con el anterior, a nivel personal, es demostrar la **capacidad de análisis y de gestión** que se posee para el desarrollo de un proyecto en un entorno realista, con la presión que ello supone, así como la utilización de todas las herramientas necesarias, que pueden ser en algunos casos desconocidas.

Finalmente, hablando de estas herramientas, otro de los objetivos fundamentales en este ámbito, es observar la **capacidad de adaptación** a los diferentes lenguajes de programación y elementos que se han propuesto para el desarrollo, ya que muchos de estos no se han utilizado previamente. Por lo tanto, en lo personal, supone un importante esfuerzo que se afronta con gran interés.



### 3. Estado del arte y viabilidad del proyecto

En este apartado, se valora cuál es la situación general respecto a la tecnología que se pretende implementar, observando como referencia, a aquellos sistemas que son similares al propuesto y que ya existen. Posteriormente, se podrá valorar qué viabilidad tendrá la idea, así como extraer algunas conclusiones.

#### 3.1. Otras plataformas similares

En el ámbito de la transmisión de contenidos audiovisuales a través de internet, existen plataformas que han dominado el mercado en los últimos años. Se podría decir, que estas plataformas, son en parte resultado de la evolución de internet y el declive de los medios convencionales a causa de las nuevas necesidades de comunicación.

##### 3.1.1. Video en diferido gratuito

La plataforma que lleva más tiempo activa en este entorno, quizá es **YouTube** [1]. Esta plataforma surgió en el 2005 de la mano de una serie de creadores independientes de Estados Unidos, y posteriormente, fue adquirida por Google en 2006. El punto importante aquí, es que es el sitio web de este tipo más utilizado en internet. Funciona mediante una aplicación web basada en la tecnología de HTML5, que utiliza para sus reproductores en línea. La idea de la plataforma es principalmente, la retransmisión de contenido en diferido, de modo que los canales suben videos y los usuarios pueden verlos e interactuar con ellos a la carta. En los últimos años, la plataforma también se ha centrado en el aspecto del contenido en directo, sin embargo, es un tema que no ha desarrollado del todo y que otros de sus competidores tratan mejor, lo que provoca que los usuarios prefieran transmitir en directo en otras plataformas. Por último, hay que comentar que la plataforma es casi en su totalidad gratuita, y que los beneficios se obtienen de la publicidad del sitio. En la Figura 1, se puede ver un ejemplo de este tipo de contenido.

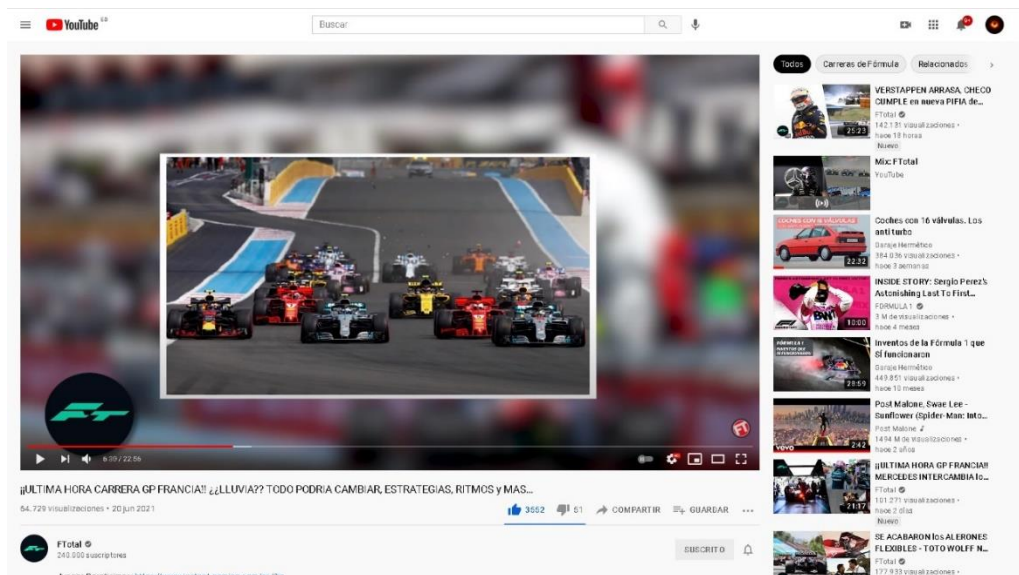


Figura 1: YouTube

Hay otras plataformas que han intentado seguir la línea de YouTube, y que han tenido bastante éxito, aunque no en una escala comparable. Sus cambios respecto a YouTube quizás están más centrados en el modelo de negocio, por ejemplo, **IGTV**, que está más centrada en el ámbito de las redes sociales y del entretenimiento rápido, o **Vimeo**, que es una plataforma libre de publicidad y con más personalización.

Por último, en cuanto a esto, me gustaría comentar también el hecho de que algunas cadenas de televisión tienen sus propias plataformas de contenido resumido a la carta, pero sin embargo no transmiten contenido en directo ya que eso lo dejan para la televisión. Estas plataformas normalmente utilizan su contenido para exponer publicidad en un modelo gratuito, pero no muy completo. Un ejemplo puede ser la página de **Cuatro**, en la Figura 2.



Figura 2: Cuatro

### 3.1.2. Video en diferido y directo de pago

Por otro lado, en cuanto a transmisión de contenido audiovisual en diferido, también han surgido en los últimos años, plataformas que se han dedicado a poner a disposición de los usuarios, contenidos relacionados con el cine y las series. Este tipo de plataformas, también tratan algunos aspectos parecidos a los de la que he presentado para este TFG, aunque llevadas quizá a un mercado distinto, ya que se trata en su totalidad de contenidos de pago, cuyo acceso requiere adquirir una suscripción mensual. Algunos ejemplos de este tipo de plataformas serían **Amazon Prime Video** [2], **Disney +** o **Netflix**. Igual sucede con plataformas híbridas entre videos y retransmisión en directo, pero destinadas a otros públicos, como por ejemplo **DAZN**, para ver deportes. En la Figura 3, se puede ver un ejemplo de este tipo de estructuras de pago.

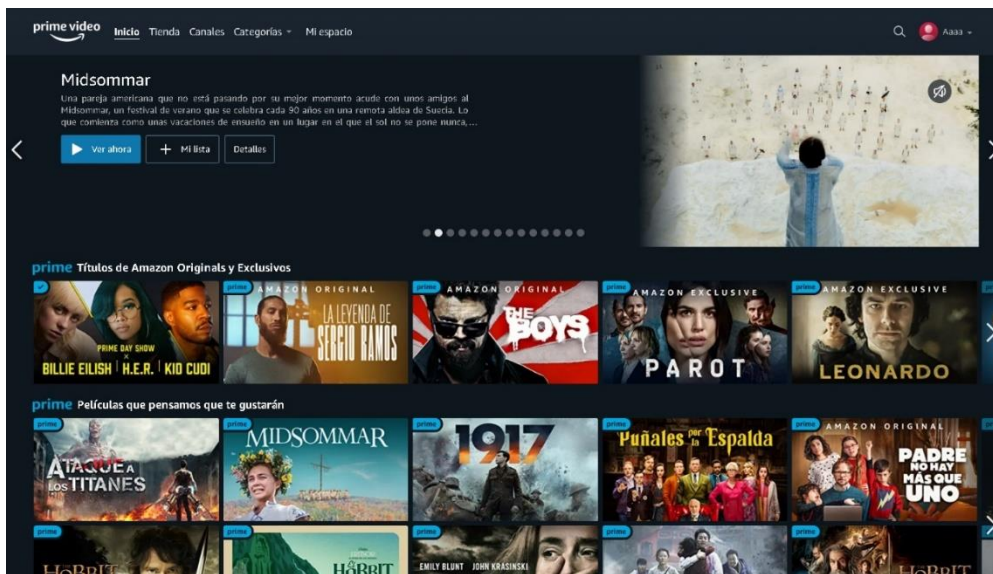


Figura 3: Prime Video

### 3.1.3. Transmisión en directo

Por otro lado, en cuanto al ámbito de la retransmisión en directo de forma exclusiva, existen múltiples plataformas que han intentado llevar a cabo este concepto, aunque quizá la más conocida es **Twitch** [3]. Se trata de una plataforma creada en 2011 y que desde 2014 es propiedad de Amazon. Es una plataforma destinada exclusivamente a la retransmisión de contenido en vivo, aunque es cierto que las transmisiones quedan grabadas en video para verlas, pero solo por un tiempo limitado. Esta plataforma ha logrado obtener audiencias muy elevadas, gracias a su modelo basado en la creación de comunidades, ya que los usuarios pueden interactuar con sus canales favoritos mediante donaciones, seguimientos o mediante un chat en vivo. Otro aspecto curioso, es que el ámbito de moderación normalmente se relega a los propios usuarios, que serán elegidos “moderadores” por los canales.

Esta plataforma, igual que YouTube, es gratuita, pero se puede pagar por suscripciones, donaciones y bits (La moneda del sitio). Se puede observar su estructura en la Figura 4.

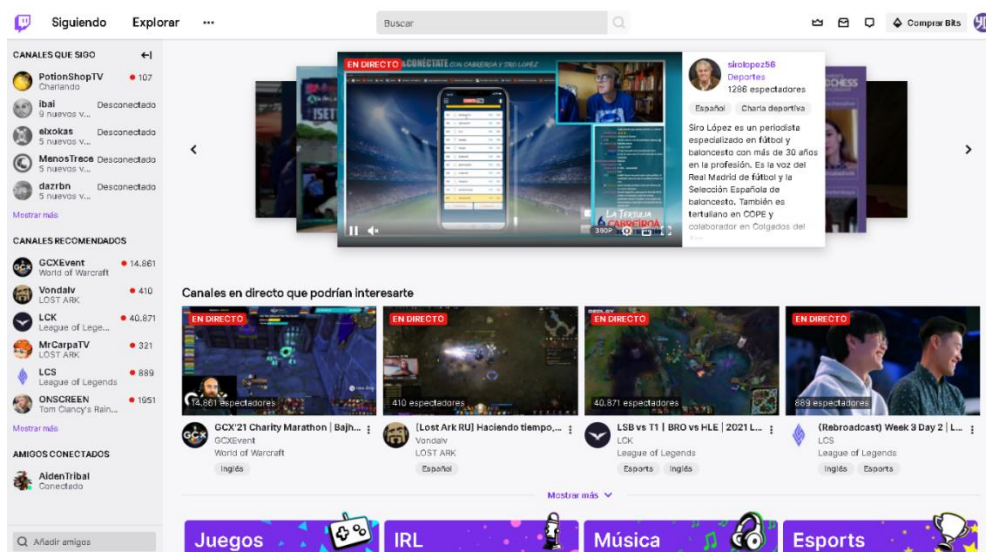


Figura 4: Twitch

### 3.1.4. Conclusión

En conclusión, se podría decir que existen múltiples plataformas con características similares a la propuesta en el TFG, pero sin embargo, ninguna de las que ha triunfado a gran escala propone un modelo de fusión entre la transmisión de contenido en directo y diferido. Además de que sus modelos están centrados en contenidos que tienen generalmente poca producción, mientras que esta plataforma está más pensada para hacer televisión.

### 3.2. Viabilidad del proyecto

Como se ha comentado anteriormente, la plataforma está centrada en la transmisión en vivo y en diferido de contenido audiovisual televisivo, respondiendo a las necesidades esenciales de modernización de los canales de televisión que actualmente transmiten a través de la antena, y que ahora más que nunca, están comprendiendo la necesidad de moverse al ámbito de internet, que es y cada vez será un mercado mayor, y que quizá las cadenas, no consiguen cubrir con sus páginas web en las que cuelgan solamente algunos contenidos. También, para aquellos que no se dedican actualmente a esto, es decir, los canales que pertenecen a particulares, se suplirá la necesidad de dar a conocer su contenido en internet. Por supuesto, la transmisión en internet en caso de canales se podría mantener en paralelo con la transmisión en televisión, aunque algunas cadenas podrán optar por cambiar de forma completa. Estas dos necesidades, pueden ser satisfechas por la plataforma, lo que podría ser determinante para la viabilidad del proyecto.

Adicionalmente, se pueden dar algunos datos que respaldan estas suposiciones. Por ejemplo, se puede ver que en España en el año 2020, el 93,2% de la población de 16 a 74 años ha utilizado Internet, lo que supone unos 32.8 millones de usuarios, 2.5 puntos más que en 2019, y la tendencia se mantiene al alza [4]. Por otro lado, en 2020 en España, aproximadamente el 93% de la población mayor de 64 años vio la televisión, mientras que en el caso de los individuos de entre 14 y 34 años se situó por debajo del 80%, lo que significa una tendencia a la baja respecto al 2019 [5]. Esto significa que la mayoría o totalidad de los canales de televisión se ven o se verán afectados por este problema.

En cuanto a la viabilidad del proyecto, también podemos observar el mercado en el que se movería este. Por ejemplo, en el mercado nacional de España, actualmente y hablando de internet, se estima que alrededor de 43 millones de españoles se conectan diariamente a Internet y que gastan de media más de 6 horas navegando [6], lo que supera por mucho al consumo de televisión.

Podríamos suponer que, si los usuarios de internet tuvieran un sistema común de acceso a los contenidos de la televisión, se generaría una captación importante de audiencia y del tiempo de expectación para los canales de televisión en internet, lo que se traducirá en beneficios para estos. Todos estos usuarios que consumen contenidos audiovisuales generan un enorme interés para las compañías, que desean publicitar sus productos en la televisión. Se calcula, si hablamos de publicidad en internet, que durante el 2020, los ingresos de publicidad en televisión online en España han sido de 45 millones de euros, y se estima que para 2024 será de unos 58 millones [7], lo que podría ser mucho mayor según algunas previsiones.

Sin duda, son aspectos muy interesantes y que podrían ser indicativos de la viabilidad de una plataforma como esta de cara al mercado, y a solucionar algunas necesidades existentes.

## 4. Conceptos teóricos

En este apartado, nos centramos en explicar algunos de los conceptos teóricos que han aparecido y aparecerán en este documento, para que se puedan entender sin necesidad de realizar una búsqueda externa. Sin embargo, en el caso de este proyecto, no existen demasiados conceptos complicados, ya que no hay una parte física en la aplicación ni tampoco una teoría complicada detrás de la misma, los elementos que se explicarán son simplemente algunos términos de conocimiento general en este ámbito.

### 4.1. Transmisiones en vivo o directo

Se trata de un tipo de transmisión de contenido, generalmente de radio o de televisión, que se realiza al mismo tiempo que se producen, y no en diferido. Esto significa coloquialmente, que lo que vemos en la transmisión, está sucediendo al mismo tiempo que lo contemplamos, o quizá con algunos segundos de retraso, pero por lo general no existe edición o cortes de ningún tipo, aunque quizá sí transiciones [8]. Ya se han comentado antes algunas plataformas que se dedican a este tipo de contenido, pero un ejemplo claro sería un partido de fútbol en la televisión. Por lo general en el sistema empleado para llevar a cabo este tipo de transmisiones, se trata de una sola fuente o servidor que distribuye a muchos receptores al mismo tiempo el contenido que recibe del emisor, o al menos esto es lo que sucede en el ámbito de este proyecto, de forma que todos observan el mismo contenido de manera simultánea, y acciones como pausar la transmisión, no tendrán ningún efecto, ya que esta sigue su curso. Por último, hay que comentar que en inglés este término se conoce como "live" o "streaming". Un ejemplo del funcionamiento o del concepto, se puede ver en la Figura 5.



Figura 5: Live Streaming

### 4.2. Transmisión de video diferida

Se trata de un tipo de transmisión en la que se transmite contenido de vídeo, es decir, que ha sido previamente grabado con una longitud y calidad determinada. En este caso la transmisión se basa en guardar y enviar, es decir, los archivos están en un servidor, y una serie de usuarios pueden acceder bajo demanda cuando quieran, de forma que cada uno puede parar o retomar lo que está viendo, de modo que no se afecta a los demás usuarios [8]. Un ejemplo de esto es un video en YouTube. Adicionalmente, en este tipo de contenido, pueden existir restricciones específicas, ya que se puede controlar previamente a la subida al lugar de almacenamiento, por

ejemplo se podrían aplicar restricciones de edad o de localización. Un esquema similar a este concepto se ve en la Figura 6.



*Figura 6: Video Streaming*

En definitiva es un concepto fácil, pero que requiere entender el hecho de que estos videos han de subirse previamente. Se podría también realizar una retransmisión en vivo sobre contenidos previamente grabados si se emitiese un video capturándolo en vivo mientras se reproduce o con algún mecanismo parecido.

## 5. Técnicas y herramientas

En este apartado, nos centramos en explicar de forma teórica práctica, las distintas herramientas y tecnologías que se han usado para el desarrollo de la plataforma de este proyecto. Principalmente, serán las herramientas utilizadas para el servicio web, y las empleadas para el servidor de streaming y de almacenamiento de video, así como algunas herramientas para la documentación y pruebas.

### 5.1. Aplicación web

Primero, nos centramos en aquellas herramientas que han afectado directamente a la parte web de la aplicación. Se presentan las más importantes únicamente, aunque puede que se utilicen algunas más que no son tan relevantes.

#### 5.1.1. Estándar web

En este caso, se utiliza el que quizás, es el estándar web más extendido y soportado en el mundo. Este se trata de HTML5 +CSS, junto con el uso de JavaScript y la utilización del protocolo HTTP. Funciona usando JavaScript para realizar el control, HTTP para la comunicación y HTML y CSS para la estructura y presentación [9].

Estos elementos se pueden tratar por separado para aclarar en qué consisten. Los trataremos en posteriores apartados.

#### 5.1.2. Vue.js

Se trata de un framework de JavaScript, que es de código abierto, y sirve para la construcción de interfaces de usuario y aplicaciones de una sola página. Su uso es muy extendido mediante el empleo de la arquitectura Modelo-Vista-Vista Modelo y por tanto, se puede utilizar para crear estructuras de páginas web de una manera relativamente simple [10].

Para hacer más sencillo el trabajo, Vue permite cargar bibliotecas y paquetes de apoyo, mediante el uso de NPM, añadiendo las características más avanzadas y necesarias para el desarrollo de aplicaciones complejas, ya que la biblioteca principal se centra sobre todo en la vista. En concreto, para este trabajo, se han ido añadiendo numerosas bibliotecas que han servido para distintos fines, como Router, Vuex o Axios. Además de todo esto, Vue permite extender el HTML básico con directivas del usuario para así añadir funcionalidad extra.

En cuanto a las características más específicas de Vue, podríamos hablar de algunos de sus elementos. Por ejemplo, los componentes. Estos extienden de los elementos básicos de HTML para encapsular el código reutilizable. Un componente normal de vue, contiene una estructura predefinida que consta de tres partes, que son las siguientes:

**<template></template>**: Es la parte del componente que encapsula la estructura del componente mediante el uso de HTML.

**<script></script>**: Esta es la parte en la que se añade la funcionalidad, normalmente usando JavaScript.

**<style></style>**: Esta es la parte en la que se usa el CSS para el diseño del componente sobre la estructura.

También es importante comentar, que en Vue, se pueden utilizar numerosos templates o plantillas que se encuentran disponibles en algunas páginas como Vuetify, y que proporcionan componentes completos de uso libre.

Por otro lado, otra de las partes más importantes de Vue, es el enrutador, que se ofrece mediante una librería oficial Vue Router, y que permite que se realice el redireccionamiento a los distintos elementos o vistas de la aplicación web. En definitiva el router oficial de Vue.

Finalmente, se pueden comentar la que sería la estructura de un proyecto Vue, y cuáles son los elementos que vienen por defecto:

- **App.vue:** Es la raíz de la aplicación, y la primera interfaz de esta. En él se sobreponen el resto de los elementos de la interfaz.
- **Main.js:** Se trata del script responsable de iniciar el proyecto. Suele incluir la lógica de construcción y la previa de esta, así como la inicialización de ítems externos.
- **Views:** Se trata de la sección que almacena los componentes de un solo archivo, es decir las interfaces o pantallas de cara al usuario final.
- **Componentes:** Se trata de la sección que contiene los componentes modulares de la aplicación, que se usarán en las vistas.
- **Router:** Se trata de la sección que contiene el index de la aplicación, y que se encarga de enrutar dentro de esta.

### 5.1.3. JavaScript

Se trata de un lenguaje de programación orientado a objetos que es débilmente tipado y dinámico. Su utilización principal viene del lado del cliente implementado principalmente en navegadores web para aportar mejoras a la interfaz de usuario [11].

Está basado en el estándar ECMAScript y en prototipos, de forma que es soportado por todos los navegadores modernos. Además se ha diseñado con una sintaxis similar a C, lo que lo hace fácil de entender para personas con algo de experiencia previa en programación.

Actualmente, además de realizar operaciones y ser el marco de la aplicación cliente, también se usa para gestionar la recepción y envío de información de servidores con ayuda de otras tecnologías, como en nuestro caso Firebase.

### 5.1.4. Vuetify

Se trata de una biblioteca de interfaz de usuario de Vue, que ofrece componentes y materiales hechos a mano y de uso libre, de forma que no se requieren habilidades de diseño [12].

### 5.1.5. Axios

Se trata de una biblioteca para la realización de peticiones HTTP mediante promesas. Estas solicitudes contra servidores son completamente configurables, y gracias a las promesas se reciben respuestas de forma que sea sencillo procesarlas [13].

### 5.1.6. Firebase

Se trata de una plataforma para desarrollar aplicaciones web de Google, de modo que se ubica en la nube y se integra en la Google Cloud Platform, usándose como herramienta para la creación y sincronización de proyectos, haciendo que sea simple el crecimiento de usuarios y la monetización [14]. En el caso particular de este proyecto, se usa principalmente como base de



datos para la información de los usuarios y de los contenidos de la plataforma, así como para la autenticación.

Firebase, además aporta a los usuarios una gran cantidad de documentación para la creación de aplicaciones utilizando la plataforma, lo que simplifica que cualquier desarrollador pueda combinar y adaptar la plataforma a sus necesidades. Se pueden comentar cuáles son los principales servicios que ofrece la plataforma de Firebase, y que servirán de algún modo al proyecto, ya que tiene otros cuantos que no son realmente necesarios en nuestro caso.

- **Firebase Auth:** Se trata del servicio ofrecido por Firebase para autenticar a los usuarios de una aplicación, utilizando solo código del lado del cliente. Incluye múltiples formas de registro, como por ejemplo correo, teléfono o Facebook. Ofrece además todos los métodos para el inicio de sesión, registro, verificación y recuperación de contraseña entre otros, haciendo fácil la creación de un sistema de autenticación, ya que se encarga de la comprobación y de la sesión del navegador.
- **Real Time Database:** Se trata de la base de datos en tiempo real que proporciona Firebase. Este servicio proporciona a los desarrolladores una API para que la información sea almacenada, sincronizada y recuperada de Firebase. Este servicio, presenta también utilidades como caché frente a desconexiones o sincronización para acceder desde cualquier dispositivo.
- **Firebase Storage:** Se trata de una sección para la carga y descarga de archivos. Entre sus características, implica que la calidad de la red no sea demasiado relevante. De esta forma, se puede utilizar para almacenar imágenes, audio video y otros contenidos.
- **Firebase Cloud Firestore:** Se trata del servicio de almacenamiento, derivado de la Cloud Platform de Google. Es una base de datos NoSQL, lo que puede complicar un poco algunos aspectos de la aplicación, ya que contiene algunas estructuras complejas. Se puede decir adicionalmente, que se organiza en forma de documentos agrupados en colecciones, y en estos se pueden definir campos de diversos tipos.

Con estas utilidades que he definido, se puede ver que Firebase ofrece funcionalidades muy interesantes para una aplicación con las características de nuestro proyecto, lo que significa que lo podremos utilizar perfectamente, al menos para el desarrollo de un primer prototipo, aunque, podría escalar bien.

### 5.1.7. HTML

En inglés, HTML son las siglas de Hiper Text Markup Language. Se trata de un lenguaje para la elaboración de páginas web, que se ha establecido como un estándar [15].

En sus distintas versiones, este ofrece una estructura básica y un código para la definición de contenido de una página web. Además, basa su filosofía de desarrollo en la diferenciación, lo que permite añadir elementos externos, sin incrustarlo en el código de la página, si no mediante una referencia al contenido.

Por otro lado, hay que mencionar que es un lenguaje de marcado, lo que permite indicar la estructura de la web mediante las etiquetas que este utiliza, facilitando así la implementación, pudiendo crear jerarquías entre los elementos, pero permitiendo una clara diferenciación.

En definitiva un lenguaje muy dinámico y con gran adaptabilidad y estructuración lógica, aunque aún existen algunas diferencias entre navegadores y webs a la hora de interpretar sus páginas.

### 5.1.8. CSS

Se trata en inglés de las siglas para Cascading Style Sheets, se trata del lenguaje de diseño gráfico que puede definir y crear la presentación de documentos escritos en lenguajes marcados [16], como por ejemplo HTML. Muy usado en general, para definir el diseño visual de los documentos web o interfaces de usuario para este tipo de lenguajes.

Sirve además para separar el formato del contenido, de forma que podamos presentar en el mismo documento marcado con diferentes estilos y la misma estructura.

## 5.2. Servidor de streaming

En esta parte nos centramos en los componentes y herramientas que se han usado en el caso del servidor que se utiliza para almacenar y transmitir los contenidos en vivo y diferido, es decir el Backend CDN y el servidor de video.

### 5.2.1. Node.js

Se trata de un entorno en tiempo de ejecución multiplataforma para servidores. Se basa en el lenguaje JavaScript, que ya se ha comentado anteriormente, y usa una arquitectura orientada a eventos, basada en el motor V8 de Google [17]. Principalmente se usa para la creación de programas de red escalables, como servidores web.

Node.js, además funciona mediante un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas concurrentes de un número de cientos de miles. De esta forma, se atiende a aplicaciones de carácter altamente concurrente, tal como sería nuestra estructura.

### 5.2.2. NPM

Se trata del sistema de gestión de paquetes de Node.js, Node Pack Manager. Se ejecuta desde la línea de comandos y puede manejar las dependencias para una aplicación, además permite a los usuarios instalar aplicaciones Node.js que se encuentren en un repositorio [18].

### 5.2.3. Nginx y RTMP

Se trata de un servidor web de alto rendimiento con múltiples usos. Por ejemplo, se puede utilizar a modo de proxy reverso, así como para balanceo de carga, como proxy de mail o como caché de HTTP [19]. Por lo tanto, nginx se puede desplegar para servir contenido HTTP dinámico en la red. Además usa un enfoque asíncrono basado en eventos, lo que permite manejar múltiples solicitudes. También se puede comentar que presenta una arquitectura modular impulsada por eventos, de forma que puede proporcionar rendimiento bajo alta carga.

Para nuestro propósito, debemos añadir a nginx, un módulo RTMP o Real Time Messaging Protocol. Este se trata de un protocolo para transmisión de audio y video a través de internet, desde un servidor a un reproductor flash. Como RTMP funciona perfectamente con peticiones HTTP, se puede combinar con Nginx para la transmisión en directo.

### 5.2.4. PM2

Se trata de un gestor de procesos para JavaScript en tiempo de ejecución de Node.js [20]. Sirve por lo tanto, principalmente para administrar procesos de Node.js en un entorno de producción. También sirve para el equilibrio automático de carga de aplicaciones así como para el auto inicio

y la monitorización. Se usa en nuestro caso principalmente para el arranque automático del servidor de subida de videos del que se hablará más adelante.

### 5.3. Herramientas de documentación y desarrollo

En esta parte, nos centraremos en explicar cuáles son y qué hacen, las herramientas que se han usado para la creación de documentos, estimaciones, código y diagramas. En general herramientas del proceso que han ayudado al desarrollo.

#### 5.3.1. Draw.io

Se trata de una herramienta, que ofrece extensas posibilidades para la creación de diagramas, gráficos, máquinas de estado, árboles de decisión, entre otros [21]. Además de su creación, una vez se ha guardado el diagrama en un archivo, se puede abrir de nuevo para editarlo. Una vez creados los diagramas se pueden exportar en numerosos formatos, como “.png” entre otros. Esta herramienta puede ser accedida desde Google drive, por lo que además de la facilidad para usarla, también se puede acceder de manera simple y guardar directamente los diagramas en drive.

#### 5.3.2. Microsoft Project

Es una herramienta o software de administración de proyectos, comercializada por Microsoft, que sirve para asistir a la administración de proyectos en distintos aspectos, como el desarrollo de planes, asignación de recursos a tareas, actualización de cargas de trabajo, y dar seguimiento al progreso [22]. En concreto, este software es muy utilizado en sus distintas versiones, para la realización de estimaciones temporales, ya que además de asignar recursos, permite asignar tiempo a las tareas, lo que da lugar a la configuración de un calendario para la determinación de tareas críticas mediante dependencias.

#### 5.3.3. Visual studio code

Se trata de un editor de código fuente de Microsoft, que funciona en distintos sistemas operativos [23]. Incluye soporte para la depuración, control integrado de Git, resultados de sintaxis, finalizador de código automática, así como refactorización y estilado de código. Todo esto además de ser personalizable .

Es compatible con numeroso lenguajes de programación, por lo tanto, este es capaz de soportar un proyecto de tipo web en vue.js sin ningún problema.

Adicionalmente, la interfaz del programa es bastante sencilla pero completa, nos permite desplegar números opciones ocultas a medida que se van necesitando, de forma que no estorben demasiado, así como la instalación de extensiones.

#### 5.3.4. EZEstimate

Se trata de una herramienta que sirve fundamentalmente para la realización de estimaciones de esfuerzo para un proyectos, en base a la métrica de UCP. Como resultado de una estimación obtenemos el número de horas por caso de uso. Sus resultados se pueden exportar en formato Excel.

### 5.3.5. JSDoc

Se trata de una sintaxis que permite generar documentación automática sobre código fuente de proyectos escritos en JavaScript en base a comentarios [24]. Es dinámico y efectivo en la creación de este tipo de documentos.

### 5.3.6. Vue Styleguidist

Se trata de una herramienta para la generación de documentación automática en un proyecto escrito en Vue, y que únicamente requiere de comentar el código [25]. En este caso es muy útil dadas las circunstancias.

### 5.3.7. Adobe XD

Se trata de un editor de gráficos vectoriales que se usa comúnmente para diseñar y crear prototipos para páginas web, de forma que puedan ser interactivos, para comprender la experiencia del usuario [26].

### 5.3.8. Google Drive

Se trata de una plataforma de almacenamiento en la nube de Google, que se ha utilizado para el almacenamiento de documentos y copias de seguridad del proyecto.

## 5.4. Herramientas de prueba

Se presentan ahora una serie de herramientas, que se han utilizado durante el desarrollo para poder demostrar el funcionamiento de algunos aspectos implementados, como por ejemplo las emisiones en directo o la visualización de los vídeos.

### 5.4.1. OBS

Se trata de las siglas en inglés de Open Broadcaster Software. Una aplicación de código abierto para grabación y transmisión de video por internet [27]. Permite la captura de fuentes de vídeo en tiempo real, así como la composición de escena, codificación y grabación de retransmisión, que se realiza a través de RTMP, enviando datos a cualquier destino que soporte este protocolo. En nuestro caso en servidor Nginx con módulo RTMP. De forma que sirve para la prueba del funcionamiento de este. Igualmente, es una opción muy viable a nivel de usuario para transmitir contenidos en la plataforma, ya que cuenta con una buena reputación y que recibe soporte de muchas de las grandes plataformas como YouTube, Twitch, entre otras.

### 5.4.2. VirtualBox

Se trata de un software de virtualización desarrollado por Oracle que sirve para diferentes arquitecturas. Gracias a este software, es posible tener sistemas operativos adicionales dentro del sistema operativo principal [28]. En nuestro caso, se ha utilizado una máquina virtual con una de las distribuciones soportadas de Ubuntu, de forma que en él están funcionando los servidores de subida de videos y transmisión, mediante el uso de Node.js y una configuración de Nginx. Los documentos en este sistema han sido creados con la herramienta nano. Todo esto con fines de realización de pruebas de transmisión.

## 6. Aspectos relevantes del desarrollo

En este apartado, nos centramos en explicar algunas de las partes fundamentales y más importantes del desarrollo de la plataforma presentada en el proyecto de fin de grado. Para realizar esto, tendremos que explicar muchos conceptos que han aparecido en los anexos presentados, desde metodologías, hasta especificaciones y estimaciones.

### 6.1. Metodología utilizada

Para la realización de este proyecto, se ha empleado una adaptación del **proceso unificado** [29], ya que es un modelo que se ha estudiado con anterioridad, lo que permitirá agilizar el proyecto. Se explican a continuación sus características principales.

El proceso unificado, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, en diferentes organizaciones con diferentes niveles de amplitud y tamaños de proyectos. El proceso unificado tiene características en dos vertientes:

- **Características generales:**
  - Basado en componentes.
  - Utiliza UML (Unified Modeling Language).
- **Características principales:**
  - Es un proceso conducido por casos de uso.
  - Centrado en arquitectura.
  - Es iterativo e incremental.

El proceso unificado se repite a lo largo de una serie de **ciclos de desarrollo**, para los que se pueden definir gran variedad de estrategias. Este ciclo de desarrollo forma la vida del sistema. Además, cada uno de estos ciclos, termina con una versión entregable del producto. Los ciclos constan de tres **fases** bien definidas.

- **Inicio:** En el que se define el alcance del proyecto y se hacen los casos de negocio.
- **Elaboración:** En la que se planifica el proyecto, especificando los casos de uso y el diseño de la arquitectura.
- **Construcción:** En la que se realiza o construye el producto.
- **Transición:** En la que el producto se establece como una versión beta y se corrigen problemas mediante mejoras de la revisión.

Un esquema de estas fases se puede ver en la Figura 7.

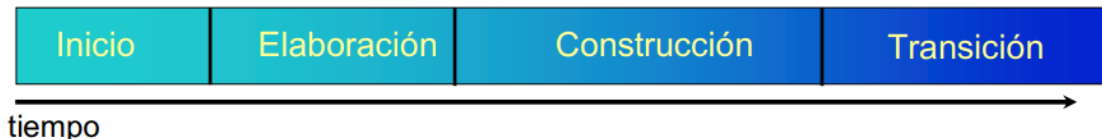


Figura 7: Fases de un Ciclo

Estas fases que hemos descrito se pueden clasificar en dos etapas: Etapa de **ingeniería**, que contiene el inicio y la elaboración, se realiza con equipos pequeños esperando actividades poco

predecibles, y por otro lado, la etapa de **Producción**, que se concentra en las fases de construcción y transición y se realiza en equipos grandes mediante actividades predecibles.

En cada uno de estos ciclos, las fases, se pueden a su vez dividir en **iteraciones** con sus resultantes incrementos.

Cada fase termina con un **hito** que se caracteriza por la disponibilidad de un conjunto de componentes de software. En cada uno de estos hitos se revisa el progreso del proyecto para identificar riesgos y evaluar la situación global. También se necesitan en estos hitos resultados tangibles. Los hitos pueden ser de dos tipos. **Principales**, al final de cada fase y **secundarios** al final de cada iteración.

Estas iteraciones mencionadas antes, son una secuencia de actividades cuyo resultado es una versión entregable. Además, estas iteraciones discurren a lo largo de **disciplinas**, que son las que organizan las actividades fundamentales de gestión y desarrollo del proyecto. Las más importantes son:

- **Modelado del negocio**
- **Requisitos**
- **Análisis**
- **Diseño**
- **Implementación**
- **Pruebas**

Un esquema de este ciclo con las distintas fases, iteraciones y disciplinas se puede observar en la Figura 8.

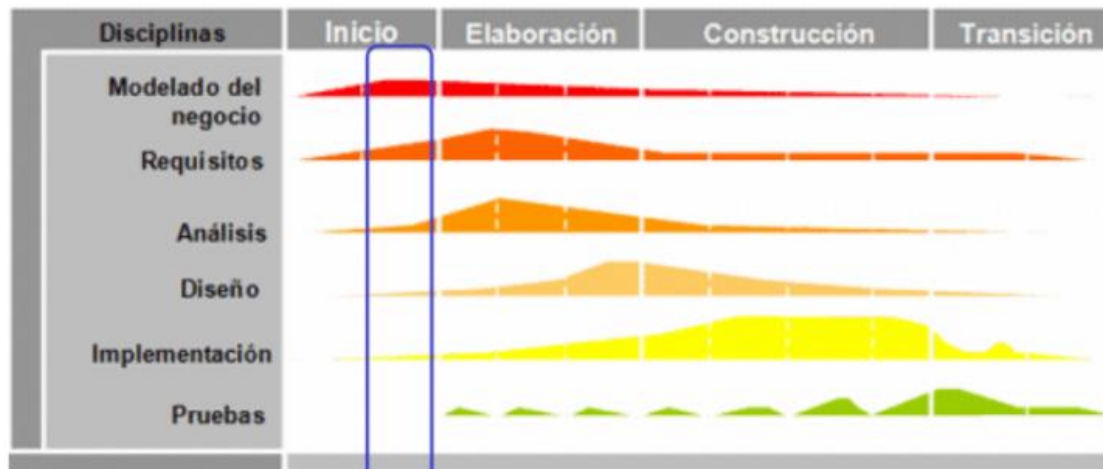


Figura 8: Proceso unificado

## 6.2. Estimación de esfuerzo

Para poder mejorar el desarrollo, y teniendo todos los requisitos, se ha realizado una estimación de esfuerzo, mediante el empleo de la herramienta llamada **EZEstimate**, que se basa en el cálculo del esfuerzo mediante los puntos de casos de uso o **UCP**, que se calculan en base a los actores, los casos de uso y los factores de complejidad del entorno y técnica.

A continuación se puede observar la realización del cálculo con la herramienta en la Figura 9.

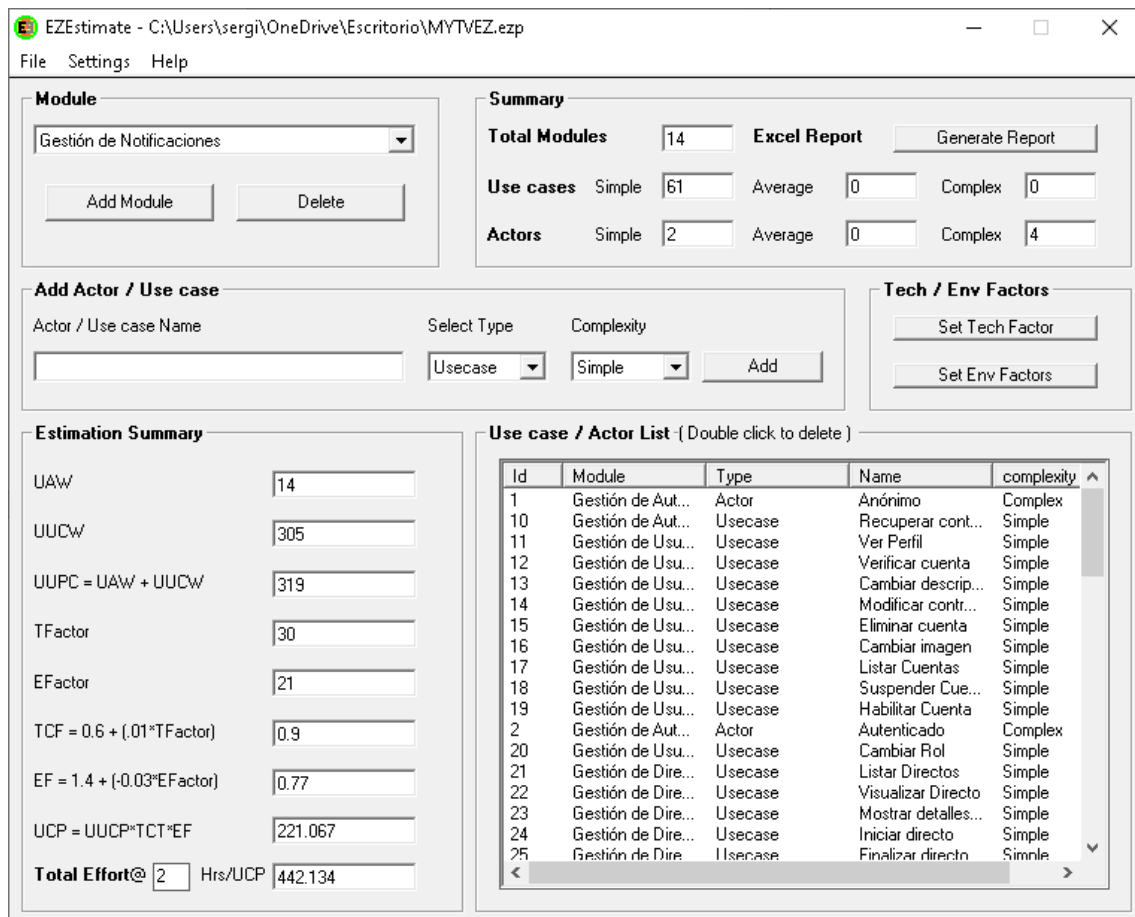


Figura 9: Cálculo y resultados de la estimación

Como resultado, se obtienen unos tiempos bastante razonables, aproximadamente 442 horas, lo que son, calculando 35 horas de trabajo por semana, unas **12 semanas y media**.

Todos los detalles de la realización de esta estimación se encuentran en el **Anexo I**.

### 6.3. Planificación temporal

También, se ha realizado una planificación temporal, para asignar a las tareas en las que se ha descompuesto el proyecto, los tiempos adecuados y los recursos disponibles. Para realizar esta planificación temporal, se ha usado **Microsoft Project**.

Esta planificación, sirve como una guía para observar la evolución del proyecto y determinar los distintos hitos que se van alcanzando, así como las tareas no se pueden retrasar por formar parte de la sección crítica.

Esta planificación temporal, mantiene un esquema como el de la Figura 10, en cuanto a las fases, iteraciones y disciplinas.

Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
	Inicio	6 días?	mié 24/02/21	lun 01/03/21	
	Iteración 1	5 días?	mié 24/02/21	dom 28/02/21	
	Modelado de Negocio	1 día?	mié 24/02/21	mié 24/02/21	
	Requisitos	3 días?	jue 25/02/21	sáb 27/02/21	
	Análisis	2 días?	jue 25/02/21	vie 26/02/21	
	Diseño	1 día?	dom 28/02/21	dom 28/02/21	
	Fin iteración 1	1 día?	lun 01/03/21	lun 01/03/21	2
	Fin inicio	1 día?	mar 02/03/21	mar 02/03/21	1
	Elaboración	8 días?	mié 03/03/21	mié 10/03/21	19
	Iteración 2	7 días?	mié 03/03/21	mar 09/03/21	
	Modelado de Negocio	2 días?	mié 03/03/21	jue 04/03/21	
	Requisitos	2 días?	mié 03/03/21	jue 04/03/21	
	Análisis	2 días?	jue 04/03/21	vie 05/03/21	
	Diseño	2 días?	vie 05/03/21	sáb 06/03/21	
	Implementación	2 días?	sáb 06/03/21	dom 07/03/21	
	Pruebas	3 días?	dom 07/03/21	mar 09/03/21	
	Fin iteración 2	1 día?	mié 10/03/21	mié 10/03/21	21
	Fin Elaboración	1 día?	jue 11/03/21	jue 11/03/21	20
	Construcción	10 días?	vie 12/03/21	dom 21/03/21	51
	Iteración 3	4 días?	vie 12/03/21	lun 15/03/21	
	Modelado de Negocio	1 día?	vie 12/03/21	vie 12/03/21	
	Diseño	1 día?	sáb 13/03/21	sáb 13/03/21	
	Implementación	1 día?	dom 14/03/21	dom 14/03/21	
	Pruebas	1 día?	lun 15/03/21	lun 15/03/21	
	Fin iteración 3	1 día?	mar 16/03/21	mar 16/03/21	53
	Iteración 4	4 días?	mié 17/03/21	sáb 20/03/21	73
	Modelado de Negocio	1 día?	mié 17/03/21	mié 17/03/21	
	Diseño	1 día?	jue 18/03/21	jue 18/03/21	
	Implementación	1 día?	vie 19/03/21	vie 19/03/21	
	Pruebas	1 día?	sáb 20/03/21	sáb 20/03/21	
	Fin iteración 4	1 día?	dom 21/03/21	dom 21/03/21	74
	Fin Construcción	1 día?	lun 22/03/21	lun 22/03/21	52
	Transición	4 días?	mar 23/03/21	vie 26/03/21	97
	Iteración 5	3 días?	mar 23/03/21	jue 25/03/21	
	Modelado de Negocio	2 días?	mar 23/03/21	mié 24/03/21	
	Diseño	1 día?	mié 24/03/21	mié 24/03/21	
	Pruebas	1 día?	jue 25/03/21	jue 25/03/21	
	Fin iteración 5	1 día?	vie 26/03/21	vie 26/03/21	99
	Fin Transición	1 día?	sáb 27/03/21	sáb 27/03/21	98

Figura 10: Estructura

En cuanto a los hitos principales de las fases del ciclo de vida, se pueden observar los siguientes resultados:

- **Fin Inicio:** Hito principal en el que se establecen las bases del proyecto y los recursos, teniendo una reunión con los tutores, y mediante la elicitación de requisitos. También se toma la decisión de qué herramientas utilizar en el desarrollo.
- **Fin Elaboración:** Hito principal en el que se da una nueva reunión con los tutores y se realizan las estimaciones. Además se da un mayor nivel de detalle a los requisitos, documentándolos y realizando diagramas. Por lo tanto se tienen los casos de uso de análisis, y algunos realizados en el dominio de la solución, comenzando por tanto con la parte del diseño del sistema, así como con la implementación de algunos componentes y sus pruebas unitarias y de integración.
- **Fin Construcción:** Hito principal en el que se ha completado el diseño realizando todos los casos de uso, así como también se diseña el despliegue. Esto además de la implementación de todos los componentes del proyecto que quedaban por hacer. También se realizan las pruebas unitarias de los componentes, las de integración, y las finales de sistema completo.



- **Fin Transición:** Hito principal en el que se realizan las pruebas en base a las correcciones finales, así como una nueva reunión con los tutores. También se termina el proyecto y la realización de anexos.

Los resultados de la planificación temporal son muy parecidos a los de la estimación de esfuerzo. Esto se ve en la Figura 11, teniendo en cuenta que se estiman aproximadamente 5 horas por día de trabajo.

	Comienzo	Fin
Actual	mié 24/02/21	jue 03/06/21
Previsto	NOD	NOD
Real	NOD	NOD
Variación	0d	0d

	Duración	Trabajo	Costo
Actual	92.5d	450h	0.00 €
Previsto	0d	0h	0.00 €
Real	0d	0h	0.00 €
Restante	92.5d	450h	0.00 €

Porcentaje completado:  
 Duración: 0%      Trabajo: 0%

Figura 11: Horas finales

Se puede sacar en conclusión un tiempo aproximado de **12 semanas y media**.

Todos los detalles de la realización de esta planificación, incluidas todas las tareas, el camino crítico, las relaciones entre las tareas, sus duraciones e hitos, se encuentran en el **Anexo I**.

## 6.4. Especificaciones de requisitos del software

Continuando con el desarrollo, se ha realizado la especificación de los requisitos del software del sistema. Para realizar la documentación de esta especificación, se ha utilizado la **metodología Durán y Bernárdez [30]**.

Toda esta información, se encuentra completa en el **Anexo II**. En los siguientes subapartados, solo se comenta de forma resumida las acciones que se han realizado sobre los requisitos, objetivos, actores, casos de uso y algunos ejemplos de las tablas utilizadas.

### 6.4.1. Participantes

En la especificación de requisitos, primero se han recogido a los **participantes del proyecto**, tanto el alumno como los tres profesores de la Universidad de Salamanca (**USAL**).

### 6.4.2. Objetivos

Posteriormente, se han recogido los **objetivos** que se consideran imprescindibles para el proyecto, y que también han sido mencionados en esta memoria.

### 6.4.3. Requisitos de información

También, se han recogido los **requisitos de información** que son necesarios para el funcionamiento del sistema.

### 6.4.4. Requisitos Funcionales

Después, se ha comenzado la especificación de los requisitos funcionales del sistema. Para ello, primero ha sido necesario especificar los **paquetes que componen el sistema**, para así, establecer una base sobre la que organizar los requisitos funcionales.

Posteriormente, también se han descrito cuales son los **actores** que interactúan con el sistema, necesarios también para definir los requisitos funcionales. Estos actores han sido especificados en tablas en el anexo mencionado, al igual que casi todos los elementos anteriores.

Una vez definidos los paquetes y los actores del sistema, se ha utilizado, para definir los requisitos funcionales del sistema, la técnica de los **casos de uso**. Estos, se han representado en tablas como las especificadas en la metodología Durán y Bernárdez, como ya se había comentado antes. Todos los casos de uso, en total 61, se encuentran especificados de forma detallada en el **Anexo II**.

Además de esto, también se han realizado **diagramas de casos de uso** para organizar los casos de uso en los diferentes paquetes del sistema. Por ejemplo, se puede ver uno de estos diagramas en la Figura 12.

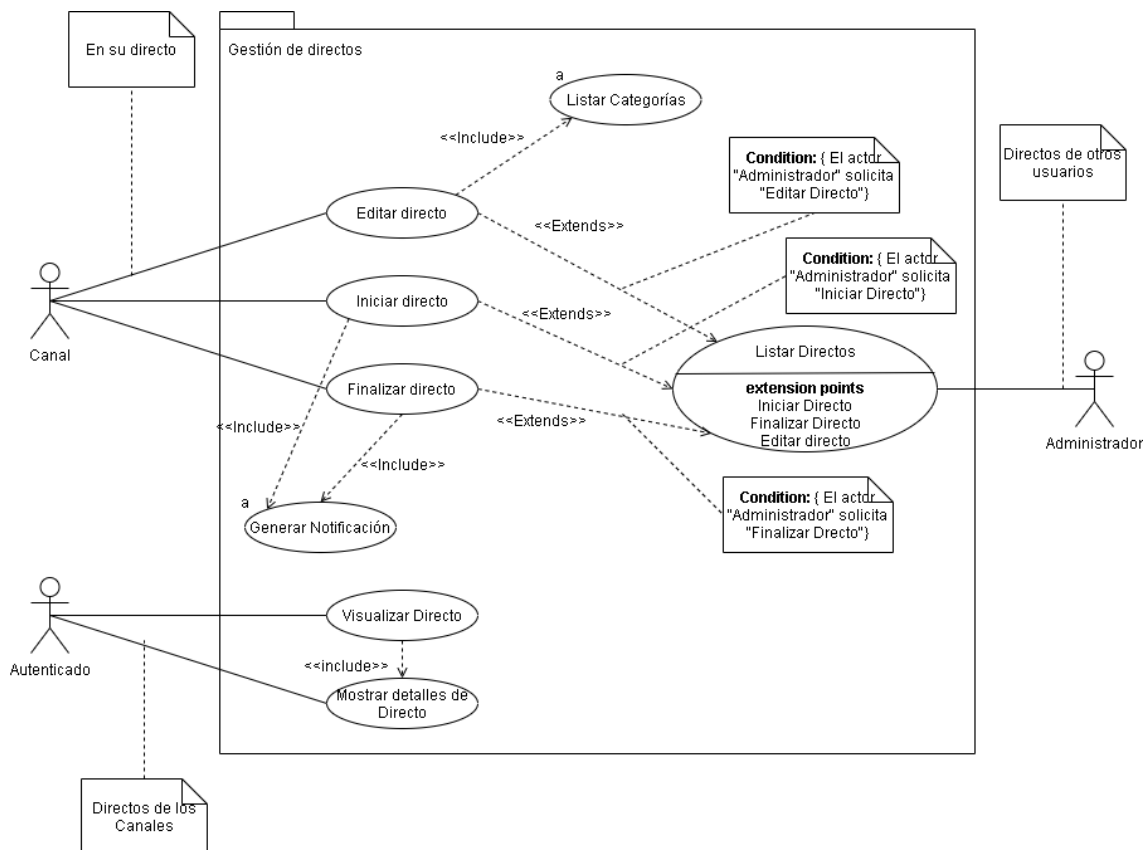


Figura 12: Diagrama de casos de uso en el paquete Gestión de Directos

#### 6.4.5. Requisitos no Funcionales

También se han registrado los requisitos no funcionales del sistema, que son los atributos de calidad del sistema. Estos se han clasificado en requisitos del producto, de la organización y externos.

#### 6.4.6. Matriz de rastreabilidad

Para finalizar, se ha realizado una matriz de rastreabilidad de los requisitos, para aclarar así las dependencias entre estos y los objetivos del sistema. Esta matriz se encuentra también en el **Anexo II**.

### 6.5. Análisis

Por otro lado, en la fase de análisis, se ha creado una estructura más precisa del sistema, de forma que se pueda dar un paso más en la transición al diseño. Para ello, se realizará un refinamiento y estructuración de los requisitos obtenidos en el apartado anterior.

Toda esta información, se encuentra detallada en el **Anexo III**, aunque a continuación veremos los aspectos más importantes de forma resumida.

#### 6.5.1. Modelo del dominio

En primer lugar, se ha realizado un **modelo del dominio**, que nos permite determinar las clases conceptuales que forman parte del dominio del problema, así como sus atributos y sus relaciones. Todo esto en base a los requisitos de información.

Para representar el modelo del dominio, se ha creado un diagrama que se puede ver en la Figura 13.

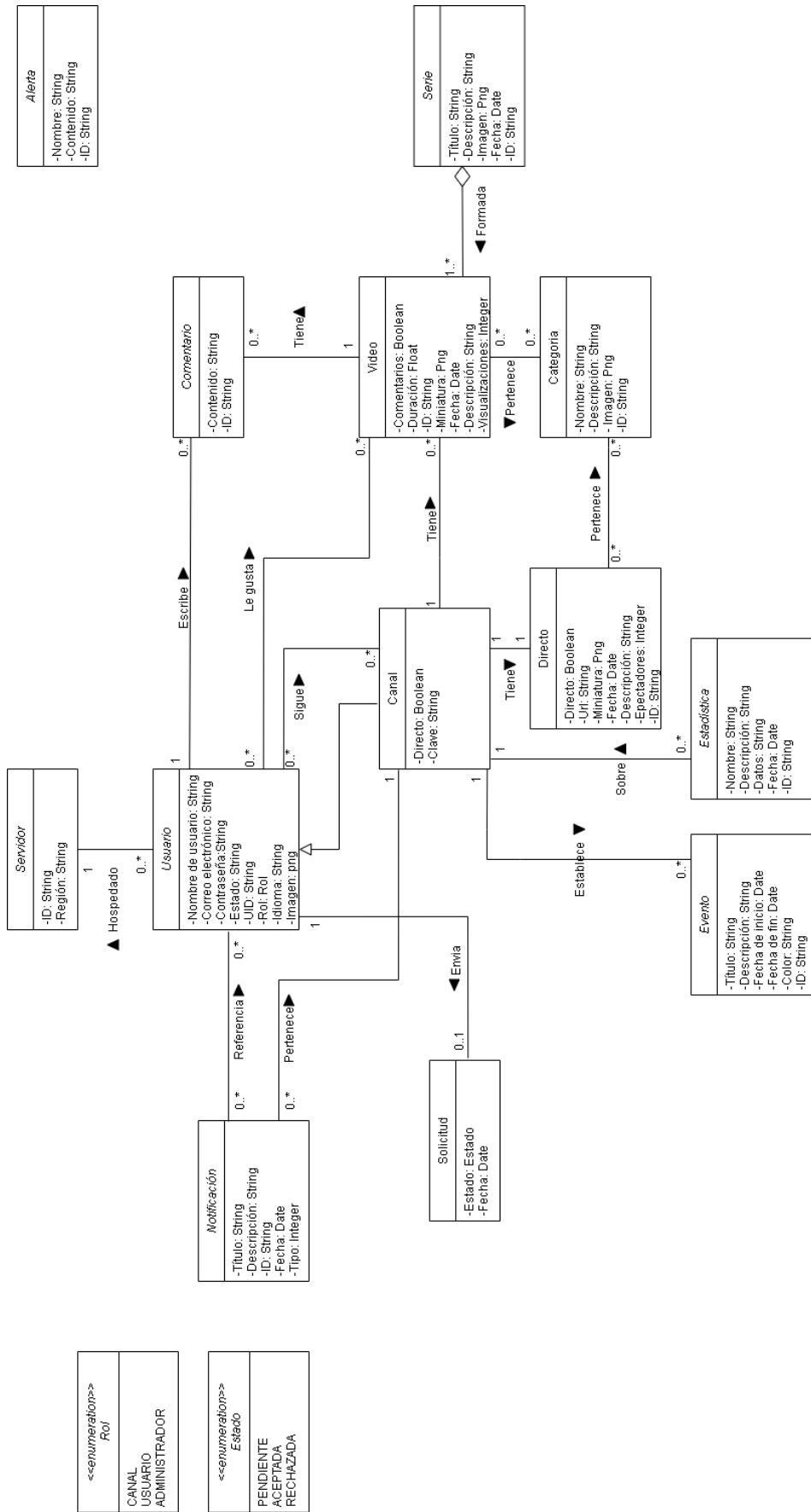


Figura 13: Modelo del dominio

Todas las clases que se presentan en él se pueden entender de forma detallada en el **Anexo III**.

### 6.5.2. Paquetes y clases de análisis

Posteriormente, se ha realizado la descomposición del sistema en unidades más elementales, mediante la identificación de las **clases de análisis** (entidad, control, interfaz), que abstraen a las clases de diseño, así como también se han identificado los **paquetes** en los que participan.

### 6.5.3. Realización de los casos de uso, modelo de análisis

Finalmente, para terminar el análisis, se ha realizado el **refinamiento de los casos de uso** mediante diagramas de secuencia, profundizando así más en estos, y mostrando su ejecución en base a las clases de análisis mencionadas anteriormente. Un ejemplo de uno de estos diagramas de secuencia se observa en la Figura 14.

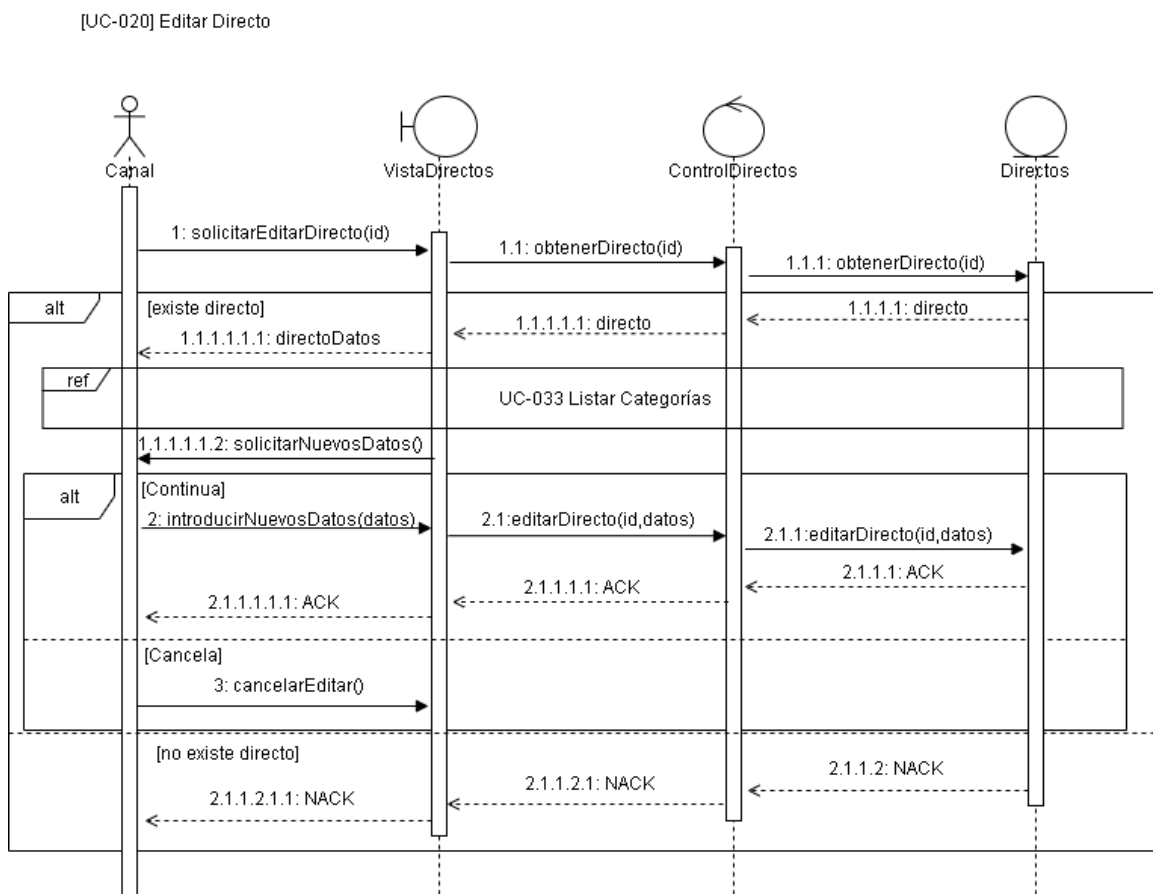


Figura 14: Diagrama de secuencia de caso de uso Editar Directo UC-020

## 6.6. Diseño

En el diseño, nos centramos en la realización física de los casos de uso, es decir su implementación en el dominio de la solución en base al análisis de requisitos del dominio del problema. Para esto, se crea un modelo de objetos que muestra cómo los requisitos, tienen impacto en el sistema.

Toda la información del diseño está detallada en el **Anexo IV**. En este apartado vemos de forma resumida algunos de los aspectos fundamentales.

### 6.6.1. Modelo de diseño

En primer lugar, es importante elegir qué utilizar y que no en el diseño, por lo tanto, para empezar se han escogido algunas tecnologías que se pretenden utilizar, para así elegir mejor los patrones. Algunas de estas son:

- **Vue.js:** Un framework para el desarrollo web. Su uso principalmente se basa en el patrón **MVVM**, mediante la utilización de **HTML**, **CSS** y **JavaScript**.
- **Firebase:** Un servicio de base de datos de Google con buenas capacidades para la gestión de usuarios y de datos.
- **Nodejs:** Un entorno en tiempo de ejecución multiplataforma para servidores.
- **Nginx:** Un servidor proxy de alto rendimiento que puede servir contenido con HTTP.

#### 6.6.1.1. Patrones arquitectónicos y de diseño

En este caso, se ha elegido un patrón en base a la tecnología detallada anteriormente. Este se usará en el diseño del sistema, ya que ofrece una solución abstracta a un problema común. Este patrón es:

##### **Modelo-vista-modelo de vista:**

También conocido como **MVVM**, es un patrón arquitectónico, destinado a organizar un sistema para el desarrollo de interfaces con lenguaje marcado [31]. Se utiliza mucho en aplicaciones web como Vue.js.

Este patrón, organiza el sistema desacoplando la interfaz de usuario de la lógica de aplicación. Similar al modelo vista controlador, pero cambiando el controlador por el ViewModel, que se encarga de gestionar al modelo. Este patrón se construye con 3 capas:

- **Vista:** La capa, más externa, encargada de la comunicación con el usuario. Esta vista es muy activa, conteniendo comportamientos y eventos.
- **Modelo de Vista o ViewModel:** La capa intermedia, encargada de la lógica de presentación, siendo una extensión abstracta de la interfaz, comunicando modelo y vista de forma bidireccional.
- **Modelo:** Capa en la que residen los datos del sistema, y que representa a la lógica del negocio. Contiene información de los objetos, pero no las acciones para manipularlos.

Conseguimos mediante este patrón una alta cohesión y bajo acoplamiento. Se puede observar su esquema en la Figura 15.

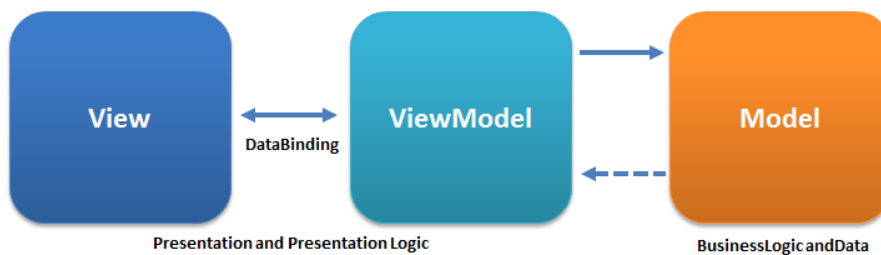


Figura 15: Patrón MVVM

### 6.6.1.2. Subsistemas de diseño

Después de definir los patrones, se descompone el sistema en una serie de subsistemas de diseño, para manejarlo mejor. En nuestro caso se ha descompuesto en dos paquetes:

- **Aplicación web:** Subsistema superficial encargado de interactuar con los usuarios, ofreciéndoles acceso, así como distintas interfaces y opciones en función de sus roles. También contiene las llamadas necesarias para gestionar la base de datos Firebase.
- **Backend CDN (Control Delivery Networks):** Subsistema API encargado de gestionar las peticiones de subida de los videos a un servidor Node para su posterior transmisión.

En la Figura 16, se muestra una imagen en la que se han representado estos subsistemas.

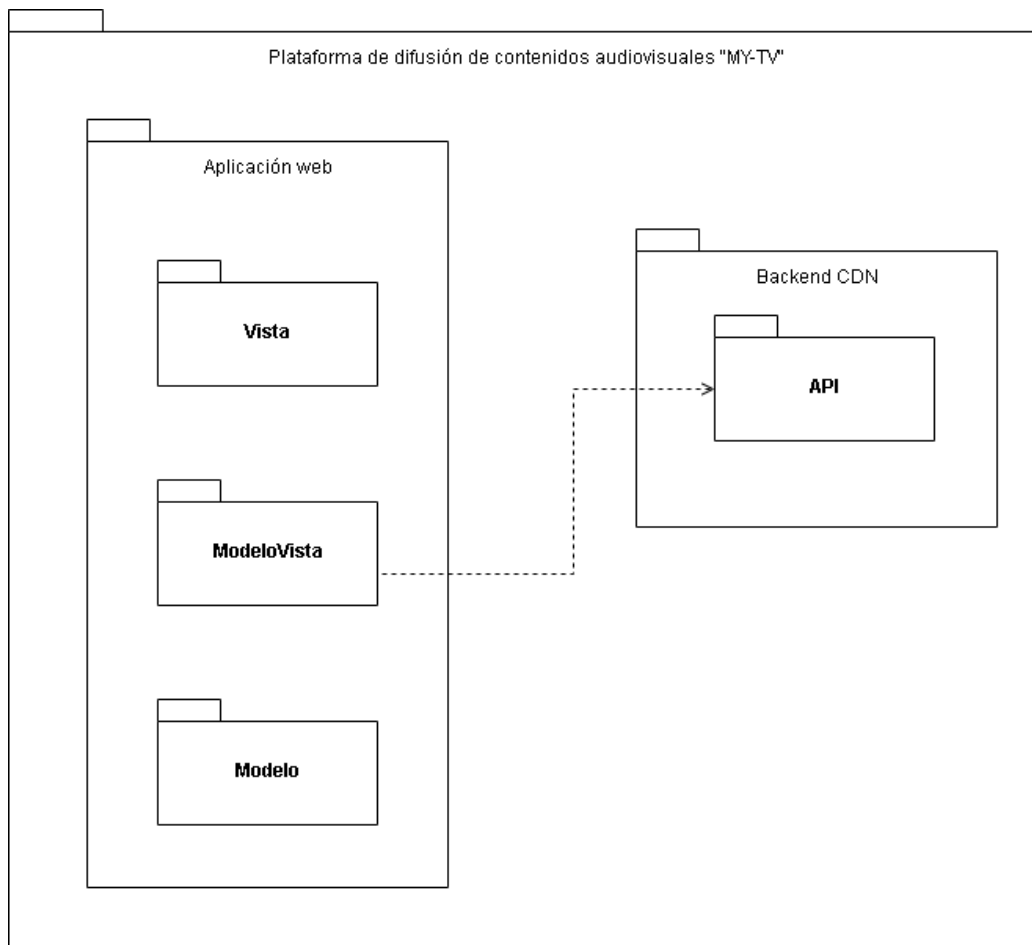


Figura 16: Diseño Subsistemas

### 6.6.1.3. Clases de diseño

En este punto, se detalla de forma más específica cada subsistema y sus clases de diseño, especificando para cada paquete las capas del MVVM con sus funciones, métodos y atributos más importantes.

#### 6.6.1.3.1. Aplicación web

Se trata del subsistema superficial de monitorización y acceso a la información de la aplicación. Detallamos ahora sus elementos relevantes para las 3 capas del Patrón MVVM:

**Vista:** En esta se pueden encontrar una serie de componentes ".vue" que conforman la vista de la aplicación web, estando de cara al usuario. Estos componentes se han representado en la Figura 17.

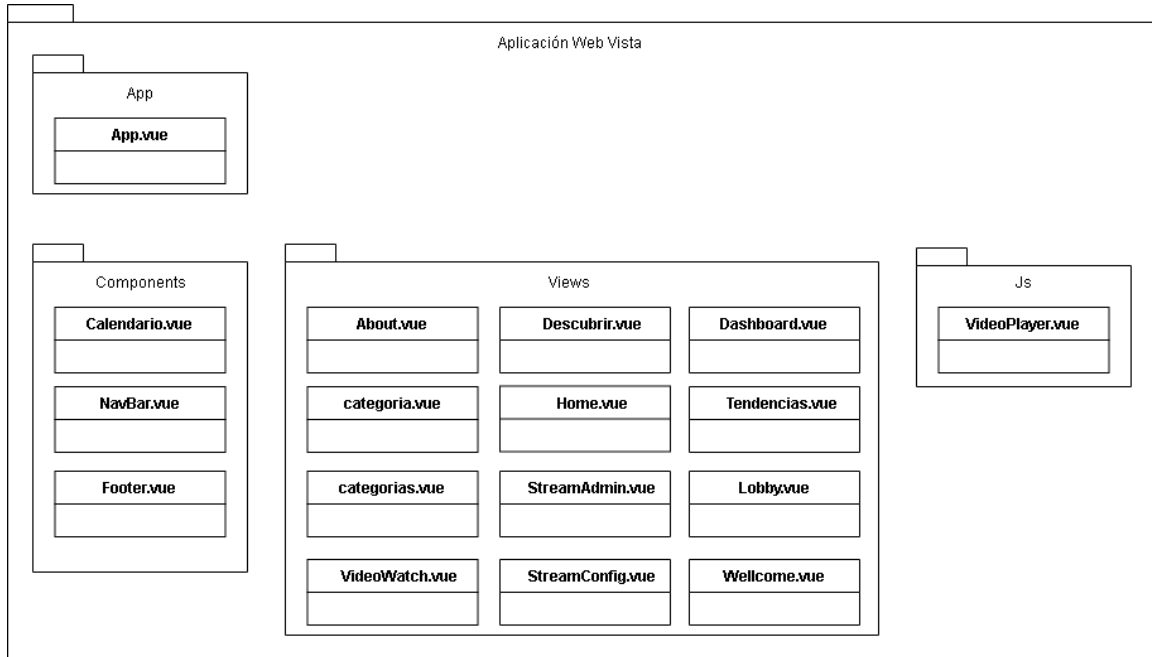


Figura 17: Vista

**ModeloVista:** Los elementos de esta capa son responsables de la corriente de datos del modelo a la vista y viceversa. Se trata de una serie de métodos que están en los componentes de la vista.

Estos elementos han sido representados en un diagrama que se puede observar en la Figura 18, junto con las funciones específicas del autenticador de firebase en la Figura 19, que se utilizarán para el mantenimiento y gestión de sesiones en el navegador, así como los inicios de sesión.



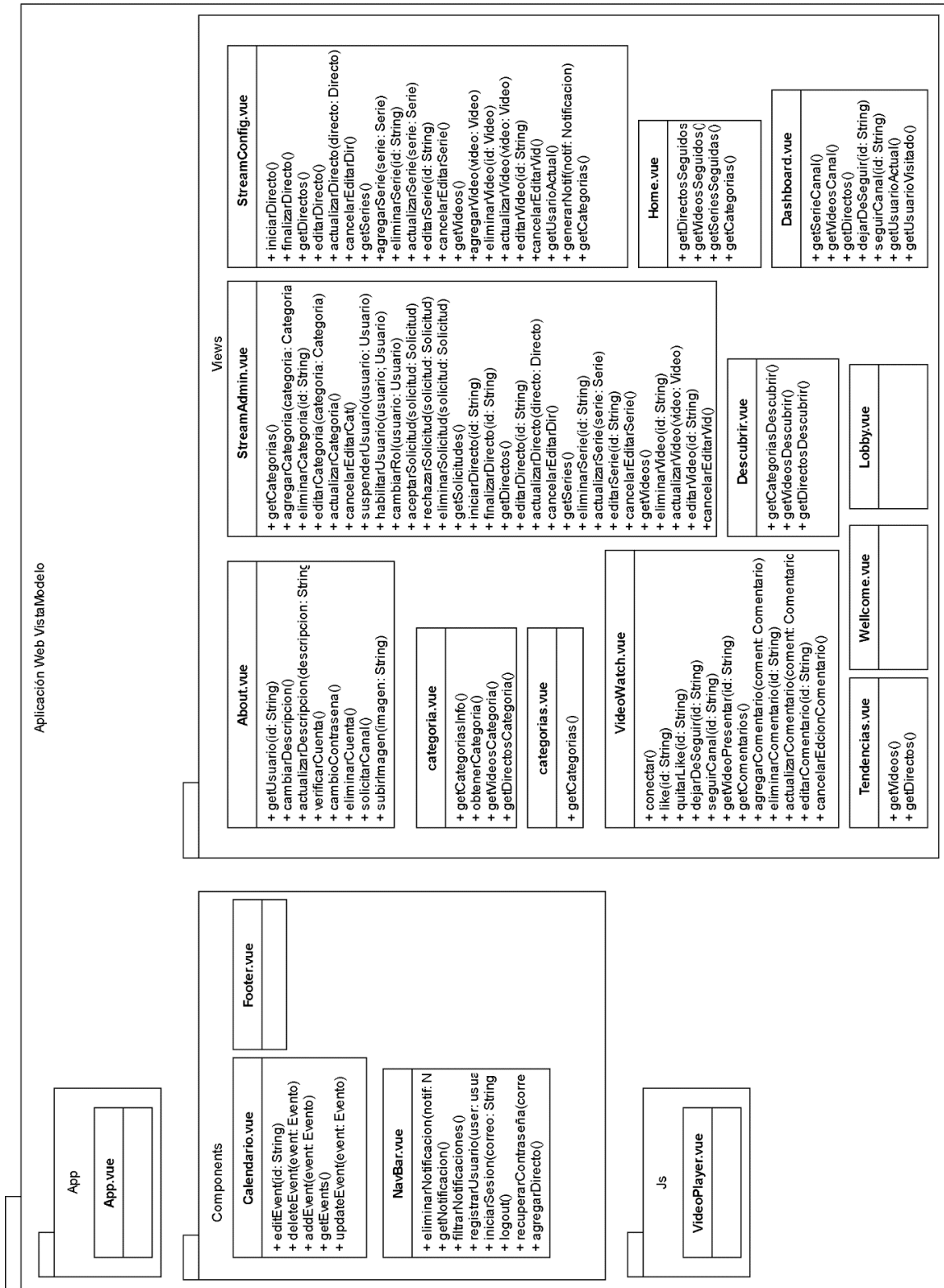


Figura 18: ModeloVista

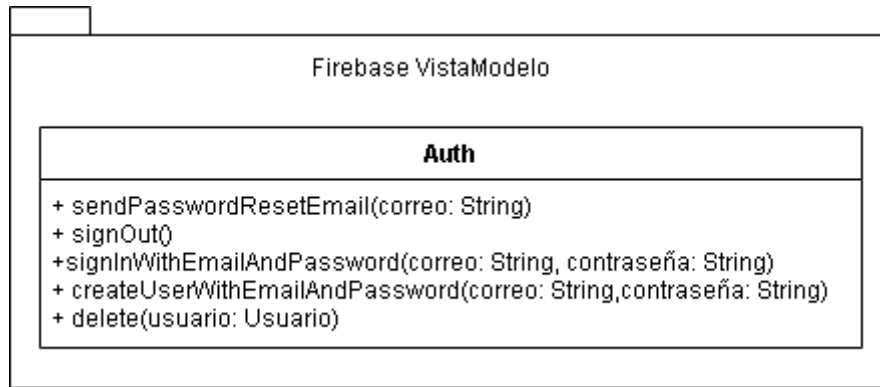


Figura 19: Auth

**Modelo:** En este modelo, se encuentran todos los datos necesarios para el funcionamiento del sistema. Sucede que al utilizar JavaScript, tratamos, solo atributos y no objetos. Para una mejor comprensión de la información, se pueden organizar los atributos en entidades, como se observa en la Figura 20. Todos los atributos se pueden observar en la Figura 21.

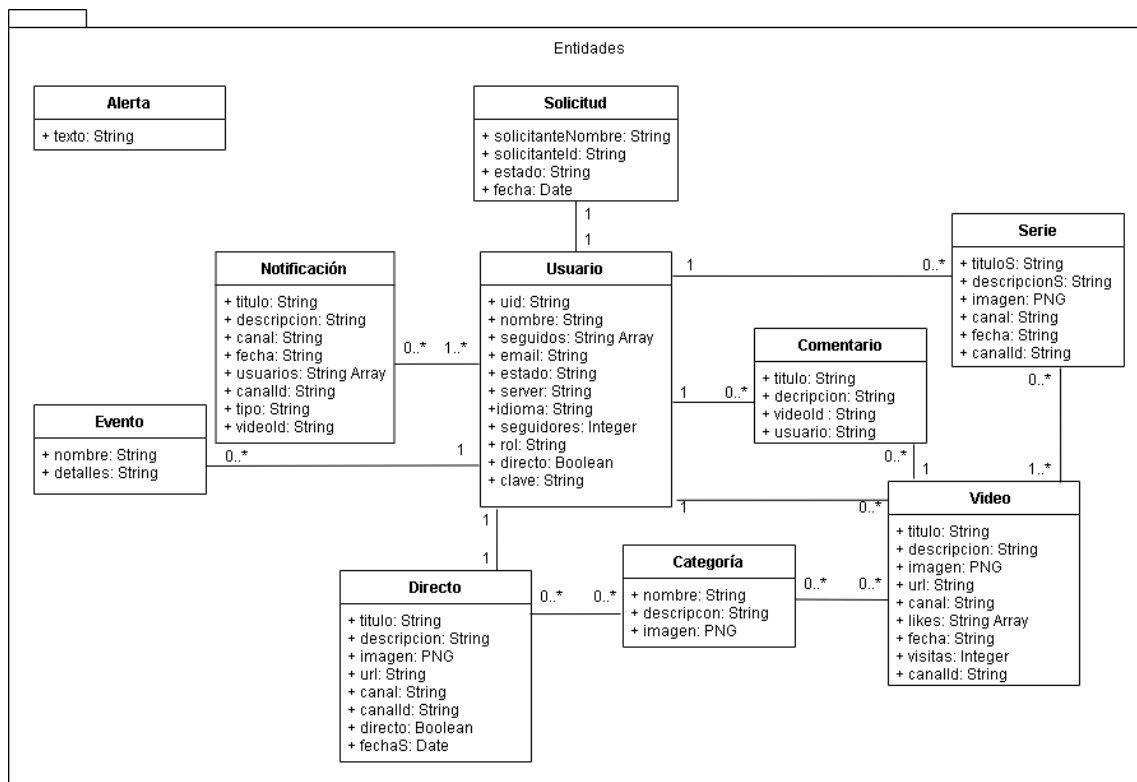


Figura 20: Entidades

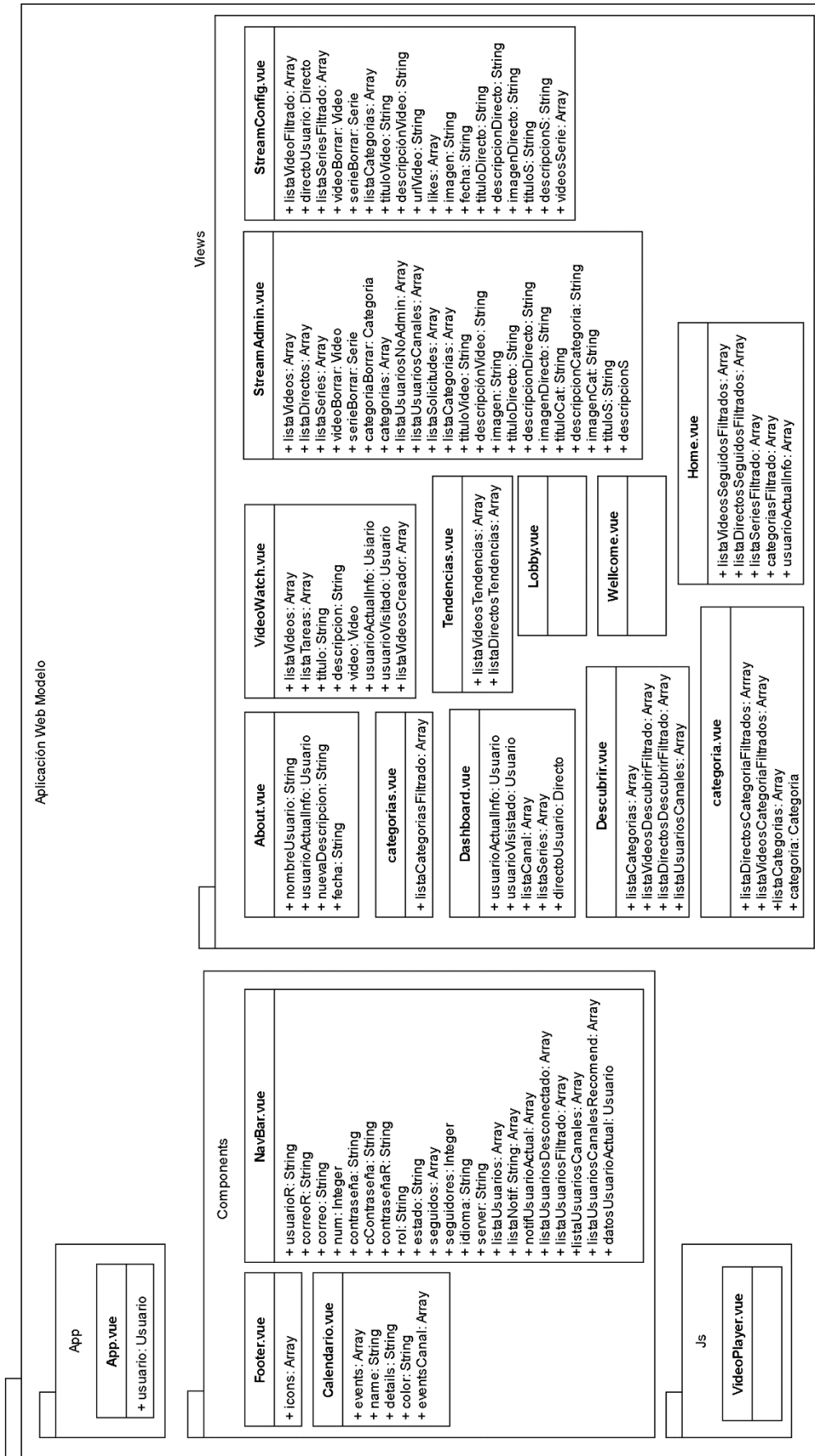


Figura 21: Modelo

Por otro lado, también es necesario el uso de las funciones firebase para el acceso a los datos estáticos de la base de datos. Esto se puede ver en la Figura 22.

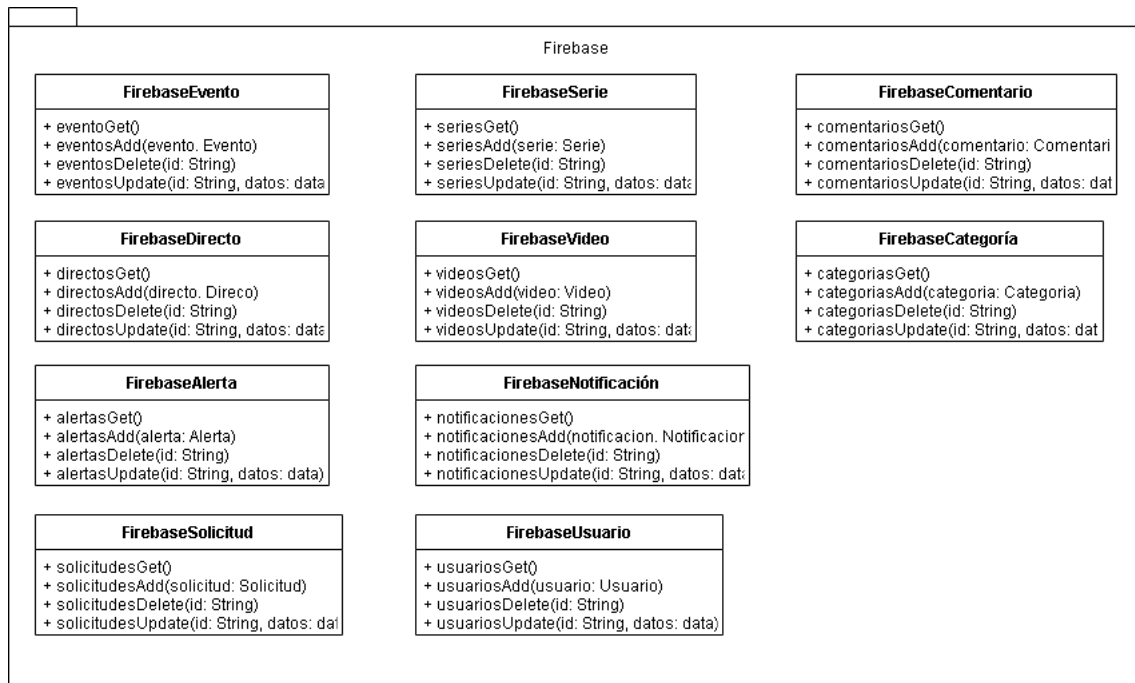


Figura 22: Modelo Firebase

#### 6.6.1.3.2. Backend CDN

Se trata del segundo subsistema, que es una API (Interfaz de programación de aplicaciones) para un servidor de almacenamiento Node soportado en una estructura UNIX, que es utilizado como backend para asegurar la localización de los archivos de vídeo. Este contiene un módulo que define las funcionalidades necesarias para ello, se presenta en un diagrama en la Figura 23.

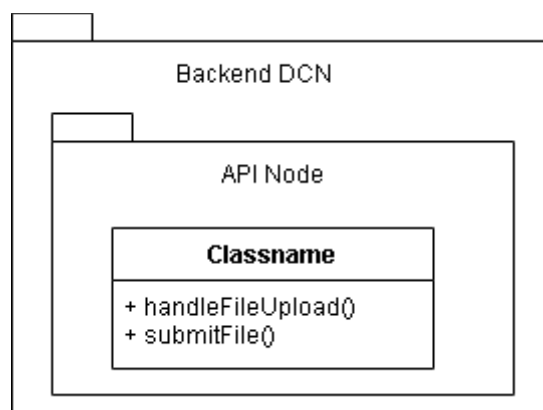


Figura 23: Backend CDN

#### 6.6.1.4. Vista arquitectónica

Mediante toda la información recopilada del diseño, especialmente las clases de diseño, se puede establecer una vista general de la arquitectura en base al patrón utilizado. Se ha elaborado un diagrama que se puede ver en la Figura 24.

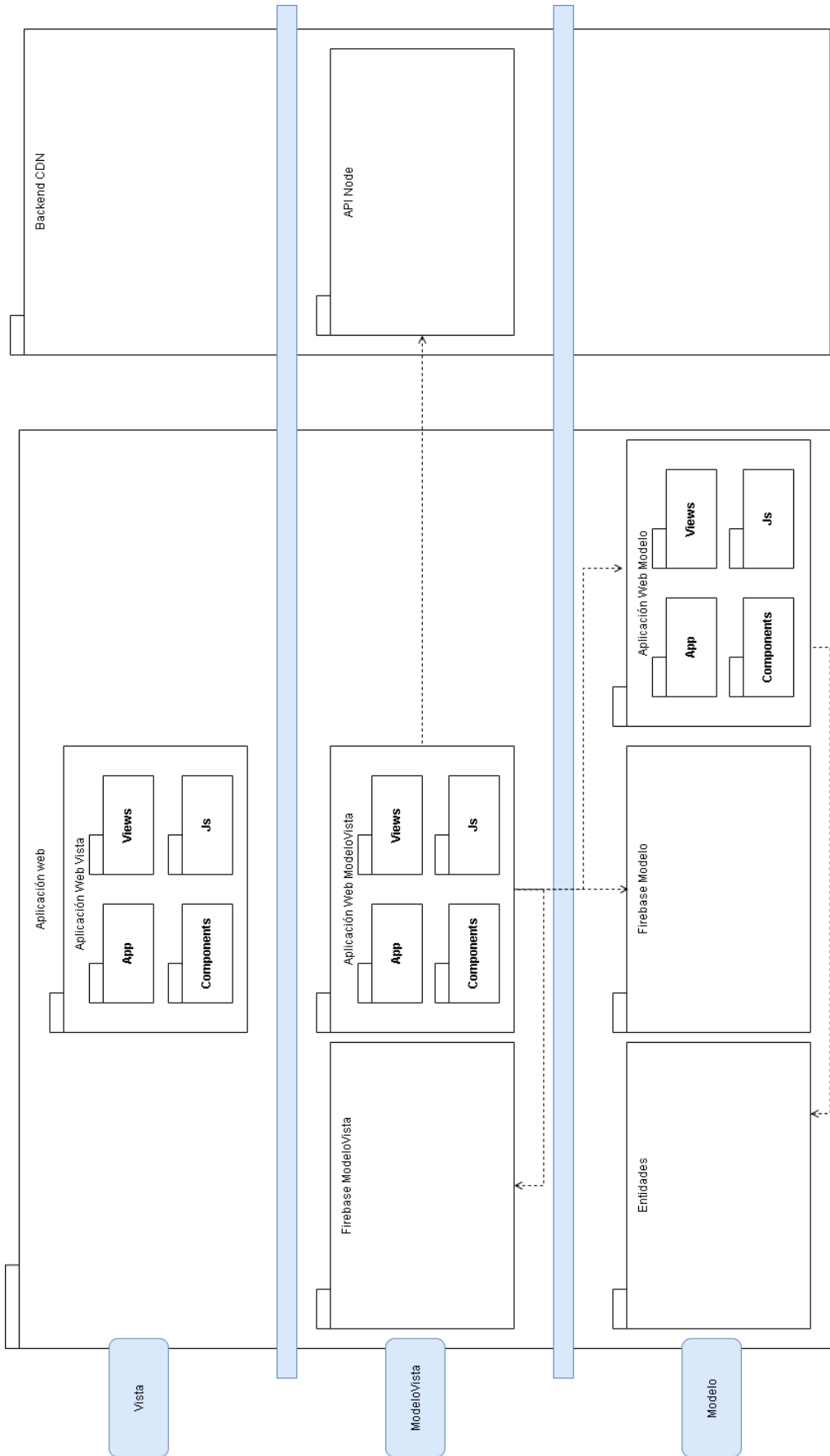


Figura 24: Arquitectura

### 6.6.1.5. Realización de los casos de uso modelo de diseño

En este punto, se ha realizado una revisión y afinamiento de los requisitos. Para ello, se representa la ejecución e interacción de los casos de uso en el modelo del diseño, con el uso de diagramas. Así se puede ver la interacción de los actores y las clases de diseño comentadas anteriormente, generando un mayor nivel de detalle.

Se ha realizado este proceso para cada uno de los 61 casos de uso. Esto se encuentra detallado en el **Anexo IV**. Aquí se muestra un ejemplo de uno de los diagramas resultantes en la Figura 25, en este caso, el registro.

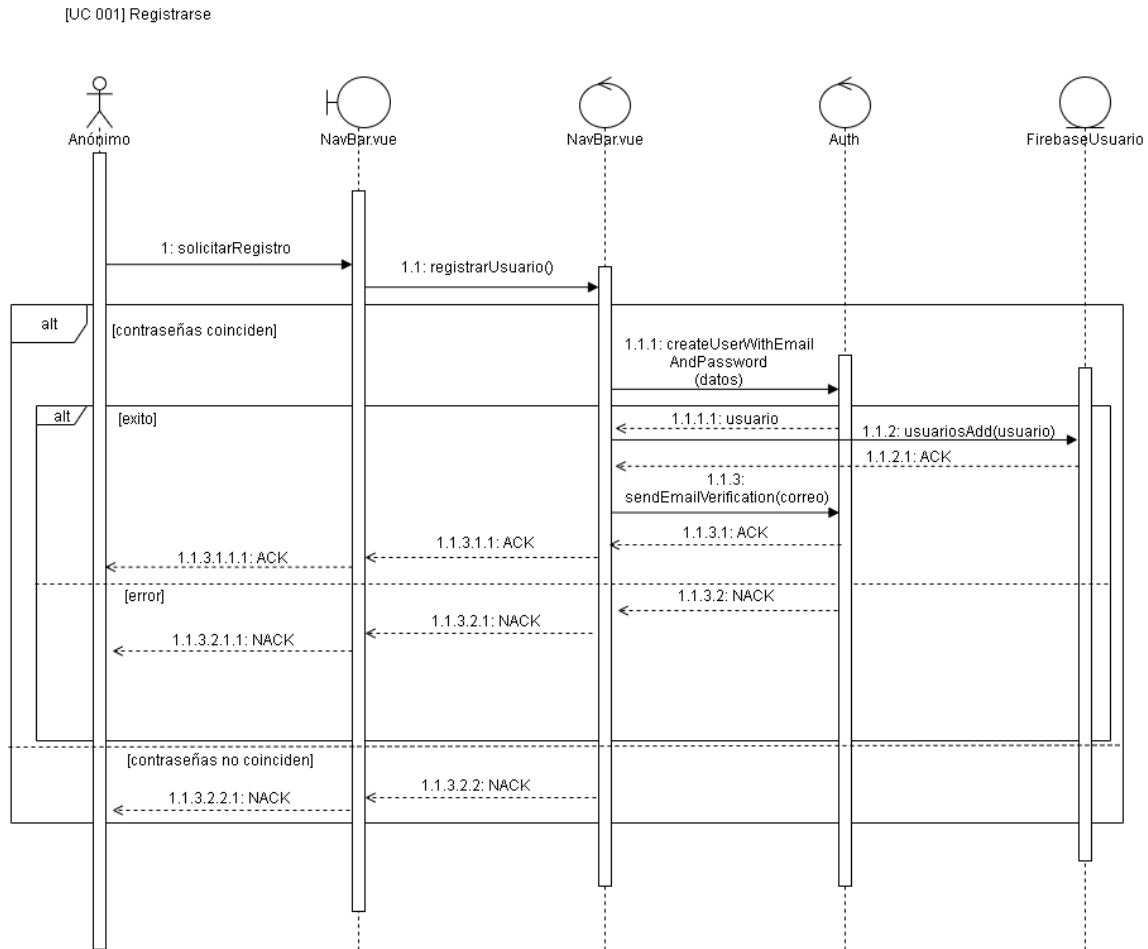


Figura 25: Diagrama de secuencia de caso de uso Registrarse UC-001

### 6.6.2. Diseño de la base de datos

Para el diseño, también se ha realizado una evolución del modelo del dominio, creando una solución en la que los datos se almacenan en una base de datos Firebase. Estas bases de datos no son tradicionales, sin embargo, se representa de una forma más tradicional para hacernos mejor a la idea, aunque discrepe un poco del posible resultado final. Se presenta en la Figura 26.

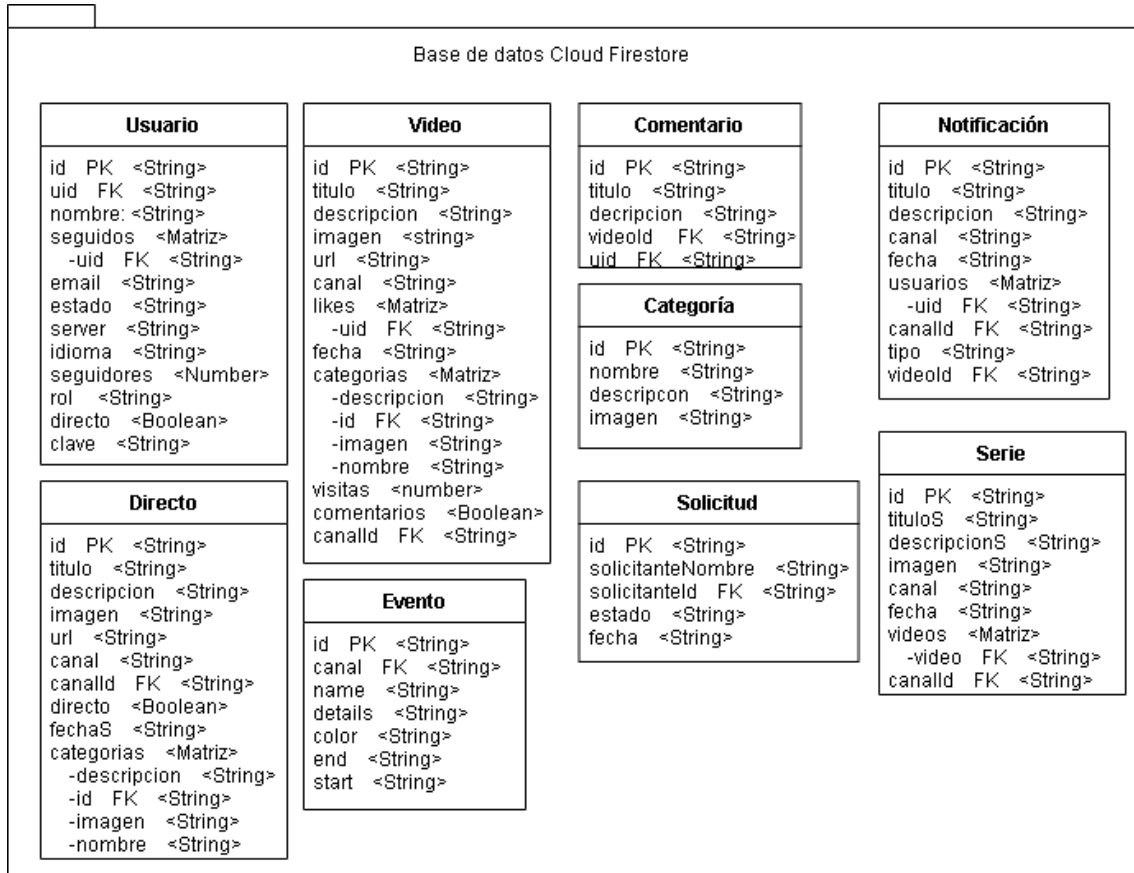


Figura 26: Firebase

### 6.6.3. Modelo de despliegue

Finalmente, en este apartado, nos centramos en desplegar el sistema . Para ello se han detallado los nodos que se desplegarán y sus relaciones. Estos nodos son los siguientes:

- **Ciente web:** Dispositivo con navegador web con el que los usuarios interactúan. Un artefacto en el cual se despliega la web. En él se pueden realizar todas las acciones del sistema por parte de todos los usuarios con distintos roles. Es el que se comunica con todos los demás nodos a través de internet.
- **Servidor Web:** Servidor en el que se encuentran todos los datos para el funcionamiento y despliegue de la aplicación web.
- **Servidor Firebase:** Servidor Firebase, que almacena casi todos los elementos relevantes del sistema menos los “.mp4”, y desde el que se realizan modificaciones, adiciones y eliminaciones, así como la autenticación de usuarios.
- **Servidor CDN:** Este es un servidor Node, que se encarga de recibir y almacenar los archivos “.mp4” que suben los canales de la plataforma.
- **Servidor de video:** Servidor de alto rendimiento NGINX, encargado de servir los videos y directos cuando recibe peticiones de los clientes, gracias a módulos **RTMP** y **HTTP**.

En la Figura 27, se presenta un esquema del modelo de despliegue, que contiene los nodos y la comunicación.

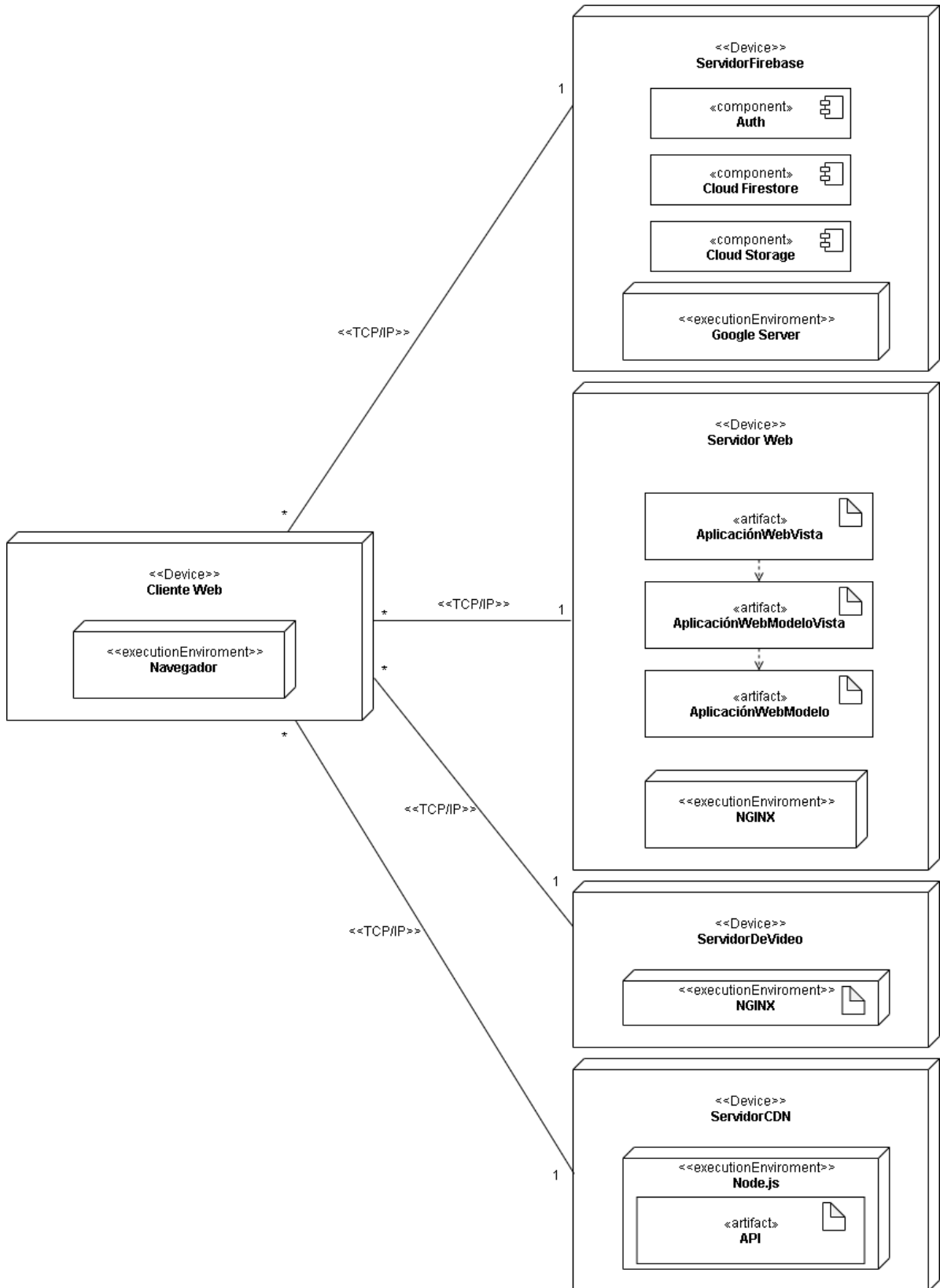


Figura 27: Despliegue



## 6.7. Implementación

En esta sección, nos centramos en la parte del desarrollo en la que se implementa el sistema. En nuestro proyecto, se ha tenido que implementar un prototipo funcional de la aplicación web, así como la implementación de los servidores en una distribución UNIX para hacer funcional el sistema. Esta parte, ha sido quizá la que más tiempo ha llevado debido al desconocimiento de muchas de las herramientas utilizadas y a la dificultad implícita.

Para la implementación, se han utilizado una serie de herramientas. Estas son las que ya han sido descritas en el apartado correspondiente de esta memoria, y por lo tanto, no repetiremos las explicaciones, aunque sí las volveremos a mencionar para indicar por qué se han elegido.

Por lo tanto, a continuación, trataremos los aspectos fundamentales del desarrollo del proyecto, pero antes, es importante comentar que para acceder al código, configuración o cualquier tipo de documentación técnica, es preferible hacerlo mediante el **Anexo V**.

### 6.7.1. Aplicación web

El elemento de aplicación web, es quizá el más importante del sistema, ya que es el nexo con el usuario, y por tanto la pieza central que une los distintos subsistemas.

Esta aplicación web, se ha desarrollado utilizando una serie de herramientas, aunque la principal es **Vue.js**. Se ha elegido vue.js como framework central, principalmente, por la gran cantidad de documentación disponible sobre este, lo que mejora en gran medida la adaptabilidad, algo muy necesario por el desconocimiento en este tipo de lenguajes. Además de esto, también se elige por su capacidad de escalado en base al uso de plantillas y bibliotecas, de forma que podamos centrarnos en la funcionalidad, y además manteniéndose siempre en un espacio en disco moderado.

También se han utilizado, junto con vue.js, otras tecnologías como **JavaScript**, para la funcionalidad, HTML Y CSS para la presentación, y **Firebase** como base de datos. JavaScript se ha elegido principalmente por lo bien que funciona con Vue, lo que se debe en parte a que este es multiplataforma, además de ser muy fácil de aprender. Por otro lado, Firebase se ha utilizado básicamente, porque gracias a que es Google el que se encarga de gestionar el servidor, nosotros podemos centrarnos en el desarrollo del frontend sin preocuparnos demasiado del backend, todo esto además de la facilidad para implementar reglas de seguridad, así como las copias de seguridad y su capacidad de escalado.

Además de estas tecnologías, se han usado una serie de bibliotecas en Vue.js, necesarias para un desarrollo más ágil y completo. Algunas de estas bibliotecas son:

- **video.js y vue-video-player**: Para la realización de un reproductor de video.
- **vue-scrollto**: Para la realización de direccionamiento a secciones de una misma página.
- **vue-router**: para el direccionamiento en la aplicación web.
- **Firebase**: Bibliotecas para gestionar acceso a los distintos servicios de Firebase, entre ellas, firebase/app, /firestore, /storage, /auth.

Se han utilizado otras bibliotecas, algunas de ellas mencionadas antes como Vuetify, Axios, o Vuex, y otras que vienen por defecto con vue o que no son tan importantes.

Posteriormente a la definición de los elementos que hemos utilizado, pasamos a ver algunos aspectos relevantes de la implementación del prototipo funcional de la aplicación web.

En primer lugar hay que comentar que con la información obtenida del análisis y antes de implementar, se realizó un **prototipo** previo en **Adobe XD**. Este prototipo se ha creado principalmente para hacerse a la idea de cómo serían los requisitos del sistema de una forma conceptual, para así evitar ir a ciegas, y conocer un poco las pantallas de este. Un prototipo de estas características puede usarse para hacer test de usuario en fases tempranas, para asegurarse de que se satisfacen todas las peticiones antes de implementar nada.

Remarcar también que la elección de colores, disposición de elementos y tipografía, se ha realizado en base distintas pruebas con paletas de color, plantillas de texto, dibujos en papel y criterios personales, eligiendo aquellos elementos que mejor representaban al sistema.

Se puede observar una de las pantallas del prototipo no funcional en la Figura 28, en este caso, la página del video.

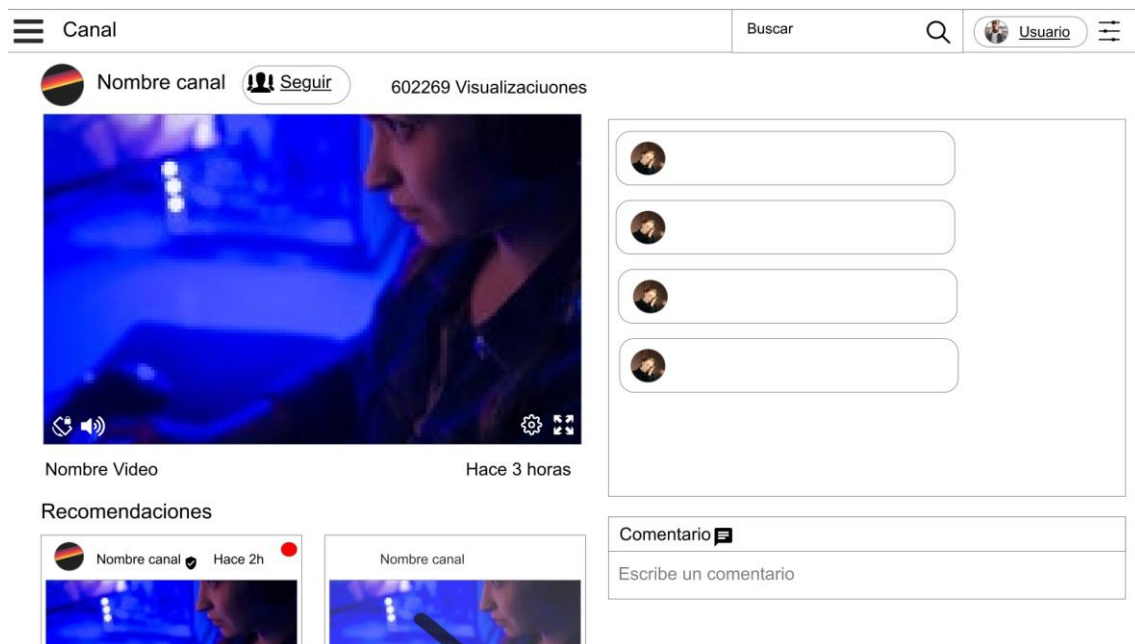


Figura 28: Video Prototipo

El resto del desarrollo se ha realizado de forma incremental, de modo que se ha empezado por las funcionalidades básicas, acabando en los elementos más complicados como por ejemplo, la administración. De esta manera, mediante las pruebas de las funcionalidades básicas, se han generado en el sistema y en la base de datos, elementos con los que trabajar en las tareas más complejas. Gracias al prototipo en Adobe, se han podido ir tomando elementos necesarios de Vuetify para ir completando los diseños, a la vez que se iban estableciendo sus funcionalidades.

Se muestran algunos de los resultados del desarrollo en la Figura 29, en este caso, correspondiente a la página del video.

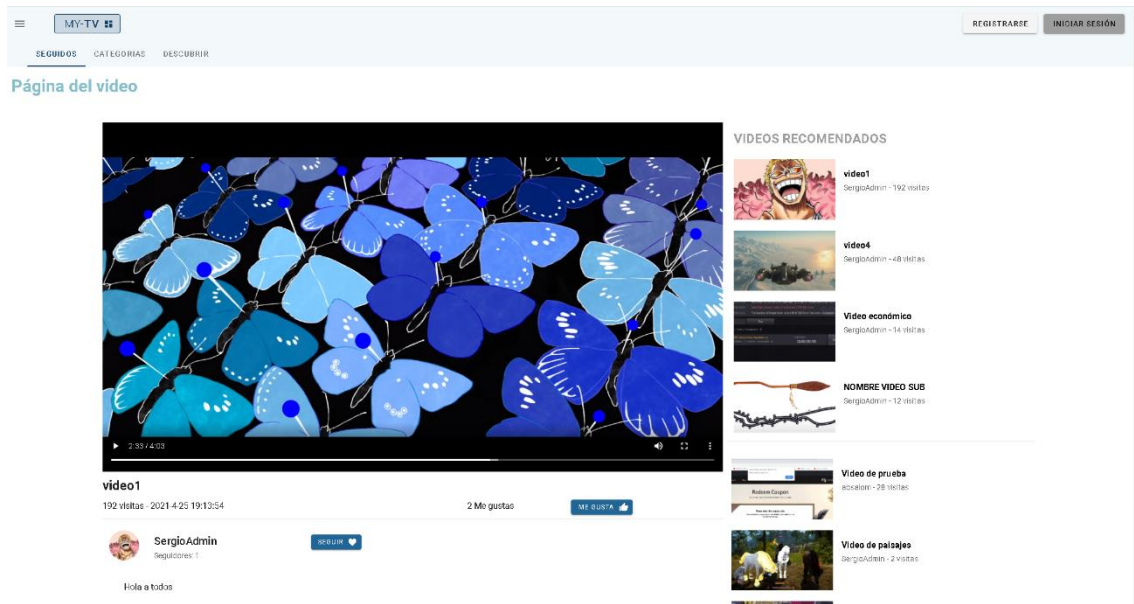


Figura 29: Video

Como se ha comentado antes, también es fundamental la implementación de una API integrada con la aplicación web para la gestión de los elementos de la base de datos. Esta se basa principalmente en la subida, recuperación y modificación de datos del Firestore y Storage de Firebase. Por ejemplo, en la Figura 30, se muestra la modificación de un video de Firestore.

```
import { db } from "../firebase";

await db.collection("videos").doc(this.selectedTarea.id).update({
  titulo: this.titulo,
  descripcion: this.descripcion,
  imagen: this.imagendesc,
  categorias: this.categoriasVideo,
  comentarios: this.checkbox,
});
```

Figura 30: Collection Update

En cuanto a la autenticación, ocurre lo mismo, pero utilizando funciones del módulo Auth de Firebase. Un ejemplo de esta conexión a Firebase, se puede ver en la Figura 31, en el caso del acceso a Auth para iniciar sesión.

```
firebase
  .auth()
  .signInWithEmailAndPassword(this.correo, this.contraseña)
  .then((user) => {
    this.$router.push({ name: "Tendencias" }).catch((err) => {});
    this.correo = "";
    this.contraseña = "";
  })
```

Figura 31: Auth

Adicionalmente, hay que comentar, que se han tenido que establecer reglas de seguridad para Firebase, de forma que dependiendo del rol o autenticación, se puedan realizar unas u otras acciones.

### 6.7.2. Backend CDN

Este Backend, es un servidor **Node** para la subida de videos a un sistema UNIX en el que se encuentra el servidor de video.

En cuanto al desarrollo de este, se han tenido que utilizar algunas herramientas relacionadas con la ejecución de servidores. La herramienta principal en este aspecto es el entorno de ejecución Node.js, que se ha elegido principalmente porque al usar JavaScript, ya no se tiene que aprender un lenguaje nuevo desde cero, debido a que ya lo usamos en Vue. También se ha escogido porque es capaz de atender a múltiples peticiones mediante eventos, lo que facilita la concurrencia en las peticiones de subida, además de la facilidad de instalación de paquetes con npm.

La implementación de este ha consistido en la configuración del servidor Node mediante el uso de 3 módulos o paquetes, instalados con **npm**. Estos 3 módulos son:

- **HTTP**: Protocolo de comunicación para la transferencia de información.
- **fs**: Utilizado para almacenar los vídeos subidos a una localización específica.
- **Formidable**: Usado para el análisis de los archivos subidos.

Posteriormente, se ha utilizado **pm2** para la gestión y monitorización del servidor Node, de forma que este pueda iniciarse solo y realizar ajustes de carga.

Por otro lado, en la parte del cliente, se utiliza la biblioteca Axios para la realización de peticiones, como ya se comentó anteriormente. Un ejemplo del uso de algunos de estos elementos se encuentra en la Figura 32, en el caso de Axios.

```
axios
  .post(this.urlUpload, formData, {
    headers: {
      "Content-Type": "multipart/form-data",
    },
  })
  .then(function () {
    console.log("Exito");
  })
```

Figura 32: Axios

### 6.7.3. Servidor de video

Finalmente, se ha desarrollado el servidor de video. Como ya se había comentado, se utiliza un **servidor de alto rendimiento Nginx**. Este se ha escogido fundamentalmente porque, aparte de ser multiplataforma, es muy apto para el streaming de archivos “.mp4”, permitiendo soportar una carga relatante elevada, gracias a que utiliza menos memoria (Utiliza compresión), lo que permite también atender a más solicitudes, aunque esto disminuye su flexibilidad.

Lo que se ha tenido que implementar en este caso, ha sido la configuración de la transmisión de video, mediante el uso de dos módulos.

En primer lugar, se ha añadido un módulo de **HTTP** que establece los puertos en los que se prestan los servicios, así como las localizaciones de carpetas donde se almacenan los videos y los trozos de los directos.

Después se ha configurado un **módulo RTMP** para establecer el servicio de livestream al que se aplican algunas configuraciones. Un ejemplo de esto se puede observar en la Figura 33, en este caso, módulo RTMP para el directo.

```
rtmp {
    server {
        listen 1935;
        chunk_size 4000;

        application livestream {
            live on;
            #record off;
            hls on;
            hls_path /etc/nginx/temp/livestream;
            hls_fragment 3;
            hls_playlist_length 60;
            deny play all;
        }
    }
}
```

Figura 33: RTMP

## 6.8. Pruebas

Una vez finalizado el sistema, y durante su desarrollo, se han ido realizando una serie de pruebas, que pretenden determinar si el sistema se comporta correctamente, y en definitiva si cumple con los objetivos propuestos. Estas pruebas, también sirven para saber si ha habido errores en el desarrollo, de forma que se puedan corregir o eliminar.

Para ello, se han organizado baterías de pruebas de distinta índole, que se han aplicado en las distintas etapas del desarrollo. Estas pruebas, han sido principalmente de tres tipos:

- **Unitarias:** Cada vez que se finaliza la creación de un componente, y durante el desarrollo de este, se realizan pruebas de distintos tipos para comprobar sus posibilidades de funcionamiento, y así poder detectar posibles errores y corregirlos durante su desarrollo y antes de pasar al siguiente componente
- **Integración:** A medida que se han ido implementando los distintos componentes, se han probado de manera conjunta, de forma que se garantizase su funcionamiento combinado, y así solucionar los posibles errores que pudiesen surgir en la interacción.
- **Sistema completo:** Finalmente, una vez que se ha desarrollado todo el sistema, se han realizado pruebas completas de todas las funcionalidades del sistema para asegurar el correcto desarrollo.

## 6.9. Funcionalidades

Una vez explicada la implementación, se muestran algunas de las funcionalidades más importantes del sistema. Esto se realizará categorizando por las acciones de los diferentes usuarios en función de sus roles.

Para acceder a la información detallada sobre las funcionalidades e interfaces del sistema, se puede consultar el **Anexo VI**.

### 6.9.1. Usuario anónimo

Se trata del usuario que no se ha registrado o que no ha iniciado sesión. Será el que menos interacciones puede tener con el sistema, aunque puede acceder a todos los contenidos y tiene también acceso a acciones relacionadas con la autenticación.

#### 6.9.1.1. Barra de opciones anónimo

Es una barra que muestra algunas de las utilidades más importantes de la aplicación. La barra está disponible para todos los tipos de usuarios, solo que con distintas opciones.



Figura 34: Barra de aplicación Anónimo

En el caso del usuario Anónimo, la barra muestra las opciones de **iniciar sesión** y **registrarse**, Figura 34.

#### 6.9.1.2. Autenticación de usuarios

Se ofrece al usuario anónimo, la opción de **iniciar sesión** mediante una interfaz sencilla con 2 campos y un botón en el que se indica el inicio de sesión, tal como se muestra en la Figura 35.

Figura 35: Inicio de sesión

Se ofrece al usuario Anónimo una interfaz para **registrar una cuenta**, esta contiene un cuestionario y un botón para el registro, como el que ve en la Figura 36.

The screenshot shows a registration form with the following fields and buttons:

- Header: **Registro de usuario**
- Text: Introduce tus datos
- Form fields: Nombre de usuario, Contraseña, Confirmar contraseña, Correo electrónico
- Buttons: **REGISTRARSE** (dark blue), **REGRESAR** (light blue)

Figura 36: Registrarse

En caso de que el registro sea correcto, el usuario es redirigido al menú principal de la aplicación y para completar el registro, se manda un enlace de verificación al correo, como se ve en la Figura 37.



Figura 37: Correo Verificación

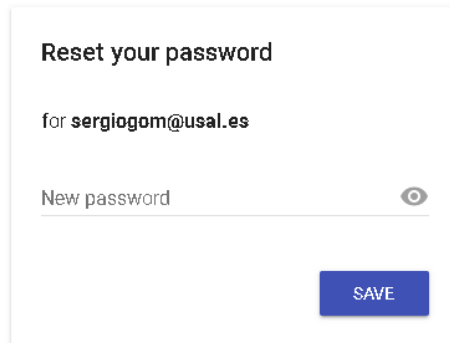
Si el usuario no puede iniciar sesión porque no recuerda su contraseña, tiene la opción de **recuperar la contraseña**. Para ello ha de aportar el correo sobre el que está registrada su cuenta. Se puede ver en la Figura 38.

The screenshot shows a password recovery form with the following fields and buttons:

- Header: **Recuperación de contraseña**
- Text: Introduce tus correo electrónico para que mandemos un enlace de recuperación
- Form field: Correo electrónico
- Buttons: **ENVIAR CORREO DE RECUPERACIÓN** (dark blue), **REGRESAR** (light blue)

Figura 38: Recuperación de contraseña

Si el correo es válido, se manda un email, que si abre, redirige a una ventana de la aplicación donde puede establecer una nueva contraseña, Figura 39.



Reset your password

for **sergiogom@usal.es**

New password

SAVE

Figura 39: Formulario contraseña

### 6.9.1.3. Páginas de la aplicación anónimo

El usuario Anónimo, aunque no tenga acceso a ciertas opciones, sí que tiene acceso a todo el contenido del sistema. Por ello, se le presentan mediante una serie de interfaces esos contenidos. Estas interfaces, son a las que pueden acceder todos los usuarios, independientemente de su rol, y son expositores de contenido.

Aunque existen muchas páginas de contenido con distintas temáticas, vemos las más relevantes, por ejemplo en la Figura 40, se puede ver la página de descubrir, que tiene un **expositor de directos** para cada canal activo, además de las listas de directos y videos de la plataforma. También presenta un filtro avanzado por etiquetas y nombre.

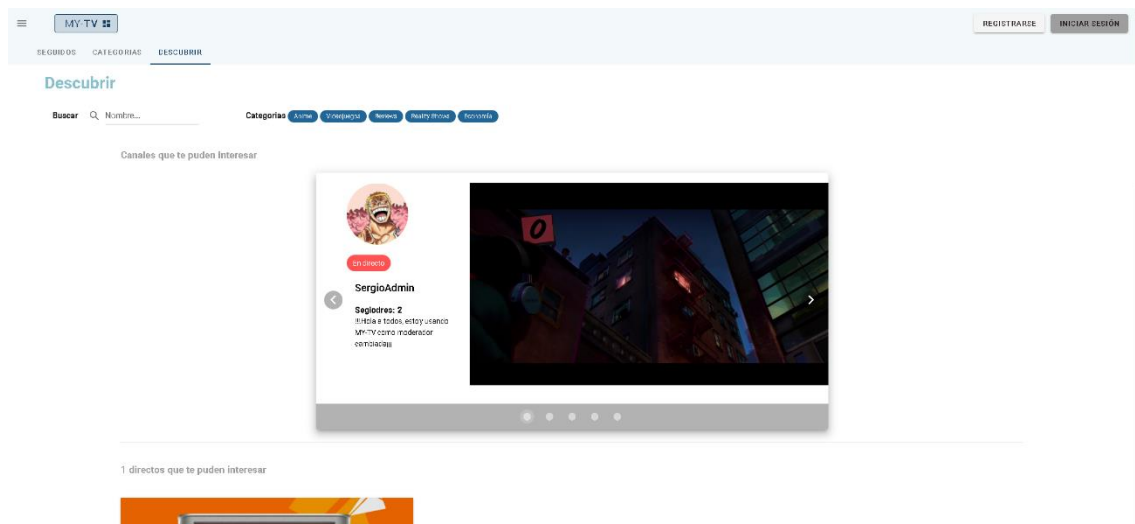


Figura 40: Descubrir

En la Figura 41, se puede ver la página de categorías, que muestra las **categorías** de la plataforma, con sus descripciones, además de un filtro de búsqueda por nombre.



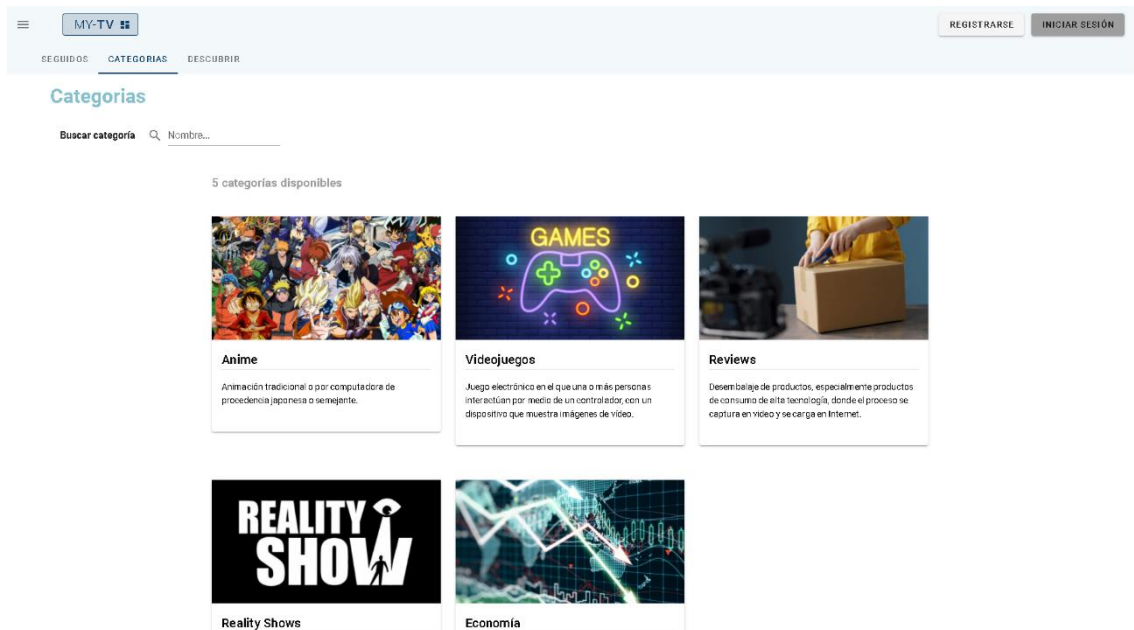


Figura 41: Categorías

Por otro lado, en la Figura 42, se puede ver una de estas categorías con todos los **directos y videos** disponibles en ella, así como un nuevo buscador.

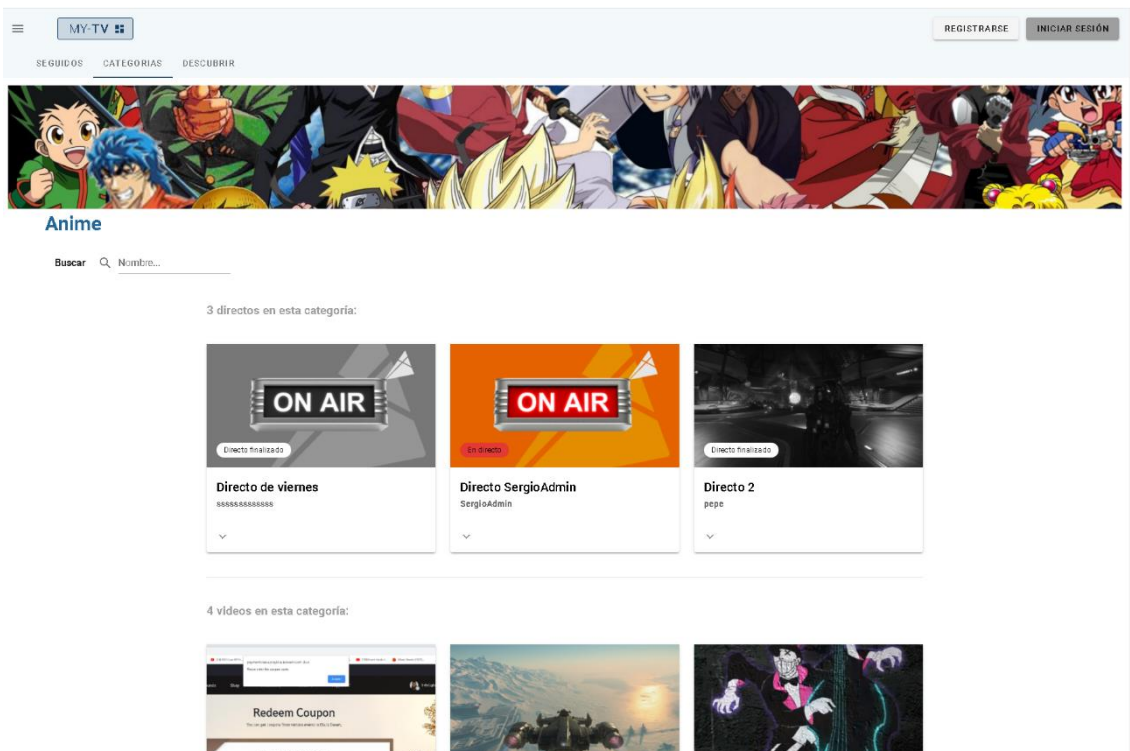


Figura 42: Categoría

En todas las páginas anteriores, en las cartas que muestran los videos, los directos y categorías, se puede desplegar para mostrar sus descripciones y etiquetas. Si presionamos sobre una etiqueta de categoría, iríamos directamente a la página de esta, lo mismo sucede con los videos y directos.

Por supuesto, existe la **página del video**, que es la que contiene el reproductor de video. Se puede ver en la Figura 43.

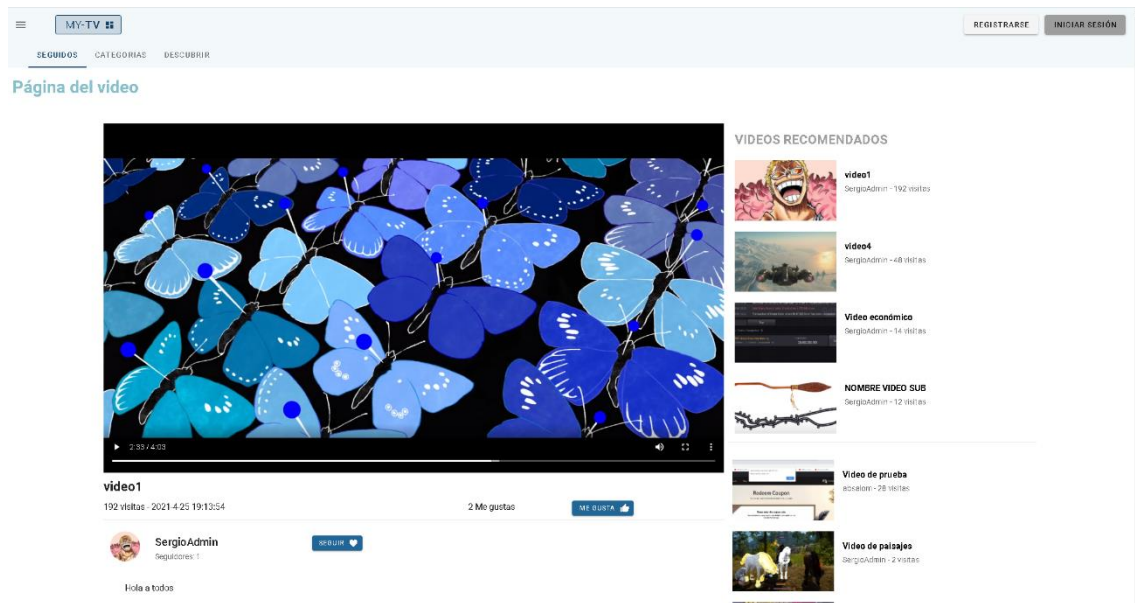


Figura 43: Página del Video 1

Justo debajo del reproductor aparecen los detalles del video y una sección destinada al canal, que sirve para acceder a este. Igualmente, se puede observar un botón que sirve para **dar me gusta** y otro que sirve para **seguir al canal**. A la derecha, también aparece una lista de videos recomendados, siendo algunos del canal del video y el resto de otros canales.

Más abajo, aparece la sección de comentarios, Figura 44, en la que se muestra una caja de texto donde podemos **escribir un comentario y añadirlo**.

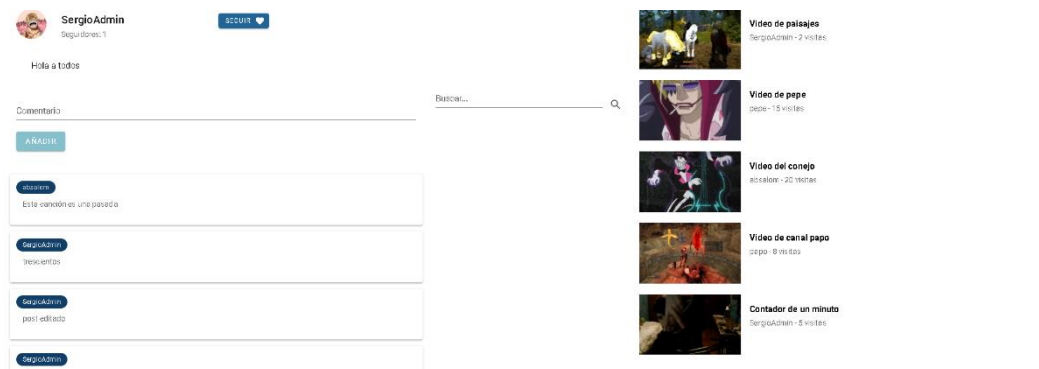


Figura 44: Página del vídeo 2

El usuario anónimo, sólo puede visualizar los comentarios, si intenta comentar, dar me gusta o seguir, saltará la ventana de inicio de sesión. Algunos usuarios con otros roles tienen permisos sobre estos comentarios, pudiendo añadirlos, modificarlos y eliminarlos.

Después, está la **página del Directo**, que es la página en la que se reproduce el directo de un determinado canal, se puede ver en la Figura 45. Esta página, da información acerca del directo y del canal, ofreciendo también la opción de seguirlo.

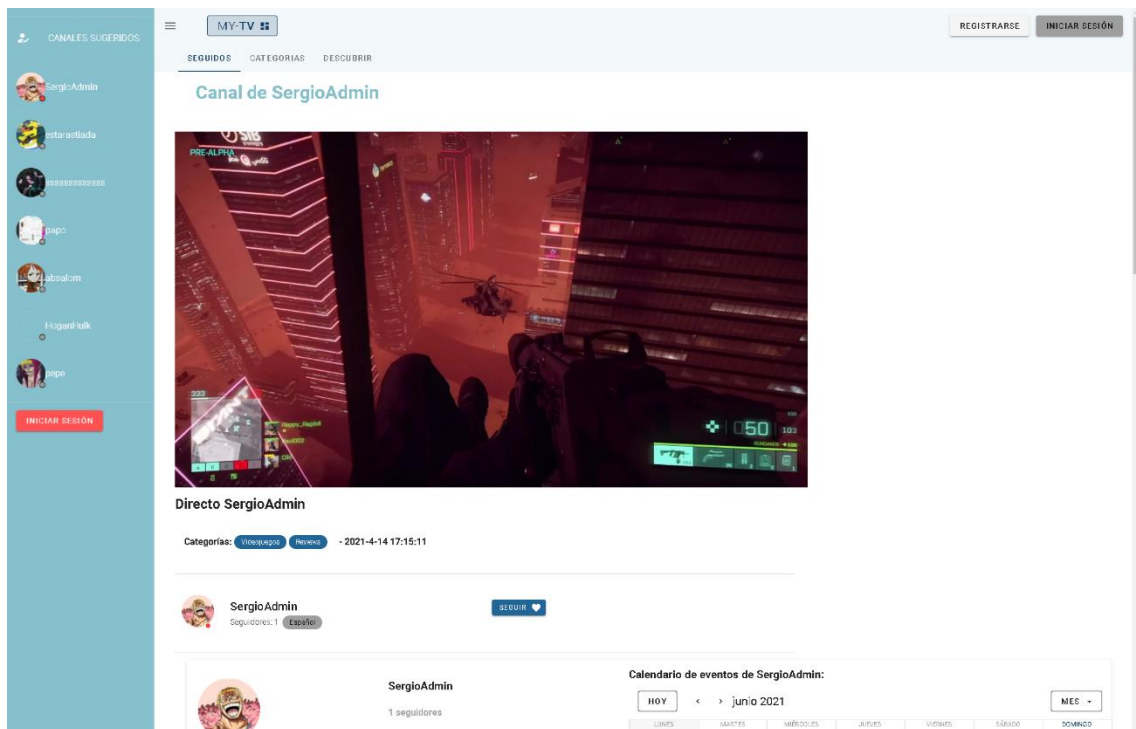


Figura 45: Página del directo 1

En la sección media de la pantalla del directo, Figura 46, se encuentra el **calendario del canal**, conteniendo los eventos programados por este. Se puede acceder a estos eventos para observar su descripción.



Figura 46: Página del directo 2

En la página del canal también aparece una **lista de videos y series**, que contiene todos los contenidos del canal. Si se despliega una serie se muestran sus videos.

#### 6.9.1.4. Menú lateral anónimo

Este menú existe para todos los usuarios, pero en él, aparecen una serie de elementos que son distintos dependiendo del rol. En el caso del usuario Anónimo, Figura 47, se muestra en la parte inferior un botón para iniciar sesión, y en la superior, un tag que indica canales recomendados.

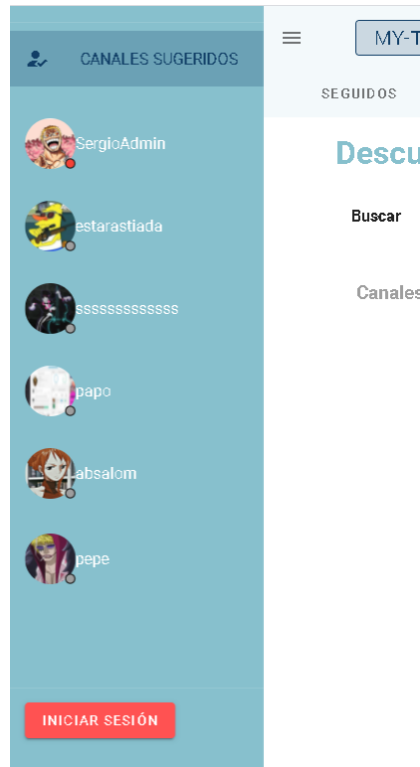


Figura 47: Menú lateral de navegación Anónimo

#### 6.9.2. Usuario autenticado

Se trata del usuario que tiene sesión iniciada, y por lo tanto, puede llevar a cabo acciones como dar me gusta, comentar o seguir, modificar sus datos desde su perfil, entre otras.

##### 6.9.2.1. Barra de aplicación autenticado

Se trata de la misma barra de la que ya habíamos hablado antes, solo que en esta ocasión cuenta con dos iconos adicionales, tal como se puede ver en la Figura 48.

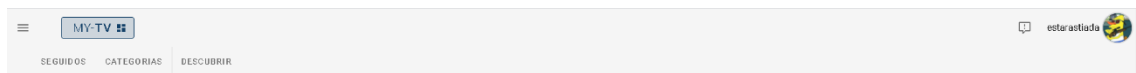


Figura 48: Barra de aplicación usuario Autenticado

El icono, con la imagen del usuario, abre un menú desplegable que muestra datos del usuario, además de una serie de opciones, tales como **cerrar sesión** o **acceder al perfil**. Se puede observar en la Figura 49.

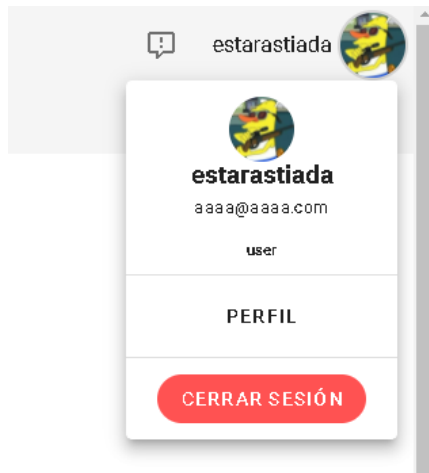


Figura 49: Desplegable Usuario

El otro icono, sirve para **listar las notificaciones** del usuario, que podrá **eliminar** o utilizar para llegar al contenido que anuncia.

Podemos por tanto, **acceder al perfil**. Este perfil, se ve en las Figuras 50 y 51.

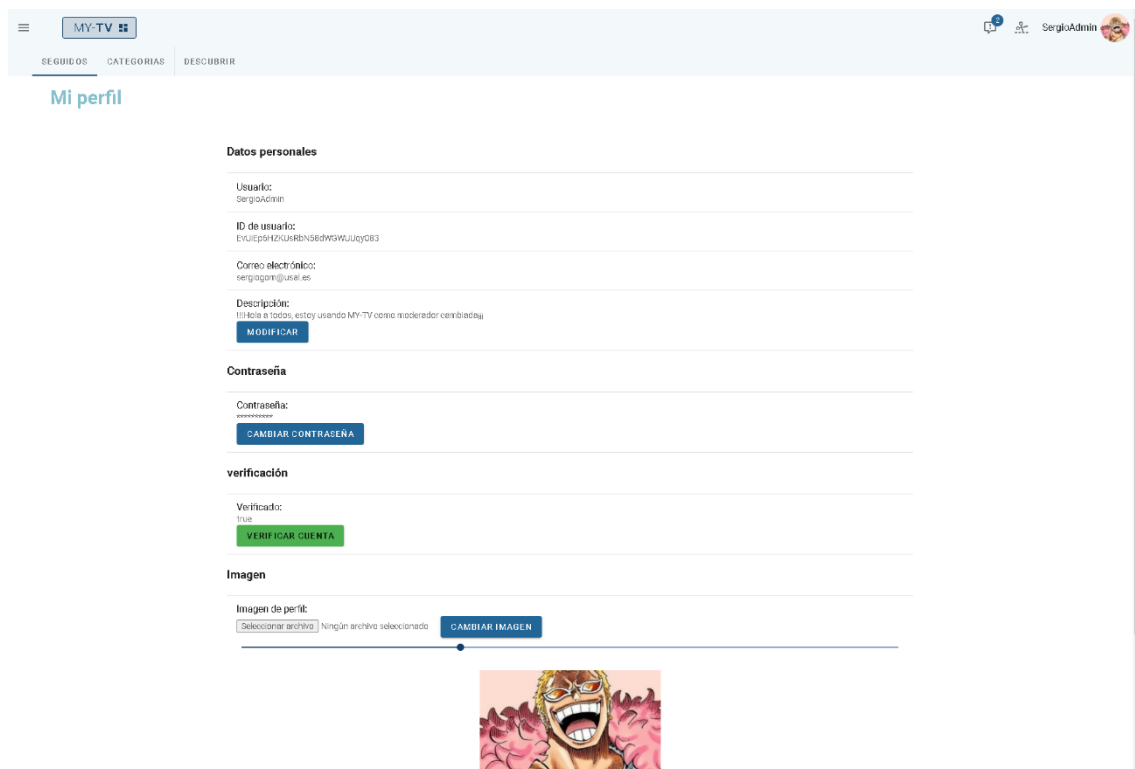


Figura 50: Perfil 1

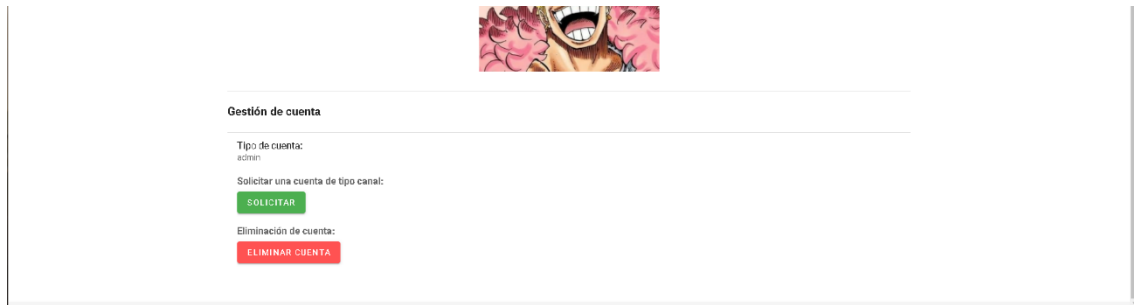


Figura 51: Perfil 2

### 6.9.2.2. Opciones de usuario autenticado

Hay una serie de opciones a las que se puede acceder desde el perfil visto anteriormente y que puede realizar el usuario autenticado las vemos a continuación:

- El usuario puede **cambiar la descripción**, para esto, se abre una pestaña que contiene una caja de texto en la que se presenta la descripción antigua, que se puede modificar.
- El usuario puede **verificar su cuenta**, para ello, el sistema presenta una interfaz en forma de ventana con un botón, que si el usuario presiona, el sistema envía un correo que contiene un mensaje personalizado con un enlace para verificar la cuenta.
- El usuario puede **cambiar su contraseña**, para ello, se ofrece una interfaz con un botón que si se presiona, el sistema manda un correo al usuario, que contiene un mensaje para cambiar la contraseña.
- El usuario también tiene una opción para **cambiar la imagen de perfil**, con un botón para subir la imagen desde el explorador de archivos, y otro para cambiarla.
- Finalmente, la opción para **eliminar la cuenta**, para la que se muestra una pestaña, con un campo de texto para escribir el nombre de usuario y un botón para aceptar la eliminación. Si el nombre coincide se elimina la cuenta y sus resto del sistema.

### 6.9.2.3. Seguimiento y me gusta

Los usuarios autenticados tienen la posibilidad de acceder a la sección de **seguidos**, que contiene los videos y directos de los canales a los que siguen. Por lo tanto, estos, en la páginas de un video o en la de un directo, pueden **Seguir y dejar de seguir** a usuarios canales, tal y como se puede ver en las Figuras 52 y 53.



Figura 52: Seguir



Figura 53: Dejar de seguir

Lo mismo sucede para **dar y quitar me gusta**, pero desde las páginas de un video únicamente. En estas páginas de video, los usuarios autenticados, además, pueden **escribir comentarios**, y **modificarlos** si lo desean, así como **eliminarlos**.

#### 6.9.2.4. Menú lateral autenticado

Es el mismo componente que en el caso del anónimo, con la excepción de que para los usuarios autenticados, aparecen los canales seguidos y no hay inicio de sesión. Se ve en la Figura 54.

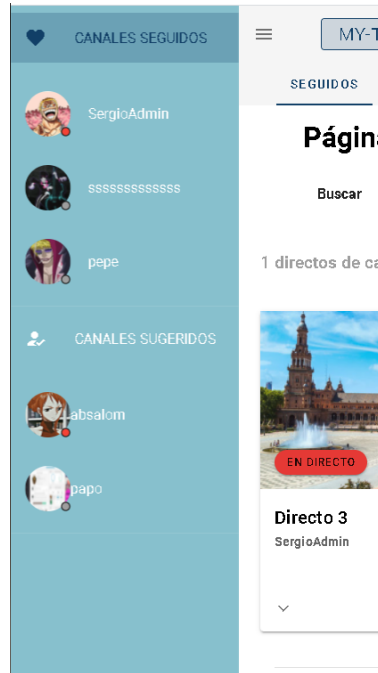


Figura 54: Menú lateral Autenticado

#### 6.9.3. Usuario canal

Este usuario puede subir y transmitir contenido en la plataforma, por lo tanto, realizará modificaciones en los datos del sistema, aunque solo sobre sus propios contenidos.

##### 6.9.3.1. Barra aplicación canal

La barra de aplicación del usuario canal contiene un nuevo icono para acceder a la configuración de su canal, que se ve en la Figura 55.

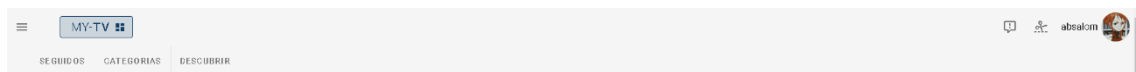


Figura 55: Barra de aplicación canal

Este icono también aparece en el desplegable del avatar del usuario.

##### 6.9.3.2. Configuración de canal

El usuario canal, puede llevar a cabo toda su actividad de gestión de contenido en la pantalla de configuración, accesible desde el botón de la barra de aplicación. Esta página presenta numerosas opciones al usuario, que veremos poco a poco.

En primer lugar, en la Figura 56, se ve el **menú de acceso rápido**, que nos desplaza por las secciones de la página. En esta misma figura, está la opción de configuración de directo, en la que aparecen los datos de transmisión, entre ellos, la **clave de transmisión**, que se puede revelar. Así como el botón para **iniciar** y **finalizar** el directo, junto con un botón que revela un desplegable para **editar el directo**.

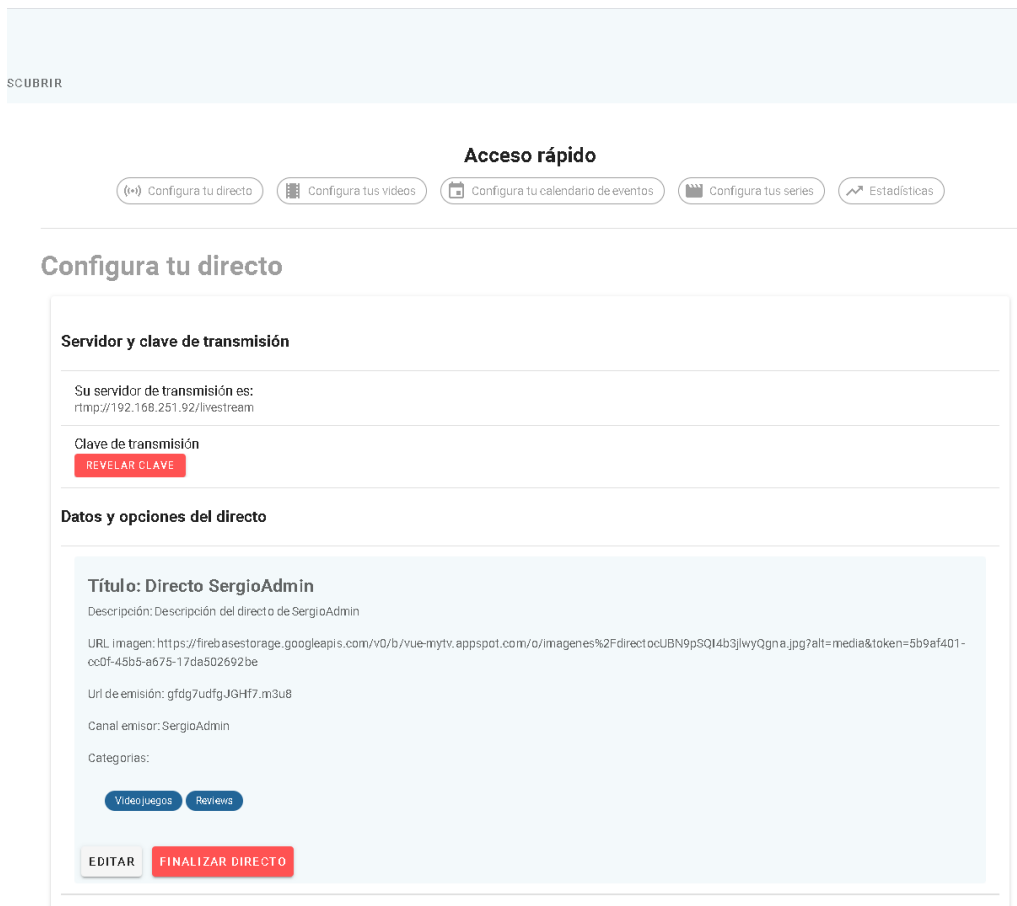












Figura 56: Configuración 1

Posteriormente, en la Figura 57, se encuentra la sección de configuración de videos, en la que aparece el formulario para **subir un video**, así como la información de los videos actuales del canal, que se ven en una lista de contenido. En la derecha de cada video de la lista hay dos opciones, una con un lapicero, que sirve para **editar**, y la otra con una papelera para **borrar**. En caso de editar, se despliega un nuevo formulario sustituyendo al de subir video, que presenta los datos del video a modificar.



Configura tus videos

**Tus Videos**

Título	Canal	Vistas	Likes	Fecha	Comentarios	Categorías	Acciones
video1	SergioAdmin	153	2	2021-4-25 19:13:54	<input checked="" type="checkbox"/>	1	 
video4	SergioAdmin	48	3	2021-4-25 19:15:30	<input type="checkbox"/>	2	 
Video económico	SergioAdmin	14	1	2021-5-4 0:32:54	<input checked="" type="checkbox"/>	3	 
NOMBRE VIDEO SUB	SergioAdmin	12	1	2021-5-22 16:28:3	<input checked="" type="checkbox"/>	2	 
Contador de un minuto	SergioAdmin	5	0	2021-6-28 22:50:41	<input type="checkbox"/>	1	 

Items por page: 5 1-5 of 6 < >

---

**Subir videos**

Título del video

---

Descripción

---

Miniatura del video  
 Ningún archivo seleccionado

Subir archivo de video  
 Ningún archivo seleccionado

Categorías de su video: 0

Seleccione las categorías con las que coincide su video: 5

Desmarque esta casilla si desea desactivar los comentarios

GUARDAR VIDEO

Figura 57: Configuración 2

Posteriormente, está la **configuración de eventos**, Figura 58, en la que aparece el calendario del canal. Si pulsamos sobre un evento, este se abre, pudiendo **editar**lo o **eliminar**lo. También se pueden **añadir** eventos con el formulario desplegable, accesible desde la opción agregar.

< > junio 2021

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
31	Jun 1	2	3	4	5	6
	Evento táctico					
7	8	9	10	11	12	13
Evento táctico						
14	15	16	17	18	19	20
21	22	23	24	26	26	27
28	29	30	Jul 1	2	3	4
Evento del canal						

Figura 58: Configuración 3

Luego, está el apartado de **configuración de series**, Figura 59. Este, contiene un formulario para **añadir una serie**, así como una lista con las series del canal, con opciones para **eliminar** y **editar**.

### Tus Series

	Título	Canal	Videos	Fecha	Acciones
▼	Serie de videos	SergioAdmin	3	2021-6-3 1:56:55	✎ 🗑
▼	Serie 2 del canal	SergioAdmin	2	2021-6-27 0:5:12	✎ 🗑

Rows per page: 5 1-2 of 2 < >

---

### Añadir serie

Título de la serie

Descripción de la serie

Videos de la serie: 0

Videos que puede añadir: 6

video1
video4
Video económico
NOMBRE VIDEO SUB
Contador de un minuto
Video de paisajes

AÑADIR SERIE

Figura 59: Configuración 4

Finalmente, está la sección de **estadísticas**, Figura 60, que muestra los visitantes del último directo del canal, y los datos de los videos más importantes del canal en función de sus acceso.

### Último directo

Nombre	Visitas
Directo SergioAdmin	4

---

### Tus mejores videos

Título	Visitas	Fecha
video1	193	2021-6-25 19:13:54
video4	48	2021-6-25 19:15:30
Video económico	14	2021-5-4 0:32:54
NOMBRE VIDEO SUB	12	2021-5-22 15:28:3
Contador de un minuto	5	2021-6-26 22:50:41
Video de paisajes	2	2021-6-26 23:3:7

Figura 60: Configuración 5

#### 6.9.4. Usuario Administrador

Es el usuario con más permisos en el sistema. El administrador, puede moderar a otros, por lo tanto, puede realizar modificaciones en todos los datos del sistema, tanto sobre sus contenidos como sobre los contenidos de los demás.

##### 6.9.4.1. Barra de aplicación administrador

La barra de aplicación del usuario administrador, en su desplegable del avatar, contiene la opción para acceder a la página de administración. Como se ve en la Figura 61.



Figura 61: Barra de aplicación desplegable administrador

##### 6.9.4.2. Administración

El usuario administrador, puede desempeñar sus actividades de control sobre el resto de los usuarios del sistema en esta página. Con el fin de facilitar la gestión, como se puede ver en la Figura 62, solo se muestra una lista de desplegables con las distintas opciones de moderación, que se irán viendo estas opciones a continuación.

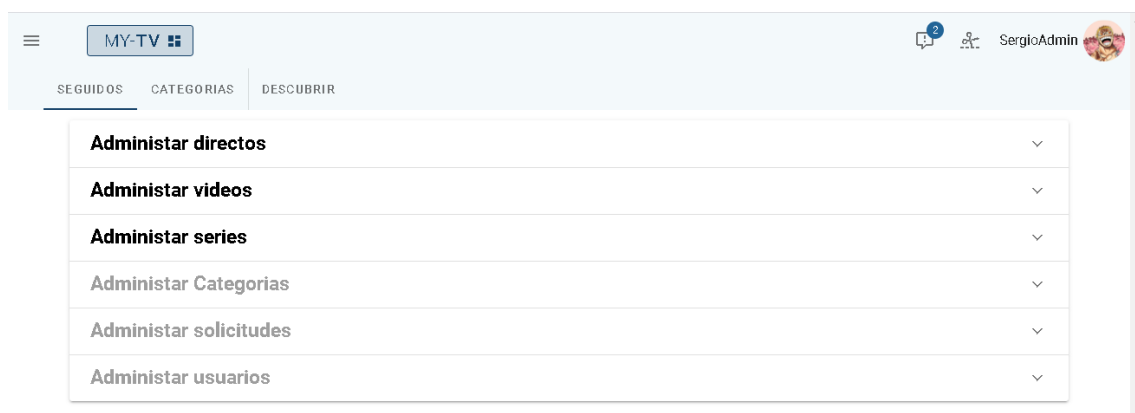


Figura 62: Administración 1

En primer lugar, en la Figura 63, está la **administración de directos**, en la que aparece una lista con los directos de todos los usuarios de la plataforma. Esta lista, tiene una columna con dos botones, uno para **editar el directo** y otro para **iniciar y finalizar ese directo**. La opción de editar abre un formulario desplegable que muestra los datos del directo a editar.

**Administrar directos**

Directos de los canales

	Título	Canal	CanalID	Directo	Categorías	Acciones
▼	Directo de viernes	ssssssssssss	Ueo7nqNjzFYHpsmLaAs0IV7juRh1	<input type="checkbox"/>	2	
▼	Jugando a Valorant	absalom	fuisv831qjU7Pr03M2a2z0UUQ0h2	<input type="checkbox"/>	1	
▼	Directo SergioAdmin	SergioAdmin	EvUIEp6HZKUsRbN58dWGWUJqy083	<input checked="" type="checkbox"/>	2	
▼	Directo de papo	papo	23Y0XyPduBY5Vn4L6PwBBfbY0UE2	<input type="checkbox"/>	2	
▼	Directo 2	pepe	7oUEOXIyINPvQae3Iu6yPK8gQA23	<input type="checkbox"/>	3	

Rows per page: 5 1-5 of 5

Figura 63: Administración de directos

Posteriormente, la **sección de videos**, en la Figura 64, que contiene una lista con los videos de la plataforma y sus datos, así como una columna de acciones, con una acción para **editar video**, y otra opción para **eliminar**. La opción de editar abre un formulario desplegable con los datos del video a editar.

**Administrar videos**

Videos de la plataforma

	Título	Canal	Visitas	Likes	Fecha	Comentarios	Categorías	Acciones
▼	Video de prueba	absalom	29	3	2021-4-25 19:11:9	<input checked="" type="checkbox"/>	2	
▲	video1	SergioAdmin	193	2	2021-4-25 19:13:54	<input checked="" type="checkbox"/>	1	
Descripción: Hola a todos								
▼	video4	SergioAdmin	48	3	2021-4-25 19:15:30	<input type="checkbox"/>	2	
▼	Video de paisajes	SergioAdmin	2	1	2021-6-26 23:3:7	<input checked="" type="checkbox"/>	1	
▼	Video de pepe	pepe	15	0	2021-4-25 19:16:34	<input checked="" type="checkbox"/>	1	

Rows per page: 5 1-5 of 10

Figura 64: Administración videos

Después, se puede desplegar la pestaña de **Administración de series**, Figura 65, en la que aparece una lista de series de los canales de la plataforma, junto con dos opciones, la primera para **editar** y la otra para **eliminar**. La opción de editar abre un formulario desplegable que contiene los datos y videos de la serie a editar.

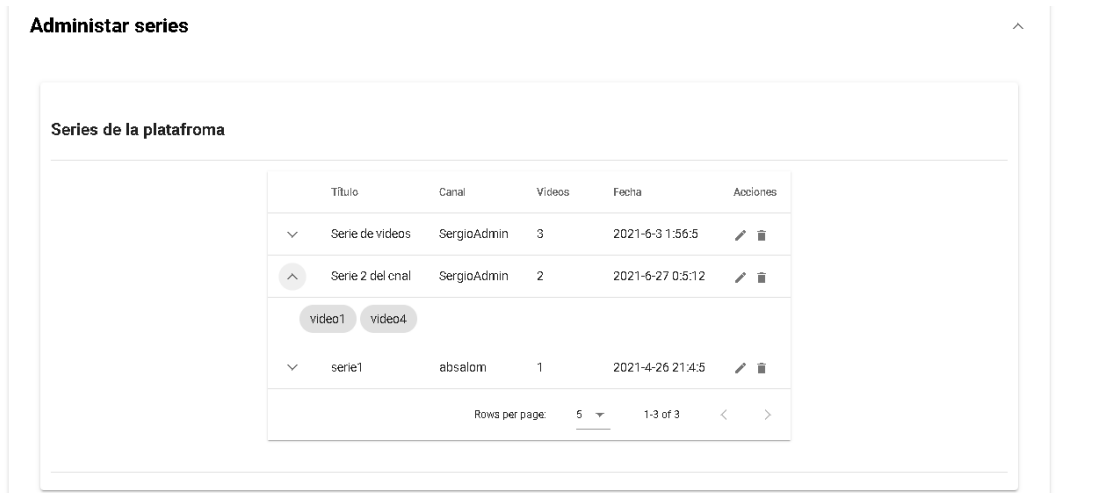


Figura 65: Administración de series

Posteriormente, la **administración de categorías**, Figura 66. En esta, se muestra el formulario para **añadir una categoría**, y justo después, una lista de categorías, mostrando sus datos y dos opciones, una para **editar** y otra para **eliminar** la categoría. La opción de editar abre un formulario desplegable con los datos de la categoría.

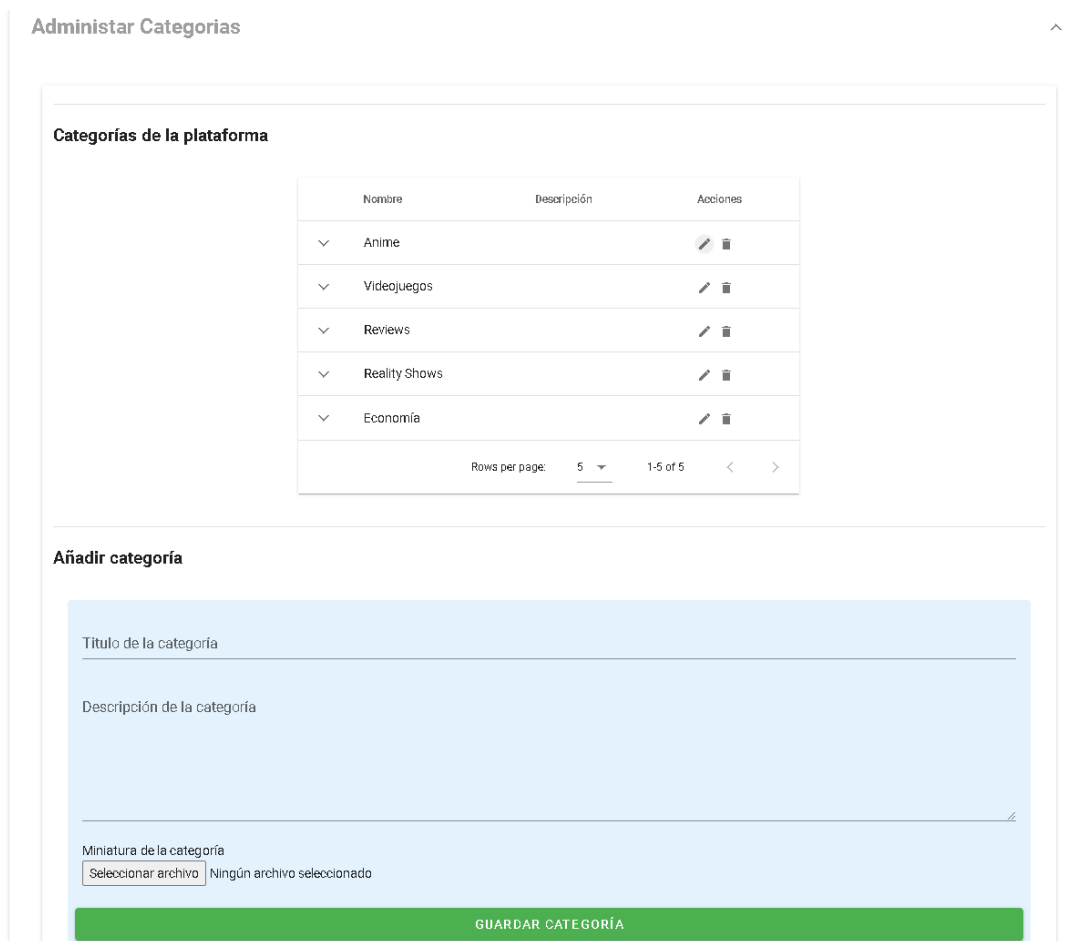


Figura 66: Administrar categoría

Posteriormente, está la sección de **administración de solicitudes**, en la Figura 67. En este apartado, podremos ver la lista de solicitudes del sistema, con su información, así como 2 opciones. En primer lugar, si la solicitud está en estado pendiente, se puede o bien **aceptar**, o bien **rechazar**, y si está en estado rechazado o aceptado, se puede **eliminar**, presionando la papelera.

inistar solicitudes

**Solicitudes para administrar**

ID Solicitante	Nombre Solicitante	Estado	Fecha	Acciones
EvUjEp6HZkUsRbN58dWGWUJuy0B3	SergioAdmin	rechazada	2021-5-25 19:11:9	
XDRgoXkNJGaUDuGahDpDwKTKyut2	estaracliada	pendiente	2021-5-5 0:4:5	

Rows per page: 5 1-2 of 2 < >

ministar usuarios

Figura 67: Administración solicitudes

Por último, podemos acceder a la **sección de administración de usuarios**, Figura 68. Una vez dentro de esta sección, se puede visualizar una lista de los usuarios del sistema con sus datos. Además, aparecen dos acciones, la primera para **suspender o habilitar al usuario** y la segunda para el **cambio del rol del usuario**.

nistar usuarios

**Usuarios de la plataforma**

Nombre	Identificador	Rol	Correo	Seguidores	Suspensión	Rol
estaracliada	XDRgoXkNJGaUDuGahDpDwKTKyut2	user	aaaa@aaaa.com	0		
ssssssssssss	Ueo7nqNjzFYHpsmLaAs0IV7juRh1	canal	varo@hhhh.com	1		
papo	23YDXyPduBY5Vn4L6PwB8fY0UE2	canal	papo@pepe.com	2		
Estado: !!!Hola a todos, estoy usando MY-TV!!!						
absalom	fuisv831qjU7Pr03M2a2z0UUQ0h2	canal	sergiopaminecraft@gmail.com	3		
HoganHulk	g08WsgQ9PKMEbnfxEZK1qQfsQaj2	user	hogan@gmailf.com	0		

Rows per page: 5 1-5 of 6 < >

Figura 68: Administración de usuarios

La suspensión se ve de forma más detallada en la Figura 69, saliendo en rojo los suspendidos.

absalom	fuisv831qjU7Pr03M2a2z0UUQ0h2	canal	sergiopaminecraft@gmail.com	3		
HoganHulk	g08WsgQ9PKMEbnfxEZK1qQfsQaj2	suspendido	hogan@gmailf.com	0		

Rows per page: 5 1-5 of 6 < >

Figura 69: Suspensión usuarios

Por otro lado, el cambio del rol despliega una ventana como la de la Figura 70, donde podemos seleccionar el nuevo rol o cancelar.



Figura 70: Cambiar rol

#### 6.9.4.3. Administrar comentarios

El administrador también tiene los privilegios para eliminar los comentarios de cualquier usuario en cualquier video, aparte de poder eliminar y editar sus propios comentarios. Para ello, se crea una interfaz como la de la Figura 71, accesible en las páginas de los videos.

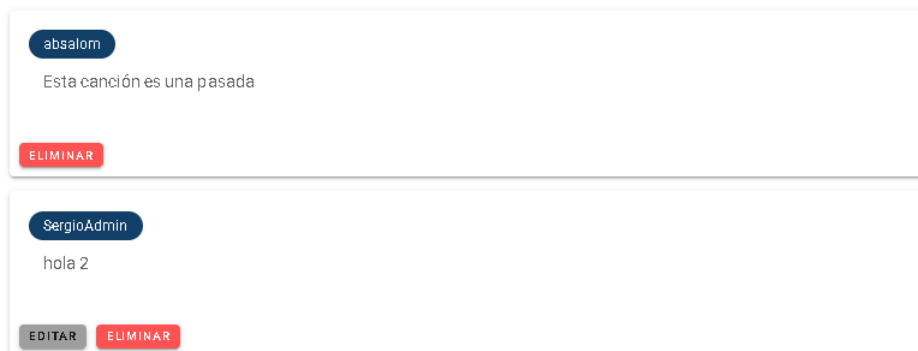


Figura 71: Administrar comentario

Respecto al usuario administrador, este también puede realizar todas las funcionalidades de un canal.

Por último, quiero repetir que lo que se ha mostrado aquí solo son algunos de los aspectos más relevantes, y que por tanto, faltan muchas interfaces y opciones que no se han mostrado. Se ha hecho esto para no hacer demasiado largo el documento. Los detalles están en el **Anexo VI**.

## 7. Conclusiones

Una vez se ha realizado el proyecto, y se ha pasado por todas las fases de este, se pueden sacar en claro una serie de conclusiones que demuestran, en parte, el éxito de los objetivos del proyecto y los objetivos personales.

Para comenzar con las conclusiones, es importante mencionar, que según mi opinión, se ha logrado cumplir todos los requisitos que se habían propuesto para el sistema, al menos los funcionales. Los requisitos no funcionales, son algo más complejos de probar, ya que en algunas ocasiones se requeriría un estudio con un mayor número de usuarios o la realización de pruebas de carga, por ejemplo. También, en base a esto, se puede decir que se han cumplido los **objetivos del sistema**, que habían sido detallados al principio de esta memoria y en el Anexo II. Por lo que se podría considerar que el proyecto ha terminado de forma correcta.

Esta asunción de que se han cumplido los objetivos se puede comprender en base a los resultados del prototipo funcional entregado:

- **Gestión de usuarios:** Se ha cumplido dado que tenemos múltiples usuarios con roles que se pueden moderar y que tienen distintas tareas y niveles de acceso.
- **Gestión de Directos:** Se ha cumplido porque se pueden emitir, visualizar y moderar directos.
- **Gestión de Vídeos:** Se ha cumplido porque se pueden subir, visualizar y moderar videos.
- **Gestión de Series:** Se ha cumplido porque se pueden crear, visualizar y moderar series.
- **Gestión de Categorías:** Se ha cumplido porque se pueden crear y utilizar categorías.
- **Gestión de Eventos:** Se ha cumplido porque se pueden añadir y visualizar eventos.
- **Gestión de Solicitudes:** Se ha cumplido porque se pueden generar y administrar solicitudes.
- **Gestión de Comentarios:** Se ha cumplido porque se pueden añadir y moderar comentarios.
- **Gestión de Seguimiento:** Se ha cumplido debido a que se puede seguir a los canales.
- **Gestión de Me gusta:** Se ha cumplido porque un usuario puede indicar que le gusta un vídeo.
- **Gestión de Estadísticas:** Se ha cumplido ya que se recopilan datos de la actividad de los usuarios en los contenidos de la plataforma.
- **Gestión de Alertas:** Se ha cumplido porque se pueden mostrar alertas a los usuarios.
- **Gestión de Notificaciones:** Se ha cumplido ya que se generan notificaciones cuando es necesario.

Todos estos objetivos, se han completado mediante el seguimiento de un proceso de desarrollo de software completo, desde los requisitos, hasta la obtención de un prototipo funcional.

Es gracias precisamente al hecho de que se ha desarrollado un sistema completo, que se podría hablar también de la realización de los **objetivos personales**. Esto se puede ver por ejemplo, en que durante el transcurso del proyecto, se han aprendido o mejorado numerosas técnicas relacionadas con el desarrollo, así como se han aprendido herramientas que son muy útiles para el ámbito de la informática, y que servirán para futuros proyectos. También, como ya he comentado, se ha logrado por primera vez, poner en común todos los conocimientos captados a lo largo de los años, en un entorno completamente práctico, en vez de en exámenes o trabajos parciales, comprendiendo así el esfuerzo que significa un desarrollo completo, algo que va más



allá de posibles estimaciones de esfuerzo o planificaciones temporales. Esto ha servido también para mejorar la autoorganización, algo que será útil en el futuro.

Por supuesto, también es muy importante entender, que al ser la primera vez que se ha realizado un proyecto de esta magnitud, se han cometido errores, y con seguridad, el resultado final no es perfecto, porque todavía quedan muchas cosas por aprender, sin embargo, de aquellos errores que si me he dado cuenta, podré tenerlos presentes en futuras ocasiones. Básicamente, se ha obtenido una experiencia que es imposible adquirir sin realizar algo de esta magnitud.

Me gustaría añadir, que todo esto, ha sido posible gracias a la motivación que ha generado el hecho de que el proyecto tratase un tema muy de mi interés. Entiendo que en otras ocasiones, cuando se tengan que realizar proyectos que no nos gustan tanto, no existirá este tipo de motivación, aunque habrá otras razones para hacerlo. Al menos, ya no será la primera vez que se aplica un proceso de desarrollo de software completo, por lo que se tendrán conocimientos para organizarse, así como se tendrá una mayor experiencia en la elección y uso de las herramientas adecuadas.

## 8. Dificultades y líneas futuras

En este apartado, se exponen las dificultades encontradas durante el desarrollo del proyecto, así como las soluciones que se han aplicado, y las posibles mejoras que se podrían realizar de cara al futuro

### 8.1. Problemas y soluciones

Como es lógico, y como ya he comentado durante las conclusiones, en la realización del proyecto, se han encontrado numerosas dificultades que han retrasado el avance de este.

En primer lugar, se podría decir que un gran problema en todo proyecto, son los **plazos**. Aquí también es el caso, ya que todos los plazos de entrega suponen una presión añadida. Al principio parecía que había bastante tiempo, y que si empezaba pronto, podría terminar antes de los exámenes finales, pero con el cansancio de los exámenes anteriores, las prácticas de empresa, que han consumido casi todos los horarios de mañana, así como los trabajos y exámenes adicionales, se ha terminado la planificación de forma muy ajustada.

Por otro lado, y la dificultad que más ha pesado a lo largo del desarrollo, ha sido el **desconocimiento de la mayoría de las herramientas utilizadas**. Durante la carrera se han aprendido y practicado muchos programas muy útiles, algunos de los cuales se han podido utilizar en este proyecto, y otros han supuesto una base para el desarrollo, pero sin embargo, nunca había programado ninguna aplicación web completa, por lo que existían solo ligeros conocimientos sobre HTML o CSS. Lo mismo ha ocurrido con JavaScript, aunque parecido a C, ha supuesto sus dificultades. Por otro lado, en cuanto al desarrollo de los servidores, aunque sí se había probado antes en algunas asignaturas, nunca había realizado nada parecido a un servidor de streaming de video, lo que ha supuesto aprender muchos nuevos conceptos.

Otro de los posibles problemas, es la elección de **Firestore como gestor de base de datos**. Esto se debe, a que como ya he comentado en anteriores ocasiones, firestore ofrece una base no relacional, lo que significa que a la hora de almacenar estructuras complejas de datos, sea necesaria la aplicación de procedimientos de filtrado en otras estructuras de datos de la base para poder presentar todo el contenido de manera adecuada. Esto ha supuesto un mayor tiempo de desarrollo, y ha producido que el programa presente algunas redundancias que se podrían haber evitado. Aun así, considero que la utilización de Firestore merece la pena, ya que a esta escala de forma correcta, e implementar algo más complejo podría suponer una mayor pérdida de tiempo. Esto, además de la utilidad de las funciones de autenticación y otras ventajas que ya se han comentado en el apartado de implementación, son las razones de su elección.

Algo parecido a Firestore, sucede con **JavaScript**, ya que en algunas ocasiones, se echan de menos algunas características como estructuras de datos más dinámicas u opciones de filtrado más completas. Sin embargo, igual que antes, su simplicidad y facilidad de uso, además de sus posibilidades, hacen que merezca la pena.

Todo esto, se ha juntado con el pobre rendimiento de mi equipo, lo que limita en muchas ocasiones las pruebas de los servidores de video, tanto por retardos como por falta de memoria.

## 8.2. Mejoras para el futuro

Ahora que se puede ver el resultado del proyecto realizado, y que ya se han extraído una serie de conclusiones, se puede proponer una serie de ideas o mejoras que han quedado en el tintero y que podrían ampliar bastante el proyecto de cara al futuro.

En cuanto a estas posibles mejoras, entraría la modificación de la base de datos firebase por una estructura propia, que nos permitiese escalar sin necesidad de depender de servicios externos como es Google. Esto significa la **implementación de un servidor propio** para almacenar los elementos que actualmente se almacenan en Firebase, algo parecido a la decisión que se tomó con los videos, debido a la necesidad de mantenerlos en la misma memoria que el servidor de video.

Otra de las posibles mejoras a nivel de aplicación estaría relacionada con el **análisis de datos**, de forma que en base a los datos de los usuarios en la plataforma, se pudiese establecer un sistema de recomendaciones que se base en los gustos individuales de los usuarios en vez de en el número de visitas u otros criterios genéricos. Esto viene de la mano de la implementación de algún motor de búsqueda básico, ya que el actualmente los resultados se filtran por exactitud y categorías.

Por otro lado, y siendo más ideas que se podrían implementar para mejorar la experiencia del usuario y la funcionalidad de la aplicación, tenemos las siguientes posibles implementaciones:

- **Chat en directo:** Se podría añadir en la página de los directos un chat, que se podría habilitar por el canal para que los usuarios de su comunidad escribieran en tiempo real.
- **Sistema de estadísticas completo:** Un sistema de estadísticas avanzado para los usuarios canales, más allá de las visitas de sus videos o espectadores de sus directos, por ejemplo, se podrían recoger datos de las regiones, edades u otros. Además se podrían extender las estadísticas al lado del usuario, con datos como horas de visualización o contenidos favoritos, entre otras posibilidades.
- **Visitantes de un directo:** Se podría crear un mecanismo para establecer el número de visitantes simultáneos en un directo.
- **Opciones de personalización:** Como por ejemplo perfiles de usuario públicos con banner o colores para personalizar la página de los canales, entre otros.
- **Sistema de propuesta de categorías:** Los canales podrían proponer categorías para sus contenidos, que podrán ser establecidas como categorías por los administradores.
- **Directos grupales:** La posibilidad de que varios canales se uniesen para hacer un directo común y que este apareciese a la vez en sus respectivas páginas o en una combinada para varios canales como una especie de grupo.
- **Sistema de amigos:** La posibilidad de agregar amigos en la plataforma para ver que están viendo o hablar con ellos.
- **Sistema de donaciones:** Para aportar económicamente a la plataforma más allá de la publicidad, se podrán hacer donaciones directas a los canales, implementando medios de pago como PayPal.

## 9. Referencias

- [1] YouTube
  - <https://www.youtube.com/> (Accedido: 28/06/2021)
  - <https://es.wikipedia.org/wiki/YouTube> (Accedido: 28/06/2021)
- [2] Amazon prime video
  - [https://es.wikipedia.org/wiki/Amazon\\_Prime\\_Video](https://es.wikipedia.org/wiki/Amazon_Prime_Video) (Accedido: 28/06/2021)
- [3] Twitch
  - <https://www.twitch.tv/> (Accedido: 28/06/2021)
  - <https://es.wikipedia.org/wiki/Twitch> (Accedido: 28/06/2021)
- [4] INE. - "Población que usa internet".
  - [https://www.ine.es/ss/Satellite?L=es\\_ES&c=INESeccion\\_C&cid=1259925528782&p=1254735110672&pagename=ProductosYServicios%2FPYSLayout](https://www.ine.es/ss/Satellite?L=es_ES&c=INESeccion_C&cid=1259925528782&p=1254735110672&pagename=ProductosYServicios%2FPYSLayout) (Accedido: 28/06/2021)
- [5] Statista. - "Televisión por grupo de edad".
  - <https://es.statista.com/estadisticas/476132/porcentaje-de-espectadores-de-television-en-espana-por-edad/>(Accedido: 28/06/2021)
- [6] RTVE. - "La pandemia de internet"
  - <https://www.rtve.es/noticias/20210210/sube-tiempo-usuarios-redes-sociales-espana-confinamiento-pandemia/2074699.shtml> (Accedido: 28/06/2021)
- [7] Statista. - "Evolución de la publicidad en TV online".
  - <https://es.statista.com/estadisticas/486489/prevision-de-ingresos-por-publicidad-en-tv-online-en-espana/> (Accedido: 28/06/2021)
- [8] Streaming
  - <https://es.wikipedia.org/wiki/Streaming> (Accedido: 28/06/2021)
- [9] Estándares web
  - [https://es.wikipedia.org/wiki/Est%C3%A1ndares\\_web](https://es.wikipedia.org/wiki/Est%C3%A1ndares_web) (Accedido: 28/06/2021)
- [10]Vue.js.
  - <https://es.wikipedia.org/wiki/Vue.js> (Accedido: 28/06/2021)
  - <https://vuejs.org/>(Accedido: 28/06/2021)
- [11]JavaScript
  - <https://es.wikipedia.org/wiki/JavaScript>(Accedido: 28/06/2021)
- [12]Vuetify
  - <https://vuetifyjs.com/en/>(Accedido: 28/06/2021)
- [13]Axios
  - <https://desarrolloweb.com/articulos/axios-ajax-cliente-http-javascript.html>(Accedido: 28/06/2021)
- [14]Firebase
  - <https://es.wikipedia.org/wiki/Firebase> (Accedido: 28/06/2021)
- [15]HTML
  - <https://es.wikipedia.org/wiki/HTML> (Accedido: 28/06/2021)
- [16]CSS
  - [https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada) (Accedido: 28/06/2021)
- [17]Node.js
  - <https://es.wikipedia.org/wiki/Node.js> (Accedido: 28/06/2021)
  - <https://nodejs.org/es/> (Accedido: 28/06/2021)

- [18]NPM
  - <https://es.wikipedia.org/wiki/Npm> (Accedido: 28/06/2021)
- [19]NGINX
  - <https://en.wikipedia.org/wiki/Nginx>(Accedido: 28/06/2021)
  - <https://nginx.org/en/> (Accedido: 28/06/2021)
- [20]PM2
  - [https://en.wikipedia.org/wiki/PM2\\_\(software\)](https://en.wikipedia.org/wiki/PM2_(software)) (Accedido: 28/06/2021)
- [21]Draw.io
  - <https://www.diagrams.net/> (Accedido: 28/06/2021)
  - <https://www.mediawiki.org/wiki/Extension:DrawioEditor?action=view> (Accedido: 28/06/2021)
- [22]Microsoft Project
  - [https://es.wikipedia.org/wiki/Microsoft\\_Project](https://es.wikipedia.org/wiki/Microsoft_Project) (Accedido: 28/06/2021)
- [23]Visual Studio Code
  - [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code) (Accedido: 28/06/2021)
- [24]JSDoc
  - <https://es.wikipedia.org/wiki/JSDoc> (Accedido: 28/06/2021)
- [25]Vue Styleguidist
  - <https://vue-styleguidist.github.io/> (Accedido: 28/06/2021)
- [26]Adobe XD
  - <https://www.adobe.com/es/products/xd.html>(Accedido: 28/06/2021)
  - [https://es.wikipedia.org/wiki/Adobe\\_XD](https://es.wikipedia.org/wiki/Adobe_XD)(Accedido: 28/06/2021)
- [27]OBS
  - [https://es.wikipedia.org/wiki/Open\\_Broadcaster\\_Software](https://es.wikipedia.org/wiki/Open_Broadcaster_Software) (Accedido: 28/06/2021)
- [28]VirtualBox
  - <https://es.wikipedia.org/wiki/VirtualBox> (Accedido: 28/06/2021)
- [29]García Peñalvo, F. J., García Holgado, A. - "Transparencias de ingeniería de software I".  
Introducción al Proceso unificado.
  - <https://repositorio.grial.eu/bitstream/grial/1948/1/7.%20PU-2020.pdf> (Accedido: 28/06/2021)
- [30]Duran Toro, A., Bernárdez Jiménez, B. - "Metodología de Elicitación de requisitos de sistemas Software"
  - <http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf> (Accedido: 28/06/2021)
- [31]Patrón MVVM
  - <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel> (Accedido: 28/06/2021)
  - [https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93modelo\\_de\\_vista](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93modelo_de_vista) (Accedido: 28/06/2021)