DUOC UC – Sede Puerto Montt  
Escuela de Informática y Telecomunicaciones

Logotipo

El contenido generado por IA puede ser incorrecto.  
INFORME DE AVANCE – PROYECTO APT  
Sistema de Traducción de Voz a Lengua de Señas Chilena (LSCh) – “CatchAI”

Integrantes: Eyleen Collado (Product Owner), Marcos Bombalas (Scrum Master / Desarrollador), Luis Gómez (Procesamiento de Voz y LLM), Marcos Hernández (Modelado/Avatar 3D)  
Profesor guía: Giocrisrai Godoy – Germán Barrientos  
Fecha: 6 de octubre de 2025

# 1. Introducción

El presente informe académico expone el estado actual del proyecto “CatchAI”, correspondiente al segundo sprint (en curso) del proceso de implementación del Sistema de Traducción de Voz a Lengua de Señas Chilena (LSCh). El propósito de la iteración es consolidar la arquitectura técnica y documental que sustenta el MVP: un flujo funcional de traducción que conecta reconocimiento de voz, procesamiento lingüístico intermedio y animación 3D mediante un avatar.  
  
Durante el periodo comprendido entre la semana 5 y la semana 9, el equipo ha concentrado esfuerzos en cuatro frentes complementarios: (i) integración técnica del pipeline audio → texto → glosa → animación; (ii) modularización del cliente Godot y corrección de transiciones; (iii) construcción del componente de contextualización semántica soportado por LLM y servicios locales; y (iv) desarrollo progresivo del avatar 3D con foco en expresividad y compatibilidad con el pipeline. En paralelo, se han resguardado los aspectos metodológicos y de trazabilidad (sprints, Kanban y redacción del DAS), de acuerdo con los criterios de evaluación del Proyecto APT para la entrega del 6 de octubre de 2025.

# 2. Desarrollo del Proyecto

## 2.1 Aporte de Marcos Bombalas (Integración)

## Marcos Bombalas, desde su rol de Scrum Master y desarrollador principal del cliente, ha sido responsable de la consolidación técnica del pipeline de integración entre el modelo de transcripción Whisper y el motor de animaciones Godot, eje central del funcionamiento del MVP. Su trabajo se ha centrado en la refactorización completa de la arquitectura del cliente, transformando una estructura monolítica en un sistema modular y escalable compuesto por múltiples scripts especializados, cada uno encargado de una parte del flujo operativo: comunicación en red (WsClient.gd), gestión de cola y despacho de glosas (GlossQueue.gd) y control de animaciones con blending (AnimController.gd).

## Gracias a esta división de responsabilidades, se logró mejorar significativamente la legibilidad del código, reducir los acoplamientos internos y optimizar el mantenimiento del sistema, facilitando futuras iteraciones del producto.

## Además, desarrolló la corrección del error de repetición de señas, que generaba bloqueos en el avatar al ejecutar la misma animación consecutivamente. Implementó una lógica de transición temporal que introduce un estado intermedio (idle) entre señas, asegurando un retorno fluido al estado base y eliminando artefactos visuales durante la reproducción.

## En paralelo, reestructuró la escena principal de Godot, añadiendo nodos dedicados para el control del reproductor de animaciones, la interfaz de usuario de pruebas, la gestión de la cola de glosas y el cliente WebSocket, mejorando la separación funcional de componentes. También verificó la conexión estable entre el servidor local Whisper y el cliente, validando la transmisión de datos en formato JSON y su mapeo directo a animaciones específicas del AnimationPlayer.

## Los resultados de su implementación se reflejan en un flujo validado de entrada y salida (audio → texto → glosa → animación), con una latencia promedio de 1 a 2 segundos en pruebas locales, lo que confirma la viabilidad técnica del sistema base.

## En conjunto, estas mejoras han incrementado la confiabilidad del MVP, su mantenibilidad y la velocidad de desarrollo del equipo, garantizando que los componentes puedan evolucionar sin comprometer la estabilidad del sistema general.

## 2.2 Aporte de Luis Gómez (Procesamiento de Voz, Servicio Local y Contextualización Semántica)

## Luis Gómez, en su rol de desarrollador enfocado en el procesamiento de voz y el módulo de lenguaje, ha centrado su trabajo en la optimización del flujo de transcripción y contextualización semántica, fortaleciendo la interacción entre el modelo de reconocimiento Whisper Live, el servicio local de comunicación y el glosario de la Lengua de Señas Chilena (LSCh).

## Durante el desarrollo del sprint, modificó el formato de recepción de las transcripciones generadas por Whisper Live, permitiendo un manejo más eficiente de los datos de entrada y una mejor compatibilidad con el flujo general del sistema. Además, creó un dataset especializado para el diccionario LSCh, utilizado como base para la normalización de glosas y la generación de correspondencias entre texto y seña.

## Uno de los hitos más relevantes de su trabajo fue la implementación de un servicio local mediante FastAPI, que permite enviar y recibir instrucciones desde distintas fuentes, actuar como punto intermedio entre módulos y ofrecer una interfaz interactiva para pruebas de usuario y validaciones de flujo. Esta estructura no solo otorga independencia funcional a cada componente, sino que también habilita un entorno de experimentación que acelera la depuración y la integración entre subsistemas.

## Complementariamente, Gómez desarrolló un modelo de contextualización de lenguaje natural que traduce el texto transcrito en secuencias coherentes con el glosario LSCh, utilizando modelos de lenguaje de gran escala (LLM) y bases de datos vectoriales. Este módulo semántico representa una ampliación significativa del MVP, ya que permite mejorar la fluidez y la naturalidad de las traducciones, reduciendo el uso del deletreo como solución de respaldo y acercando el comportamiento del sistema a una interpretación más contextual y humana.

## Entre las limitaciones actuales se destaca el alto consumo computacional derivado de la ejecución simultánea de los modelos Whisper y LLM, sumado al uso de tokens gratuitos que restringen el volumen de procesamiento continuo. Estas condiciones obligan a una gestión cuidadosa de los recursos y al ajuste de parámetros para evitar latencias elevadas.

## Los próximos pasos de su línea de trabajo incluyen la integración completa del módulo semántico con el pipeline operativo, la reducción de los tiempos de respuesta mediante estrategias de optimización de cómputo y la evaluación de mecanismos de cacheo y priorización de tokens para sostener un rendimiento estable en el entorno de pruebas.

## 2.3 Aporte de Marcos Hernández (Avatar 3D y Compatibilidad de Pipeline)

Marcos Hernández, encargado del desarrollo del avatar 3D para la interpretación de señas, ha centrado su trabajo en la construcción de un modelo humanoide optimizado para la reproducción de movimientos complejos, expresiones faciales y gestualidad manual propia de la Lengua de Señas Chilena (LSCh). Su labor representa un eje fundamental dentro del proyecto, ya que el avatar constituye el componente visual principal del sistema y el medio a través del cual se materializa la traducción de voz a seña.

Durante este periodo,Marcos Hernández realizó un cambio estratégico en la línea de modelado, descartando la utilización de un modelo base de DAZ Studio debido a incompatibilidades con el pipeline técnico del proyecto y adoptando un modelo propio desarrollado en iteraciones sucesivas. Este enfoque permitió mantener control total sobre la topología, las proporciones anatómicas y la optimización del rendimiento, asegurando la compatibilidad con las herramientas de rigging y animación de Blender y Godot.

El proceso de desarrollo se estructuró en tres versiones iterativas:

* Versión 1, centrada en un bloqueo rápido de geometría con polígonos simples para validar proporciones y escalas corporales;
* Versión 2, que consolidó una forma humanoide completa, con topología limpia, loops definidos en manos y rostro, y un rig estable para pruebas iniciales de señas;
* Versión 3, actualmente en curso, que incorpora mayor detalle anatómico, mejor definición de musculatura y la inclusión de expresiones faciales básicas, necesarias para una comunicación visual más expresiva y precisa.

De manera paralela, el desarrollador implementó revisiones de compatibilidad por iteración, evaluando la correcta exportación e importación de los modelos en formato GLB/FBX y realizando pruebas de poses críticas de señas para detectar posibles artefactos o colapsos de malla. Esta metodología ha permitido anticipar y mitigar riesgos asociados a la articulación de dedos, la claridad de las expresiones y el balance entre detalle y rendimiento en tiempo real.

En el ámbito de la calidad visual, se definió un conjunto mínimo de blendshapes y rig facial, suficientes para representar emociones y movimientos finos sin incurrir en el denominado “efecto uncanny valley”. Además, se ha considerado la posibilidad de aplicar niveles de detalle (LOD) para optimizar el desempeño en tiempo real del avatar durante la ejecución del pipeline de animaciones.

## A la fecha, el objetivo correspondiente al Sprint 2 —presentar un modelo humanoide operativo y funcional dentro del entorno Godot— se encuentra plenamente cumplido, y los avances de la Versión 3 permiten adelantar parcialmente el trabajo planificado para el Sprint siguiente. Estas mejoras han fortalecido la dimensión visual y técnica del proyecto, asegurando una base sólida para la fase de animación avanzada y sincronización con el sistema de interpretación semántica.

## 2.4 Aporte de Eyleen Collado (Documentación)

Eyleen Collado, en su rol de Product Owner y coordinadora general del proyecto CatchAI, ha liderado la construcción integral de la documentación técnica, la planificación estratégica y el control metodológico del desarrollo, asegurando la coherencia entre los objetivos académicos y los entregables técnicos.

Su labor ha consistido en articular la visión global del sistema, orientar las decisiones técnicas del equipo, garantizar la trazabilidad de los procesos y establecer las bases que sustentan la gestión del proyecto tanto en sus aspectos funcionales como no funcionales.

Entre sus principales aportes destaca la elaboración completa del marco documental que soporta el proyecto, compuesto por los siguientes artefactos:

* Documento de Arquitectura del Sistema (DAS), en el cual se describen las vistas: lógica, de procesos y de desarrollo, junto con los diagramas de clases, secuencia, comunicación y flujo de datos, que definen la estructura modular del sistema y su interacción entre componentes.
* ERS – Especificación de Requisitos del Software, documento que establece los requerimientos funcionales y no funcionales del MVP, la clasificación de prioridades, los criterios de aceptación y los indicadores de desempeño (KPI).
* Documento de Especificación de Casos de Uso, que detalla los actores, flujos principales, alternativos y postcondiciones de cada interacción relevante dentro del sistema.
* Plan de Riesgos y Matriz de Riesgos, donde se identifican las amenazas técnicas, metodológicas y operativas, junto con sus estrategias de mitigación.
* Matriz RACI, que define las responsabilidades de cada integrante del equipo en las fases de desarrollo, documentación, pruebas y control de calidad.
* Planificación de Sprints, Product Backlog, Escenarios de Calidad y Casos de Prueba, Planilla de Requerimientos, y Trazabilidad RF–RNF, que permiten monitorear de forma continua el cumplimiento de los objetivos iterativos.
* EDT (Estructura de Desglose del Trabajo), Monte Carlo de Costos y Plazo y Gantt principal (Proyecto\_Capstone.mpp), los cuales conforman la base de control de tiempo, recursos y secuencia de actividades dentro del ciclo de vida del proyecto.

A través de esta estructura,ha logrado consolidar un ecosistema documental que garantiza la trazabilidad entre las etapas de análisis, diseño, implementación y control, facilitando el seguimiento del avance por entregables y la coordinación entre los miembros del equipo. Este conjunto de documentos —todos elaborados por ella— constituye el núcleo de referencia para las evaluaciones de sprint, las revisiones internas y las entregas académicas intermedias.

Asimismo, ha desempeñado un rol activo en la supervisión del cumplimiento de la metodología Scrum, coordinando reuniones, revisión de historias de usuario y control de los flujos en el tablero Kanban. De igual forma, ha sido responsable de la coherencia entre los artefactos técnicos y los requerimientos académicos del Proyecto APT, alineando la documentación con los criterios evaluativos institucionales y las rúbricas de avance.

Durante este sprint, su trabajo ha permitido mantener una visión global de integración entre los módulos desarrollados por el equipo —procesamiento de voz, semántica, cliente Godot y avatar 3D—, asegurando la compatibilidad conceptual y técnica de todos los componentes. Además, ha administrado el repositorio de evidencias del proyecto, donde se almacenan los documentos, versiones de los diagramas, archivos de código y minutas de seguimiento, centralizando así el conocimiento generado y su respaldo verificable.

Gracias a esta gestión integral, el proyecto cuenta actualmente con un andamiaje metodológico sólido y actualizado, que no solo sustenta la ejecución técnica, sino que también asegura la continuidad, coherencia y viabilidad del desarrollo a lo largo de las siguientes iteraciones del plan general.

# 3. Objetivos y Metodología Ajustada

El objetivo general del Sprint 2 es disponer de un pipeline operativo Whisper → (normalización/semántica) → glosa → animación en Godot, documentado y medible. Como ajustes metodológicos, se consolidó la práctica de “entregable documentado” (ninguna funcionalidad se cierra sin su respaldo en los artefactos del proyecto), se implementó validación cruzada entre integrantes y se añadieron mediciones ligeras de desempeño (latencia percibida y estabilidad de cola) para construir una línea base. Se mantiene Scrum como marco, con duración de 3 semanas por sprint y revisiones semanales de seguimiento.

# 4. Evidencias de Avance

La evidencia principal del desarrollo se encuentra en el repositorio oficial de GitHub del equipo CatchAI, donde residen los módulos de código, la historia de commits, ramas de trabajo y versiones de los componentes. Este control de versiones permite verificar la modularización del cliente Godot, la integración de Whisper y FastAPI, y la incorporación progresiva del modelo semántico basado en LLM y embeddings. Asimismo, el repositorio incluye la estructura del avatar 3D y los scripts de soporte al pipeline. Complementariamente, se dispone de capturas y renders de las versiones V2 y V3 del avatar, además de un video corto que muestra el flujo voz → texto → seña en ejecución. Los artefactos de gestión (plan de sprints, Kanban y borrador del DAS) acompañan y documentan estos avances.

Pantalla de juego de computadora

El contenido generado por IA puede ser incorrecto.

Para facilitar las evidencias se anexan los vínculos a las carpetas correspondientes:  
  
[Carpeta de Videos](https://drive.google.com/drive/folders/1rS9lFl0vvnJRBKexaKN-HyIipIDE4xX9)

[Carpeta de Evidencias](https://drive.google.com/drive/folders/1JoIYOskCfemKHLlKVTPI_pdcwUYnyycd?usp=drive_link)

[Github con Documentación y repositorio de código](https://github.com/eypino/CatchAI)

# 5. Análisis del Avance (respecto de planificación y Gantt)

En relación con la carta Gantt y la planificación inicial, el proyecto presenta los siguientes estados parciales al cierre de la semana evaluada: reconocimiento de voz y transcripción operativos; cliente Godot modular funcional con corrección de transiciones; pipeline general integrado con deletreo de respaldo; servicio local FastAPI disponible; modelo semántico en integración; y avatar humanoide funcional con expresividad en progreso. La documentación del DAS avanza con prioridad en vistas lógica y de procesos, pendiente de incorporar diagramas formales y contratos JSON en la siguiente iteración del sprint.

# 6. Riesgos, Restricciones y Mitigaciones

• Cómputo limitado para ejecución simultánea de Whisper, LLM y cliente Godot: se mitigará con batching de solicitudes, ajuste de ventanas de contexto, cacheo de embeddings y, cuando sea posible, uso de ejecución escalonada.  
• Cobertura acotada del diccionario LSCh (gatilla deletreo): se prioriza ampliación incremental enfocada en glosas de alta frecuencia.  
• Compatibilidad Blender–Godot (rig y exportación): se estandariza exportación GLB con pruebas de ida y vuelta por versión del avatar.  
• DAS en desarrollo: se agenda cierre de vistas clave (lógica, procesos, desarrollo) y anexos con diagramas para v0.7 antes de finalizar el sprint.

# 7. KPI e Indicadores de Desempeño

Con base en los requisitos no funcionales definidos previamente, los KPI de seguimiento para el MVP son:  
• Latencia extremo a extremo (voz → seña) en entorno local: objetivo ≤ 10 s para entradas de hasta 1 minuto; objetivo de operación típica 1–2 s en frases cortas.  
• Continuidad del pipeline en 10 solicitudes consecutivas sin caída, con recuperación de errores y retorno a idle (manejo de repetición de señas).  
• Cobertura del diccionario LSCh: porcentaje de glosas presentes vs. glosas solicitadas; fallback por deletreo cuando corresponda.  
• Documentación técnica: DAS v0.7 con vistas y diagramas, artefactos de sprint y Kanban actualizados antes del cierre del sprint.

# 8. Plan de Trabajo Inmediato (Semana 3 del Sprint 2)

• Integrar el módulo semántico (LLM + base vectorial) al flujo de producción, verificando coherencia lingüística y reducción del deletreo.  
• Consolidar la tabla central de enrutamiento glosa → animación y validar nomenclaturas con el diccionario.  
• Completar DAS v0.7 con vistas lógica/procesos/desarrollo, diagramas (clases, comunicación, secuencia) y contratos JSON.  
• Ampliar el diccionario LSCh con un lote incremental de glosas de alta frecuencia y preparar evidencia audiovisual del pipeline actualizado.  
• Mantener el repositorio GitHub con ramas por funcionalidad, PRs de revisión y etiquetado de versión del sprint.

# 9. Conclusión

A la fecha, el proyecto cuenta con un flujo técnico operativo Whisper → (normalización/semántica) → glosa → animación, un cliente Godot modular con transiciones corregidas, un servicio local FastAPI y un avatar humanoide funcional en evolución. La documentación de sprint y Kanban está al día y el DAS se encuentra en redacción con avance relevante en vistas clave. En esta etapa, el foco del equipo se orienta a cumplir los KPI establecidos para el MVP: latencia y estabilidad del pipeline, coherencia lingüística del módulo semántico, ampliación del diccionario LSCh y finalización del DAS.