| Caso de Uso | Codificación de Audio |
| --- | --- |
| ID | RF-1 |
| Descripción | El sistema debe transformar la codificación del audio para que sea reconocible y procesable por otros módulos. |
| Actores | Servidor, FFMPEG |
| Flujo principal | 1. El usuario inicia el servicio de reconocimiento de voz.  2. El sistema recibe el audio y lo envía al servidor para codificación.  3. FFMPEG transforma el audio a un formato adecuado para procesamiento.  4. El sistema confirma la correcta codificación y sigue al módulo de interpretación. |
| Precondiciones | El audio debe estar grabado y accesible. |
| Postcondiciones | El audio se codifica en un formato compatible. |

| Caso de Uso | Codificación de Video |
| --- | --- |
| ID | RF-2 |
| Descripción | El sistema debe transformar la codificación de video para procesamiento de lengua de señas. |
| Actores | Servidor, FFMPEG |
| Flujo principal | 1. El usuario envía un video que contiene lengua de señas.  2. El sistema envía el video al servidor.  3. FFMPEG convierte el video a un formato compatible con el módulo de detección de señas.  4. El sistema confirma que el vídeo está codificado y listo para detección. |
| Precondiciones | El video debe estar grabado en un formato reconocible |
| Postcondiciones | El video se convierte para su análisis de señas. |

| Caso de Uso | Diseño de la arquitectura de Red Neuronal |
| --- | --- |
| ID | RF-3 |
| Descripción | Diseñar una arquitectura de red neuronal para el procesamiento de video. |
| Actores | Ariel, Jennifer, Mattias |
| Flujo principal | 1. Se diseña la arquitectura de la Red Neuronal Convolucional.  2. Se entrena la CNN para que detecte las señas.  3. Se implementa la CNN dentro del sistema. |
| Precondiciones | Definición de los requisitos de precisión y rendimiento. |
| Postcondiciones | La red neuronal está lista para procesar señales. |

| Caso de Uso | Transcripción de Audios |
| --- | --- |
| ID | RF-4 |
| Descripción | Convertir audio en texto para facilitar su procesamiento en el sistema. |
| Actores | Vosk Model |
| Flujo principal | 1. El usuario graba o carga un archivo de audio.  2. El sistema envía el audio al modelo de Vosk.  3. El modelo convierte el audio en texto.  4. El sistema almacena el texto para su posterior uso o interpretación. |
| Precondiciones | El audio debe estar en un formato compatible. |
| Postcondiciones | El texto transcrito se almacena en el sistema. |

| Caso de Uso | Generación de Querys |
| --- | --- |
| ID | RF-5 |
| Descripción | Crear consultas en la base de datos para manejar la información procesada. |
| Actores | Pymongo, MongoDB, FastAPI |
| Flujo principal | 1. El usuario envía una solicitud para consultar datos específicos.  2. FastAPI interpreta y genera la query necesaria.  3. Pymongo envía la query a la base de datos MongoDB.  4. MongoDB devuelve los resultados a la aplicación. |
| Precondiciones | Los datos deben estar almacenados en MongoDB. |
| Postcondiciones | Los datos deben estar almacenados en MongoDB. |

| Caso de Uso | Plataforma de Carga de la Base de Datos |
| --- | --- |
| ID | RF-6 |
| Descripción | Proporcionar una plataforma para cargar datos (palabras e imágenes) en la base de datos. |
| Actores | Pymongo, FastAPI |
| Flujo principal | 1. El administrador de la base de datos accede a la plataforma de carga.  2. Selecciona archivos de palabras o imágenes para cargar.  3. FastAPI procesa y envía los datos a MongoDB a través de Pymongo.  4. MongoDB almacena los datos. |
| Precondiciones | Acc4so a la plataforma de carga. |
| Postcondiciones | Los datos están almacenados en MongoDB y accesibles. |

| Caso de Uso | Creación de Base de Datos |
| --- | --- |
| ID | RF-7 |
| Descripción | Configurar y crear una base de datos no relacional en MongoDB para almacenar y gestionar datos de audio, video y texto procesados por el sistema. |
| Actores | MongoDB |
| Flujo principal | 1. Se crea un Cluster dentro de Mongo Atlas  2. Se crea una Base de Datos dentro del Cluster.  3. Se exportan las credenciales de la Base de Datos para su futura implementación. |
| Precondiciones | Cuenta de Mongo Atlas. |
| Postcondiciones | La base de datos en MongoDB está creada, estructurada, y lista para recibir y almacenar datos en colecciones adecuadas para el proyecto. |

| Caso de Uso | Inicio de Sesión / Registro en Developer Portal |
| --- | --- |
| ID | RF-8 |
| Descripción | Integrar servicio de Firebase para la autenticación con correo, github y google. |
| Actores | Firebase, MongoDB, Developer Portal |
| Flujo principal | 1. El usuario ingresa al Developer Portal.  2. Se dirige al inicio de sesión/registro.  3. Selecciona el tipo de registro (correo, github o google).  4. Se registra con el proveedor y se crean los datos básicos del usuario (Nombre, Correo, APIKeys, etc). |
| Precondiciones | Developer Portal desarrollado y desplegado. |

| Caso de Uso | Generación de APIKeys |
| --- | --- |
| ID | RF-9 |
| Descripción | Habilitar la generación de APIKeys dentro del Developer Portal. |
| Actores | MongoDB, Developer Portal, Servidor |
| Flujo principal | 1. El usuario ingresa al Developer Portal.  2. Se dirige a su perfil.  3. Se dirige a sus APIKeys.  4. Genera una nueva APIKey. |
| Precondiciones | Developer Portal desarrollado y desplegado. |

| Caso de Uso | Revocación de APIKeys |
| --- | --- |
| ID | RF-10 |
| Descripción | Habilitar la eliminación de APIKeys dentro del Developer Portal. |
| Actores | MongoDB, Developer Portal, Servidor |
| Flujo principal | 1. El usuario ingresa al Developer Portal.  2. Se dirige a su perfil.  3. Se dirige a sus APIKeys.  4. Elimina la APIKey seleccionada. |
| Precondiciones | Developer Portal desarrollado y desplegado. |

| Caso de Uso | Documentación del Servicios |
| --- | --- |
| ID | RF-11 |
| Descripción | Habilitar la documentación del Servicio SignAI. |
| Actores | Developer Portal |
| Flujo principal | 1. El usuario ingresa al Developer Portal.  2. Se dirige a la documentación.  3. Revisa cómo integrar la solución en el entorno. |
| Precondiciones | Developer Portal desarrollado y desplegado. |