



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto**

**Markdown2Video: Sistema para la Conversión  
Automatizada de Presentaciones Marp a Video y  
Documentos Markdown**

Curso: Calidad y Pruebas de Software

Docente: *Ing. Patrick Jose Cuadros Quiroga*

Integrantes:

<b><i>Calizaya Ladera, Andy Michael</i></b>	<b><i>(2022074258)</i></b>
<b><i>Camac Melendez, Cesar Nikolas</i></b>	<b><i>(2022074262)</i></b>
<b><i>Fernandez Villanueva, Daleska Nicolle</i></b>	<b><i>(2021070308)</i></b>

**Tacna – Perú  
2025**



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
3.0	DNFV	CNCM	CNCM	21/06/2025	Versión Original

# **Markdown2Video: Sistema para la Conversión Automatizada de Presentaciones Marp a Video y Documentos Markdown**

## **Documento de Especificación de Requerimientos de Software**

**Versión 3.0**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
3.0	DNFFV	CNCM	CNCM	21/06/2025	Versión Original

## ÍNDICE GENERAL

<b>1. INTRODUCCIÓN</b>	<b>4</b>
1.1. Propósito (Diagrama 4+1)	4
1.2. Alcance	4
1.3. Definición, siglas y abreviaturas	5
<b>2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS</b>	<b>6</b>
2.1.1. Requerimientos Funcionales	6
2.1.2. Requerimientos No Funcionales – Atributos de Calidad	7
2.1. Restricciones	9
<b>3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA</b>	<b>10</b>
3.1. Vista de Caso de uso	10
1.1.1. Diagramas de Casos de uso	10
3.2. Vista Lógica	10
3.2.1. Diagrama de Subsistemas	10
3.2.2. Diagrama de Secuencia (vista de diseño)	11
3.2.3. Diagrama de Colaboración (vista de diseño)	14
3.2.4. Diagrama de Objetos	14
3.2.5. Diagrama de Clases	15
3.2.6. Diagrama de Base de datos (relacional o no relacional)	16
3.3. Vista de Implementación (vista de desarrollo)	16
3.3.1. Diagrama de arquitectura software (paquetes)	16
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)	17
3.4. Vista de procesos	17
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)	17
3.5. Vista de Despliegue (vista física)	18
3.5.1. Diagrama de despliegue	18
<b>4. ATRIBUTOS DE CALIDAD DEL SOFTWARE</b>	<b>19</b>

## 1. INTRODUCCIÓN

### 1.1. Propósito (Diagrama 4+1)

El propósito de este documento es definir y comunicar la arquitectura de software para el sistema "Markdown2Video" utilizando el modelo 4+1 de Kruchten. Este modelo se ha elegido porque permite abordar la complejidad del sistema desde múltiples perspectivas, asegurando que todos los interesados (desarrolladores, docente, administradores de sistemas) tengan una visión clara y relevante. La arquitectura global se presenta como un sistema web por capas, diseñado para separar la interfaz de usuario, la lógica de negocio y el acceso a datos. Este enfoque tiene como objetivo principal facilitar la mantenibilidad, promover la reutilización de componentes y asegurar que el sistema sea robusto y escalable a futuro.

Las decisiones de diseño han sido fuertemente influenciadas por los requisitos tanto funcionales como no funcionales. Por ejemplo, el requisito funcional de soportar múltiples formatos de exportación (PDF, PPT, video) ha llevado a la creación de un subsistema de conversión modular y extensible. A su vez, el requisito no funcional de **confiabilidad** ha sido clave, especialmente al interactuar con herramientas externas como FFmpeg; por ello, la arquitectura aísla estos procesos para gestionar fallos de manera controlada. Se ha priorizado la **simplicidad de desarrollo y la mantenibilidad** sobre una escalabilidad masiva, optando por una arquitectura monolítica en PHP. Asimismo, se ha favorecido la **potencia de conversión** al depender de FFmpeg, aceptando la restricción de portabilidad que esto implica, ya que la herramienta debe estar preinstalada en el entorno de despliegue.

### 1.2. Alcance

El proyecto abarca el desarrollo de una aplicación que incluya las siguientes funcionalidades:

- Editor de Código Marp: Permite a los usuarios escribir presentaciones en lenguaje Markdown con soporte para Marp.

- Vista Previa en Tiempo Real: Mostrar en una interfaz visual cómo se verá la presentación en diapositivas mientras se edita el código.
- Conversión a Video: Implementar un sistema que transforme automáticamente las diapositivas en un archivo de video con transiciones predefinidas.
- Opciones de Personalización: Incluir configuraciones para ajustar la resolución, duración de las diapositivas y otros parámetros del video.
- Compatibilidad Multiplataforma: Desarrollar la herramienta para que sea accesible en diferentes sistemas operativos.

### 1.3. Definición, siglas y abreviaturas

Sigla/Abreviatura	Definición
Markdown	Lenguaje de marcado ligero para la escritura de texto estructurado.
MP4	Formato de video ampliamente compatible y de alta compresión.
Marp	Es un conjunto de herramientas de código abierto que permite crear presentaciones de diapositivas utilizando sintaxis de Markdown.
TTS	Texto a voz (Text-To-Speech), tecnología que convierte texto en audio hablado.

## 2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

### 2.1.1. Requerimientos Funcionales

RF-ID	Nombre	Descripción del Requerimiento Funcional	Prioridad
RF-01	Gestión de Cuentas de Usuario	El sistema debe permitir el registro de nuevos usuarios, la autenticación (inicio de sesión) y la modificación de los datos del perfil (nombre, correo, contraseña).	Alta
RF-02	Editor de Contenido	El sistema debe proveer un editor de texto que soporte la sintaxis de Markdown y Marp, con una vista previa que se actualice en tiempo real.	Alta
RF-03	Gestión de Proyectos/Presentaciones	El sistema debe permitir al usuario guardar sus presentaciones con un nombre, verlas en un historial personal, editarlas y eliminarlas.	Alta
RF-04	Gestión de Imágenes	El sistema debe permitir al usuario subir imágenes (validando formato y tamaño), almacenarlas y utilizarlas dentro de sus documentos Markdown/Marp.	Alta
RF-05	Importación de Archivos Markdown	El sistema debe permitir al usuario importar un archivo .md desde su dispositivo para cargarlo en el editor.	Alta
RF-06	Exportación a PDF (Desde Markdown)	El sistema debe convertir el contenido del editor Markdown estándar a un archivo PDF estilizado y permitir su descarga.	Alta
RF-07	Exportación a PDF (Desde Marp)	El sistema debe convertir el contenido del editor Marp a un archivo PDF que conserve el formato de diapositivas y permitir su descarga.	Alta
RF-08	Exportación a Video (MP4)	El sistema debe convertir una presentación Marp en un archivo de	Alta

		video .mp4.	
<b>RF-09</b>	<b>Descarga de Código Fuente</b>	El sistema debe permitir al usuario descargar el código fuente de su presentación actual en un archivo .md.	Alta
<b>RF-10</b>	<b>Compartir Presentaciones</b>	El sistema debe ofrecer una funcionalidad para que un usuario envíe por correo electrónico el archivo .md de una de sus presentaciones guardadas a otro destinatario.	Alta
<b>RF-11</b>	<b>Gestión de Plantillas</b>	El sistema debe mostrar al usuario una galería de plantillas predefinidas que pueden ser utilizadas como punto de partida para crear un nuevo documento.	Alta
<b>RF-12</b>	<b>Exportación a HTML (Desde Markdown)</b>	El sistema debe convertir el contenido del editor Markdown estándar a un archivo HTML auto-contenido y permitir su descarga.	Alta
<b>RF-13</b>	<b>Exportación a HTML (Desde Marp)</b>	El sistema debe convertir una presentación Marp en un archivo HTML interactivo que conserve las diapositivas y permitir su descarga.	Alta

### 2.1.2. Requerimientos No Funcionales – Atributos de Calidad

<b>RNF-ID</b>	<b>Descripción del Requerimiento no funcional</b>	<b>Prioridad</b>
RNF-01	El sistema debe ser capaz de procesar y convertir un documento Markdown/Marp de tamaño promedio (ej. 10 diapositivas con texto e imágenes) a formato PDF o HTML en menos de 10 segundos para no afectar la experiencia de usuario..	Alta
RNF-02	El sistema debe proteger las cuentas de usuario mediante el almacenamiento seguro de contraseñas (usando hashing y salting) y asegurar que solo los usuarios autenticados puedan	Medio

	acceder a sus perfiles y contenido	
RNF-03	La interfaz de usuario, tanto en el dashboard como en el editor, debe ser intuitiva y fácil de usar, permitiendo que un usuario sin conocimientos técnicos previos pueda crear y exportar un documento en su primer uso sin necesidad de un tutorial.	Alta
RNF-04	La aplicación web debe ser completamente funcional y visualmente consistente en las dos últimas versiones de los principales navegadores: Google Chrome, Mozilla Firefox y Microsoft Edge.	Alta
RNF-05	El contenido visual del archivo exportado (PDF, HTML) debe ser una representación fiel de la vista previa en el editor, manteniendo la estructura, estilos, imágenes y diagramas con una precisión mínima del 95%.	Alta
RNF-06	El sistema desplegado en la web debe tener un tiempo de actividad (uptime) mínimo del 99.5%, asegurando que la plataforma esté accesible para los usuarios en todo momento.	Media
RNF-07	La arquitectura del sistema debe estar diseñada para soportar un aumento en el número de usuarios concurrentes y en el volumen de conversiones sin una degradación significativa del rendimiento.	Media
RNF-08	El código fuente debe seguir principios de diseño de software (como MVC y SRP), estar debidamente comentado y estructurado en módulos para facilitar futuras actualizaciones, correcciones de errores y la adición de nuevas funcionalidades.	Media
RNF-09	El sistema debe manejar los errores de forma elegante, mostrando al usuario mensajes claros y útiles (ej. "Formato de imagen no válido") en lugar de errores técnicos o páginas en blanco, guiándolo hacia la solución.	Alta



## 2.1. Restricciones

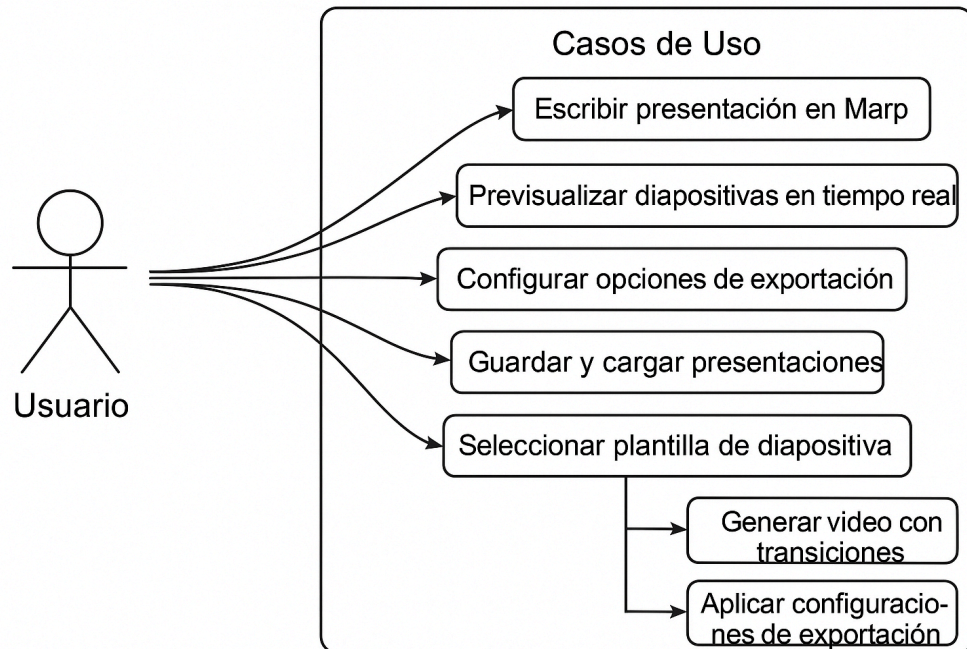
El desarrollo y la funcionalidad del sistema "Markdown2Video" están sujetos a importantes restricciones técnicas y de dependencia. El proyecto debe ser implementado utilizando la pila tecnológica definida, que consiste en PHP 8 para el backend, MySQL 8 para la base de datos, y HTML, CSS y JavaScript para el frontend. Más críticamente, la funcionalidad principal de conversión de video depende por completo de herramientas externas. Esto significa que la calidad, el rendimiento de la codificación y los formatos de video soportados están directamente limitados por las capacidades y la configuración. De manera similar, la calidad y naturalidad de la narración en audio están restringidas por las limitaciones del servicio de Texto a Voz que se integre, incluyendo los idiomas disponibles y los límites de uso de su API.

Adicionalmente, el proyecto opera bajo las limitaciones propias de un contexto académico, con un cronograma fijo que va del 31 de marzo al 18 de junio de 2025 y un equipo de desarrollo predefinido. Estas restricciones de tiempo y recursos implican que el alcance del software se centrará en las funcionalidades de conversión esenciales, excluyendo características avanzadas de edición de video como líneas de tiempo complejas o efectos visuales personalizados. Operativamente, al ser una aplicación web, su uso requiere una conexión a internet estable, tanto para que el usuario acceda a la plataforma como para que el servidor se comunice con APIs externas. Finalmente, el proyecto debe cumplir con las licencias de todo el software de código abierto utilizado, lo cual podría condicionar su distribución o uso futuro.

### 3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

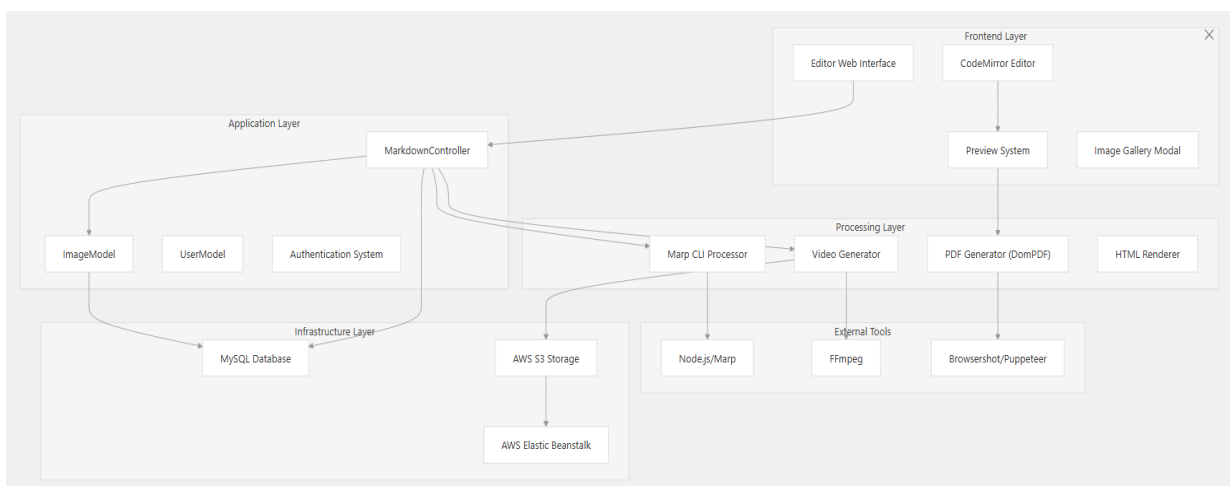
#### 3.1. Vista de Caso de uso

##### 1.1.1. Diagramas de Casos de uso



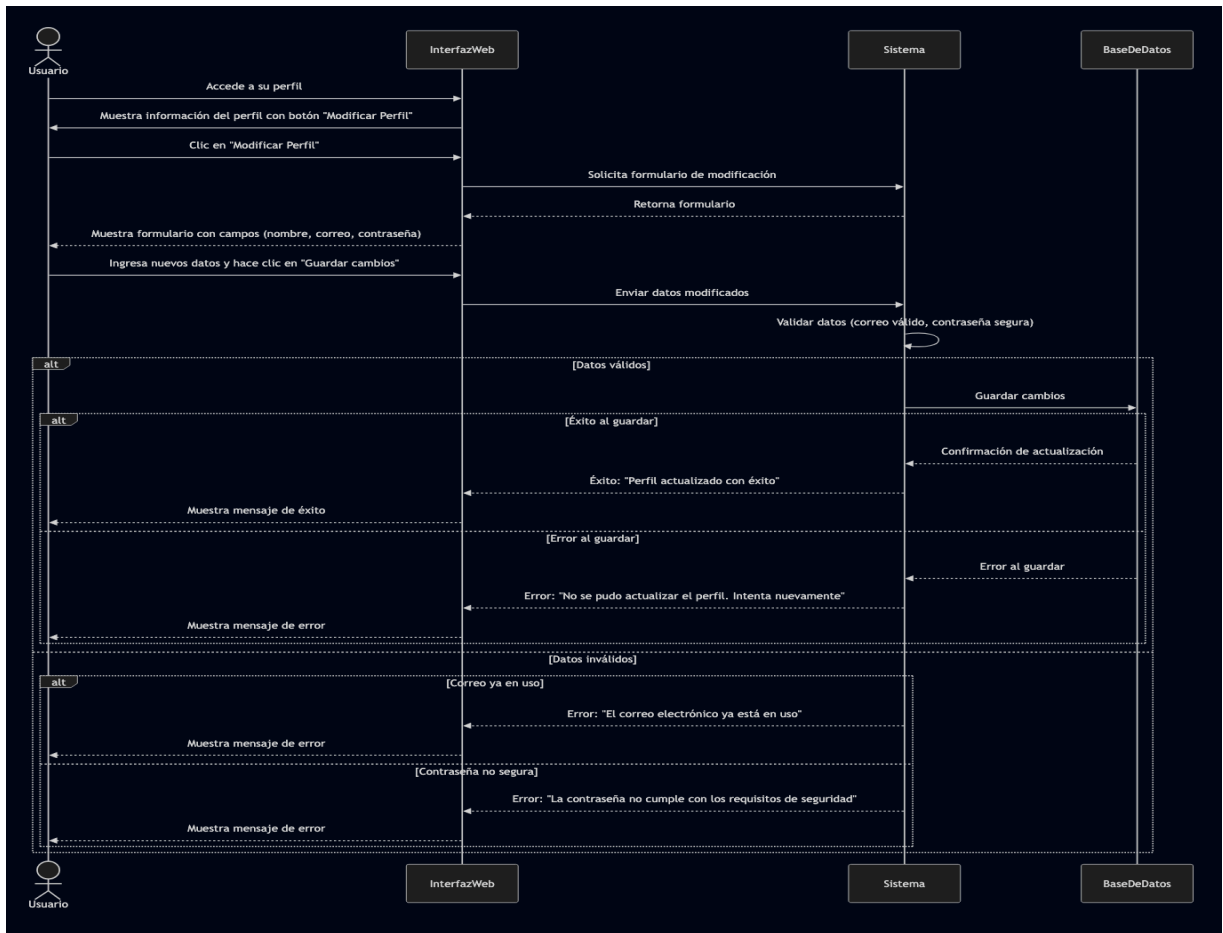
#### 3.2. Vista Lógica

##### 3.2.1. Diagrama de Subsistemas

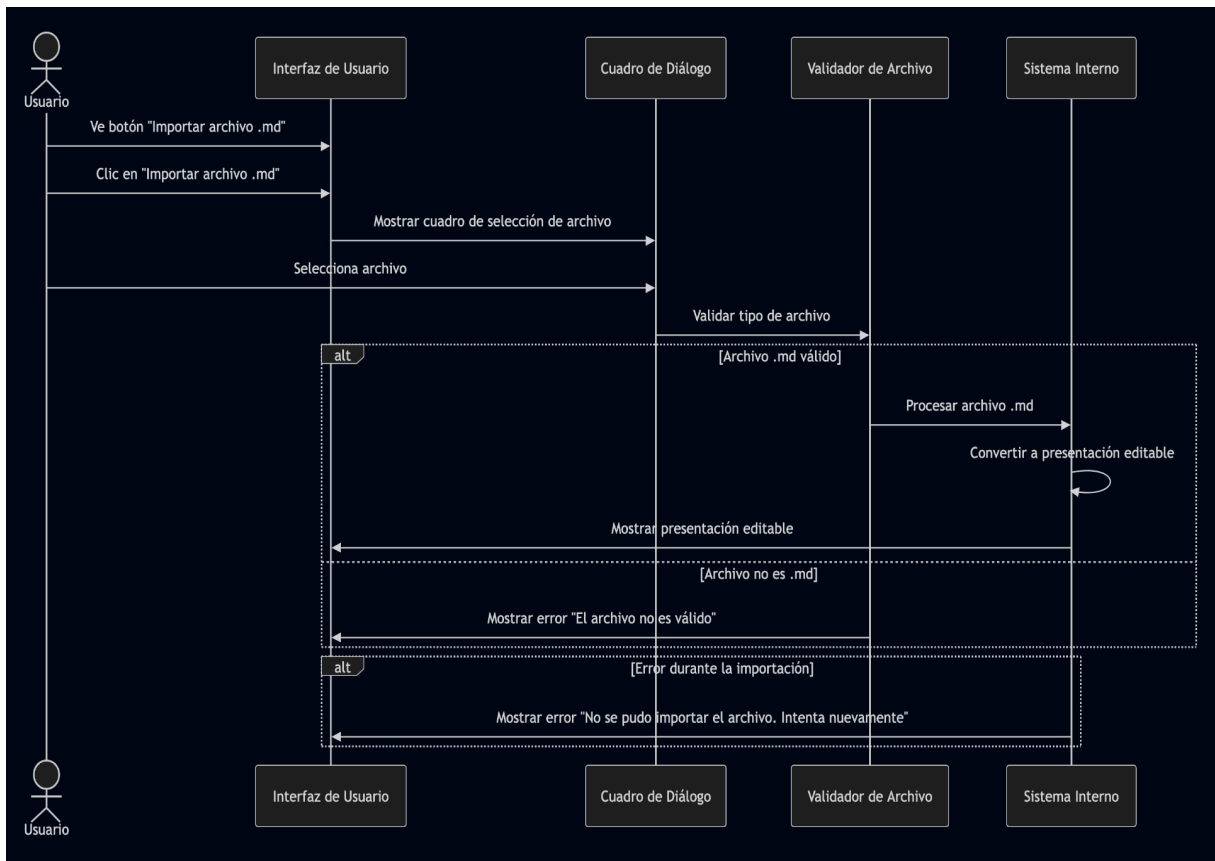


### 3.2.2. Diagrama de Secuencia (vista de diseño)

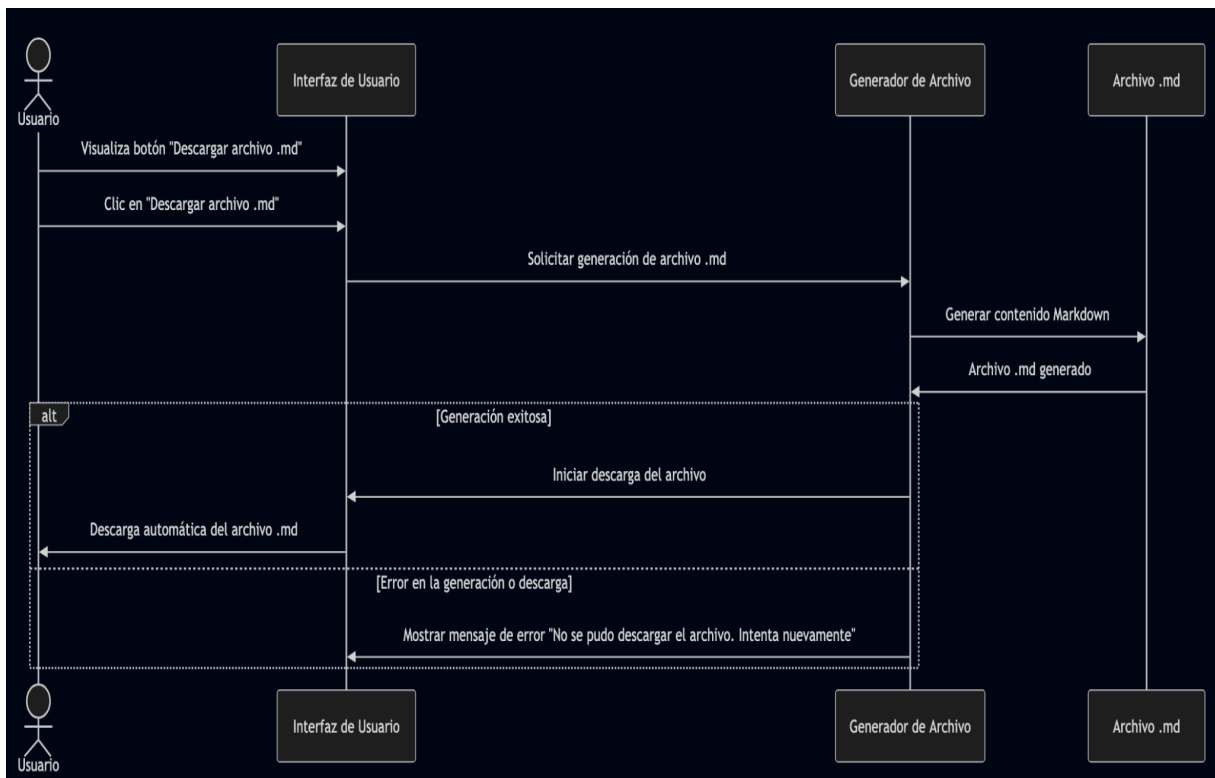
#### Modificar Perfil



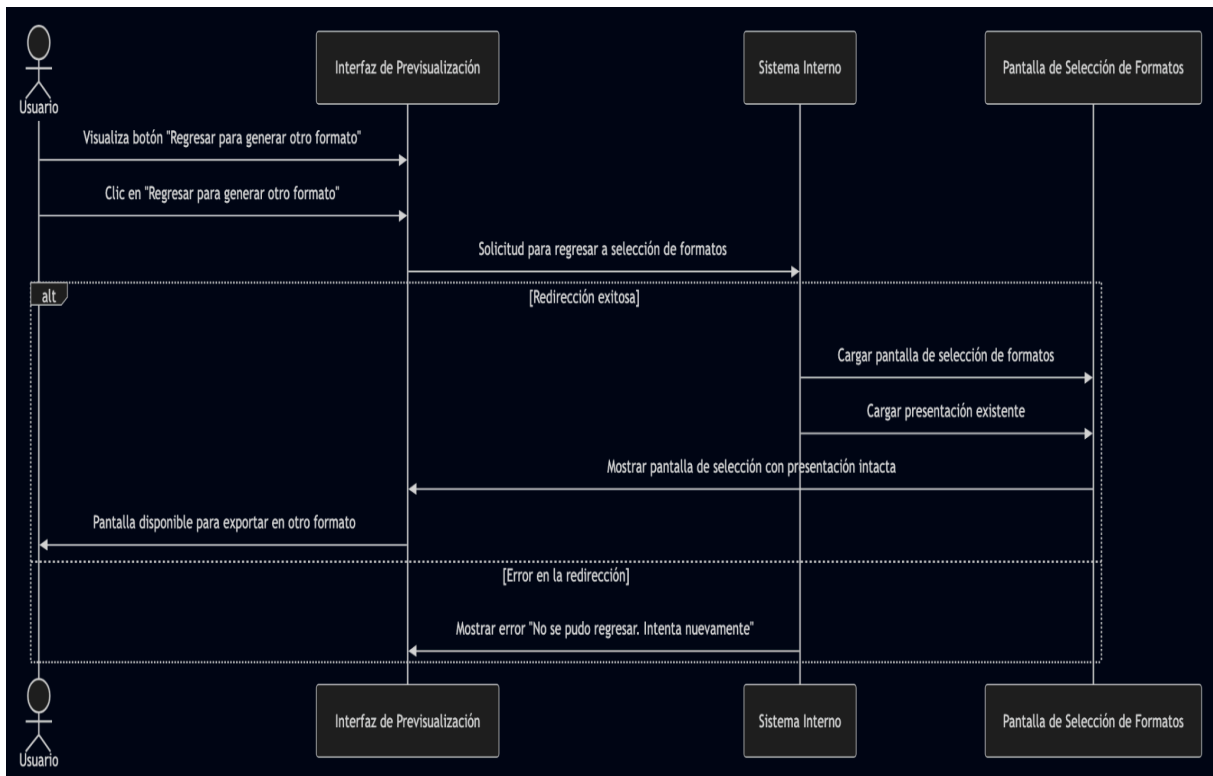
Importar Archivo .md:



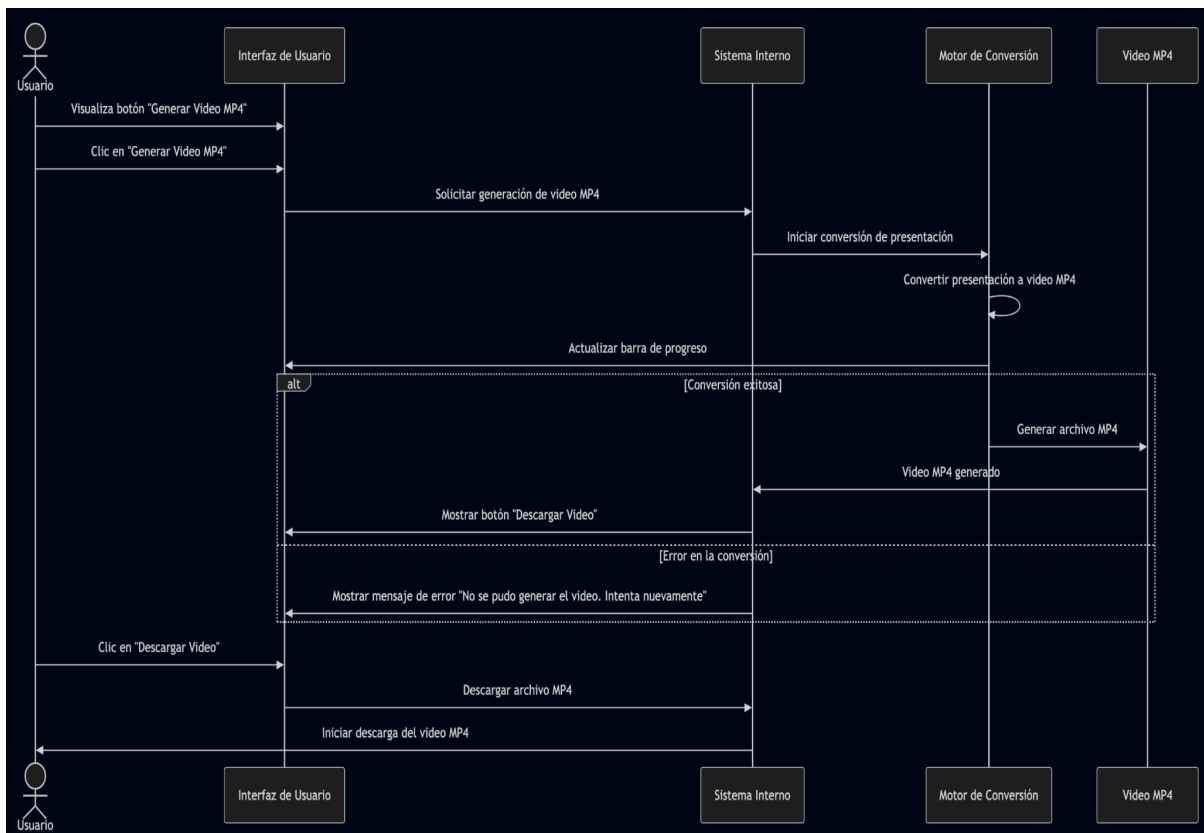
### Descargar Archivo .md:



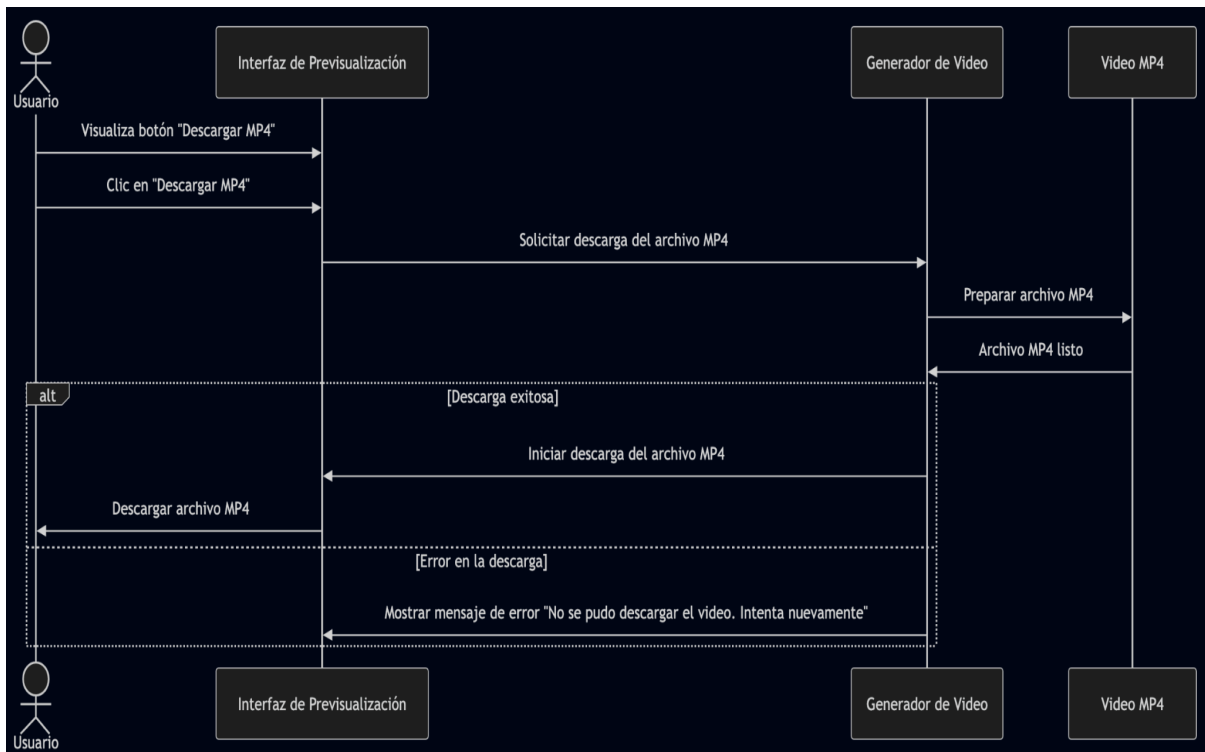
### Regresar desde la previsualización de vídeo MP4:



## Generar video MP4:

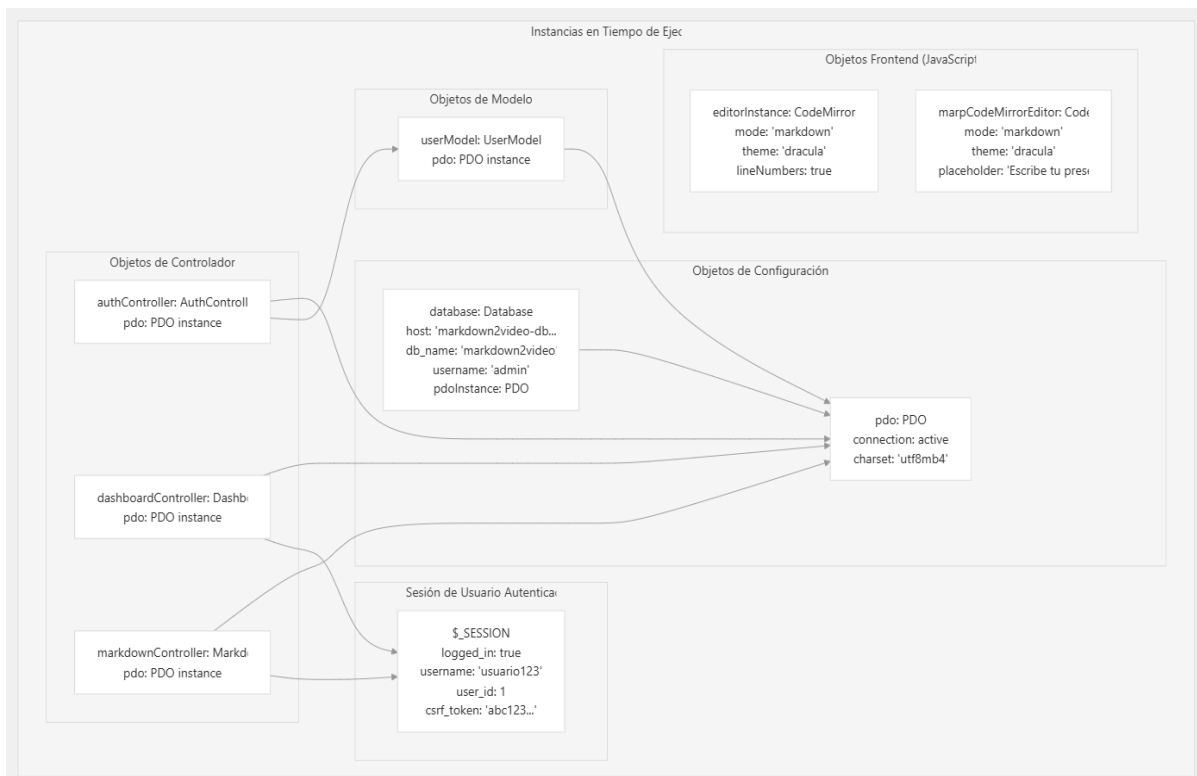


## Descargar video MP4:

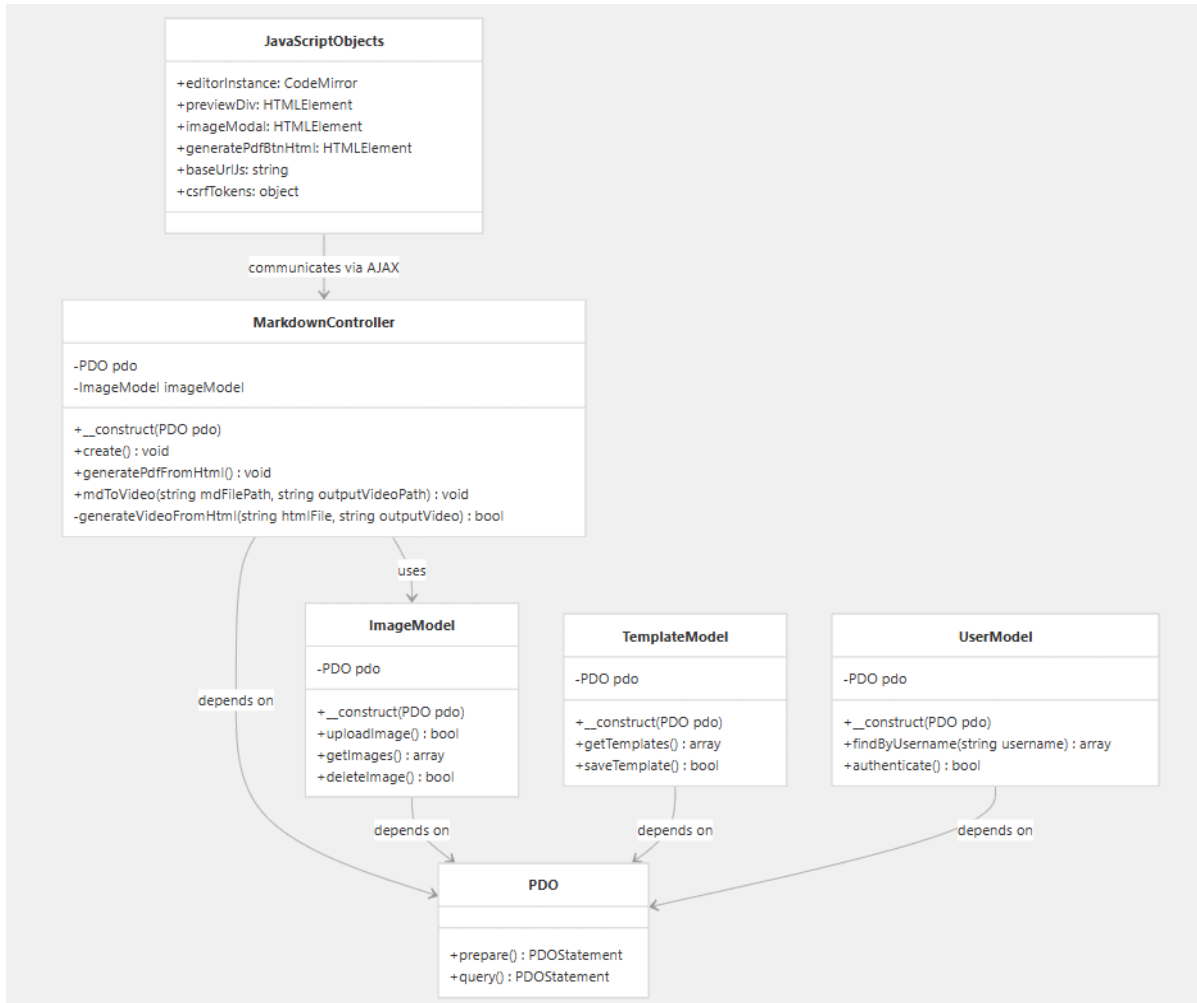


### 3.2.3. Diagrama de Colaboración (vista de diseño)

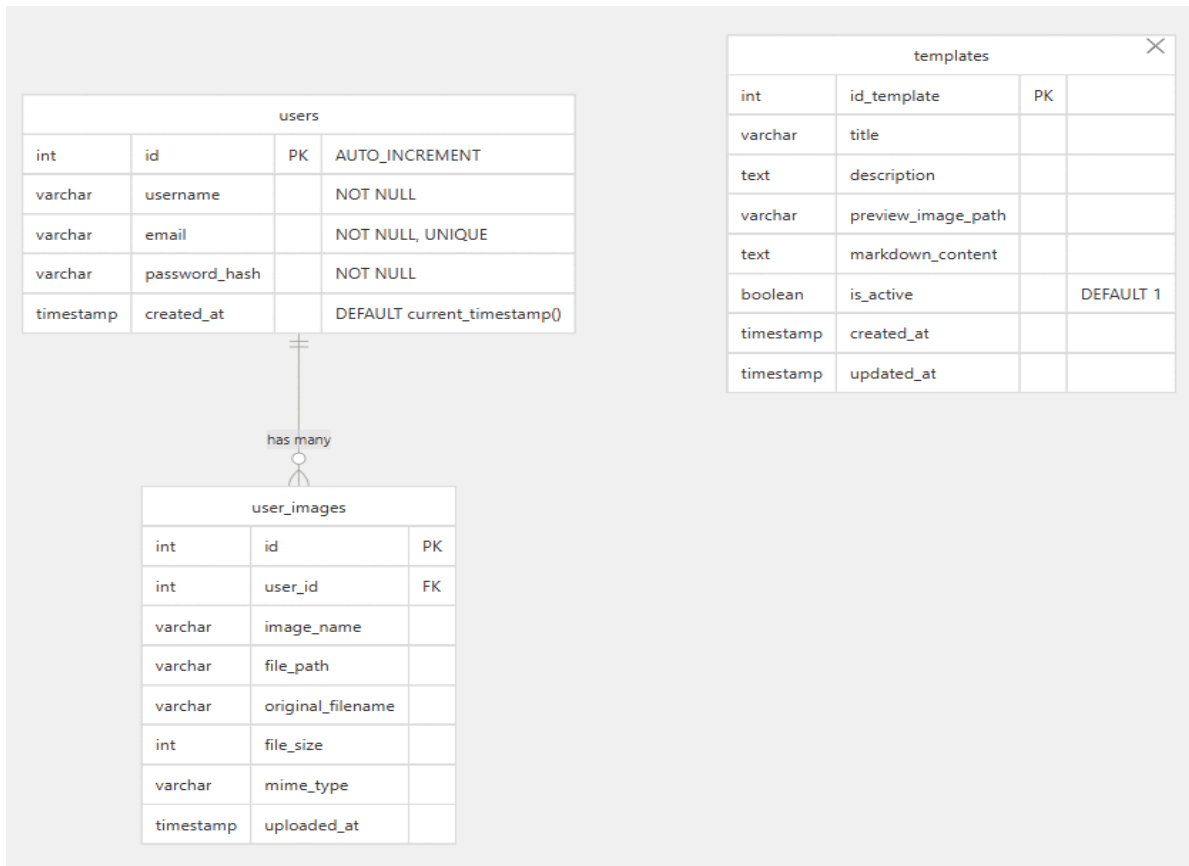
### 3.2.4. Diagrama de Objetos



### 3.2.5. Diagrama de Clases

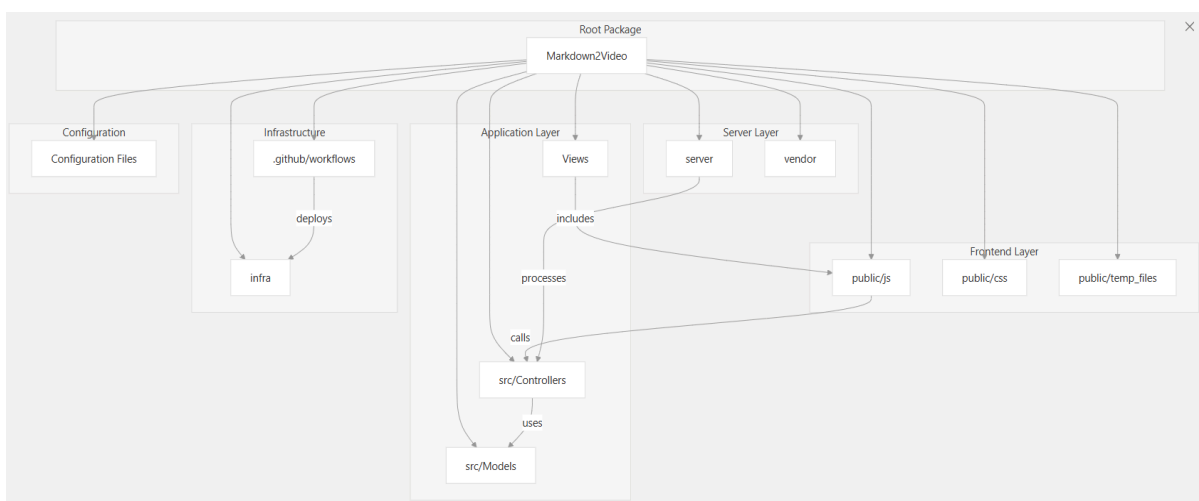


### 3.2.6. Diagrama de Base de datos (relacional o no relacional)



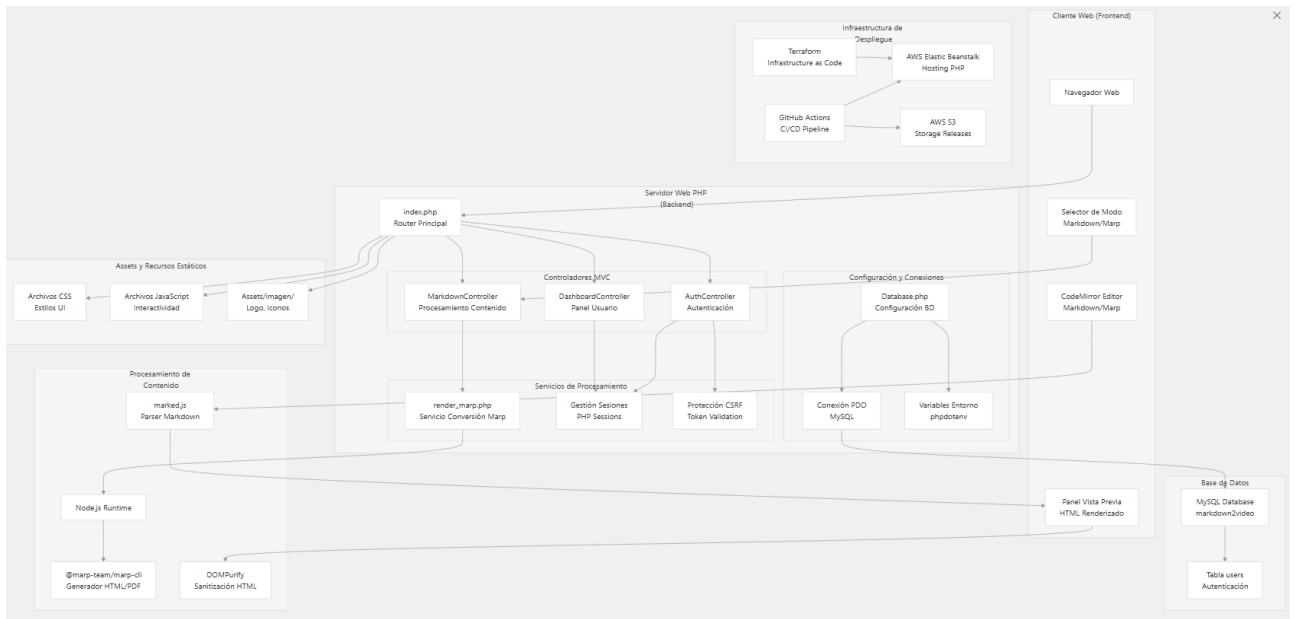
### 3.3. Vista de Implementación (vista de desarrollo)

#### 3.3.1. Diagrama de arquitectura software (paquetes)



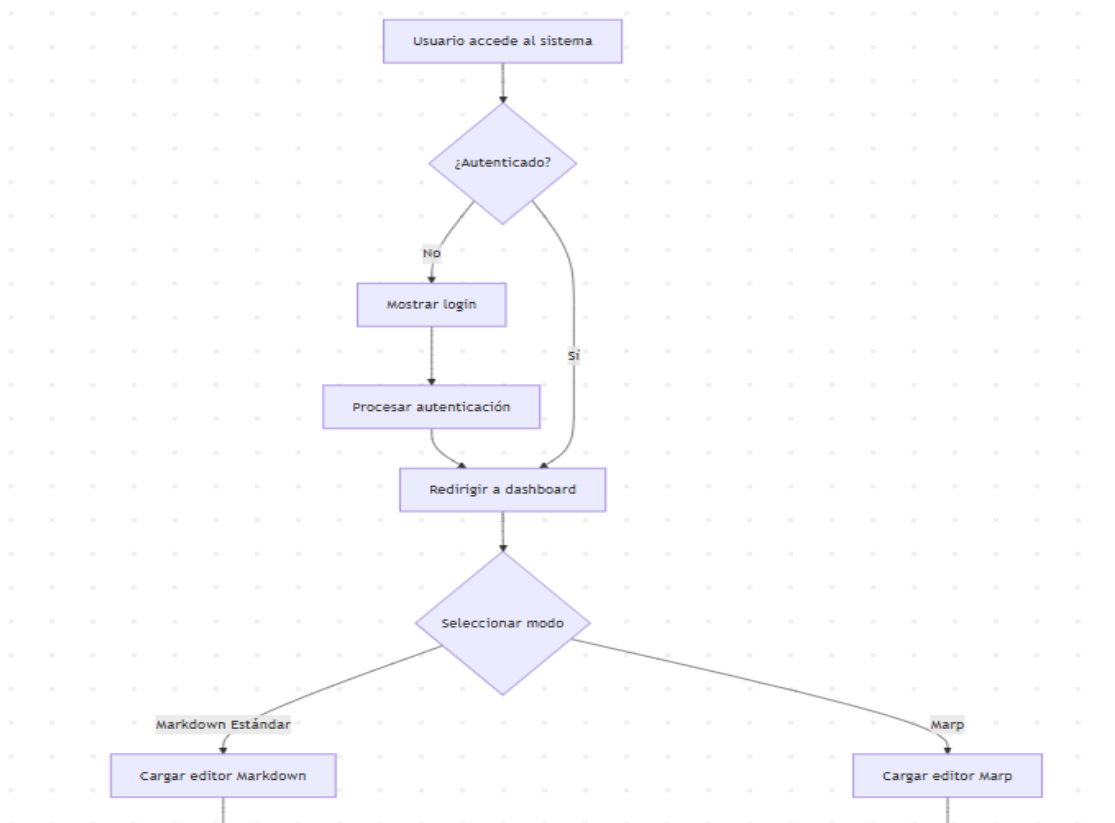


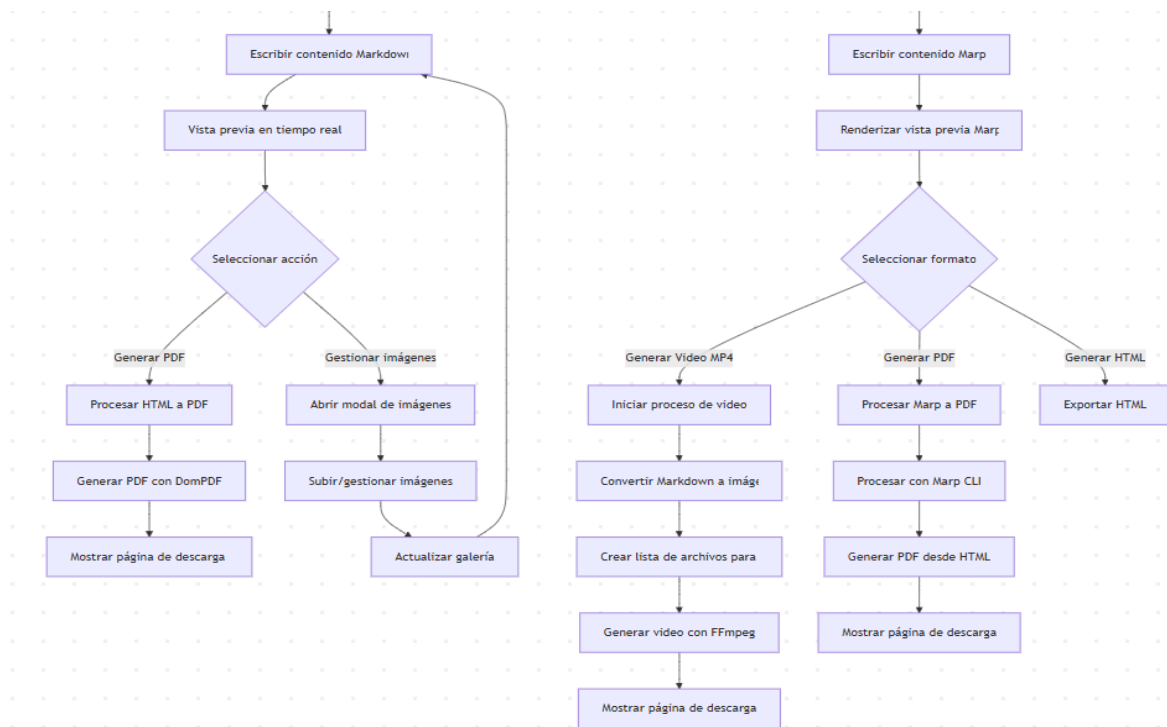
### 3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)



### 3.4. Vista de procesos

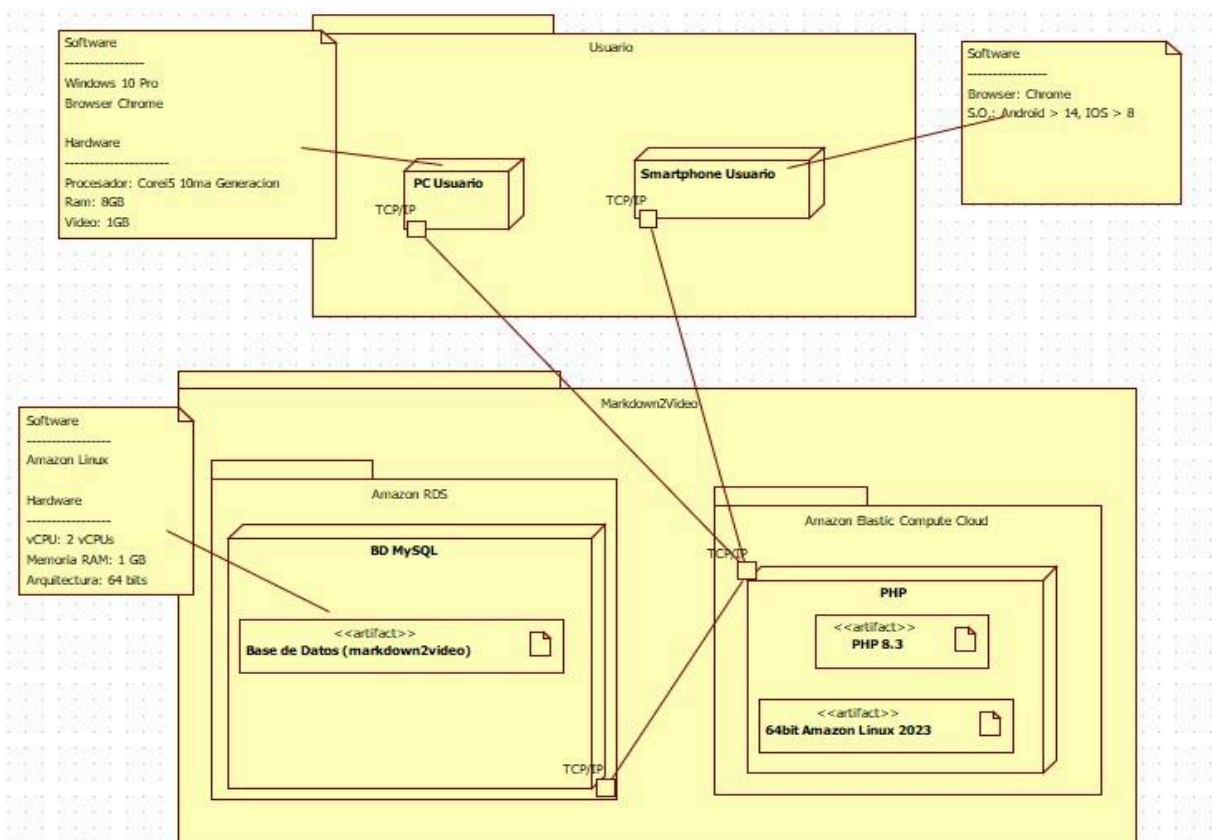
#### 3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)





### 3.5. Vista de Despliegue (vista física)

#### 3.5.1. Diagrama de despliegue



## 4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

### Escenario de Funcionalidad

La funcionalidad del sistema se evalúa en un escenario clave donde un estudiante, operando el sistema en condiciones normales, importa un archivo .md válido que incluye sintaxis de Marp, imágenes y texto. Al hacer clic en "Generar Video MP4", el sistema debe procesar correctamente el archivo, renderizar las diapositivas con sus imágenes correspondientes y generar una narración mediante la tecnología TTS para producir un video final en formato .mp4. La calidad de esta función se mide por la capacidad del sistema para generar dicho video en menos de 5 minutos, asegurando que sea descargable y que su contenido visual y auditivo refleje fielmente el documento Markdown de entrada.

### Escenario de Usabilidad

La usabilidad se centra en la experiencia del usuario, especialmente para aquellos sin experiencia previa con la herramienta. Se considera un escenario en el que un nuevo usuario, sin consultar ninguna documentación, intenta crear su primera presentación a partir de un texto simple y exportarla como PDF. El sistema demostrará ser usable si el usuario puede, de manera intuitiva, encontrar el editor, pegar su texto, observar la previsualización en tiempo real y localizar el botón de exportación para descargar su archivo. El éxito se mide si esta tarea se completa en menos de 3 minutos y con menos de cinco clics principales, sin encontrar mensajes de error que generen confusión.

### Escenario de confiabilidad

La confiabilidad del sistema se prueba ante situaciones de fallo inesperado. Un escenario crítico ocurre cuando un componente externo, como el proceso de FFmpeg, falla a mitad de una conversión de video debido a una imagen corrupta. En esta situación, el subsistema de conversión debe ser capaz de detectar el error sin que la aplicación se bloquee, registrar el incidente en los logs del sistema, eliminar los archivos temporales generados y mostrar un mensaje claro y útil al usuario, como "Error al generar el video. Por favor, revise el formato de sus imágenes". La medida de confiabilidad es la capacidad del

sistema para recuperarse del fallo en menos de dos segundos, sin corromper otros procesos y notificando al usuario sin exponer detalles técnicos.

### **Escenario de rendimiento**

El rendimiento del sistema se evalúa bajo condiciones de carga máxima. Se plantea un escenario donde 20 usuarios concurrentes inician sesión y acceden a su historial de presentaciones simultáneamente, simulando un pico de uso como el inicio de una clase. El servidor de aplicaciones y la base de datos deben procesar todas estas solicitudes de autenticación y consultas de manera eficiente. El rendimiento se considerará aceptable si el tiempo de carga de la página de historial para el 95% de los usuarios es inferior a 4 segundos

### **Escenario de mantenibilidad**

La mantenibilidad se refiere a la facilidad con la que el sistema puede ser modificado o extendido. Un escenario de prueba para este atributo consiste en que un desarrollador necesite añadir una nueva funcionalidad, como la opción de "Exportar como GIF animado". Gracias a una arquitectura de software modular, el desarrollador debería poder implementar este cambio creando una nueva clase conversora, modificando el controlador y el servicio correspondientes, y añadiendo el botón en la interfaz. La alta mantenibilidad se confirma si el cambio se puede realizar afectando un máximo de cuatro archivos, sin impactar la lógica de las funciones de exportación existentes, y si la tarea completa, incluyendo las pruebas, se finaliza en menos de un día de trabajo.

### **Otros Escenarios**

Finalmente, otro aspecto del rendimiento se mide en la capacidad de respuesta de la interfaz durante el uso normal. Este escenario se enfoca en la experiencia de edición: cuando un usuario escribe texto en el editor de Markdown, la ventana de previsualización en tiempo real tiene que actualizarse para reflejar los cambios de manera fluida. La calidad de esta interacción se medirá por una latencia inferior a 300 milisegundos entre la pulsación de una tecla y la actualización visible en la pantalla de previsualización para una presentación de tamaño promedio.