UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ

EXTENSIÓN EN EL CARMEN

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Creada Ley No. 10 – Registro Oficial 313 de Noviembre 13 de 1985

PROYECTO INTEGRADOR

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERAS EN TECNOLOGÍAS DE LA INFORMACIÓN

TEMA

APLICACIÓN WEB CON PNL PARA LA IMPLEMENTACIÓN DEL MÉTODO ANALÍTICO-SINTÉTICO EN CLASES DE LA ULEAM, EL CARMEN.

AUTOR

MATTEO JUNIOR VELEZ HOLGUIN

TUTOR

ING. CESAR AUGUSTO SINCHIGUANO CHIRIBOGA, MG.

EL CARMEN, 2026

CERTIFICACIÓN DEL TUTOR

Uleam ELOY ALFARO DE MANABÍ	NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A).	CÓDIGO: PAT-04-F-004
	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1
		Página 1 de 1

CERTIFICACIÓN

En calidad de docente tutor de la Extensión El Carmen de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **Velez Holguin Matteo Junior**, legalmente matriculado en la carrera de Tecnologías de la Información, periodo académico 2025-2026, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"APLICACIÓN WEB CON PNL PARA LA IMPLEMENTACIÓN DEL MÉTODO ANALÍTICO-SINTÉTICO EN CLASES DE LA ULEAM, EL CARMEN".**

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

El Carmen, 31 de julio de 2026.

Lo certifico,

Ing. César Augusto Sinchiguano Chiriboga
Docente Tutor
Área: Tecnologías de la información

TRIBUNAL DE SUSTENTACIÓN

Universidad Laica Eloy Alfaro de Manabí

Extensión El Carmen

Carrera de Ingeniería en Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

Título del Trabajo de Titulación:

APLICACIÓN WEB CON PNL PARA LA IMPLEMENTACIÓN DEL MÉTODO ANALÍTICO-SINTÉTICO EN CLASES DE LA ULEAM, EL CARMEN

Modalidad:

Proyector Integrador

Autor:
Matteo Junior Velez Holguin

Tutor:

Ing. Cesar Augusto Sinchiguano Chiriboga, Mg

Tribunal de Sustentación:

- **Presidente**: Ing. Mora Marcillo Alex Bladimir, Mg. _____
- **Miembro:** Ing. Quiroz Valencia Arturo Patricio, Mg. _____
- **Miembro:** Ing. Villarreal Cobena Angel Wilson, Mg. _____

Fecha de Sustentación:

27 de agosto de 2026

UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ

EXTENSIÓN EN EL CARMEN

DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: **Aplicación Web con Pnl para la Implementación del Método Analítico-Sintético en Clases de La Uleam, El Carmen,** corresponde exclusivamente a: **VELEZ HOLGUIN MATTEO JUNIOR** con CI. **0927857862** y los derechos patrimoniales de la misma corresponden a la Universidad Laica "Eloy Alfaro" de Manabí.

Velez Holguin Matteo Junior

0927857862

DEDICATORIA

Los últimos meses de este proyecto se sostuvieron en algo más simple que la inteligencia artificial o el procesamiento de lenguaje: la insistencia de seguir, aunque el avance fuera lento. Ese tipo de constancia rara vez se construye solo.

Antes que nada, gracias a Dios, por mantenerme de pie en los tramos donde las fuerzas y la certeza empezaban a fallar, y por permitirme llegar hasta acá con el proyecto terminado.

A mi familia le debo la parte que ningún capítulo de este proyecto puede documentar: los turnos en que alguien preguntó cómo iba el proyecto sin entender del todo de qué se trataba, la paciencia frente a las noches dedicadas a corregir código en vez de estar presente, y la costumbre de dar por hecho que este día iba a llegar, mucho antes de que yo mismo estuviera seguro de eso. Ese tipo de confianza sostenida en el tiempo no se agradece una sola vez, se agradece siempre.

A Mayerli Rivera, mi novia, gracias por estar desde el primer día a mi lado, apoyándome en cada momento y en cada etapa de este proceso. Hubo tramos donde sostener el ánimo me costaba más que el código mismo, y ahí estuvo ella, sin soltarme y apoyandome.

A los amigos que acompañaron el proceso desde dentro y fuera de la carrera, Mateo Vasconez, Washington Briones, Josue Bravo, Karen Montalván y Jefferson Bone, gracias por las conversaciones que no tenían nada que ver con transcripciones ni modelos de lenguaje, y que servían justamente para eso: para recordar que había vida además del proyecto.

Y a los compañeros de aula con quienes se compartió examen tras examen, entrega tras entrega, gracias por convertir cinco años de exigencia en algo más llevadero de lo que habría sido en solitario.

Dedico este trabajo a todos ellos, con la certeza de que ninguna línea de este documento habría sido posible sin ese respaldo constante.

-Matteo Junior Velez Holguin

AGRADECIMIENTO

Antes de entrar en los detalles técnicos y metodológicos de este trabajo, hay un agradecimiento que no puede quedar fuera. A Dios, por sostenerme en los tramos donde el proyecto parecía estancado y la salida no era clara. A mi familia, por acompañar de cerca cada etapa sin necesitar entender del todo en qué consistía. A mi novia, porque estuvo presente desde el inicio y sostuvo el ánimo cuando más hacía falta, y a mis amigos, por recordarme que había vida más allá del proyecto incluso en las semanas donde parecía ocuparlo todo.

El desarrollo de este proyecto habría sido considerablemente más difícil sin la orientación del Ing. César Augusto Sinchiguano Chiriboga, tutor de este proyecto, cuya revisión constante de cada avance marcó buena parte de las decisiones metodológicas tomadas a lo largo del trabajo.

Los docentes y estudiantes de Ingeniería en Tecnologías de la Información que respondieron las encuestas y entrevistas de este proyecto aportaron algo que ningún dato bibliográfico podía ofrecer: una descripción de primera mano del problema que ULEAMTranscribe intenta resolver. Ese aporte queda reconocido aquí.

A quienes acompañaron de cerca este proceso, gracias por la paciencia durante los meses en que el proyecto ocupó buena parte del tiempo que debía repartirse en otras cosas. Nadie necesitó entender qué era un pipeline de transcripción, ni por qué una tarjeta gráfica con poca memoria complicaba tanto las cosas, para seguir preguntando cómo iba el avance. Gracias también a quienes ayudaron a probar la aplicación sin ser parte de la carrera, y a quienes ofrecieron la distracción justa en los momentos de más tensión.

Por último, un agradecimiento a la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, espacio donde se formó tanto el conocimiento técnico como el criterio necesario para desarrollar este proyecto.

-Matteo Junior Velez Holguin

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE ANEXOS

RESUMEN

El presente proyecto de titulación tiene como finalidad el desarrollo de ULEAMTranscribe, una aplicación web que transcribe grabaciones de clases universitarias y genera resúmenes académicos estructurados mediante Procesamiento de Lenguaje Natural, destinada a la comunidad estudiantil de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen. Revisar grabaciones completas de clase para repasar el contenido demanda un tiempo considerable, y los puntos clave suelen quedar dispersos entre explicaciones extensas, lo que dificulta un estudio individual eficiente.

La investigación se fundamentó en un análisis bibliográfico sobre procesamiento de lenguaje natural, reconocimiento automático del habla y el método analítico-sintético, junto con la recolección de datos mediante encuestas a 80 de 92 estudiantes de Ingeniería en Tecnologías de la Información y entrevistas a 10 de 12 docentes de la carrera. Los resultados mostraron que una buena parte de los estudiantes dedica más de media hora para ir a localizar un único tema dentro de una grabación, y que los docentes consideran la terminología técnica mixta de cada una de las materias como el principal riesgo de fidelidad de cualquier sistema de resumen automático, así que en función de esto se realizó una propuesta de aplicación que sea capaz de generar informes académicos de forma organizada en función de las etapas de análisis y de síntesis de la información escritos de forma que se preserven las informaciones reales del contenido citado.

La evaluación del sistema, realizada mediante pruebas con grabaciones reales de distintas asignaturas y la interacción directa de docentes y estudiantes, confirmó que el pipeline opera dentro de la restricción de 4 GB de memoria de video del equipo de desarrollo, que las secciones fijas del informe se completan con contenido pertinente sin ajustes manuales por materia, y que la interfaz permite a los usuarios completar las tareas propuestas sin asistencia del desarrollador. Las pruebas también identificaron limitaciones puntuales, documentadas en el capítulo de verificación, relacionadas con la redacción de la sección de desarrollo en clases con procesos numerosos y con errores propagados desde la transcripción de audio a texto.

ABSTRACT

The purpose of this thesis project is the development of ULEAMTranscribe, a web application that transcribes recordings of university lectures and generates structured academic summaries through Natural Language Processing, intended for the student community of Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen. Reviewing complete lecture recordings to study demands a considerable amount of time, and key points tend to remain scattered across lengthy explanations, which makes efficient individual study difficult.

The research was based on a bibliographic analysis of natural language processing, automatic speech recognition and the analytical-synthetic method, together with data collection through surveys of 80 out of 92 students of Information Technology Engineering and interviews with 10 out of 12 faculty members of the program. The results showed that a significant share of students spend more than half an hour locating a single topic within a recording, and that faculty members identify the mixed technical terminology of each subject as the main fidelity risk for any automatic summarization system. Based on these findings, an application was proposed capable of generating academic reports organized according to the analysis and synthesis stages of information, without compromising the accuracy of the data cited.

The evaluation of the system, carried out through tests with real recordings of different subjects and direct interaction from faculty and students, confirmed that the pipeline operates within the 4 GB video memory constraint of the development equipment, that the fixed sections of the report are completed with pertinent content without manual adjustments per subject, and that the interface allows users to complete the proposed tasks without assistance from the developer. The tests also identified specific limitations, documented in the verification chapter, related to the drafting of the development section in classes with numerous processes and to errors propagated from the audio-to-text transcription.

CAPÍTULO I

INTRODUCCIÓN

El Procesamiento de Lenguaje Natural (PLN) es una rama de la inteligencia artificial orientada a interpretar, transformar y generar lenguaje humano (Giraldo Forero y Orozco Duque, 2023). En el contexto universitario, esta tecnología se ha extendido para potenciar la enseñanza y el aprendizaje (Ramírez, 2022), ayudando a enfrentar la sobrecarga de información mediante herramientas como el reconocimiento de voz a texto (Ramírez, 2022). Como disciplina que combina el procesamiento del lenguaje con ciencia cognitiva, interfaces conversacionales y recuperación de información (Jácome-Jara y Cevallos-Campoverde, 2022), el PLN permite a los estudiantes trabajar con grandes volúmenes de audio y texto de forma más ágil.

En la tarea específica de analizar recursos educativos, el PLN supera a los enfoques tradicionales que ignoran el contexto de las palabras dentro de un enunciado completo (Giraldo Forero y Orozco Duque, 2023). Técnicas como las representaciones vectoriales de palabras permiten que términos afines o equivalentes queden más próximos entre sí en un espacio multidimensional (Giraldo Forero y Orozco Duque, 2023), lo que acelera la identificación de ideas centrales. De igual modo, arquitecturas como los transformers logran capturar el significado global de un texto (Giraldo Forero y Orozco Duque, 2023), mientras que redes como las LSTM retienen información previa del discurso para refinar el análisis (Ramírez, 2022), lo que favorece una extracción de información tal como es y considerada en su contexto.

La Universidad Laica Eloy Alfaro de Manabí en su Extensión El Carmen no es ajena a esta problemática. Los alumnos graduados y/o en formación del programa de Ingeniería en Tecnologías de la Información en dicha extensión, junto con sus profesores, no son ajenos a esta dificultad, no solo por la complejidad y especificidad de los contenidos de sus asignaturas, sino, sobre todo, por la ausencia de herramientas digitales que permitan hacer una reducción automática del material de clase. Por tanto, se avanza en esta propuesta de mejora en el orden pedagógico mediante los avances de la IA.

Este proyecto responde precisamente a esa carencia, al proponer una aplicación web que integra PLN para agilizar la elaboración de resúmenes educativos. Más allá de resolver la sobrecarga de información, esta propuesta incorpora una perspectiva pedagógica basada en el

método analítico-sintético, un procedimiento que promueve un dominio más completo del contenido al descomponerlo y luego recomponerlo.

1.1 Presentación del tema

El objeto de este trabajo de titulación consiste en el diseño y desarrollo de una aplicación web que haga uso de PLN y que permita utilizar el método analítico-sintético en las aulas de la ULEAM, Extensión El Carmen. Esta idea también introduce tecnologías actuales relacionadas con la Inteligencia Artificial, tales como los large language models como Llama y el modelo de OpenAI, GPT, así como los sistemas de transcripción automática de la voz, como el modelo Whisper de OpenAI;

1.2 Ubicación y contextualización de la problemática

La Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, se ubica en el cantón El Carmen, provincia de Manabí, en la Avenida 3 de Julio. Dicha Universidad fue creada a través de la Ley No. 10, que fue inscrita en el Registro Oficial 313 el 13 de noviembre de 1985.

En la oferta académica de la Universidad se incluye la carrera de Ingeniería en Tecnologías de la Información que es asistida por un estudiantado que presenta ciertas complicaciones para lograr la comprensión de ciertas materias técnicas avanzadas. Las asignaturas del currículo demandan habitualmente procesar información compleja sobre programación, gestión de datos, infraestructura de redes, inteligencia artificial y otras áreas propias de la informática.

El nudo del problema se encuentra en las aulas de la práctica, que, posteriormente, sirven para grabar las sesiones con la intención de revisar esos audios, pero la universidad carece de herramientas digitales que permitan maximizar el aprovechamiento de esos audios, de forma que los alumnos dediquen muchas horas al estudio a partir de repasos manuales, una estrategia poco eficiente y probablemente incompleta.

Dicha circunstancia se ve agravada por otros factores relacionados con el contexto. Muchos estudiantes deben ir compaginando el estudio académico y el trabajo y/o responsabilidades familiares, lo que reduce las horas disponibles para la asimilación, así como el hecho de que el campo tecnológico es un dominio que ha de ir progresando a lo largo de toda la titulación, para lo cual se necesita una base sólida de conceptos. Una base que a menudo no se puede aportar sin recursos de estudio.

1.3 Planteamiento del problema

1.3.1 Problematización

Al examinar la situación actual en la ULEAM, Extensión El Carmen, se identifica un problema concreto relacionado con el manejo de datos educativos. Los alumnos de la carrera de Ingeniería en Tecnologías de la Información presentan significativas dificultades para asumir los contenidos de las materias, en especial en aquellas asignaturas que solo se encuentran en grabaciones de audio.

El principal problema es que hay escasez de documentos que transformen las grabaciones sonoras largas en documentos de estudio organizados y accesibles. Las grabaciones sonoras, que suelen tener una duración superior a la hora, recogen información útil con escasa regularidad a lo largo de toda la sesión. Las personas que están deseando repasar un tema concreto o atender a un examen particular invierten, al final, muchas horas en tratar de encontrar pasajes específicos de un audio extenso.

Esto plantea la necesidad de que los estudiantes cuenten con mecanismos que agilicen sus repasos cuando deben revisar grabaciones completas, sin que los puntos clave de las clases queden relegados en el estudio individual. Los enfoques digitales capaces de automatizar la captura y la síntesis de contenido educativo sin sacrificar su exactitud representan una vía concreta para atender esa necesidad.

El reto adicional es mantener el rigor pedagógico en medio de la automatización: una herramienta tecnológica debe facilitar el acceso a la información sin debilitar el desarrollo de habilidades analíticas y de síntesis en los estudiantes, equilibrando la rapidez que ofrece la tecnología con la necesidad de que los alumnos desarrollen habilidades propias en el manejo de datos.

1.3.2 Génesis del problema

El origen de este problema se remonta a la transformación de las estrategias de enseñanza universitaria. En el pasado, los estudiantes se limitaban a estudiar los apuntes realizados a mano una vez terminada la clase. Esta estrategia, implantada durante años y habitual en su quehacer, planteaba varios obstáculos: la necesidad de mantener la atención de forma continua, la

velocidad con la que cada estudiante lograba escribir, y la interpretación que cada uno hacía después de sus propias anotaciones al repasarlas.

Con el uso de la tecnología y el uso de los dispositivos de grabación, la práctica de grabar las clases en audio tuvo una difusión y pasó a ser común. Al menos en sus primeras etapas se entendía que el problema quedaba resuelto, ya que se podía acceder al contenido en clase sin depender de los apuntes. La grabación traía consigo nuevos problemas relacionados con la organización del tiempo y la productividad de las actividades de estudio.

La dificultad del problema incrementó con la profundización y especialización de los temas en la realización de los programas técnicos como este. El profesorado transmite conocimientos especializados que requieren de una comprensión minuciosa, que cubre lecturas de teorías complejas, procedimientos exhaustivos, aplicaciones prácticas que requieren ser interiorizadas. Grabar todo ese contenido produce archivos, en extensas grabaciones de audio que, paradójicamente, terminan quedando subutilizadas por lo difícil que resulta su gestión de forma eficaz.

Además, los cambios recientes en los formatos de enseñanza han intensificado el problema. El auge de herramientas educativas digitales y la conversión digital de materiales han multiplicado los audios y videos que los estudiantes deben gestionar, lo que exige soluciones para procesarlos con mayor agilidad.

1.3.3 Estado actual del problema

Actualmente, en la ULEAM Extensión El Carmen, existe un desfase entre la creciente cantidad de recursos en audio y la capacidad de los estudiantes para aprovecharlos. La carrera de Ingeniería en Tecnologías de la Información va acumulando grabaciones cada semestre como colecciones personales que, en la mayoría de los casos, no se utilizan de forma sistemática por el tiempo que exigen.

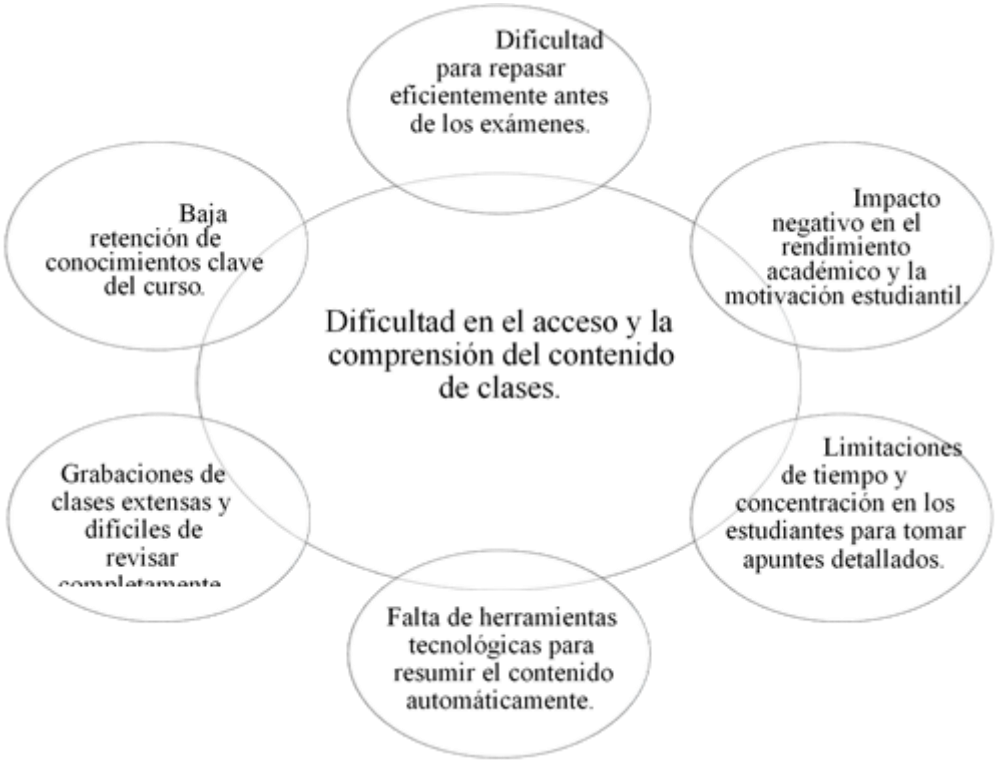
Las consecuencias afectan, y se extienden, a diversos aspectos de la formación académica, donde los estudiantes informan que les resulta muy complicado prepararse de forma adecuada para las evaluaciones cuando solo deben contar con extensos audios como única base de estudio. Obtener una información concreta a partir de esos archivos puede requerir un tiempo tal que podría ser destinado a la comprensión y/o aplicación de los conceptos. A la larga, son

muchos los que optan por no revisarlos, desaprovechando de esta forma la posibilidad de dar un refuerzo ordenado al conocimiento.

La problemática también repercute en la igualdad con respecto a los recursos que se ponen a disposición para el estudio. La diversidad de estilos de aprendizaje es dejada de lado por la ausencia de formatos alternativos. Aquellas personas que prefieren leer o repasar por escrito se topan con una dificultad adicional si las opciones consisten únicamente en las auditivas, y quienes tienen horarios apretados por trabajo o familia no pueden optar por una serie de caminos rápidos hacia contenido importante.

Desde la perspectiva institucional, se destinan esfuerzos a grabar y almacenar contenido sin mecanismos que permitan aprovecharlo al máximo en beneficio del aprendizaje. Esas grabaciones representan un recurso pedagógico valioso que permanece subutilizado por la falta de herramientas que las conviertan en material de estudio práctico y accesible.

1.4 Diagrama causa-efecto del problema



1.5 Objetivos

1.5.1 Objetivo general

Desarrollar un sistema informático que permita generar resúmenes automáticos de clases grabadas, utilizando técnicas de Procesamiento de Lenguaje Natural, con el fin de mejorar el acceso, la comprensión y el repaso del contenido académico en la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen.

1.5.2 Objetivos específicos

- Analizar los requerimientos académicos y tecnológicos necesarios para la implementación de un sistema de resumen automático de clases en la ULEAM Extensión El Carmen.
- Diseñar la arquitectura del sistema informático incorporando módulos de transcripción de audio y procesamiento de lenguaje natural.
- Implementar un prototipo funcional que procese grabaciones de clases y genere resúmenes comprensibles y coherentes.
- Evaluar la efectividad del sistema mediante pruebas con grabaciones reales de clases y la retroalimentación de los usuarios.

1.6 Justificación

El planteamiento de este trabajo se justifica desde diferentes ópticas complementarias: técnica, pedagógica, institucional y social. En el sentido tecnológico, el instrumento constituye una buena cápsula práctica para mostrar los hallazgos de la inclusión de los avances de la inteligencia artificial en el ámbito educativo real, facilitando un ejemplo sobre cómo estas innovaciones pueden modificar las dinámicas internas de la enseñanza-aprendizaje.

En el sentido pedagógico, el valor del proyecto es que concuerda con lo que convenimos en llamar analítico-sintético, el enfoque didáctico de gran tradición pedagógica que propugna la comprensión de los fenómenos a partir de su descomposición y que por tanto debe ser rehecho. Automatizar este proceso en el marco de la delimitación de PLN nos da acceso a materiales organizados que alientan el análisis crítico y la reflexión, sin perder la posibilidad de abrir parte de las habilidades cognitivas de los estudiantes. Los estudiantes cuentan con resúmenes que permiten articular con lógica las ideas fundamentales, dilucidando así el horizonte de la comprensión con preguntas pertinentes y situaciones reales para el aprendizaje.

Desde la óptica institucional, aplicar estas herramientas implica un paso adelante en la modernización de los recursos educativos de la ULEAM Extensión El Carmen, pues la institución manifiesta su esfuerzo por ser innovadora en cuanto a la educación y por integrar tecnología que se transforma en beneficio inmediato de la comunidad estudiantil, además de convertirse en un referente en el uso de inteligencia artificial en la mejora de procesos académicos.

El aspecto social se justifica por su aportación al ampliar el acceso a herramientas educativas de calidad. Al optimizar la gestión de los contenidos de clase, se beneficia a quienes por trabajar o cumplir obligaciones familiares registran poco tiempo disponible. Asimismo, los alumnos con distintos estilos de aprendizaje encuentran en esta aplicación un complemento que mejora sus estrategias de aprendizaje, logrando mayor equidad respecto a las oportunidades de éxito académico.

La tecnología es factible gracias a cómo se encuentran las soluciones existentes, así como el nivel de madurez de las tecnologías de procesamiento de lenguaje, y todas las soluciones de reconocimiento de voz. Modelos de tipo Llama, GPT o Whisper han demostrado su idoneidad para tareas similares; por este motivo, se considera que la propuesta está adecuadamente fundamentada. Todo esto, sumado a los frameworks de desarrollo web existentes y el acceso relativamente sencillo a servicios disponibles en la nube, nos permiten considerar su implementación.

El aspecto económico está dado por cómo el sistema optimiza el uso de los recursos existentes: reducir el tiempo que los estudiantes utilizan para procesar el material de estudio puede facilitar el tiempo restante en alcanzar una cierta productividad académica y, por lo tanto, quizás, en obtener mejores calificaciones. Para la universidad, aprovechar mejor las grabaciones ya existentes representa un mejor retorno de la inversión en equipos de grabación y almacenamiento de contenido educativo.

1.7 Impactos esperados

1.7.1 Impacto tecnológico

El impacto tecnológico del proyecto se extiende más allá de su implementación puntual. En primer lugar, la aplicación desarrollada sienta un precedente en la integración de inteligencia artificial dentro del entorno educativo local. Demostrar que técnicas avanzadas y de última

generación de PLN dan respuesta a problemas educativos reales empujará a otros proyectos de innovación tecnológica en la Institución y en contextos paralelos.

La incorporación del sistema generará conocimiento de tipo práctico sobre la adaptación de modelos lingüísticos de gran escala, generando condiciones de reconocimientos y positivos cambios en el ámbito universitario por su vocabulario técnico, su estructura del discurso, sus patrones temáticos que requieren modificaciones y ajustes concretos en algoritmos; y las soluciones que se encuentren para estos retos suministrarán conocimiento al existente en cuanto a las aplicaciones de PLN en la educación.

En lo que respecta al fortalecimiento institucional, la incorporación del sistema inducirá un incremento notable de las capacidades digitales de la ULEAM Extensión El Carmen, dado que la construcción y la puesta en marcha del sistema requieren un conjunto de conocimientos consolidados sobre todo a nivel de aprendizaje de máquina, de procesamiento de lenguaje, de desarrollo web y de diseño de software generando competencias útiles para futuros proyectos de transformación digital en el seno de la Institución.

1.7.2 Impacto social

El impacto social del proyecto se centra en mejorar la experiencia formativa y las posibilidades de éxito académico de los estudiantes de la ULEAM Extensión El Carmen. Contar con resúmenes automáticos de las clases resulta especialmente útil para quienes enfrentan limitaciones de tiempo por motivos económicos o familiares. Quienes combinan estudios con trabajo encuentran en la herramienta un medio para optimizar su ritmo de seguimiento de los contenidos.

La implementación del sistema facilita una provisión más equitativa de los recursos educativos. De esta forma los estudiantes que tienen ritmos distintos en cuanto a la concentración, velocidades de aprendizaje o preferencias de estudio se ven favorecidos con contenidos textuales organizados que acompañan a los originales. Esta combinación de formatos está atendiendo a criterios de diseño inclusivo para el aprendizaje, favoreciendo que un número de estudiantes más elevado tenga una llegada más efectiva a los conocimientos que se trasladan.

Además, el proyecto tiene un efecto social al formar a estudiantes desarrolladores en tecnologías de alta demanda laboral. Participar en la creación e implementación del sistema

brinda experiencia práctica con métodos de inteligencia artificial, preparando mejor a los egresados para integrarse en mercados laborales que cada vez exigen más estas competencias.

1.7.3 Impacto ecológico

Aunque no es un efecto directo, el impacto ambiental del proyecto merece consideración en una evaluación integral de sus consecuencias. Digitalizar y automatizar la elaboración de resúmenes reduce considerablemente la necesidad de materiales impresos que los estudiantes solían elaborar manualmente para resumir clases. Menos uso de papel favorece la sostenibilidad ambiental de las actividades universitarias.

La eficiencia en el tiempo que promueve el sistema conlleva beneficios ambientales adicionales. Los estudiantes que acceden a resúmenes con rapidez pasan menos tiempo en las instalaciones universitarias, lo que podría reducir el consumo de energía eléctrica, calefacción o climatización. Aunque el efecto individual parece mínimo, sumado a todo el estudiantado genera diferencias apreciables en términos ecológicos.

El diseño técnico del sistema incorpora criterios de eficiencia computacional para evitar el consumo energético innecesario. Optimizar algoritmos, aprovechar adecuadamente los recursos en la nube y adoptar prácticas de programación sostenibles contribuyen a reducir la huella de carbono de sus operaciones.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes Históricos

2.1.1 Evolución del Procesamiento de Lenguaje Natural

Las bases del Procesamiento de Lenguaje Natural (PLN) se remontan a la década de 1950, con Alan Turing y su conocido "Test de Turing" para evaluar si una máquina podía imitar la inteligencia humana. En 1954, el experimento Georgetown-IBM logró traducir automáticamente al inglés más de sesenta frases en ruso, abriendo el camino a la traducción automática. En los años sesenta y setenta se desarrollaron sistemas de conocimiento basado en reglas gramaticales, como ELIZA en 1966, capaz de simular una serie de conversaciones de tipo terapéutico reconociendo las estructuras más básicas del lenguaje.

Las décadas de 1980 y de 1990 suponen una transformación importante del área de la comprensión del lenguaje natural y un cambio de rumbo en la dirección de los enfoques probabilistas y de los enfoques basados en datos, lo que llevó a la elaboración de herramientas mucho más robustas. La incorporación de grandes corpus escritos junto con el aumento de la capacidad de cómputo permitió poner en marcha el desarrollo de modelos empíricos, mientras que ya en el siglo XXI el aprendizaje automático exploró nuevas direcciones en la comprensión del lenguaje natural, sobre todo gracias a la aparición de las redes neuronales profundas. En 2013, se presentó Word2Vec, un método para representar las palabras en forma de vectores en un espacio multidimensional capaz de captar relaciones de significado.

En el 2017 el paradigma cambió completamente con la llegada de los transformers, que fueron presentados en el artículo "Attention is All You Need" por investigadores de Google. Esta nueva arquitectura basada en mecanismos de atención fue capaz de deshacerse del procesamiento secuencial y comenzó a surgir el procesamiento en paralelo para el entrenamiento de estos modelos, que permitieron un importante avance hacia el modelo de gran escala. Posteriormente, en 2018 BERT de Google marcó un nuevo hito en el que se interpretaba el contexto de una palabra en ambas direcciones de la oración.

Posteriormente a OpenAI se le han de los modelos de la familia GPT desde su primera versión de GPT-1 en 2018, pasando por GPT-2 en 2019, hasta llegar a GPT-3 en el año 2020 que tiene 175 mil millones de parámetros, y que lograron un gran rendimiento en tareas de PLN como

generación de texto, traducción y resumen. A partir de ChatGPT, que se lanzó públicamente basado en GPT-3.5 en el año 2022, se dio un paso fundamental en la divulgación de la inteligencia artificial conversacional. Ya para 2024, GPT-4 establecía nuevas cotas en comprensión y generación de lenguaje e iba afianzando al PLN como un pilar en los ámbitos educativo, empresarial y científico..

2.1.2 Historia del Reconocimiento Automático de Voz

El Reconocimiento Automático de Voz (ASR, por sus siglas en inglés) comenzó en 1952 con "Audrey", desarrollado por Bell Labs, capaz de identificar los números del cero al nueve pronunciados por una sola voz. IBM presentó "Shoebox" en la Exposición de Seattle de 1962, que reconocía dieciséis palabras en inglés además de los dígitos. Durante la década de 1970, se consiguió que el proyecto DARPA de comprensión del habla culminara en "Harpy", el cual era capaz de reconocer más de mil palabras.

A partir de los años 80, la inercia para la creación de los métodos estadísticos fue en auge, donde los Modelos Ocultos de Markov (HMM) fueron los más consolidados en la práctica durante más de dos décadas. Dragon Dictate surgió en 1990 como el primer software comercial de dictado por voz orientado al uso de usuarios comunes. En la década de 2000, el aprendizaje de máquinas y el uso masivo de datos refinó significativamente el proceso.

La mayor innovación se produjo en el año 2010, propiciada por la adopción de las redes neuronales profundas (DNN) en el contexto del reconocimiento del habla, y Google fue la que las incorporó en el servicio de búsqueda por voz en el año 2012, lo cual conllevó una reducción drástica de la tasa de error. En el año 2016, Microsoft explicó que había logrado alcanzar una tasa de error similar a la de un ser humano en la transcripción de conversaciones, ello gracias al uso de las redes neuronales recurrentes (RNN) y LSTM para la memoria a largo plazo.

En el año 2022, OpenAI introdujo Whisper, un sistema de reconocimiento de voz basado en transformers, entrenado con 680.000 horas de audio multilingüe. Se destacó por su robustez frente al ruido y su precisión en múltiples idiomas sin necesidad de ajustes adicionales. Esta innovación ha abierto el reconocimiento de voz de alta calidad a aplicaciones educativas como la de este proyecto.

2.1.3 Desarrollo de los Métodos de Enseñanza Analítico-Sintético

El método analítico-sintético surge en la pedagogía del siglo XIX, cuando educadores buscaron alternativas a los métodos tradicionales de alfabetización. Existen personajes como Friedrich Fröbel y Johann Heinrich Pestalozzi que abanderaron la moción de realizar analítica la descomposición del conocimiento y sintética su recomposición. En torno a 1890, el método sintético-analítico-sintético o de palabras normales fue ganando terreno en escuelas de Alemania y su propagación acabó llegando a otros países.

Al iniciar el siglo XX, María Montessori, a través de su sistema educativo, fue integrando aspectos analíticos y sintéticos, partiendo de la idea de descomponer en ideas simples, para después ser capaces de construir la idea compleja. Ovide Decroly, en la década de los veintes, se ocupó del llamado "método global de lectura" a partir de unidades globales para ser descompuesto luego en componentes.

En las décadas de los sesenta y setenta, la psicología cognitiva, a partir de las aportaciones de Jerome Bruner y David Ausubel, proporcionó sólidos cimientos a este modelo. Bruner defendía el aprendizaje por descubrimiento, en el que proponía que habría que desgajar la información para conseguir construir una comprensión propia. Ausubel promovía el aprendizaje significativo, que definía cómo articular lo nuevo con lo conocido a partir de los procesos de análisis y síntesis.

Pero en los años recientes este modelo se ha hecho todavía más presente gracias a las tecnologías digitales, pues los sistemas de PLN descomponen los textos en los elementos que los componen (fase analítica) y los recomponen en resúmenes coherentes (fase sintética), constituyendo una traducción moderna de esta tradición didáctica. Dicho esto, podemos decir que en el país andino, este modelo se utiliza en la alfabetización inicial desde la actualización curricular de 1996 y se mantiene en niveles superiores para llevar a cabo el análisis intensivo y el juicio crítico.

2.2 Antecedentes de Investigaciones Relacionadas al Tema

Chancay (2024) diseñó una aplicación para lograr la gestión de los registros de inventario en pequeñas empresas textiles, incorporando los registros, la supervisión y los informes, lo cual hizo posible establecer un esquema operativo que aceleró los controles. Su aplicación es más bien comercial, pero enlaza con este proyecto en cuanto al uso de herramientas digitales para facilitar el trabajo manual y así aumentar la productividad y disminuir los errores.

Álvarez Pincay y Herrera Calle (2024) analizaron el sistema de inventario del comercial Osejos, en el cantón Jipijapa, y encontraron que la ausencia de un sistema de gestión adecuado generaba ineficiencias en el control y manejo de las mercaderías. Sus conclusiones señalan la

necesidad de modernizar el sistema mediante herramientas tecnológicas avanzadas, lo cual refuerza la pertinencia de automatizar procesos de registro y control en contextos donde tradicionalmente se manejan de forma manual, un principio que también sustenta el presente proyecto.

Guachamin y sus co-autores (2023) han creado una aplicación a medida para la compañía SOLINTEG360 de Quito, utilizando MAUI y PHP para realizar información en tiempo real, gestionar pedidos y generar reportes. Este trabajo ha permitido disminuir errores, incrementar la velocidad de los procesos y mejorar la generación de la toma de decisiones en función de los datos, o sea, es funcional y transferible al contexto universitario del proyecto.

Coloma Garófalo y Agualongo Caiza (2023) desarrollaron una aplicación móvil para el seguimiento del ganado bovino en la Hacienda San Gabriel, con módulos de gestión de bovinos, control de producción, gestión veterinaria y generación de reportes, en sustitución de los registros en papel que dificultaban el procesamiento de la información. Este trabajo revela la adaptabilidad de las soluciones tecnológicas a diferentes contextos, desde la ganadería hasta el aula.

Si bien Chancay (2024), Álvarez Pincay y Herrera Calle (2024), Guachamin et al. (2023) y Coloma Garófalo y Agualongo Caiza (2023) se centraron en aplicaciones y sistemas para el control de inventarios y la gestión de procesos en comercio, servicios y ganadería, el presente proyecto se distingue al enfocarse exclusivamente en el ámbito universitario. Más que automatizar un proceso administrativo, busca enriquecer el aprendizaje mediante el método analítico-sintético apoyado en PLN de última generación.

La incorporación de modelos lingüísticos de gran escala como GPT y BERT, junto con reconocimiento de voz mediante Whisper de OpenAI, representa un avance frente a los sistemas de gestión de datos convencionales revisados en los antecedentes citados. Este enfoque prioriza el efecto pedagógico, promoviendo la reflexión crítica y la comprensión profunda de los temas, un aspecto ausente en esos estudios previos.

2.3 Definiciones Conceptuales

Las definiciones conceptuales de este capítulo se organizan en dos ejes complementarios, que reflejan la doble naturaleza del proyecto planteada desde el título del trabajo. El primero reúne los fundamentos tecnológicos que sustentan la aplicación web con Procesamiento de Lenguaje Natural: los modelos de lenguaje, el reconocimiento de voz, la arquitectura web y el

almacenamiento de datos. El segundo reúne los fundamentos pedagógicos del método analítico-sintético y su vínculo con la síntesis automatizada de contenido educativo, el dominio específico al que se aplica la tecnología descrita en el primer eje.

2.3.1 Aplicación Web con Procesamiento de Lenguaje Natural

Este eje reúne los componentes tecnológicos que conforman ULEAMTranscribe: las técnicas de Procesamiento de Lenguaje Natural que interpretan y generan texto, los modelos lingüísticos de gran escala que ejecutan esa interpretación, el reconocimiento automático de voz que convierte el audio de las clases en texto, la arquitectura de aplicación web que expone el sistema a los usuarios, la metodología de desarrollo que organiza su construcción, y la gestión de datos persistentes que almacena la información generada.

2.3.1.1 Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural es una subdisciplina de la inteligencia artificial centrada en la interacción entre los sistemas computacionales y el lenguaje humano. De acuerdo con Hou y Huang (2025), se define técnicamente como un conjunto de métodos computacionales orientados a adquirir, interpretar y generar lenguaje cotidiano. Esta área resulta indispensable en la actualidad, ya que los datos lingüísticos humanos suelen presentarse de forma no estructurada, a diferencia de los datos tabulares convencionales.

Mediante el procesamiento del lenguaje natural (PLN), las máquinas tienen la posibilidad de procesar y analizar automáticamente enormes cantidades de texto en lenguaje natural; esto resulta clave para aplicaciones tan relevantes como la traducción automática, el análisis de sentimientos, la generación de resúmenes, los chatbots o los asistentes virtuales. El procesamiento del texto incluye etapas como la limpieza y codificación en las que, una vez se ha despojado de elementos redundantes, se convierte dicho texto a valores numéricos que pueden ser interpretados por algoritmos de aprendizaje automático.

- ***Elementos del PLN***

El PLN integra distintos niveles de análisis lingüístico que trabajan en conjunto para interpretar y generar lenguaje humano. El análisis léxico es la primera etapa, encargada de identificar y clasificar las palabras individuales del texto. Incluye la tokenización, que divide el contenido en unidades o tokens; la eliminación de palabras vacías; y la lematización o derivación, que reduce las palabras a su forma base.

El análisis sintáctico, por su parte, es el que se encarga de examinar la estructura gramatical de las oraciones, identificando la relación entre palabras y la jerarquía de los elementos, determinando así la función de cada uno de los términos, bien como sujeto, verbo y/u objeto del predicado; el análisis semántico, en cambio, analiza el significado de palabras u oraciones dentro de su contexto, más allá de la gramática y abarcando, pues, relaciones de significado, resolución de ambigüedades y comprensión del mensaje global.

Por último, el análisis pragmático incorpora el contexto amplio del uso del lenguaje, considerando el conocimiento general, la intención del hablante y las circunstancias específicas. Esto permite interpretar matices como el sarcasmo, las figuras retóricas o las insinuaciones, logrando una comprensión más completa del discurso humano.

- ***Métodos Esenciales del PLN***

La representación de palabras es el primer paso técnico del proceso, al convertir los términos en formas numéricas aptas para el procesamiento algorítmico. Los enfoques clásicos son la Bolsa de Palabras y el TF-IDF, mientras que los enfoques contemporáneos giran en torno a las representaciones vectoriales como Word2Vec, GloVe o versiones contextualizadas que comprenden matices más complejos.

El modelado del lenguaje trata acerca de estimar la probabilidad de las secuencias de palabras. Aunque los n-gramas tradicionales estuvieron de moda en otro tiempo, han sido superados por redes neuronales como las LSTM, que son aptas para capturar relaciones a más larga distancia. Más recientemente, las arquitecturas de transformers han transformado esta área de estudio porque permiten cálculos paralelos y capturan interacciones textuales más complejas.

Mediante el análisis de dependencias se determina cómo se relacionan los términos de una oración determinando una representación en forma de árbol que refleja esas conexiones, y que resulta ser un método básico para estudiar la estructura del lenguaje. El reconocimiento de entidades nombradas, por su parte, determina y clasifica las entidades concretas que contiene un texto concreto (personas, organizaciones, lugares, fechas, etc.) que permite extraer informaciones estructuradas a partir de documentos no estructurados.

2.3.1.2 Modelos Lingüísticos de Amplia Escala (LLM)

Los Modelos Lingüísticos de Amplia Escala (LLM, por sus siglas en inglés) representan un avance significativo en el campo del PLN. Según indican Ruan et al. (2025), son modelos base entrenados en "enormes repositorios de datos para ofrecer capacidades versátiles y robustas",

capaces de abordar múltiples tareas sin requerir entrenamiento específico para cada una. Estos modelos han transformado la manera en que las máquinas procesan y generan lenguaje natural, impulsando avances importantes en aplicaciones reales.

- ***BERT (Representaciones de Codificador Bidireccional de Transformers)***

BERT se describe como un "modelo de representación del lenguaje basado en transformers, destacado en múltiples tareas de PLN", según Gardazi et al. (2025). Su característica distintiva es el entrenamiento bidireccional, que incorpora información de contexto de ambos lados de una secuencia simultáneamente, a diferencia de modelos anteriores que procesaban el texto en una sola dirección, limitando así su comprensión del contexto completo.

BERT se encuentra constituido por la utilización únicamente del componente codificador del transformer. Consiste en capas apiladas de transformers con mecanismos de atención multi-cabeza y de redes de propagación hacia adelante. Las versiones estándar son BERT-Base (12 capas y 110 millones de parámetros) y BERT-Large (24 capas y 340 millones de parámetros). Con dicha forma, profundiza en el procesamiento del texto, captando relaciones complejas entre palabras.

El preentrenamiento de BERT es el resultado de dos objetivos complementarios. El primero de ellos es el Modelado de Lenguaje Enmascarado, en el que hay que predecir palabras ocultas que se han enmascarado aleatoriamente en una oración, a fin de forzar al modelo a aprender contextos en ambas direcciones; el segundo es la Predicción de la Siguiente Oración, que consiste en verificar si dos fragmentos de texto tienen una relación secuencial. De esta manera, logra crear representaciones lingüísticas ricas que podrán ser apropiadas más adelante para tareas específicas.

BERT destaca en tareas de comprensión del lenguaje, como clasificación de texto, análisis de sentimientos, respuesta a preguntas y reconocimiento de entidades. Su capacidad de integrar contexto bidireccional lo hace especialmente adecuado en situaciones donde el significado de una palabra depende tanto de lo que la precede como de lo que la sigue, logrando una comprensión más precisa del lenguaje.

- ***GPT (Transformer Generativo Preentrenado)***

A diferencia de BERT, la familia GPT prioriza la generación de texto mediante predicciones autorregresivas del siguiente token, según Ruan et al. (2025). GPT únicamente se basa en la parte decodificadora del Transformer, generando el texto de manera unidireccional de

izquierda a derecha y generando palabras de una en una. Esta dirección unidireccional se muestra óptima para las necesidades de las tareas creativas de producción de texto.

La familia GPT ha evolucionado de forma importante desde que apareció en el 2018 con la publicación de su primera versión, GPT-1, con 117 millones de parámetros, y que evidenció el valor del pre-entrenamiento en corpus de texto grandes. En el 2019, se presentó GPT-2, que pasó a 1.500 millones de parámetros, y mostró las capacidades emergentes. En el 2020 aparece GPT-3, que marca un nuevo estándar con 175.000 millones de parámetros, mostrando capacidades de razonamiento y generación creativa muy notables. Finalmente, el GPT-4 se observó en el 2023, y también resulta superior a sus anteriores, aunque el número de parámetros no fue divulgado públicamente.

La técnica de preentrenamiento autorregresivo de GPT consiste en lograr la predicción de la palabra siguiente a partir de la analizante que le precede. Gracias a esto, GPT puede generar texto de manera secuencial y tal que el texto queda en un sentido total y relevante, y puede llegar a aprender patrones del habla, información factual de interés general y patrones de razonamiento que son utilizables en diferentes tareas de forma ligera a esta factible de realizarla sin ningún ajuste sugerido o implementado.

A medida que el modelo aumenta en escala, GPT puede exhibir capacidades emergentes sorprendentes, como el aprendizaje de ejemplares pocos ejemplos a partir de instrucciones (prompts); incluso sin ejemplos previos, se puede lograr información a partir del contenido dado. También exhibe patrones de razonamiento lógico, patrones del pensamiento resolutivo de problemas del tipo numérico, así como el patrón de la generación creativa en distintos estilos de redacción.

GPT destaca en las tareas de la generación textual en diversas facetas como son el resumen automático, la traducción, la generación de código, la respuesta a preguntas, la conversación y la redacción de materiales, dicha capacidad generativa lo hace útil para cumplir el objetivo de este proyecto, que consiste en sintetizar resúmenes académicos a partir de grabaciones de clase. La competencia de GPT para procesar contextos largos y generar texto fluido y estructurado es clave para poder realizar resúmenes académicos.

- ***Mecanismo de Autoatención Multi-Cabeza***

Tanto BERT como GPT incorporan el mecanismo de autoatención multi-cabeza, que establece relaciones entre los tokens relevantes para lograr una codificación más precisa, según señalan

Ruan et al. (2025). De esta manera, el modelo identifica qué segmentos del texto son más importantes al interpretar o generar cada palabra, uno de los avances más significativos en el procesamiento de lenguaje computacional.

La autoatención viene operando a partir de la producción de tres vectores correspondientes a cada elemento de la secuencia mencionada: la consulta, la clave y el valor. De este modo, la consulta de una palabra es comparada con las claves del resto de las palabras para determinar cuál es su relevancia en el contexto. Acto seguido, los valores son ponderados en función de esas puntuaciones de atención, de ahí que obtengamos la codificación final

El hecho de que sea "multi-cabeza" implica que se están ejecutando varios procesos de atención en paralelo, cada uno de ellos dirigido a aspectos diferentes de las relaciones entre palabras. Algunas de las cabezas pueden estar dedicándose a relaciones gramaticales, otras a relaciones conceptuales o a correlaciones a más larga distancia. Al final, los resultados de todas las cabezas son combinados y refinados, lo que permite una mejor comprensión del texto.

A diferencia de arquitecturas previas, con este modelo se obtienen considerables ventajas: permite el descubrimiento de relaciones de larga distancia del texto sin las limitaciones que suponen las redes secuenciales, procesa eficientemente secuencias de longitud arbitraria, y otorga la posibilidad de un entrenamiento totalmente paralelo. Estos aspectos han hecho que se puedan construir modelos de gran tamaño y potencia, lo que ha supuesto un cambio radical en cómo se trata el lenguaje.

2.3.1.3 Reconocimiento Automático del Habla (ASR)

El Reconocimiento Automático del Habla se define como el "método para obtener la transcripción del discurso, entendido como una secuencia de palabras, a partir de la forma de onda sonora", según Alharbi et al. (2021). En esencia, el ASR convierte señales auditivas en texto escrito, permitiendo que las máquinas perciban y comprendan el habla. Esta tecnología ha evolucionado notablemente en los últimos años, desde sistemas limitados a vocabularios reducidos hasta arquitecturas capaces de transcribir conversación fluida en múltiples idiomas.

- ***Elementos del Sistema ASR***

El preprocesamiento del audio marca el inicio del proceso, en el que se trata la señal sonora original para aislar las características relevantes. Incluye la reducción de ruido ambiental mediante técnicas de filtrado, la normalización del volumen para compensar variaciones, y la

segmentación en fragmentos breves, generalmente de 20 a 40 milisegundos. Este preprocesamiento mejora directamente la precisión de las etapas posteriores.

La extracción de características se entiende como el proceso por el que se obtienen representaciones numéricas del audio manteniendo la información clave para detectar el habla. Históricamente, los Coeficientes Cepstrales en Escala Mel eran los que predominaban, que reflejan propiedades espectrales del sonido; clases más recientes en cambio utilizan representaciones frecuenciales más refinadas o incluso extraen características de la onda de sonido directamente a través de capas neuronales, prescindiendo del diseño de características manualmente.

El modelo acústico relaciona las características sonoras con fonemas o palabras. En los enfoques clásicos se incluían los Modelos Ocultos de Markov junto con las Mezclas Gaussianas para extraer transiciones y relaciones entre los estados sonoros. En las variantes más actuales, emplean las redes neuronales profundas, en especial las convolucionales y recurrentes, para extraer patrones sonoros más complejos y precisos.

El modelo de lenguaje brinda información acerca de la probabilidad de diferentes secuencias de palabras, orientadas hacia la más plausible en relación con las alternativas acústicamente similares. A pesar de que un tiempo los n-gramas fueron la forma dominante de representar el modelo de lenguaje, hoy son superados por sistemas neuronales que captan relaciones más extensas y patrones del lenguaje de mayor complejidad. Finalmente, el decodificador integra la información acústica y lingüística para producir la transcripción textual final, explorando la secuencia de palabras más probable.

- ***Whisper de OpenAI***

Whisper es un sistema "basado en transformers, que ha ganado relevancia por su utilidad en tareas de reconocimiento de voz en escenarios con interferencias o recursos limitados", según Ahlawat et al. (2025). Whisper se distingue en el ámbito del ASR por su fortaleza ante situaciones adversas, algo que ha sido históricamente un desafío para los otros sistemas de reconocimiento de la voz.

Su arquitectura es un esquema codificador-decodificador de tipo transformers. El codificador es el encargado de procesar las características del audio, en concreto espectrogramas log-Mel que representan el espectro de frecuencias; y el decodificador va generando la transcripción de forma incremental. Este diseño le permite identificar relaciones sonoras complejas y

producir transcripciones precisas, apoyándose en mecanismos de atención para priorizar los fragmentos más relevantes del audio.

En su entrenamiento, Whisper utiliza "supervisión débil junto con un manejo simplificado de los datos previos", lo que le permite generar transcripciones directas sin requerir normalizaciones complejas. El modelo se entrenó con 680.000 horas de audio multilingüe supervisado, recopilado de internet, exponiéndolo a una gran diversidad de tonos, calidades de audio, ruidos de fondo y situaciones distintas. Esta amplitud de entrenamiento le confiere una gran resistencia a situaciones adversas.

Whisper tiene unas propiedades que lo hacen diferenciarse de otros sistemas de ASR. Su habilidad multilingües, que le permite transcribir en 99 idiomas y traducir de varios idiomas al inglés, facilita su uso a nivel mundial. Es tolerante a la interferencia, ya que es capaz de funcionar bastante bien en ruido de fondo, acentos diferentes y audios irregulares. Más allá de la simple transcripción, ofrece funciones adicionales como identificación del idioma, marcado temporal y traducción, sin necesidad de módulos independientes. Su implementación sencilla facilita su despliegue, al prescindir de procesos complejos de preparación o normalización del texto. Además, su capacidad de generalización le permite adaptarse a contextos nuevos sin necesidad de reentrenamiento específico.

En el marco de este proyecto, Whisper se posiciona como la opción más adecuada para transcribir sesiones universitarias, caracterizadas por vocabulario técnico especializado, múltiples voces de docentes y estudiantes con estilos distintos, interferencias del entorno como ruido de mobiliario, conversaciones paralelas o dispositivos electrónicos, y diferencias en la calidad de captura según el equipo utilizado. Su capacidad de manejar estas condiciones sin necesidad de ajustes intensivos lo convierte en una herramienta idónea para aplicaciones educativas.

2.3.1.4 Aplicación Web

Una aplicación web es un programa que se ejecuta en un servidor remoto y se accede a través de un navegador. A diferencia del software instalado localmente, este tipo de aplicaciones no requiere instalación en el dispositivo del usuario y puede consultarse desde cualquier lugar con conexión a internet, lo que las convierte en una opción adecuada para entornos educativos que requieren acceso desde distintos dispositivos y ubicaciones.

- ***Estructura de la Aplicación Web***

Como señala Ilkun (2023), la "capa de lógica de negocio constituye el núcleo de todo sistema web", al contener el código que ejecuta las operaciones esenciales y procesa la información de entrada. Las plataformas web actuales dividen responsabilidades en capas diferenciadas, lo que facilita el mantenimiento, la escalabilidad y las actualizaciones del sistema.

El patrón de tres capas define el estándar en las aplicaciones web modernas. La capa de presentación o frontend hace referencia a la interfaz que manipulan directamente los usuarios mediante el navegador. Se implementa conforme a una serie de tecnologías como HTML5 para la estructuración del contenido, CSS3 para la presentación y adaptabilidad así como JavaScript para aportarle interactividad. Los frameworks actuales como React, Vue.js o Angular permiten una aceleración del desarrollo de las interfaces con layouts que se adaptan a diferentes tamaños de pantalla o dispositivos.

La capa de lógica de negocio o backend recoge el procesamiento central. Surte las peticiones del cliente, ejecuta el procesamiento de la PLN y de la ASR y gestiona el flujo entre la interfaz y el almacenamiento. Puede implementarse en distintos lenguajes de programación; en este proyecto se optó por Python, gracias a frameworks como Flask o Django y a su amplio ecosistema de librerías de inteligencia artificial y procesamiento de lenguaje. Otras alternativas incluyen Node.js, Java o PHP, según las necesidades específicas del proyecto.

De su parte, la capa de almacenamiento o base de datos guarda y gestiona los datos persistentes de manera organizada y eficiente y puede elegir para ello sistemas tales como PostgreSQL o MySQL, adecuados para la información estructurada, bien definida y orientada a filas, o bien elegir por alternativas NoSQL como MongoDB o Redis, que permiten una mayor flexibilidad para la información en continua evolución o de tipo híbrido. La elección vendrá determinada por las necesidades de escalabilidad, consistencia y naturaleza de los datos.

2.3.1.5 Gestión de Datos Persistentes

El manejo adecuado de la información requiere "Sistemas de Gestión de Bases de Datos (DBMS)", que "cumplen un rol central en el almacenamiento masivo de datos", tal como expone Aydın (2025). Estos sistemas deben ofrecer "capacidad de expansión y eficiencia en las consultas", características indispensables para que la plataforma funcione con fluidez ante el crecimiento de usuarios y volúmenes de información.

- ***Sistemas Relacionales***

Estos sistemas organizan los datos en tablas de filas y columnas, relacionando tablas mediante claves primarias y foráneas. Resultan óptimos para información estructurada con patrones bien definidos, donde las relaciones entre los componentes son estables y predecibles. Este modelo ha dominado el sector por su rigor conceptual y sus garantías de consistencia.

Entre sus características principales se encuentran la integridad relacional, garantizada por restricciones que mantienen la coherencia entre tablas, y el soporte de operaciones ACID: Atomicidad, que garantiza que las acciones se completen en su totalidad o no se ejecuten; Consistencia, que asegura la transición entre estados válidos; Aislamiento, que evita interferencias entre operaciones concurrentes; y Durabilidad, que preserva los cambios confirmados incluso ante fallos. Emplean SQL como lenguaje estándar de consulta, un lenguaje declarativo robusto y ampliamente difundido, que resulta apropiado para relaciones complejas que requieran uniones y búsquedas complejas.

Las más comunes son PostgreSQL, valorado por su robustez y adherencia a los estándares, MySQL, habitual en ambientes web y SQLite, liviano, ideal para pequeños proyectos o bien aplicaciones embebidas. En este trabajo este tipo de sistemas se encuentran en un ámbito de utilización válido para almacenar datos estructurados como los perfiles de usuarios, con sus credenciales, la información de los audios subidos (marcas de tiempo, duración), las transcripciones asociadas a sus archivos de origen y los resúmenes generados con sus metadatos y fechas.

- ***Sistemas No Relacionales***

Estos sistemas ofrecen un "diseño flexible que elimina las restricciones estrictas de estructura", según Aydın (2025). Se adaptan a contenidos no estructurados o híbridos que cambian con el tiempo, permitiendo ajustes rápidos sin migraciones complejas.

Sus principales variantes incluyen las bases de datos documentales, que almacenan en formatos JSON o BSON como MongoDB, adecuadas para estructuras anidadas y variables. En cuanto a las de clave-valor, entre las que podemos encontrar Redis, son más adecuadas para almacenar temporalmente sesiones donde debe ser posible recuperar rápidamente el acceso mediante identificadores únicos. Respecto a las columnares, entre las que encontraríamos Cassandra, ellas priorizan las columnas sobre las filas para hacer análisis masivos y realizar consultas de selecciones. Para acabar, respecto a las bases de datos de grafos, entre ellas estaría Neo4j, las

cuales enfatizan las relaciones entre los datos, y son idóneas para las redes, o sistemas de recomendación donde las relaciones son tan importantes como los datos en sí.

En esta propuesta, estos sistemas podrían utilizarse para metadatos variables de audios en distintos formatos, o para optimizar el almacenamiento en caché de resultados frecuentes de transcripciones y resúmenes. Aun así, dada la naturaleza mayormente estructurada de los datos del sistema, se prioriza un enfoque relacional.

2.3.2 Método Analítico-Sintético en la Educación

Este eje reúne los fundamentos pedagógicos del proyecto: el método analítico-sintético como estrategia de enseñanza, su vínculo con la generación automática de resúmenes, y los avances de la inteligencia artificial aplicados a la pedagogía universitaria. Mientras el primer eje describe las herramientas tecnológicas del sistema, este segundo eje describe el dominio educativo específico al que esas herramientas se aplican.

2.3.2.1 Método Analítico-Sintético

Tal como exponen MS, Rachmadtullah e Iasha (2021), el método analítico-sintético se define como un "planteamiento pedagógico respaldado por ejemplos que incorporan aspectos estructurales de análisis y síntesis". En el ámbito educativo, esto se traduce en una secuencia en la que el docente presenta la oración completa, procede a descomponerla en sus partes constitutivas y, finalmente, la recompone en su forma original. Aunque este enfoque tiene una larga tradición, las herramientas de procesamiento de lenguaje lo actualizan y automatizan en la actualidad.

- ***Etapa de Análisis***

En esta etapa se descompone el material hasta sus unidades más básicas, lo que facilita su comprensión. Según describen MS, Rachmadtullah e Iasha (2021), "la oración completa sirve como base para el estudio y se divide en unidades menores llamadas palabras, extendiendo esta división hasta las letras, la unidad mínima". De esta forma, esta manera ordenada de fragmentar el discurso ofrece la posibilidad de identificar y localizar aquellos elementos que pueden considerarse esenciales para su propio discurso.

Si se aplica a este mismo proyecto, el análisis se verifica mediante un proceso de PLN que puede considerarse el que descompone la grabación debidamente transcrita en sus componentes, en sucesivas etapas. Por un lado, la segmentación del texto en oraciones

fragmenta el texto en bloques semánticos, mientras que la tokenización divide cada sección, que equivale a una oración, en palabras o unidades individuales para su posterior análisis. Como una segunda idea releva las ideas centrales, se puede recurrir a distintas métricas, como la del TF-IDF, que establece la importancia de las palabras en función de la frecuencia local y de que pueden llegar a ser infrecuentes de forma global, o bien a los análisis de n-gramas que subrayan las palabras o combinaciones de palabras clave.

La extracción de expresiones clave, busca encontrar conjuntos de palabras que nos transmitan las ideas principales del contenido de la clase; el análisis gramatical, en el que se representa la estructura de las oraciones y en el que se puede ver tanto las relaciones con las ideas como su jerarquía; y, el análisis semántico, en el cual se interpreta términos y construcciones en un contexto dado y se hace por medio de representaciones vectoriales y de los modelos de lenguaje que nos puedan dar, en la reconstrucción, matices y regiones o relaciones de significado.

En cuanto a los sistemas de procesamiento de textos, este análisis busca "evaluar la correspondencia entre los vectores generados y el original, dando un mayor peso a las secciones clave", como indican Li et al. (2023). Los LLM actuales son muy sofisticados con esta tendencia, incluso identifican por sí mismos, de manera autónoma y sin reglas predefinidas, los elementos importantes, ya que están provistos para ello por los conocimientos lingüísticos y contextuales que han aprendido durante su entrenamiento.

- ***Etapa de Síntesis***

La síntesis ocurre cuando "los elementos identificados regresan a su configuración inicial, recomponiendo los componentes originales", según MS, Rachmadtullah e Iasha (2021). En la generación automatizada, esto permite que los sistemas "generen palabras y construcciones nuevas que compongan el resumen, ancladas en la fuente original", combinando el conocimiento previo para preservar la información, como detallan Li et al. (2023). Esta reconfiguración es muy relevante a la hora de elaborar resúmenes coherentes y funcionales.

La síntesis, en este caso, reordena las ideas que se han destacado durante el análisis, en una forma temporal, mediante mecanismos organizados; la agrupación por temas reune ideas semejantes en bloques coherentes, instalando el eje temático de la sesión; la construcción de oraciones genera enunciados nuevos que conectan con vigor todos los conceptos troncales de forma clara y precisa, volviendo a formular los contenidos cuando es necesario para conseguir fluidez.

La construcción de relaciones lógicas hace uso de conectores y de formas sintácticas adecuadas para vincular ideas, con un barrio de recorridos, etc. La producción del resumen genera que existe un discurso que conviene retener como representación original de la substancia del contenido sin que tenga que ver con la extensión. Por último, el ajuste de la extensión hace que el resultado sea aquella que necesita, intentado equilibrar síntesis/exhaustividad, etc.; no parte de la discusión dejando por el camino el atropello de información.

Los modelos GPT destacan en esta etapa por su capacidad generativa. Reorganizan la información de forma creativa, reformulan ideas conservando su esencia y producen texto nuevo que mantiene la coherencia y el sentido del material original. Precisamente esta capacidad de reconstrucción, creativa pero fiel a la fuente, es lo que hace valiosos a los LLM en la elaboración de resúmenes académicos de calidad.

- ***Ventajas Educativas del Método***

Una comprensión cabal del tema representa uno de los principales ejemplos de los beneficios pedagógicos del enfoque analítico-sintético. De acuerdo a Gavronskaya et al. (2022) existiría una sólida interiorización del conocimiento si esta se produce en tres fases: la evocación que restaura la información recordada; la interpretación que da sentido a esa información; y la reflexión que a continuación establece una conexión dinámica que permite consolidar los nuevos conceptos a partir del conocimiento previo. Una comprensión plena es efectuada por el alumno cuando "supera lo superficial y accede al nivel deductivo", cuando realiza la interiorización del mensaje y finalmente extrae inferencias propias, tal y como mencionan Medranda-Morales et al. (2023).

El desarrollo del pensamiento analítico surge como otro beneficio esencial. Gavronskaya et al. (2022) lo describen como un "proceso autodirigido que surge del desarrollo de habilidades como interpretar, examinar, evaluar y explicar". Implica "reflexión y lógica para decidir y resolver problemas basados en hechos", según Medranda-Morales et al. (2023). Sus indicadores incluyen delimitar el problema con precisión, evaluar la credibilidad de las fuentes, registrar observaciones de forma metódica y objetiva, y tomar decisiones fundamentadas en el análisis, tal como exponen Gavronskaya et al. (2022).

En el marco de este proyecto, si bien el sistema automatiza el análisis y la síntesis, esto potencia el razonamiento analítico en los estudiantes en lugar de eliminarlo. Al entregar resúmenes organizados, la plataforma libera tiempo y esfuerzo cognitivo que los estudiantes pueden

dedicar a tareas de mayor nivel, como cuestionar las ideas presentadas, comparar contenidos entre sesiones o fuentes distintas, integrar lo nuevo con su conocimiento previo, y formular conclusiones propias sobre la aplicación del contenido.

2.3.2.2 Síntesis Automatizada

La síntesis automatizada, o generación automática de resúmenes, es una técnica de PLN que condensa documentos extensos conservando la información esencial. En la actualidad, saturada de contenido textual, este enfoque responde a una necesidad concreta: el volumen de información disponible supera la capacidad humana de procesarlo por completo. Existen dos enfoques principales, que se diferencian en cómo generan las versiones resumidas.

- ***Síntesis Extractiva***

Esta modalidad selecciona oraciones o fragmentos importantes del texto original y los combina para formar el resumen. Selecciona y reordena partes ya existentes del texto en lugar de generar contenido nuevo, de forma similar a subrayar las secciones clave de un documento y limitarse a ellas en la revisión.

Entre sus técnicas computacionales se hallan: por un lado, los algoritmos basados en grafos, como TextRank que representan las frases como nodos de un gráfico en el que los enlaces indican afinidad sobre el nivel conceptual, concediendo especial peso a las frases más interconectadas; por otro lado, los métodos estadísticos que aplican TF-IDF para establecer una medida de la importancia de las frases en función de los términos relevantes que contienen; y, en el caso del aprendizaje automático supervisado, que aprender a predecir qué frase debe ser incluida en el resumen, con ejemplos previamente marcados por personas

Sus principales ventajas residen en, por un lado, la conservación de las debidas construcciones gramaticales de las que dispone el texto, y por otro lado, la reducción de los errores introducidos en cuestión de los hechos, pues no generan contenido nuevo. Por contra, sus desventajas son: que los resúmenes generados pueden no tener coherencia a causa de los saltos en la lectura al ir extrayendo fragmentos de diferentes partes del texto, y que son muy limitadas para integrar relaciones complejas entre ideas distantes que implican una fusión conceptual.

- ***Síntesis Abstractiva***

Esta variante genera texto nuevo que reformula y condensa el contenido original, similar a la forma en que una persona parafrasea con sus propias palabras. Requiere una comprensión

profunda del contenido y capacidades generativas, lo que representa un reto técnico mayor que el enfoque extractivo.

Sus métodos han evolucionado de forma considerable. Modelos secuencia a secuencia: la entrada se codifica en vectores que también se decodifican para dar la salida. Los mecanismos de atención dirigen la atención hacía partes del texto durante la generación. Los transformers, por ejemplo, BERT, GPT, T5 o BART son el estado del arte actual; construcciones que pueden capturar relaciones complejas y producen buenas salidas.

Entre sus ventajas se pueden citar resúmenes más naturales o fluidos, similares a los que puede componer una persona; la integración de información dispersa en ideas sintetizadas; o su flexibilidad para variar la longitud del resumen o la atención hacia ciertas partes según se desee. Entre sus desventajas, podríamos destacar el riesgo de incluir información errónea o inventada; o la mayor complejidad y necesidad de recursos computacionales más intensos para su generación y funcionamiento.

En este proyecto se prioriza el enfoque abstractivo mediante LLM como GPT y Llama, por sus resultados más naturales y útiles para los estudiantes, incorporando mecanismos de verificación para reducir el riesgo de información no verificada.

- ***Evaluación de la Calidad de los Resúmenes***

La evaluación de los resúmenes automáticos abarca distintos aspectos que reflejan su calidad. La coherencia y la cohesión determinan en buena medida la calidad de un resumen: uno bueno debe presentar una "organización lógica, con una transición clara entre ideas de una oración a la siguiente", según plantean Wu et al. (2025). De igual forma, la cohesión requiere que el texto sea "correcto gramaticalmente, con una estructura sólida y sin redundancias", según agregan Wu et al. (2025). Sterling et al. (2024) añaden que la evaluación debe considerar la "fiabilidad, exhaustividad informativa, un nivel de lectura accesible, claridad y neutralidad".

Hay herramientas automáticas de medida que aceleran o ponen en práctica el análisis sistemático de estos aspectos, por ejemplo ROUGE (Recall-Oriented Understudy for Gisting Evaluation), que compara mediante n-gramas la salida del resumen y las versiones de referencia generadas por personas, midiendo también la parte de contenido que se conserva; BLEU (Bilingual Evaluation Understudy), medida que fue concebida para la traducción automática y que aquí se utiliza para medir cómo de correcta es la salida en función de los n-gramas; o bien BERTScore que utiliza representaciones contextuales de BERT para estimar

las semejanzas semánticas con el texto de referencia, o cómo de semejantes son conceptualmente textos diferentes en función de las coincidencias literales detectadas.

Pero la evaluación humana es fundamental para un diagnóstico completa; los evaluadores analizan la coherencia, ya que sus argumentos van avanzando de manera lógica, la relevancia, ya que se abordan las temáticas más importantes, la fluidez, ya que la lectura es natural y placentera, y la fidelidad, ya que los datos son ciertos y reales en relación con el ámbito de estudio. Aunque es más costosa y demorada que las métricas automáticas, esta evaluación capta matices cualitativos que los algoritmos no logran detectar.

- ***Preservación de la Información Esencial***

Para preservar la esencia del contenido original, es necesario verificar la fidelidad factual del resumen. Esto implica "comprobar que cada elemento del resumen esté respaldado por evidencia del material original, sin invenciones ni añadidos infundados", conforme a Wu et al. (2025). La incorporación de conocimiento previo en el decodificador contribuye a "reducir las omisiones derivadas del procesamiento secuencial", como detallan Li et al. (2023). Esta verificación adquiere especial relevancia en entornos académicos, donde la exactitud determina el valor del contenido.

El hecho de que la conservación de los aspectos específicos del contexto educativo responde estrictamente a las demandas del contexto educativo, pues los estudiantes se apoyan en la validez de la información para aprender, en este proyecto, quiere decir que los resúmenes han de conservar las ideas principales que continúan el tema, las explicaciones del docente imprescindibles para dominar el área de conocimiento, las relaciones que intervinieron entre los conceptos y que comentan sus conexiones, así como los ejemplos que demuestran las aplicaciones de la teoría en situaciones concretas, pero sin alterar el discurso académico mismo ni exteriorizar información propia al aula grabada.

2.3.2.3 Avances Tecnológicos en Pedagogía

La Inteligencia Artificial Educativa (EAI, por sus siglas en inglés) "asiste a los docentes en sus estrategias de enseñanza, enriquece la experiencia de los estudiantes y promueve la renovación del entorno educativo", conforme a Kökver et al. (2025). Facilita la "automatización de tareas intelectuales complejas", como la "identificación de vacíos de conocimiento" y la elaboración de "materiales adaptados a perfiles individuales". La inteligencia artificial se presenta así como un impulsor para resolver problemas educativos persistentes.

- ***Ventajas de la IA en el Ámbito Educativo***

La personalización del aprendizaje permite que los algoritmos ajusten contenidos y ritmos según las características de cada estudiante, reconociendo diferencias en velocidad y estilo de aprendizaje. Los sistemas evaluativos pueden detectar áreas deficitarias y proporcionar apoyo extra de manera automática.

La automatización de las tareas administrativas minimiza la carga de los docentes, dado que permite, por ejemplo, la automatización de correcciones de evaluaciones de tipo objetivo, la creación de materiales como tareas o similares. Esto les permite al docente usar su tiempo para interactuar con sus estudiantes o diseñar experiencias de aprendizaje que les resulten relevantes.

El análisis de datos educacionales pone de manifiesto el progreso de estos estudiantes; en ese sentido, permite identificar cuales son las dificultades comunes y patrones de interacción con los recursos, información que ayudará en la toma de decisiones para la mejora de los cursos.

La inclusión se entiende: los mecanismos de transcripción de audio permiten el acceso a estudiantes con dificultades auditivas; por otro lado, los servicios de traducción automática favorecen el acceso a estudiantes de diferentes idiomas, lo que les permite acceder a la educación.

El soporte continuo se manifiesta en asistentes virtuales que responden a preguntas fuera del horario de clase y que permiten la resolución de preguntas frecuentes; estas herramientas no sustituyen la interacción humana en aquellos temas que exigen mayor nivel de sensibilidad.

- ***Repercusiones en el Entorno Universitario***

En el nivel de educación superior, este tipo de herramientas muestra efectos notables. Si bien las cifras de Ghorbian y Ghobaei-Arani (2026), que reportan una mejora del 33% en la "detección temprana de trastornos psicológicos" y del 27% en "eficacia terapéutica", corresponden al ámbito de la salud mental, ilustran el potencial transformador de la inteligencia artificial en contextos donde el bienestar y el rendimiento académico están relacionados.

La solución propuesta en este trabajo constituye una revolución en la sintetización del contenido académico, un procedimiento que requiere comprensión, análisis y capacidad de integración. Agiliza el repaso a través de textos y resúmenes que pueden ser interpreted por los

estudiantes a su propio ritmo: esto es, combinar el formato auditivo de las grabaciones, el formato visual de las transcripciones y el formato esquemático de los resúmenes.

Acelera el repaso ya que los puntos son enfatizados por el trabajo. Y cuando es usado por los estudiantes es una optimización del tiempo de estudio redirigiendo el esfuerzo entendible y/o un esfuerzo de registrar notas y/o escuchar el audio de manera regular en estructuras constitutivas de ensayo, hacia la reflexión y la aplicación del supresor de contenido aprendido. Y también contribuye a los diferentes tipos de estilos de aprendizaje. En definitiva, apoya a aquellos que prefieren leer, así como a los que tienen limitaciones sensoriales.

2.4 Metodología en Cascada

El Modelo en Cascada es una metodología de desarrollo de software que organiza el proceso en fases secuenciales, donde cada etapa debe completarse en su totalidad antes de iniciar la siguiente. Propuesto originalmente por Royce (1970) a modo de descripción de las prácticas en el contexto de la industria del software de la época, ello terminó por consolidarse como el modelo de referencia más habitual para proyectos con definición bien concretada desde su inicio.

Las cinco fases que tiene son: Análisis, en la que se identifica y se documenta, los requisitos del sistema; Diseño, en la que se define la arquitectura, la base de datos y la interfaz que van a permitir satisfacer esos requisitos; Implementación, en la que se codifican los componentes definidos en diseño; Verificación, en la que se prueba el sistema para delegar que cumple los requisitos especificados; y Mantenimiento, en la que se corrigen errores y se llevan a cabo ajustes tras la puesta en marcha.

Su claridad estructural que permite planear y seguir el avance del proyecto; la amplia documentación que genera en cada fase y su suficiente para proyectos académicos y de mantenimiento extendido del producto; su idoneidad para equipos de desarrollo pequeños o para un único integrante donde la coordinación de iteraciones paralelas es menos factible que en un equipo grande; en contraposición, su principal desventaja es la rigidéz ante los cambios de requerimientos una vez que el proyecto ha avanzado, ya que retroceder a fases anteriores implica un coste superior al momento que en modelos iterativos.

Este proyecto adopta el Modelo en Cascada porque el alcance funcional del sistema quedó completamente definido desde la fase de diagnóstico, documentada en el Capítulo III, y porque el desarrollo está a cargo de un único responsable dentro de un período fijo de seis meses,

condiciones bajo las cuales este modelo resulta más apropiado que alternativas iterativas como Scrum, que están pensadas para equipos que ajustan el alcance del producto sobre la marcha. Las cinco fases del modelo estructuran directamente la propuesta desarrollada en el Capítulo IV.

2.5 Conclusiones del Marco Teórico

El Procesamiento de Lenguaje Natural ha evolucionado de forma notable desde su surgimiento en la década de los 50 y se ha consolidado como un eje tecnológico del mundo actual, que nos proporciona modelos como BERT y GPT, que son sin duda el estado del arte en modelos lingüísticos de gran escala, con capacidades sin precedentes para ambos interpretar y generar lenguaje humano. Con una arquitectura basada en el Transformer y en mecanismos de autoatención, son perfectos para sistemas de resumen automático para entornos educacionales dado que combinan una buena comprensión del lenguaje y capacidades generativas sofisticadas.

Respecto al Reconocimiento Automático del Habla, el modelo Whisper de OpenAI destaca por su precisión y fortaleza, lo que lo hace perfectamente válido para hacer la transcripción de sesiones universitarias en situaciones habituales que incluyen la interferencia del contexto, vocabulario técnico y la superposición de varias voces. Su estrategia de entrenamiento con supervisión débil, su capacidad multilingüe y su resistencia a condiciones adversas lo posicionan como la herramienta más adecuada para este proyecto, superando las limitaciones de los enfoques tradicionales de ASR.

El método analítico-sintético, con raíces en tradiciones pedagógicas del siglo XIX, encuentra una versión actualizada a través de las innovaciones en PLN. Los LLM demuestran eficacia al descomponer documentos en sus ideas esenciales mediante análisis automatizado, para luego recomponerlas en resúmenes unificados mediante generación de texto. Esta traducción digital de una estrategia pedagógica consolidada busca fortalecer la comprensión profunda y el juicio analítico de los estudiantes, liberando recursos cognitivos para tareas de mayor nivel.

La estructura de las aplicaciones web, inspirada en el patrón de tres capas y materializada con frameworks como Flask que priorizan la simplicidad, ofrece una base sólida para integrar los módulos de PLN y ASR. Al separar responsabilidades, entre la capa de datos, la interfaz de usuario y la lógica de negocio, se favorece un desarrollo ordenado, un mantenimiento ágil y una escalabilidad viable del sistema completo.

En la creación de las interfaces de usuario accesibles, eminentemente en herramientas formativas y para un público que tiene grados de adopción tecnológica muy distintos, es importante aplicar condiciones de simplicidad para evitar la sobrecarga cognitiva de los usuarios, claridad para permitir que las opciones aparezcan sin ambivalencias, consistencia para reducir los tiempos de aprendizaje, retroalimentación para que el usuario obtenga la confirmación de que se ha realizado la acción correspondiente a través de un mismo proceso, inclusión para tener en cuenta la diversidad en las capacidades funcionales, así como la parecida adaptabilidad a los diferentes dispositivos. La plataforma puede en este caso incrementar el grado de satisfacción del usuario general y facilitar el tránsito en alumnos y docentes cuando se adopte la herramienta.

Los sistemas de almacenamiento de información, ya sean relacionales con sus garantías de solidez y consistencia mediante ACID, o NoSQL con su flexibilidad para contenido no estructurado, ofrecen alternativas complementarias para gestionar los datos del sistema. La elección más adecuada dependerá de las necesidades de escalabilidad, del grado de estructura de los datos, y de la eficiencia requerida en las consultas. Para esta propuesta, los sistemas relacionales resultan más adecuados dado el carácter mayormente estructurado de la información.

Por último, las innovaciones relacionadas con el desarrollo de la inteligencia artificial aplicada a la pedagogía son capaces de rentabilizar las técnicas de enseñanza y aprendizaje en la educación universitaria ya que la automatización de procesos intelectuales complejos, como la síntesis de material académico, da lugar a prácticas formativas de mayor valor: el análisis, la reflexión, la discusión, así como la aplicación creativa del conocimiento, cambiando el modelo educativo que prevalece en las dinámicas actuales, y que también en el contexto de la educación superior, se puede resumir en la transmisión de información.

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Introducción

La modalidad cuantitativa permite verificar hipótesis desde una perspectiva probabilística: si los datos la respaldan, es posible derivar de ella conclusiones generales con un grado de certeza definido. La modalidad cualitativa, en cambio, busca interpretar el fenómeno de estudio más allá de su medición numérica (Hernández-Sampieri y Mendoza, 2018). Al combinar ambos enfoques se obtiene un diseño mixto, que recopila y analiza datos cuantitativos y cualitativos dentro de un mismo estudio.

Dado el tipo de proyecto, se adoptó un enfoque cuantitativo-cualitativo: de hecho, a partir del proceso de exploración de estudiantes y docentes de Ingeniería en Tecnologías de la Información de la ULEAM Extensión El Carmen, se optó por realizar una investigación aplicada y de carácter de campo a partir de datos obtenidos a partir de encuestas y entrevistas, los cuales fueron posteriormente analizados mediante métodos inductivos y deductivos, con el objetivo último de describir la problemática existente y justificar el diseño de ULEAMTranscribe, descrito en la sección del Capítulo IV.

La revisión bibliográfica que sostiene el marco conceptual del proyecto, sobre Procesamiento de Lenguaje Natural, modelos lingüísticos masivos y reconocimiento automático de voz, se documenta como fuente secundaria en la sección 3.4 y se desarrolló extensamente durante la elaboración del Capítulo II.

3.2 Tipo de Investigación

3.2.1 Aplicada

La investigación aplicada tiene como fin analizar un problema junto con los hechos que lo rodean, de modo que la información obtenida sea útil para dar solución a una necesidad concreta más allá de ampliar el conocimiento teórico sobre el tema (Baena, 2014).

El proyecto se desarrolló mediante una investigación aplicada, dado que se intervino directamente en el proceso de estudio de estudiantes y docentes de la carrera, mediante el desarrollo de un sistema informático que transcribe y resume automáticamente las clases grabadas, con el objetivo de reducir el tiempo de repaso y facilitar la aplicación del método analítico-sintético en el aula.

3.2.2 De Campo

La investigación de campo tiene como objetivo recolectar y anotar los datos directamente de los objetos de estudio, o del medio donde ocurre el evento, sin alterar la información obtenida. Este enfoque permite obtener una visión auténtica del fenómeno al interactuar directamente con el entorno, condición necesaria para generar hallazgos que no dependan solo de la literatura previa (Hamui-Sutton, 2021).

Mediante encuestas y entrevistas se recopiló información directa de estudiantes y docentes para determinar las dificultades reales que enfrentan al procesar grabaciones de clases. Esto permitió fundamentar el diseño del sistema descrito en el Capítulo IV sobre necesidades verificadas en el propio contexto de la ULEAM Extensión El Carmen, y no sobre supuestos generales.

3.3 Métodos de Investigación

3.3.1 Método Analítico-Sintético

El método analítico consiste en descomponer un objeto de estudio en sus componentes para examinar cada uno por separado e identificar sus comportamientos específicos, un proceso indispensable para desentrañar las causas y la naturaleza de los fenómenos estudiados; el método sintético integra esos componentes ya analizados en un todo coherente (Behar Rivero, 2019). Este método se aplicó tanto al diagnóstico como al propio diseño del sistema: los procesos actuales de estudio de los alumnos se descompusieron en sus componentes básicos (grabación, almacenamiento, búsqueda, repaso), y la arquitectura del sistema web se dividió en módulos independientes (transcripción, extracción de datos, generación de resúmenes, presentación de resultados) que después se integraron en una aplicación cohesiva.

3.3.2 Método Inductivo

El método inductivo parte del análisis de hechos particulares para llegar a un enunciado general. Cuando se produce una conclusión inductiva se forma una ley o teoría, cuya finalidad no es explicar cada caso aislado sino observar la manera en que los casos se relacionan entre sí (Monroy y Nava, 2018).

Este método se empleó para comprender cómo los estudiantes de la ULEAM Extensión El Carmen enfrentan, en casos particulares, la dificultad de procesar grabaciones extensas de clases. A partir de las experiencias individuales reportadas en la encuesta y en las entrevistas,

se identificaron patrones comunes y se derivaron los problemas generales que el sistema debía resolver.

3.3.3 Método Deductivo

La deducción es el proceso lógico que permite interpretar hechos particulares a partir de su integración dentro de un conocimiento general ya verificado (Monroy y Nava, 2018).

Los fundamentos generales del método analítico-sintético y de las técnicas de Procesamiento de Lenguaje Natural, validados en la literatura revisada en el Capítulo II, se aplicaron deductivamente al caso particular de la ULEAM Extensión El Carmen, adaptándolos a las asignaturas técnicas de la carrera y a las condiciones de hardware disponibles para el desarrollo del sistema.

3.4 Fuentes de Información de Datos

Dentro de la ULEAM Extensión El Carmen se identificó que el proceso de estudio a partir de grabaciones de clase lo definen los propios docentes al grabar sus sesiones, y lo ejecutan día a día los estudiantes al intentar repasarlas; el nivel de dificultad de ese proceso se refleja directamente en el rendimiento académico y en el tiempo disponible para otras actividades.

3.4.1 Fuente Primaria - Encuesta-Cuestionario

La encuesta es un instrumento que permite estandarizar la recolección de información sobre variables específicas y facilita su posterior análisis estadístico (Hernández-Sampieri y Mendoza, 2018).

Se aplicó la encuesta a estudiantes de Ingeniería en Tecnologías de la Información de nivel intermedio y avanzado, con el fin de reconocer sus hábitos de estudio con grabaciones de clase e identificar, de forma cuantitativa, las dificultades que enfrentan y sus expectativas frente a una herramienta automatizada de generación de resúmenes.

3.4.2 Fuente Primaria - Entrevista-Guía de Entrevista

La entrevista permite una comunicación flexible y abierta, orientada a construir una visión compartida sobre la problemática estudiada, y captura matices que una encuesta cerrada no puede recoger (Hernández-Sampieri y Mendoza, 2018).

Se entrevistó a diez docentes que imparten asignaturas técnicas en la carrera, a fin de recolectar información sobre la estructura de sus clases, los conceptos que consideran esenciales, la terminología técnica que emplean, y su perspectiva pedagógica sobre el método analítico-sintético y su posible automatización.

3.4.3 Fuentes Secundarias

Las fuentes secundarias comprenden la revisión bibliográfica de literatura especializada sobre Procesamiento de Lenguaje Natural, modelos lingüísticos masivos, reconocimiento automático de voz e inteligencia artificial en educación, desarrollada extensamente en el Capítulo II a partir de artículos indexados, documentación técnica oficial de los modelos empleados (faster-whisper, Llama), e investigaciones previas sobre aplicaciones similares en contextos educativos.

3.5 Estrategia Operacional para la Recolección de Datos

3.5.1 Población

Una población es una agrupación de elementos o personas que comparten características comunes relevantes para la investigación (Asti Vera, 2015).

La población objetivo de este estudio la constituyeron estudiantes y docentes de la carrera de Ingeniería en Tecnologías de la Información de la ULEAM Extensión El Carmen que participan en asignaturas técnicas avanzadas durante el período académico 2025-2026: 120 estudiantes activos matriculados desde cuarto semestre en adelante, y 12 docentes que imparten regularmente asignaturas técnicas especializadas.

3.5.2 Muestra

La muestra es la parte del universo total, compuesta por los elementos que se van a utilizar efectivamente en el estudio (Com y Ackerman, 2013).

Dado que la población estudiantil supera las 100 personas, se calculó una muestra representativa con la fórmula estándar para poblaciones finitas, considerando un nivel de confianza del 95% ($Z = 1,96$), un margen de error del 5% ($e = 0,05$), y una proporción estimada $p = 0,5$:

Tabla 1 Cálculo del tamaño de muestra

Procedimiento	**Operación**	**Resultado**
Fórmula general	$n = (Z^2 \cdot p \cdot (1-p) \cdot N) / (e^2 \cdot (N-1) + Z^2 \cdot p \cdot (1-p))$	—
Sustitución de valores	$n = (1{,}96^2 \cdot 0{,}5 \cdot 0{,}5 \cdot 120) / (0{,}05^2 \cdot 119 + 1{,}96^2 \cdot 0{,}5 \cdot 0{,}5)$	—
Cálculo del numerador	$1{,}96^2 \cdot 0{,}5 \cdot 0{,}5 \cdot 120 = 0{,}9604 \cdot 120$	115,248
Cálculo del denominador	$0{,}05^2 \cdot 119 + 0{,}9604 = 0{,}2975 + 0{,}9604$	1,2579
División final	115,248 / 1,2579	91,6
Redondeo	Se redondea hacia arriba para no subestimar la muestra requerida	92 estudiantes

El tamaño de muestra calculado resultó en 92 estudiantes, aproximadamente el 77% de la población estudiantil. Para la población docente, dado que el número total es pequeño (12 profesores), se optó por un censo completo en lugar de un muestreo, invitando a participar a la totalidad de los docentes que imparten asignaturas técnicas.

Los 92 estudiantes fueron seleccionados de un muestreo aleatorio estratificado, que es el procedimiento correspondiente para una población que contiene grupos dentro de ella, ya que puede ser representativa en función de los grupos intermedios. Los estratos estaban determinados por el nivel académico; por estratos, Estrato 1, estudiantes de cuarto y quinto semestre (45% de la población, 41 estudiantes de la muestra); Estrato 2, estudiantes de sexto y séptimo semestre (35%, 32 estudiantes); Estrato 3, estudiantes de octavo semestre y tesistas (20%, 19 estudiantes). Dentro de cada estrato se aplicó selección aleatoria simple sobre las listas de matriculación, de modo que todo estudiante tuviera la misma probabilidad de ser seleccionado dentro de su estrato.

3.5.3 Análisis de las Herramientas de Recolección de Datos

3.5.3.1 Encuesta

El cuestionario aplicado a los estudiantes constó de nueve preguntas organizadas en cuatro secciones temáticas: hábitos actuales de estudio y uso de grabaciones (preguntas 1 y 2),

dificultades experimentadas con los métodos actuales (preguntas 3 y 4), expectativas frente a una herramienta automatizada (preguntas 5, 6 y 7), e intención de adopción y preferencias de plataforma (preguntas 8 y 9). Se utilizaron preguntas de selección única, una de selección múltiple, una de ordenamiento por prioridad, y escalas de Likert de 5 puntos, según la naturaleza de cada variable medida.

3.5.3.2 Entrevista

La guía de entrevista aplicada a los docentes constó de nueve preguntas abiertas, organizadas para indagar primero sobre la estructura típica de sus clases y la terminología técnica que emplean, luego sobre su comprensión y valoración del método analítico-sintético, y finalmente sobre los riesgos, criterios de calidad y disposición a colaborar con la validación del sistema. Las entrevistas se realizaron de forma individual, presencial o por videoconferencia, con una duración de 45 a 60 minutos, grabadas con consentimiento informado del participante.

3.5.4 Estructura de los Instrumentos de Recolección de Datos

La encuesta se implementó en formato digital mediante Google Forms, con una introducción que explicaba el propósito de la investigación, garantizaba la confidencialidad de las respuestas y solicitaba el consentimiento informado de los participantes. Se estimó un tiempo de realización de 12 a 15 minutos.

La guía de entrevistas se desglosaba en un documento que daba lugar a la formulación de preguntas generales sobre el perfil del docente (asignaturas que imparte, cantidad de años de experiencia docente con el alumnado) para iniciar una conversación que fuera más fluida antes de presentar las preguntas más centrales, agrupadas por temática, con una suficiente flexibilidad para abordar diferentes temas que pudieran emerger.

Ambos instrumentos se validaron con una prueba piloto con una muestra pequeña de alumnado y con un docente, previa a su aplicación definitiva, lo que permitió afinar la profundidad de la formulación de las preguntas e igualmente estimar con mayor aproximación el tiempo real de su contestación.

3.5.5 Plan de Recolección de Datos

El plan de recolección de datos se ejecutó de manera sistemática durante un período de cuatro semanas, con el siguiente detalle:

Tabla 2 Plan de recolección de datos

Semana	Actividad	Instrumento	Responsable
Semana 1	Finalización y validación de instrumentos mediante prueba piloto; obtención de aprobaciones institucionales.	—	Matteo Velez
Semana 2	Distribución de la encuesta digital a los estudiantes de la muestra seleccionada, con recordatorios de seguimiento.	Encuesta	Matteo Velez
Semana 3	Continuación de la recolección de encuestas; realización de entrevistas individuales con docentes.	Encuesta y entrevista	Matteo Velez
Semana 4	Cierre de la recolección de encuestas; entrevistas pendientes; depuración de la base de datos.	Encuesta y entrevista	Matteo Velez

La participación final alcanzó el 87% de la muestra estudiantil (80 de 92 estudiantes) y 10 de los 12 docentes invitados, el conjunto de datos analizado en la sección 3.5.6. Durante todo el proceso se mantuvieron los estándares éticos de investigación: consentimiento informado, confidencialidad, anonimato, y uso exclusivo de los datos con fines académicos.

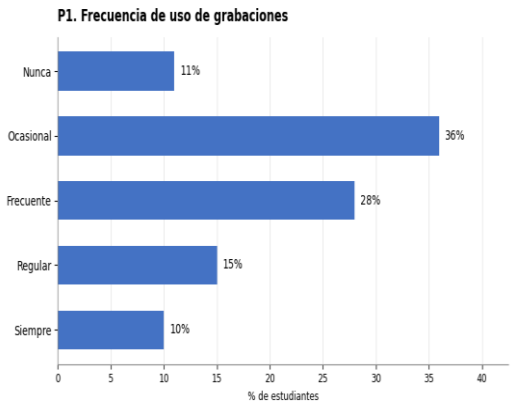
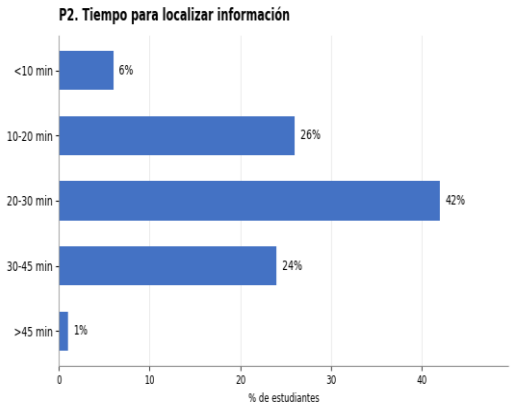
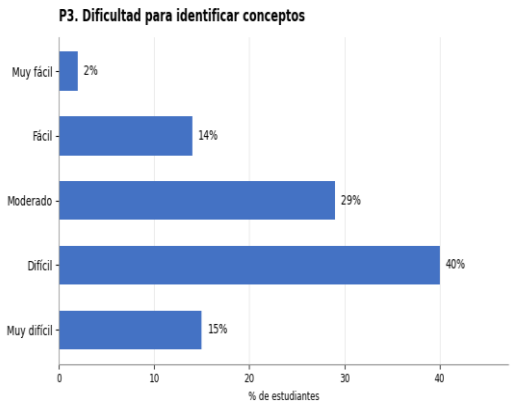
3.5.6 Análisis y Presentación de Resultados

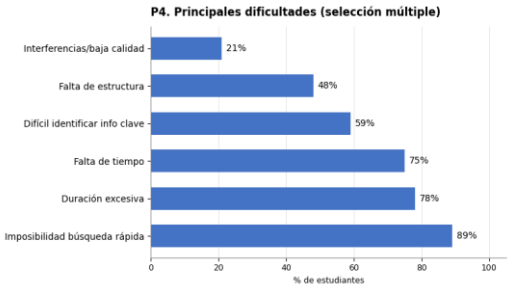
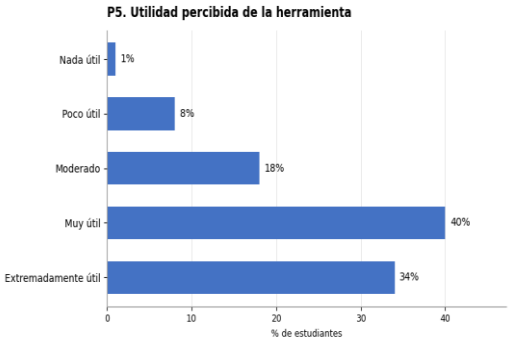
Una vez recolectada la información se procedió con su registro en una hoja de cálculo y su posterior tabulación, que se presenta a continuación.

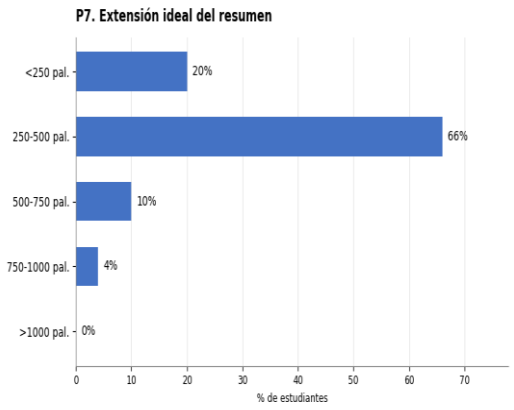
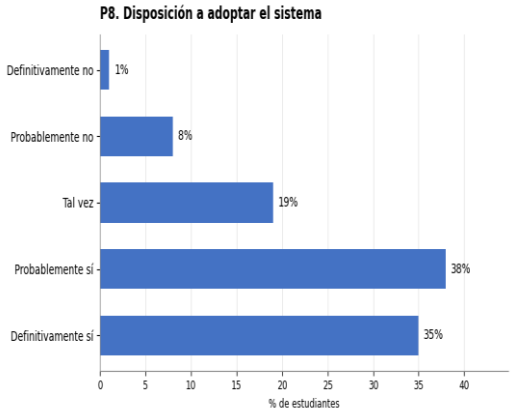
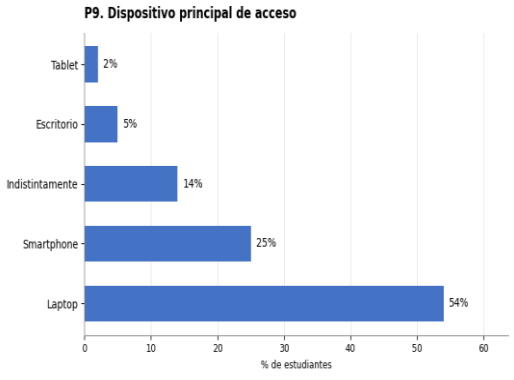
3.5.6.1 Encuesta Aplicada a Estudiantes

Resultados de las 80 respuestas válidas obtenidas sobre la muestra calculada de 92 estudiantes (sección 3.5.1.2).

Pregunta	Gráfico

1. ¿Con qué frecuencia utiliza grabaciones de audio de clases como material de estudio?	
Interpretación: El 52,5% de los estudiantes recurre a grabaciones de clase al menos una vez por semana; solo el 11,2% nunca las utiliza.	
2. ¿Cuánto tiempo invierte en promedio en localizar un tema específico dentro de una grabación?	
Interpretación: El 67,5% de los estudiantes invierte entre 20 y 45 minutos en localizar un tema específico dentro del audio.	
3. ¿Qué tan difícil le resulta identificar los conceptos principales en una clase de 60 minutos o más?	
Interpretación: El 55% de los estudiantes califica la tarea como difícil o muy difícil (media 3,51/5).	

4. ¿Cuáles son las principales dificultades que enfrenta al utilizar grabaciones de clases? (selección múltiple)	
Interpretación: La imposibilidad de búsqueda rápida (88,8%) es la dificultad más señalada, por encima de la duración del audio (77,5%).	
5. Si existiera una herramienta que generara resúmenes automáticos, ¿qué tan útil sería para su estudio?	
Interpretación: El 73,8% la considera muy útil o extremadamente útil; solo el 8,7% la considera poco o nada útil.	
6. ¿Qué características considera más importantes en un resumen automático? (ordenamiento 1-5)	
Interpretación: Identificar bien los conceptos principales y transcribir con precisión son, juntos, elegidos en primer lugar por el 87,6% de los estudiantes.	

7. ¿Cuál sería la extensión ideal de un resumen para una clase de 60 minutos?	
Interpretación: El 66,2% de los estudiantes prefiere un resumen de entre 250 y 500 palabras.	
8. ¿Estaría dispuesto a utilizar regularmente una aplicación que genere resúmenes automáticos?	
Interpretación: El 72,5% se inclina por adoptar la herramienta (probablemente sí + definitivamente sí).	
9. ¿Qué dispositivo utilizaría principalmente para acceder a la aplicación?	
Interpretación: El 53,8% accedería principalmente desde laptop, pero cerca del 40% usaría un dispositivo móvil en algún momento.	

3.5.6.2 Entrevista Aplicada a Docentes

Síntesis de las respuestas de los 10 docentes entrevistados (identificados como D1 a D10 en la sección 3.5.6.2 para preservar su confidencialidad), agrupadas por pregunta.

Pregunta	**Síntesis de los 10 docentes**	**Interpretación**
1. ¿Podría describir la estructura típica de sus clases de 60-90 minutos?	Se observa un patrón idéntico en los diez docentes. La sesión es tipo "teórico-práctica", esto es, hay de 20 a 30 minutos de fundamentación teórica y el resto de la sesión es para práctica técnica (live coding, laboratorio, casos, ejercicios, etc.).	La clase típica de la carrera prioriza la práctica sobre la exposición teórica, un dato relevante para calibrar qué proporción del audio suele ser contenido aplicado.
2. ¿Cuáles son los elementos que los estudiantes deben retener de cada clase?	Los docentes son de la idea que lo realmente importante no es el contenido lineal de las diapositivas sino la lógica de razonamiento y la capacidad para resolver problemas.	El sistema debe favorecer la lógica que une los conceptos pero no simplemente enumerar conceptos sin relación alguna.
3. ¿Qué terminología técnica podría representar un desafío para la transcripción automática?	Todo el profesorado informa de una combinación de inglés y español ("spanglish técnico"): CRUD, frameworks, queries, syscalls, hashing, overfitting y otros; se informan confusiones específicas de reconocimiento de voz como "MAC" (en referencia a una dirección de red) copiada como "Mac" (en referencia a la marca).	La terminología mixta de cada asignatura es un riesgo real de error de transcripción que debe abordarse de forma determinista, no dejarse a criterio del modelo de lenguaje.
4. ¿Cómo describiría el método analítico-	Los diez docentes vinculan la síntesis con la capacidad de descomponer ("desarmar") un sistema tecnológico	Existe consenso docente en que el método analítico-

Pregunta	Síntesis de los 10 docentes	Interpretación
sintético y su valor pedagógico?	en módulos, y esperan que un resumen conserve esa misma jerarquía lógica al reconstruirlo.	sintético ya es, de hecho, la forma en que enseñan contenido técnico.
5. ¿Qué aspectos deben preservarse en un resumen para mantener su valor educativo?	La calidad no se mide por la fluidez de la redacción sino por la fidelidad terminológica y la preservación de casos de borde (excepciones a una regla general).	Un resumen fluido pero impreciso se considera peor que uno menos elegante pero fiel al contenido técnico exacto.
6. ¿Cuáles serían los principales riesgos de automatizar la generación de resúmenes?	El riesgo más citado es la "pereza cognitiva": que el resumen sustituya el razonamiento del estudiante en lugar de apoyarlo. Un segundo grupo de docentes expresa desconfianza ante posibles alucinaciones de la IA en conceptos normativos (por ejemplo, marcos como ITIL o COBIT).	El sistema debe presentarse como apoyo al repaso, no como sustituto de asistir a clase, y debe minimizar activamente el riesgo de alucinación.
7. ¿Qué criterios utilizaría para evaluar si un resumen automático es de calidad aceptable?	Tolerancia cero a errores en acrónimos técnicos o en una sentencia de código; un solo error de este tipo se considera un fallo crítico, independientemente de la calidad general del resumen.	El estándar de calidad docente es más estricto que el de un resumen general: exige exactitud terminológica, no solo cobertura temática.
8. ¿Cómo imagina que esta herramienta podría integrarse en su enseñanza?	Disposición elevada para integrar el sistema como material de soporte en el aula virtual institucional (Moodle); un profesor propone un esquema de co-	La integración esperada por los docentes es como complemento

Pregunta	Síntesis de los 10 docentes	Interpretación
	evaluación docente-IA usando el resumen propuesto en el sistema, el profesor revisa antes de publicar.	supervisado, no como publicación automática sin revisión.
9. ¿Estaría dispuesto a permitir el uso de grabaciones de sus clases para probar el sistema?	La mayoría de los 10 docentes entrevistados (90%) se muestra favorable a utilizar grabaciones de sus clases para la práctica y el perfeccionamiento del sistema; uno de los docentes (D4) condiciona su colaboración a una revisión manual previa a cualquier publicación.	Hay un amplio grupo de docentes dispuestos a colaborar, pero no unánime ni incondicional.

3.5.7 Presentación y Descripción de los Resultados Obtenidos

La pregunta 3 de la entrevista identifica el desafío técnico más citado por los diez docentes, la mezcla de inglés y español en la terminología especializada de cada asignatura. Este hallazgo se cruza con la pregunta 3 de la encuesta, donde el 55% de los estudiantes reporta dificultad para identificar los conceptos principales de una clase extensa: si el propio vocabulario técnico es difícil de transcribir con precisión, la dificultad de identificación que reportan los estudiantes probablemente se origina, al menos en parte, en errores de transcripción de esos mismos términos y no solo en la extensión del audio.

La pregunta 6 de la entrevista pone de manifiesto que un resumen fluido y mal formulado es para los docentes algo peor que un resumen menos fluido, pero con terminología precisa. Este modo de hacer coincide con la pregunta 6 de la encuesta, donde los estudiantes sitúan la identificación de los conceptos y la precisión como los dos criterios de la calidad que más valoran, habiendo sido elegidos en primer lugar por el 87,6% de la muestra, y muy por encima de la brevedad o de la coherencia general del texto. La coincidencia entre ambos instrumentos, uno cuantitativo y aplicado a los estudiantes, el otro cualitativo y aplicado a los docentes, es la prueba más robusta para priorizar la fidelidad terminológica como criterio de diseño del sistema.

En lo que respecta a la forma de acceder a la información, la pregunta 4 de la encuesta indica que la que se refiere a la imposibilidad de buscar un tema dentro del audio, es la que tiene más dificultad (88,8%) por encima del hecho de la duración del archivo. También la pregunta 2 de la entrevista en la que se recoge la parte que consideran que deben retener de una clase, avala este resultado visto desde la óptica docente; la clase tiene valor en su lógica interna, no en su desarrollo cronológico completo. Ambas conclusiones van en la misma dirección: interesa un informe organizado por conceptos y estructurado por secciones, no una transcripción corrida ni un resumen de una única pieza.

Finalmente, la pregunta número 9 de la entrevista, acerca de la disposición docente por colaborar con el sistema, matiza el entusiasmo general, ya que solo 9 de los 10 docentes acceden sin condiciones, uno de ellos (D4) pone como condición la revisión manual antes de que se publique cualquier resumen. Esta puntual reserva, junto a la posibilidad de que la herramienta produzca "pereza cognitiva", que los docentes indicaron en la pregunta 6 de la entrevista, no aparece en la encuesta a estudiantes, ya que esta les concierne a ellos exclusivamente en su rol de responsables del contenido publicado.

3.5.8 Informe Final del Análisis de los Datos

Considerando los resultados obtenidos en las secciones anteriores, se concluye que la dificultad para aprovechar grabaciones extensas de clase es real dentro de la ULEAM Extensión El Carmen y no un supuesto de partida sin sustento: más de la mitad de los estudiantes encuestados invierte entre 20 y 45 minutos en localizar un solo tema dentro de un audio, y más de la mitad califica como difícil o muy difícil identificar los conceptos principales de una clase extensa.

No se limita a la duración de las grabaciones, sino que es también, y especialmente, la incapacidad de navegar las grabaciones temáticamente que señala el 88,8% de los alumnos, y que se ve incluso acentuada por la terminología técnica mixta que para los diez docentes a los que se entrevistó representa el principal elemento obstaculizador de la transcripción y, en consecuencia, también el del repaso. La solución que describen ambos grupos, de modo independiente, tiene en partes centrales rasgos comunes: un sistema que produzca información estructurada por conceptos educativos, leal a la terminología concreta de cada clase, elaboración que además ofrezca apoyo para el repaso sin sustituir el juicio del estudiante o el control editorial sobre lo que se publica por parte del docente.

Estos hallazgos, con sus cifras exactas, sustentan directamente el diseño de ULEAMTranscribe presentado en el Capítulo IV: la plantilla universal de secciones organizadas por concepto, la capa de verificación anti-alucinación que prioriza la fidelidad sobre la fluidez, la extracción determinista de terminología técnica mixta, y la visibilidad privada por defecto de cada clase hasta que el docente decide publicarla.

CAPÍTULO IV

MARCO PROPOSITIVO

4.1 Introducción

La Universidad Laica Eloy Alfaro de Manabí (ULEAM), concretamente con su Extensión El Carmen, ofrece la carrera de Ingeniería en Tecnologías de la Información con el objetivo de desarrollar las competencias técnicas y científicas para dar respuesta a las demandas de transformación digital de la sociedad ecuatoriana. Así, en esta situación no resulta extraño que la grabación de las clases sea una práctica común y habitual con el objetivo de proporcionar a los estudiantes la oportunidad de revisar los contenidos discutidos en sus clases y contribuir a su formación de una forma autónoma. No obstante, lo largo de nuestras vivencias docentes, hemos sido conscientes de la dificultad que implica revisar los registros de estas grabaciones de forma significativa: comprobar una grabación de 60 minutos significa revisar el mismo tiempo transcurrido de la sesión, y esto supone un obstáculo considerable para la efectividad de estudiar de forma autónoma, sobre todo, en las épocas de preparación para sus evaluaciones.

Frente a esta situación, el presente trabajo de titulación plantea el desarrollo y la implementación de ULEAMTranscribe, una aplicación web con Procesamiento de Lenguaje Natural (PLN) que automatiza, mediante un pipeline de inteligencia artificial de siete pasos, la transcripción de grabaciones de clases universitarias y la producción de informes académicos estructurados. El sistema combina la extracción determinista de datos a partir de Python y la generación de lenguaje natural mediante el modelo Llama 3.1 8B ejecutado localmente con Ollama y mediante la transcripción de voz con faster-whisper con aceleración de GPU. La presente arquitectura híbrida, a la vez determinista y generativa, permite que la solución funcione sin depender de servicios en la nube de pago o de conexiones de red de alta velocidad, y mitiga el riesgo de alucinaciones inherentes a los modelos de lenguaje, ya que fija los datos exactos (cifras, fechas, nombres) en Python en el momento anterior a la participación del LLM en la redacción.

El desarrollo de la propuesta se ha planificado en un período de seis meses, correspondiente al ciclo académico 2025-2026, y se ejecuta sobre los equipos de cómputo disponibles en el entorno del desarrollador: una laptop ASUS TUF Gaming con GPU NVIDIA como equipo principal de desarrollo, y una estación de trabajo con procesador AMD Ryzen de 16 núcleos y GPU NVIDIA RTX 2050 (4 GB VRAM) como entorno de validación. La restricción de 4 GB de VRAM, insuficiente para mantener simultáneamente en memoria el modelo de transcripción

y el modelo de lenguaje, constituye una condición de diseño determinante y obliga a una arquitectura de carga alternada entre ambos modelos, documentada en la sección 4.4.2.

El presente capítulo detalla la propuesta en su totalidad, siguiendo las cinco fases del Modelo en Cascada adoptado como metodología de desarrollo de software: Análisis, Diseño, Implementación, Verificación y Mantenimiento. Dentro de esta metodología de ingeniería de software, que se mantiene sin cambios respecto a la planificación original, la metodología interna de procesamiento de inteligencia artificial del sistema evolucionó durante la fase de implementación: la técnica inicialmente concebida de doble lectura con texto invertido fue ampliada a una estrategia de triple lectura posicional (normal, invertida y de la sección central del texto), combinada con una capa de extracción determinista y de detección de estructura semántica que no dependía únicamente del LLM. Esta evolución, sus motivaciones y su evidencia se documentan en detalle en la sección 4.4.3.3. Para cada fase del Modelo en Cascada se documentan los productos entregables generados, las decisiones técnicas tomadas y los resultados obtenidos, proporcionando una visión completa y reproducible del proceso de construcción del sistema.

4.2 Descripción de la Propuesta

La solución que se plantea para apoyar el proceso de enseñanza-aprendizaje en la ULEAM Extensión El Carmen consiste en diseñar, desarrollar y desplegar ULEAMTranscribe, una aplicación web con inteligencia artificial que opera en dos niveles: como herramienta de productividad académica para los docentes, que pueden grabar o subir sus clases y obtener automáticamente la transcripción y un informe académico estructurado, y como recurso de estudio autónomo para los estudiantes, que acceden a la transcripción completa y al informe estructurado sin necesidad de revisar el audio íntegramente, con herramientas adicionales de búsqueda, favoritos y organización por materia y etiquetas.

La propuesta se apoya en la combinación de tres tecnologías complementarias: Flask como framework web para gestionar usuarios, clases y datos y SQLite como motor de base de datos; faster-whisper para la transcripción automática con reconocimiento de voz apostando por español; y Llama 3.1 8B ejecutado en la máquina local por la plataforma Ollama para la generación del informe académico. De esta manera, toda la información tratada queda almacenada en la máquina local y no es transmitida a servicios externos lo que refuerza la confidencialidad de los contenidos académicos.

Los objetivos específicos que persigue ULEAMTranscribe son los siguientes:

- Registrar cuentas diferenciadas para docentes y estudiantes con correo electrónico institucional y contraseña, validando el formato del correo docente según el patrón nombre.apellido@uleam.edu.ec.
- Permitir a los docentes crear una clase mediante grabación en vivo desde el navegador o mediante la carga de un archivo de audio existente (formatos WAV, MP3, MP4, M4A, OGG y WEBM), asociándola opcionalmente a una materia y a etiquetas (tags) libres definidas por el propio docente.
- Realizar la transcripción automática del contenido verbalizado a partir de la utilización de faster-whisper (modelo medium, cuantización int8) añadiendo un filtrado por detección de actividad de voz (VAD) de la voz para recortar las alucinaciones en los silencios.
- Obtener un informe académico estructurado a partir de una plantilla universal capaz de aplicarse a cualquier asignatura, mediante una propuesta de estrategia híbrida que combina la extracción determinista en Python junto con triple lectura posicional sobre Llama 3.1 8B, basada en el principio revelado por Liu et al. (2023) sobre la pérdida de atención que sufre el modelo en los centros de contextos extensos.
- Detectar de forma determinista el modo retórico dominante de cada clase (narrativo, expositivo, argumentativo, técnico o instruccional) para adaptar el énfasis de la redacción sin alterar la estructura de secciones del informe, generalizando el sistema a asignaturas tan distintas como historia, programación o metodología de la investigación.
- Antes de realizar la escritura final, este sistema deberá verificar automáticamente la existencia de todos los nombres o cifras que estaban anotados de la forma que ha indicado el modelo en la transcription fuente, y por tanto eliminando de forma determinista aquellos datos que no se verifiquen (mitigación de alucinaciones).
- Proveer a los participantes un sistema para marcar como favoritas sus clases, buscar sus clases por su título o concepto, y mirar unas estadísticas básicas de las clases usadas (número de clases vistas, número de clases marcadas como favoritas, docentes que han publicado contenido).

- Llevar la cuenta de las visitas únicas por par estudiante-clase y ofrecer a los docentes una serie de estadísticas con el total de clases, número de clases públicas y horas de contenido registrado.
- Llevar a cabo el control de la visibilidad de cada clase, mediante la definición del estado de publicación (pública o privada) que el docente puede alterar en cualquier momento.
- Exponer la aplicación de forma segura mediante túnel ngrok durante las sesiones de prueba, sin requerir infraestructura de servidor dedicado.

Para el desarrollo de este trabajo de titulación se emplea el Modelo en Cascada como metodología de ingeniería de software. Este modelo lineal y secuencial se caracteriza por dividir el proceso de desarrollo en fases claramente delimitadas, donde cada etapa debe completarse en su totalidad antes de iniciar la siguiente. El Modelo en Cascada es extremadamente adecuado para ULEAMTranscribe por varios motivos: el alcance funcional del sistema queda completamente definido, ya que la investigación se realiza desde el pronóstico realizado en el Capítulo III; la duración del desarrollo sea fija (seis meses académicos); los recursos humanos son muy limitados (un único desarrollador); y el campo tecnológico del PLN y reconocimiento de voz tiene herramientas muy maduras y bien documentadas por lo que no se requiere de un modelo iterativo para la definición del alcance global del sistema. Si bien el ciclo de ingeniería de software sigue el esquema secuencial de Cascada, el componente algorítmico interno del pipeline de IA (sección 4.4.3.3) sí experimentó una iteración de mejora dentro de la fase de Implementación, hallazgo que se documenta como parte de los resultados de dicha fase y no como una desviación del modelo metodológico adoptado.

La propuesta descrita en este capítulo da cumplimiento directo al objetivo general y a los cuatro objetivos específicos planteados en el Capítulo I. La siguiente tabla traza esa correspondencia, indicando en qué fase del Modelo en Cascada se satisface cada objetivo específico:

Tabla 3 Trazabilidad entre los objetivos y las fases del Modelo en Cascada

Objetivo específico (Capítulo I)	**Fase del Modelo en Cascada que lo satisface**
OE1. Analizar los requerimientos académicos y tecnológicos necesarios para la implementación del sistema.	4.4.1 Análisis (información institucional, actores,

	requerimientos funcionales/no funcionales, diagramas UML).
OE2. Diseñar la arquitectura del sistema incorporando módulos de transcripción de audio y procesamiento de lenguaje natural.	4.4.2 Diseño (arquitectura en capas, interfaz, base de datos).
OE3. Implementar un prototipo funcional que procese grabaciones de clases y genere resúmenes comprensibles y coherentes.	4.4.3 Implementación (lenguajes, clases, métodos y funciones del pipeline de IA).
OE4. Evaluar la efectividad del sistema mediante pruebas con grabaciones reales y retroalimentación de los usuarios.	4.4.4 Verificación (pruebas de datos en frío y de datos reales).

El objetivo general (desarrollar un sistema informático que permita generar resúmenes automáticos de clases grabadas mediante Procesamiento de Lenguaje Natural, y mejorar así el acceso, la comprensión y el repaso del contenido académico en la ULEAM Extensión El Carmen) se cumple de forma transversal a lo largo de las cinco fases documentadas, y su cierre queda formalizado en la fase 4.4.5 (Mantenimiento), donde el sistema queda entregado e instalado.

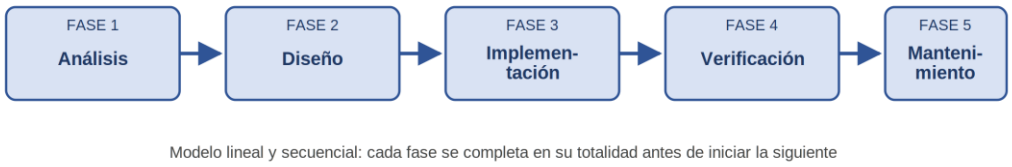


Ilustración 1 Fases del Modelo en Cascada

4.3 Determinación de Recursos

El desarrollo de un proyecto informático requiere un conjunto de recursos que permitan obtener los resultados planteados, entre los cuales se considera importante mencionar los humanos, tecnológicos y económicos.

4.3.1 Humanos

Dentro de los recursos humanos se encuentran los directos e indirectos. Para el presente proyecto y numeral se considerará solo los recursos humanos directos que participan en el proyecto.

Tabla 4 Recursos humanos

Personal	**Función**
Tutor académico	Orientación metodológica y revisión de avances del trabajo de titulación.
Docentes de ITI — ULEAM El Carmen	Para obtener los requerimientos del sistema y validar el informe generado con clases reales.
Estudiantes de ITI — ULEAM El Carmen	Con el fin de validar la usabilidad e idoneidad del informe para el estudio.
Desarrollador (autor)	Creador del sistema informático: análisis, diseño, implementación, verificación y mantenimiento.

4.3.2 Tecnológicos

En el desarrollo de la propuesta es esencial utilizar recursos tecnológicos tanto de hardware como de software. Es necesario que estos equipos cumplan con características específicas que permitan al desarrollador la creación e inferencia local de los modelos de inteligencia artificial.

Tabla 5 Recursos tecnológicos

Hardware	**Características**	**Software**	**Características**
Laptop ASUS TUF Gaming (equipo de desarrollo)	SO Ubuntu 22.04 LTS o superior CPU Intel Core i7 RAM 16 GB DDR4 GPU NVIDIA con CUDA SSD 512 GB	Backend	Python 3, Flask, Flask-Login, Flask-CORS, SQLite (sqlite3)

Estación de trabajo (equipo de validación)	SO Ubuntu 22.04 LTS CPU AMD Ryzen 16 núcleos RAM 32 GB DDR4 GPU NVIDIA RTX 2050 (4 GB VRAM, CUDA) SSD 1 TB	IA — Transcripción	faster-whisper (modelo medium, int8)
Dispositivos de prueba móvil	Android 10 o superior	IA — Redacción	Ollama + Llama 3.1 8B (API REST local)
Router Wi-Fi	Banda ancha para pruebas de carga y exposición ngrok	Editor de código	Visual Studio Code
		Control de versiones	Git / GitHub
		Exposición pública	ngrok (túnel HTTPS)

La elección de cada componente respondió a criterios verificables durante la implementación: faster-whisper se prefirió sobre la implementación de referencia de OpenAI por su menor huella de memoria en cuantización int8, condición necesaria dado el límite de 4 GB de VRAM del equipo de validación; y Llama 3.1 8B se prefirió sobre variantes de menor tamaño (como Llama 3.2 3B) porque estas últimas mostraron, durante pruebas exploratorias, mayor tendencia a alucinar contenido no presente en la transcripción fuente.

4.3.3 Económicos (presupuesto)

En los recursos económicos se toman en cuenta los elementos necesarios para el desarrollo, alojamiento y pruebas de la solución informática. Es decir, no se incluyen los equipos institucionales que docentes y estudiantes ya poseen para utilizar el sistema en producción.

Tabla 6 Presupuesto estimado del proyecto

Cantidad	Elemento	Detalle	C/U	Subtotal
1	Laptop ASUS TUF Gaming	Equipo de desarrollo (ya disponible, sin costo adicional)	$0,00*	$0,00
1	Estación AMD/RTX 2050	Equipo de validación (ya disponible)	$0,00*	$0,00
1	Stack Python/Flask/SQLite/Ollama	Software de código abierto y distribución gratuita	$0,00	$0,00
300 h	Horas de programación	Diseño, implementación y pruebas	$4,00	$1.200,00
6 meses	Internet	Sincronización con GitHub, descarga de modelos, pruebas ngrok	$25,00	$150,00
6 meses	Energía eléctrica	Consumo adicional por inferencia prolongada en GPU	$8,00	$48,00
—	Imprevistos (10%)	Reserva para gastos no planificados	—	$142,80

El uso exclusivo de software de código abierto reduce significativamente el costo total del proyecto, estimado en $1.540,80. El ítem de mayor peso presupuestal corresponde a las horas de programación, estimadas en 300 horas distribuidas a lo largo de los seis meses de desarrollo.

4.4 Desarrollo

A continuación se documenta en detalle cada una de las cinco fases del Modelo en Cascada aplicadas al desarrollo de ULEAMTranscribe.

4.4.1 Análisis

En la etapa del análisis se realiza un estudio detallado de los requisitos para el desarrollo del sistema, se obtiene la información general del contexto académico en el que opera, se especifican los actores y sus funciones, se definen los requerimientos funcionales y no funcionales, y finalmente se presentan los diagramas UML que permiten observar el comportamiento y estructura de los procesos del sistema.

El diagnóstico aplicado en el Capítulo III (encuesta a 80 estudiantes, 87% de la muestra calculada, y entrevistas a 10 docentes) confirmó empíricamente esta necesidad: el 25% de los estudiantes invierte más de 30 minutos en localizar un solo tema dentro de una grabación (sección 3.5.6.1), el 55% califica como difícil o muy difícil identificar los conceptos principales de una clase extensa (sección 3.5.6.1), y el 73,8% consideró que una herramienta de resumen automático sería muy o extremadamente útil para su estudio (sección 3.5.6.1). Los diez docentes entrevistados, por su parte, coincidieron en que la terminología técnica mixta de cada asignatura es el principal riesgo de fidelidad para un sistema de este tipo (sección 3.5.6.2), hallazgo que se traduce directamente en el diseño de la extracción determinista de términos técnicos descrita en la sección 4.4.3.3.

Tres hallazgos adicionales de la sección 3.5.7 del Capítulo III se tradujeron en requisitos concretos de este capítulo: la dificultad de búsqueda dentro del audio, señalada por el 88,8% de los estudiantes como su principal obstáculo, motivó que el informe generado se organice en secciones navegables por concepto en lugar de una transcripción corrida (requisito de generación de un informe estructurado, sección 4.4.1.4); la reserva expresada por el docente D4, que condicionó su colaboración a mantener revisión manual antes de publicar cualquier resumen, motivó que toda clase se cree en estado privado por defecto (requisito de visibilidad pública/privada); y la preocupación docente por la "pereza cognitiva" motivó que el sistema mantenga siempre disponible la transcripción completa junto al informe, en lugar de sustituir el material original.

4.4.1.1 Información de la Institución

La Universidad Laica Eloy Alfaro de Manabí (ULEAM), Extensión El Carmen, fue creada mediante Ley No. 10, inscrita en el Registro Oficial 313 del 13 de noviembre de 1985, y está ubicada en el cantón El Carmen, provincia de Manabí, en la Avenida 3 de Julio. Desde entonces oferta, entre otras carreras, la de Ingeniería en Tecnologías de la Información.

La carrera atiende a una población de 120 estudiantes activos matriculados en asignaturas técnicas de nivel intermedio y avanzado (desarrollo de software, bases de datos, redes, auditoría de TI, inteligencia artificial, entre otras), impartidas por un cuerpo de 12 docentes. El diagnóstico presentado en el Capítulo III identificó que estas asignaturas registran sus sesiones en audio con regularidad, pero la institución carece de instrumentos digitales que permitan aprovechar ese material más allá de la reproducción íntegra del archivo.

El presente trabajo de titulación se enmarca en esa extensión: el sistema ULEAMTranscribe se concibe como una herramienta de apoyo académico de uso interno, dirigida a los docentes y estudiantes de la carrera, sin fines comerciales, y su alcance de despliegue inicial se limita a las asignaturas técnicas de la ULEAM Extensión El Carmen.

4.4.1.2 Actores del Sistema

A diferencia de un sistema de gestión empresarial, organizado en torno a un orgánico funcional con cargos y jerarquías, ULEAMTranscribe se organiza en torno a tres actores con roles y permisos diferenciados, cuya interacción se describe a continuación:

- Docente: actor humano responsable de crear clases (por grabación en vivo o carga de archivo), asociarlas a una materia y a etiquetas, y controlar su visibilidad (pública o privada). Accede a un panel con estadísticas de su actividad.
- Estudiante: actor humano que se conecta a las clases públicas, ve la transcripción completa y el informe académico generado, puede buscar clases, aplazar como favoritas y consultar sus propias estadísticas.
- Sistema (pipeline de IA): actor interno que orquesta el procesamiento de cada clase (gestión de memoria GPU, transcripción, extracción determinista, triple lectura sobre Llama 3.1 8B, verificación anti-alucinación y redacción del informe) sin intervención manual del docente ni del estudiante durante su ejecución.

4.4.1.3 Funcional Procedimental

De forma análoga a una tabla de funciones y procedimientos organizacional, la siguiente tabla detalla, para cada actor humano, las funciones que puede ejecutar en el sistema y el procedimiento paso a paso que sigue cada una:

Tabla 7 Funciones y procedimientos por actor

Actor	Función	Procedimiento
Docente	Crear clase	• Seleccionar grabar en vivo o cargar un archivo de audio • Ingresar título, materia y etiquetas • Confirmar y esperar el procesamiento del pipeline de IA
Docente	Gestionar clase	• Consultar la transcripción y el informe generado • Alternar la visibilidad entre pública y privada • Eliminar la clase (y su archivo de audio) si corresponde
Docente	Consultar estadísticas	• Acceder al panel de docente • Revisar el total de clases, clases públicas y horas de contenido registradas, agrupadas por materia
Estudiante	Buscar clases	• Ingresar un término de búsqueda (título o materia) • Revisar los resultados filtrados entre las clases públicas
Estudiante	Consultar clase	• Seleccionar una clase de la lista • Alternar entre la transcripción completa y el informe académico
Estudiante	Marcar favorito	• Presionar el ícono de favorito en la clase deseada • Consultar la clase posteriormente desde la sección de favoritos

4.4.1.4 Requerimientos Funcionales

- Registrar cuentas diferenciadas para docentes y estudiantes con correo institucional y contraseña.
- Validar el formato del correo institucional del docente (patrón nombre.apellido@uleam.edu.ec) y del estudiante (patrón e + número de cédula de 10 dígitos + @live.uleam.edu.ec), solicitando el número de cédula únicamente cuando el rol seleccionado es estudiante.

- Permitir al docente crear una clase mediante grabación de audio en vivo desde el navegador.
- Posibilitar al profesor crear una clase subiendo un archivo de audio ya existente (WAV, MP3, MP4, M4A, OGG, WEBM).
- Transcribir automáticamente el audio de la clase por medio de faster-whisper.
- Generar automáticamente un informe académico estructurado a partir de la transcripción.
- Posibilitar al profesor asociar una asignatura y etiquetas libres a cada clase.
- Facilitar al profesor alternar la visibilidad de una clase entre pública o privada
- Permitir al estudiante buscar clases públicas por título o materia.
- Permitir al estudiante marcar y desmarcar clases como favoritas.
- Registrar un máximo de una visita por par estudiante-clase.
- Mostrar paneles de estadísticas diferenciados para docente y estudiante.
- Permitir eliminar una clase junto con su archivo de audio y etiquetas asociadas.

Tabla 8 Requerimientos funcionales priorizados

Código	Requisito	Prioridad	Dificultad
RF-01	Registrar cuentas diferenciadas para docentes y estudiantes con correo institucional y contraseña.	Alta	Media
RF-02	Validar el formato del correo institucional del docente y del estudiante según su patrón correspondiente.	Alta	Baja
RF-03	Permitir al docente crear una clase mediante grabación de audio en vivo desde el navegador.	Alta	Alta
RF-04	Posibilitar al profesor crear una clase subiendo un archivo de audio ya existente.	Alta	Media
RF-05	Transcribir automáticamente el audio de la clase por medio de faster-whisper.	Alta	Alta

RF-06	Generar automáticamente un informe académico estructurado a partir de la transcripción.	Alta	Alta
RF-07	Posibilitar al profesor asociar una asignatura y etiquetas libres a cada clase.	Media	Baja
RF-08	Facilitar al profesor alternar la visibilidad de una clase entre pública o privada.	Media	Baja
RF-09	Permitir al estudiante buscar clases públicas por título o materia.	Media	Media
RF-10	Permitir al estudiante marcar y desmarcar clases como favoritas.	Baja	Baja
RF-11	Registrar un máximo de una visita por par estudiante-clase.	Media	Media
RF-12	Mostrar paneles de estadísticas diferenciados para docente y estudiante.	Media	Media
RF-13	Permitir eliminar una clase junto con su archivo de audio y etiquetas asociadas.	Media	Baja

Fuente: Elaboración propia

4.4.1.5 Requerimientos No Funcionales

- El pipeline completo (transcripción + informe) debe operar dentro de los límites de VRAM del equipo de validación (4 GB), sin errores de memoria insuficiente.
- El sistema tiene que evitar al máximo la producción de contenido no verificable (alucinaciones) por medio del anclaje determinista y la verificación de la generación.
- El sistema tiene que operar sin dependencia de los servicios de IA en la nube de pago.
- La estructura del informe (Tema, Conceptos, Desarrollo, Evidencias, Conclusiones) no ha de variar independientemente de la asignatura.
- La interfaz web ha de ser responsiva y funcional en navegador de escritorio y en dispositivos móviles.
- El código fuente debe versionarse en Git con commits atómicos y descriptivos.

- El sistema debe presentar un aviso de consentimiento informado antes de iniciar cualquier grabación en vivo, notificando a los presentes que la sesión será transcrita y almacenada, en línea con el protocolo ético ya aplicado durante la recolección de datos del Capítulo III (donde las grabaciones de clase utilizadas como material de prueba se obtuvieron con consentimiento informado de docentes y estudiantes).
- Las consultas de búsqueda de clases se generarán en un intervalo razonable, sin bloquear la interfaz.

Tabla 9 Requerimientos no funcionales priorizados

Código	Requisito	Prioridad	Dificultad
RNF-01	El pipeline completo (transcripción + informe) debe operar dentro de los límites de VRAM del equipo de validación (4 GB), sin errores de memoria insuficiente.	Alta	Alta
RNF-02	El sistema debe evitar al máximo la producción de contenido no verificable (alucinaciones) mediante anclaje determinista y verificación de la generación.	Alta	Alta
RNF-03	El sistema debe operar sin dependencia de servicios de IA en la nube de pago.	Alta	Media
RNF-04	La estructura del informe no debe variar independientemente de la asignatura.	Media	Baja
RNF-05	La interfaz web debe ser responsiva y funcional en navegador de escritorio y en dispositivos móviles.	Media	Media
RNF-06	El código fuente debe versionarse en Git con commits atómicos y descriptivos.	Baja	Baja
RNF-07	El sistema debe presentar un aviso de consentimiento informado antes de iniciar cualquier grabación en vivo.	Alta	Baja

RNF-08	Las consultas de búsqueda de clases deben resolverse en un intervalo razonable, sin bloquear la interfaz.	Media	Baja

Fuente: Elaboración propia

4.4.1.6 Tratamiento de Datos Personales

Dado que ULEAMTranscribe procesa y almacena la voz de los participantes de una clase, un dato de naturaleza sensible bajo la Ley Orgánica de Protección de Datos Personales (LOPDP) del Ecuador, el diseño del sistema incorpora los siguientes lineamientos de protección de datos, coherentes con el protocolo ético ya seguido durante la recolección de datos primarios del Capítulo III (consentimiento informado, confidencialidad y uso exclusivo con fines académicos):

- Consentimiento informado en el punto de grabación: la vista de creación de clase debe informar, antes de grabar en vivo, que el audio va a ser transcrito y guardado y para que propósito.
- Minimización del alcance de acceso: una clase solo es visible para estudiantes si la docente la marca explícitamente como pública (es_publíca); por defecto, al crearse una nueva clase, esta queda en estado privado.
- Responsable del tratamiento: el docente que registra la clase es su propietario (docente_id), quien decide su publicación, edición o eliminación; el estudiante no puede acceder a una clase privada por ninguna ruta del sistema.
- Derecho de eliminación: el docente puede eliminar en cualquier momento una clase junto con su archivo de audio, sin dejar copias residuales en el servidor.
- Alcance institucional cerrado: el sistema no comparte datos con servicios de IA en la nube (todo el procesamiento es local), de modo que la voz y el contenido de las clases no salen del entorno controlado por el desarrollador y la institución.

4.4.1.7 Diagramas UML

- Casos de Uso

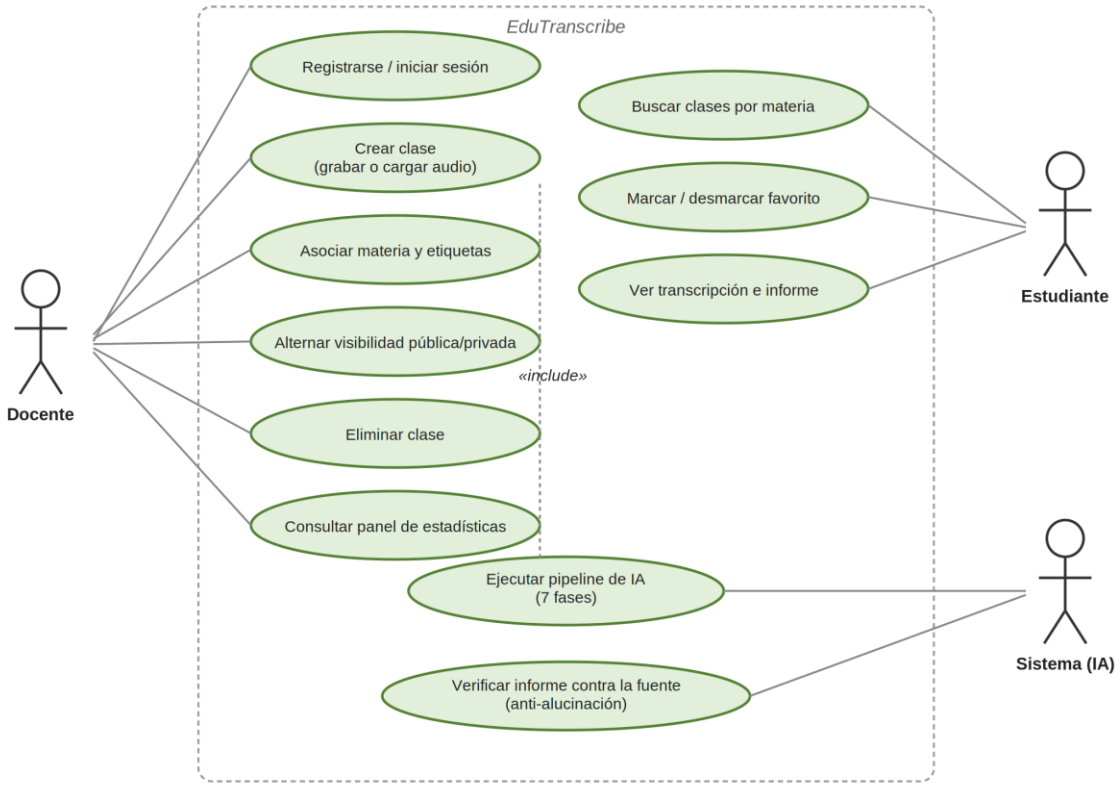


Ilustración 2 Diagrama de casos de uso de ULEAMTranscribe

Los dos casos de uso de mayor complejidad para el aporte metodológico del proyecto, Crear clase y Verificar informe contra la fuente, se detallan a continuación:

Tabla 10 Especificación del caso de uso "Crear clase"

Campo	**Descripción**
Actor principal	Docente.
Actor secundario	Sistema (pipeline de IA).
Precondición	El docente ha iniciado sesión con una cuenta de rol 'docente'.
Flujo principal	1. El docente selecciona grabar en vivo o cargar un archivo. 2. El sistema valida el formato. 3. El sistema ejecuta el pipeline de siete fases. 4. El sistema almacena la transcripción y el informe. 5. El docente visualiza el resultado.
Flujo alterno	3a. Si el archivo no está en un formato admitido, el sistema rechaza la carga con un mensaje de error.

Postcondición	La clase queda registrada con su transcripción e informe, en estado privado por defecto.

Tabla 11 Especificación del caso de uso "Verificar informe contra la fuente"

Campo	**Descripción**
Actor principal	Sistema (pipeline de IA).
Precondición	Las tres lecturas posicionales sobre Llama 3.1 8B han producido una extracción de nombres propios y cifras.
Flujo principal	1. El sistema normaliza el texto fuente y cada línea de la extracción. 2. Extrae los nombres propios y cifras de cada línea. 3. Verifica si aparecen literalmente en la fuente. 4. Conserva la línea si todos los datos están presentes. 5. Registra en el log un resumen de líneas conservadas y descartadas.
Flujo alterno	3a. Si un nombre o cifra no aparece en la fuente, la línea completa se descarta.
Postcondición	El conjunto de datos que llega a la redacción final contiene únicamente información verificable.

- Diagrama de Secuencia

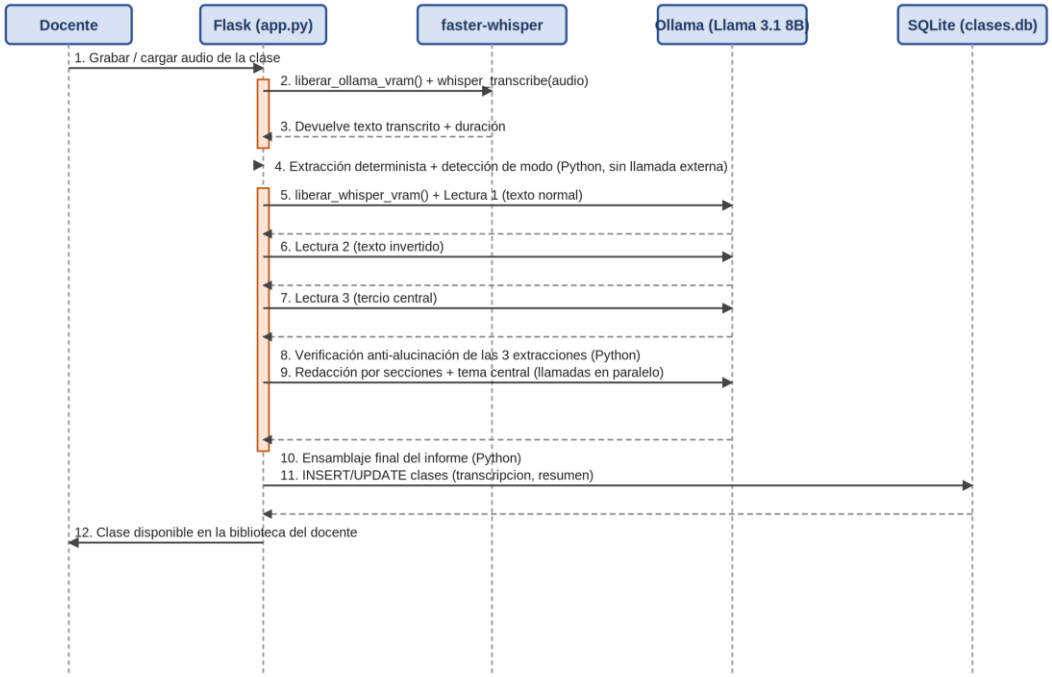


Ilustración 3 Diagrama de secuencia del pipeline de procesamiento de una clase

- Diagrama de Clases

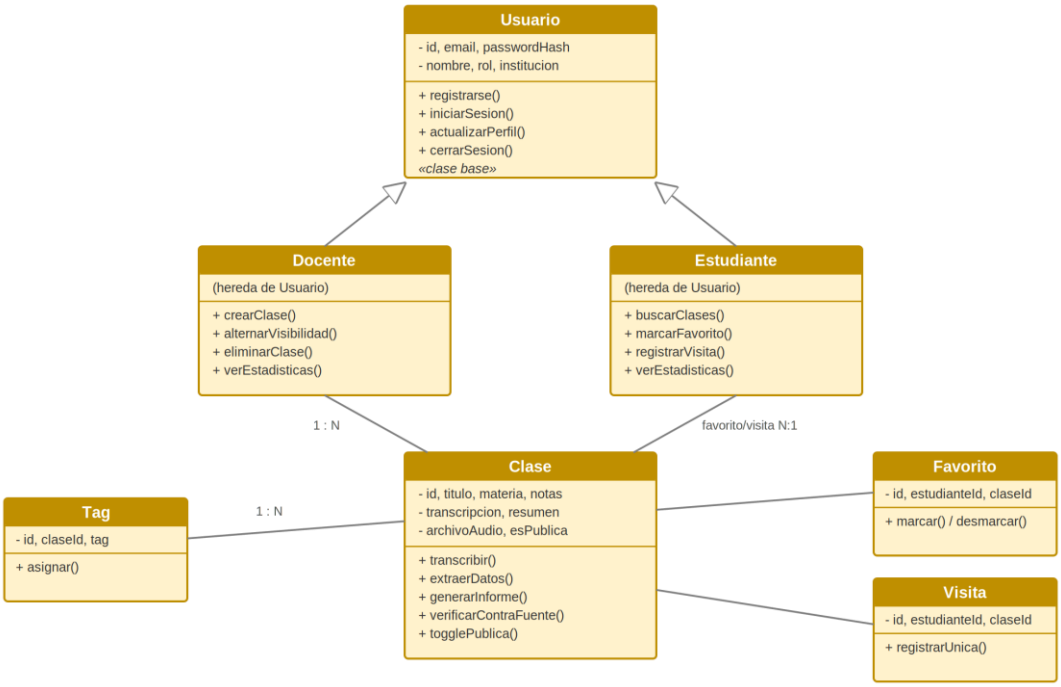


Ilustración 4 Diagrama de clases de ULEAMTranscribe

El diagrama presenta la especialización de la clase Usuario en Docente y Estudiante, y su relación con las clases Clase, Tag, Favorito y Visita. La clase Clase concentra los métodos del

pipeline de IA (transcribir, extraerDatos, generarInforme, verificarContraFuente), lo que refleja que, en la implementación actual (sección 4.4.3.2), esta lógica reside en el mismo archivo que gestiona la entidad, en lugar de estar distribuida en módulos independientes.

- Diagrama de Estados

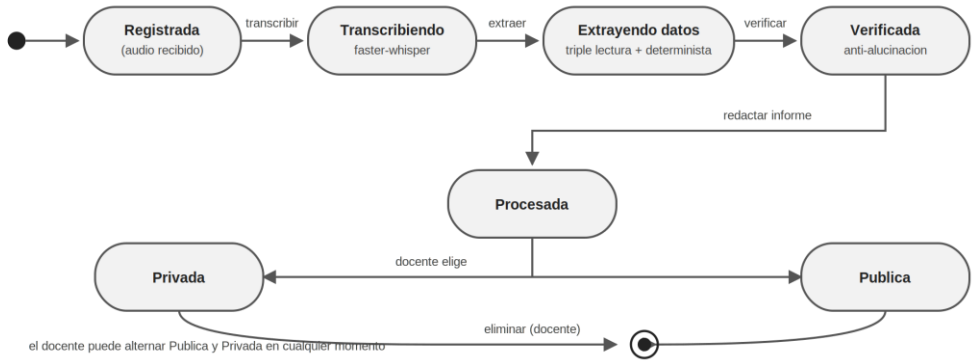


Ilustración 5 Diagrama de estados de la entidad Clase

El diagrama describe el ciclo de vida de una clase desde su registro hasta su eventual eliminación. Una vez que la clase alcanza el estado Procesada, el docente decide su visibilidad (Pública o Privada) y puede alternarla en cualquier momento posterior sin reiniciar el pipeline; la transición a Eliminada es unidireccional y remueve tanto el registro como el archivo de audio asociado.

4.4.2 Diseño

La fase del diseño se basa en la arquitectura del software, es decir, en qué conexión se va a emplear; además se realiza una planificación de los elementos a utilizarse en la interfaz (colores, iconografía, fuente, entrada, procesamiento y salida de datos), y se elabora la base de datos y su estructura, según el análisis de requerimientos.

4.4.2.1 Conexión

La conexión se efectúa mediante una arquitectura en capas, pues esta permite la identificación rápida del origen de los errores. Se trabaja en 3 capas: primero la capa de presentación, donde se muestran las plantillas HTML que ve el usuario; esta se conecta a la capa de negocio, donde se establecen las reglas del negocio y el pipeline de inteligencia artificial de siete fases (transcripción, extracción, triple lectura, verificación y redacción); y a su vez esta capa se enlaza con la capa de datos, que alberga y gestiona la información utilizada en el sistema mediante SQLite.

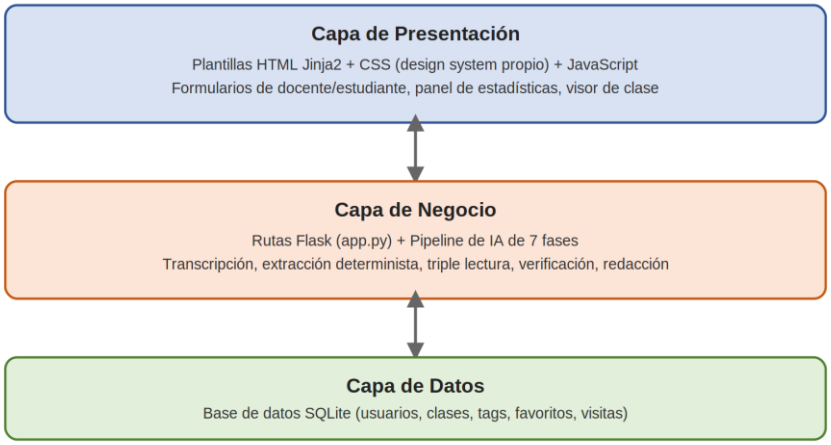


Ilustración 6 Arquitectura en 3 capas de ULEAMTranscribe

4.4.2.2 Interfaz

En la interfaz del sistema, que abarca los colores, iconografía, fuente, entrada, procesamiento y salida de datos, se persigue que el usuario interactúe con el programa de una manera intuitiva, encontrando la información necesaria sin fricción visual.

- Colores

La paleta de ULEAMTranscribe se define mediante variables CSS reutilizables en toda la interfaz (design tokens), con el propósito de mantener consistencia visual entre las vistas de docente y estudiante:

Tabla 12 Paleta de colores de ULEAMTranscribe

Color	**Valor (hex)**	**Uso**
Verde bosque (principal)	#0D3B2E / #145740 / #1C6E52	Color de marca; paneles laterales, botones primarios, encabezados.
Dorado (acento)	#C8973A / #E8C07A	Acciones destacadas y elementos de énfasis.
Crema (fondo)	#F6F1E7 / #EDE6D6 / #E0D8C5	Fondo general de la interfaz; transmite calidez y reduce la fatiga visual.
Tinta (texto)	#1A1A18 / #3A3A36 / #6B6B62	Jerarquía tipográfica: texto principal, secundario y terciario.

Azul marino (vista estudiante)	#1A1A2E / #2D2D4A / #3B3B6E	Diferenciación visual del panel de estudiante frente al de docente (verde).
Rojo / verde de estado	#C0392B / #1C6E52	Mensajes de error y de éxito (toasts, validaciones).

El verde bosque está psicológicamente conectado al crecimiento y la capacidad de concentrarse, siendo idóneo para una situación de estudio; el dorado se reserva para llamadas a la acción, evitando saturar la interfaz de color; y el fondo crema, en lugar del blanco puro, disminuye el contraste extremo y la fatiga visual en extensas sesiones de lectura de transcripciones.

- Iconografía

A diferencia de una fuente de íconos externa, ULEAMTranscribe utiliza iconografía SVG en línea (inline SVG) definida directamente en cada plantilla HTML, con trazo uniforme (stroke-width 2) y sin relleno, lo que permite que los íconos hereden el color del texto circundante mediante currentColor y se adapten automáticamente al tema de cada vista (verde para docente, azul marino para estudiante) sin depender de una librería externa.

- Fuente

El sistema emplea dos familias tipográficas complementarias: Cormorant Garamond, una fuente serif utilizada en títulos y encabezados para transmitir un carácter académico y editorial; y Nunito Sans, una fuente sans-serif de alta legibilidad utilizada en el cuerpo de texto, formularios y elementos de interfaz, donde la claridad prima sobre el carácter estético.

- Entrada de Datos

En la pantalla de entrada de datos para el registro de una nueva clase, se diseñó la distribución de la siguiente forma: el logotipo y el menú se encuentran en el lado izquierdo, el usuario autenticado en la esquina superior derecha, y el formulario de ingreso (título, materia, etiquetas, fuente de audio y visibilidad) ocupa el área central, como se observa en la siguiente imagen:

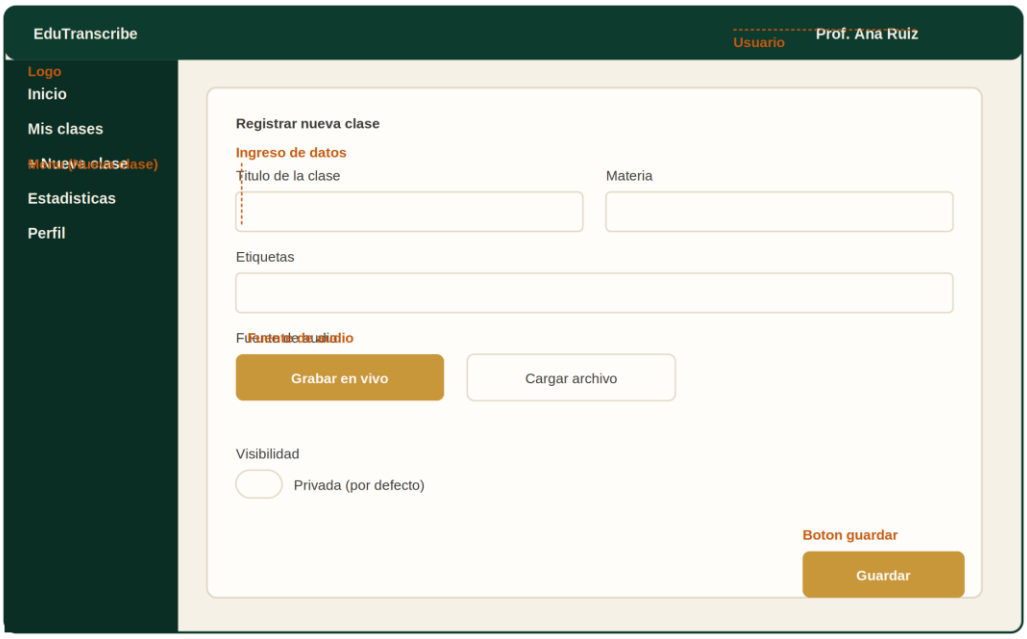


Ilustración 7 Interfaz de entrada de datos: registro de nueva clase

- Procesamiento de Datos

El procesamiento va ligado a la entrada de datos, ya que la información ingresada se transforma en un pipeline de siete fases que se ejecuta en segundo plano. En la interfaz visual, se mantiene la misma distribución de logotipo y menú, pero se añade un indicador de fase actual y una barra de progreso que permiten al docente conocer en qué etapa del pipeline se encuentra su clase sin bloquear la navegación del resto de la aplicación:

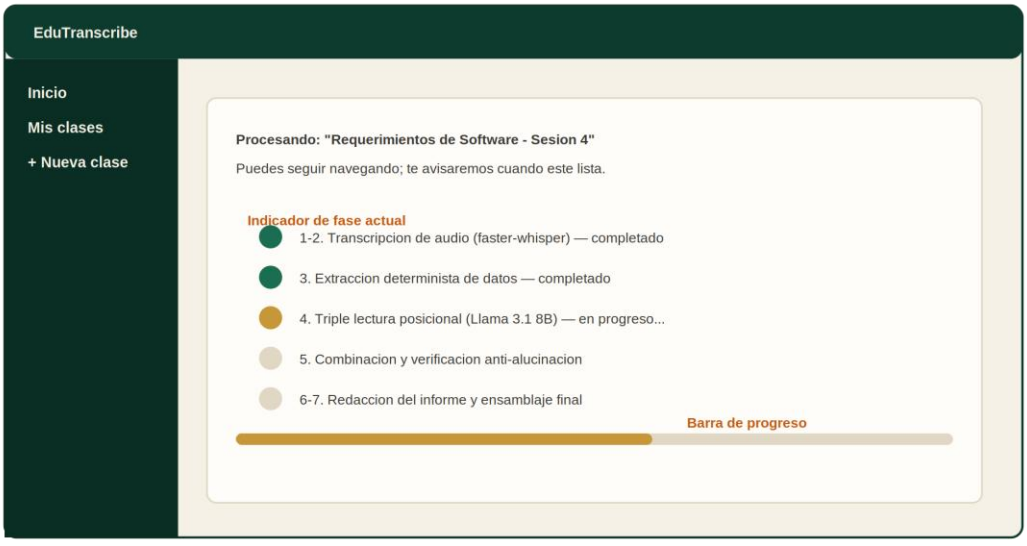


Ilustración 8 Interfaz de procesamiento de datos: pipeline de IA en progreso

- Salida de Datos

La salida de los datos se presenta una vez que el pipeline ha finalizado la interpretación de la información. La vista de detalle de clase muestra dos pestañas (informe académico y transcripción completa) y permite al estudiante marcar la clase como favorita, manteniendo la misma distribución de encabezado que las interfaces de entrada y procesamiento para un manejo consistente del sistema:

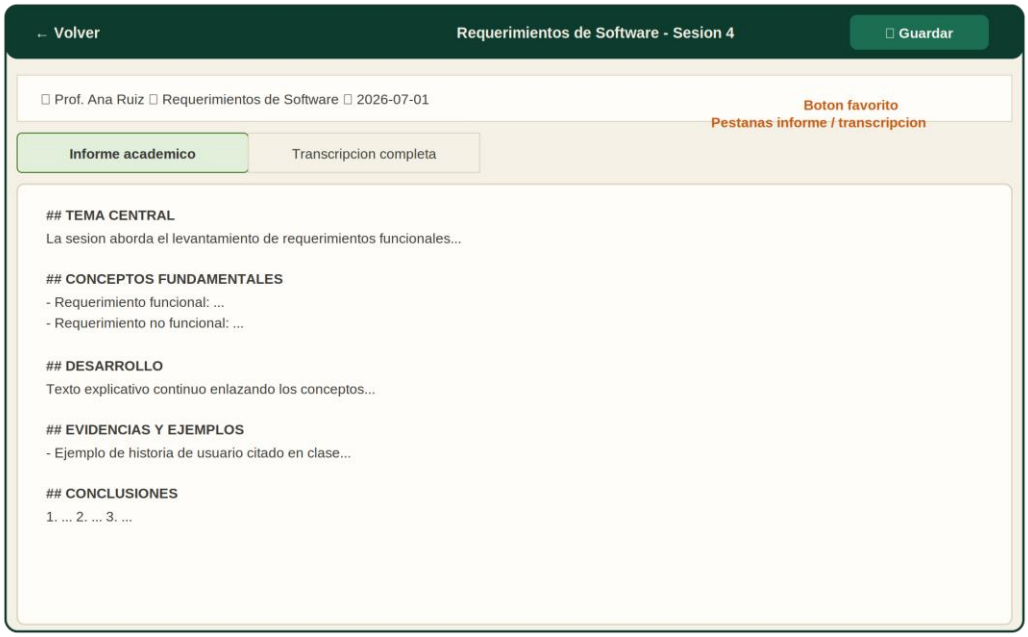


Ilustración 9 Interfaz de salida de datos: informe académico generado

4.4.2.3 Base de Datos

Una base de datos es un instrumento para recopilar y organizar información sobre usuarios, clases y su actividad. Para el presente proyecto se diseñó una base de datos a medida que permite satisfacer las necesidades específicas de ULEAMTranscribe, como es el almacenamiento de la transcripción y el informe generado junto al registro de cada clase. La base de datos se compone de 5 tablas que facilitan almacenar los datos necesarios para el funcionamiento del sistema, manteniendo relaciones entre sí para permitir la ejecución de consultas complejas (por ejemplo, el panel de estadísticas del docente).

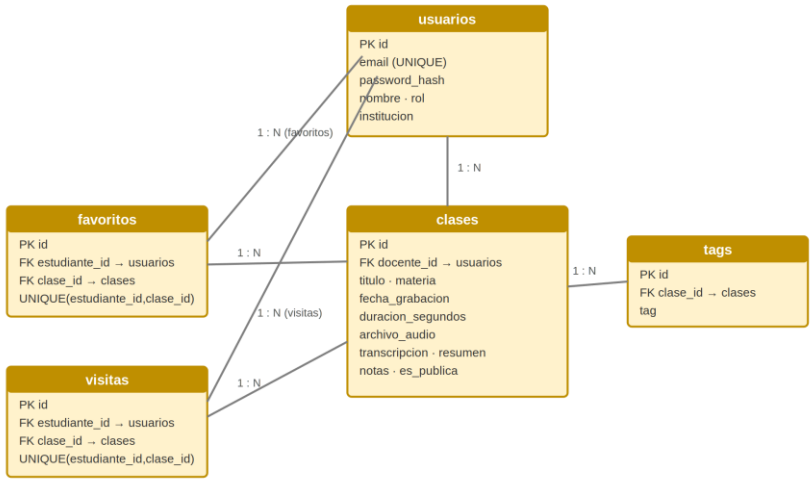


Ilustración 10 Diseño de la base de datos de ULEAMTranscribe

Tabla 13 Estructura del modelo de base de datos de ULEAMTranscribe

Entidad	**Campo**	**Tipo**	**PK/FK**	**Descripción**
usuarios	id	INTEGER	PK	Identificador autoincremental.
	email	TEXT	UNIQUE	Correo electrónico, identificador de sesión.
	password_hash	TEXT	—	Hash de la contraseña (Werkzeug).
	nombre, rol, institucion	TEXT	—	Datos del perfil del usuario.
clases	id	INTEGER	PK	Identificador de la clase.
	docente_id	INTEGER	FK → usuarios	Docente propietario.
	titulo, materia, notas	TEXT	—	Metadatos de la clase.

	transcripcion, resumen	TEXT	—	Transcripción e informe generados por el pipeline.
	archivo_audio	TEXT	—	Ruta del audio en uploads/.
	es_publica	BOOLEAN	—	Visibilidad de la clase.
tags	clase_id, tag	INTEGER/TEXT	FK → clases	Etiquetas libres del docente.
favoritos	estudiante_id, clase_id	INTEGER	FK → usuarios/clases	UNIQUE por par; evita duplicados.
visitas	estudiante_id, clase_id	INTEGER	FK → usuarios/clases	UNIQUE por par; garantiza visita única.

4.4.3 Implementación

Para el desarrollo del presente proyecto se requiere la etapa de implementación, contemplada en el Modelo en Cascada, donde se codifican los lenguajes de programación, clases y métodos definidos en el diseño para la elaboración del sistema web de transcripción y generación de informes académicos.

4.4.3.1 Lenguajes

En el sistema web se utilizó el lenguaje de programación Python 3 junto al framework Flask, el lenguaje de marcado HTML5 y el lenguaje de diseño CSS3, mediante el entorno de desarrollo integrado Visual Studio Code. Para la capa de datos se empleó el lenguaje de consulta estructurado SQL a través de SQLite (sqlite3), sin necesidad de un servidor de base de datos independiente. Los modelos de inteligencia artificial, faster-whisper y Llama 3.1 8B mediante Ollama, se invocan desde Python: el primero como librería nativa, y el segundo a través de peticiones a la API REST local de Ollama (http://localhost:11434). Esto evita invocarlo por línea de comandos y elimina la sobrecarga de varios segundos por llamada que introduce ese método.

4.4.3.2 Clases

Las clases y módulos que componen ULEAMTranscribe se implementaron dentro del archivo principal app.py (1.923 líneas en su versión vigente, rama plantilla-universal-generalidad del repositorio), que concentra las rutas Flask, la inicialización de la base de datos y el pipeline completo de inteligencia artificial. Esta decisión, consolidar en un único archivo en lugar de distribuir en módulos independientes, se sustenta en la escala del proyecto (un único desarrollador, un ciclo de seis meses) y facilitó la iteración rápida sobre el pipeline de IA.

- Conexión. – Se encarga de inicializar y gestionar la conexión con la base de datos SQLite.
- Login. – Administra el acceso de un usuario al sistema mediante Flask-Login.
- Usuario. – Gestiona el registro y perfil de docentes y estudiantes.
- Clase. – Administra el registro de una clase y orquesta las siete fases del pipeline de IA (transcripción, extracción, triple lectura, verificación, redacción).
- Tag. – Gestiona las etiquetas libres asociadas a una clase.
- Favorito. – Administra el marcado y desmarcado de clases favoritas por parte del estudiante.
- Visita. – Registra el acceso único de un estudiante a una clase.

4.4.3.3 Métodos y Funciones

El desarrollo de ULEAMTranscribe contempla un conjunto amplio de métodos y funciones; por esta razón, en el presente documento se consideró aludir a los más relevantes para el aporte metodológico del proyecto: la extracción determinista de datos, la estrategia de triple lectura posicional y la verificación anti-alucinación.

a) Extracción determinista de ejemplos técnicos
De forma paralela e independiente al LLM, esta función captura los ejemplos que el docente introduce en su discurso (marcados por disparadores como 'por ejemplo' o 'supongamos') junto con los términos técnicos literales que contienen, sin recurrir a ninguna llamada al modelo de lenguaje:

```
def capturar_ejemplos_tecnicos(text):
    """
```

```
    Capturador de DOMINIO (TI). Detecta los EJEMPLOS que el profesor introduce
    y captura la oracion completa mas los terminos tecnicos literales.
    Si no hay disparadores de ejemplo, devuelve [].
    """
    pat_disparador = re.compile(
        r'\b(?:por ejemplo|por ejemplos|como por ejemplo|un ejemplo|'
        r'supongamos|imaginen|imaginemos|miren este caso|veamos el caso|'
        r'como cuando|tomemos|pongamos que|por caso)\b', re.IGNORECASE)

    pat_tecnico = re.compile(
        r'`[^`]+`'
        r'|\b[a-z_]\w*\.[a-z_]\w*\([^)]*\)'
        r'|\b[a-z_]\w*\([^)]*\)'
        r'|\b(?:git|npm|pip|sudo|docker|kubectl)\s+[a-z][\w-]*'
        r'|\b(?:SELECT|INSERT|UPDATE|DELETE)\s+[\w*]+'
        r'|\b[A-Z]{2,}(?:[/-][A-Z]{2,})*\b'
        r'|\b(?:Python|Node|Java|React|Docker|PHP|MySQL)\s+\d+(?:\.\d+)*\b'
        r'|(?<!\w)/[\w/.\-]{3,}'
    )
    ejemplos = []
    for o in oraciones:
        if len(o) < 15 or not pat_disparador.search(o):
            continue
        tecnicos = list(dict.fromkeys(t.strip() for t in pat_tecnico.findall(o)))
        entrada = f"- {o[:220]}"
        if tecnicos:
            entrada += f"  [terminos: {', '.join(tecnicos[:5])}]"
        ejemplos.append(entrada)
    return ejemplos
```

La condición de longitud mínima (15 caracteres) y de presencia de un disparador evita capturar oraciones triviales; el patrón técnico reconoce, entre otros, comandos de terminal, sentencias SQL y llamadas a función con notación de punto, lo que permite que esta función opere igual

de bien en una clase de programación que en una de bases de datos, sin necesidad de una lista de palabras específica de una asignatura.

b) Detección de hablantes para atribución de datos
En clases con más de un interlocutor, esta función localiza las presentaciones explícitas ("me llamo...", "soy...") para delimitar los segmentos de texto correspondientes a cada hablante, evitando que el pipeline atribuya la afirmación de una persona a otra:

```
def atribuir_dato(fragmento, plano, segmentos, max_dist=2500):
    """Localiza un fragmento-dato y devuelve el hablante de su bloque,
    o None. Conservador a proposito: prefiere "sin dueno"
    sobre "dueno falso"."""
    if not segmentos:
        return None
    nucleo = re.sub(r'^[\-\*\u2022\s]+', '', fragmento).strip()[:50]
    if len(nucleo) < 10:
        return None
    pos = plano.find(nucleo)
    if pos < 0:
        return None
    for ini, fin, nombre in segmentos:
        if ini <= pos < fin:
            return nombre if (pos - ini) <= max_dist else None
    return None
```

El diseño es conservador de forma consciente. Cuando hay dudas acerca de la pertenencia de un dato (por ejemplo, si el fragmento permanece a una distancia mayor a max_dist del inicio del segmento del hablante), la función devuelve None en vez de arriesgarse a hacer una atribución errónea, perjudicando a la hora de devolver el dato de autoría en favor de sustituirlo por la atribución a la persona equivocada.

c) Verificación anti-alucinación contra la fuente
Este es el método central del aporte metodológico del proyecto: antes de la redacción final, compara cada nombre propio y cifra anotados por el LLM en las tres lecturas contra el texto original, y descarta de forma determinista cualquier línea cuyo dato no sea verificable:

```
def verificar_notas_contra_fuente(notas, texto_fuente):
    """
    Compara NOMBRES PROPIOS y CIFRAS que el LLM anoto contra el texto
    original. Elimina los que NO aparecen en la fuente (alucinaciones
    del modelo). Determinista, conservador (borra solo lo de alta
    confianza). Registra lo borrado. Devuelve (notas_limpias, borradas).
    """
    def _sin_tildes(s):
        return ''.join(c for c in unicodedata.normalize('NFD', s)
                       if unicodedata.category(c) != 'Mn').lower()

    fuente = _sin_tildes(texto_fuente)
    STOP = {'la','el','los','las','en','de','del','un','una','por','con',
            'para','fechas','eventos','personas','lugares','cifras',
            'citas','textuales','estadisticas','definiciones','terminos',
            'clave','principales','ciudad','pais','region','ano','dia',
            'tema','primera','segunda','tercera','contexto','adicional','nota'}

    lineas_ok, borradas = [], []
    for linea in notas.split('\n'):
        l = linea.strip()
        if not l:
            lineas_ok.append(linea); continue
        if l.startswith('#') or len(l) < 8:
            lineas_ok.append(linea); continue

        cuerpo = re.sub(r'^\s*[\*\-•]\s*', '', l)
        nombres = re.findall(r'\b[A-ZÁÉÍÓÚÑ][a-záéíóúñ]{2,}\b', cuerpo)
        nombres = [nb for nb in nombres if _sin_tildes(nb) not in STOP]
        cifras = re.findall(
            r'\b\d{1,3}(?:\.\d{3})+\b|\b\d+(?:[.,]\d+)?\s*(?:millones?|mil)\b', l)

        nombre_fantasma = False
        if nombres:
```

```
        presentes = sum(1 for nb in nombres if _sin_tildes(nb)[:5] in fuente)
        nombre_fantasma = (presentes == 0)
    cifra_fantasma = any(_sin_tildes(c).replace(' ', '')[:6] not in
                         fuente.replace(' ', '') for c in cifras) if cifras else False

    if nombre_fantasma or cifra_fantasma:
        borradas.append(l)
    else:
        lineas_ok.append(linea)

return '\n'.join(lineas_ok), borradas
```

El método normaliza tanto la fuente como cada línea de la extracción (elimina tildes, normaliza mayúsculas) para tolerar variaciones ortográficas menores, y actúa línea por línea: basta con que un solo nombre o cifra dentro de una línea no aparezca en la fuente para que la línea completa se descarte, en lugar de intentar corregirla o completarla, evitando así introducir una segunda oportunidad de alucinación durante la propia verificación.

d) Detección del modo retórico

Esta función clasifica de forma determinista el modo retórico dominante de una clase (narrativo, expositivo, argumentativo, técnico o instruccional) contabilizando la densidad normalizada de marcadores discursivos propios de cada modo, y sirve para decidir el orden de fusión entre la estructura semántica determinista y los datos extraídos por el LLM, todo ello sin alterar la plantilla fija y rígida de secciones del informe.

Tabla 14 Modos retóricos detectados y su efecto en la redacción

Modo retórico	**Marcadores característicos**	**Efecto sobre la redacción**
Narrativo	Fechas completas, años, expresiones temporales.	Prevalece el orden cronológico.
Expositivo	"Se define como", "consiste en", "por ejemplo".	La estructura semántica antecede a los datos del LLM.
Argumentativo	"Por lo tanto", "sin embargo", "se concluye".	Prioriza relaciones y tesis.

Técnico	Vocabulario de TI, bloques de código entre comillas.	Prioriza ejemplos técnicos capturados por el detector de dominio.
Instruccional	"Primero", "a continuación", "presiona".	Favorece la estructura de procesos.

e) Evolución de la técnica de lectura posicional

El diseño original de este módulo, concebido para contrarrestar el problema 'Lost in the Middle' (Liu et al., 2023), la tendencia de los LLM a prestar mayor atención al inicio y al final del contexto, contemplaba dos llamadas al modelo (texto normal e invertido). Durante las pruebas de implementación con clases extensas se evaluó una tercera lectura completa, que produjo tiempos de ~510 segundos y timeouts frecuentes; la solución adoptada no fue eliminar la tercera lectura, sino acotarla al tercio central del texto en lugar de repetirlo completo. La técnica vigente, normal, invertida y centro acotado, es la que se documenta como definitiva en este capítulo.

Tabla 15 Evolución del módulo de generación del informe

Dimensión	**Diseño original**	**Resultado final**
Lectura posicional	Doble lectura (normal + invertida).	Triple lectura (normal + invertida + centro acotado).
Estructura del informe	5 secciones ancladas a dominio narrativo.	Plantilla universal de 4 secciones + tema central, válida para cualquier asignatura.
Verificación de fidelidad	Confiaba en las reglas del prompt.	Verificación post-generación determinista (método c).

f) Configuración de los modelos de inteligencia artificial

Tabla 16 Parámetros de configuración del pipeline de IA

Parámetro	**Valor**	**Justificación**
Transcripción	faster-whisper, medium, int8	Equilibrio entre calidad y consumo de VRAM/CPU.
Filtrado de voz (VAD)	Activado, silencio ≥ 500 ms	Reduce cómputo y alucinación en silencios.

Modelo LLM	llama3.1:8b	Mayor fidelidad que llama3.2:3b, que alucinaba con más frecuencia.
Invocación LLM	API REST (localhost:11434)	Elimina la sobrecarga de 3-5 s por llamada del método subprocess.
Temperatura	0,2	Respuestas deterministas, fieles al texto fuente.
num_predict	1.800 tokens por llamada	Longitud suficiente por sección sin truncamiento excesivo.
Redacción por secciones	4 llamadas concurrentes (ThreadPoolExecutor, 2 workers)	REDACCION_FUSIONADA=False: cada sección se redacta por separado en lugar de en una sola llamada, lo que en pruebas preservó mejor la atención del modelo por sección.
Flask reloader	use_reloader=False	Evita una segunda instancia de faster-whisper en GPU (CUDA out-of-memory).

4.4.4 Verificación

La fase de verificación es una fase de análisis en la que se prueba y ejecuta el software para determinar si funciona adecuadamente. Se busca metódicamente localizar y corregir errores antes de entregar el producto final. En esta etapa se realizan pruebas de datos en frío (por parte del desarrollador) y pruebas de datos reales (con usuarios finales), que permiten realizar correcciones sobre el sistema.

4.4.4.1 Pruebas de Datos en Frío

Estas pruebas son realizadas por el propio desarrollador, interactuando con el sistema para verificar que funcione de acuerdo con lo planificado o bien detallar alguna observación.

Tabla 17 Prueba de datos en frío: formulario de nueva clase

Objeto	**Tipo de objeto**	**Descripción**	**Observación**

Título	Caja de texto	Permite ingresar el título de la clase.	No permite dejar el campo vacío.
Materia	Caja de texto	Permite especificar la asignatura.	Campo opcional; acepta texto libre.
Etiquetas	Caja de texto	Permite ingresar etiquetas separadas por coma.	Se dividen y almacenan como registros independientes en la tabla tags.
Fuente de audio	Botones	Permite elegir grabar en vivo o cargar un archivo.	Al elegir carga de archivo, valida la extensión antes de aceptar el envío.
Visibilidad	Interruptor (toggle)	Define si la clase es pública o privada.	Por defecto queda en privada; el docente la activa manualmente.
Guardar	Botón	Envía los datos y dispara el pipeline de IA.	Se deshabilita mientras el pipeline procesa la clase, para evitar envíos duplicados.

Tabla 18 Prueba de datos en frío: formulario de búsqueda (estudiante)

Objeto	**Tipo de objeto**	**Descripción**	**Observación**
Filtrador	Caja de texto	Permite ingresar el título o materia a buscar.	Solo devuelve coincidencias entre clases públicas.
Resultados	Lista	Presenta las clases coincidentes.	Muestra un mensaje al no encontrar coincidencias.
Favorito	Ícono	Permite marcar/desmarcar una clase desde el resultado.	Cambia de estado sin recargar la página (AJAX).

4.4.4.2 Prueba de Datos Reales

Este tipo de pruebas son realizadas por personas con características similares al usuario final, es decir, sin conocimientos de programación. En el presente proyecto se seleccionó como evaluadores a docentes y estudiantes de la carrera de Ingeniería en Tecnologías de la Información de la ULEAM Extensión El Carmen, quienes debieron interactuar con el sistema para validar su funcionamiento e intuitividad.

La validación del pipeline de inteligencia artificial, el componente de mayor aporte metodológico del proyecto, se realizó además sobre transcripciones reales de tres clases universitarias de naturaleza marcadamente distinta: requerimientos de software, metodología de la investigación y aprendizaje profundo. De este modo, la selección deliberadamente heterogénea persiguió la confirmación de que la plantilla universal de cuatro secciones se conjuga con el contenido pertinente al margen del modo retórico y de la densidad de cifras y fechas de cada asignatura, y por ello se puntuó cada clase por sección (escala 1-10) cruzando con la transcripción fuente cada afirmación del informe hasta alcanzar la nota que concernía. A su vez, las tres corridas que aquí se documentan lo hacen de forma cronológica y no como una sola medición final, ya que cada clase operó tal que al igual que los hallazgos obtenidos se susceptible de corregirse para ofrecer una respuesta bien concreta hacia el pipeline.

Tabla 19 Prueba de datos reales: calificación por clase

Clase / asignatura	**Nota global**	**Hallazgo principal**
Requerimientos de software	8,5/10 (subió desde 7,8 tras dos barridos de limpieza)	Evidencias pasó de ser la sección más floja a la mejor (100% fiel, dos listas de requerimientos verificadas); Tareas y Avisos 9/10.
Metodología de la investigación	No se le dio una calificación formal	Transcripción con puntuación muy escasa (habla corrida); sirvió principalmente para poner a prueba la re-segmentación por marcadores de discurso.

Aprendizaje profundo (Deep Learning)	8,4/10 (corrida final, con todos los arreglos acumulados)	Alta fidelidad verificada (parámetros exactos, accuracy 99,56%/91%); único defecto real: la sección Desarrollo generó una estructura de "Paso 1/2/3/4" no presente en la fuente (nota 6,5/10 en esa sección); este fue el hallazgo que motivó la prohibición explícita de encabezados fabricados (sección 4.4.5.2).

4.4.4.3 Pruebas de Rendimiento y Comparación de Configuraciones

Además de la validación funcional, se ejecutaron pruebas de rendimiento dirigidas a sustentar con evidencia, y no solo con criterio teórico, dos decisiones de configuración del pipeline: el tamaño del modelo de transcripción (faster-whisper) y el número de lecturas posicionales sobre el LLM. Ambas pruebas se ejecutaron sobre el mismo archivo de audio de prueba (un documental sobre la Primera Guerra Mundial, utilizado por su alta densidad de nombres propios, fechas y cifras, condición exigente para poner a prueba la fidelidad del sistema), en el equipo de validación (RTX 2050, 4 GB VRAM).

a) Comparación de tamaño de modelo: faster-whisper small vs. medium
Se transcribió el mismo audio con ambos tamaños de modelo mediante el script auxiliar test_transcripcion.py (sección 4.4.3.7 en versiones previas de este documento, ahora integrado como herramienta de diagnóstico del proyecto), liberando la VRAM entre una corrida y otra. El análisis de diferencias se realizó a nivel de oración con la librería difflib de Python.

Tabla 20 Comparación empírica de faster-whisper small vs. medium

Criterio	**small**	**medium**
Tiempo de transcripción	~60-70 s (orden de magnitud observado en consola; la cifra exacta no quedó registrada en esa corrida)	~126 s

Oraciones totales analizadas	5.469 palabras transcritas	Base de comparación
Tramos que difieren entre ambos	92 tramos en total, de los cuales ~60 son diferencias cosméticas (tildes, mayúsculas, puntuación) sin impacto en la extracción.	—
Fechas y cifras clave	Sobrevivieron intactas en ambos modelos (p. ej. 28 de junio de 1914, 11 de noviembre a las 11:11, 65 millones, 350.000).	Igual que small en este criterio.
Nombres propios	Mutiló nombres propios reales: "Theodor Roosevelt" por "Theodor Rusbel" o "los Balcanes" por "los balcones".	No presentó este tipo de error.
Alucinación por repetición	Existió un bucle de repetición en un fragmento ("Y no se había hecho nada de eso" repetido) que además borró contenido real expuesto en el audio.	No estuvo presente el bucle de repetición en esta prueba.
Error propio detectado	Acertó una ubicación que medium transcribió mal ya que es "Ypres, en Bélgica" y no "en México" de medium.	Cometió un error de ubicación ("México" en lugar de "Bélgica") en ese mismo tramo.

La decisión de mantener medium como configuración definitiva (documentada en la sección 4.4.3.7) no se basó en el tiempo (el ahorro de small ronda apenas los 60 segundos sobre un audio de varios minutos), sino en el tipo de error: una alucinación que borra contenido real es inaceptable en una herramienta cuyo argumento central es la fidelidad al audio fuente, mientras que el ahorro de tiempo no compensa ese riesgo. Ningún modelo resultó perfecto: medium también cometió un error de ubicación puntual, lo que confirma que la capa de verificación anti-alucinación (sección 4.4.3.3-c) es necesaria independientemente del tamaño de modelo elegido.

b) Comparación de estrategias de lectura posicional

Tabla 21 Tiempos observados por estrategia de lectura posicional

Estrategia	**Configuración**	**Tiempo observado**	**Resultado**
Doble lectura (versión inicial)	2 llamadas a llama3.1:8b (normal + invertida) sobre el mismo audio de prueba	~71 s Whisper + ~156 s Llama ≈ 227 s en total	Adoptada como línea base; capturó datos que una lectura única omitía (p. ej. el armisticio de 1914 y cifras de bajas específicas).
Triple lectura completa (variante descartada)	Tres llamadas de texto completo + redacción paralela por secciones (ocho llamadas a Ollama en total)	~510s, con frecuentes errores de espera (timeout)	Descartada por coste de tiempo; motivo por el/a que se restringe a la tercera lectura al tercio medio en vez de repetir el texto completo.
Triple lectura posicional (vigente)	3 llamadas (normal + invertida + tercio central restringido) + tema central + redacción por secciones	Esperando por la consolidación con mediciones repetidas	Adoptada como estrategia definitiva (sección 4.4.3.3-e); respeta la cobertura del centro del texto sin repetir todo el documento.

4.4.5 Mantenimiento

Después de examinar los resultados de la etapa anterior y aplicar los cambios necesarios para mejorar la aplicación, entra en marcha la fase de mantenimiento, en la cual se entrega el sistema y se documentan los lineamientos para su instalación, uso y mantenimiento posterior.

4.4.5.1 Instalación

El proceso de instalación de ULEAMTranscribe implica transferir el sistema a un nuevo entorno y configurarlo para su uso. A diferencia de un sistema alojado en un hosting compartido con panel de administración gráfico, ULEAMTranscribe se instala mediante línea de comandos sobre un entorno Linux con Python y, preferiblemente, una GPU NVIDIA compatible con CUDA.

Tabla 22 Procedimiento de instalación de ULEAMTranscribe

Paso	Acción	Comando / Descripción
A	Clonar el repositorio	git clone https://github.com/MatteoJVelezH/resumidoraudio.git
B	Crear y activar el entorno virtual	python3 -m venv venv && source venv/bin/activate
C	Instalar dependencias	pip install -r requirements.txt
D	Instalar Ollama y descargar el modelo	curl -fsSL https://ollama.com/install.sh | sh && ollama pull llama3.1:8b
E	Iniciar el servicio de Ollama	ollama serve
F	Ejecutar la aplicación	python app.py
G	Verificar el funcionamiento	Acceder a http://localhost:5000 y registrar una cuenta de docente y estudiante.
H	Exponer la aplicación (opcional)	ngrok http 5000, para pruebas remotas sin servidor dedicado.

4.4.5.2 Limitaciones Conocidas y Trabajo Futuro

Se documentan a continuación las limitaciones identificadas durante la verificación que aún no cuentan con una corrección cerrada:

- Regresión en la sección de Desarrollo: se observó en la corrida de validación de la clase de Aprendizaje Profundo (sección 4.4.4.2) que el pipeline generó una estructura de "Paso 1/2/3/4" genérica y no presente en la fuente ("inspección de archivos", "experimentos con datos"), en lugar de prosa explicativa continua. Se debe a que el detector de forma retórica contó 5 procesos en esa transcripción y la instrucción de la sección Desarrollo hace referencia a que se podría organizar el contenido como de procedimiento/pasos si describe un procedimiento, posibilidad que el modelo de 8B aprovechó para construir una estructura de pasos posible, pero no verificada. La identificación del error era la de prohibir explícitamente en la instrucción de Desarrollo la fabricación de pasos o encabezamientos no presentes en los datos facilitados.
- Propagación de errores de origen ASR: se han documentado casos concretos donde faster-whisper introduce errores que la verificación anti-alucinación no puede corregir, dado que esta solo valida contra la fuente transcrita, no contra el audio original. Ejemplos observados durante las pruebas: mutilación de un nombre propio real ("Theodor Roosevelt" transcrito como "Theodor Rusbel" con el modelo small, sección 4.4.4.3-a); y, en una transcripción de un documental con siete hablantes, omisión del límite de oración previo a la presentación de un participante ("Johannes Malo"), lo que generó arrastre de contexto hacia el hablante anterior, además de inconsistencia en la grafía de su apellido ("Malo" vs. "Malow") entre distintas menciones, un riesgo de atribución incorrecta que el mecanismo de atribución por hablante (sección 4.4.3.3-b) no puede resolver por sí solo, dado que opera sobre el texto ya transcrito.
- Migración a un sistema de colas de tareas en segundo plano, para que el servidor Flask no quede bloqueado durante el procesamiento.
- Exportación del informe en formato PDF o Word.
- Migración de ngrok a un servidor con dirección IP estática para el despliegue institucional.

CAPÍTULO V

EVALUACIÓN DE RESULTADOS

5.1 Introducción

Este capítulo interpreta, a la luz del objetivo específico OE4 (evaluar la efectividad del sistema mediante pruebas con grabaciones reales de clases y la retroalimentación de los usuarios), los resultados que la fase de Verificación del Capítulo IV registró como bitácora técnica. Mientras esa sección documentó cada corrida y cada hallazgo en el orden en que ocurrieron, este capítulo agrupa esos mismos datos por el elemento del sistema que cada prueba puso a prueba, y los contrasta con las necesidades diagnosticadas en el Capítulo III, para determinar en qué medida ULEAMTranscribe responde al problema que motivó su desarrollo.

La evaluación de un proyecto informático puede ser llevada a cabo por dos vías fundamentales. La primera consiste en el juicio de pares, esto es programadores o especialistas del área que no han participado en el proceso de desarrollo, los cuales emiten un dictamen sobre si se han cumplido determinadas especificaciones de calidad; la segunda manera consiste en la simulación: el sistema es sometido a las condiciones reales de uso para las que fue diseñado, de manera que se puede comparar el comportamiento esperado contra el comportamiento observado.

El presente proyecto adoptó la segunda vía. En lugar de un tribunal evaluador externo, ULEAMTranscribe se sometió a grabaciones reales de clases universitarias de distintas asignaturas y a la interacción directa de docentes y estudiantes de la carrera de Ingeniería en Tecnologías de la Información. A esas pruebas se sumó una comparación empírica de configuraciones técnicas bajo la restricción real de hardware del proyecto (4 GB de VRAM). Cada prueba se diseñó para verificar un elemento concreto del sistema frente a un resultado esperado definido de antemano, según se detalla en la sección siguiente.

5.2 Presentación y Monitoreo de Resultados

5.2.1 Planificación del Monitoreo

La tabla siguiente organiza los elementos evaluados durante la fase de Verificación, el método empleado para ponerlos a prueba y el resultado que se esperaba obtener en cada caso.

Tabla 23 Planificación del monitoreo

Elemento de monitoreo	**Método a aplicarse**	**Resultado esperado**
Fidelidad del informe frente a la fuente	Verificación línea por línea de nombres propios y cifras anotados por el modelo contra la transcripción, descartando lo no verificable.	Los datos concretos que aparecen reflejados en el informe provienen del audio de la clase; no aparecen datos falsificados por parte del modelo de IA.
Generalidad de la plantilla ante asignaturas heterogéneas	Prueba del pipeline para asignaturas reales sin vinculación temática.	Las secciones fijas que aparecen en el informe se rellenan con contenido correspondiente a cualquier asignatura, de modo que no hay manipulación manual, por asignatura, de los campos del informe.
Aislamiento de contenido administrativo frente a contenido académico	Prueba dirigida de una transcripción que mezcla avisos de curso con contenido técnico.	El informe cursado excluye avisos administrativos sin que de ninguna manera suponga la eliminación de contenido legítimo dentro del ámbito académico.
Rendimiento bajo la restricción de 4 GB de VRAM	Prueba comparativa empírica del pipeline (tamaño del modelo de transcripción, número de lecturas posicionales) sobre un audio de prueba.	El modelo de IA que soporta la transcripción y la conformación del informe no se equivoca, porque no le falta memoria.

Elemento de monitoreo	Método a aplicarse	Resultado esperado
Funcionalidad e intuitividad de la interfaz	Interacción directa de docentes y estudiantes con los formularios de creación de clase y búsqueda.	Los evaluadores completan las tareas propuestas sin asistencia del desarrollador.
Pertinencia del sistema frente a la necesidad diagnosticada	Contraste entre las dificultades reportadas en el diagnóstico del Capítulo III y las funciones implementadas.	Cada dificultad diagnosticada encuentra una función correspondiente en el diseño final.

5.2.2 Ejecución del Monitoreo

a) Fidelidad del informe frente a la fuente. El mecanismo verificar_notas_contra_fuente (subsección 4.4.3.3-c) se encarga de verificar cada uno de los nombres propios y cada una de las cifras que, en consecuencia, el modelo anotó en las tres lecturas posicionales que realizó, y descartar de forma determinista cualquier línea cuyo dato no se pueda verificar. La ejecución final respecto de la clase de Aprendizaje Profundo, acumulando las correcciones de las dos ejecuciones anteriores, resultó en una nota global de 8,4/10, con los valores técnicos precisos (accuracy del 99,56% y 91%) verificados contra la fuente. Respecto de la clase Requerimientos de Software, dos barridos de limpieza sobre la sección de Evidencias propiciaron que su nota se elevará de 7,8 a 8,5/10, hasta llegar a una fidelidad del 100% en los dos listados de requerimientos que debía reproducir la misma. En ambos casos se cumplió la meta esperada: un informe que tiene como origen los datos concretos del audio y no la extrapolación del modelo.

b) Generalidad de la plantilla ante asignaturas heterogéneas. Para confirmar que la plantilla universal de secciones (Tema central, Conceptos, Desarrollo, Evidencias, Conclusiones) no depende del contenido de una asignatura particular, se probó el pipeline sobre transcripciones reales de tres materias sin relación temática entre sí: Requerimientos de Software, Metodología de la Investigación y Aprendizaje Profundo (Tabla 17 del Capítulo IV). Las asignaturas de Requerimientos de Software y Aprendizaje Profundo han sido evaluadas con notas de 8,5 y 8,4 sobre 10, respectivamente. Por su parte, la asignatura de Metodología de la Investigación no obtuvo nota formal debido a que su transcripción, en forma de discurso corrido y con escasa

puntuación, ha servido en primer lugar para testar la re-segmentación del texto mediante los marcadores de discurso, cosa que hace que esto no vaya en la línea de la transformación de la calidad del informe final. Por lo demás, el resultado previsible, a saber, tres asignaturas diferentes que completan la misma estructura sin intervención manual por parte del desarrollador, se ha dado en la realización de las tres asignaturas, con el matiz de que dentro de la asignatura de Aprendizaje Profundo la sección de Desarrollo ha producido una estructura "Paso 1/2/3/4" confeccionada y ausente de la fuente, cosa que le ha costado una nota de 6,5/10 en esta sección puntual. Este resultado se recupera en la interpretación de la sección 5.3.

c) Rendimiento bajo la restricción de 4 GB de VRAM. Se procedieron a testar empíricamente dos decisiones de configuración con base en el mismo archivo de audio, un documental sobre la Primera Guerra Mundial que fue elegido por la alta densidad de nombres propios, fechas y cifras que contiene. El cambio entre el faster-whisper small y el medium (Tabla 18, Capítulo IV) demostró que el primero ahorra entre 60 y 70 segundos frente al segundo, un margen más pequeño ante el riesgo que introduce: el small mutila nombres propios reales (por ejemplo, "Theodor Roosevelt" se transcribe como "Theodor Rusbel") y genera una especie de bucle de la repetición que cancela contenido real del audio, mientras que el medium no introduce ninguno de esos dos errores en su veredicto sobre la misma prueba. Con toda esa evidencia, se mantuvo el medium como decisión de configuración definitiva, ya que el ahorro temporal no compensa una alucinación que da lugar a la desaparición de contenido real en un sistema cuya propuesta esencial era la de la fidelidad al audio fuente. En cuanto a la comparación entre lectura doble y lectura triple (Tabla 19), en este caso, los resultados confirmaron que añadir una tercera lectura completa, que había producido tiempos cercanos a los 510 segundos con errores de tiempo de espera frecuentes. El resultado esperado, un pipeline que opere dentro del límite de 4 GB de VRAM sin errores de memoria insuficiente, se cumplió en todas las corridas registradas, apoyado además en la ejecución secuencial de las llamadas a Ollama, que evitó los conflictos de memoria observados al probar la ejecución en paralelo.

d) Funcionalidad e intuitividad de la interfaz. Docentes y estudiantes de la carrera de Ingeniería en Tecnologías de la Información interactuaron directamente con los formularios de creación de clase y de búsqueda de clases públicas, sin conocimientos previos de programación (Tablas 15 y 16 del Capítulo IV). El formulario de nueva clase previno la posibilidad de envío de campos vacíos e impidió el envío de formas de guardar en el momento en que el pipeline procesaba el audio tal como había sido programado para evitar envíos duplicados, el formulario de búsqueda únicamente tenía como respuestas coincidencias entre clases públicas y marcaban

o desmarcaban como favorita una clase sin volver a cargar la página; el resultado que esperábamos: los evaluadores debían llevar a cabo ambas tareas sin ayuda del desarrollador, se verificó en las pruebas registradas.

e) Pertinencia del sistema frente a la necesidad diagnosticada. El diagnóstico de capítulo III evidenció dificultades concretas: el 67,5% de los alumnos/as invierte entre 20 y 45 minutos en localizar un tema en una grabación, el 88,8% no puede buscar una materia dentro del audio y el 55% califica como difícil o muy difícil identificar los conceptos principales en clase extensa. En el lado docente, en cambio, los diez entrevistados coincidieron en que el error terminológico, aunque el resto del resumen sea fluido, es un fallo. Cada problema desempeña una función en el resultado del diseño final: la búsqueda por materia o por título y el informe ordenado por conceptos sirven justamente para dar cuenta de la dificultad para navegar el audio por tema; la parte de Conceptos o de anclaje a la extrusión determinista antes de cualquier redacción del modelo da cuenta de las dificultades para extraer las partes principales de una clase excesivamente larga; la verificación anti-alucinaciones va de forma directa también a la falta de tolerancia ante los problemas terminológicos que los docentes no aceptan. Esta correspondencia confirma que el diseño no se apartó del problema que lo originó, con una precisión pendiente: el 73,8% de los estudiantes que en el Capítulo III anticipó la utilidad de una herramienta como esta lo hizo antes de que el sistema existiera, de modo que esa cifra describe una expectativa y no todavía una medición posterior al uso real de la aplicación por esa misma muestra.

5.3 Interpretación Objetiva

La comparación entre el diagnóstico del Capítulo III y los resultados de verificación del Capítulo IV permite valorar, punto por punto, el efecto que tuvo el diseño de ULEAMTranscribe sobre cada dificultad identificada, con la salvedad de que esta evaluación se apoya en pruebas funcionales y de fidelidad, no en una medición cronometrada del uso real de la aplicación por parte de la muestra ya encuestada en el Capítulo III.

La dificultad más mencionada entre los estudiantes, la incapacidad para buscar un tema en concreto en el interior de una grabación (88,8 %), se hace frente con dos mecanismos complementarios a la búsqueda de tema en la búsqueda por título o materia de las clases públicas y a la estructuración del informe con secciones fijas por concepto y no en una transcripción continuada. Ninguno de los mecanismos consigue eliminar por completo el tiempo de búsqueda, pero ambos mecanismos sustituyen la revisión lineal de un audio

completo por una consulta dirigida, justo lo que el 67,5 % de los estudiantes que dedicaba entre 20 y 45 minutos en esa tarea no tenía disponible hasta la llegada del sistema.

La exigencia docente de tolerancia cero frente a errores terminológicos, el criterio de calidad más estricto que arrojó el Capítulo III, encontró su respuesta técnica en la verificación anti-alucinación (sección 4.4.3.3-c), y esa respuesta se puso a prueba con resultados verificables: 100% de fidelidad en las listas de requerimientos de la clase de Requerimientos de Software, y parámetros exactos verificados en la clase de Aprendizaje Profundo. El método tiene un límite que ya indicado en el Capítulo IV: valida los datos anotados por el modelo respecto a la transcripción, no respecto al audio original, de modo que un error de transcripción, como puede ser la mutilación de un nombre propio que hemos constatado con faster-whisper small, puede llegar al informe final sin que la capa de verificación tenga tiempo de detectarlo. Este límite no inhabilita el mecanismo, sino que acota puntualmente qué puede garantizar y qué no.

Le hallazgo más destacado en el marco de todo el conjunto de procesos de verificación, donde el pipeline generó una estructura "Paso 1/2/3/4" que no estaba en la fuente en vez de una prosa explicativa, es la regresión de la sección Negociación de la clase de Aprendizaje Profundo. A diferencia de los demás resultados, este pone en evidencia el punto más débil actual del diseño: la instrucción de Negociación permitía el mantenimiento del contenido por pasos cuando el detector de modalidad retórica detectaba que la transcripción contenía cinco o más procesos, y el modelo 8B utilizó esta opción para proporcionar una secuencia plausible pero carente de verificación. La corrección queda identificada, prohibir explícitamente la fabricación de encabezados o pasos no presentes en los datos suministrados, pero no cerrada al momento de escribir este capítulo, y se documenta como tal en las limitaciones conocidas del Capítulo IV.

Sobre la restricción de hardware, el resultado es más cerrado: el pipeline completo, transcripción y redacción, operó dentro del límite de 4 GB de VRAM en todas las corridas registradas, gracias a la ejecución de faster-whisper en modo int8 y a la ejecución secuencial de las llamadas al modelo de lenguaje. La decisión de mantener la lectura triple acotada al tercio central en vez de una tercera lectura completa muestra que ese cumplimiento no fue un hecho automático: fue producto de la necesidad de descartar una variante, la lectura triplemente completa, que incumplía el margen de tiempo razonable para el usuario final, aun sin incumplir las condiciones que cumplía el límite de memoria.

Las evidencias que aparecen compiladas al final del proceso de verificación demuestran el cumplimiento del objetivo específico OE4 a través de sus componentes de fidelidad,

generalización ante cualquier otra asignatura y funcionalidad de la interfaz. Dos zonas quedan de este modo marcadas como sobresalientes del propio proceso de verificación, a saber: la manufactura ocasional de una estructura no verificada en el apartado desarrollo, y la inexistencia de un mecanismo que detecte y corrija errores de origen proveniente de la transcripción antes de ello se acabe haciendo ir al informe final.

Al contrastar los datos recogidos en el diagnóstico inicial con los tiempos de procesamiento medidos durante la validación técnica, la magnitud del cambio que introduce ULEAMTranscribe se vuelve más concreta. En la encuesta aplicada a 80 estudiantes, el 67,5% reportó dedicar entre 20 y 45 minutos a localizar un único tema dentro de una grabación de clase, mientras que el 88,8% señaló no contar con ningún mecanismo para buscar un tema específico dentro del audio. Frente a ese escenario, el pipeline completo del sistema (transcripción con faster-whisper más la doble lectura posicional sobre Llama 3.1 8B) procesó el audio de prueba en aproximadamente 227 segundos, poco menos de cuatro minutos, para producir un informe estructurado sobre la totalidad de la clase. Aun considerando que la variante de triple lectura posicional actualmente vigente añade una tercera pasada acotada al tercio central del audio (lo que eleva ligeramente ese tiempo respecto a la doble lectura, sin llegar a los 510 segundos de la variante de triple lectura completa que fue descartada), la diferencia de orden de magnitud entre los 20 a 45 minutos que un estudiante invierte manualmente en localizar un solo tema y los pocos minutos que tarda el sistema en generar un informe sobre la clase completa respalda la pertinencia práctica de automatizar este proceso. Esta correspondencia entre el problema diagnosticado en el Capítulo III y los resultados obtenidos en la validación técnica del Capítulo V confirma que ULEAMTranscribe atiende directamente la dificultad identificada como más crítica por los propios estudiantes.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- El objetivo general, desarrollar un sistema que genere resúmenes automáticos de clases grabadas mediante Procesamiento de Lenguaje Natural, se cumplió mediante un pipeline híbrido que ancla la extracción de datos exactos en Python antes de cualquier redacción del modelo de lenguaje, y verifica el resultado contra la fuente después de generarlo. La secuencia, y no el modelo de lenguaje en sí mismo, es lo que mantiene la fidelidad frente a la grabación del audio.

- En cuanto al primer objetivo específico, analizar los requerimientos académicos y tecnológicos necesarios para la implementación del sistema, el diagnóstico del Capítulo III confirmó que el hecho de que los alumnos tuvieran dificultades para utilizar las grabaciones extensas de clase como un material de estudio sí era una dificultad de la que los alumnos eran conscientes y que se podía medir: el 88,8% de los estudiantes no podían hacer el esfuerzo de encontrar qué caía bajo el tema del audio, y el 55% de los estudiantes realizaban la clasificación de difícil o muy difícil a la idea de identificar cuáles eran los conceptos principales de una extensa clase. Las pruebas del Capítulo IV y la evaluación del Capítulo V mostraron que la búsqueda por título o materia y la plantilla organizada por conceptos responden de forma directa a esas dos dificultades.

- El segundo objetivo específico, diseñar la arquitectura del sistema incorporando los módulos de transcripción de audio y de procesamiento de lenguaje natural, se reflejó en decisiones de diseño como la estrategia de triple lectura posicional (normal, invertida y tercio central), adoptada para contrarrestar la pérdida de atención del modelo en el centro de contextos extensos que describe Liu et al. (2023), demostró en las pruebas de rendimiento capturar datos que una lectura única omitía, sin incurrir en los tiempos de espera excesivos que sí produjo una tercera lectura sobre el texto completo.

- El tercer objetivo específico, implementar un prototipo funcional que procese grabaciones de clases y genere resúmenes comprensibles y coherentes, se cumplió a través de dos mecanismos complementarios. Por un lado, la plantilla universal de cuatro secciones más tema central se completó con contenido pertinente en asignaturas sin relación temática entre sí (requerimientos de software, metodología de la investigación y aprendizaje

profundo), lo que respalda la generalidad del sistema frente a distintos modos retóricos y densidades de datos, sin que el desarrollador tuviera que ajustar el pipeline por materia.

- Por otro lado, la verificación post-generación descartó de forma determinista los nombres propios y las cifras que no aparecían en la transcripción fuente, y alcanzó una fidelidad verificable del 100% en las listas de requerimientos y en los parámetros técnicos de la clase de aprendizaje profundo. Este mecanismo, más que el modelo de lenguaje en sí, es el que sostiene la promesa central del proyecto de minimizar el contenido no verificable.

- El cuarto objetivo específico, evaluar la efectividad del sistema mediante pruebas con grabaciones reales de clases y la retroalimentación de los usuarios, se abordó mediante el proceso de verificación técnico del Capítulo V y la retroalimentación recogida en la encuesta a estudiantes y las entrevistas a los diez docentes participantes. Ese mismo proceso también expuso límites que el proyecto no resuelve todavía: la sección Desarrollo puede fabricar una estructura de pasos ausente de la fuente cuando el detector de modo retórico cuenta varios procesos en la transcripción, y la capa de verificación no puede corregir errores que ya vienen introducidos por la transcripción de audio a texto.

6.2 Recomendaciones

A la institución se le recomienda instalar ULEAMTranscribe sobre un equipo con GPU NVIDIA compatible con CUDA y al menos 4 GB de VRAM, la condición mínima bajo la cual se validó el pipeline completo en este trabajo. Un equipo sin GPU dedicada puede ejecutar el sistema, pero los tiempos de transcripción y redacción se extienden de forma considerable al recaer todo el cómputo sobre el procesador. Conviene además planificar una capacitación breve para los docentes sobre el registro de clases y el control de visibilidad, y elaborar un manual de usuario dirigido a quienes no participaron en su desarrollo.

A los docentes se les recomienda mantener la revisión manual del informe antes de publicarlo para los estudiantes, en la línea de lo que propuso uno de los diez docentes entrevistados en el Capítulo III bajo un esquema de co-evaluación docente-IA. La verificación anti-alucinación reduce el riesgo de datos fabricados, pero no sustituye el criterio del docente sobre su propia asignatura, sobre todo en la sección Desarrollo, donde el proceso de verificación documentado en los Capítulos IV y V identificó el mayor margen de error.

A los estudiantes se les recomienda usar los informes generados como un apoyo para organizar el repaso y no como sustituto del audio original, particularmente cuando la clase describe un

procedimiento de varios pasos, el escenario en el que este trabajo detectó el riesgo de una estructura fabricada por el modelo.

Para aquellas personas que sigan esta línea de trabajo en posteriores trabajos de titulación o líneas de investigación relacionadas, se aconseja cerrar la corrección de la instrucción de la sección Desarrollo para prohibir que se genere explícitamente pasos, o encabezados que no se encuentren en la fuente; trasladar el procesamiento del pipeline a un sistema de colas de tareas en segundo plano, de manera que el servidor no se quede bloqueado durante la transcripción y redacción de una clase; y añadir la exportación del informe en formato PDF o Word.

Así mismo, convendría extender la validación del sistema a un número mayor de clases y asignaturas, ya que las corridas que se han documentado en este trabajo solamente han considerado cuatro materias. Una muestra mayor nos podría permitir corroborar de forma más sólida la generalidad de la plantilla en relación a asignaturas con modos retóricos poco representados en las pruebas que hemos presentado, de tipo narrativo puro o argumentativo puro o bien, explorar mecanismos que nos permitan señalar o contrarrestar errores de origen que se introducen en la transformación a la manera de un dictado desde el audio a texto, dado que la capa actual de validación confronta el informe, comparando con la transcripción y no con el audio original.

REFERENCIAS

- Ackerman, S. E., & Com, S. (2013). Metodología de la investigación. Buenos Aires, Argentina: Del Aula Taller.
- Ahlawat, H., Aggarwal, N., & Gupta, D. (2025). Automatic Speech Recognition: A survey of deep learning techniques and approaches. International Journal of Cognitive Computing in Engineering, 6, 201–237. https://doi.org/10.1016/j.ijcce.2024.12.007
- Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., Alharbi, S., Alturki, S., Alshehri, F., & Almojil, M. (2021). Automatic Speech Recognition: Systematic Literature Review. IEEE Access, 9, 131858–131876. https://doi.org/10.1109/ACCESS.2021.3112535
- Álvarez Pincay, D. E., & Herrera Calle, L. I. (2024). Sistema de manejo de control de inventario y la gestión de bodegas en el comercial Osejos, cantón Jipijapa. Ciencia y Desarrollo, 27(4), 531-539. https://doi.org/10.21503/cyd.v27i4.2759
- Asti Vera, A. (2015). Metodología de la investigación. Sevilla, España: Athenaica Ediciones Universitarias.
- Aydın, A. A. (2025). An in-depth Examination of Logical Data Models Utilized in Data Storage Systems to Facilitate Data Modeling. Gazi University Journal of Science, 38(2), 706-729. https://doi.org/10.35378/gujs.1467890
- Baena Paz, G. (2014). Metodología de la investigación. México, D.F.: Grupo Editorial Patria.
- Behar Rivero, D. S. (2019). Metodología de la Investigación. Editorial Shalom.
- Coloma Garófalo, J. A., & Agualongo Caiza, J. M. (2023). Innovación móvil para el seguimiento eficiente del ganado bovino en Hacienda San Gabriel. Polo del Conocimiento, 8(5), 1029-1043. https://doi.org/10.23857/pc.v8i5.5622
- Escobar Alvarez, R. A., Guachamin Quingaluisa, V. G., Andrade Rodríguez, I. D., & Muñoz Rodríguez, E. G. (2023). Aplicación móvil personalizada para la gestión de inventario de productos para la empresa SOLINTEG360. Revista Ingeniería e Innovación del Futuro, 2(2), 47–60. https://doi.org/10.62465/riif.v2n2.2023.52
- Gardazi, N. M., Daud, A., Malik, M. K., Bukhari, A., Alsahfi, T., & Alshemaimri, B. (2025). BERT applications in natural language processing: a review. Artificial Intelligence Review, 58, 166. https://doi.org/10.1007/s10462-025-11162-5
- Gavronskaya, Y., Larchenkova, L., Berestova, A., Latysheva, V., & Smirnov, S. (2022). The Development of Critical Thinking Skills in Mobile Learning: Fact-Checking and Getting Rid of Cognitive Distortions. International Journal of Cognitive Research in Science, Engineering and Education (IJCRSEE), 10(2), 51-68. https://doi.org/10.23947/2334-8496-2022-10-2-51-68
- Ghorbian, M., & Ghobaei-Arani, M. (2026). Large language models for mental health diagnosis and treatment: a survey. Artificial Intelligence Review, 59, 9. https://doi.org/10.1007/s10462-025-11418-0
- Giraldo Forero, A. F., & Orozco Duque, A. F. (2023). Editorial: Evolución del procesamiento natural del lenguaje. TecnoLógicas, 26(56), I-III.
- Hamui-Sutton, A. (2021). La investigación de campo en las ciencias sociales: Retos y oportunidades. Universidad Nacional Autónoma de México.

- Hernández-Sampieri, R., & Mendoza, C. (2018). Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta. McGraw-Hill Education.
- Hou, Y., & Huang, J. (2025). Natural language processing for social science research: A comprehensive review. Chinese Journal of Sociology, 11(1), 121–157. https://doi.org/10.1177/2057150X241306780
- Ilkun, O. P. (2023). Web applications as one of the modern ways of implementing decision support systems. Series: Physics & Mathematics (Taras Shevchenko National University of Kyiv). https://doi.org/10.17721/1812-5409.2023/1.13
- Jácome-Jara, M. S., & Cevallos-Campoverde, M. E. (2022). Aplicación del procesamiento de lenguaje natural para segmentar clientes en una empresa de cobranza. 593 Digital Publisher CEIT, 7(5-2), 99-113. https://doi.org/10.33386/593dp.2022.5-2.1431
- Kökver, Y., Pektaş, H. M., & Çelik, H. (2025). Artificial intelligence applications in education: Natural language processing in detecting misconceptions. Education and Information Technologies, 30, 3035–3066. https://doi.org/10.1007/s10639-024-12919-1
- Li, Y., Huang, Y., Huang, W., Yu, J., & Huang, Z. (2023). An Abstractive Summarization Model Based on Joint-Attention Mechanism and a Priori Knowledge. Applied Sciences, 13(4610). https://doi.org/10.3390/app13074610
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2023). Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, 12, 157-173. https://doi.org/10.1162/tacl_a_00638
- Medranda-Morales, N., Palacios Mieles, V. D., & Villalba Guevara, M. (2023). Reading Comprehension: An Essential Process for the Development of Critical Thinking. Education Sciences, 13(1068). https://doi.org/10.3390/educsci13111068
- Monroy Mejía, M. de los Á., & Nava Sanchezllanes, N. (2018). Metodología de la investigación. México, D.F.: Grupo Editorial Éxodo.
- MS, Z., Rachmadtullah, R., & Iasha, V. (2021). Effectiveness of the use of synthetic analytical structural methods against the ability to begin writing skills in elementary school students. Jurnal Prima Edukasia, 9(1), 16-22. https://doi.org/10.21831/jpe.v9i1.33359
- Quimis Sánchez, O. A., Pincay Pilay, M. M., & Chancay Merchán, X. Y. (2024). Aplicación móvil para la gestión de inventarios y ventas de prendas de vestir para MYPIMES del cantón Pedro Carbo. Polo del Conocimiento, 9(8), 2883-2894. https://www.polodelconocimiento.com/ojs/index.php/es/article/view/7858
- Ramírez, A. (2022). Procesamiento del lenguaje natural y aprendizaje de lenguas extranjeras: Abordaje metodológico desde la realización de una tarea lingüística. Cuaderno Activa, 14, 43-63.
- Royce, W. W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, 26, 1-9.
- Ruan, W., Lyu, Y., Zhang, J., Cai, J., Shu, P., Ge, Y., Lu, Y., Gao, S., Wang, Y., Wang, P., Zhao, L., Wang, T., Liu, Y., Fang, L., Liu, Z., Liu, Z., Li, Y., Wu, Z., Chen, J., … Liu, T. (2025). Large language models for bioinformatics. Quantitative Biology. https://doi.org/10.1002/qub2.70014

- Sterling, N. W., Brann, F., Frisch, S. O., & Schrager, J. D. (2024). Patient-Readable Radiology Report Summaries Generated via Large Language Model: Safety and Quality. Journal of Patient Experience, 11, 1-4. https://doi.org/10.1177/23743735241259477
- Wu, Y., Wan, Y., Chu, Z., Zhao, W., Liu, Y., Zhang, H., Shi, X., Jin, H., & Yu, P. S. (2025). Can Large Language Models Serve as Evaluators for Code Summarization? IEEE Transactions on Software Engineering, 51(1), 1-15. https://doi.org/10.1109/TSE.2025.3595283

ANEXOS

Anexo A

Universidad Laica Eloy Alfaro de Manabí

Periodo 2025-2 - Notificación de tutor asignado - TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Estimad@
Docente y Estudiante
Uleam

En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante:

Tema: APLICACIÓN WEB CON PNL PARA LA IMPLEMENTACIÓN DEL MÉTODO ANALÍTICO-SINTÉTICO EN CLASES DE LA ULEAM, EXTENSIÓN EL CARMEN.

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación.

Apellidos y nombres del tutor asignado: SINCHIGUANO CHIRIBOGA CESAR AUGUSTO

Apellidos y nombres del estudiante: VELEZ HOLGUIN MATTEO JUNIOR

Carrera: TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Periodo de inducción: Periodo 2025-2

Sírvasen cumplir con lo dispuesto en el Manual de Procedimientos de TITULACIÓN DE ESTUDIANTES DE GRADO: TRABAJO DE INTEGRACIÓN CURRICULAR Y UNIDAD DE TITULACIÓN https://departamentos.uleam.edu.ec/gestion-aseguramiento-calidad/files/2020/06/PAT-04-Titulacion-de-Estudiantes-de-grado-UIC-y-UT.pdf

Particular que se informa para los fines consiguientes.

Atentamente,

Comisión Académica y Responsable de Titulación.

Anexo A: Asignación de tutor

Anexo B

Analysis attestation
Compilatio Magister+ | ULEAM-ECU

MATTEO VÉLEZ
ID : c1c6afc694f0902e3a39333afh80fca78d0c0ae1d

10%
Suspicious texts

File name : MATTEO VÉLEZ.txt
Original file size : 4.7 MB
Number of words : 29,676

Submitter : CESAR SINCHIGUANO CHIRIBOGA
Submission date : July 30, 2026
Upload type : interface
analysis end date : July 30, 2026

Summary (section 1/2)

Location of suspect texts in the document :

Included in the suspicious text score :

Similarities 1%
Passages with similarities to sources found in different collections.

AI detection 5%
Texts with stylistically similar formulations to AI-generated text.
This rate is an indicator, not proof. Check with the author that he/she has mastered the knowledge mentioned in the document.

Unrecognized languages 5%
Passages in which some of the vocabulary used is not part of the language dictionary. This may be an attempt by the author to modify the text to make detection impossible.

Not included in the percentage of suspicious texts :

Texts between quotes
Passages between quotation marks, often revealing a quotation.

Anexo B: Reporte del sistema antiplagio

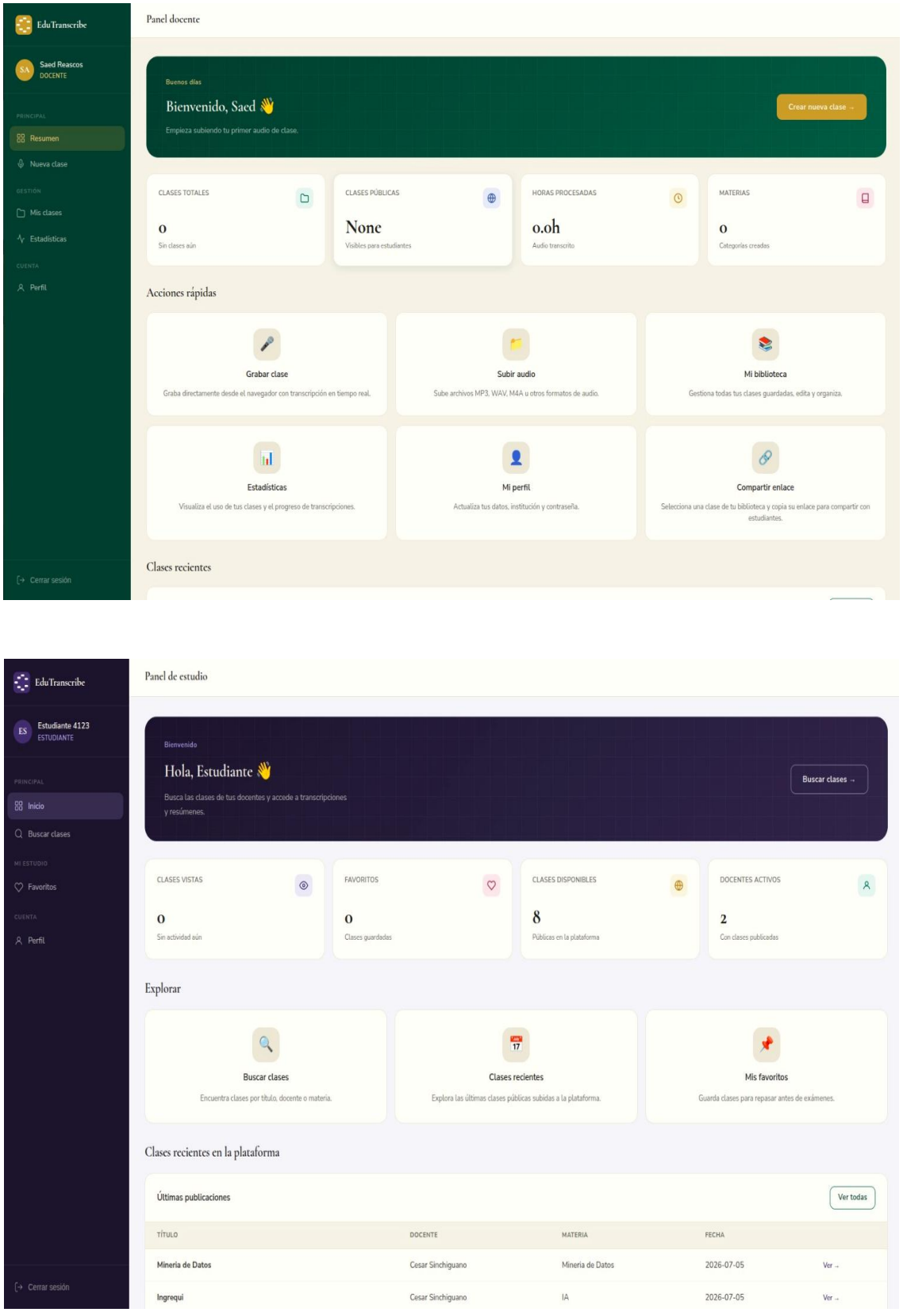
Anexo C

Anexo C: Tutoría con Docente tutor

Anexo D

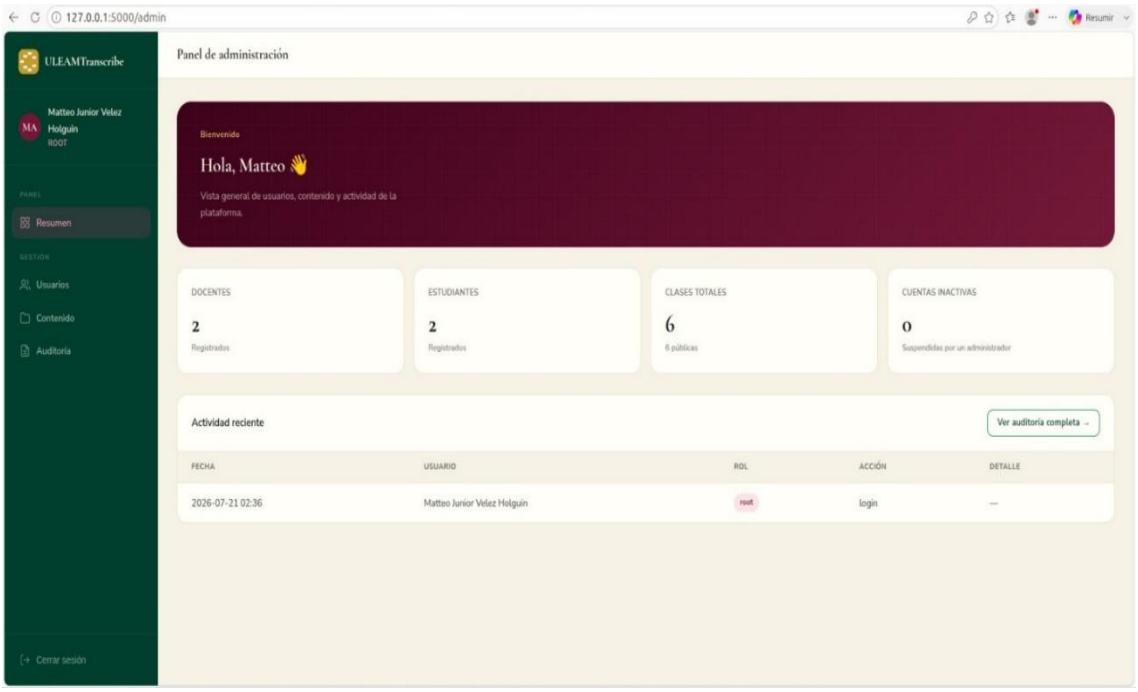
Anexo D Entrevista a coordinador de carrera

Anexo E



Anexo E Interfaz de escritorio

Anexo F



Anexo F Interfaz de escritorio para administrador

Anexo G

Anexo G Adaptación de interfaz para entorno móvil

Anexo H

Entrevista dirigida a docentes

La guía de entrevista diseñada para los docentes consta de ocho preguntas abiertas que exploran dimensiones pedagógicas, técnicas y prácticas relacionadas con la implementación del sistema.

Nombre del entrevistado: Reascos Pinchao Raul Saed

Pregunta 1: ¿Podría describir la estructura típica de sus clases y los principales tipos de contenido que aborda en una sesión de 60 a 90 minutos?

GENERALMENTE, UNA CLASE TIENE UNA DURACIÓN DE DOS HORAS (120 MINUTOS), AUNQUE UNA PARTE DE ESE TIEMPO SE DESTINA AL REGISTRO DE ASISTENCIA Y A LA EVALUACIÓN, POR LO TANTO EL TIEMPO EFECTIVO DE CLASE ES DE APROXIMADAMENTE 90 MINUTOS. DURANTE ESE TIEMPO, PRIMERO SE PRESENTA EL CONTENIDO TEÓRICO, LUEGO SE DESARROLLA UNA ACTIVIDAD PRÁCTICA PARA APLICAR LO APRENDIDO Y, FINALMENTE, SE REALIZA UNA EVALUACIÓN.

Pregunta 2: En su experiencia, ¿cuáles son los conceptos o elementos que considera absolutamente esenciales que los estudiantes deben retener de cada clase?

LA PARTE TEÓRICA ES EL ELEMENTO MÁS IMPORTANTE, YA QUE LOS CONCEPTOS FUNDAMENTALES PERMANECEN EN EL TIEMPO. AUNQUE CAMBIAN LAS HERRAMIENTAS, LOS TIPOS DE PROBLEMAS Y LOS EJEMPLOS, LA BASE TEÓRICA SE MANTIENE Y SIRVE DE FUNDAMENTO PARA SU APLICACIÓN.

Pregunta 3: ¿Qué terminología técnica especializada utiliza regularmente en sus clases que podría representar un desafío para sistemas de transcripción automática?

EN LA CARRERA SE UTILIZAN NUMEROSOS TÉRMINOS EN INGLÉS Y ABREVIACIONES, LO QUE PUEDE DIFICULTAR LAS TRANSCRIPCIONES AUTOMÁTICAS. POR ELLO, ES RECOMENDABLE CONTAR CON UN GLOSARIO DE TÉRMINOS PARA FACILITAR SU COMPRENSIÓN E INTERPRETACIÓN.

Pregunta 4: Desde su perspectiva docente, ¿cómo describiría el método analítico-sintético y qué valor pedagógico encuentra en su aplicación?

COMO SIEMPRE SE PARTE DE LA TEORÍA HACIA LA PRÁCTICA, EL MÉTODO ANALÍTICO-SINTÉTICO CUMPLE PRECISAMENTE ESA FUNCIÓN. PRIMERO SE ANALIZA LA PARTE TEÓRICA Y LUEGO SE APLICA O TRANSFIERE A LA PRÁCTICA SEGÚN LA CONFIGURACIÓN DEL PROBLEMA QUE SE ESTÉ RESOLVIENDO. ESTE MÉTODO PERMITE QUE EL ESTUDIANTE TRANSITE DE LA TEORÍA A LA PRÁCTICA Y VICEVERSA, LOGRANDO UNA MAYOR COMPRENSIÓN DE LOS CONCEPTOS.

Pregunta 5: ¿Qué aspectos específicos de una explicación o desarrollo temático considera que deben preservarse en un resumen para mantener su valor educativo?

EN TODOS LOS TEMAS ES IMPORTANTE EXTRAER LOS CONCEPTOS PRINCIPALES O ELABORAR RESÚMENES DE LOS CONCEPTOS PARA QUE SEAN CLAROS. ESOS SON LOS ELEMENTOS QUE EL ESTUDIANTE DEBE RETENER, YA QUE, AUNQUE LOS PROBLEMAS Y LAS HERRAMIENTAS CAMBIEN, LOS CONCEPTOS FUNDAMENTALES PERMANECEN Y PERMITEN APLICARLOS EN DIFERENTES CONTEXTOS.

Pregunta 6: En su opinión, ¿cuáles serían los principales riesgos o limitaciones de automatizar la generación de resúmenes de clases?

EL PRINCIPAL RIESGO SERÍA QUE EL ESTUDIANTE NO REVISE LOS RESÚMENES DE MANERA CONTINUA Y ÚNICAMENTE LOS CONSULTE EL DÍA DE LA EVALUACIÓN. EN ESE SENTIDO, EL RIESGO DEPENDE DE CÓMO EL DOCENTE UTILICE LOS RESÚMENES. SI SE EMPLEAN ÚNICAMENTE AL FINAL DEL PROCESO, EL RIESGO ES MAYOR; EN CAMBIO, SI SE UTILIZAN DURANTE EL DESARROLLO DE LA ASIGNATURA COMO UNA HERRAMIENTA DE APOYO, PUEDEN RESULTAR MUY ÚTILES.

Pregunta 7: ¿Qué criterios utilizaría para evaluar si un resumen automático de su clase es de calidad aceptable?

CONSIDERARÍA QUE UN RESUMEN ES DE CALIDAD SI MANTIENE LOS CONCEPTOS TEÓRICOS PRINCIPALES JUNTO CON DEFINICIONES BREVES Y CLARAS. LAS ACTIVIDADES PROPIAS DE LA CLASE, COMO LA REALIZACIÓN DE EJERCICIOS O EL SEGUIMIENTO DEL TRABAJO DE LOS ESTUDIANTES, NO DEBERÍAN FORMAR PARTE DEL RESUMEN. EN CAMBIO, SÍ DEBERÍAN INCLUIRSE LAS EXPLICACIONES TEÓRICAS Y LAS INSTRUCCIONES RELEVANTES IMPARTIDAS DURANTE LA CLASE.

Pregunta 8: ¿Cómo imagina que una herramienta de este tipo podría integrarse en sus estrategias de enseñanza actuales?

PODRÍA UTILIZARSE UNA HERRAMIENTA QUE, MIENTRAS EL DOCENTE REALICE LA EXPLICACIÓN TEÓRICA, GENERE AUTOMÁTICAMENTE UN RESUMEN. POSTERIORMENTE, ESE RESUMEN PODRÍA ALIMENTAR UN BANCO DE PREGUNTAS QUE SIRVA COMO BASE PARA LAS EVALUACIONES CONTINUAS REALIZADAS A LO LARGO DEL CURSO.

Firma del entrevistado:

Entrevista dirigida a docentes

La guía de entrevista diseñada para los docentes consta de ocho preguntas abiertas que exploran dimensiones pedagógicas, técnicas y prácticas relacionadas con la implementación del sistema.

Nombre del entrevistado: Arevalo Hermida Romulo Danilo

Pregunta 1: ¿Podría describir la estructura típica de sus clases y los principales tipos de contenido que aborda en una sesión de 60 a 90 minutos?

POR LO GENERAL, EN LA MAYORÍA DE CLASES TRATO DE INICIAR CON UNA EXPLICACIÓN TEÓRICA DE LOS TEMAS O CONTENIDOS QUE SE VAN A DESARROLLAR. LUEGO RESUELVO PEQUEÑOS EJERCICIOS PRÁCTICOS RELACIONADOS CON ESOS CONTENIDOS Y, FINALMENTE, LOS ESTUDIANTES REALIZAN ACTIVIDADES PRÁCTICAS PROPUESTAS PARA APLICARLO APRENDIDO.

Pregunta 2: En su experiencia, ¿cuáles son los conceptos o elementos que considera absolutamente esenciales que los estudiantes deben retener de cada clase?

LOS ASPECTOS ESENCIALES DEPENDEN DE LA CLASE, PERO GENERALMENTE SON LOS CONCEPTOS FUNDAMENTALES DE LA TEMÁTICA O DE LA PRÁCTICA QUE SE ESTÁ REALIZANDO. TAMBIÉN CONSIDERO IMPORTANTES LAS ACTIVIDADES Y EJEMPLOS PRINCIPALES DESARROLLADOS DURANTE LA SESIÓN.

Pregunta 3: ¿Qué terminología técnica especializada utiliza regularmente en sus clases que podría representar un desafío para sistemas de transcripción automática?

NORMALMENTE SE UTILIZA TERMINOLOGÍA TÉCNICA PROPIA DEL LENGUAJE O DE LA ASIGNATURA QUE SE ESTÁ IMPARTIENDO. ADEMÁS, LA MAYORÍA DE LOS LENGUAJES DE PROGRAMACIÓN EMPLEAN TÉRMINOS EN INGLÉS Y UTILIZAN SIGLAS SIGLAS QUE PODRÍAN NO TRANSCRIBIRSE DE MANERA ADECUADA.

Pregunta 4: Desde su perspectiva docente, ¿cómo describiría el método analítico-sintético y qué valor pedagógico encuentra en su aplicación?

EL ANÁLISIS Y LA SÍNTESIS SON MUY IMPORTANTES EN TODAS LAS ASIGNATURAS/ YA QUE PERMITEN QUE EL ESTUDIANTE COMPRENDA UN PROBLEMA Y SEA CAPAZ DE DESARROLLARLO. EN EL CASO DE LOS RESÚMENES DE CLASE, LA SÍNTESIS ES FUNDAMENTAL PARA SELECCIONAR LOS ELEMENTOS MÁS IMPORTANTES Y LOGRAR UN RESUMEN DE BUENA CALIDAD.

Pregunta 5: ¿Qué aspectos específicos de una explicación o desarrollo temático considera que deben preservarse en un resumen para mantener su valor educativo?

LOS ASPECTOS ESPECÍFICOS DEPENDEN DE CADA TEMA, PERO UN BUEN RESUMEN DEBE CONSERVAR LAS CARACTERÍSTICAS PRINCIPALES DE LA TEMÁTICA ABORDADA Y LOS PUNTOS MÁS IMPORTANTES DE LA PEDAGOGÍA. EN MI CASO, LA MAYORÍA DE LAS ASIGNATURAS SON PRÁCTICAS, POR LO QUE ESOS ELEMENTOS DEBEN MANTENERSE.

Pregunta 6: En su opinión, ¿cuáles serían los principales riesgos o limitaciones de automatizar la generación de resúmenes de clases?

UN RIESGO IMPORTANTE ES QUE EL ALGORITMO DEJE DE LADO ASPECTOS QUE, AUNQUE SEAN RELEVANTES, NO LOS CONSIDERE IMPORTANTES. ESTO PODRÍA PROVOCAR LA OMISIÓN DE PARTES ESENCIALES DE LA CLASE QUE DEBERÍAN MANTENERSE EN EL RESUMEN.

Pregunta 7: ¿Qué criterios utilizaría para evaluar si un resumen automático de su clase es de calidad aceptable?

CONSIDERARÍA PRINCIPALMENTE LA PRECISIÓN DE LOS CONCEPTOS Y TEMAS PRESENTADOS EN EL RESUMEN. TAMBIÉN EVALUARÍA LA EXTENSIÓN, YA QUE MIENTRAS MÁS CONCISO, CLARO Y ESPECÍFICO SEA EL RESUMEN, MEJOR CALIDAD TENDRÁ.

Pregunta 8: ¿Cómo imagina que una herramienta de este tipo podría integrarse en sus estrategias de enseñanza actuales?

SERÍA UNA HERRAMIENTAS MUY ÚTIL PORQUE AUTOMATIZARÍA LA ELABORACIÓN DE LOS RESÚMENES DE CADA CLASE. ESTO FACILITARÍA AL DOCENTE DISPONER DE MATERIAL DE APOYO PARA COMPARTIR CON LOS ESTUDIANTES EN LAS AULAS VIRTUALES. ADEMÁS, LOS RESÚMENES PERMITIRÍAN A LOS ESTUDIANTES REPASAR LOS CONCEPTOS VISTOS EN CLASE DE FORMA MÁS RÁPIDA Y ESPECÍFICA, SIN NECESIDAD DE REVISAR TODA LA GRABACIÓN DE LA CLASE O LEER GRANDES CANTIDADES DE TEXTO.

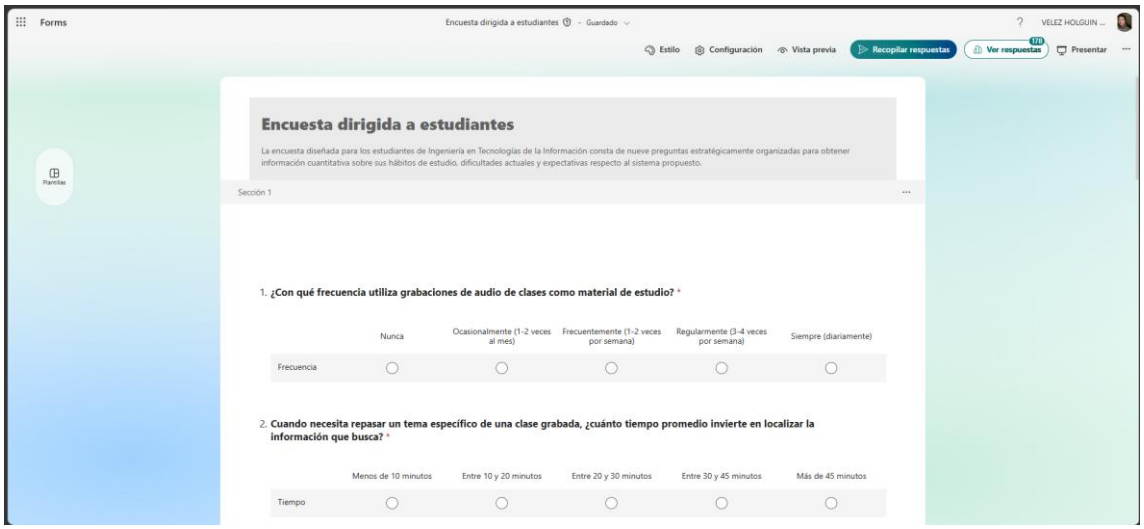
Firma del entrevistado:

Anexo H Evidencia de aplicación de entrevistas

Anexo I

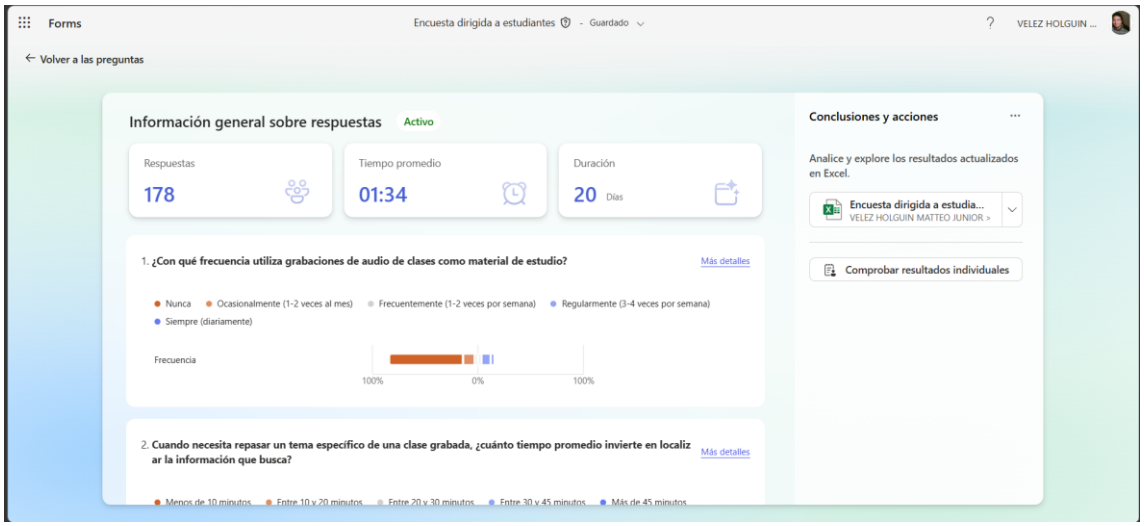
Anexo I Aplicación de entrevista

Anexo J



Anexo J Encuesta aplicada a estudiantes

Anexo K



Anexo K Resultados obtenidos de la encuesta

GLOSARIO

PLN / NLP. Rama de la inteligencia artificial y la lingüística computacional que examina cómo analizan, interpretan y producen lenguaje humano los sistemas informáticos, ya sea este lengua hablada o escrita. Agrupa métodos estadísticos, reglas gramaticales, modelos de aprendizaje profundo, y es de ahí de donde surgen tareas como la traducción automática, el resumen de texto o el reconocimiento de entidades.

LLM. Es un modelo de aprendizaje profundo entrenado con volúmenes enormes de texto para predecir y generar secuencias de lenguaje. Casi siempre se construye sobre una arquitectura transformer y se mide por la cantidad de parámetros, que llega a los miles de millones. Eso le permite resolver tareas de comprensión y generación distintas sin necesitar un entrenamiento adicional para cada una.

ASR. La abreviatura ASR se traduce a su vez como Automatic Speech Recognition, y hace referencia a la tecnología que permite pasar de una señal de audio con habla humana a un texto escrito. Dicho proceso fusiona un modelo acústico (que tiene por objetivo el reconocimiento de los distintos sonidos producidos) con un modelo de lenguaje, el cual tiene como fin el cálculo de la probabilidad de distintas secuencias de palabras, y un decodificador, que une ambos elementos por el que se produce la transcripción final.

Whisper. Es un modelo de reconocimiento de voz de código abierto, con una arquitectura transformer dividida en codificador y decodificador. Procesa el audio como espectrogramas y va generando el texto de forma progresiva. Admite tanto transcripción como traducción entre idiomas distintos.

faster-whisper. Es una variante de esa misma implementación, orientada a un menor consumo de memoria y una mayor rapidez, a partir de técnicas de optimización de inferencia, manteniendo el modelo base sin modificar.

VAD. Son las siglas de Voice Activity Detection. Es la técnica de procesamiento de señales que detecta qué segmentos de un audio contienen voz humana y cuáles son silencio o ruido de fondo, para descartar o recortar los tramos sin contenido hablado antes de seguir procesando.

Llama. Se trata de una familia de modelos de lenguaje a gran escala, que es accesible en diferentes tamaños de parámetros y cuyo objetivo es trabajar tanto en infraestructura de la nube como en hardware local de consumo.

Ollama. Es una herramienta de software que permite descargar, administrar y ejecutar modelos de lenguaje de gran escala en una computadora local. Expone una interfaz a la que otras aplicaciones le pueden enviar solicitudes sin depender de un servicio externo en la nube.

Transformer. Es una arquitectura de red neuronal que procesa una secuencia completa en paralelo, apoyada en mecanismos de atención, en vez de recorrerla elemento por elemento como hacían las redes recurrentes anteriores. Sobre esa base se construyen hoy la mayoría de los modelos de lenguaje.

BERT. Es un modelo de lenguaje, usa solo la parte codificadora de un transformer y se entrena de forma bidireccional, así que interpreta el contexto de una palabra a partir de las que la rodean en ambas direcciones a la vez. Se aplica sobre todo en tareas de comprensión de texto, como clasificar o extraer información, más que en generarlo.

GPT. Es una familia de modelos de lenguaje, usa la parte decodificadora de un transformer y se entrena de forma autorregresiva, prediciendo la siguiente palabra de una secuencia una y otra vez. Su fuerza está en generar texto extenso y coherente.

Token / Tokenización. Es un roceso mediante el cual se divide un texto en unidades más pequeñas llamadas tokens, que según el modelo pueden ser palabras completas, fragmentos de palabra o incluso caracteres sueltos.

Embedding. Es un vector numérico de dimensión fija que representa una palabra, una frase o un documento entero, construido de modo que los elementos con significados parecidos terminen ubicados cerca unos de otros dentro de ese espacio.

TF-IDF. Es un método estadístico para medir mide qué tan importante es una palabra dentro de un documento comparado con una colección más amplia de documentos: combina cuántas veces aparece en ese texto puntual con cuán rara es en el resto de la colección

N-grama. Es una secuencia de N palabras consecutivas tomada de un texto. Con una sola palabra se habla de unigrama, con dos de bigrama, y así en adelante. Sirven para estimar la probabilidad de una secuencia en los modelos de lenguaje estadísticos.

HMM. (Hidden Markov Model) Es un modelo estadístico que representa un sistema como una cadena de estados que no se pueden observar directamente, donde cada estado produce una observación visible con cierta probabilidad. Durante años fue la base del reconocimiento de voz, porque permitía relacionar sonidos con palabras.

LSTM. Es una variante de red neuronal recurrente construida para resolver el problema de olvidar información a largo plazo. Lo logra con compuertas internas que deciden qué conservar, qué actualizar y qué descartar a medida que avanza por la secuencia.

Alucinación / Hallucination. En modelos de lenguaje, la generación de información que un modelo de lenguaje genera con apariencia coherente y plausible, pero que no corresponde a hechos reales ni tiene respaldo en ninguna fuente que se le haya entregado al modelo.

Prompt. Es el texto de entrada que se le da a un modelo de lenguaje para indicarle qué tarea tiene que resolver, cómo estructurar la respuesta o qué información debe tener en cuenta al generarla

Cuantización / Quantization. Es una técnica de optimización de modelos de aprendizaje automático que reduce la cantidad de bits que se usan para representar los pesos de una red neuronal. El modelo ocupa menos memoria y corre más rápido, aunque pierde algo de precisión numérica en el proceso.

VRAM. Es la memoria integrada en una tarjeta gráfica, separada de la RAM principal del sistema, donde se guardan los datos que la GPU necesita procesar de inmediato.

GPU. Es una unidad de procesamiento diseñada originalmente como procesador especializado en renderizar gráficos, pero su arquitectura de miles de núcleos pequeños trabajando en paralelo también le sirve a las operaciones matriciales que exigen las redes neuronales.

CUDA. Es la plataforma de computación paralela de NVIDIA que permite usar la GPU para cálculos de propósito general más allá de los gráficos, incluido el entrenamiento y la inferencia de modelos de IA.

Flask. Es un framework de desarrollo web para Python. Se lo clasifica como microframework porque solo trae los componentes básicos de enrutamiento y manejo de peticiones HTTP, y deja que el desarrollador agregue las bibliotecas adicionales que necesite.

Framework. Es un conjunto de bibliotecas, convenciones y herramientas que ofrece una base estructurada para construir software: define cómo se organizan los distintos componentes de una aplicación en lugar de que cada uno se diseñe desde cero.

Backend. Es la parte de una aplicación que corre en el servidor y que el usuario nunca ve directamente. Ahí vive la lógica de negocio, el acceso a la base de datos y el procesamiento de las solicitudes que llegan desde la interfaz.

Frontend. Es la parte con la que el usuario interactúa de forma directa desde el navegador: la interfaz visual, los formularios y el comportamiento que ocurre del lado del cliente.

API REST. Es una interfaz de programación de aplicaciones que 28. sigue los principios de REST (Representational State Transfer): identifica los recursos mediante URLs y los manipula con verbos HTTP estándar como GET, POST, PUT y DELETE.

SQL / SQLite. Es el lenguaje estandarizado para definir, consultar y manipular datos en bases de datos relacionales. SQLite es un motor concreto que implementa ese lenguaje sin necesitar un proceso de servidor aparte, porque guarda toda la base de datos en un único archivo.

CRUD. Es el acrónimo de las cuatro operaciones básicas que cualquier sistema de gestión de datos permite hacer sobre un registro: crear, leer, actualizar y eliminar (Create, Read, Update, Delete).