UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE SOFTWARE

Tema:

TRANSFORMACIÓN DE VIDEOS EDUCATIVOS EN MATERIALES DE APRENDIZAJE ESTRUCTURADOS BASADA EN INTELIGENCIA ARTIFICIAL GENERATIVA

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero de Software

ÁREA: Inteligencia artificial
LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la Información y Ciencia de Datos

AUTOR: José Sebastián Camino Moreno
TUTOR: Ing. Félix Oscar Fernández Peña, MSc, PhD.

Ambato – Ecuador
julio – 2026

APROBACIÓN DEL TUTOR

En calidad de Tutor del Trabajo de Titulación con el tema: TRANSFORMACIÓN DE VIDEOS EDUCATIVOS EN MATERIALES DE APRENDIZAJE ESTRUCTURADOS BASADA EN INTELIGENCIA ARTIFICIAL GENERATIVA, desarrollado bajo la modalidad Proyecto de Investigación por el señor José Sebastián Camino Moreno, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta la conclusión, de acuerdo a lo dispuesto en el Artículo 17 de Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 6.3 del respectivo instructivo.

Ambato, julio 2026

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FELIX OSCAR
FERNANDEZ PEÑA

Ing. Félix Oscar Fernández Peña, MSc, PhD.
TUTOR

AUTORÍA

El presente trabajo de Titulación con el tema: TRANSFORMACIÓN DE VIDEOS EDUCATIVOS EN MATERIALES DE APRENDIZAJE ESTRUCTURADOS BASADA EN INTELIGENCIA ARTIFICIAL GENERATIVA, es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio 2026

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JOSE SEBASTIAN CAMINO MORENO

José Sebastián Camino Moreno
C.C 0504427758
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio 2026

José Sebastián Camino Moreno
C.C 0504427758
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor José Sebastián Camino Moreno estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, bajo la Modalidad Proyecto de Investigación, titulado TRANSFORMACIÓN DE VIDEOS EDUCATIVOS EN MATERIALES DE APRENDIZAJE ESTRUCTURADOS BASADA EN INTELIGENCIA ARTIFICIAL GENERATIVA, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 6.4 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con el señor Presidente de Tribunal

Ambato, julio 2026.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
OSCAR FERNANDO IBARRA TORRES

Ing. Oscar Fernando Ibarra Torres Mg.
PROFESOR CALIFICADOR

Santiago David Jara Moya
Time Stamping Security Data

Ing. Santiago David Jara Moya Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico este proyecto de investigación, fruto de años de esfuerzo y perseverancia, a las personas que hicieron posible que llegara hasta aquí. A mi padre, por enseñarme las primeras bases que dieron forma a mi camino profesional y darme las herramientas con las que hoy construyo mi futuro. A mi madre, mujer incansable que, sola, dio todo por sacarme adelante; este logro también es suyo. A mi pareja, que nunca soltó mi mano, incluso en los momentos más oscuros, y me dio razones para seguir de pie. A todos ellos, gracias, porque este trabajo es la prueba de que, pese a todo, se puede llegar.

José Sebastián Camino Moreno

ÍNDICE GENERAL DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN EJECUTIVO

Este trabajo aborda la transformación automatizada de videos educativos en materiales de aprendizaje estructurados mediante inteligencia artificial generativa. El objetivo fue desarrollar y validar un sistema que genere, a partir de un video, un resumen, un glosario, tarjetas de estudio y un cuestionario de opción múltiple.

Un diagnóstico aplicado a 129 estudiantes confirmó esas carencias y derivó en doce requerimientos funcionales y diez no funcionales. El aporte central fue una solución independiente, construida sobre NestJS con procesamiento asíncrono por colas. Los datos persistieron en PostgreSQL y se utilizó Prisma como ORM. La interfaz se desarrolló en Vue. Se utilizó el modelo Whisper para la transcripción de audio, y la generación de los materiales de aprendizaje estructurados se basó en un modelo de lenguaje de gran escala *opensource*. La solución se expuso en una API, lo que permitió su integración a la plataforma educativa NousAI. De esta manera se demostró su aplicabilidad en condiciones reales de uso.

Sobre un corpus de 28 videos, el procesamiento alcanzó una tasa de éxito del 92,86 %. La validez del contenido generado, valorada por 67 estudiantes, se ubicó entre 4,30 y 4,48 sobre 5. La aceptación SUS alcanzó 82,26 puntos en el subgrupo con preferencia lectoescritora, ubicando la usabilidad del sistema en el rango excelente. Los resultados confirman que la solución genera materiales válidos en un entorno académico real, lo que respalda su adopción como apoyo al proceso de aprendizaje.

Palabras clave: videos educativos, inteligencia artificial generativa, modelos de lenguaje de gran escala, reconocimiento automático de voz, materiales de aprendizaje, usabilidad.

ABSTRACT

This work addresses the automated transformation of educational videos into structured learning materials using generative artificial intelligence. The objective was to develop and validate a system that generates, from a video, a summary, a glossary, study cards and a multiple-choice quiz.

A diagnostic applied to 129 students confirmed these gaps and led to twelve functional and ten non-functional requirements. The main contribution was a standalone solution, built on NestJS with asynchronous queue processing. Data was persisted in PostgreSQL and Prisma was used as the ORM. The interface was developed in Vue. The Whisper model was used for audio transcription, and the generation of the structured learning materials relied on an *open-source* large language model. The solution was exposed through an API, which allowed its integration into the NousAI educational platform. In this way its applicability under real usage conditions was demonstrated.

Over a corpus of 28 videos, processing reached a 92.86 % success rate. The validity of the generated content, rated by 67 students, ranged between 4.30 and 4.48 out of 5. SUS acceptance reached 82.26 points in the subgroup with a read–write preference, placing the system's usability in the excellent range. The results confirm that the solution generates valid materials in a real academic setting, supporting its adoption as a tool for the learning process.

Keywords: educational videos, generative artificial intelligence, large language models, automatic speech recognition, learning materials, usability.

CAPÍTULO I

MARCO TEÓRICO

1.1 Tema de Investigación

TRANSFORMACIÓN DE VIDEOS EDUCATIVOS EN MATERIALES DE APRENDIZAJE ESTRUCTURADOS BASADA EN INTELIGENCIA ARTIFICIAL GENERATIVA

1.1.1 Planteamiento del problema

El video se ha vuelto el formato dominante de aprendizaje en la educación superior, sobre todo en disciplinas técnicas y de ingeniería. Las plataformas de contenido abierto concentran buena parte del material que los estudiantes consumen de forma autodidacta. Esta expansión no vino acompañada de recursos que atiendan la naturaleza transitoria del medio audiovisual ni las limitaciones cognitivas que impone al estudiar contenido técnico denso. El problema que aborda esta investigación es la ausencia de una solución automatizada que transforme el contenido de un video educativo en materiales de aprendizaje textuales y estructurados, accesibles de forma no lineal, consultables y alineados con prácticas de estudio activo.

Los estudiantes universitarios recurren con frecuencia a videos de clase y material audiovisual externo como insumo central para el estudio autónomo. Esta dependencia genera una carga adicional de trabajo. El estudiante termina transcribiendo a mano, resumiendo y reorganizando el contenido para poder estudiarlo de forma efectiva. Es una tarea mecánica que recorta el tiempo disponible para el análisis y la práctica. La carencia de una herramienta que automatice la generación de resúmenes estructurados, glosarios técnicos, flashcards y cuestionarios a partir del video se vuelve, en este escenario, un obstáculo concreto para la gestión eficiente del aprendizaje.

Esta investigación se desarrolla dentro del proyecto aprobado por el Consejo de Investigación e Innovación de la Universidad Técnica de Ambato mediante Resolución UTA-CONIN-2025-0261-R, de 25 de julio de 2025, titulado “Metodología para el uso de la inteligencia artificial generativa en la educación superior”. El proyecto está

coordinado por el PhD. Félix Oscar Fernández Peña, y cuenta con el Mg. Dennis Vinicio Chicaiza Castillo como coordinador subrogante. El resultado de este trabajo de titulación se integró satisfactoriamente a NousAI, plataforma educativa de la Universidad Técnica de Ambato, demostrando así la aplicabilidad de la solución propuesta como parte de una plataforma educativa institucional. La API y los componentes de procesamiento descritos en los capítulos siguientes constituyen el aporte original del autor; NousAI es el espacio de despliegue y validación en el que la solución se puso en operación.

1.2 Antecedentes investigativos

1.2.1 Contextualización del problema

Durante la presente década, y en buena medida por la crisis sanitaria global, el aprendizaje electrónico (*e-learning*) ha terminado por adoptar al video como su formato de instrucción principal. Plataformas como YouTube y otros portales de video educativo se volvieron centrales para la enseñanza de disciplinas STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), y modificaron la dinámica de transferencia de conocimientos en el entorno universitario post-pandemia [1, 2]. Pese a este despliegue masivo de contenido audiovisual, han quedado expuestas limitaciones importantes en la arquitectura cognitiva humana al momento de procesar información técnica de alta densidad.

Según la *Teoría de la Carga Cognitiva*, el consumo de material educativo en video somete al estudiante al "efecto de la información transitoria" [3]. Frente al soporte textual, donde el lector vuelve sobre cualquier fragmento, el video impone un ritmo lineal que obliga a la memoria de trabajo a retener y procesar estímulos auditivos y visuales de manera efímera. Si el contenido tiene una alta complejidad técnica, se produce una sobrecarga que degrada la retención a largo plazo [4]. Aun cuando los estudiantes recurren a mecanismos de control como la pausa y el retroceso, la falta de una estructura estática dificulta la formación de modelos mentales profundos en temas de ingeniería [5].

Esta problemática se agudiza al considerar la diversidad de perfiles de aprendizaje. Investigaciones recientes muestran que una proporción significativa de estudiantes en carreras técnicas mantiene preferencia por la modalidad de lectura y escritura (*read/write*), o presenta perfiles multimodales que requieren soporte textual para consolidar el aprendizaje [6]. Para estos estudiantes, el video lineal resulta poco eficiente,

ya que demanda un esfuerzo adicional de decodificación que no siempre se traduce en mejor desempeño académico cuando no hay materiales estructurados que lo acompañen [7].

En el plano tecnológico, el uso de sistemas de Reconocimiento Automático de Voz (ASR) ha aparecido como una posible solución, aunque su aplicación en entornos educativos técnicos sigue siendo limitada. Los modelos actuales, incluso arquitecturas avanzadas como Whisper, presentan fallos de transcripción relevantes frente a terminología científica, jerga de ingeniería o discursos colaborativos en el aula [8, 9]. Un error de transcripción en un contexto de ingeniería no es trivial. Puede alterar la lógica de un algoritmo o la precisión de una fórmula, y dejar sin valor pedagógico a la transcripción automática si no se procesa y estructura de forma adecuada [10].

En el contexto universitario, esta brecha resulta evidente. Los estudiantes dependen de grabaciones de clases y tutoriales complejos para materias de distintas áreas, e invierten horas en la transcripción manual y el resumen de estos videos para poder estudiarlos. Esta tarea mecánica reduce el tiempo disponible para el análisis crítico y la práctica experimental. La falta de una herramienta que transforme automáticamente estos recursos audiovisuales en materiales de aprendizaje estructurados se traduce en un obstáculo importante para la optimización del rendimiento académico y la gestión del conocimiento técnico.

A partir de la problemática descrita, esta investigación se orienta por la siguiente pregunta: **¿De qué manera la aplicación de modelos de lenguaje de gran escala y reconocimiento automático de voz, integrados en un sistema de software, permite transformar videos educativos en materiales de aprendizaje estructurados que respondan a las necesidades cognitivas y de formato de los estudiantes universitarios?**

1.2.2 Estado del arte

La transformación automática de contenido audiovisual educativo en materiales de estudio estructurados ha emergido como una línea de investigación activa en los últimos años, impulsada por la convergencia entre los avances en reconocimiento automático de voz y la irrupción de los modelos de lenguaje de gran escala. Los trabajos previos en este campo permiten identificar tanto el grado de madurez tecnológica alcanzado como las brechas que el presente proyecto busca atender.

Uno de los estudios más relevantes en este ámbito es el de Gonzalez y otros, quienes evaluaron el efecto de resúmenes automáticos generados con GPT-3 sobre el aprendizaje a partir de videos de clase [11]. El experimento contó con 106 participantes distribuidos en tres condiciones: video solo, resumen solo y video más resumen automático. La condición combinada obtuvo los mejores resultados en los cuestionarios de evaluación, y los estudiantes con restricciones de tiempo prefirieron los resúmenes antes que el video completo. El estudio aporta evidencia de que la generación automática de resúmenes a partir de transcripciones de video tiene un valor pedagógico medible, y no solo una conveniencia tecnológica.

En la misma línea, Kawamura y Rekimoto desarrollaron FastPerson, un sistema que combina transcripción ASR, análisis visual de fotogramas y generación de resúmenes por segmento mediante LLMs para producir versiones condensadas de videos de clase [12]. Un estudio de usuario con 40 participantes mostró que el sistema redujo el tiempo de visualización en aproximadamente un 53 % respecto a la reproducción tradicional, y mantuvo niveles de comprensión comparables. El trabajo de Kawamura y Rekimoto resulta relevante por sus resultados, pero también por su arquitectura de pipeline. La separación entre la etapa de transcripción, la etapa de segmentación semántica y la etapa de generación de contenido se anticipa al enfoque modular que adopta el presente proyecto.

El trabajo más cercano respecto al tipo de salidas generadas es el de Saha y otros, que describen un *pipeline* de microaprendizaje para cursos de ingeniería informática [13]. El sistema combina Whisper para transcribir el audio de clase con modelos GPT que generan resúmenes, cuestionarios y *flashcards* a partir de la transcripción refinada. Los autores buscan reducir el esfuerzo de autoría docente y evalúan la precisión del contenido y la percepción de los estudiantes. Los tipos de salida coinciden en buena medida con los del presente proyecto, lo que permite una comparación directa. El sistema propuesto en esta tesis añade un glosario técnico y un módulo de auditoría de la generación que no aparecen en la propuesta de Saha.

Stavrinou et al. evaluaron un sistema denominado ReelsEd que segmenta videos de clase largos en microvideos cortos mediante LLMs aplicados sobre la transcripción, con el propósito de mejorar el compromiso del estudiante con el contenido [14]. En un estudio entre sujetos con 62 participantes, los microvideos generados de forma automática produjeron mejoras significativas en engagement, puntajes de cuestionarios y eficiencia en la tarea respecto a los videos completos, sin que aumentara la carga cognitiva percibida. Aunque la salida de ReelsEd es audiovisual y no textual, el estudio aporta evidencia cuantitativa de que la reestructuración automática del contenido de

video mediante LLMs mejora los resultados de aprendizaje de forma estadísticamente significativa.

Desde una mirada histórica, el trabajo de Alrumiah y Al-Shargabi constituye un antecedente relevante de la etapa previa a los LLMs [15]. Su sistema aplicó modelos de asignación latente de Dirichlet (LDA) sobre subtítulos de videos educativos para generar resúmenes extractivos, evaluados mediante juicio humano sobre el dataset EDUVSUM. Los resultados superaron las líneas base de TF-IDF y LSA, aunque la naturaleza extractiva del enfoque, que selecciona fragmentos del texto original en lugar de generar contenido nuevo, limitaba la utilidad pedagógica de los resúmenes producidos. El paso desde métodos extractivos basados en estadísticas de frecuencia hacia generación abstractiva con LLMs es el salto cualitativo que define el estado del arte actual en este campo, y que sostiene las decisiones de diseño del sistema propuesto.

Del análisis de estos antecedentes se desprenden tres observaciones para situar la presente investigación. En primer lugar, la mayoría de los sistemas previos generan uno o dos tipos de material, en general resúmenes o cuestionarios, pero no cubren un conjunto amplio de recursos complementarios. En segundo lugar, los trabajos revisados evalúan sus sistemas mediante métricas automáticas o revisión de expertos, pero no recogen de forma sistemática la percepción de usabilidad de los usuarios finales, que son quienes deciden la adopción real del sistema. Por último, los sistemas existentes suelen construirse como prototipos de investigación, sin una arquitectura pensada para producción, lo que limita su escalabilidad y mantenibilidad en condiciones de uso institucional. El sistema propuesto busca atender las tres brechas en una misma propuesta.

1.3 Fundamentación teórica

1.3.1 El video educativo y la carga cognitiva en el aprendizaje técnico

El aprendizaje basado en video se ha consolidado como el formato dominante en la educación superior técnica, impulsado por la proliferación de plataformas de contenido abierto y el auge del modelo de aula invertida. Esa adopción masiva, sin embargo, no vino acompañada de una reflexión suficiente sobre las limitaciones cognitivas que el formato impone al estudiante. La *Teoría de la Carga Cognitiva* (CLT, por sus siglas en inglés), desarrollada en su origen por Sweller, plantea que la memoria de trabajo humana posee una capacidad de procesamiento simultáneo estrictamente limitada. A diferencia de un texto impreso, donde el lector puede volver sobre cualquier fragmen-

to con libertad, el video impone un flujo temporal continuo. Cada fotograma, palabra y diagrama desaparece del campo perceptivo antes de que el siguiente aparezca, lo que obliga a la memoria de trabajo a mantener activa la representación de segmentos anteriores mientras procesa los nuevos. Este es el llamado *efecto de información transitoria*, que aumenta la carga extrínseca y reduce los recursos disponibles para la comprensión profunda [16]. Cuando el contenido es técnicamente denso, como ocurre con la programación o el diseño de sistemas, el efecto se vuelve crítico, ya que un solo concepto mal retenido puede invalidar la comprensión de todo lo que sigue.

El efecto de atención dividida (*split-attention effect*) constituye una segunda fuente de sobrecarga cognitiva en contextos técnicos. Este efecto ocurre cuando el estudiante debe integrar mentalmente dos o más fuentes de información presentadas en ubicaciones espaciales o temporales separadas, por ejemplo el código en pantalla de un tutorial y la explicación verbal simultánea del instructor. En un estudio con estudiantes universitarios que aprendían mediante videotutoriales de software, se observó que la separación espacial entre el video y la aplicación que el estudiante debía usar en paralelo generó una mayor dilatación pupilar asociada a la tarea, indicador fisiológico de mayor demanda cognitiva, frente a configuraciones que reducían esa separación [17]. Este resultado tiene implicaciones directas para el consumo de videos educativos de ingeniería, donde el estudiante alterna con frecuencia entre el video y un entorno de desarrollo o un cuaderno de notas.

Un hallazgo de especial interés para fundamentar esta propuesta proviene de la comparación entre modalidades. Un estudio controlado con 247 estudiantes universitarios que aprendieron sobre un tema científico complejo a través de texto, video o video con subtítulos encontró rendimientos equivalentes en comprensión inmediata entre texto y video. Además, evidenció que el video subtitulado, al sobrecargar el canal visual con gráficos, narración y texto en simultáneo, perjudicó de forma estadísticamente significativa los resultados de aprendizaje profundo e inferencial [18]. Esta evidencia indica que el problema no es el video en sí, sino la ausencia de una representación textual bien estructurada que el estudiante pueda consultar de forma autónoma, no lineal y a su propio ritmo.

Las estrategias correctivas centradas en el propio medio audiovisual ofrecen un alivio parcial pero no resuelven la transitoriedad estructural del formato. El control de la velocidad de reproducción, por ejemplo, degrada el rendimiento por encima de 1.5× [19], y la segmentación en videos cortos mejora la visualización y el desempeño en exámenes pero deja intacta la condición lineal del recurso [20]. En todos los casos el estudiante sigue obligado a procesar la información en tiempo real, sin posibilidad de

navegar semánticamente por su contenido ni de disponer de un resumen que abstraiga los conceptos esenciales. Esta brecha motiva el desarrollo de un sistema que transforme de forma automática el contenido audiovisual en materiales textuales estructurados, adaptados a las necesidades cognitivas del estudiante de ingeniería.

1.3.2 Estilos de aprendizaje y la demanda de materiales textuales estructurados

El modelo VARK, propuesto por Fleming y Mills, clasifica las preferencias de aprendizaje en cuatro modalidades: visual, auditiva, lectura/escritura (*read/write*) y kinestésica. La evidencia empírica reciente no respalda de forma consistente la idea de que el estilo de aprendizaje prediga el rendimiento académico [21]. El valor operativo del modelo se sitúa en otra dirección. Permite identificar qué formato de material de estudio demanda cada perfil de estudiante para procesar la información con menor esfuerzo.

En contextos universitarios, la modalidad de lectura/escritura no es minoritaria. Un estudio con estudiantes de farmacia en Corea encontró que la preferencia read/write fue la más prevalente dentro de la cohorte analizada, por encima de las modalidades visual, auditiva y kinestésica [6]. Este hallazgo contrasta con la percepción habitual de que los estudiantes de carreras técnicas o de salud son predominantemente kinestésicos o visuales, y sugiere que una fracción considerable del estudiantado universitario procesa mejor la información cuando esta se presenta en forma de texto organizado, listas estructuradas, definiciones y resúmenes escritos.

Desde una mirada más amplia, la distinción entre modalidades unimodales y multimodales también resulta pertinente. La evidencia indica que los estudiantes multimodales, es decir, los que combinan dos o más preferencias, suelen obtener mejores resultados académicos que los unimodales [21]. Por lo tanto, enriquecer el conjunto de materiales de estudio con formatos complementarios beneficia al estudiante con independencia de su perfil dominante. El sistema propuesto en este trabajo se ubica precisamente en ese punto. Produce material textual derivado del video, pensado como apoyo complementario para los estudiantes que ya emplean video como insumo de estudio autónomo, con énfasis en quienes muestran preferencia por la lectura y la escritura. No se concibe como canal universal para perfiles que no usan el video como fuente de estudio, ni como sustituto del propio video, sino como una segunda vía estructurada para quienes lo consumen.

Por último, la preferencia por el formato textual para tareas de retención y concentración trasciende la categorización VARK. Un estudio comparativo con estudiantes

universitarios de medicina, enfermería y farmacia en Japón encontró que, con independencia de las preferencias declaradas, el formato en papel fue claramente preferido sobre el digital para actividades que demandaban memorización y concentración sostenida [22]. Aunque este hallazgo no se circunscribe al contexto de ingeniería, apunta a un principio más general. Cuando el objetivo es la retención profunda de contenido técnico denso, el material textual estructurado, sea impreso o digital, ofrece ventajas que el video, por su condición transitoria, no puede replicar sin mediación tecnológica adicional.

1.3.3 Inteligencia Artificial Generativa y Modelos de Lenguaje de Gran Escala en la generación de contenido educativo

Los Modelos de Lenguaje de Gran Escala (*Large Language Models*, LLMs) son una clase de sistemas de inteligencia artificial entrenados sobre corpus masivos de texto mediante arquitecturas de transformadores con mecanismos de atención. Su capacidad para comprender instrucciones en lenguaje natural y generar texto coherente, estructurado y contextualmente relevante los ha posicionado como base tecnológica de una nueva generación de herramientas educativas. Una revisión sistemática de 118 estudios empíricos sobre LLMs en educación identificó la generación automatizada de contenido, incluyendo resúmenes, preguntas de evaluación y materiales de estudio, como una de las categorías de aplicación más activas, aunque también señaló como limitación transversal del campo la ausencia de marcos de evaluación robustos [23].

La *Inteligencia Artificial Generativa* (GenAI) es el subconjunto de esta tecnología que no se limita a clasificar o predecir, sino que produce contenido original a partir de una entrada dada. En el contexto educativo, esta capacidad generativa se traduce en la posibilidad de tomar una transcripción de clase, un capítulo de libro o el audio de una conferencia, y transformarlo de forma automática en un conjunto de recursos pedagógicos estructurados. Una revisión de 83 artículos sobre el impacto de los LLMs en la educación superior identificó esta generación de contenido adaptativo como una tendencia dominante y subrayó la necesidad de metodologías que evalúen y garanticen la calidad del material producido [24]. Esta observación es directamente relevante para el presente trabajo, en el cual la calidad de los materiales generados se somete a una evaluación de usabilidad con usuarios, complementaria a la revisión del contenido por parte del equipo de desarrollo.

Los cuatro tipos de material que produce el sistema desarrollado en esta investigación, esto es, resúmenes estructurados, *flashcards*, cuestionarios de opción múltiple

y glosarios técnicos, corresponden a los casos de uso documentados en la literatura especializada. Abd-Alrazaq et al. describieron el potencial de los LLMs para generar precisamente estos recursos de forma personalizada, y apuntaron hacia bucles de retroalimentación iterativos que mejoraran la calidad del contenido con el tiempo [25]. Lo que en 2023 se planteaba como propuesta conceptual se ha materializado luego en implementaciones concretas con resultados medibles.

La evaluación sistemática de la calidad de estos recursos generados es un problema abierto que la comunidad científica ha empezado a abordar con metodología formal. Huang et al. desarrollaron un sistema de índices compuesto por cuatro dimensiones y veinte indicadores para evaluar recursos educativos digitales generados por IA, e identificaron la autenticidad del contenido, su precisión factual y su relevancia como los criterios de mayor peso en la valoración global [26]. Este marco conceptual respalda la decisión de diseño tomada en el presente proyecto de incluir la evaluación de usabilidad con usuarios como parte del ciclo de desarrollo, y no como una validación posterior y opcional.

1.3.4 Reconocimiento Automático de Voz y el modelo Whisper en contextos educativos

El Reconocimiento Automático de Voz (*Automatic Speech Recognition*, ASR) es la tecnología que convierte señales de audio en texto transcrito. En cualquier sistema que procese video para producir materiales textuales, el ASR es la primera etapa del pipeline. Sin una transcripción de calidad suficiente, ningún modelo de lenguaje posterior puede generar contenido pedagógico confiable, ya que los errores sistemáticos sobre palabras clave se propagan a todas las etapas siguientes. Dentro de este campo, Whisper, modelo de propósito general publicado por OpenAI en 2022, se ha posicionado como una de las referencias actuales por su tolerancia frente a variaciones de acento, velocidad de habla y calidad de grabación, gracias a la escala de su corpus de entrenamiento [27].

Las condiciones reales del aula universitaria, sin embargo, distan bastante del escenario ideal de grabación. En un estudio que evaluó Google Speech y Whisper sobre discurso colaborativo auténtico en grupos pequeños dentro de una unidad STEM de varios días, ambos sistemas produjeron tasas de error superiores a 0.82, incluso con micrófonos de consumo estándar [8]. El mismo estudio documentó una degradación adicional ante vocabulario especializado de dominio académico o técnico, como términos de ingeniería, siglas, nombres de algoritmos o notación matemática verba-

lizada, que aparece con baja frecuencia en los datos de entrenamiento generales. El ruido ambiental, los solapamientos de voz y la densidad terminológica del contenido técnico configuran así una limitación estructural del ASR en entornos educativos, particularmente relevante en áreas como programación, estructuras de datos o redes de computadoras.

La estrategia de fine-tuning ha demostrado ser una vía efectiva para mitigar estas limitaciones. Southwell et al. adaptaron Whisper a corpus de habla infantil en aulas reales mediante ajuste fino supervisado, y consiguieron mejoras sustanciales respecto al estado del arte previo en condiciones de aula desafiantes [28]. Aunque el dominio de adaptación es distinto, el principio se transfiere. Whisper es un modelo base que mejora de forma sustantiva cuando se especializa en el tipo de audio que debe procesar, lo que sostiene la decisión de incorporarlo como motor de transcripción dentro de un pipeline que combine fuentes alternativas y mecanismos de respaldo según las características de cada video procesado.

1.3.5 Materiales de aprendizaje estructurados: fundamento pedagógico de los outputs del sistema

La decisión sobre qué tipos de materiales genera el sistema propuesto no es arbitraria ni meramente tecnológica. Responde a un cuerpo consolidado de evidencia sobre qué estrategias de estudio producen retención duradera del conocimiento en educación superior. La psicología cognitiva del aprendizaje distingue entre técnicas de estudio pasivas (releer apuntes, subrayar, escuchar grabaciones repetidas) y técnicas activas basadas en la recuperación deliberada de información. La evidencia acumulada durante las últimas dos décadas señala de forma consistente que las segundas superan a las primeras para la retención a largo plazo. Los estudiantes universitarios, sin embargo, tienden a usar más las pasivas porque las perciben como más cómodas [29].

Las *flashcards* con repetición espaciada son el formato de estudio activo con mayor respaldo empírico en educación superior. El principio subyacente es el *efecto de espaciado*: distribuir las exposiciones a un mismo contenido en intervalos crecientes produce una consolidación mnémica claramente superior a la revisión masiva en una sola sesión. Un estudio cuasi-experimental con 799 estudiantes universitarios de enfermería distribuidos en 19 campus encontró que los usuarios de flashcards digitales preparadas por docentes obtuvieron puntajes finales con un tamaño de efecto de $d = 0.42$, y resultaron aproximadamente tres veces más propensos a aprobar y dos veces más propensos a obtener calificaciones sobresalientes que sus pares no usuarios,

tras controlar covariables [30]. En contextos de mayor densidad conceptual, como la pediatría clínica, el efecto escala todavía más: un diseño cuasi-experimental con estudiantes de medicina reportó un tamaño de efecto de $d \approx 0.8$ a favor del grupo con flashcards frente al estudio tradicional [31].

Los cuestionarios de opción múltiple operan bajo un principio relacionado, denominado *efecto de prueba* o *retrieval practice*. El acto mismo de intentar recuperar información de la memoria, aunque sea con apoyo de alternativas de respuesta, fortalece la huella mnémica de manera más efectiva que releer el material fuente. Una revisión de la evidencia en entornos de aula señala que la práctica de recuperación mediante MCQs supera de forma consistente al reestudio y a la inactividad para la retención a largo plazo, aunque su ventaja sobre otras técnicas activas como el mapeo conceptual es menos uniforme [32]. En el contexto de nueve cursos introductorios de STEM, la práctica espaciada mediante quizzes produjo un beneficio estadísticamente significativo sobre el rendimiento al final del curso, aunque con variabilidad entre disciplinas que subraya la importancia del diseño específico de las preguntas [33]. A mayor escala, un análisis longitudinal de 4.080 preguntas de práctica distribuidas a lo largo de dos años en un programa biomédico demostró que los ítems de recuperación reutilizados preservaron el conocimiento de forma medible frente a preguntas nuevas, y aportó evidencia de que la exposición repetida y espaciada mediante MCQs tiene efectos acumulativos sostenidos [34].

Los resúmenes estructurados y los glosarios técnicos cumplen una función habilitadora dentro de este conjunto. Su valor no reside tanto en su uso aislado como en su papel de andamiaje cognitivo. Aportan al estudiante una representación compacta y navegable del contenido que facilita las sesiones posteriores de recuperación activa mediante flashcards y cuestionarios. Revisiones de métodos de estudio en educación superior identifican la combinación de resúmenes con práctica de recuperación como una de las estrategias de mayor retorno para el estudio técnico, ya que el resumen reduce la carga extrínseca en la sesión de repaso al eliminar la necesidad de revisar el material fuente completo [29]. Un glosario técnico, en este sentido, no es un apéndice decorativo, sino un recurso de consulta rápida que ancla la terminología especializada antes de que el estudiante enfrente los mecanismos de recuperación activa.

El conjunto de estos cuatro formatos, es decir, resumen estructurado, flashcards, cuestionario de opción múltiple y glosario técnico, cubre distintos momentos y modalidades del ciclo de estudio: la comprensión inicial mediante el resumen, la consolidación terminológica mediante el glosario, y la retención a largo plazo mediante la recuperación activa con flashcards y MCQs. Esta correspondencia entre los outputs del

sistema y los mecanismos cognitivos validados por la investigación educativa es lo que distingue la propuesta de una simple herramienta de transcripción automática.

1.3.6 Arquitecturas de software para sistemas de procesamiento asíncrono

Un sistema que procesa video, ejecuta modelos de reconocimiento de voz y orquesta múltiples llamadas a modelos de lenguaje de gran escala plantea exigencias arquitectónicas distintas a las de una API convencional. Dos principios documentados en la literatura sobre sistemas modernos sirven de fundamento para abordarlas: la descomposición modular por capacidades funcionales y el procesamiento asíncrono mediante colas de trabajos. El primer principio sostiene que organizar el código en módulos de alta cohesión interna y bajo acoplamiento entre sí, con cada módulo responsable de una capacidad concreta del sistema, mejora la mantenibilidad y la evolución independiente de las partes. Un estudio sobre arquitecturas orientadas a servicios con 106 profesionales de la industria identificó la descomposición por capacidades de negocio como el criterio más utilizado en sistemas contemporáneos, y lo asoció a una mejor mantenibilidad cuando se acompaña de patrones de diseño que desacoplan las decisiones tecnológicas de la lógica funcional [35].

El segundo principio responde al carácter de larga duración de las operaciones involucradas. Una transcripción ASR o una generación con un modelo de lenguaje pueden tomar entre segundos y varios minutos. Atenderlas de forma sincrónica, bloqueando la conexión HTTP hasta que el procesamiento termine, derivaría en timeouts y en un uso ineficiente de los recursos del servidor. El patrón establecido para este escenario es el procesamiento asíncrono con colas de mensajes: la API recibe la solicitud, encola un trabajo y retorna de inmediato un identificador de seguimiento, mientras un *worker* independiente consume la cola y ejecuta el procesamiento. La literatura sobre optimización de sistemas backend identifica este mecanismo como uno de los de mayor impacto sobre el *throughput* en sistemas con operaciones de larga duración, ya que evita el bloqueo de recursos durante la espera de resultados externos y permite escalar el componente que ejecuta el procesamiento de forma independiente del que recibe las solicitudes [36].

1.3.7 Evaluación de usabilidad mediante System Usability Scale

La evaluación de un sistema educativo no se agota en su corrección técnica. Una vez que el flujo funciona y los materiales generados son válidos, queda por compro-

bar si los usuarios finales perciben el sistema como usable, pues de esta percepción depende la adopción real dentro de la práctica académica. El *System Usability Scale* (SUS) es una escala estandarizada propuesta por John Brooke en 1996 y consolidada después como una de las más utilizadas en investigación de usabilidad e interacción persona-computador [37, 38]. Se compone de diez afirmaciones sobre la experiencia de uso, que alternan formulaciones positivas y negativas, y que el participante valora en una escala Likert de cinco puntos. Las respuestas se combinan mediante una fórmula estándar que produce un puntaje global en el rango de 0 a 100, lo que permite comparar resultados entre distintos sistemas o poblaciones sin ajustes de normalización adicionales.

Para interpretar los puntajes obtenidos se utiliza la escala de referencia propuesta por Bangor, Kortum y Miller, que asocia rangos de puntaje con una escala adjetiva cualitativa y establece en 68 puntos el umbral por debajo del cual la usabilidad percibida del sistema se considera cuestionable [39]. Esta interpretación adjetiva resulta especialmente útil cuando el número de participantes es reducido, ya que facilita la lectura del resultado sin necesidad de contrastes estadísticos adicionales.

1.4 Objetivos

1.4.1 Objetivo general

Transformar videos educativos en materiales de aprendizaje estructurados utilizando modelos de lenguaje de gran escala y reconocimiento de voz.

1.4.2 Objetivos específicos

- Determinar los requisitos funcionales de la transformación de videos educativos en materiales de aprendizaje estructurados utilizando inteligencia artificial generativa.
- Implementar un sistema de transformación automatizada que convierta videos educativos en materiales de aprendizaje estructurados mediante modelos de lenguaje de gran escala y reconocimiento de voz.
- Evaluar el nivel de aceptación de la solución propuesta en estudiantes con preferencia de aprendizaje basada en lectura escritura.

CAPÍTULO II

METODOLOGÍA

Este capítulo describe la metodología que articula el trabajo en tres bloques complementarios. El primero corresponde a la metodología de validación de la pertinencia de la propuesta, que combina una encuesta a estudiantes y una revisión bibliográfica como insumos para sustentar la necesidad del sistema. El segundo describe la metodología de implementación del software, con la selección y aplicación del marco de desarrollo. El tercero presenta la metodología de evaluación de la propuesta, con el diseño del instrumento que mide la aceptación de la solución entre el público objetivo. La presente investigación adopta el tipo de investigación **aplicada**, en concordancia con el objetivo de desarrollar y validar un sistema de transformación de videos educativos en materiales de aprendizaje estructurados mediante inteligencia artificial generativa.

2.1 Materiales

Los materiales de investigación son los instrumentos mediante los cuales se recolecta, sistematiza y valida la información necesaria para sustentar el desarrollo del sistema y justificar su pertinencia. En este trabajo se emplearon dos instrumentos complementarios: una encuesta estructurada dirigida a estudiantes y un conjunto de fichas bibliográficas para el registro de las fuentes documentales.

La **encuesta** permite recoger información cuantitativa sobre los hábitos, dificultades y experiencias de los estudiantes universitarios y dimensionar con respaldo numérico la brecha que el sistema busca atender. El instrumento se organizó en doce preguntas distribuidas en tres dimensiones: hábitos de uso de videos educativos (preguntas 1 a 4), dificultades percibidas al estudiar sin material de apoyo complementario (preguntas 5 a 8) y experiencia previa con herramientas de inteligencia artificial aplicadas al aprendizaje (preguntas 9 a 12). Se administró en formato digital mediante Google Forms, lo que facilitó la distribución y la exportación ordenada de los resultados.

Las **fichas bibliográficas** cumplen la función de registro sistemático de las fuentes consultadas durante la revisión de literatura. Cada ficha reúne los datos de identificación de la fuente, un resumen del aporte principal y la relación explícita con los

objetivos de esta investigación. Con eso se garantiza la trazabilidad del proceso documental.

2.2 Métodos

2.2.1 Modalidad de la investigación

Este trabajo adopta la **modalidad de investigación aplicada**. Su propósito central no es producir conocimiento teórico nuevo, sino resolver un problema concreto y documentado: la dificultad que enfrentan los estudiantes para procesar y retener contenido educativo presentado únicamente en formato video. El conocimiento científico existente sobre procesamiento de lenguaje natural, reconocimiento automático de voz y grandes modelos de lenguaje se aplica directamente en la construcción de un artefacto de software funcional, evaluable y reproducible.

La modalidad aplicada se complementa con la **modalidad bibliográfica-documental**, que sostiene la fase de revisión de literatura y el levantamiento del marco teórico. Esta modalidad permite vincular los desarrollos propios con el estado del arte internacional y construir un marco conceptual coherente sobre el cual se toman las decisiones de diseño del sistema. Para ello se consultaron bases de datos especializadas como IEEE Xplore, ACM Digital Library y arXiv, con prioridad en publicaciones de los últimos diez años, y se elaboraron fichas bibliográficas como instrumento de registro.

La validación de la pertinencia de la propuesta se sostiene en dos planos complementarios: una **encuesta** aplicada a estudiantes universitarios, que produce datos cuantitativos sobre la frecuencia del problema, las dificultades más comunes y el nivel de familiaridad con herramientas de inteligencia artificial; y una **revisión documental** de la literatura científica especializada, registrada mediante fichas bibliográficas, que aporta el sustento conceptual para las decisiones del sistema. La metodología de desarrollo del software se trata como bloque metodológico independiente.

2.2.2 Convocatoria y muestra

La encuesta se aplicó mediante **muestreo no probabilístico por conveniencia**. La elección de este tipo de muestreo responde a la naturaleza del diagnóstico: el objetivo es caracterizar hábitos y dificultades en grupos universitarios accesibles al investigador,

no realizar inferencias poblacionales. La convocatoria se dirigió a estudiantes universitarios de dos ámbitos en los que el sistema podía evaluarse con condiciones reales de uso: el Programa Regular de Inglés del Centro de Idiomas y la carrera de Software de la FISEI, ambos de la Universidad Técnica de Ambato. La Tabla 1 muestra la matrícula activa de los grupos convocados durante el periodo lectivo en que se realizó el estudio.

Tabla 1: Grupos convocados a la encuesta

Grupo	**Estudiantes matriculados**
Programa Regular de Inglés (cursos a cargo de Mg. Wilma Villacís)	65
Carrera de Software (estudiantes que cursan o han cursado Programación Orientada a Objetos)	86
Población combinada	**151**

El formulario de Google Forms se compartió a los estudiantes a través de los docentes de los grupos convocados. Durante el período de recolección respondieron **129 estudiantes**, que constituyen la muestra efectiva del estudio ($n = 129$).

2.2.3 Recolección de información

a. Encuesta

La encuesta tiene como propósito cuantificar la prevalencia de las dificultades asociadas al uso de videos educativos y caracterizar el perfil de uso de herramientas de inteligencia artificial en el estudiantado universitario. El instrumento consta de doce preguntas distribuidas en tres dimensiones: hábitos de uso de videos educativos (P1–P4), dificultades percibidas al estudiar sin material de apoyo (P5–P8) y experiencia previa con herramientas de IA aplicadas al aprendizaje (P9–P12). Las preguntas con valoración de frecuencia emplean una escala Likert de cinco puntos (Nunca, Raramente, A veces, Frecuentemente, Siempre); el resto utiliza opción única o múltiple según corresponda al ítem. La Tabla 2 resume el enunciado de cada pregunta. Se administró en formato digital mediante Google Forms, con un período de recolección de dos semanas.

Tabla 2: Ítems del cuestionario aplicado a los estudiantes

ID	Pregunta
P1	¿Con qué frecuencia utilizas videos de YouTube u otras plataformas para estudiar los temas de tu clase?
P2	¿En qué dispositivo ves principalmente los videos educativos para estudiar?
P3	¿Qué tipo de videos utilizas con más frecuencia para estudiar?
P4	¿Cuánto tiempo en promedio dedicas a ver videos educativos fuera del horario de clase?
P5	Cuando terminas de ver un video educativo, ¿con qué frecuencia tienes un material de resumen o síntesis disponible?
P6	¿Cuál de las siguientes dificultades experimentas con más frecuencia al estudiar con videos?
P7	¿Qué tan fácil te resulta identificar las ideas principales de un video educativo sin ningún material de apoyo?
P8	¿Con qué frecuencia necesitas volver a ver un video varias veces porque no lograste retener la información suficiente la primera vez?
P9	¿Has usado alguna herramienta de inteligencia artificial (ChatGPT, Gemini u otras) para ayudarte con el estudio?
P10	¿Para qué has usado principalmente la inteligencia artificial en el contexto del aprendizaje?
P11	¿Qué tan satisfecho has quedado con los resultados que te ha dado una herramienta de IA cuando la usaste para estudiar?
P12	¿Estarías dispuesto a utilizar una herramienta que, a partir de un video, genere automáticamente un resumen, glosario y preguntas de comprensión?

Tabulación de resultados

Durante el período de recolección, **129 estudiantes** completaron el formulario. La tabulación que se presenta a continuación se realizó sobre estos 129 formularios válidos. Para cada pregunta se incluye la tabla de frecuencias y porcentajes correspondiente, junto con un breve análisis interpretativo.

Pregunta 1 (ver Tabla 2, Tabla 3).

Tabla 3: Tabulación pregunta 1 — Frecuencia de uso de videos

Opción	**Frecuencia**	**Porcentaje**
Nunca	4	3,10 %
Raramente	12	9,30 %
A veces	56	43,41 %
Frecuentemente	41	31,78 %
Siempre	16	12,40 %
Total	**129**	**100 %**

Análisis: El 44,18 % del grupo usa videos frecuentemente o siempre como recurso de estudio, y otro 43,41 % lo hace de forma ocasional. En conjunto, el 87,59 % recurre al video con algún grado de regularidad, lo que confirma que este formato ya forma parte del ecosistema de estudio del grupo encuestado. Solo el 12,40 % lo utiliza raramente o nunca, una cifra que indica que el acceso a este recurso no representa una barrera importante en la población.

Pregunta 2 (ver Tabla 2, Tabla 4).

Tabla 4: Tabulación pregunta 2 — Dispositivo de acceso

Opción	**Frecuencia**	**Porcentaje**
Teléfono celular	68	52,71 %
Computadora	49	37,98 %
Tablet	8	6,20 %
Otro	4	3,10 %
Total	**129**	**100 %**

Análisis: Más de la mitad del grupo (52,71 %) accede a los videos desde el teléfono celular. Si a esto se suma el 6,20 % que usa tablet, los dispositivos móviles se confirman como el canal principal de consumo. Este dato tiene implicaciones directas sobre el diseño del frontend del sistema. Los materiales generados deben ser legibles y navegables en pantallas pequeñas sin perder claridad estructural.

Pregunta 3 (ver Tabla 2, Tabla 5).

Tabla 5: Tabulación pregunta 3 — Tipo de video utilizado

Opción	**Frecuencia**	**Porcentaje**
Tutoriales de gramática o vocabulario	57	44,19 %
Películas o series en inglés	29	22,48 %
Clases grabadas del Centro de Idiomas	28	21,71 %
Conferencias o charlas (TED, etc.)	15	11,63 %
Total	**129**	**100 %**

Análisis: Los tutoriales de gramática o vocabulario encabezan la lista con el 44,19 %, seguidos de un bloque casi equivalente entre películas o series (22,48 %) y clases grabadas del Centro de Idiomas (21,71 %). Esta distribución muestra un perfil mixto de consumo, con predominio del contenido didáctico estructurado. Los formatos principales son compatibles con el sistema propuesto, ya que contienen un discurso orientado a objetivos de aprendizaje concretos que se presta a la generación de resúmenes y glosarios.

Pregunta 4 (ver Tabla 2, Tabla 6).

Tabla 6: Tabulación pregunta 4 — Tiempo semanal dedicado a videos

Opción	**Frecuencia**	**Porcentaje**
Menos de 30 minutos por semana	46	35,66 %
Entre 30 y 60 minutos	45	34,88 %
Entre 1 y 2 horas	28	21,71 %
Más de 2 horas	10	7,75 %
Total	**129**	**100 %**

Análisis: El 70,54 % del grupo dedica hasta una hora semanal al consumo de videos educativos. Es un tiempo moderado que refleja un uso del video como recurso de consulta puntual, no como estrategia de estudio principal. Este patrón refuerza la necesidad de aprovechar al máximo cada visualización mediante materiales de síntesis que permitan recuperar la información sin volver a invertir el tiempo completo del video.

Pregunta 5 (ver Tabla 2, Tabla 7).

Tabla 7: Tabulación pregunta 5 — Disponibilidad de material de síntesis

Opción	**Frecuencia**	**Porcentaje**
Nunca	37	28,68 %
Pocas veces	64	49,61 %
Frecuentemente	21	16,28 %
Siempre	7	5,43 %
Total	**129**	**100 %**

Análisis: El 78,29 % de los estudiantes nunca o pocas veces dispone de un material de síntesis después de ver un video. La cifra es elocuente y constituye la evidencia cuantitativa más directa de la brecha que el sistema propuesto busca cerrar. Tres de cada cuatro estudiantes se quedan sin apoyo estructurado una vez que el video termina.

Pregunta 6 (ver Tabla 2, Tabla 8).

Tabla 8: Tabulación pregunta 6 — Principal dificultad al estudiar con videos

Opción	**Frecuencia**	**Porcentaje**
La velocidad del habla es muy rápida	53	41,09 %
No entiendo suficiente vocabulario	41	31,78 %
No sé qué información es importante y cuál no	24	18,60 %
El video no tiene subtítulos o los subtítulos fallan	10	7,75 %
Sin respuesta	1	0,78 %
Total	**129**	**100 %**

Análisis: La velocidad del habla es la dificultad más reportada, con el 41,09 %, seguida del vocabulario desconocido con el 31,78 %. Ambas dificultades son atendidas de forma directa por el sistema. La transcripción automática convierte el audio en texto, lo que permite leerlo al ritmo del propio estudiante, y el glosario técnico generado cubre el vocabulario desconocido identificado en el discurso.

Pregunta 7 (ver Tabla 2, Tabla 9).

Tabla 9: Tabulación pregunta 7 — Facilidad para identificar ideas principales

Opción	**Frecuencia**	**Porcentaje**
Muy difícil	10	7,75 %
Difícil	27	20,93 %
Regular	66	51,16 %
Fácil	20	15,50 %
Muy fácil	6	4,65 %
Total	**129**	**100 %**

Análisis: Solo el 20,15 % de los encuestados considera fácil o muy fácil identificar las ideas principales de un video sin material de apoyo. La mayoría (51,16 %) se ubica en un punto intermedio y un 28,68 % reconoce abiertamente dificultad o mucha dificultad. Leído en conjunto, casi ocho de cada diez estudiantes no procesan el video con comodidad cuando están solos frente a él, lo que respalda la necesidad del resumen estructurado como salida primaria del sistema.

Pregunta 8 (ver Tabla 2, Tabla 10).

Tabla 10: Tabulación pregunta 8 — Frecuencia de revisión repetida del video

Opción	**Frecuencia**	**Porcentaje**
Nunca	11	8,53 %
Raramente	15	11,63 %
A veces	55	42,64 %
Frecuentemente	40	31,01 %
Siempre	8	6,20 %
Total	**129**	**100 %**

Análisis: El 37,21 % de los estudiantes revisa el video frecuentemente o siempre porque no retuvo la información en la primera visualización, y otro 42,64 % lo hace a veces. Ese 79,85 % que repite la visualización al menos de forma ocasional representa una ineficiencia concreta en el proceso de estudio, ya que multiplica el tiempo invertido sin garantizar mejor comprensión. El sistema busca reducir esa ineficiencia mediante materiales de síntesis que eliminen la necesidad de repetir la visualización completa.

Pregunta 9 (ver Tabla 2, Tabla 11).

Tabla 11: Tabulación pregunta 9 — Uso previo de IA en el aprendizaje

Opción	Frecuencia	Porcentaje
Sí, frecuentemente	41	31,78 %
Sí, alguna vez	60	46,51 %
No, pero me gustaría probar	17	13,18 %
No, y no me interesa	10	7,75 %
Sin respuesta	1	0,78 %
Total	**129**	**100 %**

Análisis: El 78,29 % de los estudiantes ya ha utilizado alguna herramienta de IA para estudiar, un nivel de adopción alto que muestra que la tecnología no les resulta ajena. Si se añade el 13,18 % que no la ha usado pero está dispuesto a probarla, la aceptación potencial del sistema llega al 91,47 %. La barrera de adopción tecnológica, por lo tanto, no debería ser un obstáculo importante para la implantación del sistema propuesto.

Pregunta 10 (ver Tabla 2, Tabla 12).

Tabla 12: Tabulación pregunta 10 — Uso principal de IA en el aprendizaje

Opción	Frecuencia	Porcentaje
No he utilizado IA	41	31,78 %
Resolver dudas de gramática o vocabulario	38	29,46 %
Traducir textos o frases	37	28,68 %
Generar resúmenes de textos	13	10,08 %
Total	**129**	**100 %**

Análisis: El **31,78 %** no ha utilizado IA en el aprendizaje, dato que evidencia una brecha de adopción más allá del aprendizaje de inglés. Entre quienes sí la usan predomina resolver dudas (29,46 %) y traducir (28,68 %); solo el 10,08 % la ha usado para generar resúmenes y ninguno toma el video como fuente.

Pregunta 11 (ver Tabla 2, Tabla 13).

Tabla 13: Tabulación pregunta 11 — Satisfacción con herramientas de IA actuales

Opción	**Frecuencia**	**Porcentaje**
Muy insatisfecho	5	3,88 %
Insatisfecho	9	6,98 %
Neutral	54	41,86 %
Satisfecho	42	32,56 %
Muy satisfecho	19	14,73 %
Total	**129**	**100 %**

Análisis: El 47,29 % de los estudiantes reporta estar satisfecho o muy satisfecho con las herramientas de IA existentes, pero un 41,86 % se mantiene neutral y un 10,86 % expresa insatisfacción. La alta proporción de respuestas neutrales es reveladora. Indica que la experiencia actual con IA educativa es tibia, ni claramente buena ni mala. Es un espacio intermedio donde una herramienta especializada en la transformación de videos puede aportar valor diferencial.

Pregunta 12 (ver Tabla 2, Tabla 14).

Tabla 14: Tabulación pregunta 12 — Disposición de uso del sistema propuesto

Opción	**Frecuencia**	**Porcentaje**
Definitivamente sí	61	47,29 %
Probablemente sí	38	29,46 %
No estoy seguro	25	19,38 %
Probablemente no	2	1,55 %
Definitivamente no	3	2,33 %
Total	**129**	**100 %**

Análisis: El 76,75 % de los estudiantes encuestados, casi tres de cada cuatro, manifiesta disposición positiva a utilizar el sistema propuesto, con el 47,29 % en la respuesta más contundente (“definitivamente sí”). Solo el 3,88 % se posiciona en contra. Este nivel de aceptación anticipada, junto con el 19,38 % de indecisos que podrían convertirse en usuarios tras una primera experiencia, respalda con claridad la pertinencia del desarrollo y sugiere una alta probabilidad de adopción real entre el público objetivo.

b. Fichas bibliográficas

Como instrumento de la modalidad documental, se elaboraron fichas bibliográficas para sistematizar las fuentes más relevantes consultadas durante la construcción del marco teórico. Las Tablas 15 a 19 reúnen las cinco fichas representativas de las fuentes fundamentales de esta investigación.

Tabla 15: Ficha bibliográfica 1 — LLMs en educación: revisión sistemática

Campo	**Información**
Autor(es)	Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D.
Título	Practical and ethical challenges of large language models in education: A systematic scoping review
Año	2024
Fuente	British Journal of Educational Technology, 55(1), 90–112
DOI	10.1111/bjet.13370
Resumen del aporte	Revisión sistemática de 118 estudios empíricos sobre el uso de modelos de lenguaje de gran escala en educación. Identifica la generación automatizada de contenido, que abarca resúmenes, preguntas de evaluación y materiales de estudio, como una de las categorías de aplicación más activas, y señala la ausencia de marcos de evaluación robustos como limitación transversal del campo.
Relación con la tesis	Sustenta la pertinencia del sistema dentro de una línea de investigación consolidada y respalda la decisión metodológica de incorporar una evaluación de usabilidad con usuarios finales como respuesta a la brecha de evaluación señalada por la literatura.

Tabla 16: Ficha bibliográfica 2 — Transcripción automática de voz

Campo	**Información**
Autor(es)	Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I.

Continúa en la página siguiente

(Continuación Tabla 16)

Campo	Información
Título	Robust Speech Recognition via Large-Scale Weak Supervision
Año	2023
Fuente	Proceedings of the 40th International Conference on Machine Learning (ICML)
DOI	10.48550/arXiv.2212.04356
Resumen del aporte	Describe el modelo Whisper, un sistema de reconocimiento automático de voz entrenado con 680 000 horas de audio supervisado débilmente sobre múltiples idiomas y dominios. Presenta la arquitectura encoder-decoder del modelo y sus distintas variantes de tamaño (tiny, base, small, medium, large), lo que permite seleccionar la configuración adecuada según los recursos de cómputo disponibles en cada entorno de despliegue.
Relación con la tesis	Sustenta la elección de *nodejs-whisper*, binding de Node.js sobre la implementación *whisper.cpp* del modelo Whisper, como motor de transcripción embebido en el worker del sistema. Permite ejecutar la transcripción localmente sin depender de servicios externos de pago y con control sobre la variante del modelo empleada.

Tabla 17: Ficha bibliográfica 3 — Carga cognitiva en el video instruccional

Campo	Información
Autor(es)	Yu, Q., Gou, J., Li, Y., Pi, Z., & Yang, J.
Título	Introducing support for learner control: Temporal and organizational cues in instructional videos
Año	2024
Fuente	British Journal of Educational Technology, 55(3), 933–956
DOI	10.1111/bjet.13408

Continúa en la página siguiente

(Continuación Tabla 17)

Campo	Información
Resumen del aporte	Estudio sobre la carga cognitiva en el video instruccional, con énfasis en el efecto de información transitoria y en las limitaciones de la memoria de trabajo cuando el contenido posee alta densidad técnica. Evalúa cómo las claves temporales y organizacionales reducen la carga extrínseca asociada al medio audiovisual.
Relación con la tesis	Constituye el fundamento teórico central del problema de investigación; documenta la sobrecarga cognitiva inherente al video y justifica la necesidad de transformar el contenido audiovisual en materiales textuales estructurados que el estudiante pueda consultar de forma no lineal.

Tabla 18: Ficha bibliográfica 4 — Interpretación de puntajes del System Usability Scale

Campo	Información
Autor(es)	Bangor, A., Kortum, P., & Miller, J.
Título	Determining what individual SUS scores mean: adding an adjective rating scale
Año	2009
Fuente	Journal of Usability Studies, 4(3), 114–123
DOI / URL	`https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/`
Resumen del aporte	Propone una escala adjetiva cualitativa para interpretar los puntajes individuales del System Usability Scale y establece el umbral de 68 puntos por debajo del cual la usabilidad percibida del sistema se considera cuestionable. Permite la lectura del resultado sin contrastes estadísticos adicionales, lo que resulta especialmente útil cuando el número de participantes es reducido.

Continúa en la página siguiente

(Continuación Tabla 18)

Campo	Información
Relación con la tesis	Es la fuente directa que respalda el criterio de interpretación del puntaje SUS aplicado en la evaluación de usabilidad del sistema con el subgrupo de estudiantes con preferencia lecto-escritora del piloto. Sustenta la lectura cualitativa del puntaje obtenido frente al umbral de aceptabilidad de 68 puntos.

Tabla 19: Ficha bibliográfica 5 — LLMs y generación de materiales de aprendizaje

Campo	Información
Autor(es)	Abd-Alrazaq, A., AlSaad, R., Alhuwail, D., Ahmed, A., Healy, P. M., Latifi, S., Aziz, S., Damseh, R., Alabed Alrazak, S., & Sheikh, J.
Título	Large Language Models in Medical Education: Opportunities, Challenges, and Future Directions
Año	2023
Fuente	JMIR Medical Education, 9, e48291
DOI	10.2196/48291
Resumen del aporte	Describe el potencial de los modelos de lenguaje de gran escala para generar resúmenes, cuestionarios y *flashcards* personalizadas en contextos de educación superior, apuntando hacia bucles de retroalimentación iterativos que mejoren la calidad del contenido producido a lo largo del tiempo.
Relación con la tesis	Es la fuente que respalda directamente la elección de los cuatro tipos de materiales generados por el sistema: resumen estructurado, glosario técnico, *flashcards* y cuestionario de opción múltiple. Confirma que estos cuatro *outputs* corresponden a los casos de uso documentados en la literatura especializada sobre LLMs aplicados a la generación de contenido educativo.

Triangulación de instrumentos

La triangulación entre encuesta y revisión documental permite sostener que existe una necesidad cuantificable y técnicamente abordable. Los estudiantes universitarios

que respondieron la encuesta no disponen de materiales de síntesis estructurados tras la visualización de videos educativos (78,29 % en P5), reportan dificultad para identificar ideas principales sin apoyo (79,84 % en P7) y necesitan revisar el video varias veces para retener la información (79,85 % en P8). La literatura registrada en las fichas bibliográficas respalda la pertinencia de una solución basada en inteligencia artificial generativa y aporta el criterio metodológico para su evaluación posterior.

2.3 Metodología de implementación del software

La construcción del sistema se rige por una metodología de desarrollo seleccionada de forma justificada y documentada. El propósito de este bloque es sustentar la elección del marco que ordena la planificación, las entregas y el control de cambios.

2.3.1 Selección de la metodología

Se evaluaron cuatro alternativas frecuentes para proyectos de tesis con un único desarrollador y requisitos que evolucionan durante la ejecución: **Scrum**, **Scrumban**, **Kanban** y **XP** (Extreme Programming). El descarte preliminar se realizó con una matriz de Simón sobre criterios cualitativos (Tabla 20). Las finalistas pasaron a una matriz ponderada con criterios y pesos (Tabla 21).

Tabla 20: Matriz de Simón — descarte cualitativo de metodologías

Criterio	**Scrum (base)**	**Scrumban**	**Kanban**	**XP**
Disciplina de entregas iterativas	0	0	-	0
Flexibilidad ante cambios	0	+	+	+
Adaptabilidad a equipo de un solo desarrollador	0	+	+	-
Visibilidad del flujo de trabajo	0	+	+	0
Documentación generada naturalmente	0	0	-	-
Ajuste a proyectos con investigación asociada	0	+	0	0
Curva de aprendizaje para el autor	0	0	0	-
Balance (+ netos)	0	**+4**	+2	**-2**

XP queda descartado por balance neto negativo. Kanban puro pierde frente a Scrumban en disciplina de entregas iterativas, condición exigida por el formato de tesis. Las finalistas son Scrum y Scrumban.

Tabla 21: Matriz ponderada — selección final entre Scrum y Scrumban

Criterio	**Peso (%)**	**Scrum**		**Scrumban**	
		Calif. (1-5)	Pond.	Calif. (1-5)	Pond.
Flexibilidad ante cambios	20	3	0,60	5	1,00
Adaptabilidad a equipo de un solo desarrollador	20	3	0,60	5	1,00
Disciplina de entregas iterativas formales	18	5	0,90	4	0,72
Visibilidad del flujo de trabajo	14	4	0,56	5	0,70
Documentación para tesis generada naturalmente	12	4	0,48	4	0,48
Ajuste a proyectos con componente investigativo	10	3	0,30	4	0,40
Curva de aprendizaje del marco	6	4	0,24	4	0,24
Total	**100**		**3,68**		**4,54**

Scrumban obtiene 4,54 frente a 3,68 de Scrum. La diferencia se sostiene por la flexibilidad ante cambios y la adaptabilidad a equipos reducidos. Se adopta Scrumban como metodología del proyecto: se conservan los Sprints formales como unidad de planificación y entrega, y se gestiona el trabajo diario sobre un tablero Kanban.

2.3.2 Roles aplicados al proyecto

Se diferencian dos contextos de aplicación de roles. Para la **implementación de la propuesta** (objeto de este trabajo de titulación), el PhD. Félix Oscar Fernández Peña asume el rol de **Product Owner**, y el autor asume los roles combinados de **Scrum Master y Programador**. Un equipo de una sola persona no constituye Scrumban estricto; la adopción del marco se justifica por su función ordenadora del trabajo y por la trazabilidad documental que produce, no por una composición canónica del equipo. Para la **integración del módulo en NousAI**, realizada por el equipo de plataforma, el rol de Scrum Master del equipo de integración recae en Pablo Villacrés, estudiante de la carrera de Software de la FISEI.

2.4 Metodología de evaluación de la propuesta

La evaluación del sistema se realizó en dos fases. La primera midió la usabilidad de la solución con el *System Usability Scale* (SUS), aplicado a los estudiantes después de una sesión de uso. La segunda midió la validez del contenido generado por el sistema con un cuestionario en escala Likert por cada tipo de bloque producido, y se aplicó

en un curso distinto al de la primera fase para contar con una mirada externa sobre el material.

2.4.1 Población y muestra

La evaluación se aplicó en dos grupos universitarios de la Universidad Técnica de Ambato a los que el investigador tuvo acceso. La Tabla 22 muestra la matrícula activa de los cursos involucrados.

Tabla 22: Grupos universitarios de la evaluación

Grupo	**Estudiantes matriculados**
Programa Regular de Inglés (cursos a cargo de Mg. Wilma Villacís)	65
Carrera de Software (estudiantes que cursan o han cursado Programación Orientada a Objetos)	86
Población combinada	**151**

En ambos casos se trabajó con **muestreo no probabilístico por conveniencia**. Los cursos se seleccionaron según tres criterios: la disponibilidad de los docentes colaboradores, la coincidencia del periodo lectivo con la fase de evaluación y la generación de videos por parte de los profesores sobre los temarios de inglés y de Programación Orientada a Objetos (POO).

2.4.2 Técnicas e instrumentos

La técnica de recolección de datos fue la **encuesta digital**. Se aplicaron dos encuestas distintas, una por cada fase de la evaluación: una encuesta de usabilidad y una encuesta de validez del contenido generado por el sistema. A continuación se describen los dos instrumentos correspondientes.

Cuestionario VARK y SUS extendido

El primer instrumento reúne dos cuestionarios dentro de un mismo formulario. El primero es el **cuestionario VARK** de Fleming y Mills, que clasifica las preferencias de aprendizaje en cuatro modalidades (visual, auditiva, lectura/escritura y kinestésica).

Cuando el estudiante obtiene puntajes dominantes en dos o más modalidades, el cuestionario reporta una categoría agregada llamada *Multimodal*. El segundo es el **System Usability Scale (SUS)** de Brooke [37], una escala estandarizada de diez ítems en formato Likert (1–5) que se ha consolidado como una de las herramientas más usadas para medir usabilidad percibida [38]. El SUS produce un puntaje global de 0 a 100, y la literatura toma 68 puntos como umbral interpretativo de aceptabilidad [39].

A la encuesta SUS se le agregó un bloque demográfico al inicio, dos ítems de cierre con la modalidad VARK declarada por el estudiante y un campo abierto de retroalimentación. El enunciado completo de cada ítem se reporta en el Anexo D.

- **Bloque demográfico** (caracterización de la muestra, sin información personal identificable):
  - **Q-DEM-1**: Edad del participante (respuesta numérica).
  - **Q-DEM-2**: Sexo con el que se identifica (Femenino / Masculino / Prefiero no decirlo).
  - **Q-DEM-3**: Carrera a la que pertenece (lista desplegable con la oferta académica de pregrado de la UTA).
  - **Q-DEM-4**: Nivel del Programa Regular de Inglés que cursa actualmente.
- **Bloque VARK**:
  - **Q-VARK-1**: ¿Cuál es la modalidad que le entregó el cuestionario VARK? (Visual / Auditivo / Lectura-escritura / Kinestésico / Multimodal).
  - **Q-VARK-2**: ¿Está usted de acuerdo con el resultado obtenido en el cuestionario VARK? (Sí / No / Parcialmente).
- **Bloque SUS**: diez ítems estándar en escala Likert de 1 a 5 (totalmente en desacuerdo / totalmente de acuerdo).
- **Q-FB-1**: Comentario o retroalimentación abierta sobre el sistema, ubicado al final del instrumento para que el participante responda con la experiencia completa del SUS ya procesada (campo opcional, texto libre).

La elección del SUS se sustenta en tres motivos. Es una escala independiente del dominio, aplicable a software y hardware, lo que permite usarla sobre el sistema sin adaptaciones que afecten la comparabilidad. Su formato breve de diez ítems reduce la carga cognitiva del estudiante, lo que resulta útil en una sesión en la que también se

aplica el cuestionario VARK y se usa el sistema. Por último, el puntaje global en rango fijo de 0 a 100 facilita la comparación con los valores de referencia publicados en la literatura.

Cuestionario de validez de los objetos de aprendizaje

El segundo instrumento mide la validez del contenido generado por el sistema para los cuatro tipos de bloque que produce sobre cada video: *resumen estructurado*, *flashcards*, *cuestionario de opción múltiple* y *glosario técnico*. Se aplica a estudiantes que usaron los objetos de aprendizaje en su asignatura. El material evaluado corresponde a cuatro videos del temario del curso de Programación Orientada a Objetos en JavaScript: Definición de clase, Herencia, Polimorfismo y Métodos estáticos. La escala empleada es Likert de 1 a 5, donde 5 equivale a *totalmente de acuerdo*. El cuestionario se organiza en dos niveles:

- **Valoración global por tipo de bloque**: cuatro ítems sobre la validez general del resumen, las *flashcards*, el cuestionario y el glosario producidos.
- **Valoración por videos extremos**: el estudiante señala el video que le resultó más útil y el que menos, y emite valoraciones separadas por bloque sobre cada uno. Esto permite estimar cuánto varía la calidad percibida según la calidad del video fuente.

Cada bloque incluye además un campo abierto opcional para comentarios.

2.4.3 Definición operacional del subgrupo evaluador R/W

El sistema apunta al estudiante con preferencia de aprendizaje basada en lectura y escritura (R/W). Para identificar a este subgrupo dentro de la muestra que respondió el cuestionario VARK se adoptó un **criterio estricto**: se considera dentro del subgrupo evaluador únicamente al estudiante cuya modalidad VARK declarada es *Lectura-escritura*.

El perfil *Multimodal*, que agrupa a estudiantes con dos o más modalidades dominantes y puede incluir componentes lectoescritores que la forma estándar del VARK no desagrega, se reporta de manera independiente como punto de comparación.

2.4.4 Participación efectiva

La Tabla 23 muestra la composición del grupo que participó dentro de la fase de evaluación.

Tabla 23: Participación efectiva por fase de evaluación

Encuesta	Población	Respuestas válidas	Participación
SUS (usabilidad)	151	94	62,3 %
Validez del contenido (LO)	86	67	77,9 %

En la fase de usabilidad, las 94 respuestas válidas se distribuyeron posteriormente según la modalidad VARK declarada por cada estudiante; el subgrupo evaluador con preferencia lectoescritora. En la fase de validez del contenido, las 67 respuestas se obtuvieron a partir de los objetos de aprendizaje generados por el sistema sobre los videos del temario del curso de Programación Orientada a Objetos.

2.4.5 Procedimiento

Encuesta de usabilidad

La sesión de usabilidad con los estudiantes de los cursos participantes siguió la secuencia VARK $\rightarrow$ uso del sistema $\rightarrow$ SUS extendido:

1. Se compartieron dos enlaces a los estudiantes: el cuestionario VARK y la encuesta SUS extendida.
2. Los estudiantes respondieron primero el VARK y obtuvieron su modalidad.
3. Usaron el sistema durante la sesión guiada, sobre los videos provistos por el docente.
4. Completaron la encuesta SUS extendida; en Q-VARK-1 indicaron la modalidad obtenida en el paso 2.
5. El investigador empató después cada respuesta SUS con su perfil VARK usando el dato de Q-VARK-1.

Encuesta de validez de objetos de aprendizaje

La aplicación en el curso de Programación Orientada a Objetos se realizó de la siguiente manera:

1. El docente compartió cuatro objetos de aprendizaje generados previamente por el sistema a partir de videos del temario.
2. Los estudiantes revisaron los cuatro tipos de bloque (resumen, *flashcards*, cuestionario y glosario) para cada uno de los cuatro videos.
3. Los estudiantes respondieron el cuestionario con una valoración global por bloque y otra específica sobre el video más útil y el menos útil.

2.4.6 Plan de análisis

Las respuestas del SUS se procesan con la fórmula estándar de Brooke [37]: a los ítems impares se les resta uno al valor marcado; a los ítems pares se les resta el valor marcado a cinco; la suma de los diez resultados se multiplica por 2,5 para obtener el puntaje global en escala 0–100. El cálculo se hace por estudiante. Se reporta el puntaje promedio, la desviación estándar y la proporción de estudiantes que igualan o superan el umbral de 68 puntos, tanto sobre la muestra completa como segmentada por modalidad VARK declarada. Los comentarios del campo abierto Q-FB-1 se agrupan en aspectos positivos y sugerencias de mejora.

Las respuestas de la encuesta de validez de objetos de aprendizaje se analizan con estadística descriptiva: promedio y distribución de frecuencias por ítem, comparación entre el video más útil y el menos útil indicados por cada estudiante, y resumen cualitativo de los comentarios abiertos por tipo de bloque.

CAPÍTULO III

PROPUESTA Y RESULTADOS

Este capítulo presenta la solución desarrollada y los resultados de su aplicación. Se parte del alcance del aporte y la justificación de las herramientas seleccionadas. Luego se describe la arquitectura del sistema, los requerimientos, los casos de uso, el modelo de datos, la interfaz, el manejo de errores y la seguridad. A continuación se documenta la implementación, las pruebas y el plan de despliegue. El capítulo cierra con los resultados de la evaluación piloto y la evidencia asociada a cada objetivo específico.

3.1 Alcance de la propuesta

El aporte original del autor consiste en una API y un conjunto de componentes de procesamiento que transforman videos educativos en cuatro tipos de materiales de aprendizaje estructurados. La solución abarca la ingestión del video, la gestión de estados del procesamiento, la orquestación de la transcripción, la generación de los materiales mediante agentes de contenido especializados, la persistencia de los resultados y los componentes de interfaz de usuario asociados. El resultado de este trabajo se integró satisfactoriamente a NousAI, plataforma educativa de la Universidad Técnica de Ambato, demostrando la aplicabilidad de la solución propuesta como parte de una plataforma educativa. Los componentes compartidos de NousAI, esto es, la autenticación con Microsoft SSO, el modelo de *objeto de aprendizaje* y la gestión de módulos y matrículas, se reutilizan sin ser objeto de esta investigación.

El sistema se posiciona como material de apoyo *complementario* para el estudiante que ya emplea videos como recurso principal de estudio autónomo. La evaluación de aceptación se acota mediante un filtrado VARK previo: el subgrupo evaluador del cuestionario SUS se compone de estudiantes con preferencia lectoescritora.

3.2 Análisis de procesos

El sistema modifica dos flujos: la preparación de material de apoyo a partir de un video y el acceso del estudiante a ese material.

3.2.1 Proceso actual

En el escenario previo a la incorporación del sistema, el docente selecciona o graba un video y lo comparte con sus estudiantes. Cuando desea ofrecer materiales complementarios como un resumen, un glosario, tarjetas o un cuestionario, debe elaborarlos a mano: visualizar el video completo, transcribir segmentos, redactar el resumen, identificar términos, construir preguntas. Es un proceso laborioso que en la práctica deriva en que la mayoría de los videos se distribuyan sin material complementario. El estudiante consume el video de forma lineal y construye sus propios apuntes.

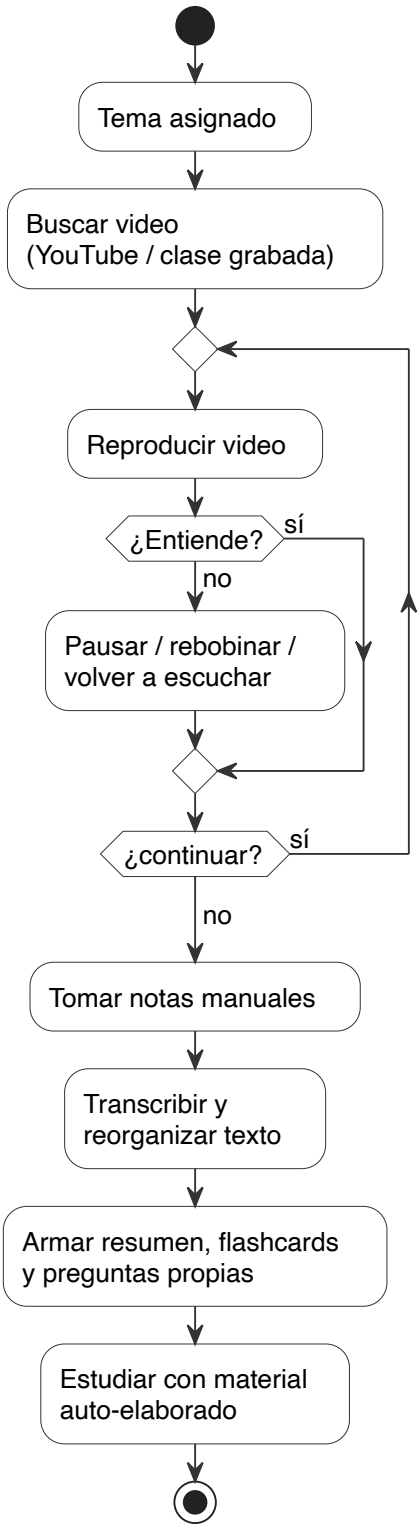

Figura 1: Proceso actual de preparación y consumo de material educativo a partir de videos

3.2.2 Proceso propuesto

El proceso propuesto introduce el sistema entre el docente y el estudiante. El docente registra el video en la plataforma, mediante carga de archivo o indicación de una URL pública, y el sistema inicia el *pipeline* de procesamiento de forma asíncrona. En el orden de uno a tres minutos por video, según duración y latencia del proveedor de LLM, el docente recibe los cuatro materiales generados de forma automática. Puede revisarlos, editarlos o solicitar la regeneración con una instrucción adicional. Una vez conforme, publica el contenido asociado al *objeto de aprendizaje* del módulo, y los estudiantes acceden al video y a los materiales desde la misma interfaz (ver Figura 2).

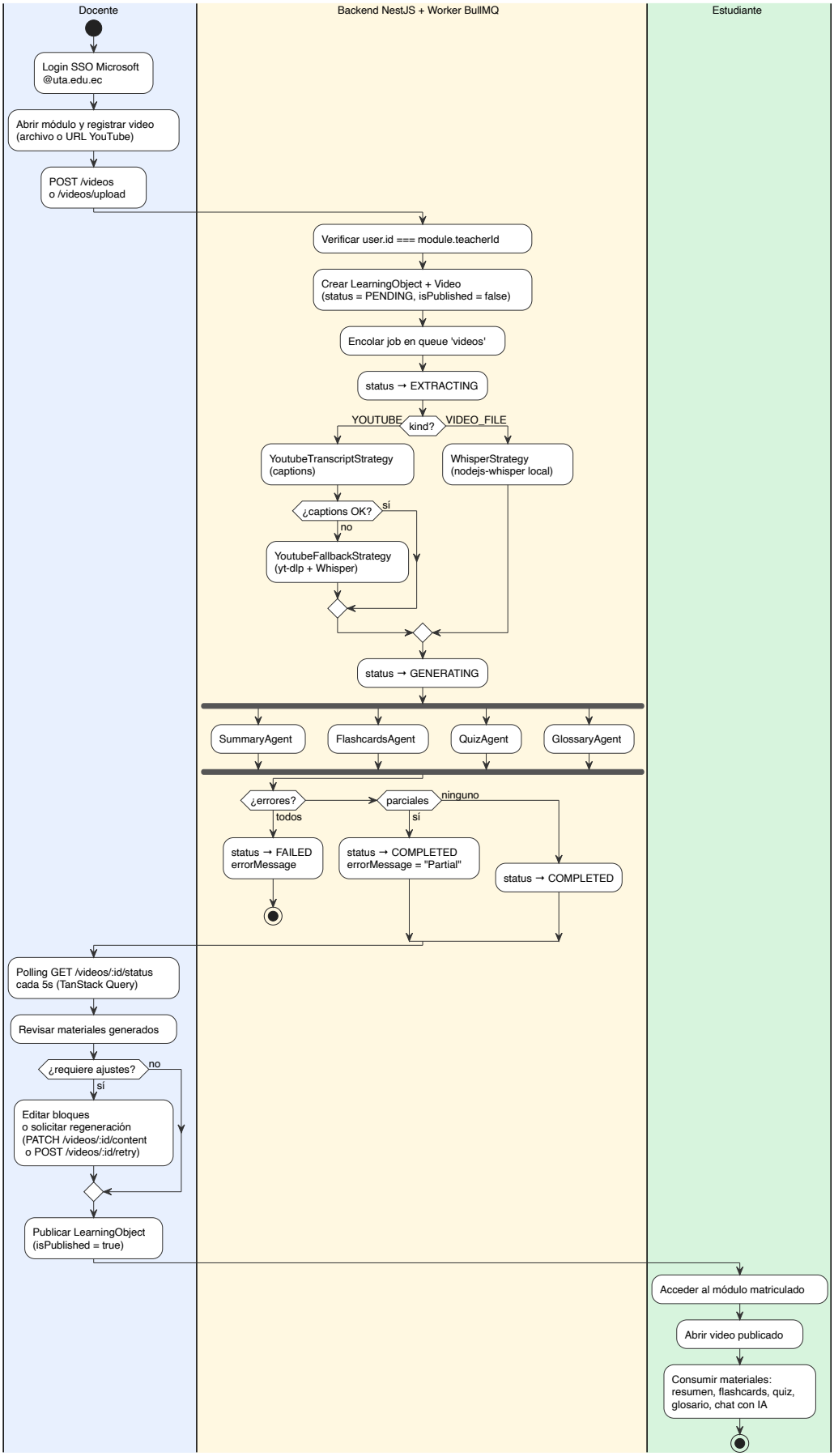

Figura 2: Proceso propuesto con el sistema de transformación automática incorporado

3.3 Justificación de herramientas

Las decisiones tecnológicas se tomaron con un análisis comparativo entre alternativas. Esta sección documenta las decisiones de mayor impacto: motor de transcripción, variante del modelo Whisper y proveedor de modelo de lenguaje.

3.3.1 Motor de reconocimiento automático de voz

La transcripción es el primer paso del *pipeline*. La selección del motor consideró cuatro alternativas frecuentes en la literatura y la industria, evaluadas sobre criterios de calidad, costo, compatibilidad con el *stack* del proyecto y privacidad de los datos. La Tabla 24 resume la comparación.

Tabla 24: Comparativa de motores de reconocimiento automático de voz

Motor	**Modalidad**	**Costo**	**Stack Node**	**Observaciones**
Whisper (OpenAI)	Local / API	Gratuito local; pago por minuto en API	Sí, vía *nodejs-whisper*	Modelo abierto, multilenguaje, sin envío de datos a terceros si se ejecuta localmente.
NVIDIA Parakeet (NIM)	Nube	*Free tier* con cuota	Sí, vía HTTP	Buena precisión en inglés; cuotas limitan procesamiento masivo.
Google Speech-to-Text	Nube	Pago por minuto	Sí, vía SDK / HTTP	Calidad alta; envío de audio a infraestructura externa.
AWS Transcribe	Nube	Pago por minuto	Sí, vía SDK / HTTP	Calidad alta; envío de audio a infraestructura externa.

Se seleccionó **Whisper** ejecutado localmente. Tres factores soportan la decisión: el costo nulo en uso recurrente (relevante en un piloto que procesa decenas de videos durante el desarrollo), la posibilidad de ejecución *on-premise* sin envío de material

académico a terceros y la disponibilidad de un *binding* maduro para Node.js (*nodejs-whisper*), compatible con el *stack* NestJS del proyecto. *Faster-whisper*, otro *binding* popular del modelo Whisper, se descartó por ser un *wrapper* de Python que requeriría introducir un segundo *runtime* solo para la transcripción. Para videos alojados en YouTube, antes de invocar a Whisper se intenta recuperar los subtítulos nativos con la biblioteca *youtube-transcript-plus*, que aprovecha los protocolos públicos de YouTube; cuando los subtítulos no están disponibles, se descarga el audio con *yt-dlp* y se delega la transcripción a Whisper local como mecanismo de *fallback*.

3.3.2 Variantes del modelo Whisper

Whisper se distribuye en seis variantes que comparten arquitectura y difieren en número de parámetros, memoria requerida y tasa de error de palabra (WER). La Tabla 25 resume las variantes disponibles a partir de la *model card* oficial de OpenAI [27].

Tabla 25: Comparativa de variantes del modelo Whisper

Variante	Parámetros	VRAM aprox.	Velocidad rel.	Observaciones
tiny	39 M	1 GB	$\sim$10$\times$	WER alto en idiomas distintos del inglés.
base	74 M	1 GB	$\sim$7$\times$	Compromiso aceptable entre velocidad y calidad.
small	244 M	2 GB	$\sim$4$\times$	WER notablemente mejor sin requerir GPU dedicada.
medium	769 M	5 GB	$\sim$2$\times$	Calidad cercana a *large* con menor memoria.
large-v3	1 550 M	10 GB	1$\times$	Mejor desempeño multilingüe; requiere GPU.
turbo	809 M	6 GB	$\sim$8$\times$	Decoder reducido; sólo ASR.

Se eligió la variante *base* como configuración por defecto, con la posibilidad de elevarla a *small* mediante la variable de entorno `WHISPER_MODEL`. La variante base ofrece la mejor relación calidad sobre tiempo de procesamiento sin requerir GPU.

3.3.3 Proveedor de modelo de lenguaje

La generación de los cuatro materiales requiere cuatro invocaciones independientes al modelo de lenguaje por cada video procesado. Esta característica vuelve al consumo de cuotas un factor decisivo durante el desarrollo. La Tabla 26 resume los proveedores integrados en la *factory* de la solución.

Tabla 26: Comparativa de proveedores de LLM integrados en la *factory*

Proveedor	**Modalidad**	**Cuota gratuita**	**Observaciones**
NVIDIA NIM	Nube	∼40 RPM por modelo, créditos iniciales	*Free tier* amplio; hospeda Llama 3.1/3.3, Nemotron y modelos con razonamiento.
Groq	Nube	∼30 RPM, 14 400 RPD aprox.	Inferencia muy rápida, pero el límite por minuto choca con cuatro llamadas paralelas por video.
Google Gemini	Nube	5–15 RPM según modelo, ∼1 000 RPD en Flash-Lite	Cuotas ajustadas; Pro 2.5 cae a 100 RPD.
OpenAI	Nube	Sin *free tier*; pago por *token*	Calidad alta y *tooling* maduro, costo no compatible con iteración masiva del piloto.
Ollama	Local	Sin cuota	Requiere GPU local; útil como alternativa *offline*.

Se seleccionó **NVIDIA NIM** como proveedor del piloto. La justificación combina dos factores. Primero, **restricción de hardware**: el servidor provisto por la DTIC para el despliegue del piloto no dispone de GPU, por lo que ejecutar un modelo de lenguaje en local mediante Ollama no es viable en ese entorno. El proveedor de LLM debe ser externo. Segundo, **patrón de uso**: cada video dispara cuatro invocaciones simultáneas, una por agente. Bajo ese patrón los *rate limits* gratuitos de Groq (30 RPM) y Gemini (5–15 RPM) se consumen rápido en series de pruebas. NVIDIA NIM ofrece la ventana más amplia entre los proveedores con *free tier* sustentable y mantiene calidad comparable a la de proveedores de pago.

3.4 Arquitectura del sistema

La arquitectura se organiza en tres capas desplegables de forma independiente: un cliente web, un API HTTP que recibe las solicitudes y un *worker* separado que consume los trabajos de la cola y ejecuta el procesamiento pesado. La comunicación entre API y *worker* se realiza a través de una cola Redis. Ambos procesos comparten una base de datos PostgreSQL con extensión `pgvector` como almacén persistente. Los proveedores externos (motor ASR y proveedores de LLM) se integran al sistema mediante adaptadores que implementan interfaces comunes. Dentro del bloque servidor, la generación se ejecuta a través de cuatro agentes especializados, uno por cada tipo de material producido (resumen, *flashcards*, cuestionario y glosario). El proveedor de

LLM se representa como un nodo intercambiable en el diagrama; los modelos concretos se configuran en tiempo de ejecución.

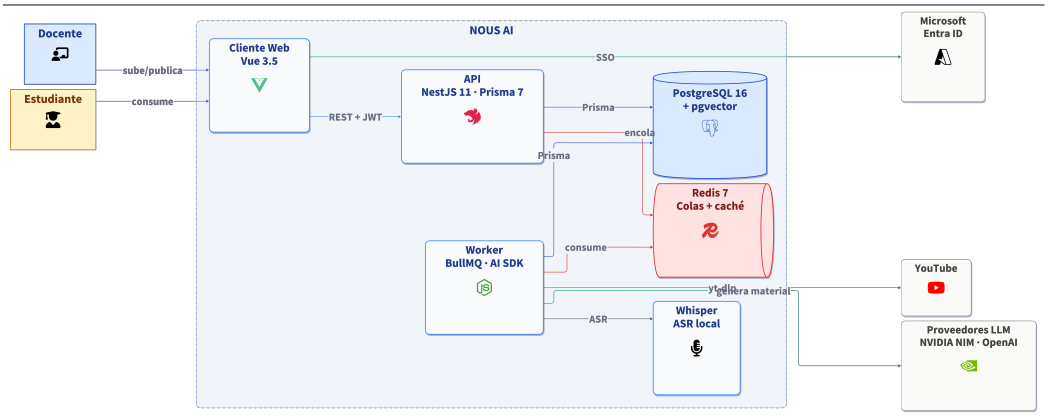

Figura 3: Arquitectura del sistema — Modelo C4 Nivel 2 (contenedores)

3.4.1 Organización modular del backend

La organización interna del backend sigue el modelo modular por *features* de NestJS. Cada *feature* agrupa en una carpeta autocontenida el controlador, los servicios, los *DTOs*, los *guards* y los *workers* asociados a una capacidad. La *feature* central de este trabajo, `src/features/videos/`, encapsula la totalidad de la lógica de transformación y se apoya en cuatro submódulos: `main/` (servicio orquestador, ingestión, máquina de estados), `transcription/` (estrategias de transcripción), `ai/` (agentes de generación, *registry* de agentes y *factory* de proveedores) y `workers/` (consumidor de la cola de procesamiento).

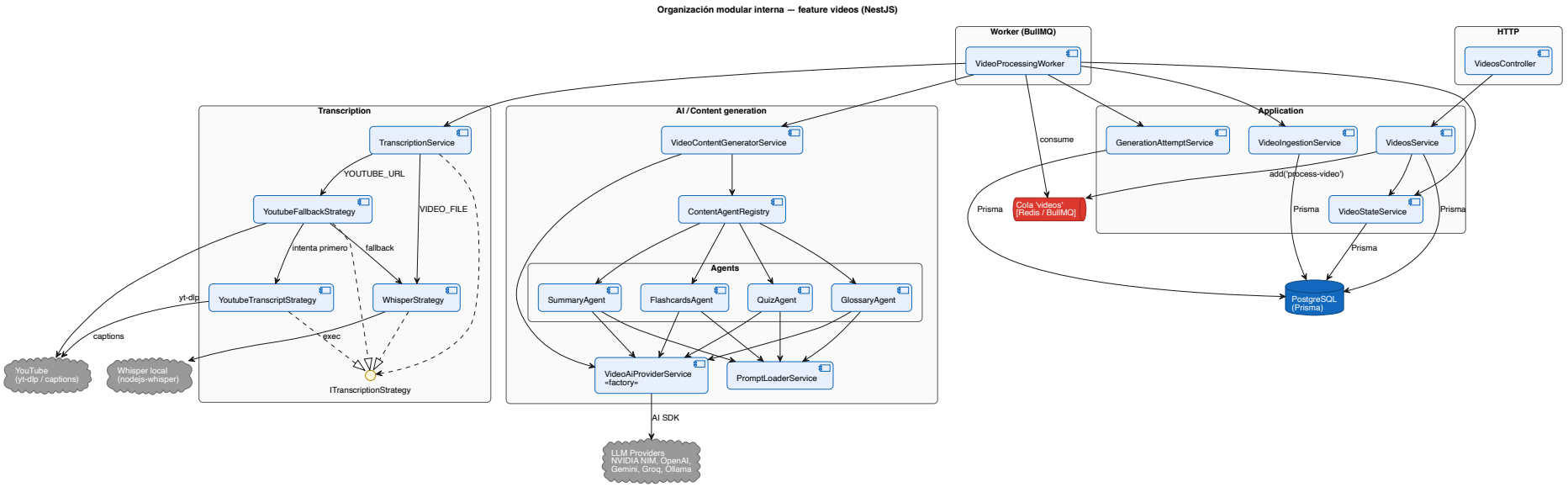

Figura 4: Organización modular interna de la *feature* **videos**

3.4.2 Patrones de diseño aplicados

Tres patrones de diseño separan las decisiones tecnológicas de la lógica funcional. El patrón **Strategy** se aplica en el servicio de transcripción: tres estrategias concretas (`YoutubeTranscriptStrategy`, `YoutubeFallbackStrategy` y `WhisperStrategy`) implementan una interfaz común `ITranscriptionStrategy`. El patrón **Factory** se aplica en el proveedor de modelo de lenguaje: el `VideoAiProviderService` mantiene un mapa interno que asocia cada proveedor configurable con una función fábrica que instancia el adaptador correspondiente en tiempo de ejecución. El patrón **Template Method** se aplica en los agentes de contenido: la clase abstracta `BaseContentAgent` define el flujo común (cargar *prompt*, invocar al modelo con esquema `Zod`, validar, recuperar) y cada subclase concreta aporta su `blockType`, su `schema` y su método `assignTo`. Los cuatro agentes son clases distintas con esquemas Zod distintos, no instancias del mismo servicio con *prompts* variables.

3.5 Requerimientos del sistema

A partir del análisis realizado y del marco conceptual revisado se consolidó el catálogo de requerimientos siguiendo el estándar IEEE 830. Los requerimientos funcionales (RF) describen capacidades. Los no funcionales (RNF) caracterizan atributos de calidad transversales.

Tabla 27: Requerimientos funcionales

ID	Descripción
RF-01	El sistema deberá permitir al docente registrar un video para procesamiento, ya sea mediante una URL pública de YouTube o mediante la carga de un archivo local.
RF-02	El sistema deberá validar el formato y la unicidad de la entrada antes de iniciar el procesamiento.
RF-03	El sistema deberá procesar cada video de forma asíncrona y retornar al cliente un identificador de seguimiento al momento de registrar la solicitud.
RF-04	El sistema deberá exponer el estado del procesamiento de cada video en cualquier momento posterior al registro.
RF-05	El sistema deberá obtener una transcripción del contenido hablado del video.
RF-06	El sistema deberá generar automáticamente cuatro tipos de materiales de aprendizaje a partir de la transcripción: resumen estructurado, *flashcards*, cuestionario de opción múltiple y glosario técnico.

Continúa en la página siguiente

(Continuación Tabla 27)

ID	Descripción
RF-07	El sistema deberá permitir al docente solicitar la regeneración de uno o varios tipos de materiales, incluyendo opcionalmente una instrucción textual para guiar al modelo.
RF-08	El sistema deberá permitir al docente editar manualmente el contenido de los materiales generados y registrar la marca de edición manual.
RF-09	El sistema deberá registrar en una tabla de auditoría cada intento de generación, incluyendo proveedor, modelo, *tokens* consumidos, tiempo de procesamiento y resultado por tipo.
RF-10	El sistema deberá permitir al estudiante visualizar el video junto con los cuatro materiales generados, organizados de forma navegable.
RF-11	El sistema deberá presentar el cuestionario en modo interactivo con corrección inmediata y explicación asociada a cada respuesta.
RF-12	El sistema deberá permitir al docente eliminar lógicamente un *objeto de aprendizaje* y sus materiales asociados.

Tabla 28: Requerimientos no funcionales

ID	Atributo	Descripción
RNF-01	Disponibilidad	El *endpoint* de consulta de estado deberá responder en menos de 300 ms para cualquier video registrado.
RNF-02	Modularidad	La generación deberá implementarse mediante cuatro agentes especializados, uno por cada tipo de material producido.
RNF-03	Escalabilidad	Los *workers* deberán poder escalarse horizontalmente de forma independiente del API, consumiendo trabajos desde la misma cola Redis.
RNF-04	Tolerancia a fallos	Los trabajos fallidos deberán reintentarse de forma automática (`attempts: 2`) con retroceso exponencial (`delay: 5000 ms`) mediante la configuración de la cola BullMQ, antes de marcar el video como `FAILED`.
RNF-05	Observabilidad	El sistema deberá exponer *endpoints* de salud (documentados en el Anexo de API) y registrar *logs* estructurados en cada transición de estado.

Continúa en la página siguiente

(Continuación Tabla 28)

ID	Atributo	Descripción
RNF-06	Trazabilidad	El sistema deberá registrar *logs* de las invocaciones a proveedores externos (transcripción y LLM) en archivos rotados en el contenedor del *worker*, junto con el *videoId* y la duración de la operación.
RNF-07	Seguridad	Todos los *endpoints* deberán estar protegidos por autenticación JWT, con excepción de los de salud. La escritura sobre videos requerirá rol `TEACHER`.
RNF-08	Compatibilidad	La interfaz web deberá ser compatible con navegadores modernos que cumplan los estándares web vigentes.
RNF-09	Verificabilidad	El sistema deberá contar con una suite de pruebas automatizadas (unitarias y de integración) con cobertura mínima del 60 % sobre los componentes críticos del *pipeline*.
RNF-10	Internacionalización	La interfaz web deberá soportar al menos dos idiomas (español e inglés) mediante un mecanismo de *i18n* configurable por el usuario.

3.6 Casos de uso

El sistema involucra dos actores principales. El **docente** (rol `TEACHER`) registra videos, supervisa el procesamiento, edita los materiales generados, solicita regeneraciones y publica el contenido. El **estudiante** (rol `STUDENT`) accede al *Learning Object* una vez publicado y consume tanto el video como los materiales generados. Adicionalmente, el proveedor externo de modelo de lenguaje participa como sistema colaborador durante la etapa de generación.

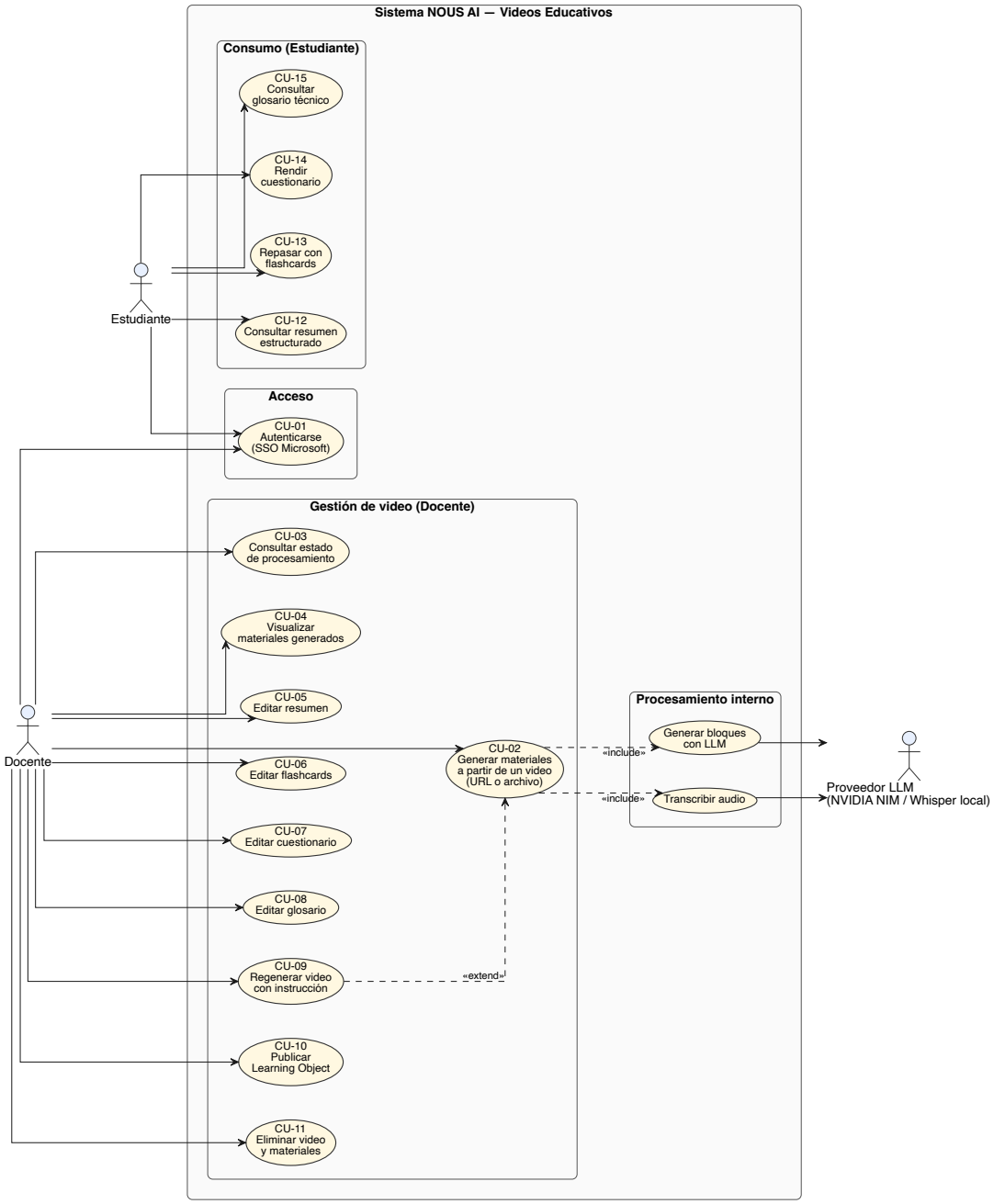

Figura 5: Diagrama de casos de uso del sistema

Tabla 29: Casos de uso principales del sistema

ID	Actor	Caso de uso
CU-01	Docente, Estudiante	Autenticarse mediante Microsoft SSO institucional
CU-02	Docente	Generar materiales a partir de un video (URL de YouTube o archivo local como flujos alternos)
CU-03	Docente	Consultar el estado de procesamiento de un video

Continúa en la página siguiente

(Continuación Tabla 29)

ID	Actor	Caso de uso
CU-04	Docente	Visualizar los materiales generados
CU-05	Docente	Editar el resumen
CU-06	Docente	Editar las *flashcards*
CU-07	Docente	Editar el cuestionario
CU-08	Docente	Editar el glosario
CU-09	Docente	Regenerar video con instrucción *(extiende a CU-02)*
CU-10	Docente	Publicar el *objeto de aprendizaje* para los estudiantes
CU-11	Docente	Eliminar un video y sus materiales asociados
CU-12	Estudiante	Consultar el resumen estructurado
CU-13	Estudiante	Repasar mediante *flashcards* interactivas
CU-14	Estudiante	Rendir el cuestionario con retroalimentación
CU-15	Estudiante	Consultar el glosario técnico

3.7 Modelo de datos

El modelo se construye sobre la entidad compartida `LearningObject` de NousAI. La relación entre un video y su *objeto de aprendizaje* es uno a uno con clave primaria compartida: el campo `learningObjectId` de la tabla `videos` actúa simultáneamente como clave primaria y como clave foránea hacia `learning_objects.id`. La eliminación de un *objeto de aprendizaje* se efectúa de forma *lógica* mediante la bandera `deletedAt`; los registros no se eliminan físicamente para preservar trazabilidad.

La tabla `videos` guarda el estado actual del procesamiento, la transcripción completa, el idioma detectado, la duración, el origen y la bandera `hasManualEdits`. Cada ejecución del *pipeline* de generación se registra como una fila en la tabla de auditoría `video_generation_attempts`. Los materiales generados no se almacenan en la tabla de videos: se persisten como bloques tipados (`SUMMARY`, `FLASHCARDS`, `QUIZ`, `GLOSSARY`) vinculados al *objeto de aprendizaje* a través de la tabla `blocks`.

La base de datos representada en la Figura 6 es la base compartida de NousAI; el diagrama presenta únicamente las tablas relevantes a esta solución. El esquema completo, incluidos los campos de *learning_objects* (*id*, *module_id*, *title*, *order_index*, *embedding*, *keywords*, *compiled_content*, *is_published*, *ai_response_id*, *has_manual_edits*, *concepts_processed*, *created_at*, *updated_at*, *type_id*), se encuentra documentado en el archivo *prisma/schema.prisma* del repositorio del proyecto.

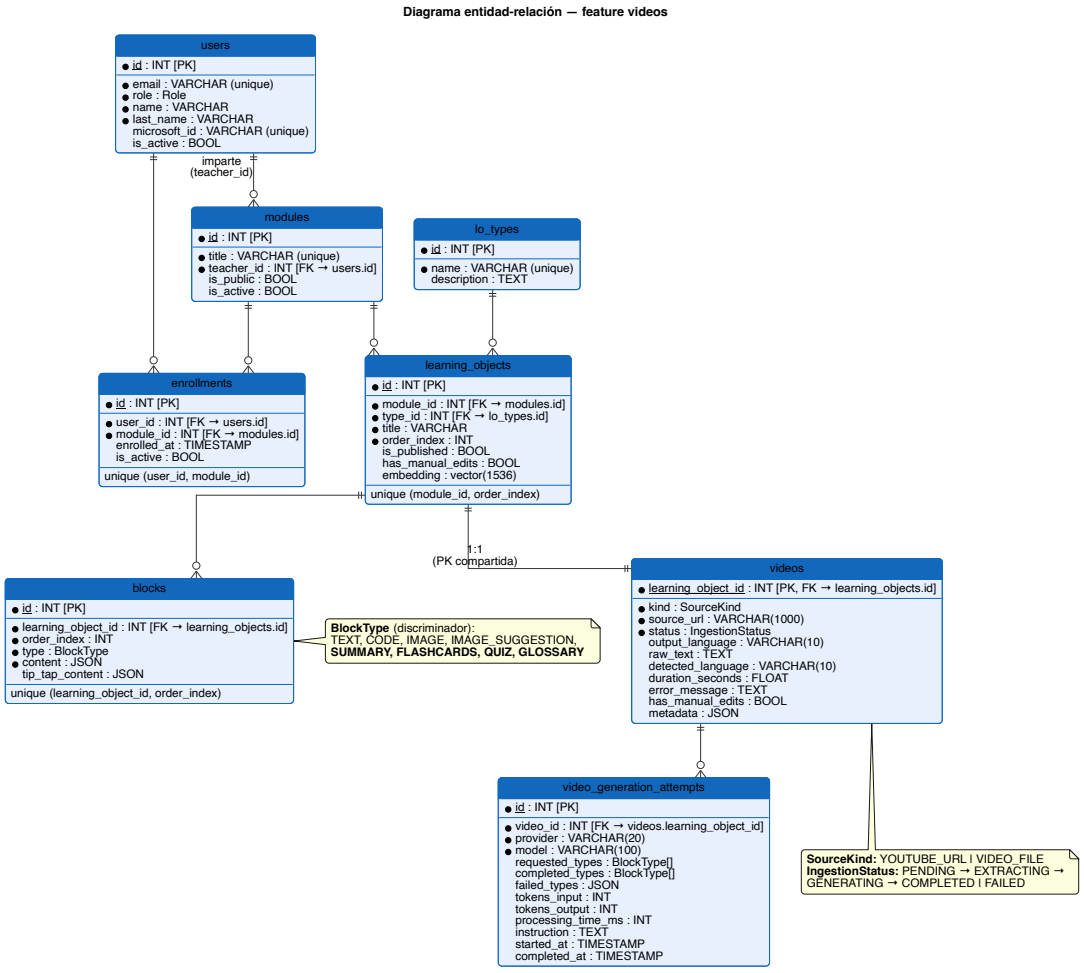

Figura 6: Diagrama relacional de la base de datos — tablas relevantes a la *feature* **videos**

3.8 Diseño del frontend e interfaz

La interfaz web se construyó con Vue 3.5 y Vite 7. La gestión de estado del cliente combina Pinia 3 (con pinia-plugin-persistedstate para sesión) y TanStack Vue Query 5 para el estado del servidor. El enrutamiento usa Vue Router 4; la capa de UI emplea Tailwind CSS 4 sobre componentes sin estilos de reka-ui 2.8. El editor de bloques se construye sobre Tiptap 3. Las validaciones de formulario usan Zod 4. Las llamadas HTTP se hacen con Axios. La estructura sigue el modelo modular por features, en paralelo al backend. El módulo features/videos/ encapsula vistas, componentes, servicios de acceso al API y composables de estado.

A continuación se presentan capturas de las pantallas principales del sistema desplegado. Cada captura corresponde a la versión actualmente en operación dentro de NousAI.

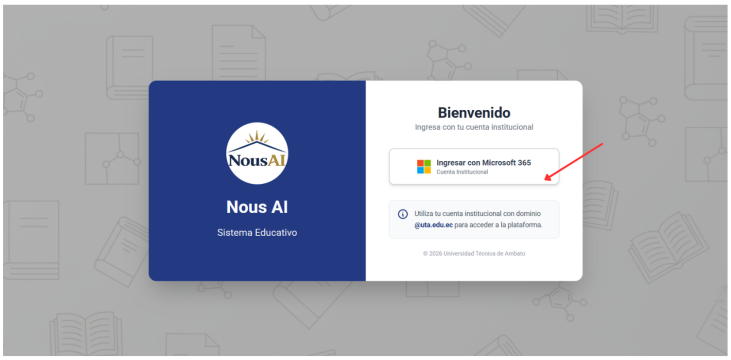

Figura 7: Pantalla de inicio de sesión vía Microsoft SSO institucional

La pantalla de inicio de sesión (Figura 7) delega la autenticación al SSO institucional de la UTA. El usuario se redirige a Microsoft Entra ID y vuelve al sistema con un JWT firmado.

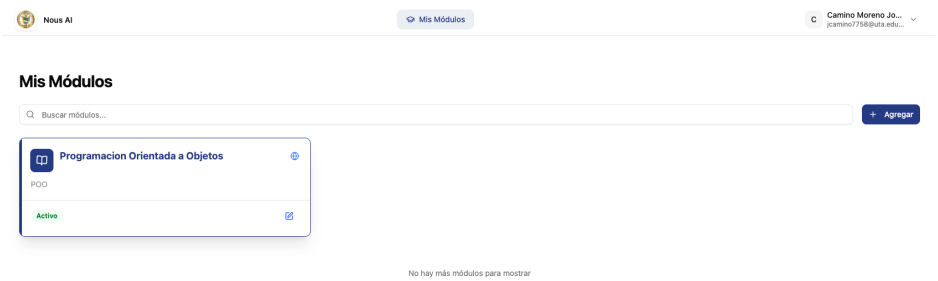

Figura 8: Dashboard del docente con sus módulos y *objetos de aprendizaje* de tipo video

El dashboard del docente (Figura 8) presenta los módulos a los que está asignado y, dentro de cada uno, los videos registrados con su estado de procesamiento.

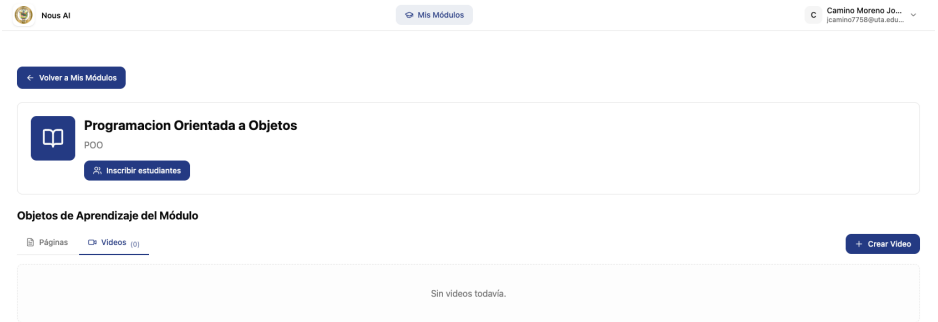

Figura 9: Vista detalle de un módulo con la lista de *objetos de aprendizaje*

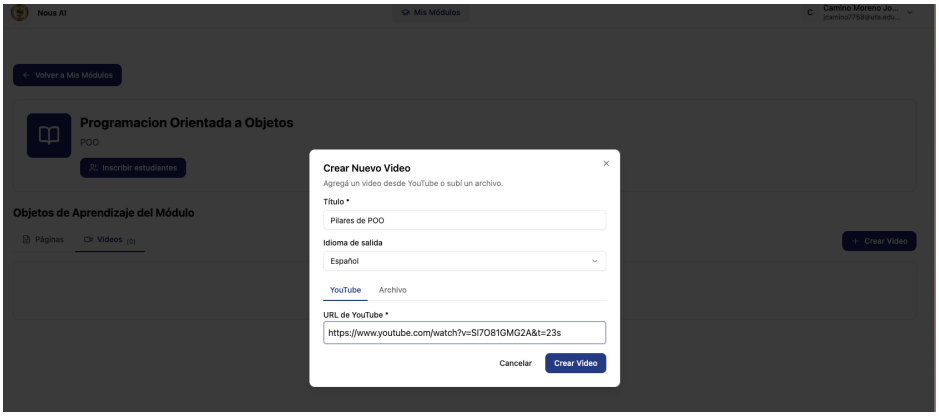

Figura 10: Diálogo de registro de video (URL pública o archivo local)

El diálogo de registro (Figura 10) acepta una URL de YouTube o un archivo local; el sistema valida la entrada y devuelve un identificador de seguimiento al cliente.

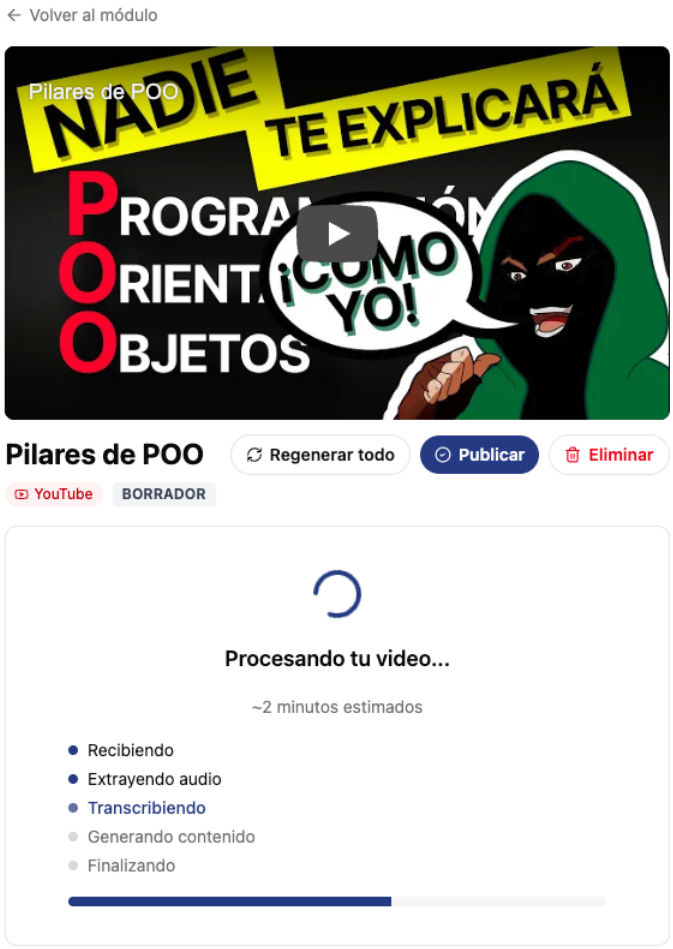

Figura 11: Estado de procesamiento en tiempo real durante el *pipeline*

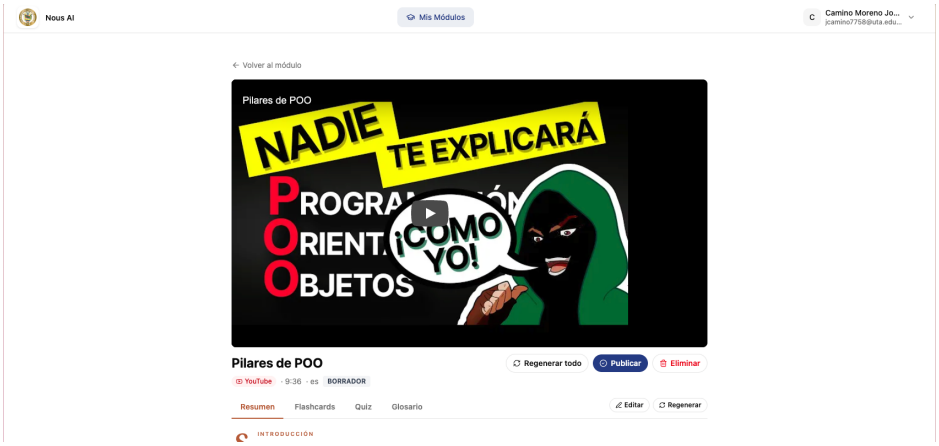

Figura 12: Detalle del video generado, con pestañas de navegación entre los cuatro materiales

Figura 13: Visualización del resumen estructurado generado por el sistema

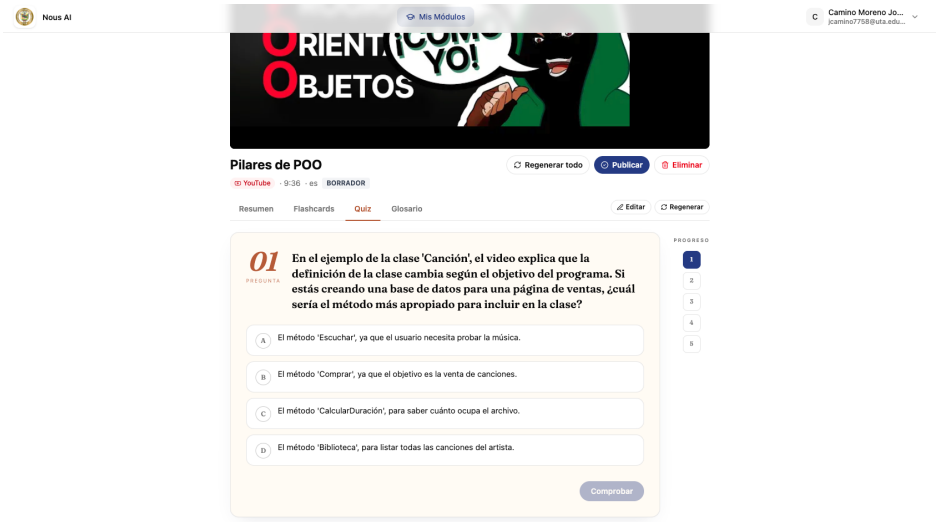

Figura 14: Cuestionario de opción múltiple con corrección inmediata

Figura 15: Glosario técnico con términos extraídos del video

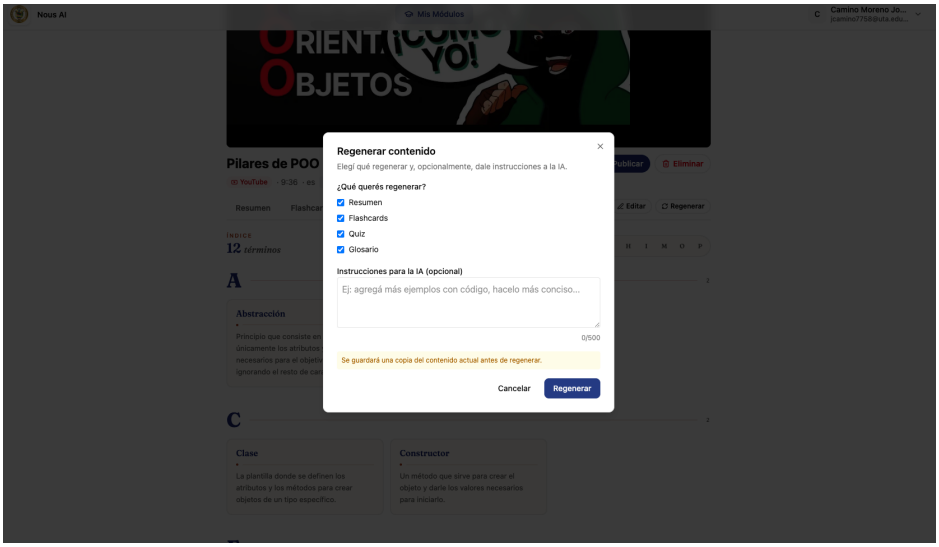

Figura 16: Diálogo de regeneración de contenido con instrucción opcional al modelo

3.9 Manejo de errores y notificaciones

El sistema separa los errores en tres niveles: errores de validación de entrada, errores de proveedor externo y errores irrecuperables del *pipeline*. Los errores de validación se devuelven al cliente con código HTTP 4xx y un cuerpo JSON con la lista de fallas detectadas. Los errores de proveedor externo se reintentan según la política BullMQ documentada en RNF-04 (`attempts: 2`, retroceso exponencial con `delay: 5000 ms`). Los errores irrecuperables marcan el video como `FAILED` y registran la causa en `video_generation_attempts`.

La interfaz notifica al docente mediante *toasts* no bloqueantes construidos sobre la librería `vue-toastification`, centralizados en el *composable* `useToast`. Los casos cubiertos son: éxito al registrar el video, transición de estado, fallo del *pipeline* con descripción legible y éxito o fallo de la regeneración. La detección de cambios se realiza mediante *polling* con TanStack Vue Query al *endpoint* de estado, con intervalo declarado en la constante `POLLING_INTERVAL_MS` (5000 ms).

3.10 Seguridad

El sistema aplica controles de seguridad en tres planos.

Autenticación. El acceso a la API requiere un JWT firmado, emitido tras el SSO institucional de Microsoft Entra ID. El JWT se valida en cada solicitud mediante un

guard global de NestJS, con excepción de los *endpoints* de salud que se mantienen abiertos para sondas de operación.

Control de acceso. Cada *endpoint* con efecto de escritura sobre videos exige el rol `TEACHER`. Los *endpoints* de lectura para estudiantes verifican que el usuario esté matriculado en el módulo al que pertenece el *objeto de aprendizaje* consultado. La verificación se realiza mediante un *guard* adicional que consulta la tabla compartida de matrículas de NousAI.

Validación de entrada. El sistema aplica tres capas de validación. En el arranque del proceso, el módulo de configuración valida todas las variables de entorno con un esquema **Joi** (`src/core/config/constants/index.ts`), de modo que el servicio solo inicia con configuración consistente (por ejemplo, `JWT_SECRET` exige al menos 30 caracteres y `VIDEO_AI_PROVIDER` debe pertenecer al conjunto {`groq`, `openai`, `google`, `nvidia`, `ollama`}). En el límite del controlador HTTP, las entradas de cada *endpoint* se validan con *DTOs* de `class-validator`. Las salidas de los modelos de lenguaje se validan contra esquemas `Zod` estrictos antes de persistirse; cuando la validación estricta falla, se intenta una recuperación con un esquema laxo que normaliza la salida y, si tampoco eso recupera, el agente correspondiente registra el error y el resto del *pipeline* continúa con los materiales que sí se generaron correctamente.

3.11 Implementación

3.11.1 Componentes clave

A continuación se documentan los componentes que mejor ilustran las decisiones arquitectónicas. El código completo está disponible en `https://github.com/Chu2409/edu-assistant-api`.

Máquina de estados del video

La máquina de estados controla las transiciones válidas del ciclo de vida de un video y evita estados inconsistentes (por ejemplo, pasar de `PENDING` a `COMPLETED` sin haber transcrito el contenido). La implementación declara la tabla de transiciones permitidas y lanza una excepción cuando se intenta una transición no contemplada.

```
1 const ALLOWED_TRANSITIONS: Record<IngestionStatus, IngestionStatus[]> = {
```

```
  [IngestionStatus.PENDING]:    [IngestionStatus.EXTRACTING, IngestionStatus.FAILED],
  [IngestionStatus.EXTRACTING]: [IngestionStatus.GENERATING, IngestionStatus.FAILED],
  [IngestionStatus.GENERATING]: [IngestionStatus.COMPLETED, IngestionStatus.FAILED],
  [IngestionStatus.COMPLETED]:  [IngestionStatus.GENERATING],
  [IngestionStatus.FAILED]:     [IngestionStatus.GENERATING, IngestionStatus.EXTRACTING],
};

canTransition(current: IngestionStatus, next: IngestionStatus): boolean {
  return ALLOWED_TRANSITIONS[current].includes(next);
}

async transition(videoId: number, next: IngestionStatus,
                 extra?: { errorMessage?: string | null },
                 client: StateClient = this.dbService): Promise<void> {
  const video = await client.video.findUnique({ where: { learningObjectId: videoId } });
  if (!this.canTransition(video.status, next)) {
    throw new BusinessException(409, `Invalid transition: ${video.status} -> ${next}`);
  }
  await client.video.update({
    where: { learningObjectId: videoId },
    data: { status: next, ...extra },
  });
}
```

Selector de estrategia de transcripción

El servicio de transcripción aplica el patrón *Strategy* para escoger la implementación según el tipo de fuente. El *worker* consume la interfaz común y permite agregar nuevas estrategias sin modificar el código consumidor.

```
export class TranscriptionService {
  private readonly strategies: Record<SourceKind, ITranscriptionStrategy>;

  constructor(youtubeFallback: YoutubeFallbackStrategy, whisper: WhisperStrategy) {
    this.strategies = {
      [SourceKind.YOUTUBE_URL]: youtubeFallback,
      [SourceKind.VIDEO_FILE]:  whisper,
```

```
    };
  }

  async transcribe(input: TranscriptionInput): Promise<TranscriptionResult> {
    const strategy = this.strategies[input.kind];
    const result = await strategy.transcribe(input);
    return { ...result, text: cleanTranscript(result.text) };
  }
}
```

Fábrica del proveedor de LLM

La elección del proveedor de LLM, comparados en la Tabla 26, se materializa mediante un registro interno de fábricas en el `VideoAiProviderService`. El cambio de proveedor se efectúa modificando la variable de entorno `VIDEO_AI_PROVIDER` sin tocar código.

```
const providerMap = new Map<string, ProviderFactory>([
  ['groq',   (cfg) => createGroq({ apiKey: cfg.GROQ_API_KEY })],
  ['openai', (cfg) => createOpenAI({ apiKey: cfg.OPENAI_API_KEY }).chat(cfg.
↪ VIDEO_AI_MODEL)],
  ['google', (cfg) => createGoogleGenerativeAI({ apiKey: cfg.
↪ GOOGLE_GENERATIVE_AI_API_KEY })],
  ['nvidia', (cfg) => wrapLanguageModel({
      model: createOpenAICompatible({ name: 'nim', baseURL: cfg.
↪ NVIDIA_BASE_URL,
            headers: { Authorization: `Bearer ${cfg.NVIDIA_API_KEY}` } }),
      middleware: [extractJsonMiddleware(), reasoningContentFallbackMiddleware
↪ ],
    })],
  ['ollama', (cfg) => wrapLanguageModel({
      model: createOpenAICompatible({ name: 'ollama', baseURL: `${cfg.
↪ OLLAMA_BASE_URL}/v1` }),
      middleware: [extractJsonMiddleware(), reasoningContentFallbackMiddleware
↪ ],
    })],
]);

getModel(): LanguageModel {
  const provider = this.config.get('VIDEO_AI_PROVIDER');
  const factory = providerMap.get(provider);
  if (!factory) throw new Error(`Unknown provider: ${provider}`);
  return factory(this.config);
}
```

Agente de contenido con validación por esquema

Cada agente hereda de `BaseContentAgent` y aporta tres elementos propios: el tipo de bloque (`blockType`), el esquema Zod que restringe la salida del modelo (`schema`) y el método `assignTo` que ubica el resultado dentro del objeto `GenerationResult` consumido por el orquestador. La clase base se encarga del flujo común: cargar el *prompt*, invocar el modelo con esquema estricto, intentar recuperación con esquema laxo si falla y agregar métricas de *tokens*.

```
async execute(model: LanguageModel, input: GenerationInput, timeout: number,
              buildProviderOptions = NOOP_PROVIDER_OPTIONS): Promise<
↪ AgentExecutionResult> {
  const prompt = this.promptLoader.getPrompt(this.taskName, input);
  const temperature = this.promptLoader.getTemperature(this.taskName);
  const maxOutputTokens = this.promptLoader.getMaxTokens(this.taskName);

  try {
    const result = await generateText({
      model,
      output: Output.object({ schema: this.schema }),
      prompt, temperature, maxOutputTokens,
      abortSignal: AbortSignal.timeout(timeout),
      providerOptions: buildProviderOptions(this.schema),
    });
    return { data: result.experimental_output, ...extractTokens(result),
↪ needsReview: false };
  } catch (err) {
    if (err instanceof NoObjectGeneratedError && this.lenientSchema && this.
↪ normalize) {
      const lenient = await this.retryWithLenientSchema(model, input, timeout)
↪ ;
      return { data: this.normalize(lenient.data), ...lenient, needsReview:
↪ true };
    }
    throw err;
  }
}
```

Los cuatro agentes definen sus esquemas Zod en módulos independientes. El listado siguiente resume la forma estricta de cada uno; los esquemas laxos asociados conservan la misma estructura pero relajan tipos escalares para tolerar respuestas parcialmente bien formadas.

Listing 3.1: Esquema Zod del bloque resumen (`summary.schema.ts`)

```ts
export const summarySchema = z.object({
  title: z.string(),
  whatYouLearn: z.array(z.string()),
  keyConcepts: z.array(z.string()),
  examples: z.array(z.string()),
  summary: z.string(),
})
```

Listing 3.2: Esquema Zod del bloque *flashcards* (`flashcards.schema.ts`)

```ts
export const flashcardsSchema = z.object({
  items: z.array(
    z.object({
      front: z.string(),
      back: z.string(),
    }),
  ),
})
```

Listing 3.3: Esquema Zod del bloque cuestionario (`quiz.schema.ts`)

```ts
export const quizSchema = z.object({
  questions: z.array(
    z.object({
      question: z.string().min(1),
      options: z.array(z.string().min(1)).length(4),
      correctAnswer: z.number().int().min(0).max(3),
      explanation: z.string().min(1),
    }),
  ).min(1),
})
```

Listing 3.4: Esquema Zod del bloque glosario (`glossary.schema.ts`)

```ts
export const glossarySchema = z.object({
  terms: z.array(
    z.object({
      term: z.string(),
      definition: z.string(),
    }),
  ),
})
```

El método `assignTo` es específico de cada subclase y dirige el contenido validado al campo correspondiente del resultado agregado. Las cuatro implementaciones se reducen a una asignación tipada:

Listing 3.5: Métodos `assignTo` de los cuatro agentes

```
// SummaryAgent
assignTo(result, data) { result.summary = data }

// FlashcardsAgent
assignTo(result, data) { result.flashcards = data }

// QuizAgent
assignTo(result, data) { result.quiz = data }

// GlossaryAgent
assignTo(result, data) { result.glossary = data }
```

Este diseño permite que el orquestador itere sobre el registro de agentes sin conocer la estructura interna de cada bloque: cada agente declara qué genera, cómo se valida y dónde se guarda.

Pipeline del *worker*

El `VideoProcessingWorker` orquesta la ejecución completa con transiciones explícitas de estado antes y después de cada etapa crítica.

```
private async handleProcess(job: Job<ProcessJobData>) {
  const { videoId } = job.data;
  await this.stateService.transition(videoId, IngestionStatus.EXTRACTING);
  const loadedVideo = await this.ingestionService.loadForProcessing(videoId);
  const transcription = await this.transcriptionService.transcribe({
    kind: loadedVideo.kind,
    url: loadedVideo.sourceUrl,
    filePath: loadedVideo.kind === SourceKind.VIDEO_FILE ? loadedVideo.
↪ sourceUrl : undefined,
    language: loadedVideo.outputLanguage,
  });
  await this.ingestionService.persistTranscription(videoId, transcription);
  await this.stateService.transition(videoId, IngestionStatus.GENERATING);
  const generated = await this.contentGenerator.generateAll({
    transcription: transcription.text,
    language: transcription.language,
    videoTitle: loadedVideo.title,
  });
  await this.persistAndFinalize(videoId, ALL_BLOCK_TYPES, generated, job.
↪ timestamp);
}
```

3.11.2 Diagramas de secuencia

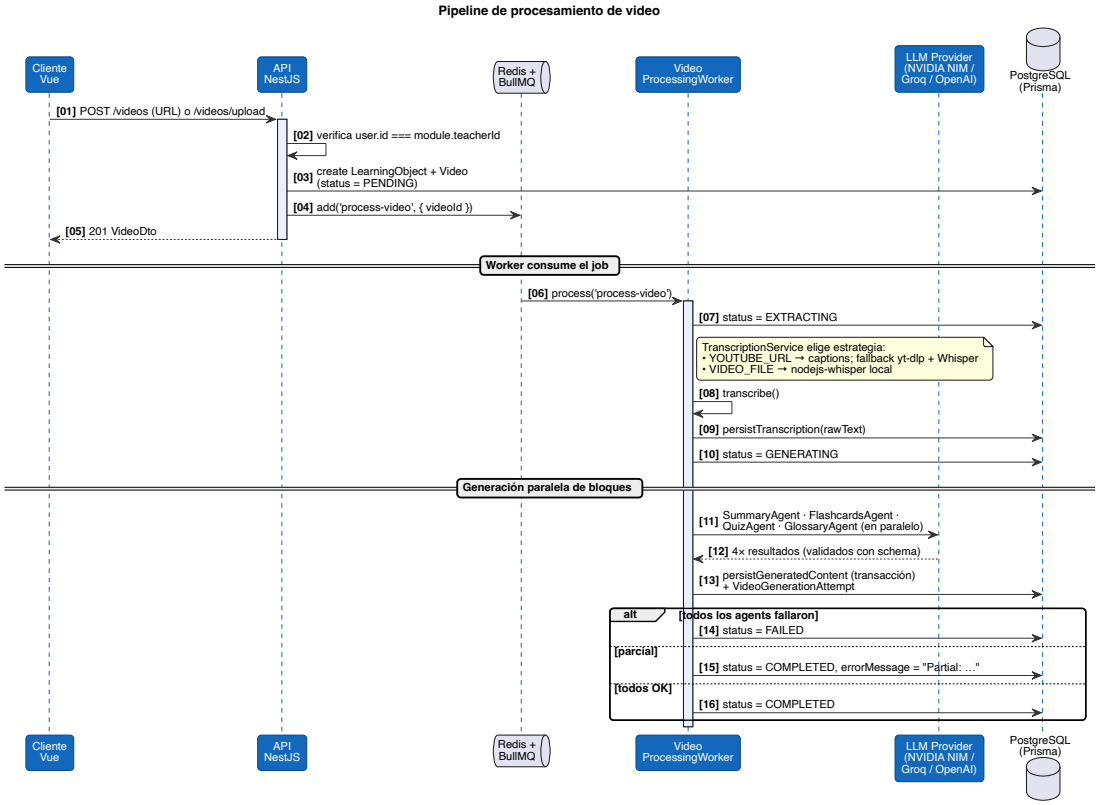

Figura 17: Diagrama de secuencia del *pipeline* de procesamiento de video

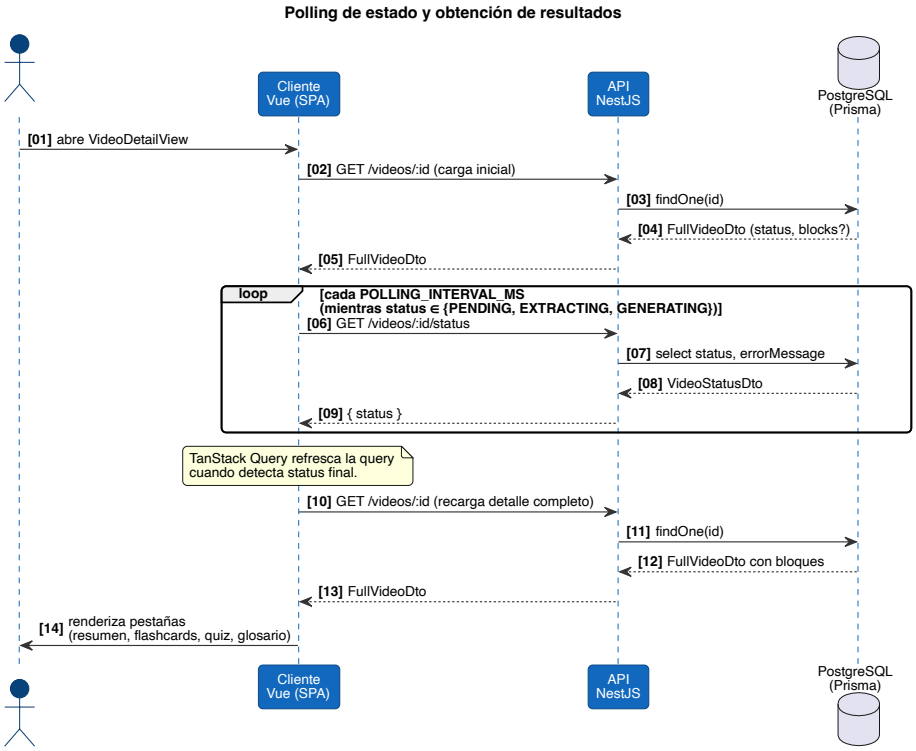

Figura 18: Diagrama de secuencia de *polling* de estado y obtención de resultados

3.12 Pruebas

El aseguramiento de la calidad se organizó en tres niveles de pruebas funcionales aplicadas durante el desarrollo: caja blanca sobre componentes críticos, caja negra sobre los *endpoints* y pruebas de integración sobre el *pipeline* completo. Las pruebas se ejecutaron de forma manual durante las iteraciones de cada Sprint y se verificaron por inspección de logs y por uso real del sistema.

3.12.1 Pruebas de caja blanca

Las pruebas de caja blanca validaron las rutas lógicas internas de los componentes críticos: la máquina de estados, el selector de estrategia de transcripción y el flujo de recuperación con esquema laxo del agente base.

Tabla 30: Casos de prueba de caja blanca

ID	Caso	Resultado
PCB-01	`PENDING` → `EXTRACTING`: transición válida	Aprobado
PCB-02	`EXTRACTING` → `GENERATING`: transición válida	Aprobado
PCB-03	`GENERATING` → `COMPLETED`: transición válida	Aprobado
PCB-04	`COMPLETED` → `GENERATING`: *retry* desde estado final	Aprobado
PCB-05	`FAILED` → `EXTRACTING`: *retry* desde error temprano	Aprobado
PCB-06	`PENDING` → `COMPLETED`: transición inválida rechazada	Aprobado
PCB-07	`COMPLETED` → `FAILED`: transición inválida rechazada	Aprobado
PCB-08	Selector `YOUTUBE_URL` → `YoutubeFallbackStrategy`	Aprobado
PCB-09	Selector `VIDEO_FILE` → `WhisperStrategy`	Aprobado
PCB-10	Recuperación con `lenientSchema` cuando el estricto falla	Aprobado

3.12.2 Pruebas de caja negra

Los casos se diseñaron a partir del catálogo de RFs y se ejecutaron manualmente contra los *endpoints* del API mediante un cliente HTTP, aplicando particiones de equivalencia y valores límite en las entradas.

Tabla 31: Casos representativos de prueba de caja negra

ID	Req.	Caso	Resultado
PCN-01	RF-01	`POST /videos` con URL válida → 201 + ID	Aprobado
PCN-02	RF-02	`POST /videos` con URL malformada → 400	Aprobado
PCN-03	RF-01	`POST /videos/upload` con archivo MP4 válido → 201	Aprobado
PCN-04	RF-02	`POST /videos/upload` con archivo `.exe` → 415	Aprobado
PCN-05	RF-04	`GET /videos/:id/status` durante procesamiento → `GENERATING`	Aprobado
PCN-06	RF-04	`GET /videos/:id/status` al finalizar → `COMPLETED`	Aprobado
PCN-07	RF-06	`GET /videos/:id` al completar → los cuatro bloques presentes	Aprobado
PCN-08	RF-07	`POST /videos/:id/retry` con `{ types: [QUIZ] }` → regenera solo quiz	Aprobado
PCN-09	RF-08	`PATCH /videos/:id/content` → `hasManualEdits=true`	Aprobado
PCN-10	RNF-07	Acceso sin JWT → 401	Aprobado
PCN-11	RNF-07	Acceso con rol `STUDENT` a `POST /videos` → 403	Aprobado

Continúa en la página siguiente

(Continuación Tabla 31)

ID	Req.	Caso	Resultado
PCN-12	RF-12	`DELETE /videos/:id` → borrado lógico	Aprobado

3.12.3 Pruebas de integración

Las pruebas de integración se ejecutaron sobre un corpus de 28 videos educativos reales, seleccionados para cubrir distintas duraciones, condiciones de audio y los dos tipos de origen (URL de YouTube y archivo cargado por el docente). Para cada video se midieron la tasa de éxito del *pipeline* completo, el tiempo total de procesamiento, los *tokens* consumidos por intento y la tasa de éxito desagregada por tipo de bloque. Los datos se obtuvieron de forma automática a partir de la tabla *video_generation_attempts*.

Tabla 32: Resumen de pruebas de integración sobre el corpus completo

Métrica	Valor
Videos en el corpus	28
Videos completados (estado `COMPLETED`)	26
Tasa de éxito a nivel de video	92,86 %
Intentos totales de generación	36
Intentos exitosos	30
Tasa de éxito a nivel de intento	83,33 %
Duración media por video (s)	416,3
Duración mediana por video (s)	322,2
Tokens de entrada acumulados	279 179
Tokens de salida acumulados	88 027
Tokens promedio por intento	10 200

Tabla 33: Tasa de éxito por tipo de bloque sobre los videos completados. Se considera un video *completado* cuando el bloque correspondiente se generó satisfactoriamente, ya sea en el primer intento o tras una regeneración manual del docente.

Tipo de bloque	Videos completados	Con bloque generado	Éxito
Resumen (SUMMARY)	26	25	96,2 %
Flashcards	26	26	100,0 %
Cuestionario (QUIZ)	26	25	96,2 %
Glosario (GLOSSARY)	26	26	100,0 %

De los 36 intentos de generación, 6 terminaron en fallo. Cuatro de los fallos correspondieron a *timeouts* del proveedor LLM en videos de mayor duración, uno a un error de parseo de la respuesta del modelo recuperado por el bloque de validación con

`zod`, y el último a un error de petición *Bad Request* atribuible al proveedor. En los casos de *timeout* parcial, los bloques que sí se generaron se persistieron correctamente y la auditoría dejó constancia de los tipos completados y fallidos por intento. Sobre el conjunto de 26 videos, únicamente 2 requirieron una regeneración manual posterior, y en ambos casos los bloques regenerados fueron precisamente el resumen y el cuestionario, lo que explica que estos dos tipos cierren con una tasa final de 96,2 % mientras que *flashcards* y glosario alcanzan el 100 %.

3.13 Despliegue

El sistema se despliega mediante Docker Compose con el archivo *docker-compose-prod.yaml*. El *compose* de producción orquesta tres servicios: el API NestJS, el *worker* NestJS y Redis, conectados sobre una red *bridge* compartida (`proxy-network`). El API y el *worker* comparten la imagen publicada en GitHub Container Registry (`ghcr.io/chu2409/edu-assistant-api:latest`); el contenedor del API ejecuta `npx prisma migrate deploy && node dist/src/main` y el del *worker* `npx prisma migrate deploy && node dist/src/worker` al iniciar. PostgreSQL con `pgvector` se utiliza como servicio externo gestionado, fuera del *compose*. El binario de `nodejs-whisper` se compila dentro de la imagen Docker durante la etapa *build*. El API expone el puerto `$PORT:-3010`; el *worker* y Redis quedan accesibles únicamente dentro de la red interna. Redis utiliza un volumen con nombre (`redis-data`) para persistir su estado y el API monta un volumen de host (`./uploads`) para los archivos de video cargados.

El comando de despliegue es:

```
1 docker compose --env-file .env.production -f docker-compose-prod.yaml up -d
```

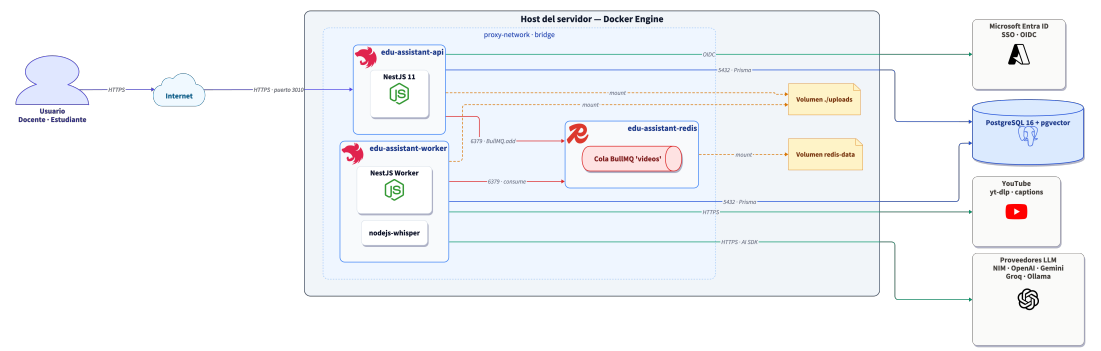

Figura 19: Diagrama de despliegue del sistema

3.13.1 Servidor de despliegue

El servidor que aloja el sistema en producción es gestionado por la Dirección de Tecnologías de la Información y Comunicación (DTIC) de la Universidad Técnica de Ambato, mediante solicitud canalizada por la Dirección de Investigación y Desarrollo (DIDE). El modelo Whisper *base* se ejecuta sobre la CPU del servidor sin requerir GPU dedicada. La latencia crece de forma lineal con la duración del video, lo que justifica la ejecución asíncrona del *pipeline* y la escalabilidad horizontal de los *workers*.

3.13.2 Configuración del entorno

La configuración se controla mediante variables de entorno declaradas en un archivo `.env.production` y validadas al arranque del proceso mediante un esquema Joi en `src/core/config/constants/index.ts`. Esta validación garantiza que el servicio inicia solo con configuración consistente: por ejemplo, `JWT_SECRET` exige al menos 30 caracteres, `VIDEO_AI_PROVIDER` debe pertenecer al conjunto {`groq`, `openai`, `google`, `nvidia`, `ollama`} y `WHISPER_MODEL` a {`tiny`, `base`, `small`, `medium`, `large-v3`}. La Tabla 34 resume las variables principales.

Tabla 34: Variables de entorno principales del sistema

Variable	**Propósito**
`DATABASE_URL`	URL de conexión a PostgreSQL con *pgvector* activado.
`REDIS_HOST`, `REDIS_PORT`	Host y puerto del Redis que actúa como *broker* de BullMQ.
`VIDEO_AI_PROVIDER`	Proveedor activo de LLM: `nvidia`, `groq`, `openai`, `google` u `ollama`.
`VIDEO_AI_MODEL`	Nombre del modelo dentro del proveedor seleccionado.
`VIDEO_AI_REQUEST_TIMEOUT`	*Timeout* máximo por agente.
`WHISPER_MODEL`	Variante de Whisper a utilizar.
`WHISPER_MODELS_DIR`	Directorio persistente de modelos Whisper.
`NVIDIA_API_KEY`, `GROQ_API_KEY`, `GOOGLE_GENERATIVE_AI_API_KEY`	Credenciales opcionales de los proveedores de LLM.
`JWT_SECRET`	Secreto de firma de los *tokens* JWT emitidos tras el SSO de Microsoft.

3.13.3 Repositorios del proyecto

El código fuente está disponible en GitHub:

- Backend (API y *worker* NestJS): `https://github.com/Chu2409/edu-assistant-api`
- Frontend (Vue 3): `https://github.com/Emi1213/edu-assistant`

3.14 Implantación y evaluación experimental

3.14.1 Plan de implantación

La implantación se planificó como un piloto controlado cuyo subgrupo evaluador son los estudiantes con preferencia de aprendizaje lectoescritora. El ámbito de aplicación fueron los cursos de los docentes colaboradores de la UTA, seleccionados por muestreo no probabilístico por conveniencia: el Programa Regular de Inglés del Centro de Idiomas y el curso de Programación Orientada a Objetos de la carrera de Software. El plan se organizó en cuatro fases secuenciales: preparación del entorno técnico, aplicación del cuestionario VARK y conformación del subgrupo evaluador, una sesión de uso del sistema por curso sobre material real de cada módulo y cierre con la aplicación del cuestionario SUS extendido.

3.14.2 Indicadores cuantitativos del sistema durante el piloto

Durante el piloto se registraron indicadores operativos del sistema sobre los videos procesados. Los valores reportados corresponden a las pruebas realizadas en el sistema.

La biblioteca *youtube-transcript-plus* aprovecha los protocolos públicos del cliente de YouTube para recuperar subtítulos cuando el video los expone. Cuando los subtítulos no están disponibles, ya sea en videos sin subtitulado autogenerado o con subtítulos restringidos, el sistema descarga el audio con `yt-dlp` y delega la transcripción a Whisper local. En el corpus de 28 videos, 20 fueron resueltos por la vía de subtítulos nativos (71,4 %) y 8 requirieron transcripción local (28,6 %).

Tabla 35: Plan de implantación del sistema

Fase	**Actividades**	**Duración**
Preparación	Provisionar el servidor con DTIC, configurar `docker compose`, validar conectividad a proveedores de LLM, cargar los modelos Whisper, registrar en el sistema los videos de cada módulo.	1 semana
Aplicación VARK	Difusión del formulario VARK entre los estudiantes de los cursos participantes, recepción de respuestas, cálculo de puntajes por modalidad e identificación del subgrupo con preferencia lectoescritora.	1 semana
Sesiones de uso	Una sesión guiada por curso, de aproximadamente 60 minutos cada una (Programa Regular de Inglés y Programación Orientada a Objetos): presentación del sistema, consumo de los videos de cada módulo y revisión de los cuatro materiales generados.	2 sesiones
Cierre con SUS	Aplicación del cuestionario SUS extendido (con bloque VARK y pregunta abierta de retroalimentación) al concluir cada sesión.	Mismo día

Tabla 36: Indicadores operativos del sistema durante el piloto

Indicador	**Valor**
Videos procesados durante la validación	28
Videos resueltos con subtítulos nativos de YouTube	20
Videos que requirieron Whisper local (archivo)	8
Idioma inglés detectado	16
Idioma español detectado	10
Idioma no determinado	2
Duración total del corpus (min)	118,0
Duración media por video (s)	416,3
Duración mediana por video (s)	322,2

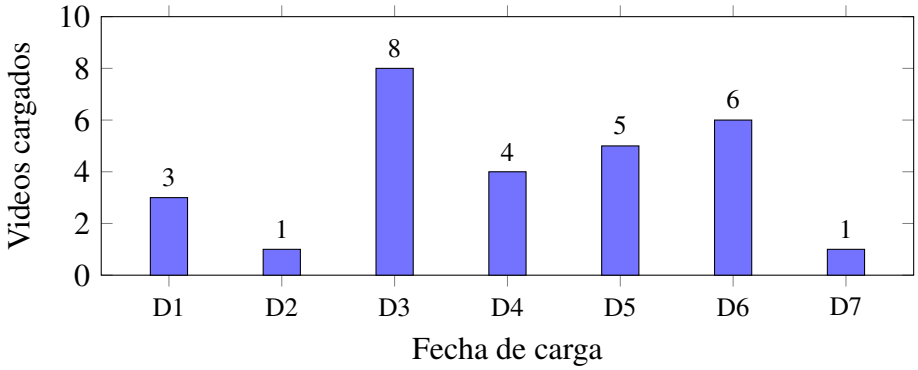

Figura 20: Adopción temporal: cantidad de videos cargados al sistema en cada una de las siete jornadas con actividad registrada (D1–D7).

La Figura 20 muestra que la carga se concentró en dos ventanas de actividad, lo cual se corresponde con las jornadas de preparación del corpus y la sesión final con usuarios.

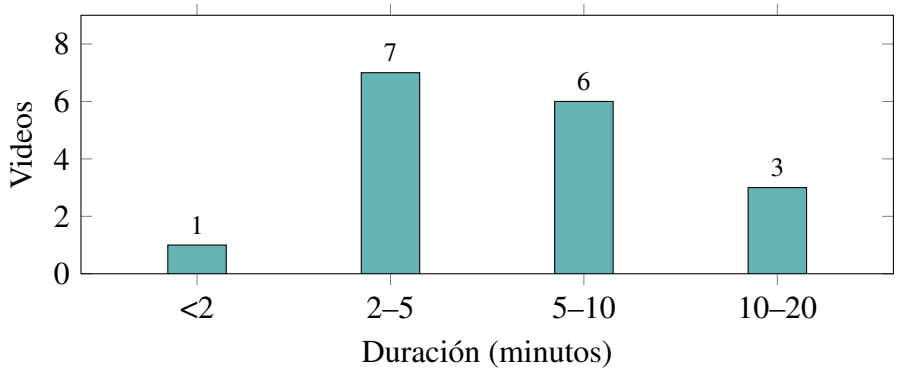

Figura 21: Distribución por duración de los 17 videos con metadato `duration_seconds` registrado.

La Figura 21 confirma que el grueso del corpus se ubica entre 2 y 10 minutos por pieza, rango habitual para material de estudio breve. Para 11 de los 28 videos el metadato de duración no quedó registrado por la fuente original, por lo que la distribución se reporta sobre los 17 casos con dato disponible.

3.15 Resultados por objetivo específico

3.15.1 Objetivo Específico 1 — Determinación de requerimientos funcionales

Se materializa en el catálogo de requerimientos presentado en las Tablas 27 y 28. La Tabla 37 resume el grado de implementación, verificable por inspección del código en los repositorios del proyecto.

Tabla 37: Verificación de requerimientos funcionales y no funcionales

Categoría	**Definidos**	**Verificados**	**% verificación**
Requerimientos funcionales (RF)	12	12	100,0 %
Requerimientos no funcionales (RNF)	10	8	80,0 %
Total	**22**	**20**	**90,9 %**

Los dos RNFs no verificados son RNF-09 (cobertura de pruebas automatizadas) y RNF-10 (internacionalización). RNF-09 quedó cubierto en la práctica con pruebas funcionales manuales aplicadas durante el desarrollo (Tablas 30 y 31); la implementación de la suite automatizada con cobertura mínima del 60 % queda como trabajo futuro. RNF-10 no se priorizó porque la población objetivo del piloto opera en español; el soporte multilenguaje del frontend queda igualmente pendiente para una fase posterior.

Cada requerimiento se derivó del cruce entre las necesidades identificadas mediante la encuesta aplicada a estudiantes y las capacidades técnicas revisadas durante el levantamiento conceptual del trabajo. La Tabla 38 resume la trazabilidad entre la evidencia empírica y los requerimientos resultantes.

Tabla 38: Matriz de trazabilidad — evidencia empírica, requerimientos y casos de uso

Hallazgo empírico	**Descripción**	**Reqs.**	**Casos de uso**
78,29 % sin material de síntesis	No disponen de un resumen tras ver un video (P5)	RF-06	CU-04, CU-12
79,84 % con dificultad media o alta	Para identificar ideas principales sin apoyo (P7)	RF-06	CU-12, CU-15
79,85 % revisa videos repetidamente	Tiempo perdido volviendo a ver el video (P8)	RF-06, RF-10	CU-13, CU-14
Dependencia de videos como insumo	Videos como recurso principal de estudio	RF-01, RF-05	CU-02
Expectativas sobre IA generativa	Disposición favorable a usar IA para generar materiales	RF-06, RF-07	CU-04, CU-09

3.15.2 Objetivo Específico 2 — Implementación del sistema

Se evidencia mediante el sistema construido, desplegado e integrado en NousAI. La Tabla 39 consolida las métricas de salida del sistema obtenidas a partir de las pruebas realizadas.

Tabla 39: Indicadores cuantitativos del sistema implementado

Indicador	**Valor**
Videos procesados (corpus de validación)	28
Tasa de éxito a nivel de video (completados / totales)	92,86 %
Tasa de éxito a nivel de intento de generación	83,33 %
Tokens de entrada acumulados	279 179
Tokens de salida acumulados	88 027
Tokens promedio por intento	10 200
Casos de prueba de caja blanca aprobados	10 / 10
Casos de prueba de caja negra aprobados	12 / 12

La distinción entre tasa de éxito a nivel de video (92,86 %) y a nivel de intento de generación (83,33 %) es relevante: la primera mide cuántos videos llegaron a estado `COMPLETED`, mientras que la segunda contabiliza los 36 intentos individuales de los agentes de generación sobre el corpus. Algunos videos requirieron más de un intento por agente antes de obtener el bloque correspondiente.

Cobertura de objetos de aprendizaje por tipo de bloque

El sistema genera cuatro tipos de bloque por video procesado: *summary*, *flashcards*, *quiz* y *glossary*. La Figura 22 reporta cuántos videos completados cuentan efectivamente con cada tipo de bloque.

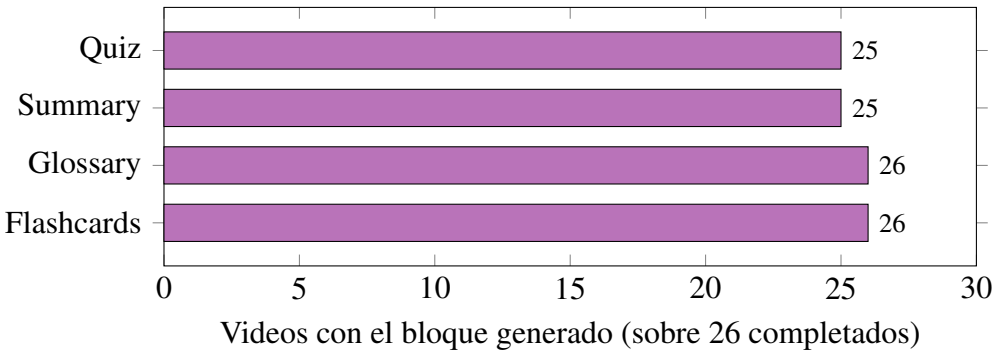

Figura 22: Cobertura de objetos de aprendizaje por tipo de bloque.

Los agentes *flashcards* y *glossary* alcanzaron cobertura total sobre los videos completados (26/26). Los agentes *summary* y *quiz* fallaron en un caso cada uno, lo cual coincide con los patrones de fallo descritos más adelante.

Rendimiento del pipeline por modelo de lenguaje

Durante la validación se contrastó el desempeño de tres modelos de lenguaje servidos por NVIDIA NIM. La Tabla 40 presenta el número de intentos, latencias y consumo de tokens por modelo.

Tabla 40: Rendimiento del pipeline por modelo de lenguaje

Modelo	**Intentos**	t_{p50} **(ms)**	t_{p95} **(ms)**	**Tok. in**	**Tok. out**
z-ai/glm4.7	24	80 872	150 464	5 563	1 559
openai/gpt-oss-120b	11	44 756	196 221	13 020	4 458
meta/llama-3.3-70b-instruct	1	33 123	33 123	2 449	1 580

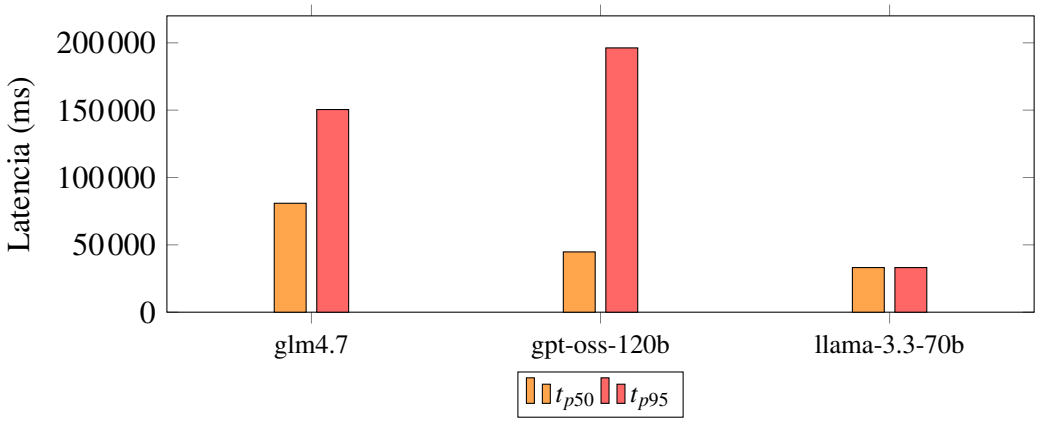

Figura 23: Latencia mediana y percentil 95 por modelo de lenguaje.

El modelo `z-ai/glm4.7` concentró el 67 % de los intentos (24/36) y exhibió la mediana de latencia más estable. El modelo `gpt-oss-120b` mostró menor mediana pero mayor dispersión, con un t_{p95} cercano a los 200 segundos y un consumo de tokens de entrada 2,3 veces superior. El modelo `llama-3.3-70b-instruct` se invocó en un único intento y se reporta a título informativo.

Eficiencia del pipeline frente a la duración del video

Para los 17 videos con metadato de duración, se contrastó el tiempo acumulado de pipeline (suma de `processing_time_ms` sobre todos los intentos) contra la duración del video fuente. La Figura 24 muestra el resultado.

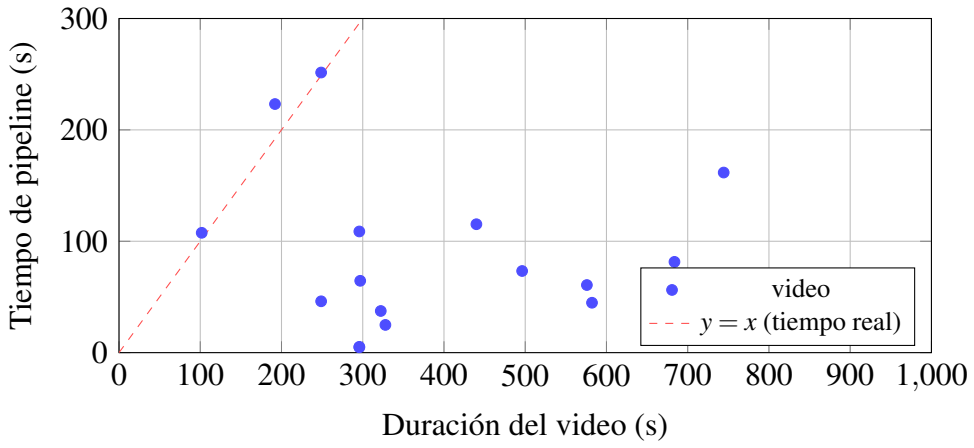

Figura 24: Tiempo total de pipeline frente a duración del video. La línea punteada marca la equivalencia con tiempo real.

Trece de los diecisiete casos (76,5 %) procesan el video en menos del 40 % de su duración. Solo dos casos superan el tiempo real, ambos correspondientes a videos breves donde el tiempo fijo de inicialización del pipeline domina sobre el procesamiento. Este patrón confirma que el costo dominante del sistema no escala linealmente con la duración del audio.

Patrones de fallo observados

De los 36 intentos de generación, 6 terminaron en fallo. La Figura 25 resume la distribución por categoría.

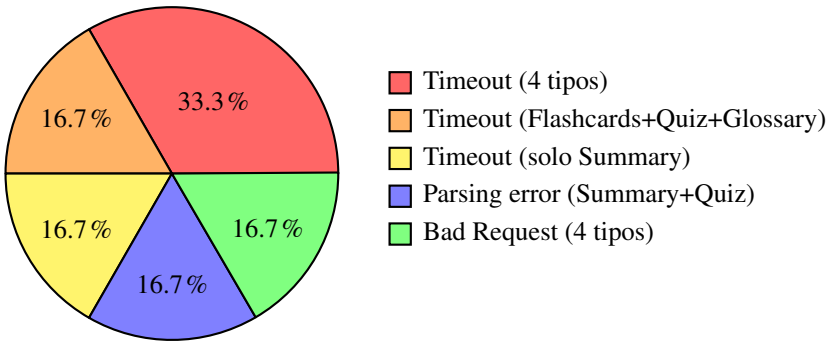

Figura 25: Distribución de los 6 intentos fallidos por categoría de error.

Cuatro de los seis fallos corresponden a *timeouts* del proveedor de LLM, lo cual concuerda con la dispersión observada en t_{p95} para `gpt-oss-120b`. Un caso correspondió a un error de parseo de la respuesta del modelo, gestionado por el bloque de validación con `zod` previsto en la implementación. El sexto fallo fue un error de petición (*Bad Request*) atribuible al proveedor.

Consumo agregado de tokens

El corpus de 28 videos consumió un total de 367 206 *tokens* (279 179 de entrada y 88 027 de salida), con un promedio de 10 200 *tokens* por intento de generación. La proporción de tokens de entrada respecto a los de salida (3,17:1) refleja el patrón habitual del sistema: las transcripciones largas se reducen a bloques estructurados más compactos.

La aplicabilidad de la solución se demuestra adicionalmente por su integración satisfactoria en NousAI: el resultado de este trabajo de titulación opera como parte de la plataforma educativa institucional.

Validez del contenido generado

Además de las métricas del *pipeline*, se aplicó una encuesta de validez del contenido producido por el sistema, descrita en el Capítulo II. La encuesta se distribuyó a los estudiantes que cursan o han cursado el curso de Programación Orientada a Objetos (86 convocados en total) y se obtuvieron 67 respuestas válidas, con un 77,9 % de participación. Las valoraciones se emitieron en escala Likert de 1 a 5 sobre los cuatro tipos de bloque que el sistema produce para los videos del temario.

Tabla 41: Validez global del contenido generado por tipo de bloque ($n = 67$)

Tipo de bloque	**Promedio**	**Desv. estándar**
Resumen	4,46	0,68
Flashcards	4,48	0,76
Cuestionario	4,46	0,65
Glosario	4,30	0,79

Los cuatro tipos de bloque se ubican por encima de 4,30 sobre 5. El mejor valorado fue *flashcards* (4,48) y el más bajo el glosario (4,30). La diferencia entre el mejor y el peor es menor a 0,2 puntos, por lo que la calidad percibida del material es similar entre los cuatro bloques.

La encuesta también pidió a cada estudiante identificar el video del temario que le resultó más útil y el que le resultó menos útil, y emitir valoraciones separadas para los cuatro bloques sobre cada uno de esos dos videos. La Tabla 42 resume el resultado.

Tabla 42: Validez del contenido en los videos extremos (más útil y menos útil) según los estudiantes encuestados ($n = 67$)

Tipo de bloque	**Video más útil**	**Video menos útil**
Resumen	4,61	3,73
Flashcards	4,60	3,99
Cuestionario	4,60	4,09
Glosario	4,42	3,96

En el video calificado como más útil, los cuatro bloques superan el 4,40 sobre 5. En el video calificado como menos útil, las valoraciones bajan a un rango entre 3,73 y 4,09. La caída más fuerte se da en el resumen, lo que sugiere que la calidad de este bloque depende del video de origen más que los otros tres.

La Tabla 43 muestra qué videos eligieron los estudiantes como más útil y como menos útil.

Tabla 43: Distribución de selecciones de los videos extremos ($n = 67$)

Video	**Más útil**	**Menos útil**
Herencia	24	7
Polimorfismo	20	27
Definición de clase	14	12
Métodos estáticos	9	21

El video sobre Herencia fue el más elegido como más útil (24 de 67) y también el menos elegido como menos útil (7 de 67). En el otro extremo, Polimorfismo fue el

más elegido como menos útil (27 de 67), aunque también recibió 20 selecciones como más útil. Esta distribución muestra que la calidad del material que entrega el sistema no es la misma para todos los videos. Los temas con una estructura más clara, como Herencia, generan material mejor valorado, mientras que los temas con más matices, como Polimorfismo, reciben opiniones más divididas.

En conjunto, la tasa de éxito del *pipeline* sobre el corpus de validación (92,86 %) y la valoración de los estudiantes sobre el material producido (entre 4,30 y 4,48 a nivel global) muestran que el sistema implementado para el segundo objetivo específico cumple las dos partes de la transformación: produce los materiales de forma estable y los materiales producidos son considerados válidos por los usuarios.

3.15.3 Objetivo Específico 3 — Evaluación de aceptación

La evaluación de aceptación se realizó con el cuestionario SUS aplicado, mediante muestreo no probabilístico por conveniencia, a estudiantes de los cursos de los docentes colaboradores de la UTA, esto es, el Programa Regular de Inglés del Centro de Idiomas y el curso de Programación Orientada a Objetos de la carrera de Software, tras una sesión de uso del sistema. El subgrupo evaluador del objetivo, definido por la modalidad VARK Lectura-escritura, se identificó mediante el filtrado VARK aplicado sobre esa muestra. La recolección concluyó con $n = 94$ respuestas válidas sobre una población combinada de 151 estudiantes. A continuación se presentan la distribución de perfiles VARK, los resultados del cuestionario SUS, su cruce por modalidad y la retroalimentación cualitativa recogida.

Distribución VARK

Tabla 44: Distribución de perfiles VARK identificados en la muestra ($n = 94$)

Perfil	**Frecuencia**	**Porcentaje**
Lectura-escritura	42	44,7 %
Visual	20	21,3 %
Multimodal	17	18,1 %
Kinestésico	11	11,7 %
Auditivo	4	4,3 %
Total respuestas válidas	**94**	**100,0 %**
Subgrupo evaluador R/W (criterio estricto)	42	44,7 %

La modalidad Lectura-escritura concentra el 44,7 % de las respuestas, y constituye el subgrupo evaluador del tercer objetivo específico ($n = 42$). El perfil Multimodal, definido en el Capítulo II como referencia y no como parte del subgrupo R/W, alcanza el 18,1 % de la muestra.

Resultados SUS

Tabla 45: Resultados del cuestionario SUS

Indicador	**Muestra completa**	**Subgrupo R/W**
Número de estudiantes	94	42
Puntaje SUS promedio	81,65	82,26
Desviación estándar	15,15	14,26
Mediana	85,0	86,25
Puntaje mínimo	25,0	47,5
Puntaje máximo	100,0	100,0
Estudiantes con puntaje ≥ 68	79 (84,0 %)	35 (83,3 %)

Sobre el subgrupo evaluador R/W el puntaje promedio se ubica en 82,26, valor que según la escala de adjetivos de Bangor sitúa la usabilidad del sistema en el rango excelente, por encima del umbral de 68 puntos que la literatura asocia al límite de aceptabilidad [39]. Sobre la muestra completa, sin segmentar por modalidad, el promedio queda en 81,65 puntos. La diferencia entre ambos valores es de menos de un punto, lo que indica que el sistema no es percibido como significativamente más usable por el subgrupo al que apunta que por el resto de los perfiles VARK. La mediana del subgrupo R/W es 86,25, que supera al promedio, lo que refleja que la mayoría de los estudiantes del subgrupo se ubicaron en el tramo alto de la escala.

Cruce VARK $\times$ SUS

El bloque Q-VARK-1 permite segmentar el puntaje SUS por modalidad VARK declarada. La Tabla 46 muestra el desglose y la Figura 26 lo representa frente al umbral de 68 puntos.

Tabla 46: Puntaje SUS promedio por modalidad VARK declarada ($n = 94$)

Modalidad VARK	**N**	**SUS promedio**
Visual	20	83,5
Multimodal	17	82,6
Lectura-escritura	42	82,3
Kinestésico	11	80,9
Auditivo	4	63,8

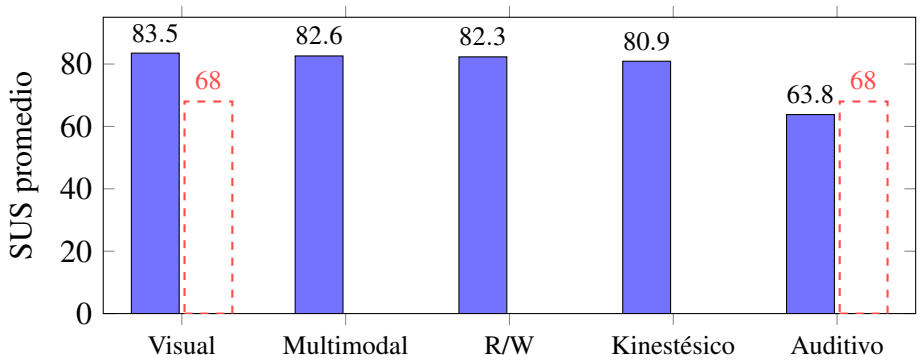

Figura 26: Puntaje SUS promedio segmentado por modalidad VARK declarada. La línea punteada marca el umbral interpretativo de 68 puntos descrito en la literatura sobre SUS.

Cuatro de los cinco perfiles se ubican en un rango estrecho, entre 80,9 y 83,5 puntos. El perfil Auditivo se aparta del resto con 63,8 puntos, pero el tamaño de su subgrupo ($n = 4$) no permite extraer una conclusión sobre ese resultado. Dentro del rango principal, el subgrupo evaluador R/W ($n = 42$, 82,3) queda al mismo nivel que los perfiles Visual (83,5) y Multimodal (82,6), lo que sostiene la pertinencia del sistema para el público al que apunta sin que ello implique pérdida de usabilidad para los otros perfiles.

El perfil Multimodal, que la literatura señala como potencialmente incluyente de la modalidad R/W [21], registra un puntaje de 82,6, prácticamente igual al del subgrupo R/W estricto (82,3). Este dato se reporta como referencia para apoyar la decisión metodológica adoptada en el Capítulo II: dado que el subgrupo R/W estricto resultó suficiente para sostener el análisis, no se incorpora al perfil Multimodal al subgrupo evaluador, pero su valoración cercana respalda la idea de que ambos perfiles responden de manera similar al sistema.

Retroalimentación cualitativa

El campo abierto Q-FB-1 recogió 58 comentarios sobre 94 respuestas (61,7 %). Las observaciones se agruparon en dos grandes categorías: valoraciones positivas sobre la experiencia con el sistema y sugerencias de mejora. La Tabla 47 resume las menciones más frecuentes en cada categoría.

Tabla 47: Comentarios abiertos agrupados por tema ($n = 58$)

Tema	**Menciones**
Valoraciones positivas	
Interfaz intuitiva y fácil de navegar	5
Integración entre las funciones del sistema	5
Claridad de los temas explicados	5
Herramienta útil para reforzar el aprendizaje	5
Glosario útil para palabras nuevas	4
Feedback automático apoya el estudio autónomo	3
Videos bien estructurados	3
Navegación clara y sencilla	3
Organización del contenido por niveles	2
Sugerencias de mejora	
Mejorar la velocidad de carga	6
Habilitar una versión sin conexión	4
Agregar más ejercicios prácticos	5
Aumentar la visibilidad de los botones en móvil	3
Sin sugerencias	
Todo funcionó correctamente	4

Las valoraciones positivas se concentran en la facilidad de uso, la claridad del contenido y la utilidad del glosario y del *feedback* automático para el estudio. Las sugerencias de mejora apuntan a tres puntos concretos: la velocidad de carga del sistema en algunos casos, la posibilidad de una versión sin conexión para zonas con mala cobertura y la visibilidad de los controles cuando se accede desde teléfono. Adicionalmente, una solicitud recurrente fue la de incorporar más ejercicios prácticos al material generado. Ningún comentario reportó errores que invalidaran el contenido producido.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones obtenidas a partir del desarrollo del trabajo y de los resultados de la implantación piloto. Adicionalmente, se incluyen recomendaciones derivadas de las limitaciones observadas durante la implantación piloto y de la retroalimentación recogida del subgrupo evaluador. Las conclusiones se organizan en torno a los tres objetivos específicos y al objetivo general del trabajo.

4.1 Conclusiones

Con el desarrollo del presente trabajo de titulación se han cumplido los objetivos planteados. Se transformaron videos educativos en materiales de aprendizaje estructurados mediante modelos de lenguaje de gran escala y reconocimiento automático de voz. El desarrollo del trabajo derivó en un sistema que, a partir de un video, genera de forma automática cuatro materiales de aprendizaje, a saber, un resumen, un glosario, *flashcards* y un cuestionario interactivo, y que se integró como un nuevo módulo dentro de la plataforma educativa institucional NousAI.

El diagnóstico aplicado a 129 estudiantes de la Universidad Técnica de Ambato evidenció tres carencias concretas en el estudio mediante video: el 78,29 % no contaba con un material de síntesis tras ver el video, el 79,84 % reportó dificultad media o alta para identificar las ideas principales sin apoyo y el 79,85 % revisaba el mismo video varias veces. A partir del cruce entre estas necesidades y las capacidades técnicas revisadas se definió un catálogo IEEE 830 de doce requerimientos funcionales y diez no funcionales, cada uno trazado mediante una matriz que vincula el hallazgo empírico con el requerimiento y el caso de uso asociado. La verificación posterior cubrió 20 de los 22 requerimientos (90,9 %): la totalidad de los funcionales y ocho de los diez no funcionales, quedando pendientes la suite de pruebas automatizadas y el soporte multilenguaje del *frontend*.

El sistema se construyó sobre NestJS, Prisma y BullMQ en el *backend* y Vue 3.5 con TanStack Vue Query en el *frontend*, integrado a NousAI mediante el modelo compartido de *objeto de aprendizaje* y autenticación con Microsoft SSO. Sobre un corpus

de validación de 28 videos, el *pipeline* alcanzó una tasa de éxito del 92,86 % a nivel de video y del 83,33 % a nivel de intento de generación (30 de 36 intentos ejecutados). De los seis fallos registrados, cuatro fueron *timeouts* del proveedor de modelo de lenguaje y los dos restantes un error de parseo y una petición rechazada; todos quedaron contenidos por la separación entre el agente de transcripción y los agentes generativos, la validación por esquema de las salidas del modelo y el retroceso exponencial configurado en BullMQ, sin interrumpir el resto de la cadena. La calidad del contenido producido se verificó con una encuesta a 67 estudiantes del curso de Programación Orientada a Objetos: los cuatro tipos de bloque, es decir, el resumen, las *flashcards*, el cuestionario y el glosario, obtuvieron valoraciones promedio entre 4,30 y 4,48 sobre una escala de 1 a 5, con una diferencia menor a 0,2 puntos entre el mejor y el peor valorado.

La aceptación de la solución se evaluó con el cuestionario SUS aplicado, mediante muestreo no probabilístico por conveniencia, a 94 estudiantes de la Universidad Técnica de Ambato provenientes de los cursos de los docentes colaboradores, esto es, el Programa Regular de Inglés del Centro de Idiomas y el curso de Programación Orientada a Objetos de la carrera de Software, con identificación previa del perfil de aprendizaje de cada participante mediante el cuestionario VARK. Sobre el subgrupo evaluador con preferencia lectoescritora ($n = 42$, criterio estricto), el puntaje SUS promedio se ubicó en 82,26 puntos sobre 100, valor que según la escala de adjetivos de Bangor y otros autores ubica la usabilidad del sistema en el rango excelente, por encima del umbral de aceptabilidad de 68 puntos, con un 83,3 % de estudiantes por encima de dicho umbral. El cruce por modalidad VARK mostró que la usabilidad se mantuvo pareja entre perfiles, con valores entre 80,9 y 83,5 puntos en cuatro de los cinco grupos, lo que indica que el sistema no penaliza a los estudiantes de otras modalidades pese a estar orientado al perfil lectoescritor.

4.2 Recomendaciones

Migrar la inferencia del modelo de lenguaje a ejecución local con Ollama cuando la Dirección de Investigación y Desarrollo (DIDE) disponga de una GPU dedicada. La fábrica de proveedores ya contempla Ollama, de modo que el cambio se reduce a la variable de entorno `LLM_PROVIDER`. Ejecutar el modelo en el propio servidor elimina el costo por *token*, mantiene el contenido educativo dentro de la infraestructura institucional y suprime la dependencia de las cuotas y los *timeouts* de los proveedores en nube, que originaron cuatro de los seis fallos registrados durante el piloto.

Añadir al *pipeline* una etapa de verificación automática de la fidelidad del contenido generado. La validación por esquema con `zod` que hoy aplica el sistema garantiza la estructura de cada bloque, pero no comprueba que su contenido se ajuste a la transcripción de origen. Se recomienda incorporar un agente verificador que, antes de publicar, contraste cada material contra la transcripción y puntúe su fidelidad, de modo que las invenciones o desviaciones del modelo se detecten y se reintenten de forma automática reutilizando el mecanismo de reintentos ya configurado en BullMQ. Con ello, el control de calidad que en este trabajo se midió mediante encuesta quedaría incorporado al propio proceso de generación.

Extender la validación del sistema a otros campos de conocimiento dentro de la universidad. El piloto se aplicó sobre el curso de inglés del Centro de Idiomas FISEI, pero la cadena de transcripción y generación no es específica del idioma ni del dominio. Una validación con corpus de áreas distintas, por ejemplo asignaturas técnicas, jurídicas o de ciencias de la salud, permitiría verificar la calidad de los materiales producidos en contextos con vocabulario especializado y estructura discursiva diferente.

Incorporar nuevos tipos de objeto de aprendizaje aprovechando la arquitectura de agentes del sistema, que admite añadir un material con solo un agente y su esquema de validación. Conviene priorizar formatos que cubran otras modalidades: mapas conceptuales y diagramas para el perfil visual, ejercicios prácticos guiados para el kinestésico y casos de aplicación para reforzar la comprensión, en línea con la petición de más ejercicios prácticos recogida durante el piloto.

BIBLIOGRAFÍA

[1] W. Breslyn y A. Green, "Learning science with YouTube videos and the impacts of Covid-19," *Disciplinary and Interdisciplinary Science Education Research*, vol. 4, 2022. DOI: 10.1186/s43031-022-00051-4

[2] J. C.-C. Lu, "Using YouTube as an Effective Educational Tool to Improve Engineering Mathematics Teaching during the COVID-19 Pandemic," *Engineering Proceedings*, 2023. DOI: 10.3390/engproc2023038024

[3] J. Sweller, "Cognitive load theory and educational technology," *Educational Technology Research and Development*, vol. 68, págs. 1-16, 2020. DOI: 10.1007/s11423-019-09701-3

[4] J. C. Castro-Alonso, B. B. De Koning, L. Fiorella y F. Paas, "Five Strategies for Optimizing Instructional Materials: Instructor- and Learner-Managed Cognitive Load," *Educational Psychology Review*, vol. 33, págs. 1379-1407, 2021. DOI: 10.1007/s10648-021-09606-9

[5] Y.-C. Lin, T.-C. Liu y S. Kalyuga, "Strategies for facilitating processing of transient information in instructional videos by using learner control mechanisms," *Instructional Science*, vol. 50, págs. 863-877, 2022. DOI: 10.1007/s11251-022-09600-w

[6] S. Lee, "Educational Implications of VARK Learning Styles: Academic Performance and Pedagogical Preferences among Korean Pharmacy Students," *Indian Journal of Pharmaceutical Education and Research*, 2025. DOI: 10.5530/ijper.20250050

[7] E. El-Saftawy et al., "Influence of applying VARK learning styles on enhancing teaching skills: application of learning theories," *BMC Medical Education*, vol. 24, 2024. DOI: 10.1186/s12909-024-05979-x

[8] J. Cao et al., "A Comparative Analysis of Automatic Speech Recognition Errors in Small Group Classroom Discourse," en *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, 2023. DOI: 10.1145/3565472.3595606

[9] S. O. Russell et al., "What automatic speech recognition can and cannot do for conversational speech transcription," *Research Methods in Applied Linguistics*, 2024. DOI: 10.1016/j.rmal.2024.100163

[10] R. Southwell et al., "Challenges and Feasibility of Automatic Speech Recognition for Modeling Student Collaborative Discourse in Classrooms," en *Proceedings of the 15th International Conference on Educational Data Mining*, 2022, págs. 302-315. DOI: 10.5281/zenodo.6853109

[11] H. Gonzalez et al., "Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience," en *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, 2023, págs. 382-393. DOI: 10.18653/v1/2023.bea-1.31

[12] K. Kawamura y J. Rekimoto, "FastPerson: Enhancing Video-Based Learning through Video Summarization that Preserves Linguistic and Visual Contexts," en *Proceedings of the Augmented Humans International Conference 2024*, 2024. DOI: 10.1145/3652920.3652922

[13] S. Saha, F. Rahbari, F. Sadique, S. Velamakanni, M. Farooque y W. Rothwell, *Next-Gen Education: Enhancing AI for Microlearning*, 2025. DOI: 10.18260/1-2--56998

[14] L. Stavrinou, A. Constantinides, M. Belk, V. Vassiliou, F. Liarokapis y M. Constantinides, "The Reel Deal: Designing and Evaluating LLM-Generated Short-Form Educational Videos," en *Proceedings of the 3rd International Conference of the ACM Greek SIGCHI Chapter*, 2025. DOI: 10.1145/3749012.3749048

[15] S. Alrumiah y A. Al-Shargabi, "Educational Videos Subtitles' Summarization Using Latent Dirichlet Allocation and Length Enhancement," *Computers, Materials & Continua*, 2021. DOI: 10.32604/cmc.2022.021780

[16] Q. Yu, J. Gou, Y. Li, Z. Pi y J. Yang, "Introducing support for learner control: Temporal and organizational cues in instructional videos," *British Journal of Educational Technology*, vol. 55, n.º 3, págs. 933-956, 2024. DOI: 10.1111/bjet.13408

[17] D. Kueker y J. Moore, "Learning from screencast software tutorials: A comparison of cognitive load in dual and single-monitor learning environments," *Journal of Computer Assisted Learning*, vol. 40, n.º 1, págs. 118-135, 2023. DOI: 10.1111/jcal.12875

[18] C. Tarchi, S. Zaccoletti y L. Mason, "Learning from text, video, or subtitles: A comparative analysis," *Computers & Education*, vol. 160, pág. 104 034, 2020. DOI: 10.1016/j.compedu.2020.104034

[19] C.-Y. Mo, C. Wang, J. Dai y P. Jin, "Video Playback Speed Influence on Learning Effect From the Perspective of Personalized Adaptive Learning: A Study Based on Cognitive Load Theory," *Frontiers in Psychology*, vol. 13, 2022. DOI: 10.3389/fpsyg.2022.839982

[20] J. Zhu et al., "The impact of short videos on student performance in an online-flipped college engineering course," *Humanities & Social Sciences Communications*, vol. 9, 2022. DOI: 10.1057/s41599-022-01355-6

[21] S. Gangadharan, K. Mezeini, S. Gnanamuthu y K. Marshoudi, "The Relationship Between Preferred Learning Styles and Academic Achievement of Undergraduate Health Sciences Students Compared to Other Disciplines at a Middle Eastern University Utilizing the VARK Instrument," *Advances in Medical Education and Practice*, vol. 16, págs. 13-28, 2025. DOI: 10.2147/amep.s491487

[22] M. Yamada, M. Sekine y T. Tatsuse, "Paper-based versus digital-based learning among undergraduate medical, nursing and pharmaceutical students in Japan: a cross-sectional study," *BMJ Open*, vol. 14, 2024. DOI: 10.1136/bmjopen-2023-083344

[23] L. Yan et al., "Practical and ethical challenges of large language models in education: A systematic scoping review," *British Journal of Educational Technology*, vol. 55, n.º 1, págs. 90-112, 2024. DOI: 10.1111/bjet.13370

[24] I. C. Peláez-Sánchez, D. Velarde-Camaqui y L. D. Glasserman-Morales, "The impact of large language models on higher education: exploring the connection between AI and Education 4.0," *Frontiers in Education*, 2024. DOI: 10.3389/feduc.2024.1392091

[25] A. Abd-Alrazaq et al., "Large Language Models in Medical Education: Opportunities, Challenges, and Future Directions," *JMIR Medical Education*, vol. 9, 2023. DOI: 10.2196/48291

[26] Q. Huang, C. Lv, L. Lu y S. Tu, "Evaluating the Quality of AI-Generated Digital Educational Resources for University Teaching and Learning," *Systems*, vol. 13, n.º 3, pág. 174, 2025. DOI: 10.3390/systems13030174

[27] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey e I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," en *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, págs. 28 492-28 518. dirección: https://arxiv.org/abs/2212.04356

[28] R. Southwell et al., "Automatic Speech Recognition Tuned for Child Speech in the Classroom," en *ICASSP 2024 – 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024, págs. 12 291-12 295. DOI: 10.1109/icassp48485.2024.10447428

[29] S. Fergus, "Are undergraduate students studying smart? Insights into study strategies and habits across a programme of study," *Journal of University Teaching and Learning Practice*, vol. 19, n.º 2, 2022. DOI: 10.53761/1.19.2.8

[30] M. Ingebrigtsen, Å. O. Miland, J. Bastesen y R. G. Sæle, "Effective, Scalable, and Low Cost: The Use of Teacher-Made Digital Flashcards Improves Student Learning," *Applied Cognitive Psychology*, 2025. DOI: 10.1002/acp.70086

[31] S. F. Durrani, N. Yousuf, R. Ali, F. F. Musharraf, A. Hameed y H. A. Raza, "Effectiveness of spaced repetition for clinical problem solving amongst undergraduate medical students studying paediatrics in Pakistan," *BMC Medical Education*, vol. 24, 2024. DOI: 10.1186/s12909-024-05479-y

[32] B. F. T. Moreira, T. S. S. Pinto, D. S. V. Starling y A. Jaeger, "Retrieval Practice in Classroom Settings: A Review of Applied Research," *Frontiers in Education*, 2019. DOI: 10.3389/feduc.2019.00005

[33] C. Bego et al., "Single-paper meta-analyses of the effects of spaced retrieval practice in nine introductory STEM courses: is the glass half full or half empty?" *International Journal of STEM Education*, vol. 11, 2024. DOI: 10.1186/s40594-024-00468-5

[34] S. C. M. Donker, M. A. T. M. Vorstenbosch, M. J. T. Gerhardus y D. H. J. Thijssen, "Retrieval practice and spaced learning: preventing loss of knowledge in Dutch medical sciences students in an ecologically valid setting," *BMC Medical Education*, vol. 22, 2022. DOI: 10.1186/s12909-021-03075-y

[35] M. Waseem, P. Liang, M. Shahin, A. D. Salle y G. Márquez, "Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective," *Journal of Systems and Software*, vol. 182, pág. 111 061, 2021. DOI: 10.1016/j.jss.2021.111061

[36] A. Smirnov, "Modern Methods of Backend System Performance Optimization: Algorithmic, Architectural, and Infrastructural Aspects," *International Journal of Advanced Research in Science, Communication and Technology*, 2025. DOI: 10.48175/ijarsct-28528

[37] J. Brooke, "SUS: a quick and dirty usability scale," en *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester e I. L. McClelland, eds., London: Taylor & Francis, 1996, págs. 189-194.

[38] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *International Journal of Human–Computer Interaction*, vol. 34, n.º 7, págs. 577-590, 2018. DOI: 10.1080/10447318.2018.1455307

[39] A. Bangor, P. Kortum y J. Miller, “Determining what individual SUS scores mean: adding an adjective rating scale,” *Journal of Usability Studies*, vol. 4, n.º 3, págs. 114-123, 2009.

ANEXOS

ANEXO A: MANUAL DE USUARIO — *FEATURE* DE VIDEOS EN NOUSAI

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SOFTWARE

MANUAL DE USUARIO

Feature de transformación de videos educativos
en materiales de aprendizaje estructurados
dentro de la plataforma NousAI

AUTOR: José Sebastián Camino Moreno
TUTOR: Ing. Félix Oscar Fernández Peña, MSc, PhD.
PROYECTO: UTA-CONIN-2025-0261-R

Ambato – Ecuador
Julio – 2026

ÍNDICE DE CONTENIDOS DEL MANUAL

1. Presentación del manual

Este documento describe paso a paso el uso de la *feature* de videos educativos integrada en la plataforma NousAI de la Universidad Técnica de Ambato. El manual está pensado para acompañar tanto al docente que registra y publica material como al estudiante que lo consume. Su finalidad es servir de guía operativa una vez la plataforma se encuentra desplegada, sin requerir conocimientos previos sobre la arquitectura interna del sistema.

2. Alcance y destinatarios

La *feature* cubre el ciclo completo de transformación de un video educativo en materiales de aprendizaje estructurados: registro del video, transcripción automática, generación asistida por inteligencia artificial de resumen, glosario, *flashcards* y cuestionario de opción múltiple, edición manual, publicación y consumo por parte del estudiante.

Existen dos perfiles de usuario:

- **Docente.** Único rol con permisos de creación, edición, regeneración y publicación. Es el responsable pedagógico del contenido.
- **Estudiante.** Consume el material una vez publicado. No puede registrar videos, editar contenido ni solicitar regeneración.

3. Requisitos previos

- Cuenta institucional Microsoft 365 activa, emitida por la Universidad Técnica de Ambato.
- Navegador web actualizado (Chrome, Edge o Firefox en versiones vigentes).
- Conexión a Internet estable. Para la carga de archivos locales se recomienda un ancho de banda superior a 5 Mbps.
- Para el docente, el video a registrar debe estar disponible como enlace público de YouTube o como archivo MP4 de hasta 500 MB.

4. Acceso a la plataforma

El ingreso se realiza mediante el inicio de sesión único de Microsoft. El usuario pulsa el botón *Iniciar sesión con Microsoft*, introduce sus credenciales institucionales y autoriza el acceso. La plataforma reconoce el rol del usuario a partir del directorio institucional y redirige al panel correspondiente.

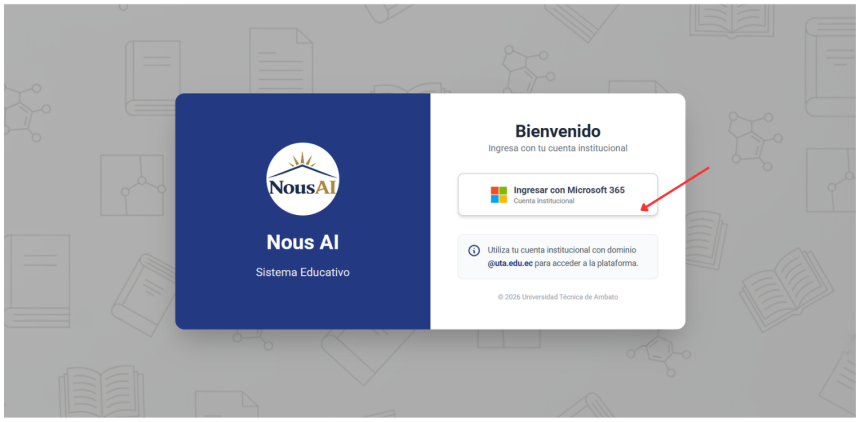

Figura 27: Pantalla de inicio de sesión con botón de autenticación Microsoft destacado mediante flecha.

5. Flujo del docente

5.1 Selección del módulo

Tras iniciar sesión, el docente visualiza el panel con la lista de módulos a su cargo. Cada módulo agrupa los objetos de aprendizaje del curso. Al pulsar sobre el módulo deseado se accede a su detalle.

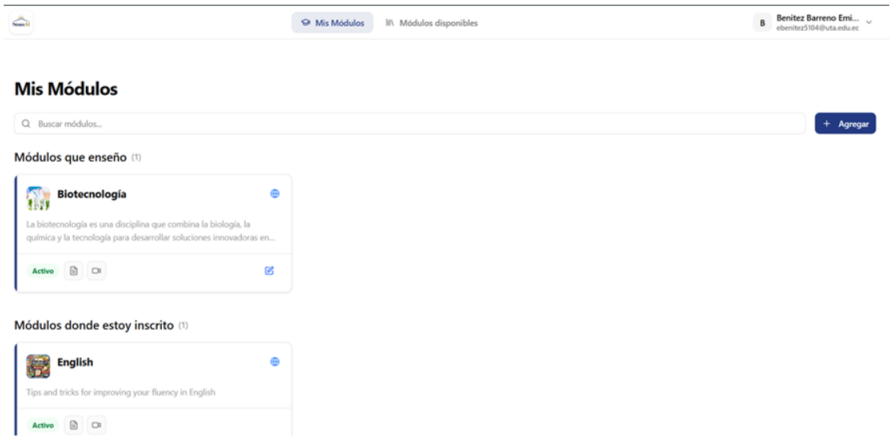

Figura 28: Panel del docente con la lista de módulos. La flecha señala la tarjeta del módulo seleccionado.

5.2 Registro de un nuevo video

Dentro del detalle del módulo, el docente pulsa el botón *Crear video*. Se abre un diálogo con dos pestañas que corresponden a las fuentes admitidas: enlace público de YouTube o archivo local MP4. El docente completa los datos requeridos (título, descripción opcional y fuente del video) y confirma. A partir de ese momento el sistema encola el video para procesamiento asíncrono.

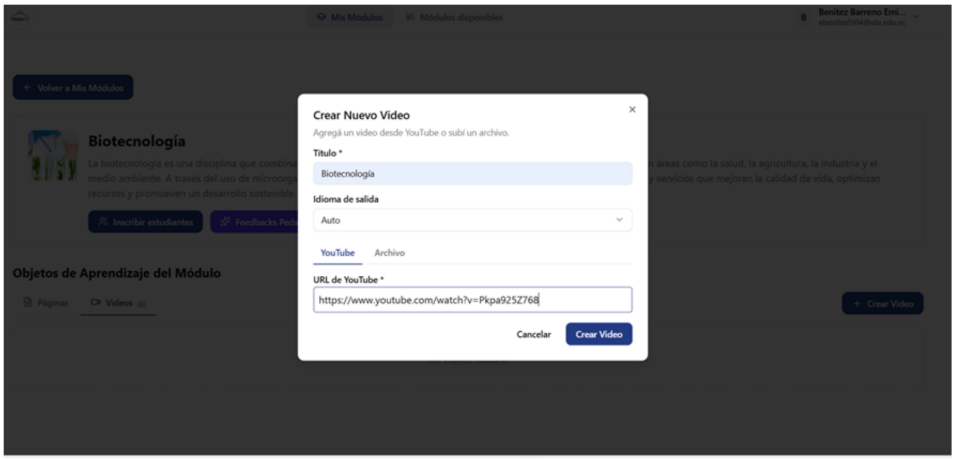

Figura 29: Diálogo de registro de video, pestaña YouTube. Las flechas indican el campo de URL y el botón *Confirmar*.

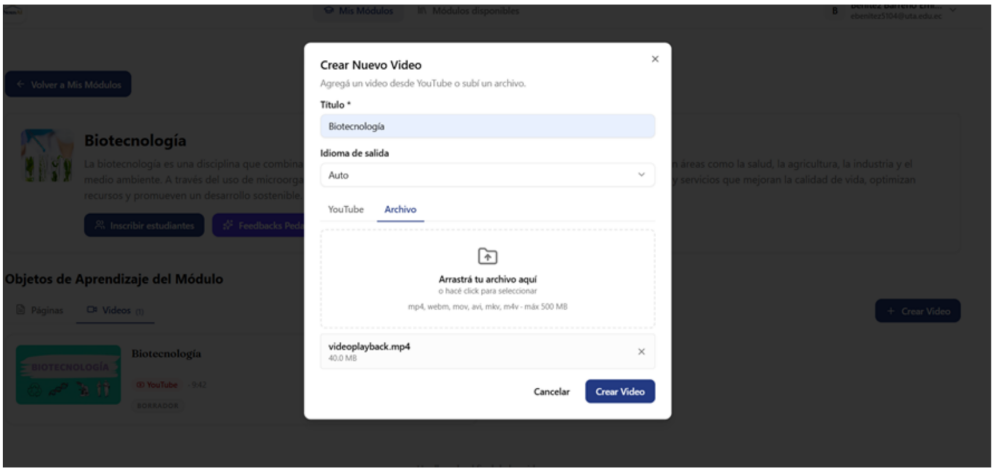

Figura 30: Diálogo de registro de video, pestaña carga local. La flecha señala el área de arrastre de archivo permitido.

5.3 Monitoreo del procesamiento

El video registrado aparece en la lista del módulo con un distintivo de estado. Los estados posibles son `queued` (en cola), `transcribing` (transcribiendo), `generating`

(generando materiales), `ready` (listo) y `failed` (con error). El docente puede recargar la vista o esperar la notificación automática cuando el procesamiento finaliza.

Figura 31: Lista de videos del módulo con sus distintivos de estado. La flecha resalta un video en estado `ready`.

5.4 Revisión de los materiales generados

Cuando el estado pasa a `ready`, el docente abre el detalle del video. La vista organiza los cuatro materiales en pestañas: resumen estructurado, *flashcards*, cuestionario de opción múltiple y glosario técnico. El docente navega entre las pestañas para revisar el contenido producido.

Figura 32: Vista de detalle del video con la barra de pestañas señalada por una flecha.

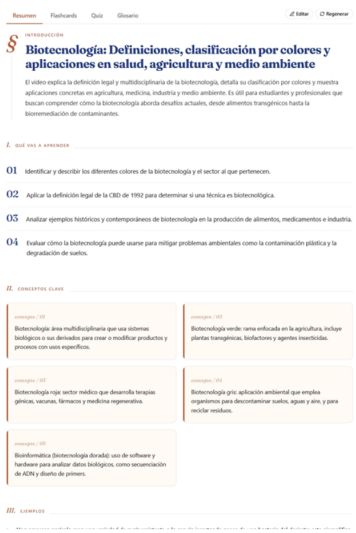

Figura 33: Pestaña de resumen estructurado, con secciones y conceptos clave.

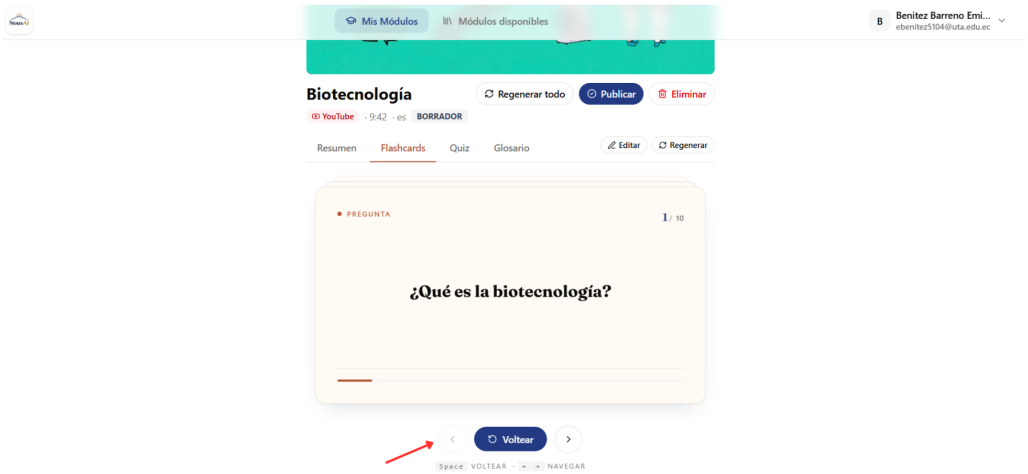

Figura 34: Pestaña de *flashcards* en modo revisión. La flecha indica el control para voltear la tarjeta.

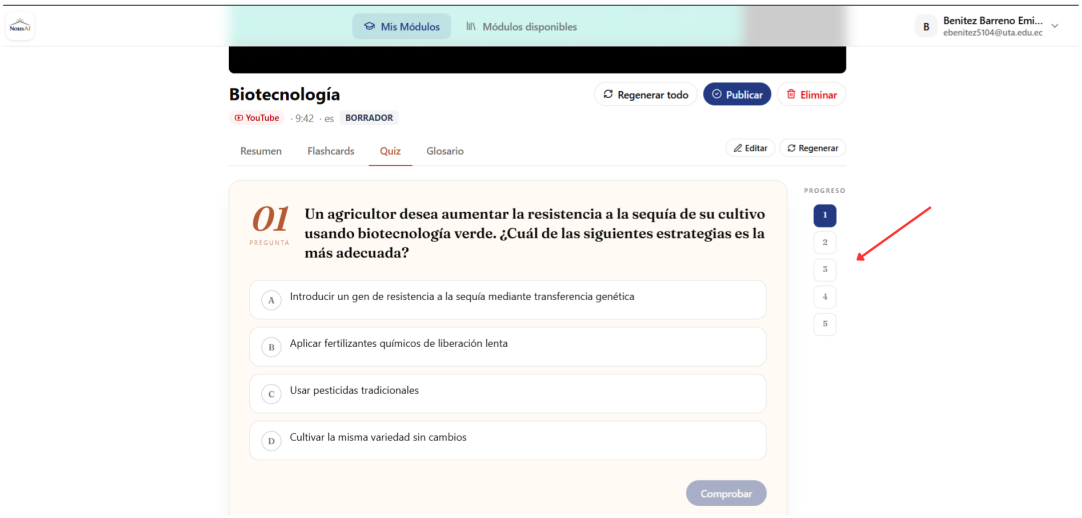

Figura 35: Pestaña de cuestionario de opción múltiple. La flecha señala una pregunta con sus alternativas.

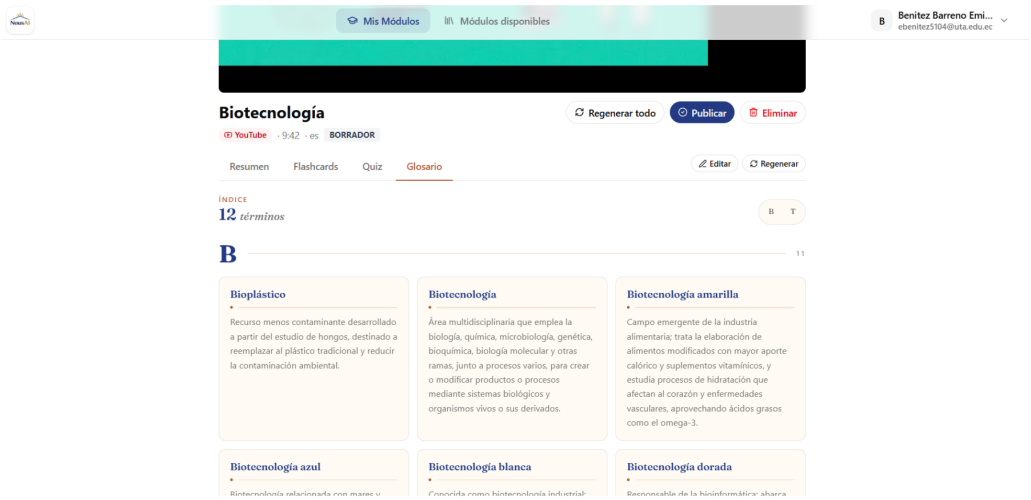

Figura 36: Pestaña de glosario técnico con términos y definiciones.

5.5 Edición del contenido

El docente puede modificar cualquiera de los materiales directamente desde la interfaz. Cada pestaña ofrece un editor enriquecido que permite ajustar texto, agregar o quitar elementos y reordenar bloques. Los cambios se persisten al guardar y el objeto de aprendizaje queda marcado como editado manualmente.

Figura 37: Editor enriquecido sobre el resumen. Las flechas indican la barra de herramientas y el botón *Guardar*.

5.6 Solicitud de regeneración

Si el resultado generado no resulta satisfactorio, el docente puede solicitar una nueva generación, total o por bloque. El diálogo de regeneración acepta una instrucción libre en lenguaje natural que el agente utiliza como guía. Esta instrucción puede indicar, por ejemplo, un nivel de detalle distinto o un enfoque pedagógico específico.

Figura 38: Diálogo de regeneración con campo de instrucción libre. La flecha señala el botón *Regenerar*.

5.7 Reintento ante fallos

Cuando el procesamiento termina en estado `failed` por causas transitorias (caída del proveedor de modelo, error de descarga, expiración de cuota), el docente dispone de un botón *Reintentar* que reinyecta el video en la cola sin necesidad de volver a registrarlo.

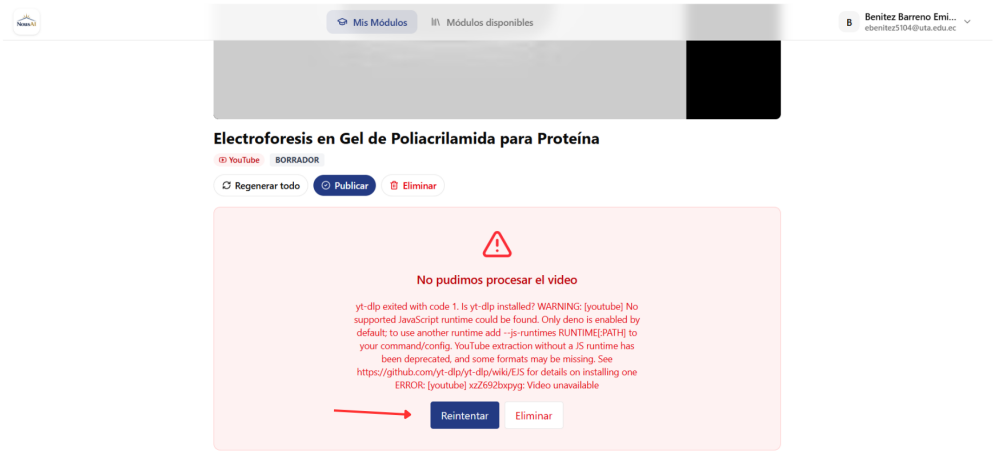

Figura 39: Video en estado `failed` con el botón *Reintentar* resaltado por una flecha.

5.8 Publicación del material

Una vez revisado, y editado si fuese necesario, el docente publica el objeto de aprendizaje. A partir de ese instante los estudiantes matriculados en el módulo lo visualizan en su panel.

Figura 40: Botón de publicación del objeto de aprendizaje, destacado mediante flecha.

6. Flujo del estudiante

6.1 Acceso al material publicado

El estudiante ingresa con su cuenta institucional, selecciona el módulo correspondiente y visualiza la lista de objetos de aprendizaje publicados. No dispone de opciones de registro, edición ni regeneración.

Figura 41: Panel del estudiante con la lista de materiales publicados.

6.2 Consumo del resumen

Al abrir un objeto de aprendizaje, el estudiante visualiza el resumen estructurado por defecto. La vista presenta el contenido organizado en secciones, con conceptos clave resaltados para facilitar la lectura.

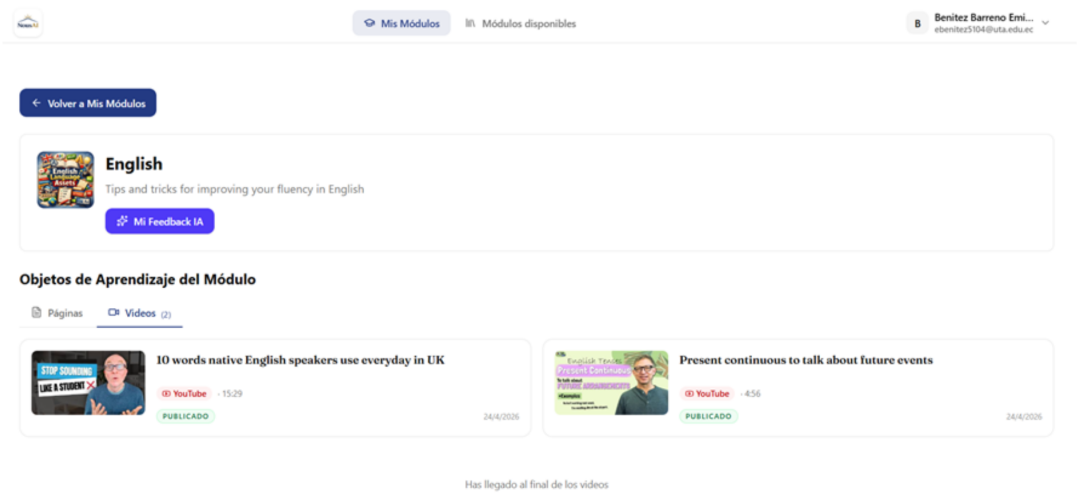

Figura 42: Vista de resumen para el estudiante.

6.3 Estudio con *flashcards*

Desde la pestaña correspondiente, el estudiante revisa las tarjetas con pregunta en el anverso y respuesta en el reverso. Cada tarjeta se voltea con un clic.

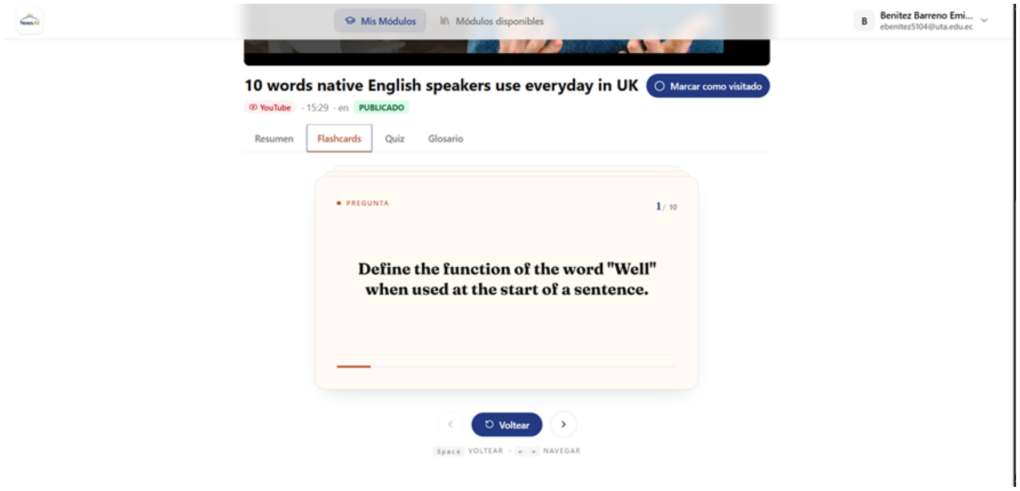

Figura 43: Tarjeta de estudio en modo estudiante. La flecha indica el control de volteo.

6.4 Autoevaluación con el cuestionario

El estudiante responde las preguntas de opción múltiple. Al enviar las respuestas, el sistema muestra la retroalimentación inmediata indicando aciertos y errores.

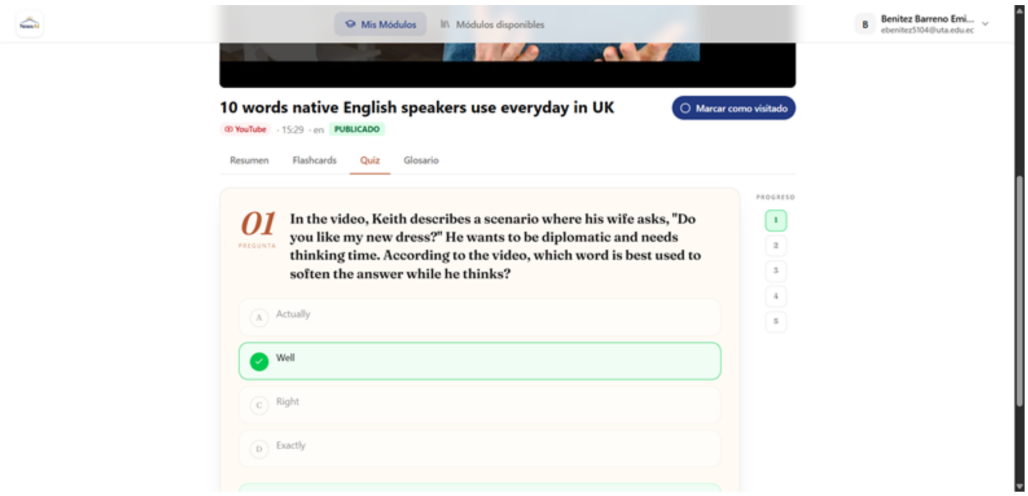

Figura 44: Cuestionario en vista estudiante con retroalimentación.

6.5 Consulta del glosario

El estudiante consulta el listado de términos técnicos extraídos del video junto con sus definiciones. El glosario se presenta ordenado alfabéticamente.

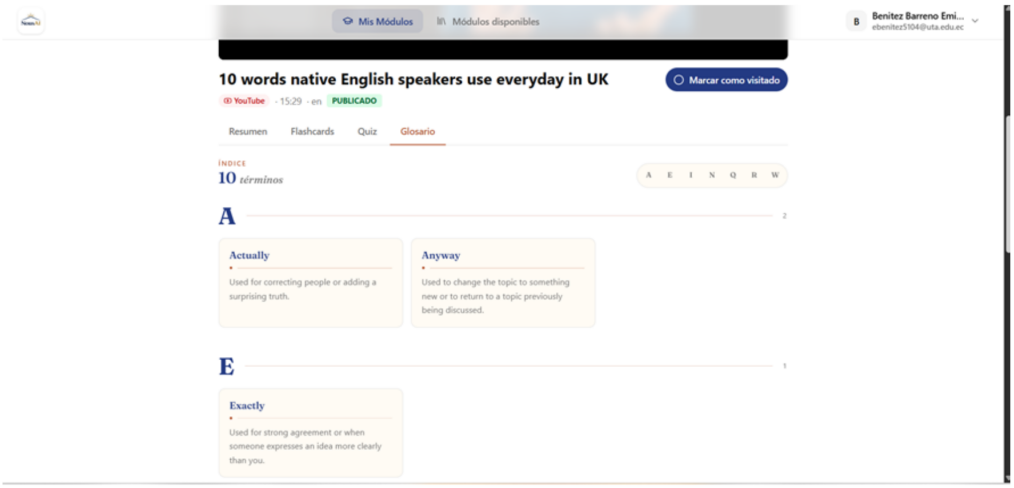

Figura 45: Glosario en vista estudiante.

ANEXO B: DOCUMENTACIÓN DE LA API

El presente anexo lista los *endpoints* REST expuestos por la API desarrollada (`edu-assistant-api`, NestJS 11), agrupados por área funcional. Para cada *endpoint* se indica el método HTTP, la ruta, el rol requerido para la autenticación y una breve descripción de su propósito. Todos los *endpoints* no marcados como públicos exigen un *token* JWT válido emitido por el servicio de autenticación de NousAI.

B.1 Endpoints de salud y observabilidad

Tabla 48: *Endpoints* de salud expuestos por la API

Método	Ruta	Auth	Descripción
GET	`/health`	Pública	Verifica que el servicio se encuentra activo. Devuelve estado y marca temporal.
GET	`/health/db`	Pública	Verifica la conectividad con la base de datos PostgreSQL mediante una consulta de prueba.

B.2 Endpoints de gestión de videos

Tabla 49: *Endpoints* del recurso `videos`

Método	Ruta	Auth	Descripción
POST	`/videos`	Docente	Registra un video a partir de una URL pública de YouTube.
POST	`/videos/upload`	Docente	Carga un archivo de video local (*multipart/form-data*).
GET	`/videos/module/:moduleId`	JWT	Lista paginada de videos del módulo indicado, con filtros opcionales.
GET	`/videos/:id`	JWT	Recupera el detalle completo del video, incluidos sus objetos de aprendizaje.

Continúa en la página siguiente

(Continuación Tabla 49)

Método	Ruta	Auth	Descripción
GET	`/videos/:id/status`	JWT	Devuelve el estado actual del procesamiento. Empleado por el cliente para *polling*.
POST	`/videos/:id/retry`	Docente	Reintenta la generación de contenido tras un fallo transitorio.
PATCH	`/videos/:id/content`	Docente	Actualiza el contenido editorial asociado al video.
DELETE	`/videos/:id`	Docente	Elimina el video y sus objetos de aprendizaje asociados.

B.3 Endpoints de objetos de aprendizaje

Tabla 50: *Endpoints* del recurso `learning-objects`

Método	Ruta	Auth	Descripción
POST	`/learning-objects`	Docente	Crea un nuevo objeto de aprendizaje.
GET	`/learning-objects/module/:moduleId`	JWT	Lista los objetos de aprendizaje de un módulo.
GET	`/learning-objects/:id`	JWT	Recupera un objeto de aprendizaje por identificador.
PATCH	`/learning-objects/reorder`	Docente	Reordena los objetos de aprendizaje dentro de un módulo.
PATCH	`/learning-objects/:id`	Docente	Actualiza los metadatos del objeto de aprendizaje.
PATCH	`/learning-objects/:id/content`	Docente	Actualiza el contenido por bloques del objeto de aprendizaje.

B.4 Endpoints de generación de contenido con IA

Tabla 51: *Endpoints* del recurso `content` (acceso restringido al rol docente)

Método	**Ruta**	**Descripción**
POST	`/content/generate-content`	Genera el contenido inicial de un objeto de aprendizaje mediante el agente de IA.
POST	`/content/regenerate-content`	Regenera o refina el contenido completo de un objeto de aprendizaje.
POST	`/content/regenerate-block`	Regenera un bloque específico dentro del contenido.
POST	`/content/expand-content`	Expande un bloque existente con detalles adicionales.
POST	`/content/extract-concepts`	Extrae automáticamente los conceptos clave del contenido.
POST	`/content/generate-image`	Genera una imagen asociada a un concepto o bloque.
POST	`/content/generate-concept`	Genera la definición textual de un concepto.
POST	`/content/generate-relations`	Genera relaciones semánticas entre objetos de aprendizaje.
POST	`/content/generate-activity`	Genera una actividad pedagógica (cuestionario, *flashcards*).

ANEXO C: CUESTIONARIO APLICADO A LOS ESTUDIANTES

Se aplicó la encuesta a estudiantes de la Universidad Técnica de Ambato con el propósito de cuantificar la prevalencia de las dificultades asociadas al uso de videos educativos y caracterizar el perfil de uso de herramientas de inteligencia artificial en el estudio autónomo. El instrumento consta de doce preguntas distribuidas en tres dimensiones: hábitos de uso de videos educativos (P1–P4), dificultades percibidas al estudiar sin material de apoyo (P5–P8) y experiencia previa con herramientas de inteligencia artificial aplicadas al aprendizaje (P9–P12). Se administró en formato digital mediante Google Forms y recogió 129 respuestas válidas.

A continuación se presenta el enunciado de cada pregunta junto con sus opciones de respuesta.

P1. ¿Con qué frecuencia utilizas videos de YouTube u otras plataformas para estudiar los temas de tu clase?
Opciones (Likert de frecuencia): Nunca / Raramente / A veces / Frecuentemente / Siempre.

P2. ¿En qué dispositivo ves principalmente los videos educativos para estudiar?
Opciones (única): Teléfono celular / Computadora portátil / Computadora de escritorio / Tableta / Otro.

P3. ¿Qué tipo de videos utilizas con más frecuencia para estudiar?
Opciones (única): Tutoriales paso a paso / Clases magistrales grabadas / Explicaciones cortas / Documentales o reportajes / Otro.

P4. ¿Cuánto tiempo en promedio dedicas a ver videos educativos fuera del horario de clase?
Opciones (única): Menos de una hora semanal / Entre 1 y 3 horas semanales / Entre 3 y 6 horas semanales / Más de 6 horas semanales.

P5. Cuando terminas de ver un video educativo, ¿con qué frecuencia tienes un material de resumen o síntesis disponible?
Opciones (Likert de frecuencia): Nunca / Raramente / A veces / Frecuentemente / Siempre.

P6. ¿Cuál de las siguientes dificultades experimentas con más frecuencia al estudiar con videos?
Opciones (única): Dificultad para retener la información / Dificultad para identificar las ideas principales / Falta de material complementario / Duración excesiva del video / Otro.

P7. ¿Qué tan fácil te resulta identificar las ideas principales de un video educativo sin ningún material de apoyo?
Opciones (Likert de facilidad): Muy difícil / Difícil / Intermedio / Fácil / Muy fácil.

P8. ¿Con qué frecuencia necesitas volver a ver un video varias veces porque no lograste retener la información suficiente la primera vez?
Opciones (Likert de frecuencia): Nunca / Raramente / A veces / Frecuentemente / Siempre.

P9. ¿Has usado alguna herramienta de inteligencia artificial (ChatGPT, Gemini u otras) para ayudarte con el estudio?
Opciones (única): Sí / No.

P10. ¿Para qué has usado principalmente la inteligencia artificial en el contexto del aprendizaje?
Opciones (única): Resolver dudas puntuales / Resumir textos o videos / Traducir textos o frases / Generar ejercicios de práctica / No he usado inteligencia artificial / Otro.

P11. ¿Qué tan satisfecho has quedado con los resultados que te ha dado una herramienta de IA cuando la usaste para estudiar?
Opciones (Likert de satisfacción): Muy insatisfecho / Insatisfecho / Neutral / Satisfecho / Muy satisfecho.

P12. ¿Estarías dispuesto a utilizar una herramienta que, a partir de un video, genere automáticamente un resumen, glosario y preguntas de comprensión?
Opciones (única): Sí / No / Tal vez.

ANEXO D: CUESTIONARIO SUS EXTENDIDO APLICADO EN LA EVALUACIÓN DE USABILIDAD

Este anexo recoge el instrumento administrado al subgrupo evaluador del sistema. El cuestionario se organiza en cuatro bloques aplicados en el siguiente orden: caracterización demográfica, identificación de la modalidad VARK del participante, escala estandarizada *System Usability Scale* (SUS) de Brooke [37] y un campo abierto final de retroalimentación. La ubicación del comentario al cierre permite que el participante responda con la experiencia completa del SUS ya procesada. El instrumento se administró en formato digital mediante Google Forms tras la sesión guiada de uso del sistema. La aplicación fue anónima y no se recolectó información personal identificable.

Bloque 1 — Demográfico

Q-DEM-1. ¿Cuál es su edad?
Tipo de respuesta: Numérica (años cumplidos).

Q-DEM-2. ¿Con qué sexo se identifica?
Opciones (única): Femenino / Masculino / Prefiero no decirlo.

Q-DEM-3. ¿A qué carrera pertenece?
Tipo de respuesta: Lista desplegable con la oferta académica de pregrado de la Universidad Técnica de Ambato, más la opción “Otra”.

Q-DEM-4. ¿En qué nivel del Programa Regular de Inglés se encuentra actualmente?
Opciones (única): Nivel I (A1 Principiante) / Nivel II (A2 Elemental) / Nivel III (B1 Preintermedio) / Nivel IV (B1+ Intermedio) / Nivel V (B2 Intermedio Alto) / Nivel VI (B2+ Avanzado) / Nivel VII (C1 Dominio Operativo Eficaz) / Nivel VIII (C1+).

Bloque 2 — VARK

Q-VARK-1. ¿Cuál es la modalidad que le entregó el cuestionario VARK?
Opciones (única): Visual / Auditivo / Lectura-Escritura / Kinestésico / Multimodal.

Q-VARK-2. ¿Está usted de acuerdo con el resultado obtenido en el cuestionario VARK?
Opciones (única): Sí / No / Parcialmente.

Bloque 3 — SUS

Los diez ítems siguientes corresponden a la versión estándar del *System Usability Scale*. Cada uno se responde en escala Likert de cinco puntos, donde 1 equivale a "Totalmente en desacuerdo" y 5 a "Totalmente de acuerdo". Los ítems impares aportan al puntaje con la fórmula $x - 1$ y los pares con $5 - x$; la suma se multiplica por 2,5 para obtener el puntaje final sobre 100, con umbral de aceptabilidad de 68 puntos [38].

SUS-01. Creo que me gustaría utilizar este sistema con frecuencia.

SUS-02. Encontré el sistema innecesariamente complejo.

SUS-03. Pensé que el sistema era fácil de utilizar.

SUS-04. Creo que necesitaría el apoyo de una persona técnica para poder utilizar este sistema.

SUS-05. Encontré que las diversas funciones del sistema estaban bien integradas.

SUS-06. Pensé que había demasiada inconsistencia en este sistema.

SUS-07. Imagino que la mayoría de las personas aprenderían a utilizar este sistema rápidamente.

SUS-08. Encontré el sistema muy difícil de utilizar.

SUS-09. Me sentí muy seguro al utilizar el sistema.

SUS-10. Necesité aprender muchas cosas antes de poder utilizar el sistema.

Bloque 4 — Retroalimentación abierta

Q-FB-1. Comentario o retroalimentación abierta sobre el sistema.
Tipo de respuesta: Texto libre, opcional.

ANEXO E: CONSULTAS SQL DE VALIDACIÓN DEL OBJETIVO ESPECÍFICO 2

Este anexo recoge las consultas SQL ejecutadas sobre la base de datos PostgreSQL del entorno de implantación de NousAI. Cada consulta sustenta uno o varios indicadores reportados en la sección de resultados del Objetivo Específico 2 (Capítulo IV). Se incluye como complemento de trazabilidad: las cifras presentadas en las tablas y figuras se obtienen directamente al ejecutar estas consultas sobre el esquema productivo del *feature* de videos.

C.1 Resumen general de procesamiento de videos

Consulta asociada a la Tabla 36 y a las cifras agregadas reportadas en la Tabla 39. Determina el número total de videos, los completados, los fallidos y los estadísticos de duración del corpus.

```
SELECT
  COUNT(*) AS total,
  COUNT(*) FILTER (WHERE status = 'COMPLETED') AS completados,
  COUNT(*) FILTER (WHERE status = 'FAILED')    AS fallidos,
  ROUND(AVG(duration_seconds)::numeric, 2)     AS dur_media_s,
  PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY duration_seconds)
    AS dur_mediana_s,
  MIN(duration_seconds) AS dur_min_s,
  MAX(duration_seconds) AS dur_max_s,
  ROUND(SUM(duration_seconds)::numeric / 60, 1) AS dur_total_min
FROM videos;
```

C.2 Distribución de videos por origen

Cuantifica la proporción de videos provenientes de URL de YouTube frente a archivos subidos directamente. Sustenta la Tabla 36.

```
SELECT source_kind, COUNT(*) AS total
FROM videos
GROUP BY source_kind;
```

C.3 Idioma detectado en los videos

Reporta el idioma identificado por Whisper o por los metadatos del subtítulo nativo. Alimenta la Tabla 36.

```
SELECT
  COALESCE(detected_language, '(sin deteccion)') AS idioma,
  COUNT(*) AS total
FROM videos
GROUP BY detected_language
ORDER BY total DESC;
```

C.4 Cobertura de bloques por tipo

Cuenta cuántos videos completados poseen efectivamente cada tipo de bloque generado por los agentes. Sustenta la Figura 22.

```
SELECT
  b.type AS block_type,
  COUNT(DISTINCT b.learning_object_id) AS videos_con_bloque
FROM blocks b
JOIN videos v ON v.learning_object_id = b.learning_object_id
WHERE v.status = 'COMPLETED'
  AND b.type IN ('SUMMARY','FLASHCARDS','QUIZ','GLOSSARY')
GROUP BY b.type
ORDER BY videos_con_bloque DESC;
```

C.5 Rendimiento del pipeline por modelo de lenguaje

Calcula intentos, latencia mediana y percentil 95, y promedios de *tokens* de entrada y salida por modelo. Sustenta la Tabla 40 y la Figura 23.

```
SELECT
  vga.provider,
  vga.model,
  COUNT(*) AS intentos,
  ROUND(AVG(vga.processing_time_ms)::numeric, 0) AS t_medio_ms,
  PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY vga.processing_time_ms)
    AS t_p50_ms,
  PERCENTILE_CONT(0.95) WITHIN GROUP (ORDER BY vga.processing_time_ms)
    AS t_p95_ms,
  ROUND(AVG(vga.tokens_input)::numeric, 0)  AS tokens_in_prom,
```

```
  ROUND(AVG(vga.tokens_output)::numeric, 0) AS tokens_out_prom
FROM video_generation_attempts vga
GROUP BY vga.provider, vga.model
ORDER BY intentos DESC;
```

C.6 Tasa de éxito a nivel de intento de generación

Mide qué porcentaje de los intentos individuales de los agentes terminaron sin error. Sustenta la Tabla 39.

```
SELECT
  COUNT(*) AS total_intentos,
  COUNT(*) FILTER (WHERE failed_types IS NULL) AS sin_fallos,
  ROUND(
    100.0 * COUNT(*) FILTER (WHERE failed_types IS NULL)
    / NULLIF(COUNT(*), 0),
    2
  ) AS pct_exito
FROM video_generation_attempts;
```

C.7 Distribución por duración del video

Agrupa los videos en rangos de duración para caracterizar el corpus. Sustenta la Figura 21.

```
SELECT
  CASE
    WHEN duration_seconds < 120  THEN '<2 min'
    WHEN duration_seconds < 300  THEN '2-5 min'
    WHEN duration_seconds < 600  THEN '5-10 min'
    WHEN duration_seconds < 1200 THEN '10-20 min'
    ELSE '20+ min'
  END AS rango,
  COUNT(*) AS videos
FROM videos
WHERE duration_seconds IS NOT NULL
GROUP BY rango
ORDER BY MIN(duration_seconds);
```

C.8 Tiempo de pipeline frente a duración del video

Compara el tiempo total dedicado por el pipeline al procesar cada video con la duración del video fuente. Sustenta la Figura 24.

```
SELECT
  v.learning_object_id AS video_id,
  ROUND(v.duration_seconds::numeric, 1) AS dur_video_s,
  ROUND(SUM(vga.processing_time_ms)::numeric / 1000, 1)
    AS pipeline_total_s,
  ROUND(
    ((SUM(vga.processing_time_ms) / 1000.0)
      / NULLIF(v.duration_seconds, 0))::numeric,
    2
  ) AS ratio_proc_vs_video
FROM videos v
JOIN video_generation_attempts vga ON vga.video_id = v.learning_object_id
WHERE v.status = 'COMPLETED'
  AND v.duration_seconds IS NOT NULL
GROUP BY v.learning_object_id, v.duration_seconds
ORDER BY pipeline_total_s DESC;
```

C.9 Patrones de fallo en intentos de generación

Clasifica los intentos fallidos por mensaje de error y por los tipos de bloque involucrados. Sustenta la Figura 25.

El campo `failed_types` es de tipo `JSONB` y almacena un arreglo de objetos con la forma {“type”: <BlockType>, “error”: <mensaje>}. La consulta descompone el arreglo y agrupa los fallos por mensaje de error y tipo de bloque.

```
SELECT
  fe ->> 'error' AS mensaje_error,
  fe ->> 'type'  AS tipo_bloque,
  COUNT(*) AS ocurrencias
FROM video_generation_attempts vga,
     LATERAL jsonb_array_elements(vga.failed_types) AS fe
WHERE vga.failed_types IS NOT NULL
GROUP BY mensaje_error, tipo_bloque
ORDER BY ocurrencias DESC;
```

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JOSE SEBASTIAN
CAMINO MORENO

C.10 Consumo agregado de tokens

Totaliza el consumo de *tokens* de entrada y salida del corpus, y calcula el promedio por intento de generación. Sustenta la Tabla 39.

```
SELECT
  SUM(tokens_input)                    AS tokens_in_total,
  SUM(tokens_output)                   AS tokens_out_total,
  SUM(tokens_input + tokens_output)    AS tokens_total,
  ROUND(AVG(tokens_input + tokens_output)::numeric, 0)
    AS tokens_prom_intento
FROM video_generation_attempts;
```

C.11 Adopción temporal del sistema

Cuenta los videos cargados por día durante el piloto. Sustenta la Figura 20.

La tabla `videos` no almacena fecha de creación propia; esta se toma del objeto de aprendizaje asociado (`learning_objects.created_at`), con el cual el video mantiene una relación uno a uno por `learning_object_id`.

```
SELECT
  DATE(lo.created_at) AS fecha,
  COUNT(*) AS videos_cargados
FROM videos v
JOIN learning_objects lo ON lo.id = v.learning_object_id
GROUP BY DATE(lo.created_at)
ORDER BY fecha;
```