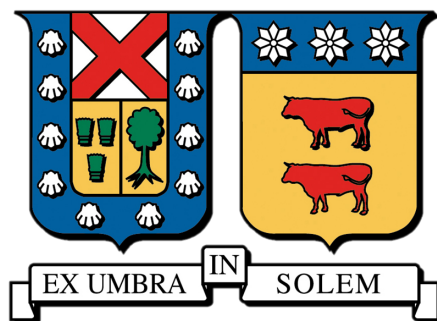


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
SANTIAGO - CHILE



“Modelo Acústico de Lenguaje Natural (NLP) para
Conversaciones Oncológicas de Salud”

Fabián Soto Morales Elías

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA:
PROFESOR CORREFERENTE:

Nicolas Jara
Matías Zañartu

Septiembre 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: _Modelo acústico de lenguaje natural para conversaciones oncológicas de salud

Nombre del candidato(a): _ Fabián Soto Elías Morales

Carrera / Grado: _ Ingeniería Civil Telemática

Campus: _ San Joaquín **Departamento:** _ Electrónica _

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, __ Nicolás Jara __, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: _ 15/10/2025 _ **Firma:** _____

Estudiante o Candidato(a):

Fecha: _ 15/10/2025 _ **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Agradezco a la Universidad Técnica Federico Santa María por la oportunidad de formarme en una carrera que me apasiona y por el entorno de aprendizaje que hizo posible este proyecto. A mis profesores, por su dedicación y por las herramientas entregadas durante estos años. En particular, a mi profesor guía, **Nicolás Jara**, por su orientación oportuna y apoyo constante, aun encontrándose en Valparaíso.

A mis compañeros y compañeras, gracias por la colaboración, la paciencia y el ánimo en cada etapa de este trabajo. En especial, a **Hsien-i Lee**, por su disposición permanente y el trabajo conjunto que permitió materializar esta plataforma.

Finalmente, a todas las personas que aportaron ideas, tiempo o críticas constructivas: su contribución fue clave para transformar este desafío académico en un aprendizaje significativo.

Resumen

En el ámbito de la salud, la comunicación entre médicos y pacientes juega un rol fundamental para el éxito de los tratamientos. Sin embargo, en la práctica clínica es común que los pacientes olviden una parte importante de las indicaciones recibidas durante la consulta. Factores como el uso de terminología técnica, el tiempo limitado de atención y el impacto emocional asociado al diagnóstico contribuyen a esta dificultad. Esta situación no solo afecta la comprensión del paciente, sino que también puede generar problemas de adherencia a los tratamientos y disminuir la calidad de vida.

Frente a este escenario, surge la necesidad de contar con herramientas tecnológicas que permitan registrar y procesar la información entregada durante la consulta médica, facilitando su acceso posterior en un formato claro y comprensible. La oportunidad de mejora consiste en desarrollar soluciones que automaticen este proceso, entregando información adaptada tanto para el paciente como para el equipo de salud.

Este trabajo propone el diseño e implementación de una plataforma web que captura, transcribe y procesa conversaciones médico–paciente mediante técnicas de procesamiento de lenguaje natural. La plataforma genera dos tipos de informes al finalizar cada consulta: uno para el paciente, redactado en un lenguaje simple y explicativo, y otro para el médico, en un formato técnico estandarizado con codificación CIE-10. La solución se construyó utilizando React para el frontend, Node.js para el backend, AssemblyAI para la transcripción de audio y modelos de OpenAI para el análisis y la generación de texto.

Las pruebas realizadas con datos simulados demuestran que la plataforma es capaz de producir informes personalizados de manera eficiente, reduciendo la pérdida de información, mejorando la comprensión de los pacientes y optimizando la documentación clínica. Este trabajo se desarrolló en colaboración con la Fundación Arturo López Pérez (FALP), institución interesada en explorar soluciones innovadoras para fortalecer la comunicación en el ámbito oncológico.

Keywords: React, Node.js, JavaScript, AssemblyAI, OpenAI, GPT-4, transcripción de audio, procesamiento de lenguaje natural, generación de informes médicos,

oncología.

Índice de figuras

2.1. actores principales, entradas y salidas de información.	14
2.2. componentes principales (frontend y backend), integraciones y flujos de datos.	15
2.3. Estructura JSON generada tras la creación del los informes	26
2.4. Ejemplo de datos ingresados completamente	28
2.5. Ejemplo de lo que entrega AssemblyAI	28
2.6. Ejemplo de texto procesado por OpenAI	29
2.7. Esquema general de los componente del prototipo	31
2.8. Pantalla de bienvenida	31
2.9. Pantalla de información del paciente: vista normal	32
2.10. Pantalla de información del paciente: campos incompletos	33
2.11. Pantalla de información del paciente: RUT incorrecto	33
2.12. Pantalla para iniciar la grabación	34
2.13. Pantalla de espera mientras se generan los informes	34
2.14. Pantalla de edición y validación de informes médicos	35
3.1. Diagrama de secuencia de grabación, transcripción y generación de informes	39

Índice de tablas

2.1. tabla de perfil de usuario	16
2.2. Módulos de la arquitectura del sistema	17
2.3. Rutas de la API utilizadas en el proyecto	27
3.1. Riesgos y Medidas de Mitigación	44

Índice general

1. Introducción	5
1.1. Hipótesis	7
1.2. Objetivos	7
1.2.1. Objetivo General	7
1.2.2. Objetivos Específicos	7
1.3. Estado del Arte	7
2. Desarrollo de la plataforma	10
2.1. Metodología	10
2.2. Requisitos del Sistema	11
2.2.1. Requisitos Funcionales	11
2.2.2. Requisitos No Funcionales	11
2.2.3. Requisitos de Ambiente	12
2.3. Tecnologías utilizadas en el proyecto	12
2.3.1. Frontend	12
2.3.2. Backend	12
2.3.3. Servicios externos	13
2.4. Diseño de Arquitectura del Sistema	14
2.4.1. Diagrama de Contexto	14
2.4.2. Diagrama de Arquitectura	14
2.4.3. Flujo de funcionamiento del sistema	15
2.4.4. Enumeración de Módulos	17
2.4.5. Enumeración de Módulos	17
2.5. Descripción de los Módulos	17

2.5.1.	Módulo de Gestión de Lógica de Negocios (Backend)	17
2.5.2.	Módulo de Interfaz de Usuario (Frontend)	19
2.5.3.	Módulo de Almacenamiento de Datos (EMR)	20
2.5.4.	Módulo de Transcripción de Audio (Servicio AssemblyAI)	21
2.5.5.	Módulo de Procesamiento de Lenguaje Natural (Servicio OpenAI GPT-4)	22
2.5.6.	Hitos Principales del Proyecto	24
2.5.7.	Revisión de Requerimientos	24
2.5.8.	Revisión del Diseño	24
2.5.9.	Entorno de Soporte y Desarrollo	25
2.6.	diseño del prototipo	25
2.6.1.	Contexto General del Prototipo	25
2.6.2.	Descripción de Componentes	25
2.6.3.	Funcionamiento del Sistema de Grabación y Generación de Informes	27
2.6.4.	Esquema general de los componente del prototipo	30
2.6.5.	Interfaces Implementadas en el Prototipo	31
3.	Resultados	36
3.1.	Verificación y Validación	36
3.1.1.	Verificación	36
3.1.2.	Validación	36
3.1.3.	Plan de Testing	37
3.1.4.	Análisis Crítico y Evaluación de Riesgos	37
3.1.5.	Diagrama de interacción	38
3.2.	Pruebas y resultados	39
3.2.1.	Pruebas y Audios de Simulación	39
	Pruebas y Validación	40
3.3.	Gestión de Riesgos	42
3.3.1.	Supuestos	42
3.3.2.	Dependencias	43

3.3.3. Restricciones	43
3.3.4. Riesgos	44
4. Conclusiones	45

Capítulo 1

Introducción

La adecuada comunicación entre los médicos y los pacientes repercute positivamente en el entendimiento del diagnóstico, en la adherencia al tratamiento y en la confianza generada hacia el sistema de salud. En la Medicina Oncológica, la duración limitada de la consulta y el contexto emocional en que se desarrolla hacen difícil la retención en el paciente de la información clínica, observándose que muchos pacientes olvidan la mayor parte de lo conversado, lo que dificulta la adherencia y la toma de decisiones compartidas.

Este problema es muy complicado, dado que el proceso de atención oncológica consiste en transmitir información muy sensible y técnica, como puede ser la comunicación de resultados de exámenes, diagnósticos, pronósticos o efectos secundarios de los tratamientos, etc. En la práctica, muchos de los pacientes no comprenden en profundidad ni llegan a interpretar correctamente aquellos mensajes del médico que pueden estar relacionados, por ejemplo, con el mismo diagnóstico, lo que provoca una sensación de ansiedad entre los pacientes o de confusión, e incluso, en algunos casos, abandono del tratamiento. Por ejemplo, en algunos estudios internacionales se ha constatado que más del 60

Sabemos que la comunicación médico paciente es uno de los aspectos del acto clínico, porque no consiste solamente en el hecho de comunicar información, sino de hacer llegar una comprensión. Un médico puede comunicar correctamente unas instrucciones, pero si el paciente no llega a comprenderlas o si no tiene un registro o un documento claro para revisitarlas posteriormente, se pierde el mensaje del

contenido clínico. En el contexto oncológico, donde los tratamientos suelen tener una larga duración y las decisiones médicas pueden tener implicaciones letales, tener un registro fiel y accesible de la consulta se vuelve básico e indispensable. Por otro lado, la carga emocional que puede conllevar un diagnóstico de cáncer puede tener influencias en la atención y memoria del paciente, lo que dificulta la capacidad para absorber la información clínica.

En este sentido, las nuevas tecnologías ofrecen una oportunidad concreta para la mejora de la adherencia y la comprensión acerca del tratamiento. De hecho, la incorporación de instrumentos de inteligencia artificial permite grabar, transcribir y procesar la información generada en la consulta, dándole forma de manera que pueda ser revisada posteriormente por el profesional médico o el paciente. Así, no solo se preserva la información técnica de la interacción, sino que además se favorece una interacción más empática, más clara y que puede ser mejor comprendida. Estas mismas herramientas, además, pueden contribuir a la reducción del peso administrativo de los profesionales médicos así como a la optimización de tiempo de atención a la clínica.

Con este telón de fondo se propone una solución alternativa que consiste en una plataforma que es capaz de grabar (en particular, de transcribir automáticamente) la conversación clínica y, mediante ciertos procesos de procesamiento de lenguaje natural, se ocupa de generar dos informes para enviar al médico y al paciente (i) el informe para el paciente, que tiene un carácter claro y accesible, que explica los términos técnicos de forma comprensible y (ii) el informe para el médico, de cómo están las cosas utilizando terminología técnica y con una estructura que cumple con los estándares de la práctica médica. La solución se basa en React y Node.js, incluye AssemblyAI para el trabajo de la transcripción y modelos de OpenAI para el análisis y la propia síntesis del texto. De este modo, la proposición ayuda a facilitar la comunicación entre los profesionales sanitarios y los pacientes oncológicos con vistas a mejorar la adherencia al tratamiento y la calidad de la atención médica.

1.1. Hipótesis

El uso combinado de transcripción automática y procesamiento de lenguaje natural puede mejorar la comunicación médico–paciente, al reducir la pérdida de información y facilitar el seguimiento clínico.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un modelo acústico de lenguaje natural aplicado al ámbito oncológico que permita capturar y procesar conversaciones médico–paciente, generando informes diferenciados en lenguaje técnico y comprensible para mejorar la comunicación y la continuidad del cuidado.

1.2.2. Objetivos Específicos

1. Implementar un sistema de captura y transcripción de audio en consultas oncológicas.
2. Diseñar un módulo de análisis de lenguaje natural para identificar roles (médico, paciente, acompañante).
3. Generar informes técnicos estandarizados mediante clasificación CIE-10.
4. Elaborar informes para pacientes en lenguaje claro, explicando términos complejos.
5. Desarrollar una plataforma web para la gestión y visualización de informes.
6. Validar la propuesta mediante pruebas de funcionalidad y claridad de informes.

1.3. Estado del Arte

La comprensión que tienen los pacientes sobre la información entregada por el médico durante una consulta es un factor crítico para garantizar la adherencia a los

tratamientos, el uso seguro de medicamentos y la participación activa en la toma de decisiones clínicas.

Un estudio realizado en el Hospital Universitario Federal de Juiz de Fora, con una muestra de 100 pacientes, evidenció que un 65 % presentó algún grado de dificultad para comprender las indicaciones médicas, y que casi un 50 % no entendía de manera clara los objetivos de los tratamientos propuestos. Estos resultados reflejan un riesgo directo en los resultados clínicos esperados [1].

De manera complementaria, [2] observaron que, si bien un 96 % de los pacientes entendía que un resultado “negativo” en una prueba de cáncer era favorable, solo un 67 % comprendía que la expresión “nódulos linfáticos positivos” implicaba diseminación tumoral. Este hallazgo refuerza la necesidad de transmitir la información en un lenguaje claro y accesible.

Asimismo, [3] reportaron que únicamente el 50 % de los pacientes hospitalizados presentaba una alfabetización en salud adecuada, y que aquellos con niveles bajos tenían tres veces más probabilidades de ser readmitidos en los primeros 90 días tras el alta. En la misma línea, [4] demostraron que la aplicación del método *teach-back* en servicios de urgencia mejoraba significativamente la comprensión de los pacientes respecto a las indicaciones médicas.

En el contexto nacional, [5] destacaron que la claridad y el acompañamiento en la comunicación constituyen ejes centrales del cuidado humanizado, subrayando la importancia de transmitir información médica en un lenguaje cercano, comprensible y adaptado a las necesidades del paciente.

La relevancia de este problema cobra mayor fuerza en oncología. La Organización Mundial de la Salud (OMS) proyecta un aumento sostenido en el número de diagnósticos de cáncer a nivel mundial en los próximos años, lo que enfatiza la urgencia de implementar estrategias que mejoren la comunicación médico-paciente y fortalezcan la adherencia a los tratamientos [6].

En este escenario, han comenzado a desarrollarse soluciones tecnológicas basadas en inteligencia artificial. Un ejemplo es la herramienta *AI Medical Transcription* de Markovate, la cual permite transcribir en tiempo real las conversaciones entre médico y paciente, interpretar el contexto clínico, generar resúmenes automáticos y ofrecer

sugerencias de codificación diagnóstica [7]. Estas funcionalidades buscan reducir la carga administrativa del médico y mejorar la claridad de la información comunicada. Sin embargo, hasta la fecha no existen estudios publicados que cuantifiquen el impacto de estas herramientas en la comprensión efectiva del paciente.

Capítulo 2

Desarrollo de la plataforma

2.1. Metodología

La metodología empleada en el desarrollo de la plataforma se estructuró en cinco pasos principales, los cuales permitieron avanzar de manera ordenada y progresiva hacia la obtención de los resultados. A continuación, se detallan cada uno de ellos:

1. **Levantamiento de requerimientos:** Se recopilaron y definieron los requisitos funcionales y no funcionales de la plataforma, considerando tanto la perspectiva del médico como la del paciente.
2. **Diseño de la solución:** Se elaboraron diagramas de arquitectura, flujo de funcionamiento y componentes principales de la aplicación, estableciendo la interacción entre el frontend, el backend y los servicios externos.
3. **Desarrollo e implementación:** Se construyeron los módulos de la plataforma utilizando tecnologías como React, Node.js, AssemblyAI y OpenAI, siguiendo un enfoque iterativo para asegurar la integración de cada componente.
4. **Validación preliminar:** Se realizaron pruebas técnicas utilizando audios simulados y casos clínicos reales, lo que permitió evaluar la funcionalidad de la plataforma, identificar limitaciones y proponer mejoras relacionadas con la transcripción, el manejo de síntomas y las condiciones acústicas.
5. **Conclusión y resultados:** Finalmente, se evaluó el desempeño de la solución en

relación con el objetivo inicial.

2.2. Requisitos del Sistema

2.2.1. Requisitos Funcionales

- **RF1:** El sistema debe iniciar la grabación cuando el doctor haga clic en el botón correspondiente en el sitio web.
- **RF2:** El modelo debe ser capaz de distinguir entre las voces de los usuarios y aislarlas del ruido ambiente, para evitar interferencias en la transcripción.
- **RF3:** Implementar correctamente el sistema de codificación CIE-10 para generar un informe dirigido al médico con un lenguaje técnico y formal.
- **RF4:** El sistema debe garantizar el envío seguro del audio grabado desde el componente web hacia el backend, para su posterior procesamiento.
- **RF5:** Al finalizar la grabación de la consulta, el sistema debe generar dos informes: uno dirigido al paciente y otro al médico, cada uno adaptado a su respectivo destinatario.

2.2.2. Requisitos No Funcionales

- **RNF1:** El sistema debe ser capaz de procesar y generar los informes médicos en un tiempo máximo de 5 minutos tras finalizar la consulta.
- **RNF2:** El informe generado debe estar libre de errores ortográficos.
- **RNF3:** La interfaz de usuario debe ser intuitiva y fácil de usar para los médicos, sin requerir formación técnica especializada.
- **RNF4:** El informe debe estar bien estructurado, destacando claramente la información más importante para facilitar su lectura y comprensión.

2.2.3. Requisitos de Ambiente

Hardware de Desarrollo

Se requerirá como hardware de desarrollo un computador capaz de soportar las distintas actividades que se irán realizando a medida que avanza el modelo. Adicionalmente, se recomienda contar con un micrófono de buena calidad para asegurar una captura de audio nítida y estable, dado que la calidad de entrada impacta directamente en la tasa de error de palabra de la transcripción.

2.3. Tecnologías utilizadas en el proyecto

Este trabajo integró diversas tecnologías de desarrollo web y servicios de inteligencia artificial, que se describen a continuación:

2.3.1. Frontend

El frontend del sistema fue desarrollado utilizando React, una biblioteca de JavaScript que permite construir interfaces web dinámicas y reactivas. Esta tecnología fue seleccionada porque actualmente es una de las más utilizadas a nivel mundial para el desarrollo de interfaces, lo que asegura una amplia comunidad de soporte y abundante documentación disponible. Asimismo, React ofrece un enfoque basado en componentes reutilizables, lo que contribuye a mantener el código ordenado, facilita la escalabilidad del sistema y reduce los tiempos de desarrollo. Gracias a estas características, React constituye una opción confiable, práctica y adecuada para aplicaciones en el ámbito clínico.

2.3.2. Backend

El backend del sistema fue desarrollado en Node.js, lo que permitió gestionar la lógica del servidor, procesar las solicitudes provenientes del frontend y comunicarse con los servicios externos como AssemblyAI y OpenAI. Node.js fue seleccionado porque ofrece un entorno ligero y eficiente para el manejo de múltiples conexiones simultáneas,

característica fundamental en este proyecto donde se trabaja con grabaciones de audio y envío de datos en tiempo real.

Otra razón relevante para su elección fue la compatibilidad con React, ya que ambos utilizan el lenguaje JavaScript, lo que facilita el desarrollo y reduce la complejidad al mantener un mismo ecosistema tecnológico. A diferencia de otros frameworks como PHP/Laravel o Django (Python), que también podrían haber sido utilizados, Node.js presenta una mayor flexibilidad para crear APIs rápidas y orientadas a la comunicación con servicios externos. Estas ventajas convierten a Node.js en una alternativa práctica y adecuada para implementar el backend de una aplicación clínica basada en inteligencia artificial.

2.3.3. Servicios externos

Para complementar el funcionamiento del sistema se integraron servicios externos basados en inteligencia artificial. La selección de estos servicios se realizó considerando criterios de precisión, facilidad de integración y relación costo-beneficio, aspectos fundamentales para un proyecto orientado al ámbito clínico.

- **AssemblyAI**: servicio de reconocimiento automático de voz (ASR) utilizado para transcribir las consultas médicas. Este servicio destaca por su capacidad de diarización, lo que permite separar las intervenciones de diferentes hablantes en una conversación. Según su documentación oficial, AssemblyAI soporta múltiples idiomas y ha sido utilizado en aplicaciones que requieren transcripción en tiempo real y con alta precisión
- **OpenAI GPT-4**: modelo de procesamiento de lenguaje natural empleado para analizar las transcripciones generadas por AssemblyAI. Su función principal en el proyecto fue identificar los roles de los hablantes (médico, paciente o acompañante), generar resúmenes diferenciados según el destinatario y extraer códigos **CIE-10** asociados a diagnósticos médicos. OpenAI GPT-4 es ampliamente reconocido por su capacidad de razonamiento y comprensión contextual, lo que lo convierte en una herramienta adecuada para tareas de clasificación y generación de informes clínicos

2.4. Diseño de Arquitectura del Sistema

En esta sección se describe cómo se relacionan los usuarios con la plataforma y cómo funciona internamente el sistema. Primero, se presenta el *diagrama de contexto*, que muestra los actores y los flujos de información de alto nivel. Luego, se expone el *diagrama de arquitectura*, donde se detallan los componentes principales (frontend y backend) y sus puntos de integración con servicios y datos.

2.4.1. Diagrama de Contexto

La figura 3.1 representa los actores principales del sistema y los flujos de información entre ellos. El médico es quien inicia la grabación de la consulta y el sistema procesa la conversación para elaborar dos informes: uno técnico, dirigido al profesional, y el otro en lenguaje claro dirigido al paciente. Este diagrama hizo posible determinar los límites funcionales del sistema. Es decir, qué información entra y qué información es entregada como resultado. La importancia de este aspecto se entiende porque se pudo orientar el diseño de los requerimientos y la interacción manejo de los usuarios y plataforma a partir de una visión más global.

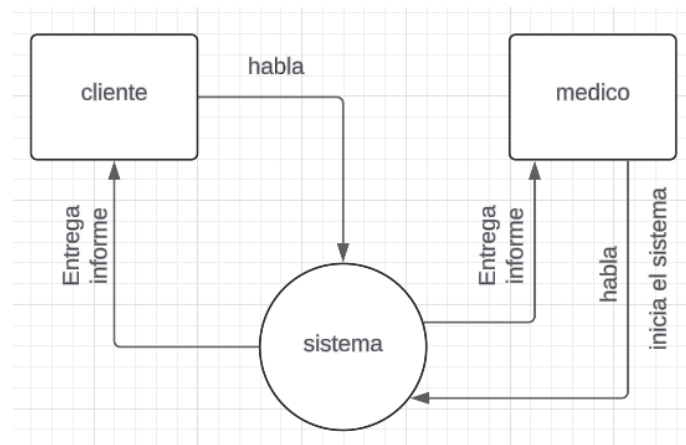


Figura 2.1: actores principales, entradas y salidas de información.

2.4.2. Diagrama de Arquitectura

La Figura 3.2 muestra la topología general del sistema, donde se incluyen los elementos de frontend (React), los de backend (Node.js) y los servicios ajenos

(AssemblyAI y OpenAI GPT-4o). Esta estructura fue generada a partir de la recogida de los requisitos técnicos con la intención de lograr que la comunicación entre los módulos y los servicios fuera eficiente, por lo que se decidió emplear una arquitectura basada en el modelo cliente-servidor con API REST, con el fin de poder garantizar la modularidad, la escalabilidad y la trazabilidad del flujo de información, desde la captura del audio hasta el artefacto de los informes clínicos.

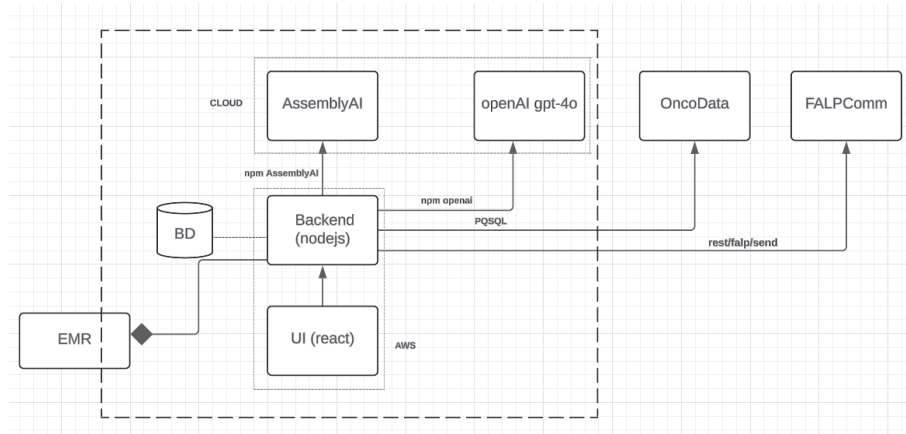


Figura 2.2: componentes principales (frontend y backend), integraciones y flujos de datos.

2.4.3. Flujo de funcionamiento del sistema

A continuación se mostrará cómo fluye la información dentro del sistema:

1. El médico inicia la grabación desde el frontend.
2. El audio se envía al backend, donde es almacenado temporalmente y reenviado a AssemblyAI.
3. AssemblyAI transcribe el audio y devuelve un diálogo segmentado por hablante.
4. La transcripción se envía a OpenAI GPT-4, que identifica los roles de cada participante (médico, paciente y acompañante). Este paso es fundamental, ya que se evita que comentarios o autodiagnósticos del paciente o acompañante se incluyan en el diagnóstico final.
5. Con el texto segmentado y limpio, OpenAI GPT-4 analiza el contenido e identifica los diagnósticos asociados, clasificándolos según la codificación CIE-10.

6. El sistema genera dos informes: uno en lenguaje claro dirigido al paciente y otro en formato técnico para el médico, ambos basados en la transcripción procesada y los códigos CIE-10.
7. Los informes se devuelven al frontend, donde el médico puede revisarlos, editarlos y validarlos antes de su registro oficial.
8. Una vez validados, el backend construye un archivo en formato JSON, que incluye los datos del paciente, el informe para el paciente, el informe para el médico y los códigos CIE-10.
9. Finalmente, este archivo se envía a través de un endpoint y se integra en la ficha clínica electrónica (EMR), asegurando que solo la información revisada y aprobada por el médico quede registrada.

Perfiles de Usuario

Perfil	Socioeconómico y cultural	Ocupacional	Etnia
Médico Oncológico	Trabajador de FALP	Médico con especialidad en oncología	25-65
Paciente	Persona común	Persona con una ocupación	10-10

Tabla 2.1: tabla de perfil de usuario

2.4.4. Enumeración de Módulos

2.4.5. Enumeración de Módulos

Módulo	Propósito	Sección
Módulo de Gestión de Lógica de Negocios (Backend)	Gestiona la lógica del negocio y la comunicación entre los módulos del sistema	§2.5.1
Módulo de Interfaz de Usuario (Frontend)	Proporciona una interfaz interactiva para médicos; se comunica con el backend para mostrar datos, visualizar informes y capturar entradas	§2.5.2
Módulo de Almacenamiento de Datos (EMR)	Almacena los datos y los informes personalizados generados	§2.5.3
Módulo de Transcripción de Audio (AssemblyAI)	Transcribe el audio de las consultas y lo segmenta por hablantes mediante la API de AssemblyAI	§2.5.4
Módulo de Procesamiento de Lenguaje Natural (OpenAI GPT-4o)	Ofrece capacidades de PLN: identificación de roles, resúmenes y propuesta de códigos CIE-10 mediante la API de OpenAI	§2.5.5

Tabla 2.2: Módulos de la arquitectura del sistema

2.5. Descripción de los Módulos

2.5.1. Módulo de Gestión de Lógica de Negocios (Backend)

■ **Propósito:**

- Centralizar y manejar la lógica de negocio dentro del sistema, asegurando que todas las operaciones de negocio se realicen de manera eficiente. Este módulo se comunica con distintas API, incluyendo AssemblyAI, OpenAI, la base de datos y otros servicios, garantizando coherencia y seguridad. Es responsable de procesar las solicitudes de los usuarios, aplicar las reglas del negocio y coordinar la comunicación entre los diferentes componentes del sistema.

■ **Alcance:**

- Recibir y procesar las solicitudes desde el frontend.

- Interactuar con el EMR para almacenar y recuperar datos clínicos.
- Coordinar la comunicación con servicios externos, como la transcripción de voz a texto (AssemblyAI) y el procesamiento de lenguaje natural (OpenAI).

■ **Dependencias:**

- Este módulo depende del frontend para recibir solicitudes y enviar respuestas. Además, depende de servicios externos para la conexión con APIs, permitiendo la generación adecuada de los informes médicos.

■ **Supuestos:**

- Se espera que los servicios externos (AssemblyAI, OpenAI) y el EMR respondan en tiempos adecuados para mantener la eficiencia del sistema.
- Se asume que el servidor que ejecuta Node.js cuenta con suficiente capacidad de procesamiento y memoria para manejar múltiples solicitudes concurrentes.
- Se requiere una conexión de red estable y rápida para la comunicación entre los módulos y los servicios externos.

■ **Restricciones:**

- Debe mantener tiempos de respuesta bajos para garantizar una experiencia de usuario fluida.
- Debe implementar medidas de seguridad para proteger los datos sensibles y evitar accesos no autorizados.

■ **Estructura General:**

• **Entradas:**

- Solicitudes del frontend para operaciones CRUD en el sistema de gestión de datos.
- Datos de audio de consultas médicas para transcripción.
- Solicitudes para generar informes personalizados.

- **Ejecución:**

- Recibir solicitud del frontend.
- Enviar datos al servicio de transcripción de voz a texto (AssemblyAI).
- Enviar el texto transcrito al servicio de procesamiento de lenguaje natural (OpenAI).
- Interactuar con el EMR para realizar operaciones CRUD.

- **Salidas:**

- Generación y entrega de dos informes personalizados, tanto para el médico como para el paciente, en formato JSON.

2.5.2. Módulo de Interfaz de Usuario (Frontend)

- **Propósito:**

- Proveer una interfaz interactiva para los médicos, permitiéndoles iniciar la grabación de consultas y generar informes personalizados.

- **Alcance:**

- El módulo de UI se encarga de la interacción directa con los usuarios. Aquí es donde el médico inicia la grabación y, una vez finalizada, se generan los informes personalizados.

- **Dependencias:**

- Depende del Módulo de Gestión de Lógica de Negocios para obtener y enviar datos.
- Depende del Módulo de Transcripción de Audio (Servicio AssemblyAI) para transcribir el audio a texto y del Módulo de Procesamiento de Lenguaje Natural (Servicio OpenAI gpt-4o) para generar informes personalizados tanto para el médico como para el paciente.

■ **Supuestos:**

- Se asume que los usuarios tendrán acceso a dispositivos con navegadores web modernos.
- Se asume que el backend y otros servicios estarán disponibles y funcionando correctamente.

■ **Restricciones:**

- Limitaciones en el tiempo de generación de los informes personalizados, debido a la dependencia de varios servicios externos como AssemblyAI y OpenAI.

■ **Estructura General:**

- **Entradas:** El médico hace clic en el botón de grabar, iniciando la grabación. La interfaz incluye cinco botones: grabar, pausar, terminar, anular y generar informe.
- **Salidas:** Visualización de los informes generados.

2.5.3. Módulo de Almacenamiento de Datos (EMR)

■ **Propósito:**

- El propósito de este módulo es gestionar de manera eficiente y segura el almacenamiento, recuperación y manipulación de todos los datos del sistema. Se almacenarán tanto los informes personalizados del paciente como los del médico, garantizando su accesibilidad para futuras consultas y análisis.

■ **Alcance:**

- Se encarga de almacenar y gestionar todos los datos necesarios para la operación del sistema, incluyendo las transcripciones de audio, resúmenes de consultas y cualquier otra información relevante.

■ **Dependencias:**

- Este módulo interactúa con el EMR, encargado de integrar los datos y asegurar que todo quede registrado.
- **Supuestos:**
 - Se espera que las consultas y operaciones en la base de datos se realicen dentro de tiempos aceptables para no afectar la eficiencia del sistema.
 - Se asume una conexión de red estable y rápida para la comunicación entre el servidor de la base de datos y otros módulos del sistema.
- **Restricciones:**
 - Debe mantener tiempos de respuesta bajos para garantizar una experiencia de usuario fluida.
 - Debe implementar medidas de seguridad para proteger los datos sensibles y evitar accesos no autorizados.
- **Estructura General:**
 - **Entradas:**
 - Solicitudes del backend para operaciones CRUD.
 - **Ejecución:**
 - El backend recibe las solicitudes del frontend y, una vez procesadas, los informes generados se almacenan en el EMR.
 - **Salidas:**
 - Entrega de datos para futuros tratamientos o análisis.

2.5.4. Módulo de Transcripción de Audio (Servicio AssemblyAI)

- **Propósito:**
 - Este módulo se encarga de convertir el audio de las consultas médicas en texto utilizando el servicio de AssemblyAI.

■ **Alcance:**

- El módulo de transcripción de audio toma las grabaciones de audio generadas durante las consultas médicas y las transcribe a texto. Este texto se utiliza posteriormente para generar informes médicos personalizados y mejorar la retención de información por parte del paciente.

■ **Dependencias:**

- Este módulo depende principalmente del backend, ya que es el backend el encargado de conectarse a los servicios externos. En este caso, utilizará la API de AssemblyAI para realizar la transcripción de audio.

■ **Supuestos:**

- Se asume que los archivos de audio proporcionados están en un formato compatible con AssemblyAI y que la conexión a internet es estable.

■ **Restricciones:**

- Las transcripciones pueden estar limitadas por la calidad del audio grabado, el ruido de fondo y la claridad del discurso. También existen limitaciones en términos de la capacidad de procesamiento del servicio AssemblyAI y el tiempo de respuesta.

■ **Estructura General:**

- **Entrada:** Archivo de audio de la consulta médica.
- **Ejecución:** Enviar el archivo de audio a AssemblyAI. Recibir la transcripción del audio en texto. Procesar y almacenar la transcripción.
- **Salida:** Texto transcrito del audio de la consulta médica.

2.5.5. Módulo de Procesamiento de Lenguaje Natural (Servicio OpenAI GPT-4)

■ **Propósito:**

- Este módulo se encarga de procesar las transcripciones de audio para generar resúmenes y respuestas coherentes utilizando el servicio de procesamiento de lenguaje natural GPT-4 de OpenAI.
- **Alcance:**
 - El módulo de procesamiento de lenguaje natural toma las transcripciones de audio y las analiza para generar informes médicos personalizados, resúmenes de consultas y respuestas a preguntas específicas basadas en el contenido de las consultas.
- **Dependencias:**
 - Este módulo depende del modulo del backend, ya que es el encargado de incluir la API de OpenAI GPT-4 para el procesamiento de lenguaje natural. También requiere el módulo de transcripción de audio, que proporciona el texto transcrito a procesar e incluir en la plataforma.
- **Supuestos:**
 - Se asume que las transcripciones de audio están en un formato adecuado para ser procesadas por GPT-4 y que la conexión a internet es estable.
- **Restricciones:**
 - Las capacidades de procesamiento están limitadas por la complejidad de las consultas médicas y la precisión de las transcripciones. También existen limitaciones en términos de la capacidad de procesamiento del servicio GPT-4 y el tiempo de respuesta.
- **Estructura General:**
 - **Entrada:** Texto transcrito de la consulta médica.
 - **Ejecución:** Enviar el texto transcrito a la API de GPT-4. Recibir el análisis y las respuestas generadas por GPT-4. Procesar y almacenar los resultados.
 - **Salida:** Informes médicos personalizados.

2.5.6. Hitos Principales del Proyecto

A continuación, se presentan los principales hitos que se han desarrollado y que están en desarrollo en el proyecto:

1. **Comprensión clara del desafío:** Identificación de los requerimientos clave tanto para médicos como para pacientes, con respecto a los informes médicos personalizados.
2. **Integración de AssemblyAI y OpenAI:** Implementación de los servicios de AssemblyAI para la transcripción de audio y de OpenAI para la identificación de roles y generación de informes.
3. **Desarrollo de prototipo funcional:** Creación del prototipo de la interfaz de usuario, con la posterior aprobación por parte del cliente.
4. **Validación con usuarios finales:** Realización de pruebas en entornos clínicos con médicos y pacientes, obteniendo retroalimentación para optimizar la plataforma.

2.5.7. Revisión de Requerimientos

Se revisaron los requisitos funcionales y no funcionales definidos en el proyecto y se evaluó si estos cumplen con los puntos establecidos. Como resultado, se ha observado que la mayoría de los requisitos funcionales han sido completados correctamente, lo que asegura el funcionamiento adecuado del sistema y su alineación con los objetivos del proyecto.

2.5.8. Revisión del Diseño

El diseño del prototipo ha cumplido con las expectativas del cliente, destacándose por su simplicidad y facilidad de uso. La interfaz ha sido desarrollada pensando en la experiencia del usuario, lo que permite a los médicos interactuar con la plataforma de manera intuitiva.

2.5.9. Entorno de Soporte y Desarrollo

El desarrollo del sistema ha sido dividido en dos áreas principales:

- **Frontend:** Desarrollado por Fabián Soto.
- **Backend:** Desarrollado por Hsien-i Lee y Fabián Soto.

2.6. diseño del prototipo

2.6.1. Contexto General del Prototipo

La aplicación está diseñada para generar dos informes personalizados: uno dirigido al médico y otro al paciente. El objetivo principal es que el paciente pueda comprender completamente su consulta médica, lo que le permitirá seguir su tratamiento de manera adecuada en el futuro. Estos informes personalizados están adaptados para garantizar que tanto el médico como el paciente reciban la información de una manera clara y relevante para sus necesidades específicas.

2.6.2. Descripción de Componentes

En la plataforma se identifican seis componentes principales, que se describen a continuación:

1. Iniciar la grabación: Este botón redirige al usuario a la sección donde puede iniciar la grabación de la consulta médica. Al presionarlo, se activa el proceso para comenzar a capturar el audio.
2. Información del paciente: En esta sección, el usuario debe ingresar los datos del paciente para su identificación y posterior búsqueda en el sistema de registros médicos electrónicos (EMR). Los campos presentes en este componente incluyen:
 - Nombre: Campo de texto donde se ingresa el nombre del paciente.
 - Apellido: Campo de texto donde se ingresa el apellido del paciente.
 - RUT: Campo de texto para el número de identificación del paciente (RUT).

- Botón Confirmar: Al presionar este botón, se validan los datos ingresados y se permite continuar a la siguiente etapa de grabación.
3. Grabar: Este componente se encarga de iniciar la grabación del audio de la consulta. Al presionar el botón “Grabar”, el sistema comienza a captar y almacenar la conversación entre el médico y el paciente.
 4. Pausar: Este botón permite pausar la grabación en cualquier momento. Es útil en caso de que se necesite un descanso en la conversación o una interrupción temporal.
 5. Terminar: Este botón finaliza la grabación y activa el proceso de generación de informes. Una vez que el audio ha sido capturado, se envía a AssemblyAI, que se encarga de segmentarlo y transcribirlo a texto. A continuación, el texto transcrito es procesado por OpenAI, que identifica los roles de los participantes y genera dos informes personalizados: uno para el médico y otro para el paciente. Los informes generados no se integran directamente en el EMR, sino que se presentan en un editor de texto dentro del frontend, donde el médico puede revisarlos, corregirlos y validarlos. Solo una vez confirmada la información, el sistema permite enviar el informe definitivo al EMR.
 6. Anular: Este botón cancela la grabación actual y permite reiniciar el proceso desde el principio. Al utilizarlo, cualquier audio capturado en la sesión actual es descartado.

Aquí, una vez que se haya completado la generación del informe, se creará un JSON con la siguiente estructura:

```
{
  "nombre": "Nombre del paciente",
  "apellido": "Apellido del paciente",
  "rut": "RUT del paciente",
  "informe_paciente": "Informe médico del paciente",
  "informe_medico": "Informe médico para el médico",
  "codigo_cie_10": "Código CIE-10 encontrado"
}
```

Figura 2.3: Estructura JSON generada tras la creación de los informes

@ 1 R L L L @				
Método HTTP	Ruta	Recibe	Entrega	Función
POST	/api/openai/roles	Texto generado por AssemblyAI	JSON con roles de hablantes	Identificar roles (doctor, paciente, acompañante)
POST	/api/cie/generarcie10	Diálogo con los roles identificados	JSON con códigos CIE-10	Generar códigos CIE-10
POST	/api/paciente/generar-informe	Diálogo con los roles identificados	Informe para el paciente	Generar informe personalizado para paciente
POST	/api/doctor/informe	Diálogo con los roles identificados	Informe para el médico	Generar informe técnico para el médico
POST	/api/transcripcion	Archivo de audio e información del paciente	JSON con datos del paciente, informes y clasificación CIE-10	Transcribir y etiquetar diálogos

Tabla 2.3: Rutas de la API utilizadas en el proyecto

2.6.3. Funcionamiento del Sistema de Grabación y Generación de Informes

El flujo de trabajo para la generación de informes médicos personalizados funciona de la siguiente manera:

1. Ingreso de Información del Paciente: Antes de comenzar la grabación, es necesario que el usuario complete toda la información del paciente en la sección correspondiente. Esto incluye ingresar el nombre, apellido y RUT del paciente. Solo después de completar esta sección, el sistema permitirá avanzar a la sección de grabación.

Información del paciente

Nombre :

Hsien

Apellido :

Lee

Rut:

21.326.227-0

Confirmar

Figura 2.4: Ejemplo de datos ingresados completamente

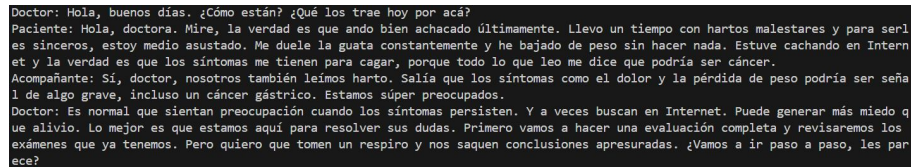
- Grabación del Audio: El proceso continúa con la grabación de la consulta médica, la cual es iniciada y finalizada por el médico.
- Envío a AssemblyAI: Una vez finalizada la grabación, el audio se envía a AssemblyAI. Esta API es la encargada de segmentar el audio y transcribirlo de audio a texto.

```
B: Hola, buenos días. ¿Cómo están? ¿Qué los trae hoy por acá?  
A: Hola, doctora. Mire, la verdad es que ando bien achacado últimamente. Llevo un tiempo con hartos malestares y para serles sinceros, estoy medio asustado. Me duele la guata constantemente y he bajado de peso sin hacer nada. Estuve cachando en Internet y la verdad es que los síntomas me tienen para cagar, porque todo lo que leo me dice que podría ser cáncer.  
C: Sí, doctor, nosotros también leímos hartos. Salía que los síntomas como el dolor y la pérdida de peso podría ser señal de algo grave, incluso un cáncer gástrico. Estamos súper preocupados.  
B: Es normal que sientan preocupación cuando los síntomas persisten. Y a veces buscan en Internet. Puede generar más miedo que alivio. Lo mejor es que estamos aquí para resolver sus dudas. Primero vamos a hacer una evaluación completa y revisaremos los exámenes que ya tenemos. Pero quiero que tomen un respiro y nos saquen conclusiones apresuradas. ¿Vamos a ir paso a paso, les parece?
```

Figura 2.5: Ejemplo de lo que entrega AssemblyAI

- Identificación de Roles en OpenAI: Después de la transcripción, el texto segmentado es enviado a OpenAI. En esta etapa, OpenAI identifica los roles de los distintos hablantes (médico, paciente y acompañantes). Esto es fundamental para asegurar que, al generar el informe, se excluya cualquier información que no debería

estar presente. Por ejemplo, el sistema filtra comentarios de autodiagnóstico o suposiciones hechas por el paciente sobre su estado de salud, que pueden no ser precisas ni útiles para el informe médico. De esta manera, el informe final se centra únicamente en la información proporcionada por el médico, asegurando que refleje un diagnóstico y tratamiento profesionales y precisos.



Doctor: Hola, buenos días. ¿Cómo están? ¿Qué los trae hoy por acá?
Paciente: Hola, doctora. Mire, la verdad es que ando bien achacado últimamente. Llevo un tiempo con hartos malestares y para serl
es sinceros, estoy medio asustado. Me duele la gasta constantemente y he bajado de peso sin hacer nada. Estuve cachando en Intern
et y la verdad es que los síntomas me tienen para cagar, porque todo lo que leo me dice que podría ser cáncer.
Acompañante: Si, doctor, nosotros también leímos harto. Salía que los síntomas como el dolor y la pérdida de peso podría ser seña
l de algo grave, incluso un cáncer gástrico. Estamos súper preocupados.
Doctor: Es normal que sientan preocupación cuando los síntomas persisten. Y a veces buscan en Internet. Puede generar más miedo q
ue alivio. Lo mejor es que estamos aquí para resolver sus dudas. Primero vamos a hacer una evaluación completa y revisaremos los
exámenes que ya tenemos. Pero quiero que tomen un respiro y nos saquen conclusiones apresuradas. ¿Vamos a ir paso a paso, les par
ece?

Figura 2.6: Ejemplo de texto procesado por OpenAI

5. Identificación de Códigos CIE-10: Tras identificar los roles, el texto procesado se envía nuevamente a OpenAI para extraer los códigos CIE-10 asociados con el diagnóstico. OpenAI identifica los términos médicos y los clasifica con los códigos CIE-10 correspondientes, según la Clasificación Internacional de Enfermedades, para asegurar un registro estandarizado y facilitar futuras consultas.
6. Generación del Informe Personalizado: Una vez identificados los roles de los hablantes, el sistema realiza dos solicitudes adicionales a OpenAI para generar los informes: una solicitud para el informe del paciente y otra para el informe del médico.
 - a) Informe del Paciente: Se entrega en un formato claro y sencillo, evitando el uso de terminología técnica compleja. En caso de que sea necesario, se incluyen explicaciones breves para garantizar la comprensión. El sistema también filtra comentarios no relevantes del paciente, como intentos de autodiagnóstico.
 - b) Informe del Médico: Se presenta en un formato más técnico y conciso, enfocado en diagnóstico y tratamiento, con un lenguaje formal y profesional.
7. Revisión y Validación en el Editor de Texto: Los informes generados no se envían directamente al sistema de registros clínicos. En cambio, se presentan primero en un editor de texto dentro del frontend. Allí, el médico puede revisar, corregir

y validar la información generada. Este paso asegura que solo la información verificada quede registrada oficialmente.

8. Inyección de Datos en el EMR: Una vez que los informes han sido revisados y validados en el editor de texto, el sistema crea un archivo en formato JSON con los siguientes campos esenciales:

- Nombre del paciente
- Apellido del paciente
- RUT del paciente
- Informe para el paciente
- Informe para el médico
- Código CIE-10

Este archivo JSON es enviado a través de un endpoint y finalmente integrado en el sistema de registros médicos electrónicos (EMR), garantizando que solo la información revisada y validada por el médico quede registrada de manera oficial y segura.

2.6.4. Esquema general de los componente del prototipo

El siguiente esquema se observa la interacción entre los principales componentes del prototipo desarrollado para la grabación y generación de informes médicos personalizados. En este flujo, se destacan tres áreas principales: el frontend, el backend, y los servicios externos (AssemblyAI y OpenAI) que facilitan la transcripción y análisis de los datos.

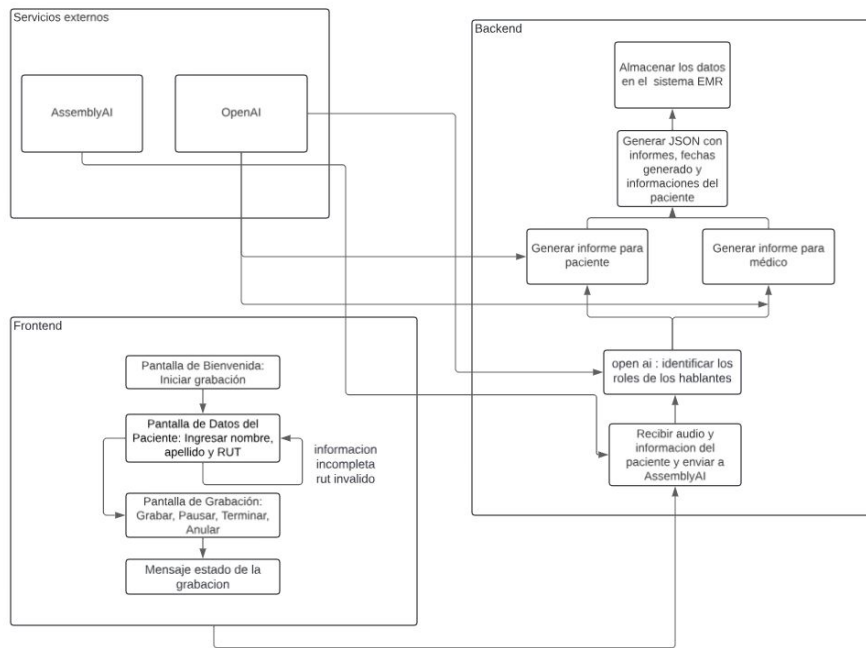


Figura 2.7: Esquema general de los componente del prototipo

2.6.5. Interfaces Implementadas en el Prototipo

Se ha implementado una interfaz de usuario intuitiva y diseñada para ser lo más accesible posible. A continuación, se describen las interfaces principales desarrolladas en la plataforma:

Pantalla de Bienvenida: Al iniciar la aplicación, el usuario es recibido con una pantalla que incluye el logotipo de la institución y un botón para comenzar la grabación de la consulta.



Figura 2.8: Pantalla de bienvenida

Pantalla de Información del Paciente: Tras seleccionar el botón “Iniciar grabación” en la pantalla de bienvenida, el usuario es redirigido a una página donde debe ingresar los datos del paciente (nombre, apellido y RUT). Esta información es esencial para la correcta identificación del paciente en el sistema de registros médicos electrónicos (EMR) y para futuras consultas. Para avanzar al proceso de grabación, todos los campos deben ser completados correctamente. Existen diferentes variaciones de esta pantalla:

- **Vista normal:** Pantalla estándar en la que el usuario introduce los datos del paciente sin errores ni advertencias.

The image shows a web browser window with a dark blue header. The main content area has a light blue background. In the center, there is a white rectangular form titled "Información del paciente". Inside the form, there are three input fields labeled "Nombre:", "Apellido:", and "Rut:". The "Rut:" field has a placeholder text "xxxxxxxx-x". Below the input fields is a green button labeled "Confirmar".

Figura 2.9: Pantalla de información del paciente: vista normal

- **Campos incompletos:** Si el usuario intenta avanzar sin completar todos los campos, se despliega un mensaje de advertencia en color rojo solicitando que se ingresen todos los datos obligatorios.

The screenshot shows a web application interface with a dark blue header containing a logo. The main content area is light blue and contains a white form titled "Información del paciente". The form has three input fields: "Nombre" with the value "Fabian", "Apellido" with the value "Soto", and "Rut" with a masked value "XXXXXXXX-X". Below the fields is a red error message "Por favor, complete todos los campos." and a green "Confirmar" button.

Figura 2.10: Pantalla de información del paciente: campos incompletos

- **RUT incorrecto:** En caso de que el RUT ingresado no sea válido, se muestra un mensaje de error en rojo indicando la corrección necesaria antes de continuar.

The screenshot shows the same "Información del paciente" form as in Figure 2.10, but with the "Rut" field containing the value "231". Below the fields is a red error message "El RUT ingresado es incorrecto." and a green "Confirmar" button.

Figura 2.11: Pantalla de información del paciente: RUT incorrecto

Pantalla de Grabación: Una vez confirmados los datos del paciente, el sistema muestra la interfaz de grabación. En ella se presentan cuatro botones principales: “Grabar”, “Pausar”, “Terminar” y “Anular”. El médico puede iniciar la grabación con el botón “Grabar”, pausarla en cualquier momento o cancelarla con “Anular”. Al presionar “Terminar”, se finaliza la grabación y el sistema comienza con el procesamiento del audio.

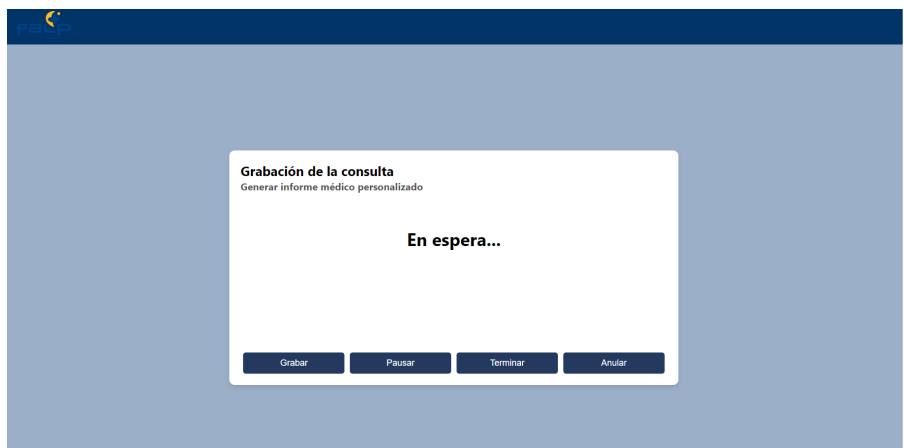


Figura 2.12: Pantalla para iniciar la grabación

Pantalla de Espera: Tras finalizar la grabación, el sistema procesa el audio y genera los informes médicos preliminares. Durante este tiempo, se muestra una pantalla de espera que informa al usuario que el sistema está trabajando en la generación de los resultados.



Figura 2.13: Pantalla de espera mientras se generan los informes

Pantalla de Edición y Validación de Informes: Una vez generados los informes (para el paciente y para el médico), estos se muestran en un editor de texto dentro de la plataforma. El médico tiene la posibilidad de revisar, modificar y validar la información antes de confirmar su envío. Solo después de esta validación, el informe definitivo es enviado al sistema de registros médicos electrónicos (EMR).

Resultados de la consulta

Datos del paciente

Nombre: Islen, Apellido: lee, RUT: 21526220

Informe para el paciente

Informe Médico

1. Síntomas: Mas experimentado dolor constante en el estómago, pérdida de peso sin motivo aparente, sensación de hiena sin haber comido mucho, vómitos y la sensación de que algo se sube por la garganta.

2. Diagnóstico: Se ha detectado una masa en la región del estómago a través de una ecografía.

Informe para el médico

1. Síntomas: Dolor abdominal constante, pérdida de peso involuntaria, sensación de plenitud gástrica, vómitos, regurgitación.

2. Diagnóstico: En estudio. Masa en la región del estómago detectada en ecografía abdominal.

3. Tratamiento: Tratamiento inicial. Ninguno especificado.

Examen complementario: Ecografía con biopsia transrectal consultada.

Guardar en EMR, Volver

Figura 2.14: Pantalla de edición y validación de informes médicos

Capítulo 3

Resultados

3.1. Verificación y Validación

3.1.1. Verificación

Se realizarán pruebas exhaustivas en los componentes clave del frontend y backend para garantizar su correcto funcionamiento. Se verificará que los botones de grabación y la interfaz de usuario operen correctamente y que el backend se comunique adecuadamente con los servicios externos, como AssemblyAI y OpenAI. Además, se realizarán pruebas con diferentes audios para asegurar que estos servicios puedan identificar correctamente a los interlocutores y generar informes precisos.

3.1.2. Validación

Para la validación preliminar se buscaron casos clínicos reales como referencia y, a partir de ellos, se generaron audios simulados que permitieron poner a prueba el funcionamiento de la plataforma en condiciones controladas. Estos audios fueron diseñados con el fin de reproducir de manera aproximada la dinámica de una consulta oncológica, lo que posibilitó evaluar la integración de los módulos y la generación de informes diferenciados para médico y paciente.

No obstante, esta validación corresponde a una etapa de laboratorio y no constituye aún una validación funcional en un entorno clínico real. Para alcanzar este objetivo será necesario realizar pruebas con un número significativo de audios reales

(idealmente más de 100) y con la participación de médicos especialistas, de modo de garantizar que los resultados obtenidos reflejen el comportamiento esperado en la práctica médica cotidiana.

3.1.3. Plan de Testing

En este paso, lo que haremos es asegurar que el sistema funcione correctamente y de manera integrada. Se evaluarán tanto el frontend como el backend, así como los servicios externos.

Las pruebas abarcarán varios aspectos. Primero, en el frontend, se verificarán los botones de grabación (iniciar, pausar, detener, cancelar) y la interfaz de usuario para garantizar que funcionen adecuadamente. Luego, en el backend, se evaluará que los endpoints se comuniquen correctamente con los servicios externos como AssemblyAI y OpenAI. Utilizaremos audios que simulan distintos niveles de ruido y diferentes interlocutores, como el médico, paciente y acompañantes, para probar la capacidad de los servicios de generar informes médicos precisos.

Realizaremos distintos tipos de pruebas, comenzando por pruebas unitarias, que verificarán que cada componente del frontend funcione de manera independiente. Después, realizaremos pruebas de integración para garantizar que el frontend y backend se comuniquen sin problemas. También realizaremos pruebas de rendimiento para medir cómo responde el sistema ante diferentes volúmenes de datos y complejidades en los audios. Finalmente, se llevarán a cabo pruebas de usuario, donde se simularán consultas médicas reales para evaluar la experiencia del usuario y la precisión de la generación de informes. En estas pruebas, los audios incluirán variaciones de ruido y cantidad de hablantes, y ajustaremos las instrucciones proporcionadas a los servicios externos para mejorar la precisión de los informes.

3.1.4. Análisis Crítico y Evaluación de Riesgos

Durante el desarrollo del proyecto, se han identificado varios riesgos que podrían afectar la precisión en la generación de informes, esto se debe a que el proceso depende en gran medida del rendimiento de AssemblyAI para la transcripción

y procesamiento de audio. Aunque AssemblyAI ha demostrado ser preciso en la identificación y distinción de voces, se han observado algunas limitaciones durante las pruebas.

Uno de los riesgos principales es la calidad del micrófono utilizado por los usuarios, en este caso, el médico. Si el dispositivo de grabación no es de buena calidad o el audio grabado presenta algún problema o inconveniente, la precisión de la transcripción puede verse comprometida, lo que afectaría directamente la exactitud del informe generado.

Otro factor de riesgo importante es la presencia de ruido ambiente o interferencias durante la grabación de la consulta. Aunque AssemblyAI tiene un buen rendimiento en situaciones controladas, en escenarios con ruido de fondo o conversaciones superpuestas, pueden producirse errores en la identificación de los hablantes. Esto podría dar lugar a transcripciones incompletas o mal interpretadas, lo que comprometería la calidad del informe.

Para mitigar estos riesgos, es esencial que se utilicen dispositivos de grabación de alta calidad y que el entorno de grabación sea lo más controlado posible, reduciendo el ruido y las interferencias. Además, será fundamental realizar pruebas exhaustivas en distintos entornos para ajustar el sistema según las condiciones reales de uso.

En cuanto a OpenAI, aunque es una herramienta potente para el procesamiento de lenguaje natural, su precisión en la generación de informes dependerá de la claridad y precisión de las instrucciones proporcionadas. Es crucial orientar adecuadamente a OpenAI para que el informe resultante sea lo más preciso posible.

Para mejorar aún más la precisión de los informes, será necesario trabajar con una variedad de audios representativos de diferentes escenarios clínicos, permitiendo así ajustar el sistema de acuerdo con los resultados obtenidos en cada caso. Esto permitirá optimizar las instrucciones y garantizar que los informes generados cumplan con las expectativas tanto del médico como del paciente, siendo coherentes y precisos.

3.1.5. Diagrama de interacción

La figura presenta la secuencia de interacciones que llevan a cabo los principales componentes del sistema durante el proceso de grabación, transcripción y generación

del informe médico. El flujo de ejecución se inicia cuando el médico presiona el botón “Iniciar grabación” desde la interfaz del frontend, de modo que se comienza la captura del audio, el cual se envía al backend. Este, a su vez, remite el archivo a la API de AssemblyAI, encargada de realizar la transcripción automática del contenido del audio.

A continuación, el texto resultante de la transcripción es procesado por OpenAI, donde se genera un informe estructurado que resume la conversación médico-paciente. Finalmente, el backend devuelve el resultado al frontend, donde es visualizado por el usuario.

Este flujo de ejecución representa la coordinación entre los distintos módulos del sistema y el uso conjunto de servicios de inteligencia artificial que permiten transformar audio clínico en información estructurada.

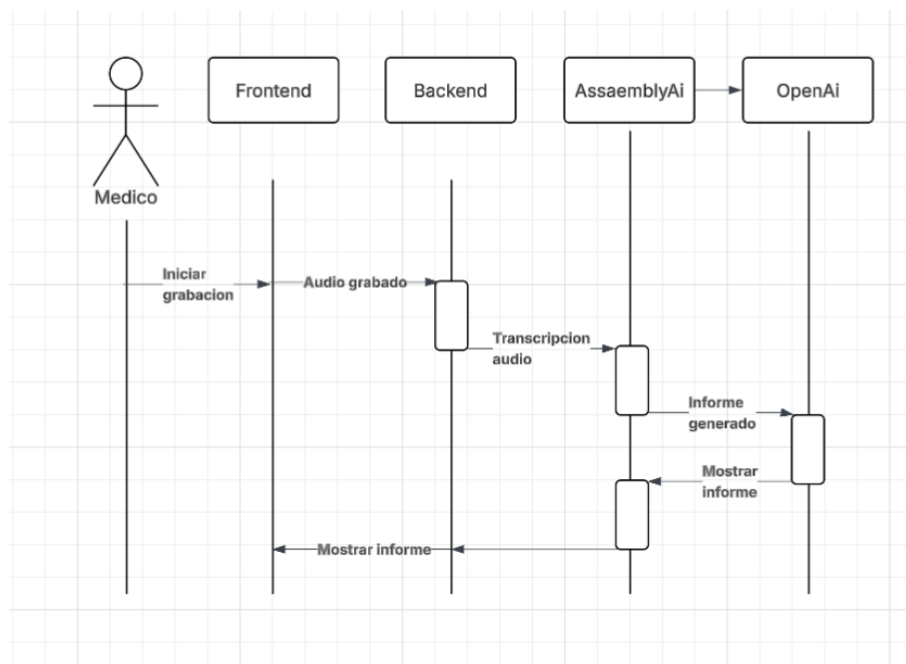


Figura 3.1: Diagrama de secuencia de grabación, transcripción y generación de informes

3.2. Pruebas y resultados

3.2.1. Pruebas y Audios de Simulación

Pruebas y Validación

El proceso de validación de la plataforma se dividió en dos etapas complementarias. La primera correspondió a una fase preliminar de carácter técnico, cuyo objetivo fue evaluar el desempeño del sistema en distintos escenarios de audio utilizando casos simulados. La segunda etapa se orientó a la exploración de casos clínicos reales en oncología, con el fin de contrastar los informes generados por la plataforma con los informes médicos de referencia.

En la fase preliminar se generaron un total de quince audios simulados, diseñados para emular diferentes contextos de consultas médicas. Estos audios permitieron analizar el comportamiento de AssemblyAI frente a situaciones con distintos números de hablantes, condiciones acústicas y variaciones en la pronunciación. Entre todos los casos, los más representativos fueron los siguientes:

- **Audio 1:** Conversación entre un paciente y un médico. El sistema segmentó correctamente la interacción, identificó a los hablantes y generó los informes sin inconvenientes.
- **Audio 2:** Conversación entre un paciente, un médico y un acompañante. AssemblyAI logró diferenciar adecuadamente a cada participante, manteniendo la coherencia en los informes generados.
- **Audio 3:** Conversación entre un paciente, un médico y dos acompañantes. Si bien el sistema produjo un informe completo y claro, presentó dificultades menores al identificar de forma precisa al segundo acompañante, sin que esto afectara de manera crítica la calidad del resultado.
- **Audio 4:** Conversación entre tres personas con ruido de fondo. El informe final se

generó correctamente, aunque algunas palabras fueron omitidas debido al ruido, lo que evidenció la sensibilidad del sistema a condiciones acústicas adversas.

- **Audio 5:** Conversación entre dos personas con ruidos fuertes e inesperados. En este caso, se observaron problemas de segmentación y errores de transcripción, lo que impactó directamente en la fidelidad del informe generado.

Una de las principales debilidades observadas en el uso de AssemblyAI corresponde a las dificultades presentadas cuando los pacientes emplean un lenguaje excesivamente coloquial, se utilizan modismos propios de la región o no articulan correctamente ciertas palabras. Estas características, frecuentes en un contexto clínico real, pueden afectar la calidad de la transcripción y, en consecuencia, la fidelidad del informe generado.

Considerando las limitaciones actuales de la tecnología de reconocimiento de voz, la estrategia más eficiente en este escenario no consiste únicamente en depender del modelo, sino en complementar el proceso con una participación activa del médico. En particular, se propone que el profesional repita o reformule, de manera clara y estructurada, las expresiones más relevantes mencionadas por el paciente. De esta forma, se logra preservar la esencia de lo expresado, pero asegurando que quede registrado en un formato comprensible y clínicamente útil, potenciando así la exactitud de los informes generados.

La segunda etapa estuvo orientada a la validación con casos clínicos reales de oncología. Para ello, se seleccionaron tres casos reales a partir de los cuales se generaron diez audios, que luego fueron procesados por la plataforma. Los resultados obtenidos fueron comparados con los informes médicos oficiales asociados a dichos casos. De esta validación se pudo observar que los informes dirigidos a los pacientes mostraron un desempeño muy positivo: la información entregada por la plataforma coincidía con la contenida en los informes médicos reales, especialmente en lo relativo a diagnóstico y tratamiento, destacando además la claridad con la que se explicaban los conceptos clínicos.

No obstante, en los informes dirigidos al médico se identificó una limitación puntual en la sección de síntomas. Mientras que el sistema tendía a describir los

síntomas de manera general, los informes médicos reales presentaban un nivel de detalle más específico y técnico. Este desfase no afectó la precisión del diagnóstico ni del tratamiento, pero evidenció la necesidad de ajustar los criterios de extracción y síntesis de síntomas para que los informes médicos generados reflejen con mayor fidelidad la especificidad clínica esperada.

En términos generales, los resultados confirman que la plataforma cumple con su propósito central: constituirse como una herramienta de apoyo para médicos y pacientes. Los informes generados demostraron ser consistentes en los aspectos esenciales de diagnóstico y tratamiento, y aunque se requiere mejorar el manejo de la sección de síntomas en los informes médicos, la validación realizada respalda el potencial del sistema para integrarse como complemento en la práctica clínica, quedando pendiente su validación definitiva en un entorno hospitalario con un mayor número de casos reales.

3.3. Gestión de Riesgos

3.3.1. Supuestos

1. **Feedback del cliente:** Se recibirá retroalimentación del cliente durante cada ciclo de desarrollo, lo que asegurará que el proyecto esté alineado con las expectativas del usuario.
2. **Calidad de audio:** Se asume que los micrófonos utilizados por los usuarios (médicos) serán de alta calidad, lo que garantizará que el audio capturado sea lo suficientemente claro para permitir transcripciones precisas a través del servicio AssemblyAI.
3. **Acceso continuo a Internet:** Se contará con acceso a internet estable y continuo para permitir la comunicación eficiente entre los módulos del sistema.
4. **Compatibilidad de dispositivos:** Los dispositivos utilizados por los médicos serán compatibles con el sistema y podrán ejecutar aplicaciones web sin problemas.

3.3.2. Dependencias

1. **Conexión a Internet Estable:** Es imprescindible contar con una conexión a internet continua y confiable en los lugares donde se utilizará el sistema, ya que la generación y envío de informes dependen de este acceso.
2. **Adquisición de Licencias y Derechos de Software:** Es necesario adquirir las licencias correspondientes para el uso de tecnologías y herramientas requeridas, incluidas las APIs externas de AssemblyAI y OpenAI.

3.3.3. Restricciones

1. **Dependencia de servicios externos:** El sistema depende de la disponibilidad y estabilidad de servicios externos como AssemblyAI y OpenAI. Cualquier interrupción o baja disponibilidad de estos servicios puede afectar el funcionamiento y la capacidad del sistema para generar informes de manera oportuna.
2. **Limitaciones en la transcripción en tiempo real:** La transcripción de audios y la generación de informes en tiempo real pueden verse limitadas por el tiempo de respuesta de servicios externos como AssemblyAI y OpenAI, así como por la capacidad de procesamiento del propio sistema, lo que podría afectar la rapidez de generación de los informes en situaciones de alta demanda.
3. **Calidad del entorno de grabación:** La calidad del audio puede verse afectada por el entorno en el que se realiza la grabación, como en consultas ruidosas o con interferencias. Esto puede dificultar la precisión de la transcripción y la correcta identificación de los hablantes.

3.3.4. Riesgos

Riesgo	Medida de Mitigación
Recursos limitados (servicios externos)	Considerar el uso de servicios alternativos en caso de que se excedan los límites de solicitudes a OpenAI o AssemblyAI.
Dependencia de servicios externos (AssemblyAI/OpenAI)	Mantener un plan de contingencia que incluya proveedores alternativos. Establecer un sistema que notifique al usuario en caso de fallas en estos servicios y permita reintentar el procesamiento.

Tabla 3.1: Riesgos y Medidas de Mitigación

Capítulo 4

Conclusiones

El presente proyecto consistió en el desarrollo de una plataforma orientada a mejorar la comunicación médico–paciente en el ámbito oncológico, a través de la grabación y transcripción automática de las consultas médicas, utilizando servicios de inteligencia artificial para procesar el contenido y generar informes diferenciados: uno técnico, dirigido al médico, y otro en lenguaje claro, pensado para el paciente. De esta forma, se buscó abordar una de las principales dificultades en este tipo de atenciones: la baja retención y comprensión de la información clínica por parte de los pacientes y sus familias.

Entre los principales logros alcanzados se encuentra la integración exitosa de tecnologías como *AssemblyAI* y *OpenAI GPT-4* en un flujo completo que permite capturar audio, transcribirlo, identificar los roles de los participantes, extraer diagnósticos con clasificación CIE-10 y finalmente generar informes personalizados. Además, se desarrolló un prototipo funcional con una interfaz amigable que demuestra la factibilidad técnica de la propuesta, validando la coherencia del sistema y su capacidad de generar información clara y diferenciada para cada usuario.

Sin embargo, es importante señalar que esta validación corresponde a una etapa preliminar, realizada en condiciones de laboratorio mediante audios simulados basados en casos clínicos reales. Esto implica que no se trata aún de una validación funcional en un entorno clínico real, lo que constituye la principal limitación del presente trabajo. Asimismo, factores como la calidad del audio, la dependencia de servicios externos y el tiempo de respuesta de las APIs representan desafíos adicionales que

deben considerarse en futuras implementaciones.

En relación con la hipótesis planteada, los resultados obtenidos permiten afirmar que el uso combinado de transcripción automática y procesamiento de lenguaje natural efectivamente contribuye a mejorar la comunicación médico-paciente, al reducir la pérdida de información y favorecer una mejor comprensión de los contenidos clínicos. No obstante, su eficacia plena dependerá de la validación en contextos reales de atención médica.

Para las próximas etapas, se espera realizar pruebas en entornos clínicos reales con un número mayor de audios y con la participación de especialistas oncológicos, con el fin de confirmar la utilidad, precisión y usabilidad de la plataforma en la práctica médica. También se planea optimizar el sistema para reducir los tiempos de procesamiento, reforzar la seguridad en el manejo de datos sensibles y ampliar sus capacidades a otras áreas de la medicina, de modo que se convierta en una herramienta útil y efectiva dentro de la salud digital.

Bibliografía

- [1] M. J. Silva et al. Evaluación de la comprensión del paciente sobre la información médica en un hospital universitario. *Enfermería Clínica*, 26(4):215–221, 2016.
- [2] M. Gotlieb et al. Patient comprehension of common medical phrases: A survey of 215 adults. *JAMA Network Open*, 5(12):e2247802, 2022.
- [3] M. Shahid et al. Health literacy and readmission risk among hospitalized patients: A prospective cohort study. *BMC Health Services Research*, 22:1248, 2022.
- [4] R. T. Griffey et al. Teach-back method improves patient understanding in the emergency department: A randomized controlled trial. *International Journal of Emergency Medicine*, 8:9, 2015.
- [5] M. Hermosilla Ávila et al. Construcción y validación de un instrumento para valorar el cuidado humanizado en enfermería en Chile. *Enfermería Global*, 15(3):344–358, 2016.
- [6] Organización Mundial de la Salud. World health organization cancer fact sheet, 2020. Accedido: 2025-09-28.
- [7] Markovate. Ai medical transcription, 2025. Accedido: 2025-09-28.