

**CICLO FORMATIVO DE GRADO SUPERIOR:**

DESARROLLO DE APLICACIONES WEB

**AUTORES:**

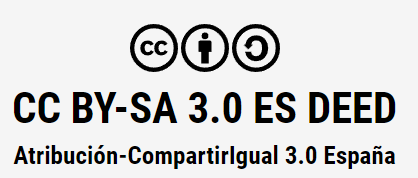
Daniel García Ayala

Daniel Alonso Lázaro

Álvaro Mateos Robledillo

**LICENCIA**

Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envie una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



**DEDICATORIAS Y AGRADECIMIENTOS**

**Daniel García Ayala**

Gracias a mi madre por ver el progreso y querer probarlo, aunque no se entere nada de cómo funcionan las cosas, al igual que con el anterior proyecto.

**Daniel Alonso Lázaro**

Μὴ γιγνώσκοντες τὴν ὕβριν ἣν ἐποίουν... Ad astra, per aspera.

**Álvaro Mateos Robledillo**

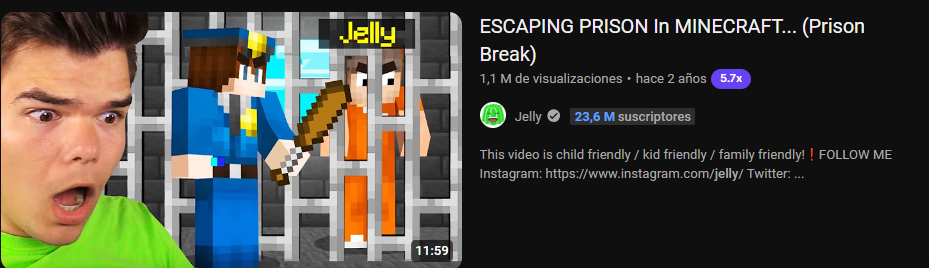
A mi novia, que siempre confía en mí y me apoya en cada paso del camino.

**RESUMEN**

Vidmentor es una aplicación web innovadora que permite a los usuarios poder generar un gran abanico de ideas para la creación de contenido en formato horizontal para la plataforma YouTube. Dicho software está diseñado para que los usuarios puedan tener una gestión óptima de su contenido, pudiendo ser organizada mediante un calendario para que el usuario pueda agilizar en lo máximo posible su tiempo, todo ello a través de una interfaz sutil y manejable, donde los usuarios pueden acceder en todo momento, tanto a sus ideas guardadas, planificaciones de estas, guionizado y post-guionizado y la generación de dichas ideas.

La idea de desarrollo viene asociada a la creciente popularidad de las redes sociales y la necesidad de las personas de generar dinero con ellas haciendo contenido viral para las mismas, intentando replicar lo que sus creadores de contenido favoritos hacen. Sin embargo, lo que el individuo que quiere dedicarse a ello ve, solo es el producto final, no tiene métodos ni medios para saber como se puede realizar una pieza visual de contenido viral, sobre todo, intentando imitar a otros creadores de contenido de su mismo nicho e idioma, debido a que es un producto que ya ha sido vendido a gran escala y, al copiarlo, solo unos pocos querrán adquirir el tuyo. No obstante, se pueden ver ejemplos de creadores de contenido que han sabido adaptar ideas angloparlantes al español y han resultado ser un caso de éxito, algunos casos de estos pueden ser el Youtubers más famoso de España *Mikecrack*, adaptando a creadores como *Jelly* o *PrestionPlayz*.

**Vídeo Original de Jelly Vídeo Original de Mikecrack**

Vidmentor se encarga de ofrecer las ideas las virales del hablante anglosajona y ofrecerlas al creador de contenido hispanohablante traducidas para poder orientar desde el primer momento el contenido a tener un alto potencial viral, ofreciendo no solo las ideas, sino las miniaturas más virales en función a la idea seleccionada. También Vidmentor se encarga de como mejorar la viralidad del vídeo tras su subida, recordando al usuario que si no ha alcanzado la viralidad esperada ha de cambiar la miniatura a otras más llamativa o modificar el título, comprobando su CTR (porcentaje de clicks por impresiones), y fomentando que el algoritmo reconozca su potencial con visitas externas a la página original, por ejemplo mediante Instagram.

La demanda por el producto se considera elevada ya que uno de cada tres jóvenes en España está interesado en dedicarse a trabajos relacionados con la creación de contenidos en internet, por lo que Vidmentor facilita la entrada de estos nuevos personajes públicos para poder trabajar de ello de manera sostenible, ofreciendo una plataforma online desde la cual se puede acceder en cualquier momento, desde todos los dispositivos que soporten un navegador para que optimicen su tiempo y sacarle máximo rendimiento y teniendo como principal foco de atención la plataforma YouTube, ya que, en cuanto a las más relevantes que hay en el mercado, es la que mayores ingresos genera y ofrece mayor sostenibilidad por visitas residuales.

En resumen, Vidmentor es un software que ofrece a cualquier persona la posibilidad de adentrarse en la creación de contenido en internet, ofreciendo ideas y miniaturas específicas para el nicho específico de contenido, gestión de producción y post-producción del mismo, y una planificación para las distintas piezas audiovisuales.

**ABSTRACT**

Vidmentor is an innovative web application designed to generate a wide range of ideas for creating horizontal format content for the YouTube platform. This software enables users to optimally manage their content, organizing it through a calendar that streamlines their time management. With a subtle and manageable interface, users can access their saved ideas, planning, scripting, post-scripting, and idea generation at any time.

The development of Vidmentor is linked to the growing popularity of social networks and the need for individuals to monetize viral content. Aspiring content creators often see only the final product without understanding the process behind viral content creation. Vidmentor addresses this need by providing methods and means to create visually appealing content, replicating the success of famous content creators in the user's own language and niche. Successful examples include Spanish YouTubers like *Mikecrack*, who have adapted ideas from English-speaking creators like *Jelly* or *PrestonPlayz* with great success.

Vidmentor offers viral ideas from English-speaking creators, translated for Spanish-speaking content creators, guiding them from the start to maximize viral potential. In addition to ideas, Vidmentor provides the most effective thumbnails based on the selected idea and tips to improve video virality after upload. This includes recommendations to change thumbnails or titles if the expected virality is not reached, checking the CTR (click-through rate), and encouraging algorithm recognition through external visits, for instance, via Instagram.

The demand for Vidmentor is high, with one in three young people in Spain interested in jobs related to internet content creation. Vidmentor facilitates the entry of these new public figures by providing an online platform accessible at any time from any device with a browser. This helps users optimize their time and maximize their output, focusing primarily on YouTube, the platform that generates the highest income and sustainability through residual visits.

In summary, Vidmentor is software that allows anyone to delve into internet content creation. It offers specific ideas and thumbnails for particular content niches, production and post-production management, and planning for various audiovisual pieces, ensuring sustainable success on the YouTube platform.

Índice de contenido

[1. INTRODUCCIÓN](#__RefHeading__2216_364215148) 10

[2 NECESIDADES DEL SECTOR PRODUCTIVO 9](#__RefHeading__4307_478974897)

[2.1 Análisis de la situación actual 9](#__RefHeading__4644_1717060311)

[2.2 Necesidades del cliente y oportunidad de negocio 9](#__RefHeading__4646_1717060311)

[2.3 El nuevo proyecto: Vidmentor 9](#__RefHeading__4648_1717060311)

[3 DISEÑO DEL PROYECTO 11](#__RefHeading__4309_478974897)

[3.1 Fases del proyecto 11](#__RefHeading__864_980353409)

[3.1.1 Análisis 11](#__RefHeading__866_980353409)

[3.1.2 Diseño 11](#__RefHeading__868_980353409)

[3.1.3 Implementación 11](#__RefHeading__870_980353409)

[3.1.4 Pruebas 12](#__RefHeading__928_1438982970)

[3.2 Objetivos a conseguir 13](#__RefHeading__874_980353409)

[3.3 Previsión de recursos materiales y humanos necesarios 13](#__RefHeading__876_980353409)

[3.4 Presupuesto económico. 13](#__RefHeading__878_980353409)

[4 PLANIFICACIÓN DE LA EJECUCIÓN DEL PROYECTO 14](#__RefHeading__4311_478974897)

[4.1 Fase de Análisis 14](#__RefHeading__938_1438982970)

[4.2 Fase de diseño 14](#__RefHeading__940_1438982970)

[4.3 Fase de Implementación 14](#__RefHeading__942_1438982970)

[4.4 Fase de pruebas 15](#__RefHeading__944_1438982970)

[5 DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN 16](#__RefHeading__4313_478974897)

[6 FUENTES 17](#__RefHeading__4319_478974897)

[7 ANEXOS 18](#__RefHeading__4321_478974897)

[7.1 Guía de estilo 19](#__RefHeading__7110_478974897)

Índice de figuras

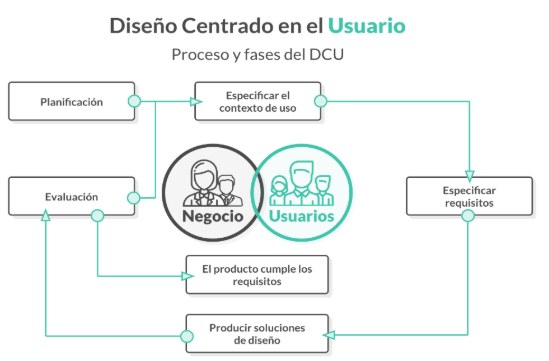
1. [Esquema diseño centrado en el Usuario 10](#DisenoUsuario)
2. [Diagrama diseño centrado en el Usuario 11](#DisenoUsuario)
3. [Etapas sprint metodologías ágiles 11](#EtapasSprint)
4. [Tablero Kanban de la herramienta Trello 12](#TableroTrello)
5. [Detalle de tarea en herramienta Trello 12](#DetalleTrello)
6. [Gráfica de ganancias de Youtube 13](#GanaciasYT)
7. [Imagen de colaboración de Sandy con Opera GX 15](#OperaGXSandy)
8. [Ejemplo 1 de público objetivo 16](#Ejemplo1TargetClient)
9. [Ejemplo 2 de público objetivo 16](#Ejemplo2TargetClient)
10. [Tabla Entidad-Relación de la Base de Datos 21](#TablaER)
11. [Tabla Usuario en la Base de Datos 22](#TablaUsuario)
12. [Tabla Token en la Base de Datos 22](#TablaToken)
13. [Tabla Tipo en la Base de Datos 23](#TablaTipo)
14. [Tabla Rol en la Base de Datos 23](#TablaRol)
15. [Tabla Detalle en la Base de Datos 24](#TablaDetalle)
16. [Tabla Intereses en la Base de Datos 25](#TablaIntereses)
17. [Tabla Rol Intereses en la Base de Datos 25](#TablaIntereses)
18. [Listado de canales estudiados 26](#ListadoCanales)
19. [Gráfico de exposición de organización de una pieza 27](#GráficoHistoria)
20. [Listado de redes sociales más usadas 27](#RRSSmásUsadas)
21. [Prototipo en figma 34](#PrototipoFigma)

# INTRODUCCIÓN

Este documento responde a la realización del **módulo de Proyecto** del CFGS en Desarrollo de Aplicaciones Web.

El objetivo principal de este proyecto es la implementación de este software web con un diseño accesible e intuitivo para generar ideas y títulos para vídeos con alto potencial viral y poder desarrollar un guion acorde a dicha idea, y tener referencia de las miniaturas más virales para la misma, poder organizar el contenido de la forma más cómoda posible y poder triunfar en las redes sociales con rapidez y esfuerzo.

Otros objetivos más a largo plazo son poder generar las etiquetas de los vídeos de manera automática para poder facilitar el posicionamiento de los vídeos en las funciones de exploración de YouTube, una función automática para la generación de contenido vertical para poder expandir el software a poder ser usado para otras plataformas como *TikTok* y poder expandir Vidmentor a otros idiomas para tener mayor escalabilidad como producto.

Para que el software se ajuste adecuadamente tanto a los contextos funcionales como tecnológicos, es imprescindible que el proceso de desarrollo se enfoque en un diseño centrado en el usuario y su experiencia1. Para lograrlo, se implementarán metodologías ágiles, que facilitan una colaboración y coordinación constante entre los desarrolladores y los clientes. Estas metodologías permiten considerar cada fase del desarrollo y recopilar continuamente la retroalimentación de los usuarios.

*Esquema diseño centrado en el Usuario*

1 *“El Diseño Centrado en el Usuario es una filosofía de diseño que tiene como objetivo la creación de productos que satisfagan las necesidades y resuelvan los problemas de los usuarios a los cuales está destinado dicho producto. El Diseño Centrado en el Usuario tiene su origen en la Ingeniería del Software, el diseño industrial y militar. Don Norman, cofundador de Nielsen Norman Group, fue la primera persona que comenzó a utilizar el término Diseño Centrado en el Usuario. El término DCU es utilizado como método para el diseño de interfaces de usuario. Cuando se diseña siguiendo la filosofía DCU el usuario debe de situarse en el centro de todo. No es posible concebir el producto desligado de su uso, el contexto o las necesidades del usuario final.”* Arias del Prado, J. (2020, 4 de marzo). Diseño Centrado en el Usuario (DCU). Todas las claves del proceso. Blog.UXABLES. <http://www.uxables.com/diseno-ux-ui/diseno-centrado-en-el-usuario-dcu-todas-las-claves-del-proceso/>

Gráfico

Descripción generada automáticamente

*Diagrama diseño centrado en el Usuario*

Etapas sprint metodologías ágiles
Para este proyecto, se adoptaron los doce principios del manifiesto Agile2 y se empleó el tablero Kanban como metodología ágil para gestionar las tareas de manera detallada. La coordinación del equipo de desarrollo se llevó a cabo mediante reuniones semanales (*sprints*), en las cuales se revisaron los avances, se definieron los pasos a seguir y se realizaron las pruebas necesarias para concluir las tareas asignadas. Estas reuniones, mayoritariamente telemáticas, utilizaron herramientas en línea para facilitar la colaboración a distancia, aunque en ocasiones específicas se realizaron encuentros presenciales para tratar temas más complejos. Este enfoque híbrido de comunicación, junto con el uso del tablero Kanban, permitió mantener una estructura sólida, fomentando la colaboración, la transparencia y la entrega continua de valor, asegurando así la eficiente ejecución del proyecto.

*Etapas sprint metodologías ágiles*

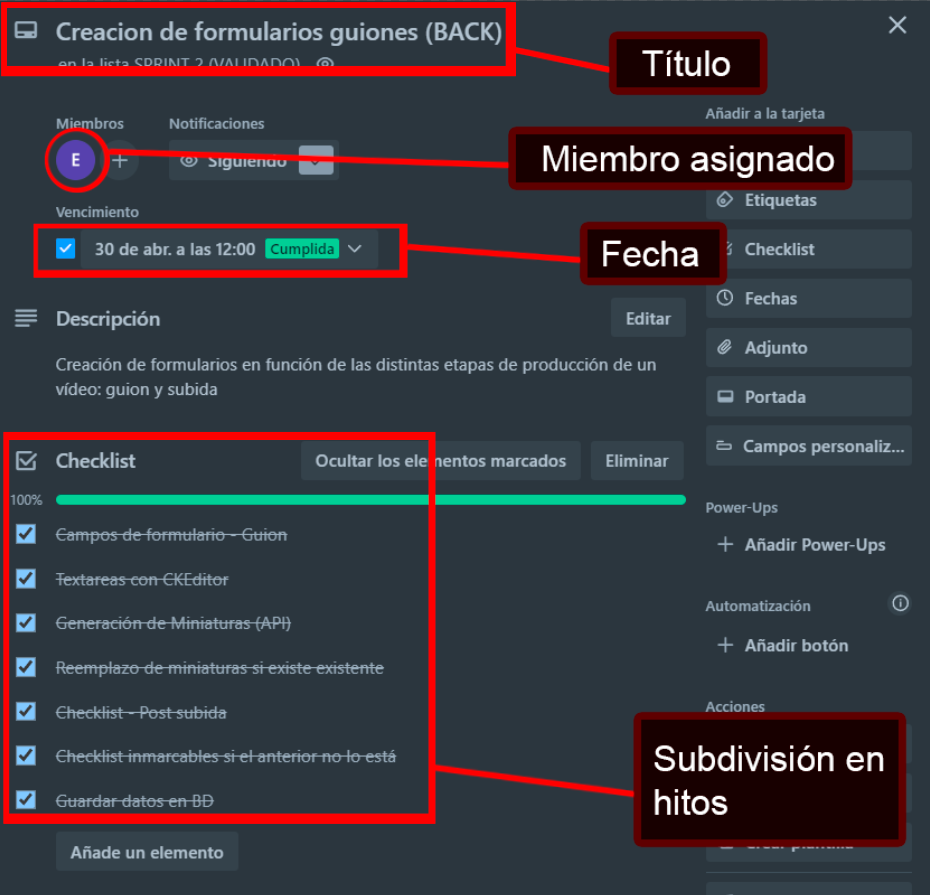
2 Cunningham, W. (2001) Principios del Manifiesto Ágil. agilemanifiesto.org

<https://agilemanifesto.org/iso/es/principles.html>

Tablero Kanban de la herramienta Trello


*Tablero Kanban de la herramienta Trello*

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

*Detalle de tarea en Herramienta Trello*

# NECESIDADES DEL SECTOR PRODUCTIVO

En la actualidad, se observa una tendencia creciente entre los jóvenes de querer dedicarse profesionalmente a las redes sociales, un mercado que ha demostrado ser económicamente viable. De acuerdo con los datos recientes, se estima que uno de cada tres jóvenes aspira a trabajar en áreas relacionadas con las redes sociales. Esta inclinación hacia las plataformas digitales puede atribuirse al reconocimiento de que es posible obtener ingresos sustanciales y sostenibles en este ámbito.

Sin embargo, a pesar de este interés masivo, no se ha encontrado una plataforma accesible y estructurada que sirva como punto de partida para aquellos que desean iniciarse en este camino. Aquí es donde *Vidmentor* juega un papel crucial. *Vidmentor* ha sido diseñada específicamente para servir como un recurso de iniciación para futuros *YouTubers*, considerando que YouTube es la plataforma que actualmente genera los mayores ingresos para los creadores de contenido. Como se puede observar en las siguientes imágenes, en el siguiente canal de nicho *gaming* con únicamente 7 millones de visitas mensuales se han logrado superar los 1400 euros de ingreso, únicamente del dinero generado por la plataforma.

Ganacias Mensuales Youtube


*Gráfica de ganancias mensuales en Youtube*

Para contextualizar estas cifras, se va a explicar inicialmente que Youtube paga en función a las visitas obtenidas, es decir, cuantas más visitas se logren en un mes, más dinero se percibe, es una relación directamente proporcional (sin tener en cuenta el nicho o CPM del canal en cuestión, ni el mes específico ya que muchos anunciantes finalizan los contratos de publicidad con este tipo de plataformas los meses de enero o junio), y las visitas van condicionadas por los clicks que genere dicho vídeo, es decir , que los usuarios estén interesados por tu producto, y un vídeo al estar constituido por un título y una miniatura (que son las partes visibles antes de entrar a este) es lo que ofrece *Vidmentor*.

*Vidmentor* es una plataforma gratuita que ofrece títulos con un alto potencial viral y miniaturas relacionadas al mismo para que desde su subida pueda tener un alto alcance viral y mediático. A su vez, ofrece una amplia gama de herramientas y recursos organizativos que guían a los usuarios a través del proceso de creación de contenido, optimización de vídeos y estrategias de crecimiento de audiencia. Al brindar estos servicios sin costo, se asegura una mayor captación de clientes, permitiendo a más futuros creadores a explorar y desarrollar sus habilidades en la creación de contenido digital.

## Análisis de la situación actual

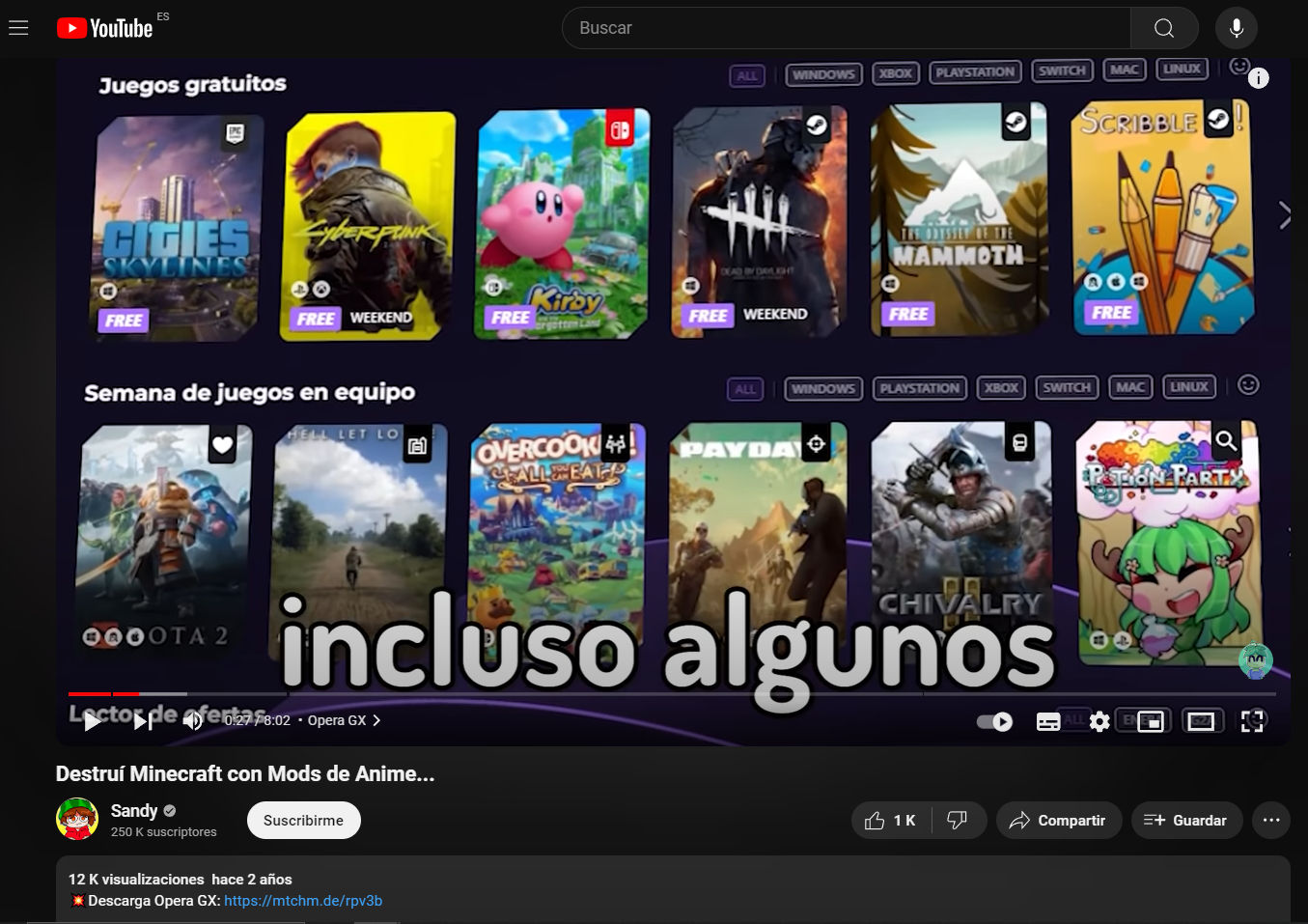
El objetivo de *Vidmentor* se define como proporcionar una plataforma integral donde se generan títulos con alto potencial mediático y viral, acompañados de miniaturas atractivas y acordes. La realización de guiones optimizados y la implementación de métodos para aumentar las visitas después de la subida del vídeo son facilitadas. Adicionalmente, se ofrece la posibilidad de distribuir el contenido en un calendario bien estructurado. Se incluye también una *landing page* que muestra tanto a los *influencers* que utilizan el software como a las marcas que han colaborado con *Vidmentor*, destacando la efectividad y el alcance de la plataforma. Así, se garantiza que los usuarios dispongan de las herramientas necesarias para maximizar el impacto de sus contenidos en las redes sociales.

Se reconoce que el mercado cuenta con varias plataformas especializadas que ofrecen herramientas valiosas para los creadores de contenido. Por ejemplo, [www.ryrob.com](file:///C:\Users\amateos.GOURMET\AppData\Local\Microsoft\Windows\INetCache\IE\M5C733W3\www.ryrob.com) se enfoca en la generación de títulos optimizados para captar la atención del público. [Simplified.com](file:///C:\Users\amateos.GOURMET\AppData\Local\Microsoft\Windows\INetCache\IE\M5C733W3\simplified.com), por su parte, se dedica a la creación de miniaturas online, proporcionando diseños atractivos y personalizables que ayudan a mejorar la visibilidad de los vídeos. Además, existen templates especializados de [Notion](https://www.notion.so/) que facilitan la organización y planificación del contenido, permitiendo a los usuarios estructurar sus ideas y proyectos de manera eficiente.

Aunque cada una de estas plataformas ofrece funcionalidades específicas, la fragmentación del proceso de creación y gestión de contenido puede resultar en una experiencia menos eficiente para los creadores. En este sentido, *Vidmentor* se presenta como una competencia realista frente a otros softwares y herramientas disponibles en el mercado. Al aunar todos estos conceptos en una única plataforma integral, *Vidmentor* ofrece una solución completa y cohesiva para los creadores de contenido. Desde la generación de títulos y la creación de miniaturas hasta la planificación y distribución del contenido, *Vidmentor* simplifica y optimiza cada paso del proceso. Esta integración de funcionalidades clave permite a los usuarios gestionar todo su flujo de trabajo de manera más fluida y eficiente, lo que lo convierte en una opción competitiva y atractiva en el mercado digital.

*Vidmentor* ha logrado consolidarse como una plataforma destacada en el mundo del contenido digital, en gran parte gracias al respaldo de influyentes *YouTubers* como *Sandy*, quien ha utilizado la plataforma para potenciar su presencia en línea. *Sandy*, en sus propias palabras, ha destacado: "Gracias a su plataforma y a su equipo de trabajo, he podido dar un salto de calidad en mi contenido". Esta afirmación evidencia el impacto positivo que *Vidmentor* ha tenido en la carrera de creadores de contenido como *Sandy*, permitiéndoles mejorar la calidad y el alcance de sus vídeos.

Además del respaldo de figuras prominentes como Sandy, *Vidmentor* ha logrado establecer sólidas relaciones con marcas reconocidas en la industria, como *Opera* y *Filmora*, entre otras. Estas marcas han confiado en *Vidmentor* como una plataforma efectiva para colaborar con creadores de contenido, lo que ha resultado en contratos exitosos y colaboraciones fructíferas. El testimonio de *Sandy* y la asociación con marcas de renombre han sido prueba del valor y la efectividad de *Vidmentor* en el panorama del contenido digital, reforzando su posición como una herramienta integral y relevante para creadores y marcas por igual.



*Vídeo de Sandy en Colaboración con Opera GX*

## Necesidades del cliente y oportunidad de negocio

Como contexto funcional se cabe mencionar que *Vidmentor* es una aplicación web que tiene como público objetivo personas que hacen uso habitual de las redes sociales y de contenido que le ofrece un valor informacional al usuario, por ello, este software va dedicado a este perfil de personas, personas mayoritariamente sin pareja, con una gran cantidad de tiempo libre, ideales emprendedores y ganas de aprender en nuevas disciplinas como SEO, scripting o edición. Este público al que se dirige son tanto hombres como mujeres entre 14 y 35 años de edad de cualquier nivel socioeconómico.



*Ejemplo 1 de público objetivo*

Una captura de pantalla de un celular con la imagen de un hombre

Descripción generada automáticamente con confianza media

*Ejemplo 2 de público objetivo*

En un escenario donde la influencia digital se convierte en una meta deseada, se destaca que uno de cada tres jóvenes en España anhela alcanzar el estatus de *influencer*, según indica un estudio del Centro Reina Sofía sobre Adolescencia y Juventud3. Este deseo es evidente, pero ¿cómo se enfrenta la carencia de conocimiento sobre cómo dar los primeros pasos en esta carrera?

La realidad es que, a pesar de la abundancia de ambición, muchos aspirantes encuentran confusa y desalentadora la ruta hacia el éxito en el mundo digital. La falta de una comprensión clara sobre cómo comenzar y en qué área enfocarse es notable. En contraste, en China, donde los jóvenes muestran preferencia por carreras más pragmáticas como la nutrición, la situación es distinta.

El fenómeno de los jóvenes españoles que anhelan convertirse en *influencers* resalta una percepción de *glamour* y éxito asociada con esta profesión emergente. Sin embargo, tras las apariencias glamorosas, se encuentra un trabajo arduo y una estrategia meticulosa que no todos comprenden.

El camino hacia el estrellato en línea no es sencillo y requiere no solo una comprensión profunda de las plataformas digitales y las tendencias culturales, sino también una voz auténtica y única que resuene con una audiencia específica. Además, la consistencia y la paciencia son clave en un mundo donde la competencia es intensa y los algoritmos cambian constantemente.

Por tanto, la falta de conocimiento sobre cómo adentrarse en el mundo de la influencia digital es una deficiencia que debe ser abordada, y como se ha explicado anteriormente, al no existir una plataforma que aúne los *softwares* relatados anteriormente *Vidmentor* ofrece una solución única para todas estas personas que se encuentran con dicho problema, ergo, no existen duplicidades en el producto.

Vidmentor está siendo preparado para expandirse a más idiomas además del español, lo cual permitirá llegar a un público más amplio y diverso. En paralelo, se está desarrollaría un algoritmo más complejo para la generación de ideas, con el objetivo de ofrecer propuestas más creativas y ajustadas a las necesidades específicas de los usuarios. Este nuevo algoritmo busca incrementar la relevancia y la utilidad del software, facilitando la creación de contenidos atractivos y originales.

3 *Según un estudio del Centro Reina Sofía sobre Adolescencia y Juventud, 1 de cada 3 menores quiere ser influencer. Los niños españoles pasan de media 4 horas al día ante pantallas, de las cuales 80 minutos las dedican a redes sociales o plataformas de vídeo online, según el último informe de Qustodio.”* Transmitido vía *La Vanguardia* en la noticia*:* [https://www.lavanguardia.com/tecnologia/20220727/8432422/que-quieres-mayor-1-3-menores-espana-claro-influencer-pmv.html#:~:text=Seg%C3%BAn%20un%20estudio%20del%20Centro,el%20%C3%BAltimo%20informe%20de%20Qustodio.](https://www.lavanguardia.com/tecnologia/20220727/8432422/que-quieres-mayor-1-3-menores-espana-claro-influencer-pmv.html%23:~:text=Seg%C3%BAn%20un%20estudio%20del%20Centro,el%20%C3%BAltimo%20informe%20de%20Qustodio.)

La inclusión de inteligencia artificial en Vidmentor también sería una idea de futuro desarrollo, con la finalidad de ofrecer miniaturas completamente libres de uso, diseñadas para captar la atención de los espectadores y mejorar la apariencia visual de los vídeos. Además, se está trabajaría en una inteligencia artificial especializada en la generación de etiquetas o tags, optimizadas para el posicionamiento de vídeos en las plataformas de vídeo. Estas etiquetas ayudarán a los creadores de contenido a aumentar su visibilidad y alcance, mejorando su presencia en los motores de búsqueda y recomendación de vídeos.

Estas mejoras tecnológicas y funcionales tienen como objetivo la captación de más clientes, al ofrecer un conjunto de herramientas más robusto y versátil que satisfaga las necesidades crecientes de los creadores de contenido. Con una base de usuarios más amplia y satisfecha, Vidmentor podrá consolidar su posición en el mercado y atraer a nuevos usuarios interesados en optimizar su proceso de creación de contenido multimedia.

Además, se está contemplando la creación de una nueva sección de *management*, destinada a los *influencers* que utilicen la plataforma. Esta sección permitirá a los *influencers* solicitar servicios de gestión y representación, facilitando el contacto con marcas interesadas en colaboraciones y patrocinios. A través de esta nueva línea de negocio, tanto los *influencers* como *Vidmentor* podrán generar ingresos adicionales. Los *influencers* tendrán acceso a oportunidades comerciales que les permitirán monetizar su contenido de manera más efectiva, mientras que *Vidmentor* podrá beneficiarse de las comisiones y acuerdos establecidos.

Para mejorar la competitividad en el mercado, se trabajaría en perfilar y desarrollar con precisión todos los cambios y mejoras mencionados anteriormente. Cada nueva funcionalidad y mejora estaría siendo cuidadosamente diseñada e implementada, con el objetivo de ofrecer un producto de alta calidad que satisfaga las expectativas de los usuarios y se mantenga a la vanguardia de las tendencias tecnológicas y de mercado.

En un principio, no serían establecidas oficinas físicas para Vidmentor, ya que la operativa inicial se realizaría completamente en modalidad de teletrabajo. Esta decisión se fundamenta en la flexibilidad y eficiencia que el teletrabajo ofrece, permitiendo a todos los miembros del equipo trabajar desde cualquier ubicación sin la necesidad de una oficina física centralizada.

No obstante, en caso de que se implemente la sección de *management* mencionada anteriormente, se considerarían instauradas unas oficinas de pequeña empresa en Madrid capital. Esta sección de *management* tiene como objetivo proporcionar servicios de gestión y representación a los *influencers*, facilitando el contacto con marcas y patrocinadores. Dado que este tipo de actividades podría requerir un espacio físico para reuniones y coordinación, se ha planteado la posibilidad de establecer dichas oficinas.

Aunque los clientes de Vidmentor serían de toda el área hispanohablante, se considera que la ubicación más lógica para las oficinas sería Madrid, la capital española. Madrid se considera el lugar más céntrico y accesible dentro del contexto del mundo hispanohablante, lo que facilitaría tanto las interacciones comerciales como la atracción de talento y la colaboración con otras empresas y entidades.

Estas oficinas de pequeña empresa en Madrid permitirían a *Vidmentor* contar con un espacio físico adecuado para la gestión de la sección de *management*, así como para posibles reuniones y eventos relacionados con el crecimiento y expansión de la empresa. Además, esta ubicación estratégica en la capital española facilitaría la creación de una red de contactos y colaboradores clave en el sector.

Estos planes a futuro reflejan la capacidad de *Vidmentor* para adaptarse a las necesidades emergentes del mercado y de sus operaciones, combinando la eficiencia del teletrabajo con la posibilidad de contar con un espacio físico cuando sea necesario para actividades específicas. La instauración de oficinas en Madrid sería un paso importante en la evolución de la empresa, permitiendo una gestión más efectiva y cercana de la nueva línea de negocio, al tiempo que se mantiene la flexibilidad operativa que caracteriza a *Vidmentor*.

## El nuevo proyecto: Vidmentor

**2.3.1Tipo de proyecto**

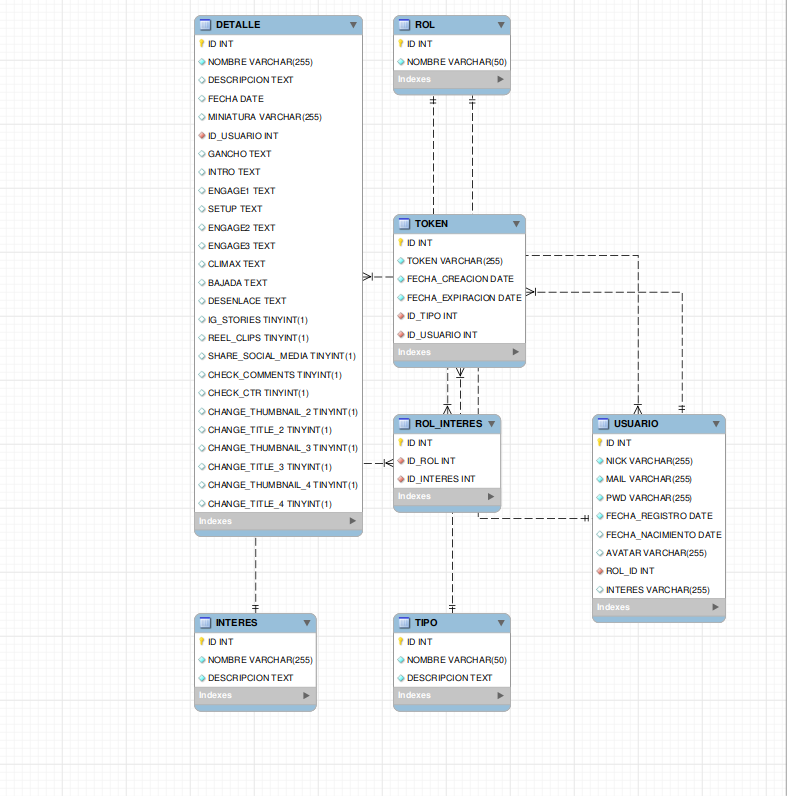
Como resumen de lo anteriormente mencionado, se daría respuesta a las necesidades detectadas mediante varias iniciativas estratégicas. En primer lugar, *Vidmentor* se expandiría a más idiomas para atender a un público más amplio. Se desarrollaría un algoritmo más complejo para mejorar la generación de ideas, asegurando propuestas más creativas y relevantes para los usuarios.

La inteligencia artificial se integraría para ofrecer miniaturas libres de uso y etiquetas optimizadas para el posicionamiento de vídeos, lo cual mejoraría la visibilidad y el atractivo del contenido creado por los usuarios. Estas mejoras tecnológicas facilitarían la captación de más clientes, proporcionando herramientas avanzadas que satisfacen sus necesidades.

Para abordar nuevas líneas de negocio, se implementaría una sección de *management* que los *influencers* podrían solicitar. Esta sección permitiría conectar a los *influencers* con marcas, creando oportunidades de monetización y generando ingresos tanto para los *influencers* como para *Vidmentor*. La implementación de oficinas en Madrid se consideraría para gestionar esta sección de manera más efectiva, dado su carácter central y accesible. Con todo ello se darían respuesta a las necesidades detectadas en los puntos anteriores.

Para concretar el tipo de proyecto a desarrollar en la empresa hay que hacer la diferenciación de sus distintos apartados para el desarrollo del software, el cual primero de ellos sería la base de datos.

En el diseño de la base de datos para el proyecto, se ha considerado esencial la creación de varias tablas interrelacionadas para gestionar eficientemente la información. Se describirá el mapa conceptual de la base de datos según las siguientes tablas y relaciones:



*Modelo entidad relación de la base de datos*

A continuación se detallarán cada una de las tablas del modelo:

**Tabla USUARIO**

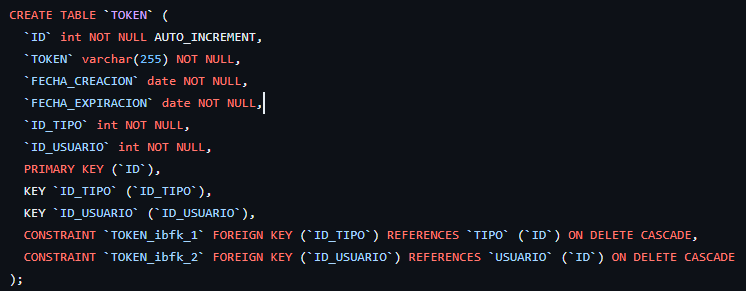
* Propósito: Guardar información detallada de cada usuario de la plataforma proporcionado inicialmente desde el registro de usuarios o creación de cuenta.
* Campos:
  + ID: Identificador único de cada usuario (autoincremental).
  + NICK: Apodo o nombre de usuario.
  + MAIL: Dirección de correo electrónico del usuario.
  + PWD: Contraseña del usuario.
  + FECHA\_REGISTRO: Fecha en que el usuario se registró en la plataforma.
  + FECHA\_NACIMIENTO: Fecha de nacimiento del usuario.
  + AVATAR: URL de la imagen de perfil del usuario, por defecto 'default.jpg'.
  + ROL\_ID: Identificador del rol asignado al usuario, referenciado desde la tabla ROL.
  + INTERES: Intereses del usuario, guardados como una cadena de texto, este
* Relaciones: Se relaciona con la tabla ROL mediante ROL\_ID y con varias otras tablas como TOKEN y DETALLE para vincular tokens y detalles de contenido a los usuarios.

TableUsuarios


*Tabla de Usuario en Base de Datos*

**Tabla TOKEN**

* Propósito: Gestionar los tokens de autenticación y otros tipos de tokens utilizados en la plataforma para la regulación de creación de usuarios, sesiones y cambios de contraseña, entre otras cosas.
* Campos:
  + ID: Identificador único de cada token (autoincremental).
  + TOKEN: Cadena de texto que representa el token.
  + FECHA\_CREACION: Fecha en que el token fue creado.
  + FECHA\_EXPIRACION: Fecha en que el token expira.
  + ID\_TIPO: Identificador del tipo de token, referenciado desde la tabla TIPO.
  + ID\_USUARIO: Identificador del usuario asociado al token, referenciado desde la tabla USUARIO.
* Relaciones: Se relaciona con las tablas TIPO y USUARIO mediante las claves foráneas ID\_TIPO y ID\_USUARIO.

 *Tabla de Token en Base de Datos*

**Tabla TIPO**

* Propósito: Definir los diferentes tipos de tokens que pueden ser utilizados en la plataforma.
* Campos:
  + ID: Identificador único de cada tipo de token (autoincremental).
  + NOMBRE: Nombre del tipo de token.
  + DESCRIPCION: Descripción detallada del tipo de token.
* Relaciones: Se utiliza como clave foránea en la tabla TOKEN para especificar el tipo de cada token.

Texto

Descripción generada automáticamente

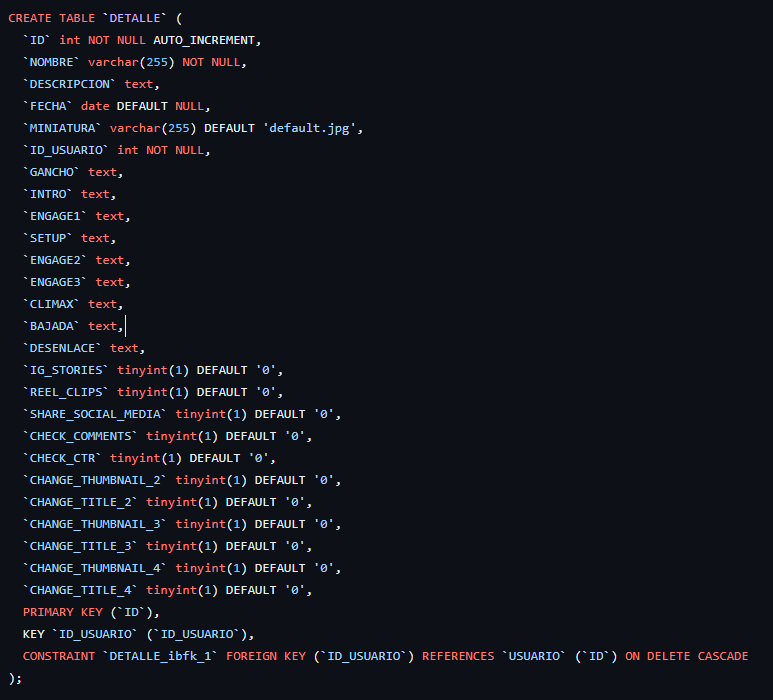
*Tabla de Tipo en Base de Datos*

**Tabla ROL**

* Propósito: Almacenar los diferentes roles que pueden asignarse a los usuarios dentro de la plataforma, en principio solo sería “usuario”, pero como se ha explicado anteriormente la idea sería que existieran “usuario”, “admin” y “empresa”.
* Campos:
  + ID: Identificador único de cada rol (autoincremental).
  + NOMBRE: Nombre descriptivo del rol.
* Relaciones: Se utiliza como clave foránea en la tabla USUARIO para asignar un rol específico a cada usuario.

Tabla Rol
 *Tabla de Rol en Base de Datos*

**Tabla DETALLE**

* Propósito: Almacenar los detalles de los vídeos y guiones creados por los usuarios.
* Campos:
  + ID: Identificador único de cada detalle (autoincremental).
  + NOMBRE: Nombre del vídeo o guion.
  + DESCRIPCION: Descripción del contenido.
  + FECHA: Fecha de publicación del contenido.
  + MINIATURA: URL de la miniatura del contenido, por defecto 'default.jpg'.
  + ID\_USUARIO: Identificador del usuario que creó el contenido, referenciado desde la tabla USUARIO.
  + GANCHO, INTRO, ENGAGE1, SETUP, ENGAGE2, ENGAGE3, CLIMAX, BAJADA, DESENLACE: Diferentes segmentos del guion.
  + Varias columnas booleanas (IG\_STORIES, REEL\_CLIPS, SHARE\_SOCIAL\_MEDIA, CHECK\_COMMENTS, CHECK\_CTR, CHANGE\_THUMBNAIL\_2, CHANGE\_TITLE\_2, CHANGE\_THUMBNAIL\_3, CHANGE\_TITLE\_3, CHANGE\_THUMBNAIL\_4, CHANGE\_TITLE\_4) para opciones de gestión del contenido en redes sociales.
* Relaciones: Se relaciona con la tabla USUARIO mediante ID\_USUARIO.

*Tabla Detalle de la Base de Datos*

Como se ha podido observar en el modelo Entidad-Relación existen dos tablas adicionales denominadas “INTERESES” y “ROL\_INTERESES” las cuales a efectos prácticos no realizan ninguna función dentro de la aplicación, sin embargo, como se ha mencionado anteriormente, se quiere llevar el algoritmo de recomendaciones a algo más complejo y este sería el primer cambio que se llevaría a cabo, sin embargo, a efectos prácticos, todavía no tienen funcionalidad.

Texto

Descripción generada automáticamente Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

*Tabla Intereses de la Base de Datos Tabla Rol Intereses de la Base de Datos*

Concretar además el tipo de proyecto a desarrollar en la empresa:

* Implantación de software
* Implantación de sistemas en red
* Seguridad de redes
* Desarrollo de un sistema de información...

**2.3.2 Características requeridas al proyecto**

El proyecto se centra en abordar la falta de conocimiento y orientación que enfrentan los jóvenes aspirantes a *influencers*, especialmente en España, donde uno de cada tres jóvenes aspira a este estatus.

En términos de qué tiene que hacer el proyecto, su objetivo principal es proporcionar recursos y guías claras sobre cómo dar los primeros pasos en el camino hacia la influencia digital. Esto incluiría desde guías prácticos sobre la creación de contenido hasta estrategias de construcción de marca personal y audiencia las cuales modifica libremente el usuario mediante su *dashboard*.

Los elementos diferenciadores del proyecto radican en su enfoque integral y accesible. En lugar de simplemente ofrecer consejos genéricos, el proyecto se esfuerza por adaptar la información a las necesidades y aspiraciones específicas de los jóvenes aspirantes a *influencers*. En Vidmentor se han realizado estudios específicos de cómo funciona a día de hoy la aplicación de Youtube, se ha estudiado minuciosamente el algoritmo de recomendación de vídeos para saber qué táctica se ha de seguir en cuanto al SEO, se han revisado cientos de canales para valorar su contenido y sus estadísticas, comparándolo no solo los canales hispanohablantes, sino los angloparlantes para poder analizar los casos de éxito que ha habido en cuanto a ellos y poder saber cuánto es el nivel de ingresos a los que estos canales son sometidos, llegando a la conclusión que todo se basaba en ideas llamativas e innovadoras para el público hispanohablante, los cuales son copias o inspiraciones de los *Youtubers* de habla inglesa.

[](https://www.youtube.com/embed/NMnJaO4Qhrg?feature=oembed) *Listado de canales estudiados*

*Vidmentor* también se distinguiría por su enfoque práctico, proporcionando herramientas y recursos tangibles que los jóvenes puedan utilizar para desarrollar sus habilidades y su presencia en línea de manera efectiva. Esto podría incluir desde herramientas de estructuración de vídeo para que el producto final pueda tener una mayor retención de audiencia, hasta elementos que pueden llevar a que si un vídeo ha dado comienzo con unos números bajos, puedan volver a alzarse y alcanzar las visitas y horas de visualización estimadas con el mismo, con el objetivo de brindar orientación práctica y apoyo personalizado. A continuación se muestra una imagen la cual explica de manera más gráfica como una pieza de contenido audiovisual ha de ser distribuida, ya puede ser para un vídeo para redes sociales o incluso una película profesional:

Gráfico de exposición de una historia


*Gráfico de exposición de organización de una pieza*

Con todos estos conceptos que poca gente sabe y haciendo unos estudios minuciosos con ello, se pueden verificar fácilmente los elementos diferenciadores que ofrece *Vidmentor* frente a su competencia, siendo todos ellos orientados a un entorno específico que sería la creación de contenido audiovisual para la plataforma Youtube, plataforma la cual es la que mejor monetiza el contenido subido a su propia plataforma y siendo la segunda plataforma más utilizada a nivel mundial con 2.491 millones de usuarios y un billón de horas de vídeo diarias4 y todo ello con un constante crecimiento y avalado por Google.

 *Listado de redes sociales más usadas*

4 *Según un estudio de documentado por Sheila Martín para el diario MKParadise, aquí se puede constantar la noticia:* [*https://mkparadise.com/redes-sociales-mas-utilizadas#2\_YouTube\_2491\_millones\_de\_usuarios*](https://mkparadise.com/redes-sociales-mas-utilizadas%232_YouTube_2491_millones_de_usuarios)

Para todo ello se ha tenido que hacer uso de varias tecnologías para poder desarrollar con exactitud las ideas que propone *Vidmentor*, para ello se han listado todas dando justificación de su uso:

* MySQL: fue seleccionado como el sistema de gestión de bases de datos principal debido a su robustez, rendimiento y amplia compatibilidad con otras tecnologías. Nos permitió almacenar y organizar eficientemente una gran cantidad de datos relacionados con usuarios, configuraciones, contenido generado por usuarios y otra información crítica para el funcionamiento del software. Además, su capacidad para gestionar transacciones de manera segura garantizó la integridad de los datos en todo momento.
* **PHP**: se utilizó como el lenguaje de programación principal para el desarrollo del *backend* del software. Gracias a su amplia disponibilidad de recursos, su facilidad de aprendizaje y su compatibilidad con diferentes sistemas operativos y servidores web, PHP se mostró como una opción sólida para la implementación de la lógica del negocio. Desde la gestión de usuarios hasta la manipulación de datos en la base de datos MySQL, PHP nos permitió crear una aplicación web dinámica y escalable.
* **HTML, JS, CSS**: Estas tecnologías formaron la columna vertebral del desarrollo *frontend* de nuestro software. HTML proporcionó la estructura básica para nuestras páginas web, permitiéndonos organizar y presentar contenido de manera semántica y accesible. CSS fue esencial para estilizar y diseñar la interfaz de usuario, asegurando una experiencia visualmente atractiva y coherente en todas las plataformas y dispositivos. JavaScript, por su parte, agregó interactividad y dinamismo a nuestras páginas, permitiendo a su vez dar uso a las distintas *APIs* o tecnologías que se van a explicar a continuación.
* YouTube API: La integración de la *YouTube API* fue fundamental para enriquecer la funcionalidad de nuestro software con respecto al contenido de vídeo. Nos permitió acceder y manipular datos directamente desde la plataforma de YouTube, lo que posibilitó funciones como la búsqueda a través de los intereses del usuario para la recomendación de títulos e ideas acordes con su contenido y poder filtrar las miniaturas por este para poder ofrecer con exactitud un producto altamente viral, pudiendo filtrar todo por los vídeos realizados por la comunidad angloparlante.
* mymemory.translated: Esta *API* fue seleccionada para proporcionar funcionalidades de traducción automática dentro de nuestro software. Al integrarse, permitió que las ideas generadas por la *Youtube API* pudieran ser traducidas de una manera eficiente y realista y se pudiera traer el contenido de habla inglesa a la comunidad hispanohablante.
* CKEditor: La integración de CKEditor en *Vidmentor* permitió ofrecer a los usuarios una experiencia de edición de texto enriquecida y familiar. Con CKEditor, los usuarios pueden crear y editar contenido de manera similar a como lo harían en un procesador de texto convencional, con funciones avanzadas como el formato de texto, la inserción de imágenes y la gestión de enlaces, lo que facilita la creación de contenido atractivo y de calidad.
* Tailwind: se utilizó como *framework* de CSS para agilizar el desarrollo *frontend* y mantener un diseño visual coherente en todo el software. Su enfoque en clases utilitarias permitió construir rápidamente diseños flexibles y responsivos sin tener que escribir una gran cantidad de CSS personalizado. Además, su configuración predefinida y su extensa documentación facilitaron la colaboración entre el equipo de desarrollo *frontend*.
* Photoshop: desempeñó un papel crucial en el diseño visual del software, especialmente en la creación del logotipo y la producción de imágenes personalizadas que se tuvieron que dar uso en situaciones puntuales de la web para dar un enfoque más profesional y mayor sensación de cercanía al usuario con respecto a su objetivo con YouTube.
* Trello: para la gestión eficiente de tareas y la organización del equipo, se empleó Trello como una herramienta colaborativa. Con Trello, se pudieron crear tableros personalizados donde se podía visualizar y seguir el progreso de las tareas en tiempo real. La capacidad de asignar tareas, establecer fechas límite y adjuntar archivos relevantes facilitó la coordinación y la colaboración entre los miembros del equipo, asegurando que cada tarea fuera completada de manera oportuna y eficiente, como se han podido ver en las imágenes expuestas en la introducción.
* Slack: para mantener una comunicación fluida y efectiva entre los miembros del equipo, se utilizó *Slack* como la plataforma principal de mensajería instantánea. *Slack* ofreció canales de comunicación dedicados para diferentes equipos y proyectos, lo que permitió organizar las discusiones de manera estructurada y evitar la pérdida de información.
* Github: para poder trabajar los distintos miembros desarrolladores en conjunto y mantener un orden en cuanto al desarrollo se utilizó Github para tener un repositorio común.

Después se hizo uso de otras herramientas como *Whatsapp* para el contacto, o *Discord* por sus comodidades a la hora de poder retransmitir en vivo la propia pantalla del ordenador, *Cinema 4D* para pruebas de renderizado de logo y demás tecnologías, sin embargo, al no haber tenido tanta presencia en el desarrollo o el producto final solo se han mencionado y no se dará una explicación amplia de su uso debido a que se dieron uso en casos puntuales y de corta duración.

En una primera aproximación a los recursos tanto humanos como materiales necesarios para el desarrollo del proyecto, se estima que sería necesario contar con un equipo compuesto por al menos tres personas. Este equipo estaría encargado de llevar a cabo las diferentes etapas del desarrollo del software, incluyendo el diseño, la programación, la gestión de bases de datos y la integración de APIs, entre otras tareas, las cuales serían distribuidas en, una persona encargada de realizar el *frontend* de la página web, otra encargada del desarrollo del *backend* y el despliegue del software y otra encargada de dirigir al equipo, gestionar el uso de las *APIs* y recursos y poder ser polivalente en cuanto al *backend* como *frontend*.

En términos de recursos materiales, cada miembro del equipo requeriría al menos un ordenador con acceso a Internet mediante conexión WiFi. Estos ordenadores servirían como herramientas fundamentales para el desarrollo del software, permitiendo a los desarrolladores escribir código, probar funcionalidades y realizar otras tareas relacionadas con el proyecto. Además, sería necesario contar con una instalación de red *WiFi* que proporcione conexión a Internet a los ordenadores del equipo. Esta instalación incluiría un *router* *WiFi* que permita la conexión inalámbrica de los dispositivos al acceso a Internet. El *router* sería el centro de la red y proporcionaría la conectividad necesaria para desplegar la plataforma y acceder a recursos externos, como *APIs* y servicios en la nube.

**2.3.3 Obligaciones fiscales, laborales y de prevención de riesgo**

Dentro del marco del nuevo emprendimiento, resulta imperativo comprender detalladamente las principales obligaciones fiscales, laborales y de prevención de riesgos que emergen con su establecimiento y desarrollo. Estas responsabilidades constituyen una red interconectada de aspectos que demandan una gestión meticulosa, con el fin de asegurar tanto el cumplimiento normativo como la integridad y bienestar de los empleados y la propia empresa. En este sentido, la atención a estas obligaciones no solo es un requisito legal, sino un pilar fundamental para la sostenibilidad y el éxito a largo plazo del negocio en el entorno empresarial actual.

Como empresa de sociedad limitada dedicada a proporcionar un software gratuito que ofrece ideas de vídeos para YouTube, la empresa deberá inscribirse en el Registro Mercantil como sociedad limitada, lo que implica cumplir con los requisitos legales para la constitución de la empresa y la redacción de los estatutos sociales. Este proceso establece la base legal para la operación de la empresa y define su estructura y responsabilidades. En términos de obligaciones fiscales, la empresa estará sujeta al impuesto sobre sociedades, que grava los beneficios obtenidos durante el ejercicio fiscal. Es importante llevar una contabilidad adecuada y presentar declaraciones fiscales periódicas, cumpliendo con los plazos establecidos por la legislación fiscal.

Además del impuesto sobre sociedades, la empresa deberá considerar otras obligaciones fiscales, como el IVA si ofrece servicios o productos sujetos a este impuesto, en el caso de ofrecer el servicio de *management* mencionado con anterioridad. Si la empresa realiza transacciones con clientes en otros países de la Unión Europea, también deberá cumplir con las normativas fiscales relacionadas con el IVA intracomunitario, en el caso de la expansión a otros idiomas también mencionada anteriormente.

Otra obligación fiscal importante es la retención e ingreso a cuenta del impuesto sobre la renta de los trabajadores. Esto implica calcular y retener parte del salario de los trabajadores para luego ingresarlo en Hacienda. Además de estas obligaciones fiscales específicas, la empresa también deberá cumplir con otras obligaciones administrativas, como la presentación de cuentas anuales y el cumplimiento de las obligaciones laborales y de seguridad social, si tiene empleados.

Dentro del contexto de abordar las obligaciones laborales y de prevención de riesgos, la empresa deberá asegurarse de cumplir con las normativas laborales vigentes en cuanto a la contratación de personal. Esto incluye la elaboración de contratos de trabajo que definan claramente las condiciones laborales, los derechos y responsabilidades tanto de la empresa como de los empleados, así como el cumplimiento de los requisitos legales en cuanto a salario mínimo, jornada laboral, vacaciones y otros aspectos laborales relevantes. A su vez, al operar en régimen de teletrabajo, la empresa deberá prestar especial atención a la gestión del tiempo y la conciliación laboral. Esto implica establecer mecanismos efectivos de seguimiento y control de la jornada laboral, así como promover una cultura organizacional que fomente el equilibrio entre la vida laboral y personal de los empleados.

En cuanto a las obligaciones de prevención de riesgos laborales5, la empresa deberá proporcionar a los empleados equipos ergonómicos adecuados, como sillas y mesas ergonómicas, así como garantizar la seguridad de las conexiones informáticas y la protección de datos en el entorno de trabajo remoto. Asimismo, la empresa deberá ofrecer formación y capacitación en materia de prevención de riesgos laborales a los empleados que trabajan en régimen de teletrabajo, asegurándose de que estén familiarizados con los protocolos de seguridad y salud en el trabajo, así como con los procedimientos de actuación en caso de emergencia.

5 *Todo ello según la Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales publicado en el BOE de forma genérica, para el teletrabajo no se ha podido recopilar una información extensa debido a su inexistencia o que el formato de búsqueda ha sido escaso en cuanto a resultados. Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales:* [*https://www.boe.es/buscar/act.php?id=BOE-A-1995-24292*](https://www.boe.es/buscar/act.php?id=BOE-A-1995-24292)

**2.3.4 Ayudas/subvenciones**

Para la puesta en marcha del proyecto de una empresa de sociedad limitada de nueva creación, se pueden considerar diversas ayudas y subvenciones.

En primer lugar, se pueden solicitar subvenciones para la creación de empresas de base tecnológica. Estas ayudas, concedidas por el Ministerio de Economía y Competitividad, están destinadas a financiar los gastos de constitución y puesta en marcha de la empresa, así como para promover la innovación y la competitividad en el sector tecnológico. Estas subvenciones pueden cubrir una variedad de gastos, como los honorarios de abogados para la constitución de la empresa, los costos de registro mercantil, el desarrollo inicial del software, la adquisición de equipos informáticos y otros gastos relacionados con el inicio de la actividad empresarial.

Además, se pueden solicitar ayudas para la contratación de personal altamente cualificado. Estas subvenciones, concedidas por el Ministerio de Empleo y Seguridad Social, tienen como objetivo fomentar la contratación de personal con formación en áreas tecnológicas y digitales, como ingenieros informáticos, diseñadores de software y especialistas en marketing digital. Estas subvenciones pueden cubrir parte de los costos salariales de los nuevos empleados durante un período determinado, lo que puede suponer un alivio financiero significativo para la empresa durante los primeros años de actividad.

Por último, se pueden solicitar ayudas para el desarrollo de proyectos de I+D+i. Estas subvenciones, concedidas por el Centro para el Desarrollo Tecnológico Industrial (CDTI) y otras instituciones públicas y privadas, están destinadas a financiar proyectos de investigación, desarrollo e innovación en el ámbito de las tecnologías de la información y la comunicación. Estas subvenciones pueden cubrir una amplia gama de actividades, como la investigación de mercado, el diseño y desarrollo de nuevos productos y servicios, la mejora de la eficiencia operativa y la optimización de los procesos empresariales.

Es importante destacar que la concesión de estas ayudas está sujeta a la evaluación de la viabilidad y el potencial de crecimiento del proyecto empresarial.

# DISEÑO DEL PROYECTO

Dando por hecho la viabilidad del proyecto, en este apartado se concretarán las fases necesarias para llevarlo a cabo, y cumplir con los objetivos que se establezcan, teniendo en cuenta los recursos necesarios.

## Fases del proyecto

El desarrollo de este proyecto se llevará a cabo en cuatro fases: análisis, diseño, implementación y pruebas, que pasan a detallarse a continuación.

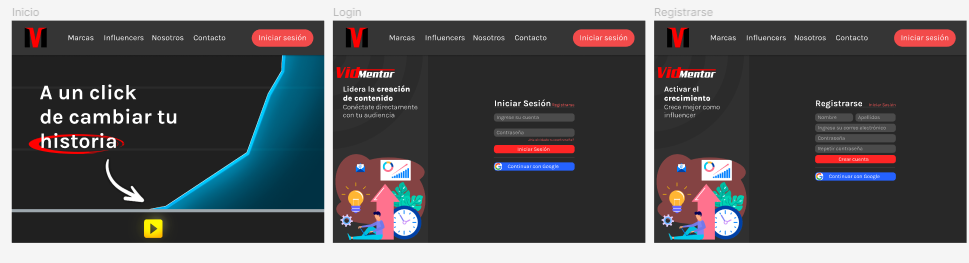
### Análisis

Estableciendo los requisitos del proyecto, se pueden distinguir entre los funcionales y los no funciones:

* Funcionales:
  + Registro de usuarios por formulario con autentificación de creación de cuenta.
  + Cambio de contraseña por token.
  + Formulario previo al *dashboard* para categorizar los intereses/temática del canal del usuario y pudiendo ser rehecho para cambiar la temática.
  + Generación de ideas extraídas de canales angloparlantes traducidas al español, las cuales pueden ser guardadas en el listado del usuario.
  + Vista de ideas guardadas tanto en vista general como de calendario, para poder realizar un *drag and drop*.
  + Formulario de detalle donde el usuario podrá guionizar su vídeo y se le ofrecerá las miniaturas más virales para poder replicar.
  + Formulario de detalle donde el usuario podrá marcar y desmarcar al gusto del usuario las distintas estrategias probadas en la post-subida.
  + *Landing page* mostrando marcas e *influencers* con los que Vidmentor ha trabajado.
* No funcionales:
  + Interfaz atractiva e intuitiva basada en el minimalismo, la accesibilidad y la ‘responsividad’.
  + Optimización del sitio web para mejorar el rendimiento del sitio web y la eficiencia en el procesamiento de datos.
  + Claridad y transparencia en la organización y estructuración de los contenidos de la aplicación, para proporcionarle al usuario una navegación eficaz y ágil.

### Diseño

Tanto la *landing page* como el *dashboard* del usuario son la cara visibles y principal del proyecto dónde los usuarios navegaran para poder acceder a las distintas funcionalidades que ofrece el software.

Por tanto, es importante tener claro su diseño, cómo será su estructura básica, para tener una idea igual de clara a la hora de ponerse a desarrollar su interfaz, para ello se ha hecho uso del servicio gratuito que ofrece *figma*, a la hora de poder realizar prototipos del portal web para poder dar una aproximación visual del producto que se está buscando para luego ser desarrollado, aunque sea susceptible a cambios. Esta tecnología permitió crear las distintas pestañas de la página web para agilizar el proceso de desarrollo.  


*Prototipo en figma*

Debido a su diseño previo, ya se podía hacer una distinción de las distintas pestañas a desarrollar:

* *Landing page* (Inicio)
* Inicio de sesión
* Registro de usuarios
* *Dashboard*, el cual incluye:
  + Generación de ideas
  + Distribución de ideas guardadas
  + Formularios de detalle
  + Formulario de intereses
  + Modificar perfil

Una vez definidas sus distintas pestañas se podía dar comienzo con el desarrollo del diseño, para ello, a continuación, se explicarán una a una sus distintas partes, pero antes se va a detallar la guía de estilos y colores seleccionada:

**GUÍA DE ESTILOS**

Para la selección de colores se ha utilizado la misma paleta de colores que utiliza Youtube en su modo oscuro, tanto los colores grises como los rojos, y para los colores azules, se han seleccionado debido a los colores que ofrecen las gráficas de estadísticas de Youtube.

Logotipo, nombre de la empresa

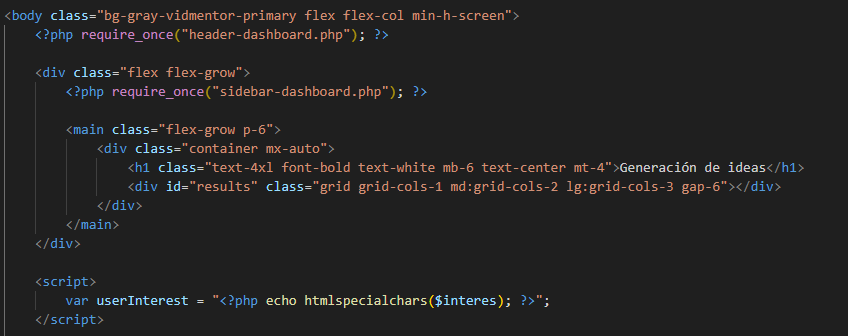
Descripción generada automáticamente

*Guía de estilos de colores*

**GENERACIÓN DE IDEAS**

En el siguiente código se puede observar como en la etiqueta body se están dando uso a etiquetas de las librerías de *Tailwind* para dotar de propiedades de responsividad y tamaño al cuerpo de la web, y clases propias para hacer uso de la paleta de colores que corresponde con *Vidmentor*. Acto seguido se puede observar que, mediante php, se hace un llamamiento para que se impriman tanto el *header* como el *sidebar*, ya que son archivos independientes que son comunes en todas las pestañas del *dashboard*. Seguidamente encontramos distintas etiquetas que tienen como clases las propias de *Tailwind* para dotar de responsividad y colores los distintos componentes de la web, hasta llegar a la etiqueta id con id “results”, donde se puede observar la clase grid, para poder crear una cuadrícula, para poder disponer los distintos elementos (los cuales serán explicados más adelante debido a que son generados por Javascript) y al lado del número de columnas que genera usa serie de letras (“md” y “lg”, esto es debido a que, según el tamaño que adopte la pantalla donde se está visualizando la web, serán distinto el número de columnas que acepte), todo ello con un gap de 24px.

En cuanto a la etiqueta script, para poder pasar valores desde php a Javascript se está haciendo uso de la creación de dicha variable en el propio archivo php para que la forma de pasar esos valores sea más cómoda y rápida.



*Código HTML de ideas.php*

Una vez cargado el html, se puede observar que se hace uso de la función “search”, en la cual se pasa como parámetro la variable mencionada anteriormente.

DOMContentLoaded ideas.php
 *Uso de JS para ideas.php*

En el siguiente código se observa cómo se utiliza JavaScript y jQuery para realizar una búsqueda en la API de YouTube y procesar los resultados obtenidos.

La función *search* se ha diseñado para aceptar un parámetro *query* que representa la consulta de búsqueda, en este caso los intereses obtenidos de la parte php. Se verifica si query no está vacío (query !== ''). Si la consulta no está vacía, se realiza una petición GET a la API de YouTube utilizando jQuery.

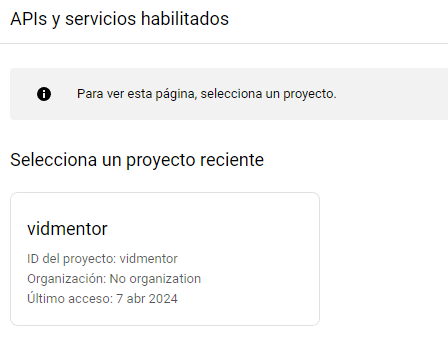
La llamada a la API se realiza con el método $.get, que envía una solicitud HTTP GET a la URL especificada (*'https://www.googleapis.com/youtube/v3/search*') con los parámetros proporcionados:

* *part*: *'snippet'*: Se especifica que se desean obtener los detalles del vídeo (como título, descripción, miniatura, etc.).
* *q*: *query*: Se pasa la consulta de búsqueda a la API, los intereses mencionados anteriormente.
* *key*: *'AIzaSyD8bPxnG\_Rr0v5bIok4iu8xAnjtOGR\_ZOM*': Se utiliza una clave API para autenticar la solicitud, más adelante, cuando se terminen de explicar los distintos campos se explicará como se obtuvo y las limitaciones que muestra. Se agradece que su visibilidad y uso sea limitada, ya que está vinculada a la cuenta de uno de los alumnos/desarrolladores.
* *maxResults*: *10*: Se limita el número de resultados a 10.
* *type*: *'video'*: Se especifica que se desean solo resultados de vídeos.
* *videoDuration*: *'long'*: Se filtran los resultados para incluir solo vídeos largos (formato horizontal).
* *regionCode*: *'US'*: Se especifica el código de región (Estados Unidos).
* *relevanceLanguage*: *'en'*: Se especifica el idioma de relevancia (inglés).

Texto

Descripción generada automáticamenteAl ser la solicitud es exitosa, se ejecuta la función proporcionada como segundo argumento de $.get, la cual recibe los datos devueltos por la *API* y llama a la función *showResults* con los elementos (*items*) de los datos recibidos.

*Función de búsqueda de ideas*

**

Para obtener una clave *API* de YouTube, primero se accedió a *Google Cloud Console*. Una vez dentro, se creó un nuevo proyecto. Después de configurar el proyecto, se habilitó la *API* de YouTube. Esto se realizó navegando a "*API* y Servicios" dentro del menú de Google Cloud Console, y luego a la "Biblioteca". Aquí, se buscó "YouTube Data *API* v3" y se habilitó para el proyecto en cuestión. Una vez habilitada la API, se crearon las credenciales necesarias para utilizarla. Esto se hizo yendo a "*API* y Servicios" y luego a "Credenciales". Se hizo clic en "Crear credenciales" y se seleccionó "Clave de API". Esto generó una clave API que se puede utilizar para realizar solicitudes a la *API* de YouTube.

*Google Cloud Console - Servicio API*

Esta clave es esencial para autenticar las solicitudes y acceder a los datos proporcionados por YouTube. Con la versión gratuita se tiene un límite de cuotas el cual funciona por créditos diarios consumidos, y cada información obtenida tiene un coste distinto de los mismos, para más información acerca del consumo de estos consultar: <https://developers.google.com/youtube/v3/determine_quota_cost?hl=es-419>

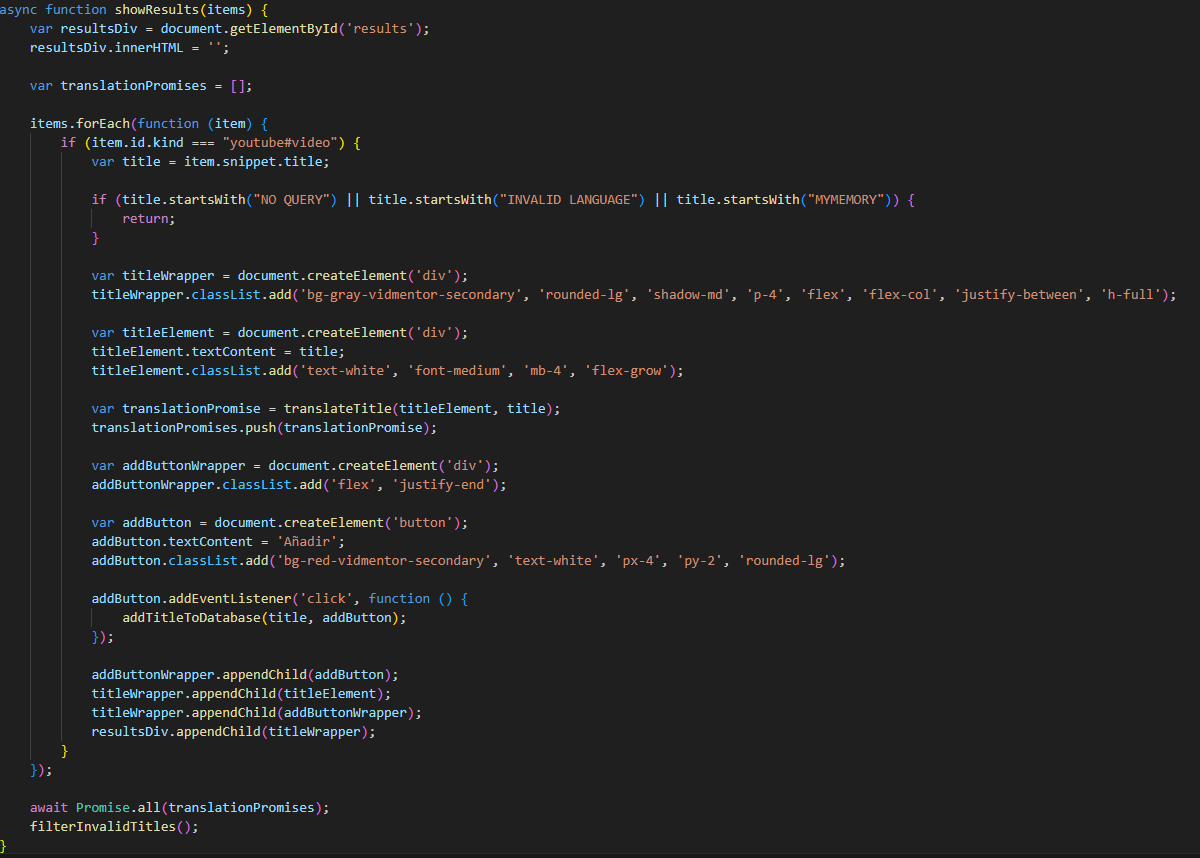
Continuando con la explicación del código de *front-end*, en la siguiente captura se puede observar la función *showResults,* dentro de la función, se comienza obteniendo una referencia al elemento HTML con el id *'results'*,mencionado con anterioridad para poder imprimir los resultados de la búsqueda . Luego, se limpia el contenido de este elemento utilizando *innerHTML*.

A continuación, se itera sobre cada elemento en el array *items*. Por cada elemento, se verifica si el tipo de vídeo es *"youtube#video".* Si no lo es, se ignora y se pasa al siguiente elemento.

Para los vídeos válidos, se extrae el título del vídeo desde *item.snippet.title*. Se realiza una serie de comprobaciones en el título para verificar si comienza con ciertas cadenas específicas (*'NO QUERY'*, *'INVALID LANGUAGE*' o *'MYMEMORY'*). Si el título comienza con alguna de estas cadenas, se ignora y se pasa al siguiente elemento. Esto es debido al servicio de traducción gratuito utilizado que puede dar tanto errores, como puede agotarse sus traducciones diarias gratuitas por lo tanto, se hacen comprobaciones para no mostrar dichos errores al usuario.

Para los títulos válidos, se crea un contenedor de título (*titleWrapper*) utilizando elementos HTML. Se añade el título como contenido de un elemento div y se le aplican estilos mediante clases CSS de Tailwind para dar una apariencia de “*cards*” al usuario. Además, se crea un botón "*Añadir*" que permite añadir el título a una base de datos cuando se hace clic en él (esta parte se detallará en la parte de implementación) con la función *addTitleToDatabase*.

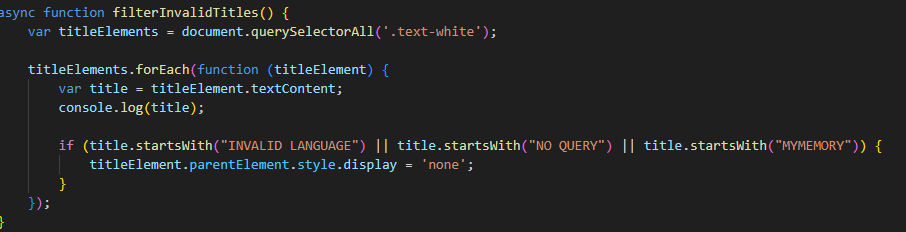
Finalmente, se espera a que todas las promesas de traducción (*translationPromises*) se resuelvan utilizando *Promise.all*. Después, se llama a la función *filterInvalidTitles* para filtrar los títulos inválidos como seguridad por los errores mencionados anteriormente que puede dar el traductor.

 *Función showResults() del script de ideas.php*

La función *filterInvalidTitles* es una función asincrónica que se encarga de filtrar y ocultar los títulos inválidos en la interfaz de usuario después de que se hayan mostrado los resultados de una búsqueda de vídeos.

Primero, la función selecciona todos los elementos con la clase CSS .text-white (ósea los títulos impresos con anterioridad), que se espera que sean los elementos que contienen los títulos de los vídeos mostrados en la página.

Luego, se itera sobre cada uno de estos elementos de título. Para cada elemento de título, se obtiene el texto del título y se comprueba si comienza con algunas cadenas las cuales hemos mencionado antes. Si el título comienza con alguna de estas cadenas, se oculta el contenedor del título estableciendo su estilo de visualización (*display*) a *'none'*.



*Función filterInvalidTitles() del scipt de ideas.php*

El resultado final sería el siguiente, tanto en versión *desktop*, como *mobile*:

Diseño ideas.php
Captura de pantalla de un celular

Descripción generada automáticamente

*Diseño final de ideas.php*

**IDEAS GUARDADAS Y CALENDARIO**

Al igual que con la anterior pestaña que se ha explicado, la etiqueta del tanto del body como las internas tienen los mismos valores que el anterior, ya que este diseño se decidió hacer por componentes para una mayor sencillez en el código, siendo estos *ideas\_guardadas.php* y calendar\_raw.php, los cuales se explicaran como subapartados de esta pestaña, los apartados de Javascript se detallarán de forma global, en vez de por los 2 apartados ya que existe un apartado *drag and drop* que involucra ambos archivos.

*Texto

Descripción generada automáticamente*

*Cuerpo HTML de calendar.php*

* **IDEAS GUARDADAS**

Debido a que la obtención de datos en este archivo forma parte del *back-end* que se hará la debida explicación en el apartado de implementación, como resumen de lo que se hace en dicha parte para saber lo que se va a imprimir aquí son todas las ideas guardadas en la base de datos asociadas al usuario.

Dentro de esta captura podemos observar que es el mismo sistema de *cards* que se ha utilizado anteriormente en la generación de ideas, sin embargo, este tiene la peculiaridad de tener 2 botones en él, “*Eliminar*” el cual eliminará la *card* de manera visible y actualizará la base de datos y “*Editar*” que abrirá el archivo *forms\_container*, pasando como parámetro el ID de la idea.

Captura de pantalla de computadora

Descripción generada automáticamente

*Cuerpo HTML impreso con PHP de calendar.php*

* **CALENDARIO *RAW***

En el siguiente código se puede observar como para la creación del calendario no se ha dado uso de las clases de Tailwind, sino que se ha optado por hacerlo en CSS Vanilla para una mayor comodidad en el desarrollo debido a la complejidad de su desarrollo.

Texto

Descripción generada automáticamenteEn el HTML se puede observar que únicamente son distintos contenedor, unos dentro de otros, donde se ve información de lo que sería un calendario, como el nombre del mes o los días de la semana, a excepción de dos botones que actuarán de flechas para poder ir pasando de meses.

*Contenido HTML de calendar\_raw.php*

Por la parte del Javascript, se define un array “*dayNames”* que contiene los nombres abreviados de los días de la semana. También se define un array “*monthNames”* que contiene los nombres de los meses del año.

Cuando el documento está listo, se obtiene la fecha actual mediante la creación de un objeto *Date*. Se extraen el mes y el año actuales utilizando los métodos “*getMonth*” y “*getFullYear*” del objeto Date, respectivamente.

Se intenta obtener una fecha guardada de las cookies utilizando la función “*getCookie”* con la clave "*savedDate*", las cuales se explicarán más adelante. Si se encuentra una fecha guardada, se divide la cadena en partes para extraer el año y el mes, los cuales se actualizan en las variables “*currentYear”* y “*currentMonth”*, para que el calendario se dirija al mes específico donde con anterioridad lo había dejado el usuario.

A continuación, se llama a la función “*renderCalendar”* con el mes y el año actuales para renderizar el calendario en la interfaz de usuario, esta función también se explicará seguidamente.

Se añade un evento de clic a los elementos con la clase “*.prev-month”,* botones mencionados anteriormente. Al hacer clic, se decrementa el mes actual. Si el mes actual es menor que 0, se establece a 11 (diciembre) y se decrementa el año actual. Luego, se vuelve a renderizar el calendario con el nuevo mes y año, y se guarda la fecha actualizada en las cookies utilizando la función “*saveDate”*.

De manera similar, se añade un evento de clic a los elementos con la clase *“.next-month*”. Al hacer clic, se incrementa el mes actual. Si el mes actual es mayor que 11, se establece a 0 (enero) y se incrementa el año actual. Luego, se vuelve a renderizar el calendario con el nuevo mes y año, y se guarda la fecha actualizada en las *cookies* utilizando la función “*saveDate*”. *Inicialización JS de calendar\_raw.php*

La función “*saveDate*” se utiliza para guardar una fecha en las cookies del navegador. Se toma como argumentos el año (*year*) y el mes (*month*), y se construye una cadena de fecha en el formato "*año-mes*". Esta cadena se asigna a una cookie llamada "*savedDate*", lo que permite que la fecha se guarde en el navegador del usuario.

Texto

Descripción generada automáticamenteLa función “*getCookie*” se utiliza para recuperar el valor de una *cookie* específica por su nombre. Se toma como argumento el nombre de la cookie (*name*). Primero, se obtienen todas las cookies del documento y se dividen en un array utilizando *split*(";"). Luego, se itera sobre este array de *cookies*. Para cada *cookie*, se divide la cookie en un par nombre-valor utilizando *split*("="). Si el nombre de la *cookie* coincide con el nombre buscado, se devuelve el valor de la *cookie* después de decodificarlo con “*decodeURIComponent*”. Si no se encuentra ninguna coincidencia, se devuelve *null*.

*Funciones saveDate y getCookie  
 de calendar\_raw.php*

Para la función “*renderCalendar*”, primero se calcula el número de días en el mes especificado utilizando *new Date(year, month + 1, 0).getDate().* Esto se hace creando una fecha con el primer día del mes siguiente y restando un día, lo que da como resultado el último día del mes actual. Luego, se determina el día de la semana del primer día del mes utilizando *new Date(year, month, 1).getDay().*

A continuación, se construye una cadena de texto que muestra el nombre del mes y el año, utilizando los arrays “*monthNames*” y “*year*”. Este texto se inserta en el elemento con la clase *“.month-year*”.

El cuerpo del calendario *(“.calendar-body”)* se vacía para asegurarse de que no haya contenido previo. Luego, se crea una fila *(“.calendar-row”)* y se añaden los nombres de los días de la semana a esta fila. La fila se agrega al cuerpo del calendario.

Para alinear correctamente los días del mes, se crean celdas vacías en la primera fila hasta que se alcanza el primer día del mes. Esto se hace añadiendo divs con la clase “*.calendar-day*” vacíos.

Se itera sobre cada día del mes, creando un “*div*” con la clase “*.calendar-day*” para cada día. La fecha se formatea en una cadena y se almacena en el atributo “*data-date*” del “*div*” para posteriormente poder hacer el *drag and drop* mencionado con anterioridad. Si la fecha está presente en la base de datos, se añade una clase adicional “*highlighted*” al “*div*”.

Se añaden los días del mes a la fila actual. Si la fila tiene siete días (la semana completa), se añade al cuerpo del calendario y se crea una nueva fila.

Finalmente, si la última fila no tiene siete días, se completa con celdas vacías hasta que tenga siete días. La fila se añade al cuerpo del calendario.



*Función render Calendar de calendar\_raw.php*

Seguidamente se tiene el script “*calendar\_drag.js*” el cual se ha mencionado anteriormente, pero al ser un *drag and drop* convencional, se detallará todo el código a la vez, en vez de hacerlo por funciones como se ha hecho anteriormente.

Texto

Descripción generada automáticamenteLa función “*drag*” se llama cuando se comienza a arrastrar un elemento. Utiliza *event.dataTransfer.setData("text", event.target.id)* para almacenar el id del elemento arrastrado, permitiendo que se recupere posteriormente al soltarlo. La función “*drop*” se llama cuando un elemento arrastrado se suelta sobre un elemento destino. Primero, se llama a *event.preventDefault()* para evitar el comportamiento predeterminado del navegador, luego se recupera el id del elemento arrastrado usando *event.dataTransfer.getData("text").* También se obtiene la fecha del elemento destino usando *event.target.dataset.date*. Posteriormente, se realiza una solicitud AJAX POST a ideas\_guardadas.php, enviando el id de la idea y la fecha (no se detalla más debido a que esta parte va en conjunto con la explicación *back-end* de la aplicación). En caso de éxito, se almacena una bandera en “*localStorage”* para mostrar una alerta y se recarga la página. En caso de error, se muestra un mensaje de error en la *Funciones drag&drop del calendar.php*

consola (esto será explicado más adelante debido a que se le da uso junto con unas alertas).

Ahora para saber qué días el usuario ha planificado subir vídeo, ha de tener marcado en el día en el que se sube y acceder a qué vídeo es, para ello, cuando el cursor del ratón entra en un elemento con la clase “.*highlighted*”, se ejecuta el evento “*mouseenter*”. Se obtiene la fecha del día resaltado y se realiza una solicitud AJAX POST a “*get\_idea\_name.php*”, enviando la fecha para obtener los nombres de las ideas asociadas. En caso de éxito, se separan los nombres de las ideas y se construye un contenido de “*tooltip”* (*tooltipContent*). El contenido del “*tooltip”* se añade al elemento resaltado y se ajusta la posición del “*tooltip”* para que se muestre sobre el elemento.

Cuando el cursor del ratón sale de un elemento con la clase “*.highlighted*”, se ejecuta el evento “*mouseleave”*, eliminando el *“tooltip”* *(.hover\_mark)* del elemento resaltado.

Texto

Descripción generada automáticamente *Función de visualización de vídeo en calendario*

Texto

Descripción generada automáticamenteCon el “*localStorage*” mencionado con anterioridad, se le da uso para la siguiente alerta, ya que se exige actualizar la página para poder actualizar el calendario con las clases “*.highlighted”*, así que se crea el “*localStorage*” para que al actualizarse, se muestre la siguiente alerta para verificar que se ha efectuado el cambio en la base de datos, y posteriormente destruirlo.

*Función de alerta al actualizar fechas con drag&drop.*

Para el diseño del calendario, como se ha mencionado anteriormente, se ha hecho con CSS Vanilla de la siguiente manera. Resumidamente, el cuerpo del calendario, utiliza “*display: flex*” y *“flex-wrap: wrap*” para organizar los días del mes en una cuadrícula flexible. Cada fila de días se muestra como una fila flexible. Los días individuales del calendario cuentan con un tamaño fijo, y un borde gris para distinguirlos. La propiedad “*position: relative*” se añade para facilitar la colocación de elementos hijos posicionados absolutamente dentro de los días del calendario.

*Diseño del calendario en CSS Vanilla*

El resultado final junto con la alerta se vería de la siguiente manera:

Interfaz de usuario gráfica

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

*Diseño final de calendar.php*

**FORMULARIOS DE DETALLE**

En cuanto al código HTML de los formularios de detalle, sigue siendo igual de estructura que las pestañas mencionadas anteriormente, sin embargo, esta tiene la peculiaridad de que tiene su propio menú de opciones para poder elegir qué formulario desea rellenar el usuario, mostrándose por defecto el formulario del guion, y gracias a los ids añadidos y al código Javascript con *jQuery* para ir mostrando o escondiendo el que no ha sido presionado.

*Texto

Descripción generada automáticamenteTexto

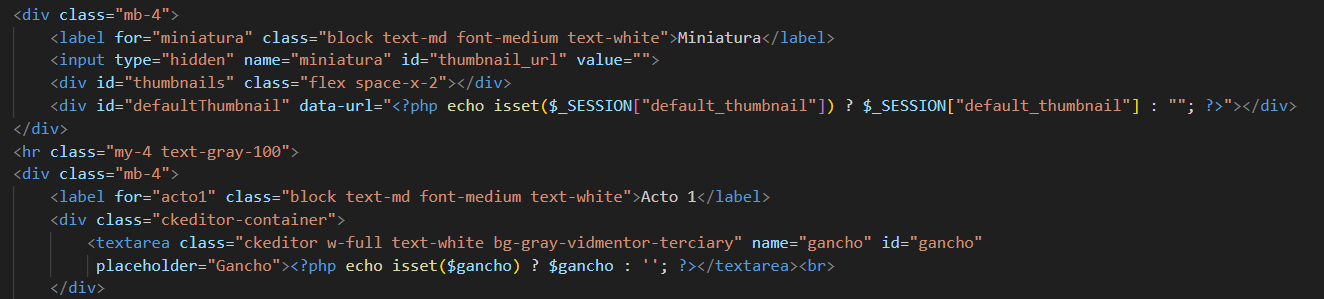
Descripción generada automáticamenteCódigo HTML de forms\_container.php JS para mostrar u ocultar formularios*

En cuanto a los dos formularios que se pueden elegir, se explicarán uno a uno listándolos:

* **FORMULARIO DE GUION**

En base al HTML y PHP que se puede observar, es un formulario normal con sus campos con las distintas clases de CSS para poder tener un diseño responsivo, por lo tanto no se va a profundizar en exceso en las distintas clases utilizadas, sino en el diseño global que tiene dicho formulario y peculiaridades de PHP que lleguen a afectar al diseño visual.

Comenzando por su diseño, el formulario está dividido en las distintas partes que tiene un vídeo, las contribuciones que ha de tener en la subida, tal como el título (el cual está bloqueado, es decir, que no se puede modificar), la descripción, la fecha (para que los usuarios de móvil que no puedan realizar el *drag and drop* puedan modificarla libremente) y la miniatura, y la siguiente división de actos que se ha mostrado antes en el gráfico de cómo se ha de relatar una historia, teniendo cada uno sus campos a rellenar. Cada campo tiene como contenido que, si existe la variable de su campo específico, se muestre el contenido de la variable para poder ver qué ha sido guardado con anterioridad, pasando esto en todos los campos tanto de texto como de fecha excepto en el de miniatura ya que, en su caso, al ser miniaturas generadas por Javascript, se ha de sobrescribir la primera que se imprima para poder mostrar la guardada. Todo esto se hace gracias a las llamadas a la base de datos que se detallarán más adelante en la parte de implementación.



*Formulario HTML del guion*

También, como en Javascript se van a necesitar algunos datos de la base de datos (título y miniatura), estas serán pasadas al archivo por creación de variables y atributos.

Texto

Descripción generada automáticamente

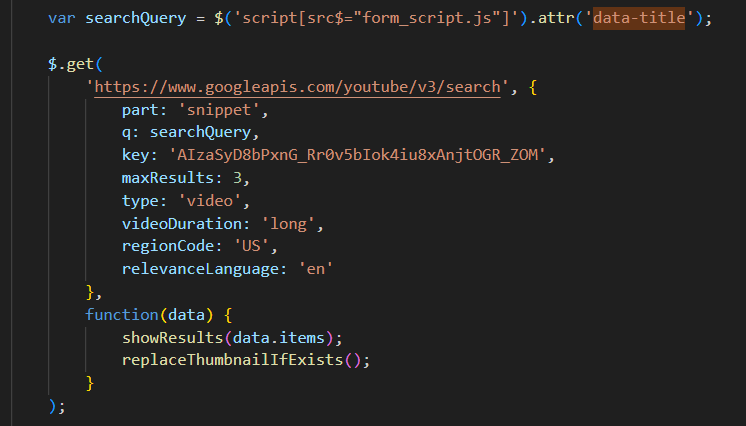
*Creación de variables para JS del guion*

En la parte de Javascript, al hacer clic en un elemento con la clase “*.ckeditor*” (los cuales son todos los *textarea* a excepción del de la descripción), se ejecuta una función que identifica el área de texto correspondiente. Si hay un editor activo (*activeEditor*) y su id es diferente del área de texto actual, el editor activo se destruye utilizando *CKEDITOR.instances[activeEditor].destroy()*. Luego, se actualiza *activeEditor* con el id del área de texto actual y se inicializa *CKEditor* en ese área con una barra de herramientas básica.

Se añade un manejador de eventos de clic al documento para detectar clics fuera del contenedor del editor *(.ckeditor-container)*. Si se hace clic fuera y hay un editor activo, dicho editor se destruye y *activeEditor* se establece en *null*.

Finalmente, se añade un manejador de eventos de entrada (*input*) a todas las áreas de texto. Cada vez que se introduce texto en una de estas áreas, su contenido HTML se actualiza para reflejar el texto actual, utilizando *textarea.html(textarea.text())*.

 *Función para crear o destuir CKEditor en guiones*

Seguidamente, en el mismo “*(document).ready*” se puede observar como se le vuelve a dar uso a la *API* de Youtube para poder hacer la generación de miniaturas en función del título, el cual se está almacenando en “*searchQuery*” debido a que se ha pasado como atributo en el apartado de HTML, después de ello se ejecutan las funciones “*showResults*” y “*replaceThumbnailIfExists*” que se explicarán a continuación.

*Generación de miniaturas en guiones*

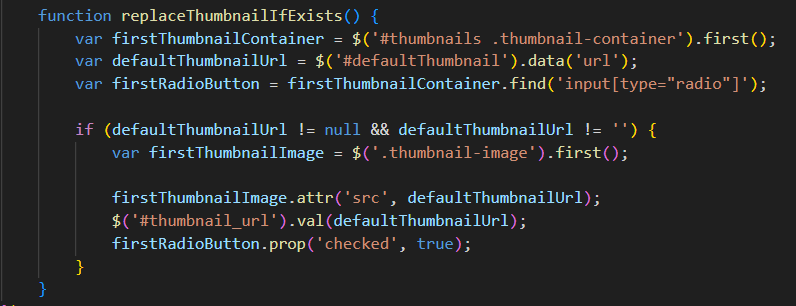
La función “*showResults”* se encarga de mostrar una lista de resultados de vídeos de YouTube en un contenedor de miniaturas. Primero, se selecciona el elemento con id “*thumbnails*” y se vacía su contenido para asegurarse de que no haya resultados anteriores. Para cada elemento en la lista de resultados *items*, se verifica si el tipo de id es “*youtube#video*”. Si es así, se extraen el id del vídeo, el título y la URL de la miniatura del vídeo. Luego, se crea un contenedor para la miniatura utilizando un “*div*” con la clase “*.thumbnail-container*”, que se configura para ser flexible y centrar su contenido.

Dentro del contenedor de la miniatura, se añade una imagen de la miniatura con la URL extraída y se le asigna la clase “*.thumbnail-image*”. También se crea un botón de radio con el tipo radio, el nombre video, y el valor del id del vídeo. Después, se agrega un evento de clic a la imagen de la miniatura, de modo que al hacer clic en la imagen, el botón de radio correspondiente se selecciona, y el valor de la URL de la miniatura se asigna al elemento con id “*#thumbnail\_url*”.

Al final, el contenedor de la miniatura, que incluye la imagen y el botón de radio, se añade al contenedor de miniaturas. Este proceso se repite para cada elemento en la lista de resultados, asegurando que todas las miniaturas de los vídeos se muestren correctamente, que, en este caso, como especificamos en la búsqueda, son 3.

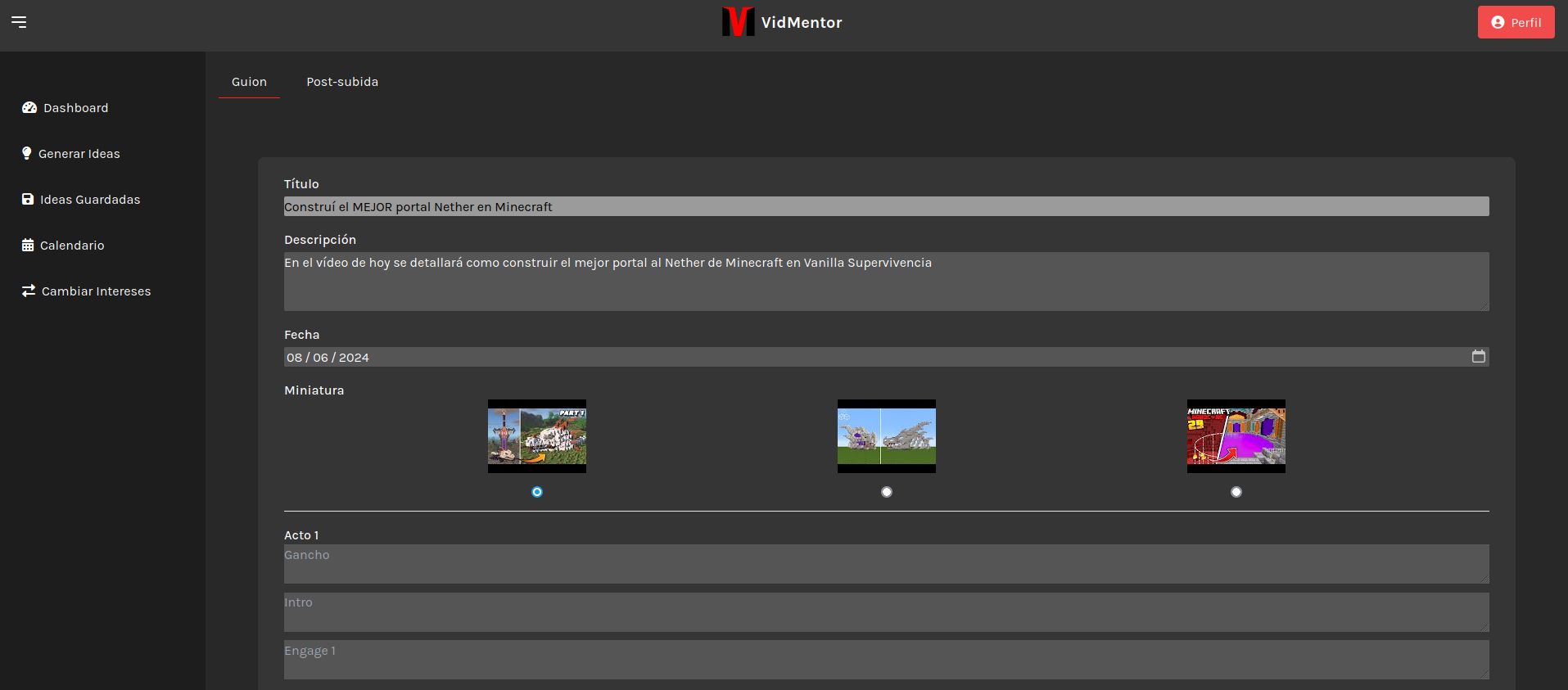
 *Función showResults de guiones*

Para función “*replaceThumbnailIfExists*” se verifica si hay una miniatura por defecto y la reemplaza en la primera posición de la lista de miniaturas, si es necesario. Primero, se selecciona el primer contenedor de miniaturas dentro del elemento con id “*#thumbnails*” y se almacena en la variable “*firstThumbnailContainer*”. Luego, se obtiene la URL de la miniatura por defecto desde el atributo *data-url* del elemento con id “*#defaultThumbnail”* y se almacena en la variable “*defaultThumbnailUrl*”. También se selecciona el primer botón de radio dentro del primer contenedor de miniaturas y se almacena en la variable “*firstRadioButton*”.



*Función replaceThumbnailIfExists de guiones*

Cabe destacar que como método de mejor diseño, al iniciar la página se destruyen todos las instancias de los editores de texto *CKEditor*.



*Diseño final del formulario de guion*

* **FORMULARIO POST-SUBIDA**

Para el formulario post-subida no se detallará en exceso el HTML ni se mostrará captura del mismo ya que, es igual que el formulario visto anteriormente, pero con *checkboxes*, por tanto, se deduce que puede se información redundante para aportar, la única diferencia destacable es el listado utilizado en forma de escalera para cierta parte de las opciones, ya que, son un “*li*” dentro de otro, aplicándose un *margin-left* a cada uno de ellos.

Detallando el apartado de Javascript, cuando el documento está listo, se define y ejecuta la función “*toggleSubElements*” para habilitar o deshabilitar ciertos elementos según el estado de un *checkbox* con id “*#checkCTR*”. Esta función comprueba si el *checkbox* está marcado, y luego habilita o deshabilita los elementos con los ids que se pueden observar en la captura del código en función de ese estado. Si el *checkbox* no está marcado, se eliminan las decoraciones de texto de los elementos con la clase “*.form-check-label*”.

La función “*toggleSubElements*” se ejecuta inicialmente para establecer el estado correcto de los elementos y, un evento “*change*” se asocia al *checkbox* *“#checkCTR*” para que, cuando su estado cambie, se ejecute toggleSubElements y disableNextCheckboxes, al igual que con “.form-checkbox” (solo este último).

La función “*disableNextCheckboxes*” verifica el estado del *checkbox* *“#checkCTR*” y deshabilita o habilita los *checkboxes* posteriores (los dispuestos en escalera) en función de su estado y del estado de los *checkboxes* anteriores. Si un *checkbox* anterior no está marcado, se deshabilitan los siguientes, se desmarcan y se elimina cualquier decoración de texto de sus etiquetas.. Al final, “*disableNextCheckboxes”* se ejecuta para asegurar que todos los *checkboxes* estén en el estado correcto al cargar la página.

Texto

Descripción generada automáticamente*Código JS del formulario de checkboxes*

Captura de pantalla de computadora

Descripción generada automáticamente

*Diseño final del formulario de checkboxes*

**CAMBIAR INTERESES**

En el código HTML del formulario de intereses, se puede observar que se está trabajando sobre un formulario distinto a los que hemos visto con anterioridad, debido a que no existe una etiqueta formulario, y todo el formulario se desarrollar a raíz de Javascript, para el se hace uso de evento *onclick* para ir pasando de pregunta, también se hace uso de un input oculto con el id de usuario para poder pasarlo al Javascript, esto es otro método que no se ha visto. Se hace uso de distintos colores como rojo y azul para dar mayor diversidad a la pestaña y que no sea tan monótona

*Texto

Descripción generada automáticamenteCódigo HTML de intereses.php*

En el Javascript, se pueden observar distintas funciones, pero se va a empezar detallando la que ofrece las distintas preguntas para el usuario (“*nextQuestion*”), ya que se usa para actualizar dinámicamente el contenido de una sección de preguntas en función de la elección del usuario. Primero, se selecciona el elemento con id “*#questions”* y se limpia su contenido. Dependiendo de la opción elegida, se establece un interés seleccionado y se inserta una nueva pregunta con opciones relevantes en el elemento *questionsDiv*.

Si se elige *“Vlogs”*, se establece el interés en “*Vlogs*” y se añade una nueva pregunta sobre si los vlogs serán diarios o informativos. Cada opción tiene un botón que, al hacer clic, llama a la función *finalQuestion* o *showInformativeInput*.

Para “*Videojuegos”*, se establece el interés en “*Videojuegos”* y se añade una pregunta sobre el estilo del contenido. Las opciones permiten elegir entre genéricos o específicos, y cada opción llama a la función *finalQuestion* o *showGameInput*.

Si se elige “*Desarrollo*”, el interés se establece en “*Desarrollo*” y se añade una pregunta sobre el tipo de desarrollo. Las opciones permiten elegir entre programación o desarrollo personal, llamando a *finalQuestion*.

Para “*Reacciones”*, se establece el interés en “*Reacciones”* y se añade una pregunta sobre el tipo de reacciones. Las opciones permiten elegir entre música o contenido viral, cada una llamando a *finalQuestion*.

 *Función nextQuestion en intereses*

Tanto la función *showInformativeInput* como *showGameInput* llevaban a un formulario escrito donde se puede detallar con exactitud qué tipo de videojuego o qué tipo de vlogs sube el usuario a su canal, completando así el formulario, y siendo procesado (detalles en implementación).

*Texto

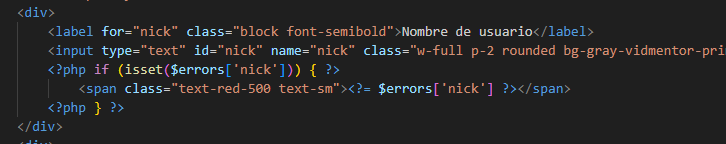
Descripción generada automáticamenteFunciones showInformativeInput y showGameInput de intereses*



*Diseño final del formulario intereses*

**EDICIÓN DE PERFIL**

Al igual que relatar en el formulario de *checkboxes*, en este tampoco se mostrará todo el código HTML ya que es demasiado amplio y no ofrece información extra que no se haya visto, a excepción de lo que se mencionará a continuación. Como novedad, en este formulario, se hace uso de PHP para poder mostrar errores que haya podido haber en la parte *back-end* al procesar la solicitud de envío de formulario. Para ello, únicamente se identifica si existe un error en relación a ese campo con el array de errores obtenido y, en caso de que exista, se muestra.



*Detección de errores en el formulario de perfil*

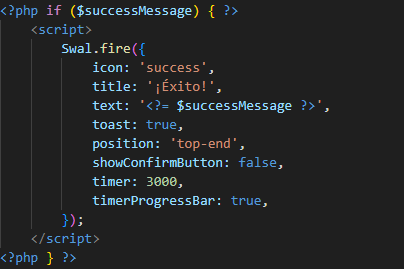
Pantalla de computadora

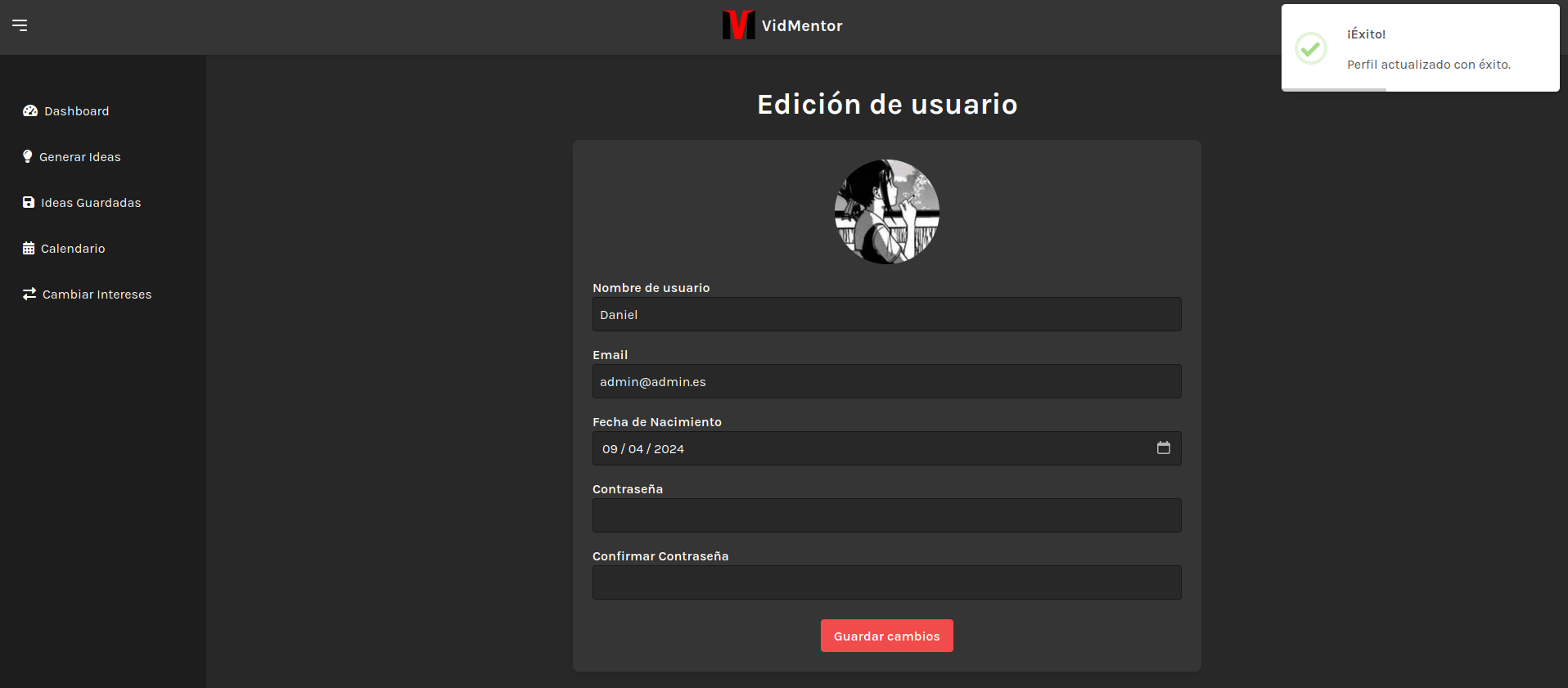
Descripción generada automáticamenteTexto

Descripción generada automáticamenteA su vez también, otra peculiaridad de este formulario es que la imagen del usuario, al ser cargada en el formulario, pueda ser previsualizada en el perfil gracias a la función *previewImage*. En este, se crea un nuevo objeto *FileReader*, que permite leer el contenido de archivos. Se define un evento *onload* para este objeto, que se activa cuando la lectura del archivo se completa con éxito. Dentro de este evento, se obtiene la URL de datos codificados en base64 de la imagen y se asigna al atributo “*src*” de una etiqueta de imagen, que se supone que tiene el id *“#imagePreview*”, para mostrar la vista previa de la imagen seleccionada. Finalmente, se inicia la lectura del contenido del archivo seleccionado utilizando el método *readAsDataURL* del *FileReader*.

*Función previewImage del formulario de usuario*

También para que el usuario reciba *feedback* de cuando su perfil ha sido actualizado con éxito, se vuelve a hacer uso de *SweetAlert* cuando el formulario ha sido exitoso.

*Ejecución de SweetAlert al actualizar información de usuario*



*Diseño final del formulario de edición de perfil*

### Implementación

En este apartado se detallará la parte *back-end* de cada una de las distintas partes de la web. Para ello se seguirá la misma división que en el apartado de diseño, contando con los archivos extra acordes para su correcto desarrollo:

• *Landing page* (Inicio)

• Inicio de sesión

• Registro de usuarios

• *Dashboard*, el cual incluye:

* Generación de ideas
* Distribución de ideas guardadas
* Formularios de detalle
* Formulario de intereses
* Modificar perfil

**COMMON FUNCTIONS**

ads

**DASHBOARD**

asd

### Pruebas

Son muchas pruebas que pueden realizarse en un proyecto, para eliminar los posibles errores y garantizar su correcto funcionamiento. Los casos de prueba establecen las condiciones/variables que permitirán determinar si los requisitos establecidos se cumplen o no. A continuación se detallan algunos de los casos de prueba que se ejecutarán para comprobar la correcta construcción de este proyecto.

A modo de ejemplo se facilitan los campos de una posible plantilla para definir casos de pruebas, que deberá modificarse como se estime oportuno en función del proyecto y las pruebas que se consideren oportunas realizar.

* Fecha/autor/[ versión a probar]
* Caso de prueba

Identificador del caso de prueba ( nombre único)

* Descripción

Breve explicación sobre el objetivo del caso de prueba

* Condiciones de ejecución

Descripción de las condiciones de ejecución que se deben cumplir antes de iniciar el caso de prueba, por ejemplo, que se haya realizado correctamente el login en el sistema...

* Entrada

Datos necesarios para poder ejecutar la prueba

* Resultado esperado

Valor esperado para el correcto funcionamiento del proyecto

* Resultado obtenido

Valor de salida al ejecutar el caso de prueba

* Evaluación

Comparación del valor esperado y obtenido para concluir, finalmente, si el aspecto chequeado por el caso de prueba confirma el correcto funcionamiento del proyecto o elevación detallada del correspondiente error.

A modo de ejemplo se facilitan los campos de una posible plantilla para reportar los errores detectados en los casos de prueba ejecutados.

* Fecha/autor/[ versión probada ]
* Caso de prueba
* Evaluación

Comparación del valor esperado y obtenido para concluir, finalmente, si el aspecto chequeado por el caso de prueba confirma el correcto funcionamiento del proyecto o elevación detallada del correspondiente error.

* Posible causa de error

Detallar la posible causa o causas que han podido generar el error detectado.

* Posible corrección

Detallar la posible forma de corregir el problema

* Áreas afectadas

Detallar qué áreas (módulos, componentes, documentos,...) se verán afectados al implementar la corrección.

## Objetivos a conseguir

Se pueden diferenciar objetivos que son para el desarrollo (cumplimiento de los requisitos técnicos) del proyecto o bien del negocio, de la empresa, financieros,...

* Extender el uso en la empresa de nuevas metodologías, estándares, procedimientos de trabajo...
* Abrir el mercado internacional
* Satisfacer los requisitos del cliente con un valor añadido
* Facturar 100.000.00 en 2015
* Tener 150 clientes el primer año.
* Contar con 3000 visita en la página web el primer mes

## Previsión de los recursos materiales y humanos necesarios

Se tendrá en cuenta las herramientas y la formación necesaria para desarrollar las actividades que requiere el proyecto, así como el tiempo para llevarlo a cabo.

## Presupuesto económico.

Detallar el coste económico de los recursos anteriormente establecidos.

# PLANIFICACIÓN DE LA EJECUCIÓN DEL PROYECTO

A continuación se detallan las actividades/tareas/procedimientos por cada una de las fases del proyecto previamente establecidas.

[Estas tareas podrán variar en función del proyecto]

## Fase de Análisis

1. Estudio de las necesidades a cubrir
2. Estudio de la situación actual
3. Establecimiento de los requisitos del proyecto
4. Valoración comparativa de las posibles soluciones
5. Identificación de las necesidades que implica el nuevo proyecto en la empresa.

Para solventar los problemas que plantea el proyecto puede ser necesario contratar personal, formarlo en determinadas metodologías/herramientas, comprar equipos...

1. Estudio de viabilidad de la solución elegida teniendo en cuenta no solo los beneficios económicos.
2. Corrección de posibles errores

## Fase de diseño

1. Preparación del entorno de diseño
2. Diseño de la arquitectura
3. Diseño de los interfaces
4. Diseño de los datos
5. Diseño de los procedimientos
6. Corrección de posibles errores

## Fase de Implementación

1. Preparación del entorno de implementación
2. Desarrollo de la arquitectura
3. Desarrollo de los interfaces, los datos y los procedimientos
4. Corrección de posibles errores

## Fase de pruebas

1. Preparación del entorno de pruebas
2. Ejecución de las pruebas y reporte de los errores encontrados

# DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN

* Definir el procedimiento de evaluación de las **incidencias** que puedan presentarse durante la realización de las diferentes actividades.
* Definir el procedimiento para gestionar los posibles **cambios** en los recursos y en las actividades.

# FUENTES

Incluir las páginas web, biografía,,,consultadas.

# ANEXOS

Cualquier añadido se incluirá en este apartado.

Por ejemplo, a continuación se incluyen unas pautas a tener en cuenta a la hora de elaborar la documentación del proyecto.

## Guía de estilo

Obviamente, este anexo deberá ser eliminado del documento final a entregar.

* **Título del proyecto**

Elegir un nombre llamativo y relacionado con la temática que va a tratar.

* **Figuras y tablas**

Cualquier figura, tabla... incluida en el documento deberá tener un título a pie de página..

Incluir tablas, gráfico, mapas conceptuales...que ayuden a leer y comprender el documento.

* **Índices**

Además del índice de contenidos, ya incluido en la plantilla, se añadirá a continuación el índice de figuras, si fuera necesario.

* **Redacción**

Se evitarán las mayúsculas, salvo en los títulos y poco más.

No se emplearán formas personales (instalamos, seleccionamos...) en su lugar se utilizarán formas impersonales ( instalar, se instalará, seleccionar, se selecciona,...).

Se evitará la voz pasiva (casi siempre traducción literal del inglés). En vez de: es desarrollado para cumplir... mejor: se desarrolla para cumplir...

Se evitarán los párrafos largos.

Se utilizarán las viñetas para facilitar la lectura del documento.

* **Formato**

El documento se generará en formato pdf.

* **Entrega**

Todo el material del módulo Proyecto (documentos, ficheros fuentes, herramientas...) se entregará en formato electrónico, en una carpeta comprimida:

CICLO-CURSO-TITULO DEL PROYECTO