*Vicerrectoría Académica* 

*Dirección de Servicios Académicos*

*Subdirección de Servicios a Escuelas*

**[Sistema Avatar para Traducción de Voz a Lengua de Señas Chilena (LSCh) CatchAI]**

**(SAD) Software Architecture Document Versión 2.0**

**Identificación de Documento**

|  |  |
| --- | --- |
| **Identificación** | Documento de Arquitectura de Software de CatCh AI |
| **Proyecto** | Sistema Avatar para Traducción de Voz a Lengua de Señas Chilena (LSCh) CatchAI |
| **Versión** | 2.0 |

|  |  |
| --- | --- |
| **Documento mantenido por** | Eyleen Collado |
| **Fecha de última revisión** | 29/10/2025 |
| **Fecha de próxima revisión** | 29/10/2025 |

|  |  |
| --- | --- |
| **Documento aprobado por** | Giocrisrai Godoy |
| **Fecha de última aprobación** | 28/10/2025 |

**Historia de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 27/10/2025 | 1.0 | Creación al desarrollo documento “DAS”. | Eyleen Collado |
| 12/10/2025 | 2.0 | Inserción de diagramas y mockups | Eyleen Collado |

Tabla de contenido

[1. Introducción 5](#_Toc211216258)

[1.1. Contexto del Problema 5](#_Toc211216259)

[1.2. Propósito 5](#_Toc211216260)

[1.3. Ámbito 6](#_Toc211216261)

[1.4. Definiciones, acrónimos y abreviaciones 7](#_Toc211216262)

[1.5. Referencias 7](#_Toc211216263)

[1.6. Resumen ejecutivo 8](#_Toc211216264)

[1.7. Representación 8](#_Toc211216265)

[2. Metas y Restricciones de la Arquitectura 9](#_Toc211216266)

[2.1 Metas de la arquitectura 9](#_Toc211216267)

[2.2 Restricciones de la Arquitectura 9](#_Toc211216268)

[2.3 Otros antecedentes y consideraciones 10](#_Toc211216269)

[3. Vista +1 11](#_Toc211216270)

[3.1 Modelo de Casos de Uso 11](#_Toc211216271)

[3.2 Especificación de Casos de Uso Relevantes 12](#_Toc211216272)

[3.3 Especificación de los Escenarios de Calidad Relevantes 17](#_Toc211216273)

[4. Vista Lógica 22](#_Toc211216274)

[4.1. Diagrama de Clases 22](#_Toc211216275)

[4.2. Diagrama de comunicación 24](#_Toc211216276)

[4.3. Diagrama de Secuencia 26](#_Toc211216277)

[4.4. Modelo Entidad Relación (a futuro) 27](#_Toc211216278)

[5. Vista de Componentes 29](#_Toc211216279)

[5.1. Diagrama de Componentes y Conectores 29](#_Toc211216280)

[5.2. Diagrama de Paquetes 30](#_Toc211216281)

[6. Vista de Procesos 33](#_Toc211216282)

[6.1. Diagrama de Actividades 33](#_Toc211216283)

[7. Vista Física 35](#_Toc211216284)

[7.1. Diagrama de Despliegue 35](#_Toc211216285)

[8. Decisiones de Diseño y Selección de Alternativas 38](#_Toc211216286)

[8.1. Elección del Entorno de Ejecución 38](#_Toc211216287)

[8.2. Arquitectura Modular y Orquestada 38](#_Toc211216288)

[8.3. Selección de Herramientas Técnicas 39](#_Toc211216289)

[8.4. Estrategia de Comunicación entre Componentes 39](#_Toc211216290)

[8.5. Manejo de Errores y Logs 39](#_Toc211216291)

[8.6. Diseño del Flujo de Procesamiento (Pipeline) 40](#_Toc211216292)

[8.7. Consideraciones de Seguridad, Privacidad y Gobernanza de la IA 40](#_Toc211216293)

[9. Análisis de Reutilización 43](#_Toc211216294)

[9.1. Reutilización de Componentes Técnicos 43](#_Toc211216295)

[9.2. Reutilización de Recursos y Modelos 44](#_Toc211216296)

[9.3. Reutilización Arquitectónica 44](#_Toc211216297)

[10. Mockups 45](#_Toc211216298)

[11. Anexos 46](#_Toc211216299)

## Introducción

Este documento presenta la arquitectura de software del proyecto CatchAI, un sistema diseñado para traducir audio o video a Lengua de Señas Chilena (LSCh) mediante el uso de tecnologías de reconocimiento de voz y animación 3D.

El enfoque adoptado corresponde al modelo arquitectónico 4+1 de Kruchten, el cual organiza la arquitectura en diferentes vistas que permiten abordar tanto los requisitos funcionales como los no funcionales del sistema. A través de este informe se busca proporcionar una descripción clara de cómo el sistema está estructurado, qué componentes lo conforman y cómo estos interactúan entre sí.

El desarrollo de CatchAI surge de la necesidad de fomentar la inclusión de la comunidad sorda en ámbitos educativos, comunicacionales y sociales, ofreciendo un mecanismo que reduzca la dependencia de intérpretes humanos en contextos donde estos no estén disponibles. El prototipo inicial (MVP) funciona de manera local: recibe un archivo de audio o video desde una página web HTML, transcribe su contenido a texto mediante un modelo STT, consulta un diccionario de señas chilenas y reproduce automáticamente la animación de un avatar en Blender/Godot.

En esta versión del documento, se presentan las diferentes vistas arquitectónicas que conforman la solución, con un énfasis en los casos de uso más representativos, las restricciones tecnológicas del entorno local y las oportunidades de crecimiento hacia funcionalidades futuras como la exportación automática de video, respuestas en JSON y transmisión en vivo mediante protocolos estándar.

### Contexto del Problema

La inclusión de personas sordas en entornos educativos, comunicacionales y de difusión masiva sigue siendo una necesidad urgente. En Chile, la Lengua de Señas Chilena (LSCh) constituye el medio principal de comunicación para esta comunidad, pero su presencia en medios audiovisuales es escasa. Los intérpretes humanos son un recurso limitado y costoso, lo que dificulta contar con traducciones en tiempo real o en contextos de bajo presupuesto.

CatchAI busca abordar este problema mediante un prototipo que traduzca audio o video a señas chilenas, utilizando modelos de reconocimiento de voz y un avatar 3D animado en Blender/Godot. El sistema permitirá cargar archivos, grabar audio desde micrófono o ingresar video como entrada, para luego reproducir automáticamente la animación correspondiente.

### Propósito

El propósito de este documento es describir la arquitectura de software del sistema CatchAI bajo el enfoque del modelo 4+1 de Philippe Kruchten. El informe detalla cómo se organiza el sistema en sus distintas vistas (casos de uso, lógica, procesos, física e implementación), con el fin de:

* Proporcionar una visión clara de los componentes principales del sistema y sus interacciones.
* Establecer las restricciones tecnológicas que condicionan el desarrollo del MVP.
* Definir las responsabilidades de cada módulo y sus dependencias.
* Entregar lineamientos para el desarrollo incremental y la evolución hacia una solución más robusta en el futuro.

### Ámbito

El sistema CatchAI se concibe como un prototipo local orientado a la traducción de audio y video a Lengua de Señas Chilena (LSCh). Su alcance en esta primera versión (MVP) se limita a:

Entradas:

* Archivos de video en formatos estándar (MP4, MKV, AVI).
* Archivos de audio en formatos WAV, MP3 o AAC.
* Grabación de audio desde micrófono a través de una interfaz web HTML.

Procesamiento:

* Transcripción de voz a texto mediante el modelo Whisper en ejecución local.
* Consulta de un diccionario básico de señas chilenas con aproximadamente 50 frases iniciales.
* Selección de la seña correspondiente o, en caso de ausencia, uso de deletreo manual en LSCh.
* Reproducción automática de la animación del avatar en Blender/Godot.

Salidas:

* Animación en tiempo real visualizada en Blender/Godot.
* Captura opcional de la animación mediante software externo (OBS Studio) para su difusión.

En síntesis, el ámbito del MVP está centrado en validar la cadena funcional mínima (voz→texto→seña→animación) en un entorno local, garantizando que las bases tecnológicas y arquitectónicas permitan una evolución progresiva hacia versiones más completas y con mayor grado de integración.

### Definiciones, acrónimos y abreviaciones

|  |  |
| --- | --- |
| **ACRÓNIMO** | **DESCRIPCIÓN** |
| LSCh | Lengua de Señas Chilena |
| STT | Speech-to-Text, proceso de transcripción de voz a texto |
| MVP | Minimum Viable Product, primera versión funcional mínima del sistema |
| OBS | Open Broadcaster Software, software para captura de video y transmisión |
| API | Application Programming Interface, interfaz de programación de aplicaciones |
| JSON | JavaScript Object Notation, formato estándar de intercambio de datos (previsto en versiones futuras) |
| RTMP | Real-Time Messaging Protocol, protocolo de transmisión en vivo (previsto en versiones futuras) |
| WebRTC | Web Real-Time Communication, estándar de comunicación en tiempo real por navegador (previsto en versiones futuras) |

### Referencias

A continuación, se listan las referencias a otros documentos:

* Casos de Usos
* Planilla de Requerimientos
* Especificación de requerimientos de software
* Diagrama de actividad
* Diagrama de despliegue
* Documento de Especificación de Casos de Uso

Estos documentos se encuentran todos disponibles dentro del apartado de Anexos.

### Resumen ejecutivo

El proyecto CatchAI propone un sistema local de traducción de voz a señas chilenas que, en su versión inicial, recibe audio o video como entrada, transcribe el contenido mediante un modelo de reconocimiento de voz (Whisper) y reproduce automáticamente la animación de un avatar en Blender/Godot. El MVP no contempla aún la exportación automática de video ni la entrega de resultados por API, pero establece la base tecnológica para alcanzar esas funcionalidades en fases posteriores.

La arquitectura definida bajo el marco 4+1 proporciona una visión integral de los casos de uso, componentes lógicos, procesos, restricciones físicas y despliegue del sistema, asegurando que las decisiones técnicas estén alineadas con los objetivos de inclusión y accesibilidad que guían el proyecto.

### Representación

La arquitectura del sistema se adhiere al enfoque del framework 4+1 y sigue las recomendaciones del proceso unificado. En esta versión del documento, se incluyen las siguientes vistas:

* **Vista +1:** describe los casos de uso más significativos, presenta a los actores involucrados y proporciona una descripción de sus casos de uso asociados. Asimismo, aborda los escenarios de calidad más relevantes para la arquitectura.
* **Vista Física:** detalla las restricciones tecnológicas, normativas, estándares y otros factores que influyen en las decisiones arquitectónicas, tanto del producto como del proceso de desarrollo.
* **Vista Lógica:** presenta la arquitectura del sistema en diferentes niveles de refinamiento, identificando los módulos lógicos principales, sus responsabilidades y dependencias. Incluye diagramas de módulos y de componentes/conectores.
* **Vista de Procesos:** se centra en los procesos involucrados en la ejecución del sistema, detallando sus relaciones de comunicación y sincronización.
* **Vista de Implementación:** describe los componentes de deployment construidos y sus dependencias, mostrando cómo se despliega el sistema en el entorno local.

## Metas y Restricciones de la Arquitectura

A continuación, se revisan las metas y restricciones de la arquitectura.

### 2.1 Metas de la arquitectura

De acuerdo con las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

* **Desempeño:** El sistema debe proporcionar tiempos de respuesta de 2 segundos en los horarios de mayor concurrencia de uso del mismo.
* **Tolerancia a fallos:** El servidor debe estar construido y optimizado para que en las horas y fechas de mayor demanda no se exija más del 90% de uso de la CPU. Además, se deben implementar progresivamente medidas para minimizar fallas en la generación, anulación y modificación de boletas.
* **Seguridad:** El sistema debe implementar un sistema de acceso diferenciado que otorgue diferentes niveles de acceso a grupos de usuarios según las características de su perfil. También, se debe cifrar los datos y contraseñas, y cumplir con las normas WCAG para el contraste de colores. Además, se debe registrar automáticamente todas las operaciones de modificación o actualización de datos.
* **Modificabilidad/Reusabilidad:** El sistema debe contar con componentes separados y autónomos para gestionar de manera eficiente tareas específicas. Además, se deben implementar mejoras en el manejo de errores relacionados con las operaciones de boletas, lo que se traduce en una reducción constante de incidencias y fallos.
* **Operatividad:** El sistema deberá ser monitoreado constantemente para garantizar una disponibilidad y accesibilidad del 98% las 24 horas del día. Además, se debe proporcionar un manual de usuario en línea y mostrar mensajes de error de manera oportuna y clara.

### 2.2 Restricciones de la Arquitectura

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* **Tiempo de construcción**: se cuenta con un plazo estrecho de tiempo para su construcción, 4 semanas según la planificación.
* **Infraestructura**: se cuenta con servidores de aplicación replicados y con una base de datos MySQL en estructura de clúster.
* **Otros componentes de software**: no se considera la adquisición y licenciamiento de otros componentes de software más allá de los sistemas operativos.

### 2.3 Otros antecedentes y consideraciones

La empresa desarrolladora cuenta con un framework que considera los siguientes componentes que permiten satisfacer los requerimientos arquitectónicos:

* Framework de inyección de dependencias, con esto se soporta la encapsulación y modularización de componentes para facilitar la mantenibilidad del sistema. Asimismo, privilegia el performance en tiempo de ejecución dado que es un framework liviano.
* Framework de seguridad, con esto se soporta la meta de seguridad.

## Vista +1

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

### 3.1 Modelo de Casos de Uso

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.

Ilustración 1- Diagrama General de casos de uso

### 3.2 Especificación de Casos de Uso Relevantes

Los casos de uso considerados más relevantes para el desarrollo de la arquitectura fueron determinados según los siguientes criterios:

* Su implementación involucra varios componentes del sistema y requiere coordinación entre módulos de procesamiento, animación y captura.
* Su ejecución representa puntos críticos del flujo de conversión voz-a-seña, afectando directamente la calidad y comprensión de los resultados.
* Incluyen interacciones entre distintos actores (Usuario, Sistema, Administrador Técnico) y manipulación de archivos multimedia.
* Representan los escenarios más complejos y de mayor riesgo técnico dentro del MVP, debido a la dependencia de hardware y software local.

A continuación, se listan los casos de uso relevantes, los cuales pueden encontrarse con su especificación detallada en el documento “Especificación de Casos de Uso CatchAI”:

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Nombre del Caso de Uso** | **Actores Involucrados** | **Prioridad** |
| CU-01 | Cargar Archivo | Usuario | Alta |
| CU-02 | Grabar Audio | Usuario | Alta |
| CU-03 | Transcribir Audio | Sistema | Alta |
| CU-04 | Preprocesar Texto | Sistema | Media |
| CU-05 | Consultar Diccionario | Sistema | Alta |
| CU-06 | Seleccionar Seña | Sistema | Alta |
| CU-07 | Deletrear | Sistema | Media |
| CU-08 | Reproducir Animación | Sistema | Alta |
| CU-09 | Capturar Animación | Sistema | Media |
| CU-10 | Mantener Diccionario | Administrador Técnico | Media |
| CU-11 | Revisar Logs | Administrador Técnico | Media |

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 2- Diagrama de Caso de uso 1 cargar archivo

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 3- diagrama de caso de uso 2 grabar audio

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 4 - diagrama de caso de uso 3 transcribir audio

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 5- diagrama de caso de uso 4 preprocesar texto

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 6 - diagrama de caso de uso 5 consultar diccionario

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 7 - diagrama de caso de uso 6 seleccionar seña

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 8 - diagrama de caso de uso 7 deletrear

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 9 - diagrama de caso de uso 8 reproducir animación

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 10 - diagrama de caso de uso 9 capturar animación

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 11 - diagrama de caso de uso 10 mantener diccionario

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 12 - diagrama de caso de uso 11 revisar logs

### 3.3 Especificación de los Escenarios de Calidad Relevantes

A continuación, se presentan los escenarios de calidad que guían las decisiones arquitectónicas del MVP de CatchAI.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Clasificación** | **Atributo** | **Estímulo** | **Entorno** | **Artefacto** | **Respuesta esperada** | **Medida de respuesta** |
| EQ-01 | Eficiencia del desempeño | Rendimiento local | Usuario carga un archivo ≤1 min | Ejecución local estándar | Pipeline CU-01→CU-03→CU-05→CU-06/07→CU-08 | Procesar y reproducir animación sin bloqueos | ≤10 s por archivo (95% de pruebas) |
| EQ-02 | Eficiencia del desempeño | Latencia en transmisión en vivo | Flujo de audio en tiempo real | Navegador + proceso local | Captura / Transcripción / Animación | Retardo máximo controlado entre audio y seña | ≤10 s (90% de pruebas) |
| EQ-03 | Fiabilidad | Procesamiento continuo | Ejecución de 10 solicitudes seguidas | Sesión de operación local | Orquestador del pipeline | Sin caídas ni reinicios en ejecuciones consecutivas | 100% de solicitudes completadas |
| EQ-04 | Fiabilidad | Manejo de errores | Archivo inválido o interrupción en el flujo | Operación normal | Validador / pipeline | Notificar error y continuar operativo | 100% solicitudes continúan activas |
| EQ-05 | Seguridad | Protección de archivos temporales | Fin del proceso | Postprocesamiento local | Gestor de archivos | Borrado automático de archivos temporales | Limpieza ≤ 24 h |
| EQ-06 | Seguridad / Fiabilidad | Gestión de logs | Solicitud procesada | Operación normal | Módulo de registros | Registrar datos de entrada y salida cifrados | 100% de solicitudes registradas |
| EQ-07 | Compatibilidad / Portabilidad | Compatibilidad multiplataforma | Ejecución en sistemas distintos | Entorno Windows / Ubuntu | Sistema completo | Ejecución estable sin cambios de código | 100% de entornos validados |
| EQ-08 | Fiabilidad / Exactitud | Reproducibilidad de resultados | Misma entrada en equipos distintos | Ejecución local en diferentes PCs | Pipeline completo | Mismos resultados en animación y texto | 100% de pruebas reproducibles |
| EQ-09 | Exactitud funcional | Correctitud semántica | Texto dentro del vocabulario inicial (50 frases) | Diccionario activo | Módulo de consulta | Se selecciona la seña correcta | ≥95% de aciertos |
| EQ-10 | Fiabilidad / Exactitud funcional | Cobertura ante ausencia de seña | Palabra no registrada | Diccionario / Deletreo | Módulo de deletreo | Generar secuencia legible de deletreo | Comprensibilidad ≥90% |

**ID: EQ-01**

**Nombre**: Rendimiento del sistema en entorno local

**Sinopsis:** El sistema debe procesar archivos de audio o video de hasta 1 minuto y generar la animación correspondiente en menos de 10 segundos.

**Entorno**: Ejecución local estándar sin GPU, operando desde navegador web.

**Cambio en el entorno:** Se recibe un archivo de audio o video y se inicia el pipeline de procesamiento completo.

**Comportamiento esperado:** El sistema procesa la entrada y genera la animación de forma fluida sin interrupciones.

**Medida:** Procesar archivos de ≤1 min en ≤10 s, con cumplimiento en el 95% de las pruebas.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Global (RNF.01)

**ID: EQ-02**

**Nombre:** Latencia de respuesta en transmisión en vivo

**Sinopsis:** El sistema debe mantener una latencia máxima de 10 segundos entre la entrada de voz capturada en vivo y la animación generada.

**Entorno:** Captura de audio en tiempo real mediante micrófono local.

**Cambio en el entorno:** Flujo continuo de audio o transmisión en vivo.

**Comportamiento esperado**: La animación se genera en sincronía con la voz, manteniendo un retardo controlado.

**Medida:** Latencia ≤10 s entre final del audio y animación, con 90% de cumplimiento en pruebas.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Módulo de entrada de audio y motor de animación (RNF.02)

**ID: EQ-03**

**Nombre:** Procesamiento continuo sin fallas

**Sinopsis:** El sistema debe soportar la ejecución continua de solicitudes sin interrupciones ni reinicios.

**Entorno:** Sesión local de pruebas y uso operativo en entorno controlado.

**Cambio en el entorno:** Ejecución de múltiples solicitudes consecutivas.

**Comportamiento esperado:** El sistema procesa al menos 10 solicitudes seguidas sin fallos, generando animaciones válidas en todos los casos.

**Medida:** Procesamiento de 10 solicitudes consecutivas con 100% de éxito.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Global (RNF.03)

**ID: EQ-04**

**Nombre:** Manejo de errores y estabilidad del servicio

**Sinopsis:** El sistema debe mantener su estabilidad ante la carga de archivos no válidos o errores durante el proceso.

**Entorno:** Flujo normal de ejecución del sistema.

**Cambio en el entorno:** Carga de archivo inválido, corrupto o no soportado.

**Comportamiento esperado:** El sistema notifica el error al usuario, conserva la estabilidad y permite continuar con nuevas solicitudes.

**Medida:** 100% de las solicitudes permanecen activas después del error.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Módulo de validación y entrada (RNF.04)

**ID: EQ-05**

**Nombre:** Protección y eliminación de archivos temporales

**Sinopsis:** Los archivos generados durante el procesamiento deben protegerse y eliminarse automáticamente dentro de las 24 horas posteriores.

**Entorno:** Ejecución local con almacenamiento temporal en disco.

**Cambio en el entorno:** Finalización del proceso o sesión.

**Comportamiento esperado:** Los archivos temporales quedan protegidos y son eliminados en el plazo establecido.

**Medida:** Cumplimiento total en inspecciones de limpieza y seguridad de archivos.

**Prioridad Arquitectónica:** Media

**Aplicación:** Módulo de gestión de archivos (RNF.05)

**ID: EQ-06**

**Nombre:** Registro cifrado de logs del sistema

**Sinopsis:** El sistema debe registrar automáticamente los datos de entrada y salida de cada solicitud en logs cifrados.

**Entorno:** Operación normal en entorno local.

**Cambio en el entorno:** Ejecución de solicitudes o procesos consecutivos.

**Comportamiento esperado:** Los logs se generan automáticamente y se almacenan con cifrado para auditoría.

**Medida:** 100% de las solicitudes generan un registro cifrado.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Global (RNF.06)

**ID: EQ-07**

**Nombre:** Compatibilidad multiplataforma

**Sinopsis:** El sistema debe poder ejecutarse correctamente en Windows 10/11 y Ubuntu 22.04 sin requerir modificaciones de código.

**Entorno:** Instalación y ejecución en distintos sistemas operativos.

**Cambio en el entorno:** Ejecución del sistema en plataformas diferentes.

**Comportamiento esperado:** El sistema mantiene la funcionalidad completa en todos los entornos validados.

**Medida:** 100% de las pruebas exitosas en ambos sistemas operativos.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Global (RNF.11)

**ID: EQ-08**

**Nombre:** Reproducibilidad de resultados

**Sinopsis:** El sistema debe generar resultados idénticos frente a la misma entrada en diferentes equipos o sesiones.

**Entorno:** Ejecución local en equipos distintos con configuración equivalente.

**Cambio en el entorno:** Procesamiento del mismo archivo en entornos diferentes.

**Comportamiento esperado:** Los resultados son idénticos en texto y animación, sin variaciones perceptibles.

**Medida:** 100% de pruebas reproducen salidas iguales ante mismas entradas.

**Prioridad Arquitectónica**: Media

**Aplicación:** Global (RNF.12)

**ID: EQ-09**

**Nombre:** Correctitud semántica de señas reconocidas

**Sinopsis:** Las señas seleccionadas deben corresponder correctamente al texto reconocido en el diccionario LSCh.

**Entorno:** Operación normal con vocabulario inicial de 50 frases.

**Cambio en el entorno:** Procesamiento de texto reconocido dentro del vocabulario.

**Comportamiento esperado:** El sistema selecciona la seña adecuada y la reproduce correctamente.

**Medida:** Precisión ≥95% en pruebas internas.

**Prioridad Arquitectónica:** Alta

**Aplicación:** Módulo de diccionario y selección de señas

**ID: EQ-10**

**Nombre:** Cobertura ante ausencia de señas

**Sinopsis:** Cuando el texto no tiene seña registrada, el sistema debe generar una secuencia de deletreo legible y comprensible.

**Entorno:** Operación normal en ejecución local.

**Cambio en el entorno:** Entrada de palabra sin coincidencia en el diccionario.

**Comportamiento esperado:** El sistema produce una secuencia de señas del alfabeto manual LSCh.

**Medida:** Comprensibilidad ≥90% en pruebas internas.

**Prioridad Arquitectónica:** Media

**Aplicación:** Módulo de deletreo

## Vista Lógica

La vista lógica representa la estructura estática y dinámica del sistema CatCh AI, destacando los componentes principales, su interacción y los flujos de información que se producen durante la ejecución. Esta vista se apoya en tres diagramas fundamentales: diagrama de clases, diagramas de comunicación y diagramas de secuencia, los cuales permiten comprender la organización interna del sistema y su comportamiento frente a diferentes estímulos de entrada.

### Diagrama de Clases

El diagrama de clases ilustra la arquitectura interna del sistema en términos de objetos y sus relaciones.

En él se identifican las clases Orquestador, ExtractorAudio, STT (Whisper), Normalizador, Diccionario, SelectorSeña, MotorAnimación y Logs, junto con la clase UI HTML, que representa la interfaz de entrada del usuario.

Cada clase cumple una responsabilidad específica dentro del flujo de procesamiento:

* **UI HTML** gestiona la recepción de archivos o grabaciones desde el navegador.
* **Orquestador** coordina el flujo completo del sistema, invocando los módulos necesarios en función del tipo de entrada.
* **ExtractorAudio** se encarga de obtener la pista de audio cuando la entrada es un video.
* **STT (Whisper)** realiza la transcripción de voz a texto mediante reconocimiento automático de voz.
* **Normalizador** adapta el texto transcrito al formato lingüístico esperado por el diccionario.
* **Diccionario** contiene la base de datos de señas conocidas y sus identificadores asociados.
* **SelectorSeña** determina si la seña existe o debe representarse mediante deletreo.
* **MotorAnimación** ejecuta la animación correspondiente utilizando Blender o Godot.
* **Logs** registra los eventos del proceso y métricas de desempeño para auditoría técnica.

Este diagrama permite visualizar la modularidad del sistema, la separación de responsabilidades y las dependencias mínimas entre clases, facilitando la mantenibilidad y escalabilidad futuras.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 13 - diagrama de clases

### Diagrama de comunicación

Los diagramas de comunicación muestran cómo los objetos colaboran entre sí mediante el intercambio de mensajes numerados.

En esta versión del sistema se incluyen dos escenarios principales dentro del mismo diagrama:

#### Comunicación - Archivo

Representa el flujo cuando el usuario carga un archivo de audio o video a través de la interfaz HTML.

El sistema extrae el audio (si aplica), lo transcribe, normaliza el texto y busca su correspondencia en el diccionario de señas. Si la palabra no se encuentra, se activa el módulo de deletreo. Finalmente, el motor de animación reproduce la seña seleccionada y los resultados se registran en el módulo de logs.

Los mensajes se numeran del 1 al 8, reflejando el orden exacto de las interacciones entre módulos.

#### Comunicación – Micrófono

Describe el flujo cuando la entrada proviene de una grabación directa de audio desde el navegador.

En este caso no interviene el extractor de audio, sino que el flujo parte directamente en el módulo STT (Whisper). La secuencia continúa con la normalización, búsqueda en el diccionario y animación, siguiendo la misma lógica que el escenario anterior.

Este diagrama enfatiza la continuidad del procesamiento y la baja latencia requerida para mantener la respuesta en tiempo real.

#### Diagrama de Comunicación para ambos casos

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 14 - diagrama de comunicación

Ambos diagramas reflejan la interacción distribuida entre componentes internos y su orquestación centralizada, lo que asegura la coherencia del sistema aun cuando los módulos se ejecuten en equipos o entornos locales distintos.

### Diagrama de Secuencia

Los diagramas de secuencia complementan los de comunicación mostrando la dimensión temporal del proceso.

Se detallan los eventos que ocurren en orden cronológico desde que el usuario ingresa una entrada hasta la generación final de la animación.

Cada secuencia evidencia los puntos de sincronización, el manejo de excepciones (por ejemplo, archivos no válidos o ausencia de seña en el diccionario) y la independencia del flujo de procesamiento respecto de la fuente de entrada.

#### Diagrama de Secuencia – Archivo

#### Diagrama El contenido generado por IA puede ser incorrecto.

Ilustración 15 - diagrama de secuencia con carga de archivo

#### Diagrama de Secuencia – Micrófono (sin latencia)

#### Imagen que contiene Diagrama El contenido generado por IA puede ser incorrecto.

Ilustración 16 - diagrama de secuencia con micrófono

### Modelo Entidad Relación (a futuro)

El Modelo Entidad–Relación (MER) conceptual proyecta la futura estructura de datos que permitirá formalizar la trazabilidad completa del flujo voz → texto → seña → animación en versiones posteriores del sistema CatchAI.

Aunque el MVP actual no implementa una base de datos relacional, este modelo define las entidades lógicas y relaciones necesarias para una futura persistencia estructurada y auditable, sirviendo como base de referencia para la evolución del sistema hacia un entorno escalable y reutilizable.

El objetivo del MER es establecer un modelo conceptual independiente de la tecnología, que capture los elementos más relevantes del dominio, sus relaciones y cardinalidades, anticipando la futura necesidad de almacenamiento, trazabilidad, control de versiones y métricas operativas.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 17 - modelo entidad relación (futuro)

## Vista de Componentes

La vista de componentes detalla cómo se descompone el sistema en módulos ejecutables y bibliotecas con responsabilidades claras, y cómo se enlazan mediante conectores bien definidos. En el MVP, CatCh AI adopta una arquitectura modular orientada a proceso: una capa de interfaz mínima en HTML que permite cargar o registrar audio; un núcleo de orquestación que coordina el pipeline; un conjunto de servicios técnicos especializados (extracción de audio, transcripción, normalización semántica y resolución de señas); y un subsistema de animación que materializa la seña elegida en un avatar 3D. La comunicación entre componentes prioriza conectores locales de muy bajo acoplamiento: HTTP local para la entrada, invocaciones de proceso/biblioteca para las transformaciones, y paso de referencias por sistema de archivos temporal para los artefactos intermedios. Esta organización favorece la reproducibilidad, la portabilidad entre Windows y Ubuntu y la operación sin infraestructura en la nube, en línea con los requisitos no funcionales priorizados.

### Diagrama de Componentes y Conectores

El diagrama de componentes y conectores representa la topología lógica de módulos que intervienen en el pipeline. La UI HTML actúa como el conector al sistema, exponiendo un formulario de carga/grabación y remitiendo la solicitud al Orquestador local mediante un endpoint HTTP. El Orquestador es el corazón del MVP: recibe la referencia de entrada, persiste temporalmente el archivo, invoca a ExtractorAudio para extraer el canal de audio cuando la fuente es un vídeo y encadena la llamada a Whisper (STT) para obtener la transcripción. El texto resultante es procesado por el Normalizador, que aplica limpieza y segmentación y consulta el Diccionario de LSCh para resolver la seña; cuando no existe, deriva al Selector de Seña para generar un deletreo. Con la referencia de animación resuelta, el Motor de Animación (Godot/Blender según el caso de uso) carga la animación desde el repositorio de assets en GLB/FBX y la reproduce sobre el avatar. Durante todo el flujo, el componente de Logs registra trazas, tiempos y errores, y el File System Temporal encapsula los artefactos intermedios con políticas de limpieza. Los conectores son explícitos: HTTP local para la UI, llamadas de proceso o biblioteca para ExtractorAudio/Whisper/Normalizador/Selector, y rutas de archivo para los assets de animación. Esta configuración minimiza dependencias externas y permite aislar fallas por etapa con reintentos acotados.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 18 - Diagrama de componentes

### Diagrama de Paquetes

El diagrama de paquetes agrupa el diseño lógico en cuatro dominios de responsabilidad. El paquete Interfaz contiene la UI HTML, hojas de estilo y el manejador mínimo que envía solicitudes al orquestador; su contrato es estable y deliberadamente simple para reducir fricción en pruebas. El paquete Procesamiento de Señal reúne ExtractorAudio (FFmpeg), STT (Whisper) y los adaptadores necesarios para transformar datos binarios en texto; desde el punto de vista de dependencias, es un paquete “hacia adentro” del que dependen las capas superiores, pero que no conoce la animación. El paquete NLP y Resolución de Signos encapsula el Normalizador, la lógica de consulta del Diccionario y el Selector de Seña/Deletreo; su frontera es textual: recibe oraciones o tokens y devuelve identificadores de animación o secuencias de deletreo. Por último, el paquete Animación y Assets administra el árbol de animaciones, la carga de GLB/FBX, los rigs y la reproducción con Godot o Blender; depende solo de identificadores y rutas, de modo que puede evolucionar con nuevas bibliotecas sin cambiar contratos aguas arriba. Esta partición reduce el acoplamiento, facilita pruebas por paquete y permite trazabilidad directa entre requisitos funcionales y módulos, cumpliendo las metas de modificabilidad y reusabilidad.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 19 - diagrama de paquetes

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 20 - mapa de dependencias

## Vista de Procesos

La vista de procesos describe la dinámica de ejecución del sistema y la coordinación entre actividades concurrentes. En CatCh AI, el pipeline se estructura como una sucesión controlada de etapas con colas de trabajo y almacenamiento temporal. Una solicitud creada por la UI se encola y es atendida por workers especializados: primero extracción de audio, luego transcripción, normalización y, finalmente, resolución y reproducción. Cada etapa produce artefactos temporales y señales de avance que el orquestador utiliza para decidir reintentos o abortos controlados. Este enfoque desacopla duraciones heterogéneas, como la transcripción y la carga de animaciones, y permite sostener procesamientos consecutivos sin caída, objetivo que se refleja en los RNF de rendimiento (tiempos de respuesta para entradas ≤1 minuto), procesamiento continuo y manejo robusto de errores.

### Diagrama de Actividades

El diagrama de actividades formaliza el flujo técnico end-to-end. La ejecución comienza con la recepción de un archivo de audio o vídeo; si la fuente es vídeo, se extrae el canal de audio. La pista resultante se transcribe con Whisper y el texto se normaliza para eliminar ruido, segmentar y estandarizar glosas. Con el texto limpio, el sistema verifica la existencia de la seña en el diccionario: si está disponible, genera una lista de animaciones a reproducir; en caso contrario, compone una secuencia de deletreo conforme a LSCh. Paralelamente, el flujo contempla la preparación y mantenimiento de assets de animación mediante FreeMoCap y Blender, que actualizan el repositorio GLB/FBX y alimentan el árbol de animaciones. En la fase final, el motor de animación carga la seña (o la animación de error) y la ejecuta en el avatar, cerrando el ciclo con el registro de métricas. Las decisiones y bifurcaciones del diagrama hacen explícitos los puntos de control de calidad (existencia de animación, errores de extracción o transcripción) y sostienen la trazabilidad de tiempos y fallos.

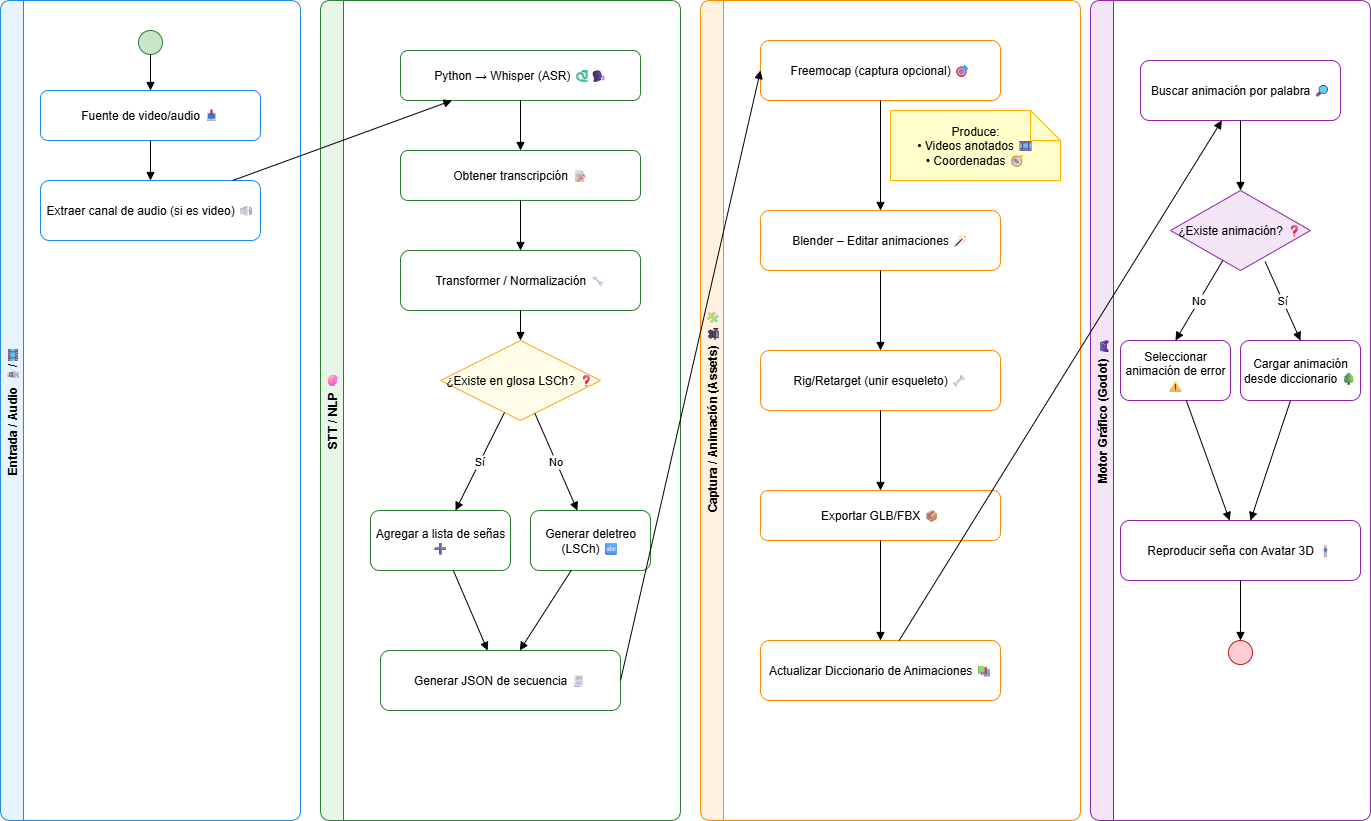


Ilustración 21 - diagrama de actividades

## Vista Física

La vista física precisa cómo se despliega el sistema en hardware y sistemas operativos reales, así como los canales de comunicación entre nodos. El MVP se concibe para funcionar íntegramente en entornos locales, sin servicios en la nube: el navegador del usuario actúa desde el mismo equipo o desde otro host en la LAN, mientras que el procesamiento y la reproducción se ejecutan en una estación de trabajo con Windows 10/11 o Ubuntu 22.04. Los artefactos se intercambian a través de un sistema de archivos temporal y los módulos técnicos (FFmpeg, Whisper, Normalizador y el motor de animación) se orquestan en el mismo host para minimizar latencia y dependencias. Esta topología facilita la reproducibilidad de resultados, simplifica el soporte y se ajusta a las restricciones de recursos del proyecto. En escenarios de proyección, la UI, los servicios y el equipo de captura/emisión pueden separarse en hosts distintos de la misma red local, manteniendo el mismo contrato de comunicación.

### Diagrama de Despliegue

El diagrama de despliegue muestra dos configuraciones coherentes con el plan de evolución. En la configuración base, una única estación de trabajo aloja el sistema operativo, el runtime de la aplicación, las herramientas técnicas (FFmpeg, Whisper y Godot/Blender) y los artefactos locales: archivos temporales, configuraciones y logs cifrados. El navegador del usuario interactúa con el orquestador vía HTTP local para enviar el archivo y consultar el estado de procesamiento. La salida es la reproducción en ventana del avatar, que puede ser capturada por OBS si se requiere difusión posterior. En la configuración futura, la solución se distribuye en la LAN: un host sirve la UI, un segundo host concentra el procesamiento y un tercero ejecuta el motor de animación y la captura; las transferencias se realizan por HTTP y por archivos/colas locales, manteniendo la política de archivos temporales y el cifrado de logs. Esta vista consolida los atributos de calidad de compatibilidad, reproducibilidad y operatividad definidos para el MVP y deja un camino claro para escalar sin cambiar contratos de integración.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 22- diagrama de despliegue (actual)

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 23- diagrama de despliegue (futuro)

## Decisiones de Diseño y Selección de Alternativas

El desarrollo del MVP de CatCh AI se enmarca en un contexto de recursos materiales y humanos limitados, pero con un objetivo claro: demostrar la factibilidad de un sistema capaz de traducir voz a señas chilenas (LSCh) mediante la integración de modelos de reconocimiento de voz, procesamiento lingüístico y animación 3D. A lo largo del diseño, se analizaron distintas alternativas técnicas y conceptuales, y se tomaron decisiones estratégicas en torno a la arquitectura, herramientas, interfaces, flujo de procesamiento y gestión de datos. A continuación, se detallan las principales decisiones adoptadas y su justificación.

### Elección del Entorno de Ejecución

La primera decisión relevante fue prescindir del uso de servicios en la nube para la ejecución y entrenamiento de los modelos.

Si bien las plataformas cloud (AWS, GCP, Azure, Huawei Cloud) ofrecen ventajas en escalabilidad, disponibilidad y potencia de cómputo, su uso resultó inviable para este MVP por dos razones principales:

* Limitaciones presupuestarias y de conectividad, que impiden mantener entornos virtuales o GPU’s remotas.
* Necesidad de control total sobre los artefactos y dependencias, debido a la experimentación intensiva con modelos locales y herramientas de animación.

Por ello, se optó por una arquitectura 100 % local, ejecutable en estaciones de trabajo con Windows 10/11 o Ubuntu 22.04, utilizando dependencias de código abierto. Esta decisión permite garantizar reproducibilidad, independencia de red y control del rendimiento, además de alinearse con los requerimientos no funcionales RNF.01 (tiempo de respuesta local) y RNF.11 (compatibilidad multiplataforma).

### Arquitectura Modular y Orquestada

Inicialmente, se consideró integrar todo el flujo en un solo script monolítico de Python. Sin embargo, esta aproximación dificultaba la trazabilidad de errores y el reemplazo futuro de componentes.

Se decidió, en cambio, adoptar una arquitectura modular, basada en un orquestador principal que coordina subsistemas especializados: extracción de audio, transcripción, normalización de texto, consulta de diccionario y renderización.

Cada módulo puede ejecutarse de manera independiente y comunicar sus resultados mediante archivos JSON o rutas de archivo temporales.

Esta elección responde a criterios de modificabilidad, mantenibilidad y extensibilidad, permitiendo sustituir Whisper por otro motor de reconocimiento o reemplazar el módulo de animación sin alterar el flujo general. También facilita la futura evolución hacia una API escalable o una versión distribuida.

### Selección de Herramientas Técnicas

Durante la fase de exploración, se evaluaron varias herramientas para cada función clave:

* **Reconocimiento de voz (STT):** se compararon Google Speech API, Vosk, DeepSpeech y Whisper. La decisión se inclinó por Whisper (OpenAI) debido a su precisión con acentos hispanos, su licencia abierta y su capacidad de ejecución local en CPU, lo que reduce la dependencia de conectividad.
* **Procesamiento y normalización lingüística:** se optó por un pipeline basado en Python y Transformers ligeros, descartando el uso de servicios NLP externos. La normalización se implementó con expresiones regulares y modelos livianos de tokenización.
* **Captura y animación 3D:** se descartaron motores propietarios (Unity, Unreal Engine) por requerir licencias o dependencias específicas. Se seleccionó Blender para edición y retargeting, y Godot Engine para reproducción, debido a su integración nativa con formatos GLB/FBX, su soporte multiplataforma y su bajo consumo de recursos.
* **Captura de movimiento:** para experimentación con gestos manuales se eligió FreeMoCap, una solución open-source compatible con múltiples cámaras y hardware heterogéneo, que permite capturar coordenadas de movimiento sin sensores especializados.

Esta combinación permite mantener una línea tecnológica homogénea en torno a software libre, coherente con las limitaciones de infraestructura y el objetivo del proyecto.

### Estrategia de Comunicación entre Componentes

Otra decisión crucial fue la forma en que los módulos se comunican entre sí. Se analizaron dos estrategias:

* Comunicación por API REST local.
* Comunicación por intercambio de archivos (JSON y rutas de video).

Se seleccionó la segunda opción para el MVP, ya que evita la necesidad de levantar servicios web permanentes y simplifica la depuración.

Cada etapa del proceso deja un artefacto tangible (transcripción, texto normalizado, animación generada) que puede auditarse manualmente.

En futuras iteraciones, la arquitectura permitirá reemplazar los archivos por endpoints REST cuando el sistema se escale hacia una API formal.

### Manejo de Errores y Logs

Dado que no existe supervisión humana directa durante el flujo, se adoptó un mecanismo de registro automático de eventos que documenta cada fase del pipeline.

Los logs se almacenan localmente en formato cifrado y se conservan junto con los metadatos de los archivos procesados por 24 horas, conforme al RNF.05 (protección de archivos).

Esta decisión responde a la necesidad de auditar el comportamiento del sistema sin interrumpir el flujo de trabajo, garantizando continuidad ante fallos y facilitando la detección de errores persistentes.

### Diseño del Flujo de Procesamiento (Pipeline)

El flujo de procesamiento fue diseñado como un pipeline secuencial con bifurcaciones controladas, en el que cada nodo actúa como un proceso autocontenido.

El diseño prioriza la estabilidad sobre la concurrencia, ya que el hardware disponible no permite ejecución paralela de alta carga.

No obstante, el orquestador está preparado para procesar múltiples solicitudes de manera secuencial sin pérdida de estado, cumpliendo el RNF.03 de procesamiento continuo.

Las decisiones de bifurcación (por ejemplo, existencia o ausencia de seña en el diccionario) se documentan explícitamente en el diagrama de actividades, permitiendo aislar cuellos de botella y medir los tiempos por fase.

### Consideraciones de Seguridad, Privacidad y Gobernanza de la IA

En el contexto del desarrollo del MVP de CatCh AI, la seguridad y la ética en el manejo de los datos adquieren relevancia, dado que el sistema trabaja con material audiovisual y resultados generados por inteligencia artificial. Aunque no se manejan datos personales sensibles ni almacenamiento en la nube, se ha considerado desde la arquitectura la aplicación de principios compatibles con las normas ISO/IEC 42001:2023 (Sistemas de Gestión de IA) e ISO/IEC 23894:2023 (Gestión de Riesgos en IA), asegurando que incluso en un entorno académico y experimental, la tecnología se utilice de forma segura, responsable y trazable.

El diseño general del sistema se rige por tres principios esenciales: seguridad de la información, transparencia del procesamiento y control humano verificable.

En primer lugar, se implementó una política de almacenamiento temporal controlado. Todos los archivos procesados —ya sean de audio, video o animaciones generadas— se guardan únicamente durante el tiempo necesario para ejecutar el pipeline y validar su resultado. Estos archivos se almacenan en un directorio local protegido por permisos del sistema operativo, evitando el acceso de otros usuarios del equipo. Una vez concluido el procesamiento, los archivos pueden eliminarse manualmente o mediante scripts de limpieza, garantizando así el principio de minimización de datos establecido por la ISO/IEC 42001.

En segundo lugar, la arquitectura integra un sistema básico de registro de eventos (logs) que documenta las operaciones ejecutadas en cada etapa del proceso: carga del archivo, extracción de audio, transcripción, normalización y generación de animación. Si bien los registros no están cifrados, se almacenan localmente y de manera controlada dentro del mismo entorno del orquestador, sin exposición a redes externas. Este enfoque cumple con el principio de trazabilidad y transparencia algorítmica, permitiendo revisar el comportamiento del sistema en caso de errores o fallos sin comprometer información sensible.

Además, el flujo se diseñó bajo el enfoque de seguridad por aislamiento. Ningún módulo realiza conexiones hacia internet ni transmite datos a terceros. Todos los procesos, desde Whisper hasta Godot, operan en el mismo entorno local, lo que reduce significativamente la superficie de ataque y elimina riesgos asociados a la comunicación con servicios externos. En concordancia con los lineamientos de confidencialidad, integridad y disponibilidad (CIA), esta arquitectura minimiza vulnerabilidades y facilita el control total de los datos.

Por otra parte, el desarrollo se alinea con los principios de gobernanza responsable en IA definidos por la ISO/IEC 42001. Aunque el sistema actual no incorpora mecanismos automatizados de decisión sobre personas ni procesamiento de datos personales, sí aplica criterios de transparencia técnica, explicabilidad del proceso y supervisión humana. Cada resultado puede ser reproducido o auditado mediante la revisión de los archivos intermedios y los logs del sistema. Esto permite demostrar que las decisiones tomadas por los modelos (por ejemplo, el reconocimiento de una palabra o la selección de una seña) son verificables, observables y reversibles por un operador humano.

Todas las decisiones de diseño en materia de seguridad se orientan hacia un objetivo mayor: que CatCh AI evolucione en el futuro hacia un sistema que no solo cumpla con los requisitos funcionales, sino que también pueda integrarse en un marco de gestión ética de la inteligencia artificial, conforme a los estándares internacionales.

#### Tabla 8.7.1 – Alineación del MVP con los Principios de la Norma ISO/IEC 42001:2023

|  |  |  |
| --- | --- | --- |
| **Principio de la ISO/IEC 42001:2023** | **Descripción general del principio** | **Aplicación en el sistema CatCh AI (versión MVP local)** |
| **Transparencia y explicabilidad** | El sistema debe permitir comprender de forma clara cómo se generan los resultados de IA y en qué condiciones. | Cada etapa del pipeline (extracción, transcripción, normalización y animación) deja artefactos intermedios y logs que evidencian el proceso. Las decisiones del modelo Whisper y del normalizador son revisables, garantizando la posibilidad de auditoría y explicación del comportamiento del sistema. |
| **Trazabilidad y registro del ciclo de vida** | Debe existir la capacidad de rastrear las operaciones y sus resultados en todas las fases del procesamiento. | Se conserva un registro local en texto plano con eventos y resultados de cada módulo. Aunque no está cifrado, se encuentra aislado del usuario final y permite reconstruir el flujo de procesamiento en caso de revisión técnica o auditoría académica. |
| **Seguridad y protección de la información** | La arquitectura debe minimizar la exposición y el riesgo de acceso no autorizado. | Los datos se almacenan solo durante el tiempo de ejecución y en directorios protegidos por permisos de sistema. No existen conexiones a internet ni transferencias externas. Los archivos se eliminan manual o automáticamente tras su uso, cumpliendo la política de almacenamiento temporal seguro. |
| **Minimización y retención de datos** | La IA debe utilizar solo los datos estrictamente necesarios y mantenerlos el tiempo mínimo indispensable. | Los archivos de entrada (audio o video) se procesan una sola vez y luego se eliminan. No se almacenan datos de usuarios ni se conservan copias. El sistema no crea repositorios persistentes ni bases de datos. |
| **Control humano y supervisión verificable** | Las decisiones automatizadas deben poder ser revisadas o anuladas por un operador humano. | Aunque el sistema opera de forma automática, todos los resultados son verificables mediante los artefactos producidos (textos, animaciones, logs). Un operador puede validar manualmente la correspondencia entre el audio original y la seña generada. |
| **Responsabilidad y rendición de cuentas** | Debe existir claridad sobre los responsables del diseño, uso y mantenimiento de la IA. | El proyecto se desarrolla en un entorno académico bajo la supervisión del equipo de desarrollo de DUOC UC. Las decisiones de arquitectura y ejecución están documentadas, y cada iteración de software cuenta con responsable asignado. |
| **Gestión de riesgos en IA (ISO/IEC 23894)** | Los riesgos asociados al uso de IA deben identificarse, evaluarse y mitigarse durante el ciclo de vida del sistema. | El Plan de Riesgos del proyecto incorpora una categoría específica de riesgos tecnológicos de IA: latencia, precisión del modelo, errores de transcripción y fallas en la generación de animaciones. Cada uno cuenta con medidas de mitigación y seguimiento definidas. |
| **Confiabilidad técnica y reproducibilidad** | Los resultados deben ser consistentes ante las mismas condiciones de entrada. | Los modelos de IA operan con parámetros fijos, garantizando resultados reproducibles en distintos equipos. Los RNF.11 y RNF.12 aseguran compatibilidad multiplataforma y reproducibilidad de salidas idénticas. |
| **Ética y no discriminación** | Los sistemas de IA deben evitar sesgos, discriminación o usos indebidos. | El MVP no toma decisiones sobre personas ni aplica análisis semánticos que impliquen juicios o sesgos. El diseño se limita a traducción de voz a seña, sin interpretación subjetiva. Futuras versiones contemplarán validación con intérpretes certificados para reforzar imparcialidad. |

## Análisis de Reutilización

El análisis de reutilización tiene como objetivo identificar los componentes, librerías y estructuras del sistema que pueden ser aprovechados en iteraciones futuras o en otros proyectos que compartan características técnicas o funcionales con CatCh AI. Dado que el MVP se ha diseñado bajo un enfoque modular y de bajo acoplamiento, su arquitectura presenta una base sólida para extender, reemplazar o adaptar elementos sin afectar el núcleo funcional del sistema. Esta capacidad de reutilización es resultado directo de las decisiones de diseño adoptadas en las secciones anteriores: independencia tecnológica, procesamiento por etapas y comunicación mediante archivos intermedios o interfaces simples.

En su estado actual, el sistema se compone de módulos bien definidos —Orquestador, STT (Whisper), Normalizador, Diccionario, Motor de Animación, Logs y Gestión de Archivos Temporales—, los cuales funcionan como bloques autónomos que se comunican entre sí a través de datos estructurados. Esta estructura permite que varios de ellos puedan ser reutilizados íntegramente, ya sea dentro del mismo dominio (traducción e interpretación automática) o en áreas de desarrollo afines (procesamiento multimedia, educación accesible, o investigación de interfaces inclusivas).

### Reutilización de Componentes Técnicos

#### Módulo de Orquestación

El Orquestador constituye el punto de entrada principal del sistema y su lógica de coordinación puede reutilizarse en cualquier pipeline secuencial de procesamiento de datos o multimedia. Su capacidad de controlar etapas, validar salidas intermedias y registrar eventos lo convierte en un componente genérico aplicable en sistemas de análisis de voz, visión o texto. Con pequeñas modificaciones, puede integrarse a nuevas arquitecturas mediante llamadas REST o colas de tareas, conservando la misma estructura modular.

#### Motor STT (Whisper)

El módulo de transcripción puede ser empleado de forma independiente en otros proyectos que requieran reconocimiento de voz en español. Su ejecución local, sin conexión a internet, lo hace especialmente valioso en entornos con restricciones de red o en proyectos educativos y de investigación donde la privacidad es prioritaria. Además, su API interna puede adaptarse fácilmente a flujos de entrada distintos (micrófono, video, o transmisión en vivo).

#### Módulo de Normalización y Tokenización

El pipeline lingüístico desarrollado para CatCh AI —que transforma texto crudo en frases normalizadas y compatibles con el diccionario LSCh— puede reutilizarse en otros contextos donde se requiera limpieza, estandarización o segmentación de texto. Este módulo, por su naturaleza, es útil en sistemas de subtitulado automático, chatbots o preprocesamiento de corpus lingüísticos para lengua chilena.

#### Diccionario LSCh y Selector de Seña

El componente que consulta y selecciona animaciones en base a palabras o glosas puede ser extendido como módulo semántico reutilizable en proyectos relacionados con educación inclusiva o traducción multimodal. Su estructura de índices y referencias a archivos GLB/FBX es fácilmente adaptable a otros idiomas de señas o a repositorios de animaciones 3D. A largo plazo, este módulo podría evolucionar hacia una API de consulta de señas reutilizable en plataformas educativas, gubernamentales o mediáticas.

### Reutilización de Recursos y Modelos

Además de los módulos funcionales, el sistema incorpora recursos reutilizables en forma de modelos, datasets y scripts auxiliares. El modelo Whisper puede mantenerse como componente estándar para futuros desarrollos de transcripción. Los scripts de Blender empleados en el rigging y exportación de animaciones pueden transformarse en plantillas para generar repositorios de señas en 3D en distintos contextos. Asimismo, las configuraciones del pipeline de captura con Freemocap pueden adaptarse a nuevas sesiones de grabación o a la creación de un dataset nacional de LSCh animada, aportando valor más allá del proyecto actual.

### Reutilización Arquitectónica

Desde el punto de vista estructural, la arquitectura de CatCh AI sigue un patrón de capas y servicios independientes, donde cada módulo tiene interfaces definidas y contratos explícitos. Este patrón puede replicarse en otros sistemas de IA que combinen procesamiento secuencial, análisis multimodal y salida gráfica. Por ejemplo, la arquitectura puede servir como base para proyectos de:

* Reconocimiento de gestos corporales.
* Interpretación de texto a movimiento.
* Generación audiovisual a partir de modelos lingüísticos.

La reutilización arquitectónica se traduce en un marco flexible de integración: la sustitución de un módulo (por ejemplo, Whisper) no altera el comportamiento de los demás, siempre que se mantenga el formato de comunicación entre etapas. Esto permite escalar o migrar el proyecto hacia microservicios o contenedores en fases futuras.

## Mockups

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración 24 - Mockup Página de inicio

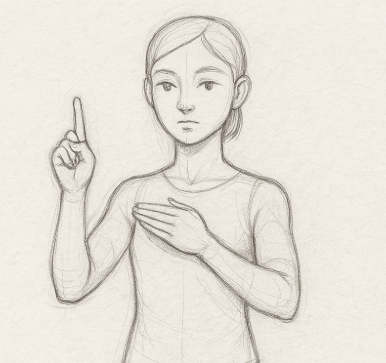


Ilustración 25 - Mockup de avatar 3d

## 11. Anexos

[Planilla de Requerimientos](https://docs.google.com/spreadsheets/d/188pjUmcWKTqnXb6MYQZPqOsCzOVTr-lt/edit?usp=drive_link&ouid=101055761493027937484&rtpof=true&sd=true)

[Documento de Especificación de Casos de Uso](https://docs.google.com/document/d/1-pjCre4w1OdcvfUHcCoKdReFe3-MI64Q/edit?usp=drive_link&ouid=101055761493027937484&rtpof=true&sd=true)

[Escenarios de Calidad y Resumen de Casos de Uso](https://docs.google.com/spreadsheets/d/1Ac19YFbx-AoeIAZlBqB9biYhq1eE2SLF/edit?usp=drive_link&ouid=101055761493027937484&rtpof=true&sd=true)

[Carpeta de Diagramas de DAS](https://drive.google.com/drive/folders/1hunGI1XzyDqsothaE9_vYP_8erwHBDH3?usp=drive_link)

[Especificación de Requisitos de Software](https://docs.google.com/document/d/1cRU1c2ob3wLzS4pPbFW6RWnxxIgMB5x7/edit?usp=drive_link&ouid=101055761493027937484&rtpof=true&sd=true)