*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 | 2.1 | Creación al desarrollo documento “DAS”. | Eyleen Collado |

Tabla de contenido

[1. Introducción 5](#_Toc209912315)

[1.1 Contexto del Problema 5](#_Toc209912316)

[1.2 Propósito 5](#_Toc209912317)

[1.3 Ámbito 6](#_Toc209912318)

[1.4 Definiciones, acrónimos y abreviaciones 6](#_Toc209912319)

[1.5 Referencias 7](#_Toc209912320)

[1.6 Resumen ejecutivo 8](#_Toc209912321)

[1.7 Representación 9](#_Toc209912322)

[2. Metas y Restricciones de la Arquitectura 10](#_Toc209912323)

[2.1 Metas de la arquitectura 10](#_Toc209912324)

[2.2 Restricciones de la Arquitectura 10](#_Toc209912325)

[2.3 Otros antecedentes y consideraciones 10](#_Toc209912326)

[3. Vista de Casos de Uso y Escenarios de Calidad 12](#_Toc209912327)

[3.1 Modelo de Casos de Uso 12](#_Toc209912328)

[3.2 Especificación de Casos de Uso Relevantes 12](#_Toc209912329)

[3.3 Especificación de los Escenarios de Calidad Relevantes 20](#_Toc209912330)

[4. Vista Lógica 22](#_Toc209912331)

[4.1 Diagrama de Clases 22](#_Toc209912332)

[4.2 Diagrama Modelo Entidad Relación 22](#_Toc209912333)

[4.3 Diagrama de Secuencia 23](#_Toc209912334)

[5. Vista de Componentes 24](#_Toc209912335)

[5.1 Diagrama de Componentes y Conectores 24](#_Toc209912336)

[5.2 Diagrama de Paquetes 25](#_Toc209912337)

[6. Vista de Procesos 27](#_Toc209912338)

[6.1 Diagrama de Actividades 27](#_Toc209912339)

[7. Vista Física 28](#_Toc209912340)

[7.1 Diagrama de Despliegue 28](#_Toc209912341)

[8 Decisiones de Diseño y Selección de Alternativas 29](#_Toc209912342)

[9. Análisis de Reutilización 30](#_Toc209912343)

[10. Mockups 31](#_Toc209912344)

[10.1 Mockup de Ingreso de Factura electrónica 31](#_Toc209912345)

[10.2 Mockup de Reserva de Cita por parte del cliente 31](#_Toc209912346)

[10.3 Mockup de Registro de Nuevo Proveedor 32](#_Toc209912347)

[10.4 Mockup de Registro de nuevo cliente por parte del empleado 33](#_Toc209912348)

[10.5 Mockup de Generación de Informes 34](#_Toc209912349)

[11. Anexos 35](#_Toc209912350)

## 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 actores
* 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 de Casos de Uso y Escenarios de Calidad:** 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 a 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 cluster.
* **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 de Casos de Uso y Escenarios de Calidad

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.

### 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.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

### 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 CatChAI, 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.

### 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.

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.

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

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

### 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.

## 5. Vista de Componentes

### 

### 5.1 Diagrama de Componentes y Conectores

Diagrama de componentes de Registro de facturas.



Diagrama de componentes de Reserva de citas



Diagrama de componentes de Recepción de Producto



### 5.2 Diagrama de Paquetes

Diagrama de paquetes de Recepción de Producto



Diagrama de paquetes de Registro de Factura

Diagrama de paquetes de Reserva de Citas



## 6. Vista de Procesos

### 

### 6.1 Diagrama de Actividades



## 7. Vista Física

### 7.1 Diagrama de Despliegue

**Ilustración 6: Diagrama de Despliegue**

## 8 Decisiones de Diseño y Selección de Alternativas

## 9. Análisis de Reutilización

## 10. Mockups

## 11. Anexos

Documento de especificación de casos de uso

Diagrama de casos de uso

Diagrama de clases

MER

Diagrama de secuencia

Diagrama de despliegue

Diagrama de componente de registro de factura

Diagrama de componente de recepción de producto

Diagrama de componente de reserva de cita

Diagrama de paquetes de registro de factura

Diagrama de paquetes de recepción de producto’

Diagrama de paquetes de reserva de cita

Diagrama de actividad de registro de factura