



**UNIVERSIDAD DE
CÓRDOBA**



LICENCIATURA EN
INFORMÁTICA

Acreditada de Alta Calidad
MEN Res. 10710 25/05/17

Documento
técnico
para
proyectos
de Diseño
de Software

Documento de Propuesta de Diseño de Software I, II y III

VIDEO RECORDING

AUTORES:

- Bruno Aguirre Mayerlis Paola
- Briceño Ojeda Cesar Luis
- Moreno Zapata Yerson David
- Álvarez Ortega José Fernando
- Flórez Causil Arturo José

TUTOR:

Alexander Enrique Toscano Ricardo.



BREVE RESEÑA

Diseñar un componente para la plataforma administradora de contenidos que se encargue de grabar pantalla, permitiendo la grabación de esta misma por completo o de una región en específico según la necesidad del usuario ofreciendo diferentes resoluciones y calidades de vídeo con la posibilidad de grabar audio del sistema o del micrófono local almacenando las grabaciones en un formato de video compatible (MP4, AVI, etc.). Organizando las grabaciones por fecha y hora o nombres de archivo Ofreciendo opciones para compartir las grabaciones por correo electrónico, plataformas en la nube o enlaces directos, implementando herramientas básicas de edición como cortar, unir y eliminar partes de las grabaciones. Dando opción de añadir títulos, créditos y marcas de agua a las grabaciones.

Ofreciendo opciones para ajustar el brillo, contraste y volumen de las grabaciones y botones de fácil acceso para iniciar, detener y pausar la grabación.

Mostrar información detallada como el tiempo real de la duración de la grabación, la fecha (hora, día, mes, año), el peso del archivo, el formato y el espacio disponible en disco. Permitiendo la configuración de las opciones de grabación antes de iniciar el proceso.

Tabla de contenido

Documento de Propuesta de Diseño de Software I, II y III	2
Tabla de contenido	4
Etapas 1 Diseño de la Aplicación y Análisis de Requisitos	7
Introducción	7
Propósito del Documento	7
Etapas 1 Diseño de la Aplicación y Análisis de Requisitos	7
Etapas 2: Persistencia de Datos con Backend – Servidor	7
Etapas 3: Consumo de Datos y Desarrollo Frontend – Cliente	8
Alcance del Proyecto	8
Definiciones y Acrónimos	9
Descripción General	10
Objetivos del Sistema	10
Interfaz	10
Conceptos de las entidades	11
Funcionalidad General	11
Usuarios del Sistema	11
Restricciones	11
Requisitos Funcionales	11
Casos de Uso	11
Descripción detallada de cada caso de uso	13
Diagramas de Secuencia	25
Prioridad de Requisitos	30
Requisitos No Funcionales	31
Requisitos de Desempeño	32

Requisitos de Seguridad	32
Requisitos de Usabilidad	32
Requisitos de Escalabilidad	32
Modelado E/R	32
Caracterización de los datos Diagrama de Entidad-Relación	32
Diagrama relacional	32
Descripción de Entidades y Relaciones	32
Reglas de Integridad	32
Anexos (si es necesario)	32
Diagramas Adicionales	32
Referencias	32
Etapla 2: Persistencia de Datos con Backend	33
Introducción	33
Diseño de la Arquitectura de Backend	33
Elección de la Base de Datos	33
Implementación del Backend	34
Conexión a la Base de Datos	34
Pruebas del Backend	34
Etapla 3: Consumo de Datos y Desarrollo Frontend	35
Introducción	35
Creación de la Interfaz de Usuario (UI)	36
Programación Frontend con JavaScript (JS)	36
Consumo de Datos desde el Backend	37
Interacción Usuario-Interfaz	37
Pruebas y Depuración del Frontend	37
Implementación de la Lógica de Negocio en el Frontend	38

Etapas 1 Diseño de la Aplicación y Análisis de Requisitos

Introducción

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación de software de tipo educativo, comercial, OVA, componente o módulo de aplicaciones. Se divide en tres etapas para facilitar el entendimiento y aplicación a gran escala en la asignatura de diseño de software.

Etapas 1 Diseño de la Aplicación y Análisis de Requisitos

Esta etapa cumple la tarea de recoger todas las competencias desarrolladas en todas las áreas de formación del currículo de la licenciatura en Informática y Medios Audiovisuales y ponerlas a prueba en el diseño y análisis de un producto educativo que se base en las teorías de aprendizaje estudiadas, articule las estrategias de enseñanza con uso de TIC y genere innovaciones en educación con productos interactivos que revelen una verdadera naturaleza educativa. Estos productos deben aprovechar las fortalezas adquiridas en las áreas de tecnología e informática, técnicas y herramientas, medios audiovisuales y programación y sistemas, para generar productos software interactivos que permitan a los usuarios disfrutar de lo que aprenden, a su propio ritmo. Todo esto en el marco de un proceso metodológico (metodologías de desarrollo de software como MODESEC, SEMLI, etc.) que aproveche lo aprendido en la línea de gestión y lo enriquezca con elementos de la Ingeniería de Software.

Etapas 2: Persistencia de Datos con Backend – Servidor

En la etapa 2 se continúa con los lineamientos de la etapa 1, para seguir adicionando elementos de diseño e implementación de software, enfocados en el desarrollo de APIs, servidores o microservicios que permitan soportar aplicaciones cliente del software educativo; en este sentido, el curso presenta los conceptos de los sistemas de bases de datos, su diseño lógico, la organización de los sistemas manejadores de bases de datos, los lenguaje de definición de datos y el lenguaje de manipulación de datos SQL y NoSQL; de tal manera que los estudiantes adquieran las competencias para analizar, diseñar y desarrollar aplicaciones para gestionar y almacenar grandes cantidades de datos, mediante el uso de técnicas adecuadas como el diseño y modelo lógico y físico de base datos, manejo de los sistemas de gestión de bases de datos, algebra relacional, dominio del lenguaje SQL como herramienta de consulta, tecnología cliente / servidor; igualmente, se definirán los elementos necesarios para el acceso a dichas bases de datos, como la creación del servidor API, utilizando tecnologías de vanguardia como node.js, express, Nest.js, Spring entre otros; para, finalmente converger en el despliegue de la API utilizando servicios de hospedaje en la nube, preferiblemente gratuitos. También podrá

implementar servidores o API 's con inteligencia artificial o en su defecto crear una nueva capa que consuma y transforme los datos obtenidos de la IA. El desarrollo del curso se trabajará por proyectos de trabajo colaborativo que serán evaluados de múltiples maneras, teniendo en cuenta más el proceso que el resultado.

Etapas 3: Consumo de Datos y Desarrollo Frontend – Cliente

La etapa 3 el estudiante está en capacidad de establecer la mejor elección de herramientas de consumo de datos y técnicas en aras de lograr el mejor producto a nivel de software o hardware acorde a los requerimientos funcionales y no funcionales del problema a solucionar. En este punto el estudiante puede consumir los datos a través de un cliente que puede ser una aplicación de celular, una aplicación de escritorio, una página web, IoT(internet de las cosas) o incluso, artefactos tecnológicos. El diseño gráfico es de los requisitos esenciales en la capa de presentación, por lo tanto, se requieren los cursos de diseño gráficos vistos previamente. Los elementos anteriores nos permiten elegir el paradigma y tecnología para desarrollar nuestras aplicaciones, teniendo en cuenta que podríamos desarrollar aplicaciones de tipo cliente.

Alcance del Proyecto

El proyecto consiste en desarrollar un componente de grabación de vídeo, permitiendo la captura de pantalla completa o por zonas, grabación de audio del sistema o micrófono, con opciones de calidad y resolución ajustables, así como herramientas básicas de edición como recorte de video a futuro podría incluirse el envío de vídeos por correo electrónico, edición avanzada, visualización de datos de video y espacio disponible en el sistema; con el fin de mejorar la comunicación, el aprendizaje, la accesibilidad, la colaboración y la documentación de los usuarios.

El alcance del proyecto puede estar sujeto a ajustes según las necesidades y requerimientos adicionales que surjan durante el desarrollo. La flexibilidad y la comunicación efectiva serán clave para el éxito del proyecto. Cabe resaltar que la implementación del componente se llevará a cabo en tres fases durante tres semestres académicos. La primera fase se enfoca en la documentación detallada del componente y sus funcionalidades.

Para esta primera versión se trabajará las siguientes funcionalidades:

- Grabar pantalla completa
- Grabar pantalla por zona
- Capturar audio
- Iniciar grabación
- Pausar Grabación
- Finalizar grabación
- Guardar grabación.

Funcionalidades Futuras

- Descargar grabación
- Buscar Video

- Lista de videos

- Datos de video

Definiciones y Acrónimos

Grabación de pantalla: Proceso de capturar y guardar en formato de vídeo lo que se muestra en la pantalla de una computadora o dispositivo.

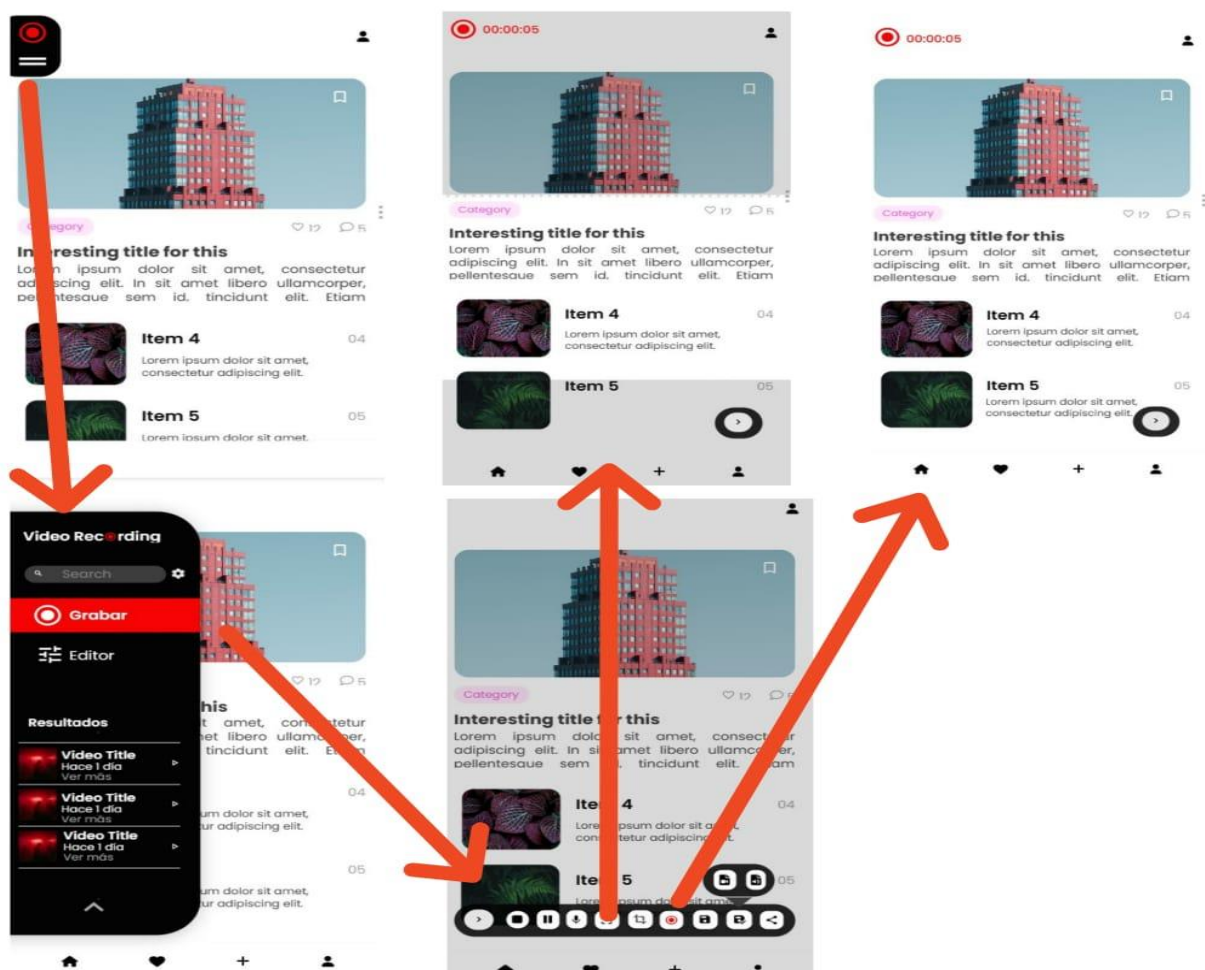
- 1. Resolución de vídeo:** La cantidad de píxeles que componen una imagen de vídeo, determinada por el número de píxeles en anchura y altura (por ejemplo, 1920x1080 píxeles para resolución Full HD).
- 2. Calidad de vídeo:** La medida de la claridad y fidelidad de una grabación de vídeo, que puede estar influenciada por factores como la resolución, la tasa de bits y el códec utilizado.
- 3. Audio del sistema:** El sonido generado por el sistema operativo y las aplicaciones en ejecución en el dispositivo.
- 4. Micrófono local:** Dispositivo de entrada de audio que captura sonido del entorno cercano al dispositivo, como la voz del usuario.
- 5. Formato de vídeo compatible:** Formato de archivo de vídeo que puede ser reproducido por una amplia gama de reproductores y dispositivos, como MP4 (MPEG-4), AVI (Audio Video Interleave), etc.
- 6. Organización por fecha y hora:** Agrupar las grabaciones de pantalla según el momento en que fueron realizadas, facilitando la búsqueda y gestión.
- 7. Edición de vídeo:** Proceso de modificar y mejorar el contenido de vídeo, que puede incluir recorte, unión, eliminación de partes, añadir títulos, créditos y marcas de agua, entre otros.
- 8. Brillo y contraste:** Parámetros que afectan la luminosidad y diferencia entre las partes más claras y oscuras de una imagen de vídeo.
- 9. Volumen de las grabaciones:** Nivel de sonido de las grabaciones de vídeo.
- 10. Inicio, pausa y detención de la grabación:** Acciones para iniciar, pausar y detener el proceso de grabación de pantalla.
- 11. Duración de la grabación:** El tiempo total que ha estado activa la grabación.
- 12. Peso del archivo:** El tamaño en bytes del archivo de vídeo grabado.
- 13. Espacio disponible en disco:** La cantidad de almacenamiento libre en el dispositivo donde se guardarán las grabaciones.
- 14. Opciones de grabación:** Configuraciones que el usuario puede ajustar antes de iniciar el proceso de grabación, como resolución, calidad, fuente de audio, entre otros.

Descripción General

Objetivos del Sistema

El objetivo principal de este componente dentro del software educativo es potenciar la capacidad de los usuarios para capturar y compartir contenido visual de sus pantallas de manera eficiente. Ofrece funcionalidades que van desde la grabación y edición hasta la organización y distribución de dicho contenido. Este componente busca fomentar la creatividad y mejorar la comunicación en el ámbito educativo, permitiendo a educadores y estudiantes crear y compartir material visual de alta calidad de forma sencilla. Facilita la colaboración en proyectos educativos al posibilitar la edición y compartición rápida de contenido entre usuarios

Interfaz



Conceptos de las entidades

Funcionalidad General

Usuarios del Sistema

Los siguientes usuarios pueden interactuar con el sistema dependiendo de las funcionalidades

Funcionalidades	Administrador	Docente	Docente invitado	Alumno
Grabar pantalla completa en tiempo real	✓	✓	✓	✓
Grabar pantalla por zona en tiempo real	✓	✓	✓	✓
Capturar audio en tiempo real	✓	✓	✓	✓
Iniciar grabación	✓	✓	✓	✓
Pausar grabación	✓	✓	✓	✓
Finalizar grabación	✓	✓	✓	✓
Guarda grabación	✓	✓	✓	✓
Descargar grabación	✓	✓	✓	✓
Buscar video	✓	✓	✓	✓
Listar videos	✓	✓	✓	✓
Datos del video	✓	✓	✓	✓

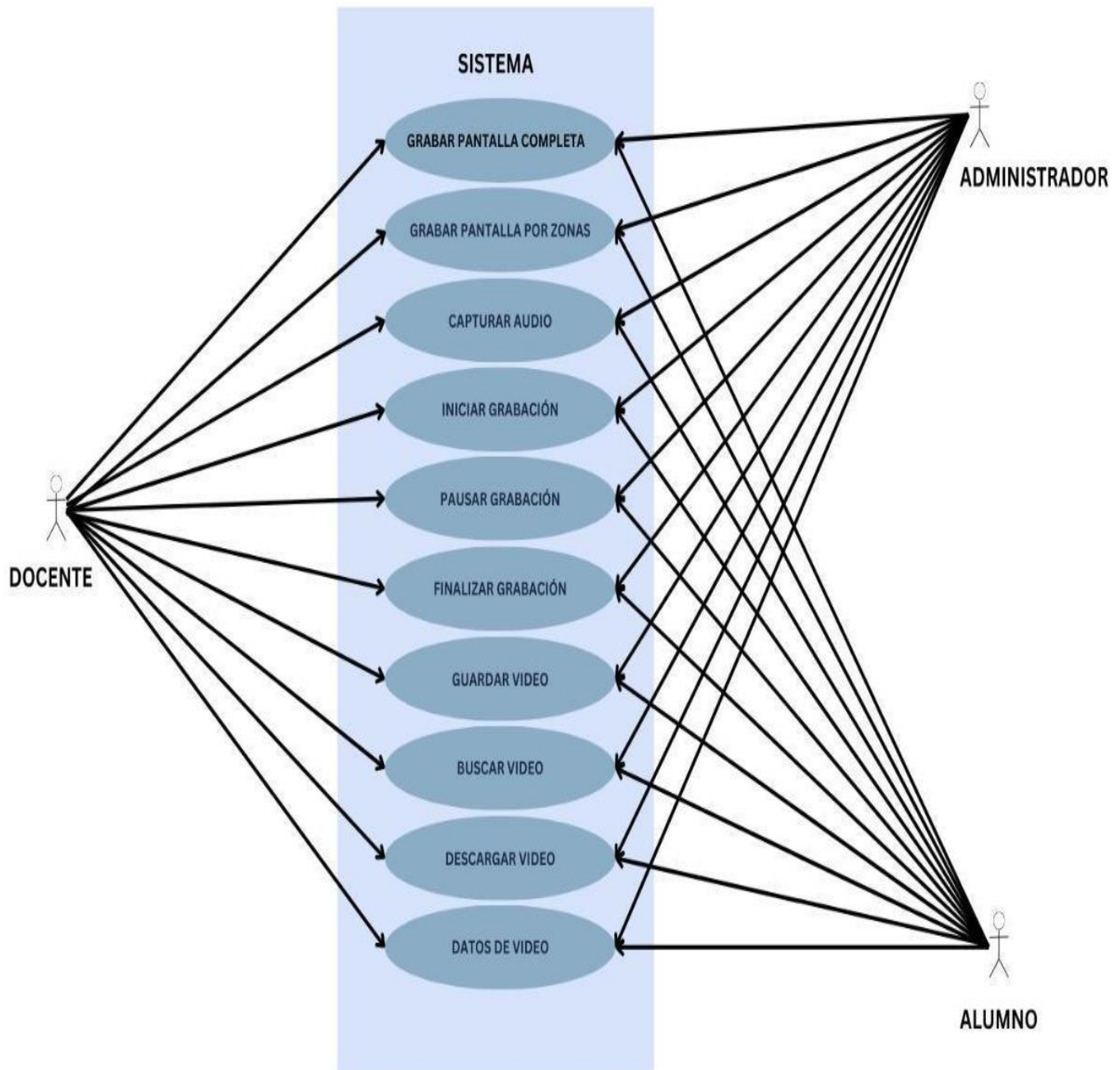
Restricciones

Requisitos Funcionales

Casos de Uso

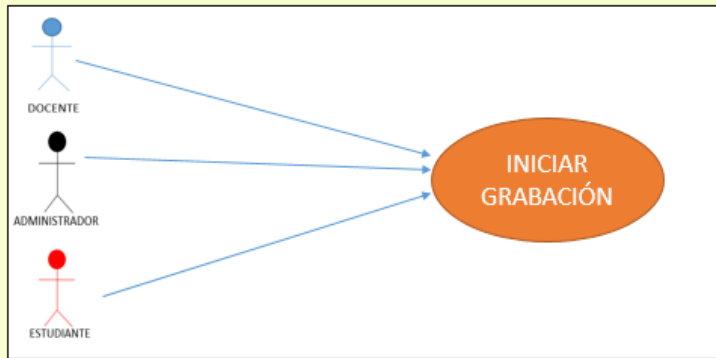
- Grabar pantalla completa
- Grabar pantalla por zona
- Capturar audio
- Iniciar grabación
- Pausar Grabación
- Finalizar grabación
- Guardar grabación
- Descargar grabación
- Buscar Video
- Lista de videos

- Datos de video



Descripción detallada de cada caso de uso

CU-1



Iniciar grabación
 Flujo: Iniciar grabación
 Prueba: Variable iniciar grabación .

Iniciar grabación
 Flujo: SOIG, IG, IIG

Urgencia: 5
 Esfuerzo: 4

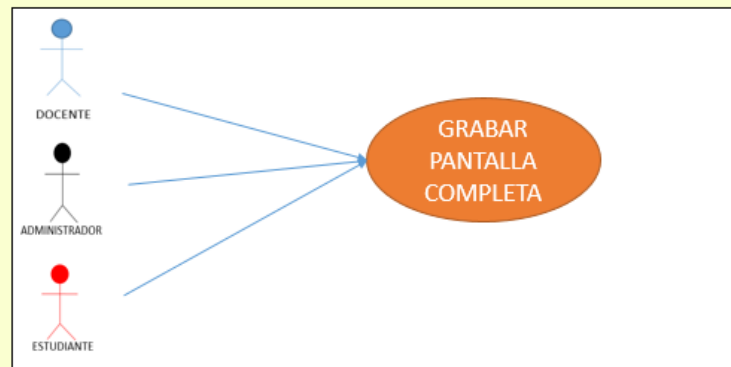
SOIG: Selecciona la opción iniciar grabación
 IG: Iniciar la grabación
 IIG: Indica que inicio la grabación

CASO No. 1 Iniciar grabación

ID:	CU-1	
Nombre	Iniciar grabación	
Actores	Docente - Docente invitado -Administrador-Alumno	
Objetivo	Permitir iniciar una grabación	
Urgencia	5	
Esfuerzo	4	
Precondicione s	Estar autenticado, permisos para grabar audio y video otorgados	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Iniciar grabación”	
		Inicia la grabación
		Indica que inició la grabación
Flujo Alternativo 1	Selecciona la opción “Iniciar grabación”	

		Muestra Advertencia (Falta de espacio, puede grabar, pero no puede guardarse)
	Da la opción de liberar espacio	

CU-2



Urgencia: 5
Esfuerzo: 3

IGPC: Iniciar Grabación de Pantalla Completa
 CU-1: Se efectúa el caso de uso 1
 VG: Visualiza la grabación
 DG: Detiene la grabación
 OGG: Da la opción de guardar grabación
 SGG: Selección guardar la grabación
 MOAG: Muestra opciones de almacenamiento para la grabación
 SGG: Selecciona donde se guardara la grabación
 PGG: Pulsa guardar la grabación
 GG: Guarda la grabación

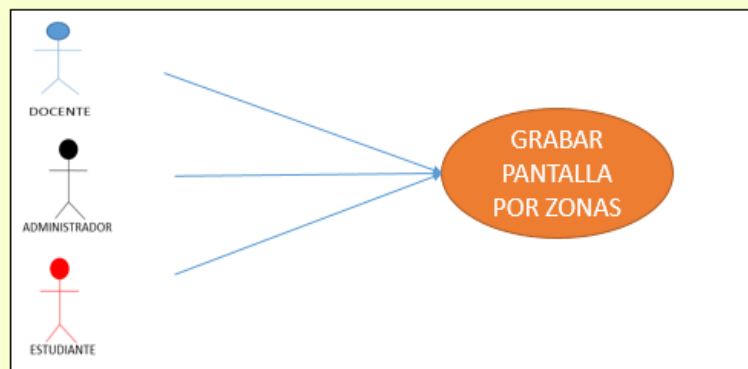
Grabar pantalla completa
 Flujo: Grabar pantalla completa
 Prueba: variable de grabar pantalla completa

GRABAR PANTALLA COMPLETA
 Flujo: IGPC, CU-1, VG, DG, OGG, SGG, MOAG, SGG, PGG, GG

CASO No. 2 Grabar Pantalla completa

ID:	CU-2
Nombre	Grabar Pantalla Completa
Actores	Docente - Docente invitado -Administrador-Alumno
Objetivo	Permitir Grabar pantalla completa
Urgencia	5
Esfuerzo	3
Precondicione s	Estar autenticado, deben haber sido otorgados los permisos de grabar audio y video

Flujo Normal	Actores	Sistema
	Iniciar grabación de pantalla completa	
		Se Efectúa el CU-1
	Visualiza la grabación	
	Detiene la grabación	
		Da opción de guardar la grabación
	Selección opción guardar la grabación	
		Muestra opciones de almacenamiento para la grabación
	Selecciona donde se guardará la grabación	
	Pulsa Guardar la grabación	
		Guarda la grabación
Flujo Alternativo 1	Inicia la grabación	
		Se Efectúa FLUJO 1 ALTERNATIVO CU1
	Da la opción de liberar espacio	



Urgencia: 4

Esfuerzo: 3

SOGZ: Selecciona la opción grabar por zonas

MBDZG: Muestra botón de selección para definir la zona a grabar

DZ: Define la zona

IGZD: Inicia la grabación de la zona definida

IIG: Indica que inició la grabación

DG: Detiene la grabación

CU- 2: Se efectúa el CU-2

VDG: Visualiza datos de la grabación

Grabar pantalla por zonas

Flujo: Grabar pantalla por zonas

Prueba: Seleccionar zona

Grabar pantalla por zonas

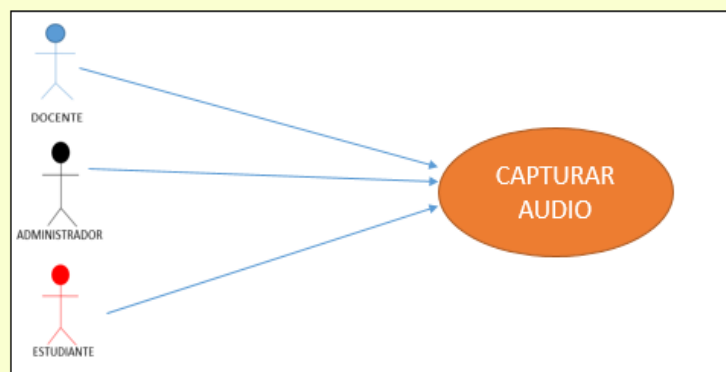
Flujo: SOGZ, MBDZG, DZ, IGZD, IIG, DG, CU- 2, VDG

CASO No. 3 Grabar Pantalla por zonas

ID:	CU-3	
Nombre	Grabar Pantalla por zonas	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir Grabar zonas específicas de la pantalla	
Urgencia	4	
Esfuerzo	3	
Precondicione s	Estar autenticado, haber definido la zona a grabar, permisos de grabar audio y video otorgados	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Grabar por zonas”	
		Muestra botón de selección para definir la zona a grabar
	Define la zona	

		Inicia la grabación de la zona definida
		Indica que inició la grabación
	Detiene la grabación	
		Se Efectúa CU 2
	Visualiza datos de guardado	
Flujo Alternativo 1	Define la zona a grabar	
		Muestra error (Zona fuera de limite)
	Redefine la zona	

CU-4



Urgencia:4

Esfuerzo: 2

SOCA: Selecciona la opción Capturar audio
 IMGGA: Inicia módulo de grabación de audio
 MV: Muestra módulo de volumen
 DG: Detiene la grabación
 OGAG: Da la opción de guardar audio grabado
 SGA: Selecciona guardar audio
 MOA: Muestra opciones de almacenamiento
 SLA: Selecciona lugar de almacenamiento
 GA: Guarda el audio

Capturar audio
 Flujo: Capturar audio
 Prueba: Audio grabado.

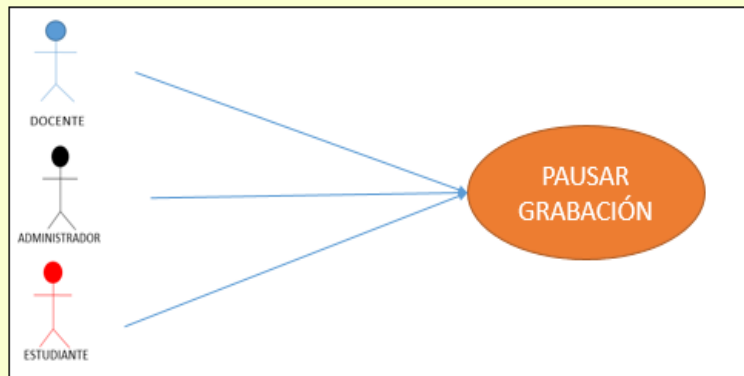
Capturar Audio
 Flujo: SOCA, IMGGA, MV, DG, OGAG, SGA, MOA, SLA, GA

CASO No. 4 Capturar audio

ID:	CU-4	
Nombre	Capturar audio	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir capturar solo el audio	
Urgencia	4	
Esfuerzo	2	
Precondicione s	Estar autenticado, permisos para grabar audio otorgados	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Capturar audio”	
		Inicia módulo de grabación de audio
		Muestra módulo de volumen
	Detiene la grabación	
		Da la opción de guardar audio grabado
	Selecciona guardar audio	
		Muestra opciones de almacenamiento
	Selecciona lugar de almacenamiento	
		Guarda el audio
Flujo Alternativo 1	Selecciona la opción de capturar audio	
		Muestra error (Micrófono no disponible)

	Configura micrófono, Intenta de nuevo	
--	--	--

CU- 5



Urgencia: 3
Esfuerzo: 2

SOPG: Selecciona la opción pausar grabación
PG: Pausa la grabación
IGP: Indica que la grabación ha sido pausada
IGN: Inicia grabación nuevamente
SG: Sigue grabando por donde iba

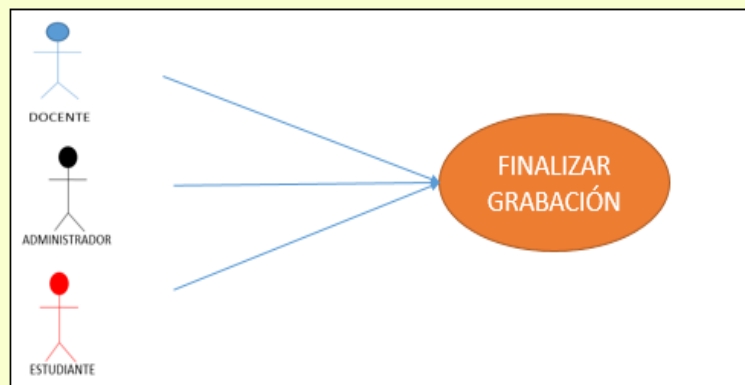
Flujo: Pausar grabación
Prueba: Variable pausar grabación.

Pausar grabación
Flujo: SOPG, PG, IGP, IGN, SG

CASO No. 5 Pausar grabación

ID:	CU-5	
Nombre	Pausar grabación	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir pausar una grabación en curso	
Urgencia	3	
Esfuerzo	2	
Precondiciones	Estar autenticado, una grabación en curso	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Pausar grabación”	
		Pausa la grabación
		Indica que la grabación ha sido pausada
	Inicia grabación nuevamente	
		Sigue grabando por donde iba
Flujo Alternativo 1	Selecciona la opción “Pausar grabación”	
		Muestra advertencia (Grabación avanzada, desea pausar?)
	Cancela la operación	

CU- 6



Urgencia: 5
Esfuerzo: 2

SOFG: Selecciona la opción Finalizar grabación
MOA: Muestra Opción de almacenamiento
SLA: Selecciona lugar de almacenamiento
SGG: Selecciona guardar grabación
GG: Guarda la grabación

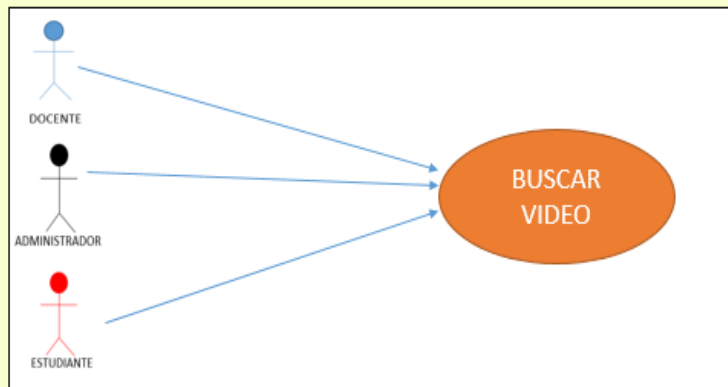
Finalizar grabación
Flujo: Finalizar grabación
Prueba: Video finalizado

Finalizar grabación
Flujo: SOFG, MOA, SLA, SGG, GG

CASO No. 6 Finalizar grabación

ID:	CU-6	
Nombre	Finalizar grabación	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir finalizar una grabación en curso	
Urgencia	5	
Esfuerzo	2	
Precondiciones	Estar autenticado, una grabación en curso	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Finalizar grabación”	
		Muestra Opción de almacenamiento
	Selecciona lugar de almacenamiento	
	Selecciona guardar grabación	Guarda la grabación
Flujo Alternativo 1	Selecciona la opción “Finalizar grabación”	
		Muestra error. Problemas de almacenamiento)
	Cancela la operación	

CU- 7



Urgencia: 3
Esfuerzo: 2

SOBV: Selecciona la opción Buscar videos
ICB: Indica los criterios de la búsqueda
MR: Muestra los resultados

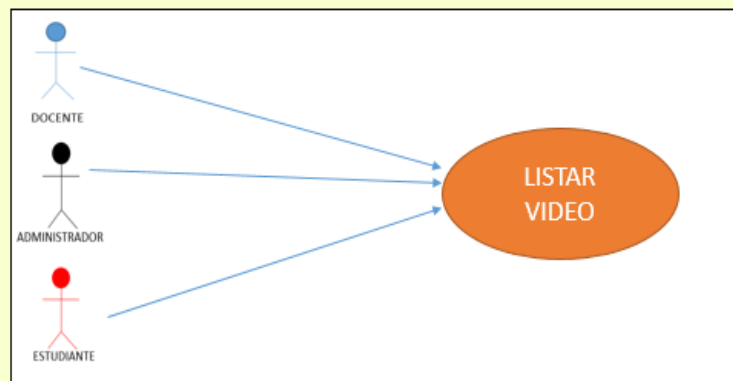
Buscar video
Flujo: Buscar video
Prueba: Variable buscar video.

Buscar video
Flujo: SOBV, ICB, MR

CASO No. 7 Buscar video

ID:	CU-7	
Nombre	Buscar video	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir buscar videos en el sistema	
Urgencia	3	
Esfuerzo	2	
Precondiciones	Estar autenticado	
Flujo Normal	Actores	Sistema
	Selecciona la opción “Buscar videos”	
	Indica los criterios de la búsqueda	
		Muestra los resultados
Flujo Alternativo 1	Selecciona la opción “Buscar videos”	
	Define los criterios de búsqueda	
		Muestra error (No hay resultados)
	Cancela la operación	

CU- 8



Urgencia: 3
Esfuerzo: 2

SOLV: Selecciona la opción listar videos
MLVD: Muestra lista de videos disponibles
SV: Selección video
VV: Visualiza video

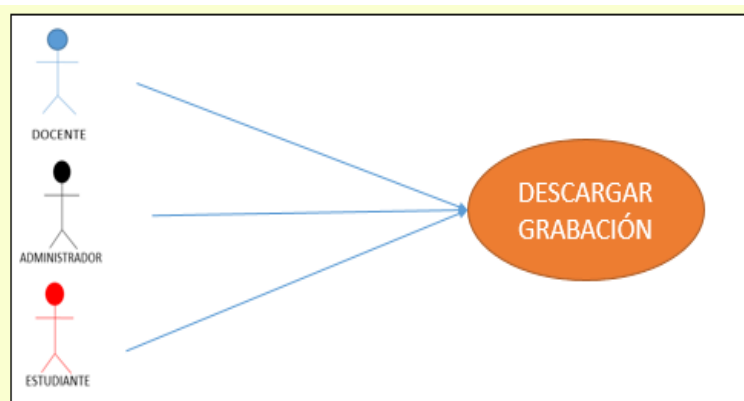
Listar video
Flujo: Listar video
Prueba: Lista de videos.

Listar video
Flujo: SOLV, MLVD, SV, VV

CASO No. 8 Listar video

ID:	CU-8	
Nombre	Listar video	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir buscar videos en el sistema	
Urgencia	3	
Esfuerzo	2	
precondiciones	Estar autenticado	
Flujo Normal	Actores	Sistema
	Selecciona la opción “listar videos”	
		Muestra lista de videos disponibles
	Selección video	
	Visualiza video	
Flujo Alternativo 1	Selecciona la opción “Listar videos”	
		Muestra error (Lista vacía)
	cancela la operación	

CU- 9



Urgencia: 4

Esfuerzo: 3

PLG: Presiona lista de grabaciones

MLGD: Muestra las grabaciones disponibles

SG: Selecciona la grabación

PAD: Prepara el archivo para descargar

SCU: Selecciona carpeta de ubicación

D: Descarga

VD: Visualiza descarga

Flujo: Descargar grabación

Prueba: Variable descargar grabación.

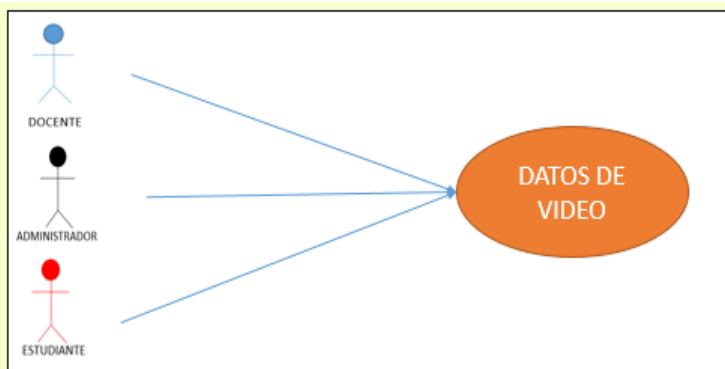
Descargar grabación

Flujo: PLG, MLGD, SG, PAD, SCU, D, VD

CASO No. 9 Descargar grabación

ID:	CU-9	
Nombre	Descargar grabación	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir descargar una grabación	
Urgencia	4	
Esfuerzo	3	
precondiciones	Estar autenticado, tener una grabación disponible	
Flujo Normal	Actores	Sistema
	Presiona lista de grabaciones	
		Muestra las grabaciones disponibles
	Selecciona la grabación	
		Prepara el archivo para descargar
	Selecciona carpeta de ubicación	
		Descarga
Flujo Alternativo 1	Visualiza Descarga	
	Selecciona grabación para descargar	
		Muestra error (Problemas de conexión)
	Cancela la operación	

CU- 10



Urgencia: 3
Esfuerzo: 2

SOVDV: Selecciona la opción ver datos de video
MDVS: Muestra los datos de video seleccionado
VD: Visualiza datos

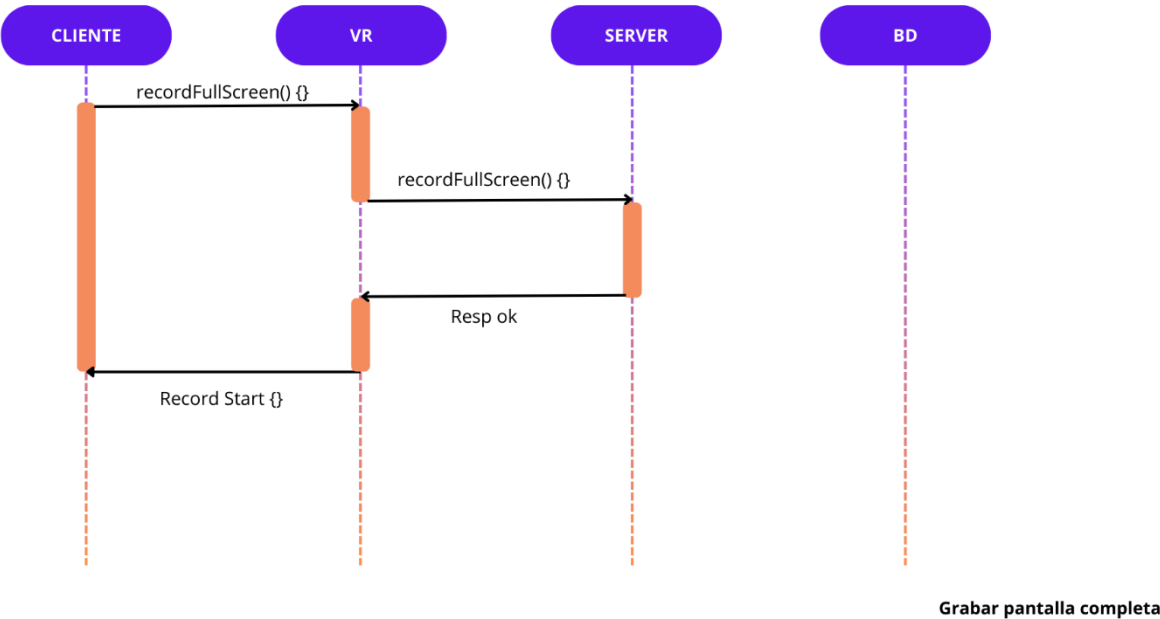
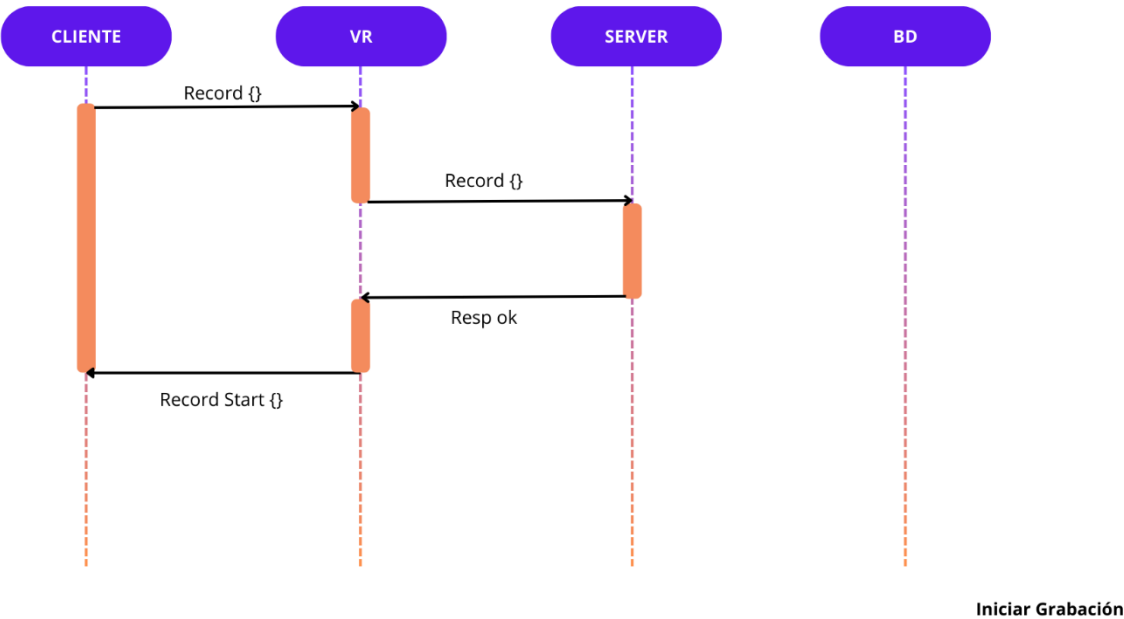
Flujo: Datos de video
Prueba: Variable datos de video.

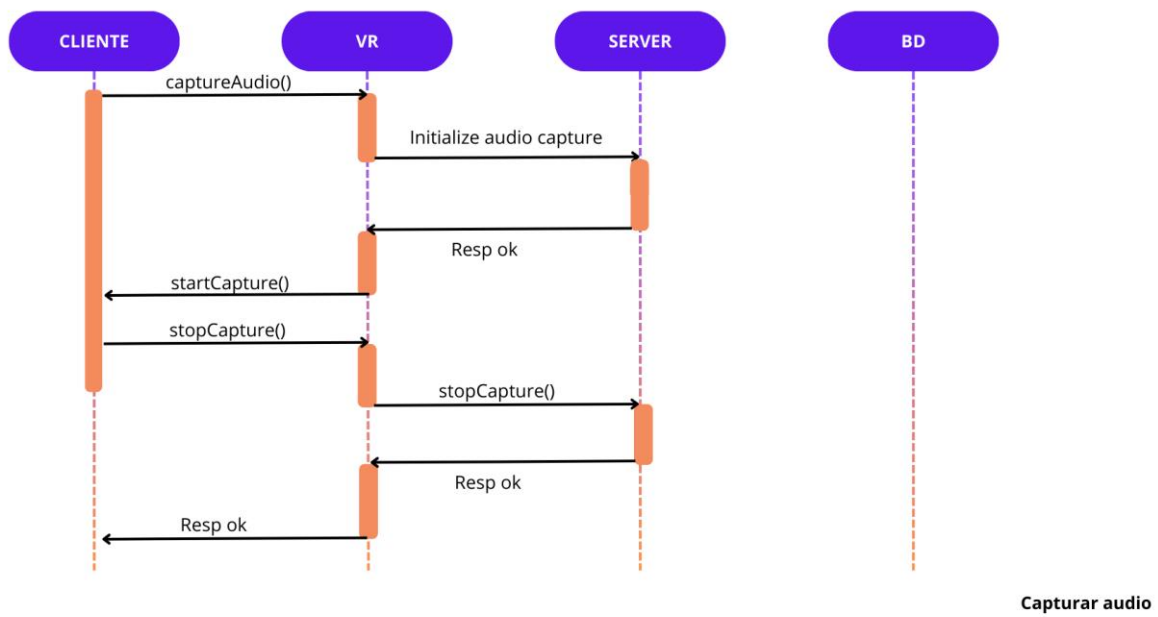
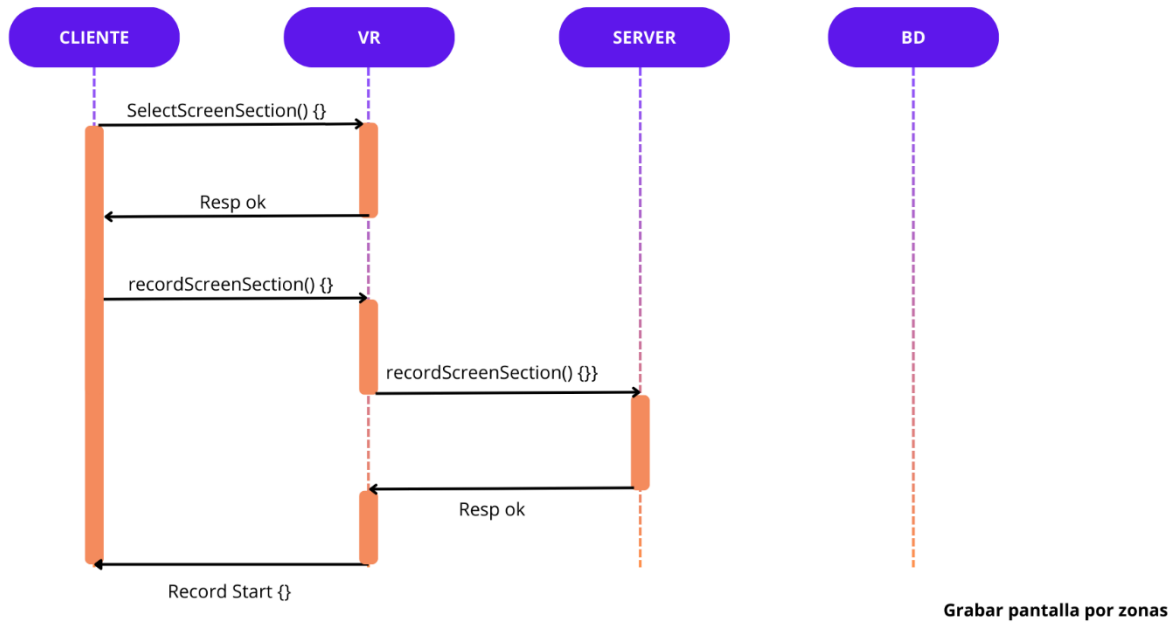
Datos de video
Flujo: SOVDV, MDVS, VD

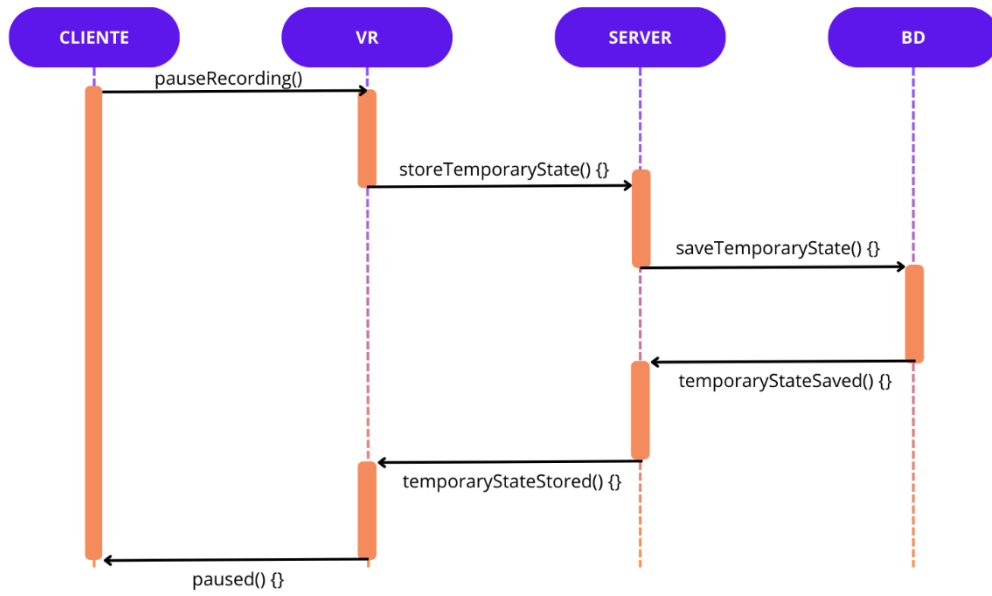
CASO No. 10 Datos de video

ID:	CU-10	
Nombre	Datos de video	
Actores	Docente – Docente invitado -Administrador-Alumno	
Objetivo	Permitir ver los datos de un video grabado	
Urgencia	3	
Esfuerzo	2	
precondiciones	Estar autenticado, tener una grabación disponible	
Flujo Normal	Actores	Sistema
	Selecciona opción de ver datos de video	
		Muestra los datos del video seleccionado
	Visualiza los datos	
Flujo Alternativo 1	Selecciona opción para ver datos de video	
		Muestra error (Datos no disponibles)
	cancela la operación	

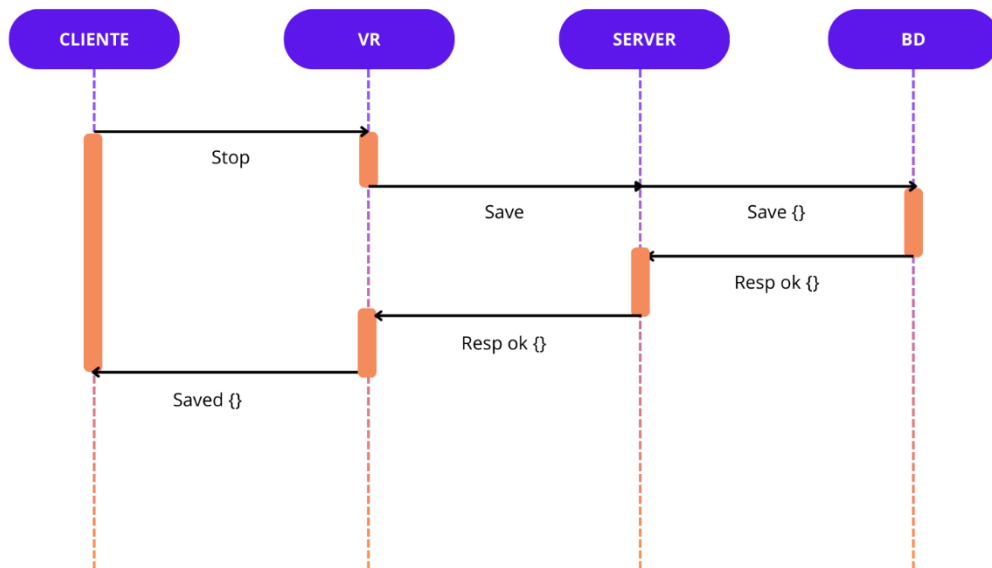
Diagramas de Secuencia



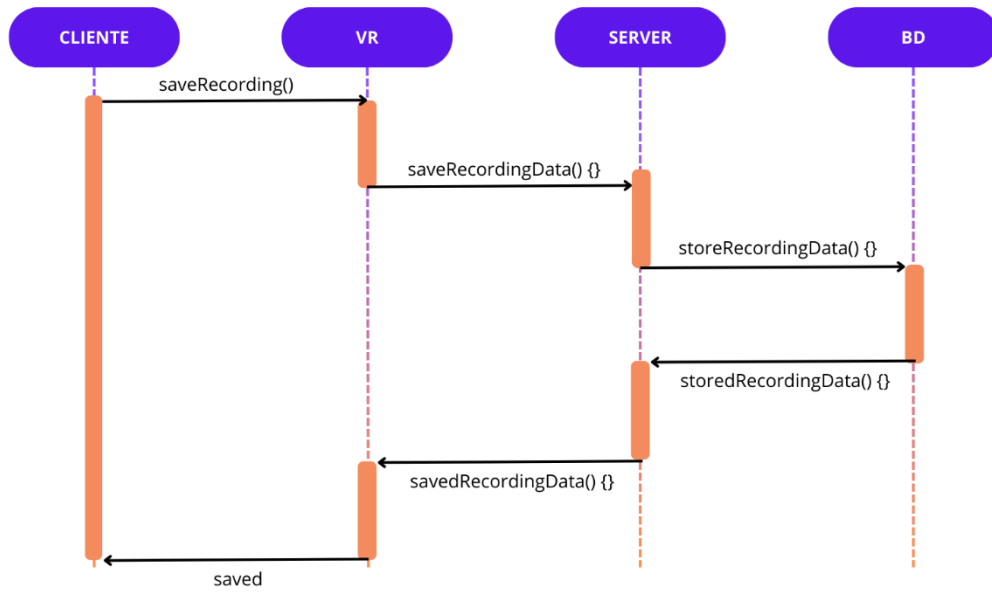




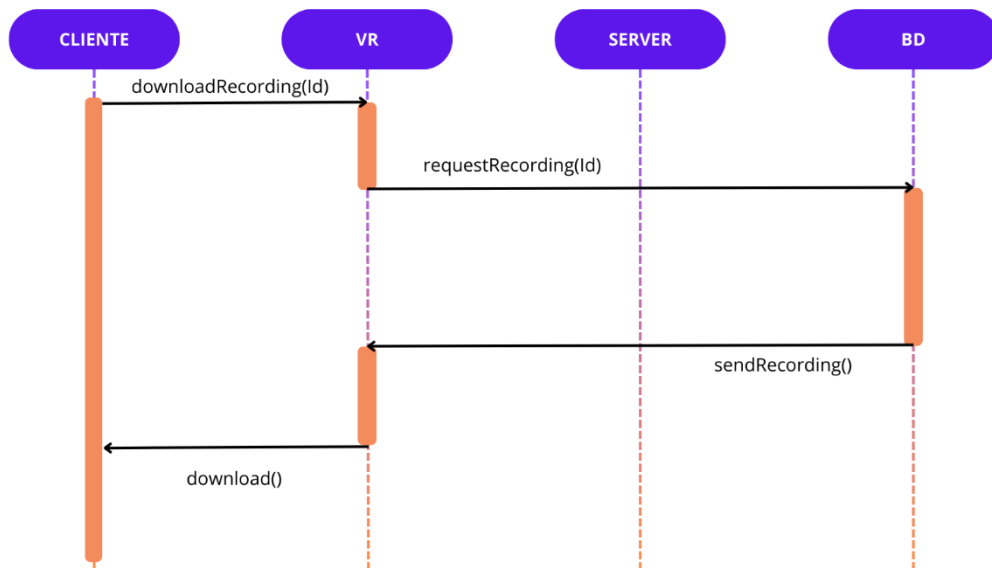
Pausar grabación



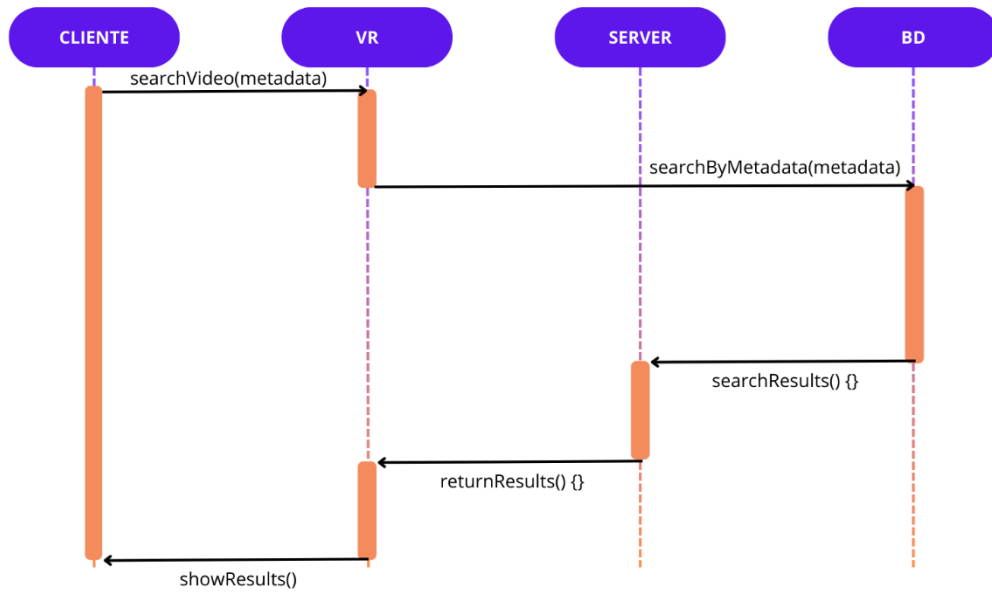
Finalizar grabación



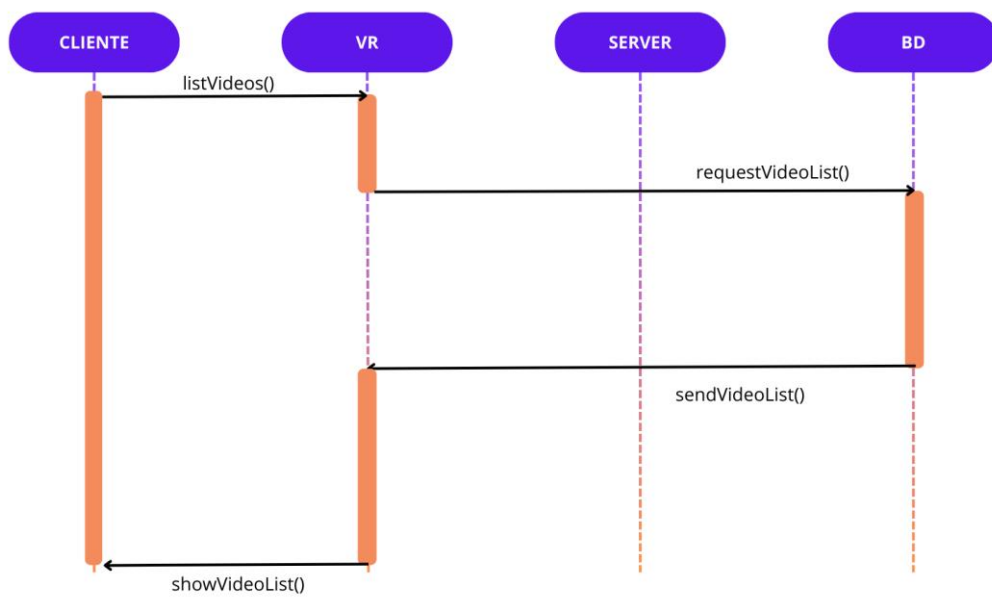
Guardar grabación



Descargar grabación



Buscar video



Lista de videos

Prioridad de Requisitos

A partir del análisis de requerimientos, funcionalidades y el proceso de diseño thinking, se concreta la siguiente matriz de prioridad de requerimientos.

Para la interpretación se tiene en cuenta la siguiente escala con sus valores.

Eje de Urgencia:

- Obligatoria (5)
- Alta (4)
- Moderada (3)
- Menor (2)
- Baja (1)

Eje de Esfuerzo:

- Muy alto (5)
- Alto (4)
- Medio (3)
- Bajo (2)

	Urgencia					
Esfuerzo		1-Baja	2-Menor	3-Moderada	4-Alta	5-Obligatoria
	5-Muy alto	5	10	15	20	25
	4-Alto	4	8	12	16	20
						CU-1
	3-Medio	3	6	9	12	15
					CU-3 CU-9	CU-2
	2-Bajo	2	4	6	8	10
				CU-7 CU-5 CU-8 CU-10	CU-4	CU-6
		1	2	3	4	5

Requisitos No Funcionales

Rendimiento:

El sistema debe poder grabar pantallas a una tasa mínima de 30 cuadros por segundo (FPS) con resoluciones de hasta 1920x1080 (Full HD).

El tiempo de respuesta del sistema para iniciar o pausar una grabación debe ser inferior a 2 segundos.

Las herramientas de edición (cortar, unir, eliminar) deben ejecutarse en menos de 5 segundos para videos de hasta 10 minutos.

Escalabilidad:

El sistema debe ser capaz de manejar múltiples usuarios concurrentes sin afectar el rendimiento, especialmente en la carga y descarga de grabaciones desde la nube.

Se debe permitir la posibilidad de aumentar la capacidad de almacenamiento para los videos según la demanda.

Usabilidad:

La interfaz de usuario debe ser intuitiva y accesible para usuarios con diferentes niveles de experiencia técnica.

Las opciones clave (iniciar, pausar, detener grabación) deben ser accesibles con un máximo de dos clics o interacciones.

El sistema debe ofrecer atajos de teclado para iniciar, pausar y detener grabaciones.

Seguridad:

El sistema debe cumplir con los estándares de cifrado para la transmisión y almacenamiento de los videos en la nube (SSL/TLS).

Solo los usuarios autenticados deben tener acceso a las grabaciones.

Debe haber un control de acceso que garantice que solo ciertos usuarios puedan realizar acciones específicas (ej. administrador puede eliminar videos, usuarios estándar sólo pueden visualizar).

Compatibilidad:

El sistema debe ser compatible con los principales navegadores web (Chrome, Firefox, Edge, Safari).

Las grabaciones deben guardarse en formatos de video ampliamente aceptados como MP4 y AVI, que sean reproducibles en cualquier dispositivo moderno.

Portabilidad:

La aplicación debe ser capaz de ejecutarse en diferentes sistemas operativos, incluyendo Windows, macOS y Linux.

La arquitectura debe permitir su fácil integración con otros módulos o plataformas de administración de contenidos mediante APIs.

Fiabilidad:

El sistema debe asegurar una tasa de éxito del 99.9% en la grabación y almacenamiento de videos.

En caso de fallo de grabación, el sistema debe notificar al usuario y permitir la recuperación automática de archivos temporales.

Mantenibilidad:

El sistema debe seguir principios de diseño modular para facilitar futuras actualizaciones de funcionalidades, tanto en la parte de frontend como backend.

Debe contar con una documentación detallada para desarrolladores que incluya instrucciones de configuración, instalación y pruebas.

Disponibilidad:

El sistema debe estar disponible al 99.5% del tiempo, con interrupciones permitidas solo para tareas de mantenimiento planificado.

Tolerancia a fallos:

En caso de que el sistema se cierre inesperadamente, debe haber mecanismos para reanudar la grabación desde el último punto guardado o recuperar grabaciones no finalizadas.

Requisitos de Desempeño

Velocidad: Latencia máxima de 100 ms para la grabación y procesamiento de video; herramientas de edición deben procesar un video de 10 minutos en menos de 5 segundos.

Uso de memoria: No debe superar el 60% de la memoria disponible en grabaciones Full HD.

Almacenamiento: Videos de 10 minutos en Full HD no deben ocupar más de 1 GB.

Tiempo de respuesta: Iniciar o detener la grabación debe tardar menos de 2 segundos; exportar un video de 5 minutos debe tomar menos de 10 segundos.

Rendimiento en tiempo real: Sin interrupciones en la grabación, con sincronización perfecta entre audio y video.

Subida a la nube: Tiempo de subida menor a 15 segundos por cada 100 MB de video.

Optimización de recursos: Uso del procesador no debe superar el 70% en grabaciones Full HD.

Ancho de banda: Transmisión eficiente sin saturar la red y reproducción en línea con latencia inferior a 200 ms.

Capacidad concurrente: Soporte para al menos 50 usuarios concurrentes sin pérdida de calidad.

Requisitos de Seguridad

Autenticación: Los usuarios deben iniciar sesión con credenciales seguras (contraseñas o autenticación de dos factores) para acceder al sistema de grabación y edición.

Autorización: Diferentes niveles de acceso para usuarios según su rol (administradores, usuarios estándar, etc.), limitando funcionalidades según permisos.

Cifrado de datos: Los videos y archivos de audio deben estar cifrados tanto en almacenamiento como durante la transmisión para evitar accesos no autorizados.

Protección de datos personales: Cumplir con las normativas de privacidad (como GDPR) y evitar la captura de datos personales sin consentimiento.

Almacenamiento seguro: Los archivos se deben almacenar en servidores seguros con copias de seguridad y protección contra accesos no autorizados.

Control de sesiones: Las sesiones de usuario deben expirar después de un tiempo de inactividad y no se debe permitir múltiples inicios de sesión desde diferentes ubicaciones.

Registro de actividad: Registrar eventos críticos como inicio de sesión, intentos de acceso no autorizado, y modificaciones en archivos para auditorías de seguridad.

Protección contra ataques: Implementar protección contra ataques de fuerza bruta, inyecciones de código y accesos indebidos a las APIs del sistema.

Respaldo y recuperación: Garantizar que los datos de grabaciones y ajustes del usuario puedan ser restaurados en caso de pérdida o corrupción.

Requisitos de Usabilidad

Interfaz intuitiva: Se asegurará que el diseño sea limpio y sencillo, para que cualquier usuario, sin importar su nivel de experiencia, pueda navegar fácilmente por las opciones disponibles.

Minimización de pasos: Se asegurará que el flujo de trabajo esté diseñado de manera eficiente, permitiendo a los usuarios grabar y gestionar sus archivos con el menor número de pasos posible.

Accesibilidad: Se considerará la diversidad de los usuarios, incluyendo características como atajos de teclado y opciones de accesibilidad para facilitar el uso a personas con discapacidades.

Coherencia visual y funcional: Se mantendrá un diseño consistente en toda la plataforma, tanto en términos visuales como funcionales, para que los usuarios puedan familiarizarse rápidamente con la estructura y las herramientas.

Interacción fluida: Se asegurará que la respuesta de las acciones dentro de la plataforma sea rápida y coherente, evitando retrasos o comportamientos inesperados que puedan confundir al usuario.

Requisitos de Escalabilidad

Adaptabilidad al crecimiento de usuarios: Se asegurará que la plataforma soporte un aumento en el número de usuarios simultáneos sin degradar su rendimiento.

Ampliación de funcionalidades: Se diseñará la arquitectura de manera modular para permitir la incorporación de nuevas características sin afectar las existentes.

Gestión eficiente de recursos: Se optimizarán los recursos del sistema para manejar una mayor carga de trabajo a medida que la demanda crezca.

Soporte para múltiples resoluciones y calidades: Se garantizará que la plataforma maneje eficientemente grabaciones en diferentes resoluciones y calidades, sin comprometer la experiencia del usuario.

Modelado E/R

Caracterización de los datos Diagrama de Entidad-Relación

1 Grabar Pantalla Completa

- Entidades:

1. **Grabación de Pantalla**

- Atributos: ID, Nombre de archivo, Duración, Fecha y hora de inicio

2. **Usuario**

- Atributos: Nombre de usuario, Tipo de usuario

- Relaciones:

- **Usuario realiza Grabación**

- Relación de uno a muchos: Un usuario puede realizar múltiples grabaciones de pantalla.

2. grabar Pantalla Por Zonas

Entidades:

- **Usuario**

- Atributos: Nombre de usuario, Tipo de usuario

- **Grabación por Zonas**

- Atributos: ID, Nombre de archivo, Duración, Fecha y hora de inicio, Coordenadas de inicio, Coordenadas de fin

Relaciones:

- **Usuario realiza Grabación por Zonas**

- Relación uno a muchos

(1): Un usuario puede realizar varias grabaciones por zonas.

3. Capturar Audio

- **Usuario**

- Atributos: Nombre de usuario, Tipo de usuario

- **Captura de Audio**
 - Atributos: ID, Nombre de archivo, Duración, Fecha y hora de inicio

Relaciones:

- **Usuario realiza Captura de Audio**
 - Relación uno a muchos

(1): Un usuario puede realizar varias capturas de audio.

4. iniciar Grabación

- **Usuario**
 - Atributos: Nombre de usuario , Tipo de usuario
- **Iniciar Grabación**
 - Atributos: ID de Inicio ,Fecha y hora de inicio
 - Relaciones:
 - ID de Usuario referencia a Usuario

Relaciones:

- **Usuario realiza Iniciar Grabación**
 - Relación uno a muchos

(1): Un usuario puede iniciar varias sesiones de grabación de pantalla.

5. Pausar Grabación.

Entidades:

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Pausa de Grabación**
 - Atributos: ID, Fecha y hora de inicio de pausa, Fecha y hora de fin de pausa

Relaciones:

- **Usuario realiza Pausa de Grabación**
 - Relación uno a muchos

(1): Un usuario puede realizar varias pausas durante una grabación.

6. Finalizar Grabación

Entidades:

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Finalización de Video**
 - Atributos: ID, Nombre del video, Duración total, Fecha y hora de finalización

Relaciones:

- **Usuario finaliza Video**
 - Relación uno a muchos

(1): Un usuario puede finalizar varios videos.}

7. Guardar grabación.

Entidades:

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Guardar grabación**
 - Atributos: ID, Nombre del video, Duración total, Fecha y hora de grabación

Relaciones:

- **Usuario guarda grabación**
 - Relación uno a muchos

(1): Un usuario puede guardar varios videos.}

8. Descargar Grabación

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Descarga de Grabación**
 - Atributos: ID, Nombre de archivo, Fecha y hora de descarga

Relaciones:

- **Usuario realiza Descarga de Grabación**
 - Relación uno a muchos

(1): Un usuario puede descargar varias grabaciones.

9. Buscar Video

Entidades:

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Buscar Video**
 - Atributos: ID, Término de búsqueda, Fecha y hora de búsqueda

Relaciones:

- **Usuario realiza Búsqueda de Video**
 - Relación uno a muchos

(1): Un usuario puede realizar varias búsquedas de video.

10. Listar Video

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Lista de Video**
 - Atributos: ID, Nombre del video, Categoría, Duración, Fecha de publicación

Relaciones:

- **Usuario lista Video**
 - Relación uno a muchos

(1): Un usuario puede listar varios videos.

11. Datos De Video

- **Usuario**
 - Atributos: Nombre de usuario, Tipo de usuario
- **Datos de Video**
 - Atributos: ID, Nombre del video, Duración, Formato, Tamaño, Fecha de creación

Relaciones:

- **Usuario accede a Datos de Video**
 - Relación uno a muchos

(1): Un usuario puede acceder a los datos de varios videos.

Diagrama de Entidad-Relación

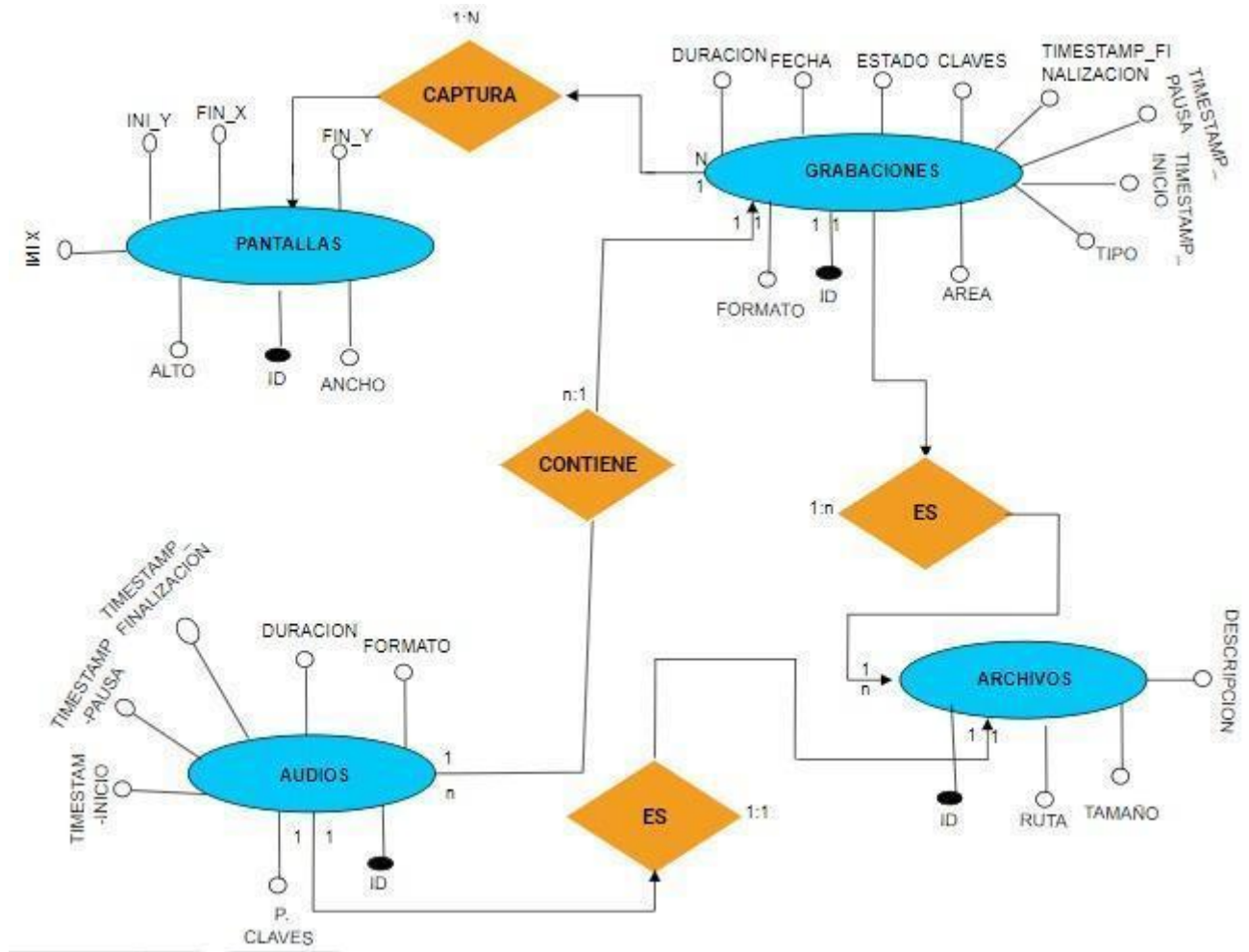
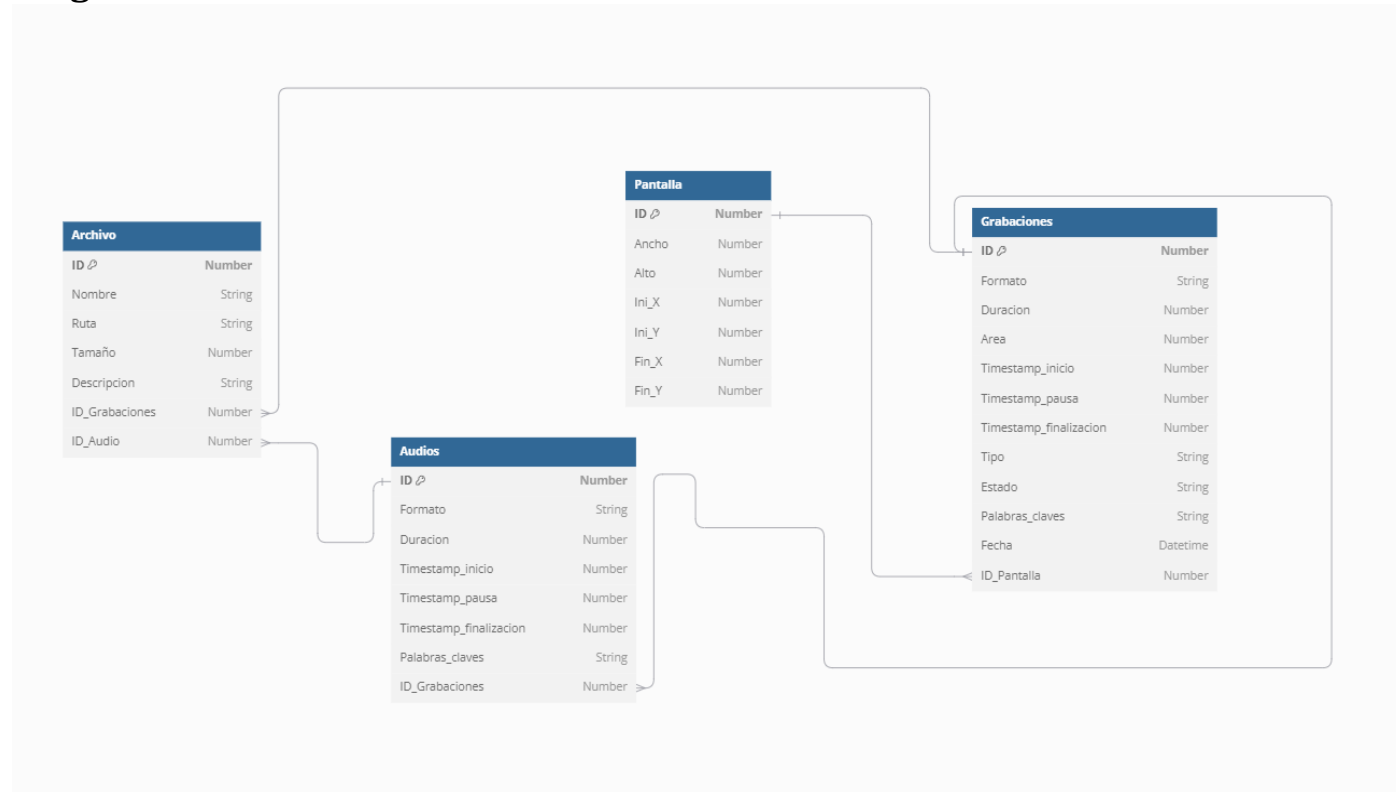


Diagrama relacional



Script del modelo relacional :

```

Table Archivo {
  ID integer [primary key]
  Nombre varchar
  Ruta varchar
  Tamaño integer
  Descripcion varchar
  ID_Grabaciones integer [ref: > Grabaciones.ID]
  ID_Audio integer [ref: > Audios.ID]
}
  
```

```

Table Pantalla {
  ID integer [primary key]
  Ancho integer
  Alto integer
  Ini_X integer
  Ini_Y integer
  Fin_X integer
  Fin_Y integer
}
  
```



```

Table Audios {
  ID integer [primary key]
  Formato varchar
  Duracion integer
  Timestamp_inicio integer
  Timestamp_pausa integer
  Timestamp_finalizacion integer
  Palabras_claves varchar
  ID_Grabaciones integer [ref: > Grabaciones.ID]
}

```

```

Table Grabaciones {
  ID integer [primary key]
  Formato varchar
  Duracion integer
  Area integer
  Timestamp_inicio integer
  Timestamp_pausa integer
  Timestamp_finalizacion integer
  Tipo varchar
  Estado varchar
  Palabras_claves varchar
  Fecha datetime
  ID_Pantalla integer [ref: > Pantalla.ID]
}

```

Descripción de Entidades y Relaciones : **Archivo**

Descripción: Esta entidad representa un archivo multimedia que puede contener información de una grabación o de un audio específico. Se incluyen atributos como el nombre, ruta, tamaño y descripción del archivo.

Atributos:

ID (Number): Identificador único del archivo.

Nombre (String): Nombre del archivo.

Ruta (String): Ubicación del archivo en el sistema de almacenamiento.

Tamaño (Number): Tamaño del archivo en bytes.

Descripción (String): Información adicional o descripción del archivo.

ID Grabaciones (Number): Relación hacia la entidad Grabaciones, identificando la grabación asociada.

ID_Video (Number): Relación hacia la entidad Audios, indicando el audio

vinculado.

Pantalla

Descripción: Define las dimensiones y posiciones de un área de pantalla seleccionada para grabar o visualizar.

Atributos:

- ID (Number): Identificador único de la pantalla.
- Ancho (Number): Ancho de la pantalla en píxeles.
- Alto (Number): Altura de la pantalla en píxeles.
- Ini_X (Number): Coordenada X inicial del área de pantalla.
- Ini_Y (Number): Coordenada Y inicial del área de pantalla.
- Fin_X (Number): Coordenada X final del área de pantalla.
- Fin_Y (Number): Coordenada Y final del área de pantalla.

Audios

Descripción: Contiene detalles de un archivo de audio, tales como formato, duración y timestamps específicos que permiten identificar momentos clave dentro del audio.

Atributos:

- ID (Number): Identificador único del audio.
- Formato (String): Formato del archivo de audio (por ejemplo, MP3, WAV).
- Duración (Number): Duración del audio en segundos.
- Timestamp Inicio (Number): Marca de tiempo de inicio en segundos.
- Timestamp Pausa (Number): Marca de tiempo en la que se pausó el audio.
- Timestamp Finalización (Number): Marca de tiempo de finalización del audio.
- Palabras Claves (String): Palabras clave asociadas con el audio para facilitar su búsqueda.
- ID Grabaciones (Number): Relación hacia la entidad Grabaciones, indicando la grabación a la que pertenece el audio.

Grabaciones

Descripción: Esta entidad almacena información detallada de cada grabación, incluyendo formato, duración, área de grabación, y datos de timestamp. También identifica la pantalla asociada a la grabación.

Atributos:

- ID (Number): Identificador único de la grabación.
- Formato (String): Formato del archivo de grabación (por ejemplo, MP4, AVI).

Duracion (Number): Duración de la grabación en segundos.

Area (Number): Área de la pantalla grabada en píxeles.

Timestamp_inicio (Number): Marca de tiempo de inicio de la grabación.

Timestamp_pausa (Number): Marca de tiempo de pausa de la grabación.

Timestamp_finalizacion (Number): Marca de tiempo de finalización de la grabación.

Tipo (String): Tipo de grabación (por ejemplo, pantalla completa, área específica).

Estado (String): Estado actual de la grabación (por ejemplo, activo, finalizado).

Palabras_claves (String): Palabras clave para facilitar la búsqueda de la grabación.

Fecha (Datetime): Fecha y hora en que se realizó la grabación.

ID_Pantalla (Number): Relación hacia la entidad Pantalla, indicando la pantalla asociada.

Relaciones

Relación entre Archivo y Grabaciones:

Descripción: Un archivo puede estar asociado a una grabación específica, lo que permite que un archivo almacene los datos de una grabación realizada.

Cardinalidad: Un archivo puede estar relacionado con una sola grabación (uno a uno).

Relación entre Archivo y Audios:

Descripción: Un archivo puede tener vinculado un audio, lo que facilita la organización y almacenamiento de los archivos de audio de forma individual.

Cardinalidad: Un archivo puede estar relacionado con un solo audio (uno a uno).

Relación entre Grabaciones y Pantalla:

Descripción: Una grabación puede estar asociada a un área de pantalla específica que se desea grabar.

Cardinalidad: Una grabación está vinculada a una sola área de pantalla, pero una pantalla puede ser usada en múltiples grabaciones (uno a muchos).

Relación entre Grabaciones y Audios:

Descripción: Una grabación puede contener un archivo de audio asociado que se sincroniza con el contenido grabado.

Cardinalidad: Una grabación puede estar vinculada a un solo archivo de audio, pero un audio puede ser utilizado en múltiples grabaciones

Integridad Referencial entre "Archivo" y "Grabaciones": Cada registro en la tabla Archivo debe estar asociado con una grabación existente en la

tabla Grabaciones a través de la clave foránea ID_Grabaciones.

Integridad Referencial entre "Archivo" y "Audios": Cada registro en la tabla Archivo debe estar asociado con un audio existente en la tabla Audios a través de la clave foránea ID_Audio.

Integridad Referencial entre "Audios" y "Grabaciones": Cada registro en la tabla Audios debe estar asociado con una grabación existente en la tabla Grabaciones a través de la clave foránea ID_Grabaciones.

Integridad Referencial entre "Grabaciones" y "Pantalla": Cada registro en la tabla Grabaciones debe estar asociado con una pantalla existente en la tabla Pantalla a través de la clave foránea ID_Pantalla.

Coleccion No SQL

```
{
  "_id": ObjectId,
  "nombre": "string",
  "ruta": "string",
  "tamaño": Number,
  "descripcion": "string",
  "grabacion": {
    "id_grabacion": ObjectId,
    "formato": "string",
    "duracion": Number
  },
  "audio": {
    "id_audio": ObjectId,
    "formato": "string",
    "duracion": Number
  }
}
```

```
{
  "_id": ObjectId,
  "ancho": Number,
  "alto": Number,
  "coordenadas_iniciales": {
    "x": Number,
```

```
    "y": Number
  },
  "coordenadas_finales": {
    "x": Number,
    "y": Number
  }
}
```

```
{
  "_id": ObjectId,
  "formato": "string",
  "duracion": Number,
  "timestamp": {
    "inicio": Number,
    "pausa": Number,
    "finalizacion": Number
  },
  "palabras_claves": ["string"],
  "grabacion_id": ObjectId
}
```

```
{
  "_id": ObjectId,
  "formato": "string",
  "duracion": Number,
  "area": Number,
  "timestamp": {
    "inicio": Number,
    "pausa": Number,
    "finalizacion": Number
  },
  "tipo": "string",
  "estado": "string",
  "palabras_claves": ["string"],
  "fecha": ISODate,
  "pantalla_id": ObjectId
}
```

Etapa 2: Persistencia de Datos con Backend

Introducción:

En esta etapa del proyecto "**Video Recording**", se establecerá la infraestructura de persistencia de datos en el backend. Este componente es esencial para almacenar y gestionar de manera estructurada las grabaciones y la información asociada, permitiendo que los datos puedan guardarse, recuperarse y organizarse de forma eficaz. Esto asegura una integración robusta entre el componente de grabación de pantalla y el backend.

Durante el semestre anterior, se trabajó en la documentación inicial del proyecto, y se espera que el desarrollo del frontend sea abordado en el próximo semestre, permitiendo una experiencia completa de gestión de grabaciones para los usuarios finales

Propósito de la Etapa:

El objetivo de esta etapa es garantizar la persistencia y administración confiable de los datos generados por el sistema. Esto incluye el almacenamiento de las grabaciones y otros datos necesarios para su administración, facilitando la integración entre el frontend y el backend y permitiendo que los usuarios accedan y gestionen sus archivos de manera eficiente.

Alcance de la Etapa:

En la Etapa 2 se diseñará la base de datos para almacenar y organizar las grabaciones de manera estructurada y eficiente, implementando una API para permitir la futura comunicación entre el frontend y el backend, lo cual se desarrollará en una etapa posterior. Además, se gestionarán los metadatos necesarios para facilitar la administración de los archivos, junto con la persistencia de las opciones de edición, como cortes y eliminaciones de segmentos. Opcionalmente, se incluirá un control de acceso y permisos para definir quién puede visualizar o modificar los archivos de grabación.

Definiciones y Acrónimos

Definiciones

1. **MongoDB:** Sistema de gestión de bases de datos NoSQL, orientado a documentos, que utiliza un formato de almacenamiento basado en BSON (Binary JSON). MongoDB es conocido por su escalabilidad, flexibilidad y

- facilidad para manejar grandes volúmenes de datos no estructurados.
2. **Mongo Atlas:** Servicio administrado de bases de datos MongoDB en la nube, proporcionado por MongoDB Inc. Permite a los desarrolladores implementar, gestionar y escalar bases de datos MongoDB sin tener que preocuparse por la infraestructura o el mantenimiento.
 3. **Node.js:** Entorno de ejecución para JavaScript que permite ejecutar código JavaScript en el servidor. Basado en el motor V8 de Google Chrome, es conocido por su alta eficiencia y su modelo de programación asíncrona y orientado a eventos.
 4. **Backend:** Parte del desarrollo de una aplicación que se encarga de la lógica de negocio, almacenamiento de datos, y la comunicación con otras aplicaciones o servicios. Se ejecuta en un servidor y no es accesible directamente por los usuarios finales, pero es esencial para el funcionamiento de la aplicación.
 5. **Capa de datos:** Capa en una arquitectura de software que se encarga de gestionar el acceso a la base de datos y de garantizar que los datos se almacenen, recuperen y manipulen de forma eficiente. A menudo, se maneja mediante consultas y comandos SQL (en bases de datos relacionales) o comandos específicos de bases de datos NoSQL.
 6. **Service:** Componente de software que contiene la lógica de negocio y ofrece funcionalidades específicas a otras partes del sistema, como la interacción con la base de datos, el procesamiento de datos, o la ejecución de tareas relacionadas. Los servicios en una arquitectura de software suelen estar desacoplados de la interfaz de usuario.
 7. **Controller:** Componente dentro de un patrón de diseño MVC (Modelo-Vista-Controlador) que maneja las solicitudes de los usuarios, coordina la ejecución de las operaciones del servicio y gestiona la respuesta que se envía al usuario. Los controladores gestionan la interacción entre la capa de presentación y la capa de datos.
 8. **Repository:** Patrón de diseño que proporciona una capa de abstracción entre la capa de acceso a los datos y el resto de la aplicación. Los repositorios gestionan la lógica de consulta y persistencia de datos, y permiten realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de forma más estructurada.

Acrónimos

1. **MongoDB:** NoSQL DB, orientado a documentos, basado en BSON (Binary JSON).
2. **API (Application Programming Interface):** Conjunto de funciones y protocolos que permiten que las aplicaciones se comuniquen entre sí.
3. **CRUD (Create, Read, Update, Delete):** Operaciones básicas realizadas en bases de datos para gestionar los datos.
4. **MVC (Model-View-Controller):** Patrón de diseño de software que separa la lógica de negocio, la interfaz de usuario y el control de la aplicación en tres componentes distintos.
5. **SQL (Structured Query Language):** Lenguaje estándar para gestionar y manipular bases de datos relacionales.
6. **NoSQL (Not Only SQL):** Tipo de bases de datos que no utilizan un modelo

relacional. MongoDB es un ejemplo de base de datos NoSQL.

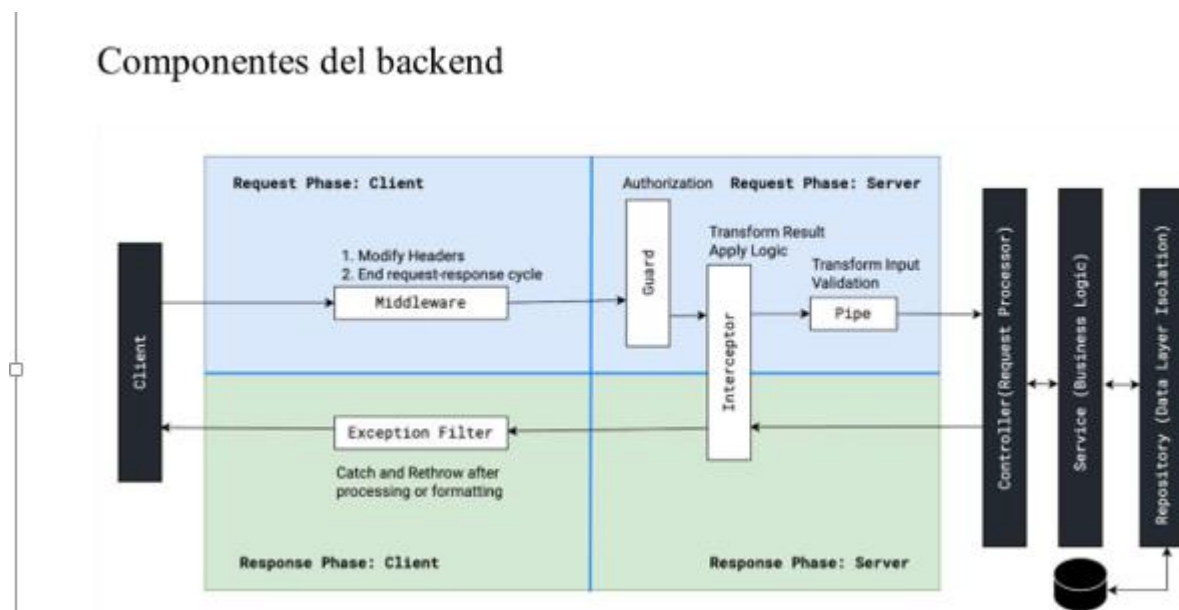
7. **JSON (JavaScript Object Notation):** Formato de intercambio de datos basado en texto que es fácil de leer y escribir para los humanos, y fácil de analizar y generar por las máquinas. Se usa comúnmente para el intercambio de datos entre clientes y servidores.

Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta:

La arquitectura backend del componente "Video Recording" utiliza **Nest.js** en **Node.js**, organizada en módulos para manejar la grabación, almacenamiento y recuperación de videos. Incluye un controlador para gestionar las solicitudes, un servicio para la lógica de grabación, y una base de datos para almacenar metadatos de las grabaciones. También cuenta con un sistema de autenticación para asegurar el acceso y, opcionalmente, una cola de procesamiento para tareas intensivas. Todo esto está expuesto mediante una API REST que permite la integración con otros servicios o un frontend, garantizando un sistema seguro y escalable.

Componentes del Backend

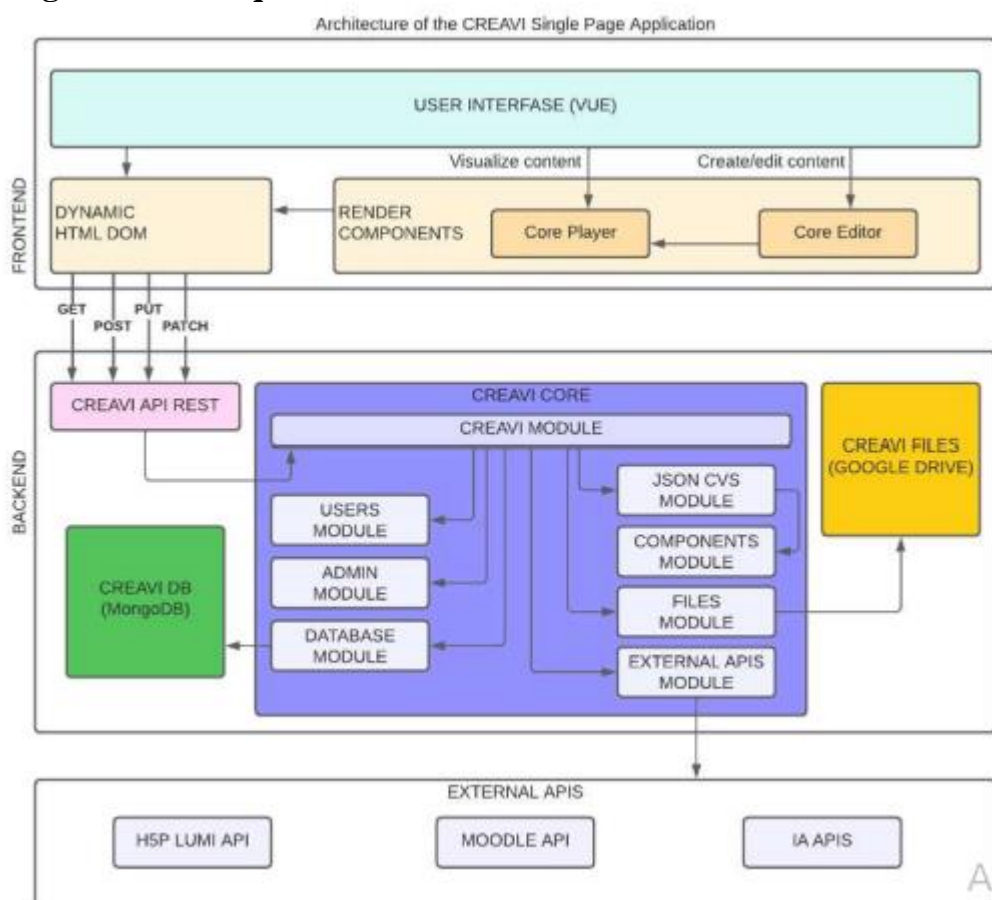


En nuestro proyecto de backend de grabación de video, seguimos una estructura similar al diagrama ejemplo proporcionado. Cuando un cliente envía una solicitud, primero

pasa por un **Middleware**, que se encarga de modificar los encabezados y gestionar el ciclo de solicitud-respuesta. Luego, un **Guard** valida la autorización de la solicitud, asegurándonos de que el usuario tenga los permisos necesarios. Una vez autorizada, la solicitud pasa por un **Pipe** que transforma y valida los datos de entrada, garantizando que cumplan con los formatos requeridos. Después, un **Interceptor** gestiona la transformación de los resultados y aplica la lógica necesaria antes de llegar al controlador. Si se presenta un error, un **Filtro de Excepciones** lo captura y lo reformatea, enviando una respuesta coherente al cliente.

En la fase de procesamiento, los **Controladores** manejan las solicitudes y las dirigen a los **Servicios**, donde implementamos la lógica de negocio que procesa y transforma los datos de la grabación. Los **Repositorios** se encargan de la interacción con la base de datos, lo que nos permite asegurar el aislamiento de los datos y la persistencia de la información. Este enfoque nos permite manejar autorizaciones, validaciones y la gestión de errores de manera estructurada, asegurando que nuestro backend sea eficiente y seguro para las funcionalidades de grabación de video.

Diagramas de Arquitectura



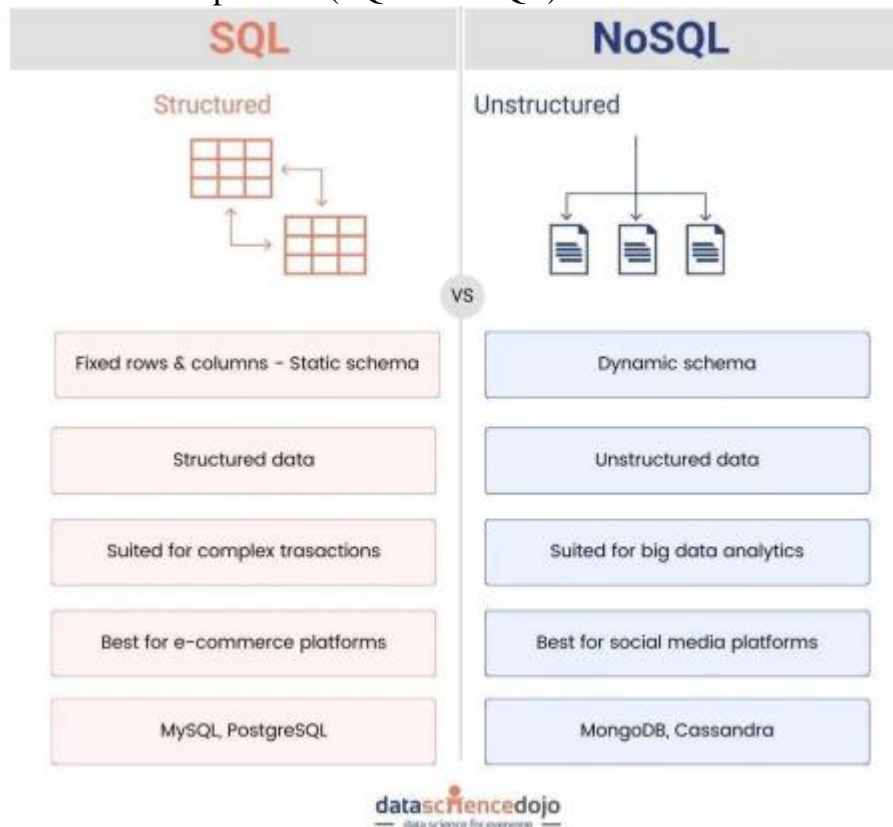
El diagrama anterior muestra la arquitectura de la aplicación de una sola página (SPA) de CREAVI, compuesta por una interfaz de usuario que incluye componentes para la visualización y edición de contenido, un backend que maneja la lógica de la aplicación mediante módulos, y una base de datos MongoDB. En esta arquitectura, se

integrará el componente "Video Recording ", un grabador de pantalla que será muy útil para satisfacer las necesidades de los usuarios dentro de la plataforma grabando la pantalla completamente o por secciones, con audio o sin audio según la necesidad

En el frontend, se incorporarán componentes para la captura de video y retroalimentación en los módulos de visualización y edición de las grabaciones. En el backend, se añadirá un nuevo módulo dentro del núcleo de la plataforma, con endpoints en la API REST para manejar solicitudes de comandos relacionados con la grabación de video, lógica para interpretar y ejecutar estos comandos (como iniciar, pausar o detener la grabación), y generación de respuestas visuales o de estado. Esto mejorará significativamente la accesibilidad y usabilidad de la plataforma, permitiendo a los usuarios crear, editar y gestionar grabaciones de pantalla de manera eficiente.

Elección de la Base de Datos

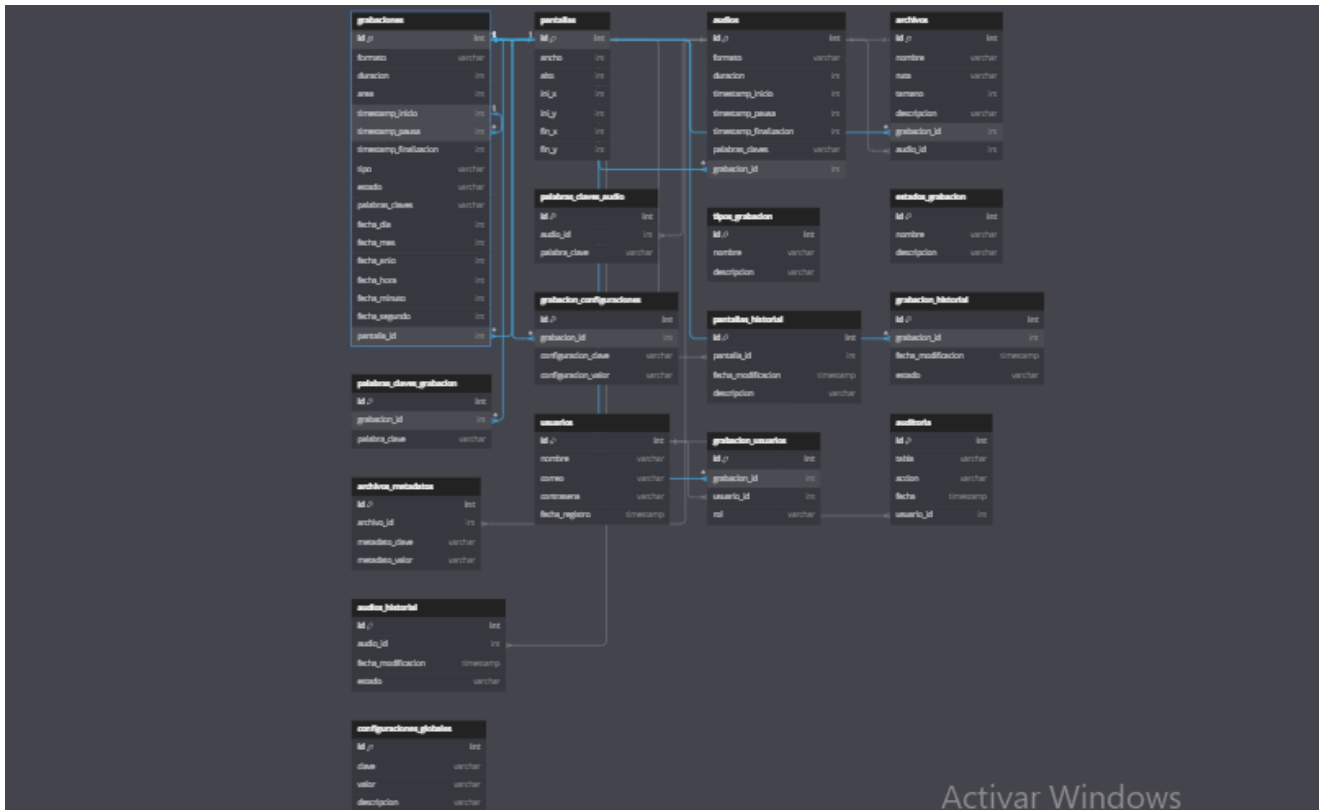
Evaluación de Opciones (SQL o NoSQL)



Justificación de la Elección

En el proyecto de grabador de pantalla, una base de datos NoSQL es la opción más adecuada debido a la flexibilidad y escalabilidad que ofrece. Al manejar grandes volúmenes de datos no estructurados, como los metadatos de las grabaciones y los registros de actividad, NoSQL permite un almacenamiento más eficiente y rápido. Además, su capacidad para escalar horizontalmente facilita el manejo de crecientes cantidades de datos a medida que el sistema crece, sin necesidad de reestructurar la base de datos. Esto optimiza tanto el rendimiento como la disponibilidad de la plataforma.

Diseño de esquema de bases de datos



Las colecciones del componente **Video Recording** en la plataforma **Creavi** gestionan diferentes aspectos de las grabaciones de pantalla. La colección **Grabación** almacena información sobre cada grabación, incluyendo detalles como formato, duración, área grabada, timestamps de inicio, pausa y finalización, tipo, estado, palabras clave y la fecha de grabación. La colección **Pantalla** define las propiedades de las pantallas o áreas grabadas, como dimensiones y coordenadas. **Audio** contiene los datos de los archivos de audio asociados, como formato, duración y timestamps, junto con las palabras clave. Por último, la colección **Archivo** gestiona los archivos generados durante la grabación, incluyendo información sobre el nombre, ruta, tamaño, descripción y referencias a las grabaciones y audios correspondientes. Juntas, estas colecciones permiten un control completo sobre las grabaciones, su almacenamiento y gestión dentro de la plataforma Creavi.

Implementación del Backend

Elección del Lenguaje de Programación

El componente **Video Recording** se implementa utilizando **Node.js** y **NestJS**, un framework modular y escalable para crear aplicaciones backend. **Node.js** proporciona un entorno de ejecución eficiente para manejar múltiples solicitudes concurrentes, mientras que **NestJS** facilita la estructura del proyecto y la gestión de dependencias, permitiendo un desarrollo más organizado y mantenible. En este caso, se utilizan las colecciones de **MongoDB** para almacenar información sobre las grabaciones de pantalla, los archivos multimedia asociados (video y audio), y las pantallas grabadas. **Mongoose** se utiliza para interactuar con la base de datos, y los esquemas están definidos en NestJS para reflejar las colecciones como **Grabación**, **Audio**, **Archivo** y **Pantalla**. La arquitectura se basa en servicios que gestionan la lógica de negocio, y controladores que manejan las solicitudes HTTP para interactuar con los datos. Esto permite un manejo eficiente de las grabaciones de video y audio, así como un almacenamiento organizado de los archivos generados durante el proceso de grabación.

Creación de la Lógica de Negocio

Estructura del Servicio en NestJS

Un servicio en NestJS es responsable de la lógica de negocio, gestionando las operaciones con las bases de datos y exponiendo métodos que pueden ser llamados por los controladores. Para el componente de **Video Recording**, el servicio manejará la creación de grabaciones, la vinculación de audios y archivos, la gestión de pantallas grabadas, y más. Este servicio es el lugar donde se implementa la lógica de negocio principal, como la creación de grabaciones, la adición de audios y archivos relacionados, y la gestión de las pantallas grabadas.

```
import { Injectable } from '@nestjs/common';
import { InjectModel } from '@nestjs/mongoose';
import { Model } from 'mongoose';
import { Grabacion } from '../schemas/grabacion.schema';
import { Audio } from '../schemas/audio.schema';
import { Archivo } from '../schemas/archivo.schema';
import { Pantalla } from '../schemas/pantalla.schema';

@Injectable()
export class VideoRecordingService {
  constructor(
    @InjectModel(Grabacion.name) private grabacionModel: Model<Grabacion>,
    @InjectModel(Audio.name) private audioModel: Model<Audio>,
    @InjectModel(Archivo.name) private archivoModel: Model<Archivo>,
    @InjectModel(Pantalla.name) private pantallaModel: Model<Pantalla>,
  ) {}

  // Crear una nueva grabación
  async createGrabacion(grabacionDto): Promise<Grabacion> {
    const newGrabacion = new this.grabacionModel(grabacionDto);
    return newGrabacion.save();
  }
```

```
const newAudio = new this.audioModel(audioDto);
return newAudio.save();
}

// Crear un archivo asociado a una grabación y audio
async createArchivo(archivoDto): Promise<Archivo> {
  const newArchivo = new this.archivoModel(archivoDto);
  return newArchivo.save();
}

// Obtener el audio de una grabación específica
async getAudioByGrabacionId(grabacionId: number): Promise<Audio[]> {
  return this.audioModel.find({ grabacionId }).exec();
}

// Obtener todos los archivos asociados a una grabación
async getArchivosByGrabacionId(grabacionId: number): Promise<Archivo[]> {
  return this.archivoModel.find({ grabacionId }).exec();
}

// Obtener la pantalla grabada por su ID
async getPantallaById(id: number): Promise<Pantalla> {
  return this.pantallaModel.findOne({ id }).exec();
}

// Crear una nueva pantalla grabada
async createPantalla(pantallaDto): Promise<Pantalla> {
  const newPantalla = new this.pantallaModel(pantallaDto);
  return newPantalla.save();
}
```

El **VideoRecording Controller** interactúa con el servicio para exponer estas funcionalidades a través de una API RESTful.

```
import { Controller, Post, Body, Get, Param } from '@nestjs/common';
import { VideoRecordingService } from '../video-recording.service';
import { Grabacion } from '../schemas/grabacion.schema';

@Controller('video-recording')
export class VideoRecordingController {
  constructor(private readonly videoRecordingService: VideoRecordingService) {}

  @Post('grabacion')
  createGrabacion(@Body() grabacionDto: Promise<Grabacion>): Promise<Grabacion> {
    return this.videoRecordingService.createGrabacion(grabacionDto);
  }

  @Get('grabacion/:id')
  getGrabacionById(@Param('id') id: number): Promise<Grabacion> {
    return this.videoRecordingService.getGrabacionById(id);
  }

  @Post('audio')
  createAudio(@Body() audioDto: Promise<any>): Promise<any> {
    return this.videoRecordingService.createAudio(audioDto);
  }

  @Post('archivo')
  createArchivo(@Body() archivoDto: Promise<any>): Promise<any> {
    return this.videoRecordingService.createArchivo(archivoDto);
  }

  @Get('pantalla/:id')
  getPantallaById(@Param('id') id: number): Promise<any> {
    return this.videoRecordingService.getPantallaById(id);
  }
}
```

Desarrollo de Endpoints

En nuestro proyecto "Video Recording", los endpoints son fundamentales para la gestión de los recursos asociados con la grabación de video, audio, pantallas y archivos. Estos endpoints proporcionan las funcionalidades necesarias para interactuar con los datos de la plataforma de manera eficiente. Se implementan operaciones CRUD (Crear, Leer, Actualizar, Eliminar) a través de los métodos HTTP estándar.

Por ejemplo, en la colección de **Archivos**, el endpoint GET /archivos permite obtener todos los archivos, mientras que POST /archivos se usa para cargar un nuevo archivo en el sistema. Para manipular archivos específicos, se usan rutas dinámicas como GET /archivos/{id}, que recupera la información de un archivo con un ID específico, o PATCH /archivos/{id}, para actualizar los detalles de un archivo en particular. Además, DELETE /archivos/{id} permite eliminar un archivo por su ID.

En el caso de **Audios**, el endpoint GET /audios se utiliza para listar todos los audios disponibles, y POST /audios se usa para registrar un nuevo archivo de audio. Para manejar audios específicos, GET /audios/{id}, PATCH /audios/{id} y DELETE /audios/{id} permiten obtener, modificar o eliminar audios según el ID proporcionado.

Para la gestión de **Pantallas**, los endpoints permiten obtener y crear pantallas con GET /pantallas y POST /pantallas. Para manejar pantallas específicas, se implementan los métodos GET /pantallas/{id}, PATCH /pantallas/{id} y DELETE /pantallas/{id}.

Finalmente, en la colección de **Grabaciones**, los endpoints permiten interactuar con las grabaciones de video, ofreciendo funcionalidades similares a las otras colecciones, con los métodos GET /grabaciones, POST /grabaciones, PATCH /grabaciones/{id} y DELETE /grabaciones/{id}.

Todos estos endpoints están diseñados para interactuar con la base de datos de manera eficiente, y las respuestas devueltas permiten validar las operaciones realizadas, mientras que herramientas como Postman facilitan el proceso de prueba y documentación de la API.

Recurso	Método	Descripción	Ejemplo de Endpoint
Archivos	GET	Obtener todos los archivos registrados.	/archivos
	POST	Crear un nuevo archivo.	/archivos
	GET	Obtener un archivo específico por ID.	/archivos/{id}
	PATCH	Actualizar un archivo específico por ID.	/archivos/{id}
	DELETE	Eliminar un archivo específico por ID.	/archivos/{id}
Audios	GET	Obtener todos los audios registrados.	/audios
	POST	Crear un nuevo archivo de audio.	/audios
	GET	Obtener un audio específico por ID.	/audios/{id}
	PATCH	Actualizar un archivo de audio específico por ID.	/audios/{id}
	DELETE	Eliminar un archivo de audio específico por ID.	/audios/{id}
Pantallas	GET	Obtener todas las pantallas registradas.	/pantallas
	POST	Crear una nueva pantalla.	/pantallas
	GET	Obtener una pantalla específica por ID.	/pantallas/{id}
	PATCH	Actualizar una pantalla específica por ID.	/pantallas/{id}
	DELETE	Eliminar una pantalla específica por ID.	/pantallas/{id}
Grabaciones	GET	Obtener todas las grabaciones registradas.	/grabaciones
	POST	Crear una nueva grabación.	/grabaciones
	GET	Obtener una grabación específica por ID.	/grabaciones/{id}
	PATCH	Actualizar una grabación específica por ID.	/grabaciones/{id}
	DELETE	Eliminar una grabación específica por ID.	/grabaciones/{id}

Conexión a la Base de Datos

En el contexto de tu proyecto "Video Recording", la conexión a la base de datos se realiza a través de NestJS y TypeORM, Mongo DB es la base de datos utilizada. El archivo app.module.ts configura la conexión, utilizando el módulo TypeOrmModule para establecer la conexión con la base de datos y realizar las operaciones necesarias para manejar las colecciones de Grabación, Audio, Archivo y Pantalla.

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { ArchivosModule } from './archivos/archivos.module';
import { AudiosModule } from './audios/audios.module';
import { PantallasModule } from './pantallas/pantallas.module';
import { GrabacionesModule } from './grabaciones/grabaciones.module';
import { MongooseModule } from '@nestjs/mongoose';

@Module({
  imports: [
    MongooseModule.forRoot(
      'mongodb+srv://cbricenoojeda42:GWAPA0ZTviZLWqyg@cluster0.avxwc.mongodb.net/db_video_recording?retryWrites=true&w=majority&appName=Cluster0',
    ),
    ArchivosModule,
    AudiosModule,
    PantallasModule,
    GrabacionesModule,
  ],
  controllers: [AppController],
  providers: [AppService],
})
export class AppModule {}
```

Desarrollo de Operaciones CRUD

El desarrollo de operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en NestJS Y mongo DB se realiza principalmente a través de **servicios** y **controladores**. A continuación, te explico cómo desarrollar cada una de las operaciones CRUD en el contexto de tu proyecto de **Video Recording** para entidades como **Grabaciones**, **Audios**, **Pantallas** y **Archivos**.

```
Import { Injectable } from '@nestjs/common';
Import { InjectRepository } from '@nestjs/typeorm';
Import { Repository } from 'typeorm';
Import { Grabacion } from './grabacion.entity';
```

```
@Injectable()
export class GrabacionService {
  constructor(
    @InjectRepository(Grabacion) private grabacionRepository:
    Repository<Grabacion>,
```

```
) {}
```

```
// Crear una nueva grabación
```

```
async create(grabacionData: Partial<Grabacion>): Promise<Grabacion> {  
  const newGrabacion = this.grabacionRepository.create(grabacionData);  
  return await this.grabacionRepository.save(newGrabacion);  
}
```

```
// Obtener todas las grabaciones
```

```
async findAll(): Promise<Grabacion[]> {  
  return await this.grabacionRepository.find();  
}
```

```
// Obtener una grabación por su ID
```

```
async findOne(id: number): Promise<Grabacion> {  
  return await this.grabacionRepository.findOne(id);  
}
```

```
// Actualizar una grabación
```

```
async update(id: number, grabacionData: Partial<Grabacion>):  
Promise<Grabacion> {  
  await this.grabacionRepository.update(id, grabacionData);  
  return this.findOne(id);  
}
```

```
// Eliminar una grabación
```

```
async remove(id: number): Promise<void> {  
  await this.grabacionRepository.delete(id);  
}  
}
```

Manejo de Transacciones

Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapa 3: Consumo de Datos y Desarrollo Frontend

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend

