



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

Tema:

**SISTEMA DE TUTORÍA DOCENTE BASADO EN IA GENERATIVA
SUPERVISADA.**

Trabajo de titulación, modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Software

ÁREA: Inteligencia artificial

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

AUTOR: Zhu Ordóñez Erick Daniel

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

Ambato – Ecuador

julio de 2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE TUTORÍA DOCENTE BASADO EN IA GENERATIVA SUPERVISADA, desarrollado bajo la modalidad Proyecto de Investigación por el señor Zhu Ordóñez Erick Daniel, 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 su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2026.

Ing. Félix Oscar Fernández Peña, MSc, PhD.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE TUTORÍA DOCENTE BASADO EN IA GENERATIVA SUPERVISADA 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.

Zhu Ordóñez Erick Daniel

C.C. 0707047643

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.

Zhu Ordóñez Erick Daniel

C.C. 0707047643

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Zhu Ordóñez Erick Daniel, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE TUTORÍA DOCENTE BASADO EN IA GENERATIVA SUPERVISADA, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, julio 2026.

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Hernan Fabricio Naranjo Avalos, Mg.

PROFESOR CALIFICADOR

Ing. Leonardo David Torres Valverde, Mg.

PROFESOR CALIFICADOR

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
ÍNDICE GENERAL DE CONTENIDOS	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	xi
ÍNDICE DE ANEXOS	xiv
RESUMEN EJECUTIVO	xv
ABSTRACT	xvi
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.1.1 Planteamiento del problema	1
1.2 Antecedentes investigativos	3
1.2.1 Mediación conversacional con el estudiante	3
1.2.2 Retorno de información al docente	6
1.2.3 Retroalimentación personalizada al estudiante	8
1.2.4 Producción asistida de material didáctico	10
1.2.5 Riesgos del uso no supervisado y supervisión humana	12
1.2.6 Lecturas del estado del arte	14
1.3 Fundamentación teórica	15
1.3.1 Inteligencia Artificial Generativa y Grandes Modelos de Lenguaje	15
1.3.2 Sistemas de Tutoría Inteligente	17

1.3.3	Aprendizaje Autorregulado y retroalimentación educativa	18
1.3.4	Analítica del Aprendizaje	19
1.3.5	Consideraciones éticas y diseño responsable	20
1.4	Objetivos	21
1.4.1	Objetivo general	21
1.4.2	Objetivos específicos	22
CAPÍTULO II. METODOLOGÍA		23
2.1	Materiales	23
2.2	Métodos	25
2.2.1	Modalidad de la investigación	25
2.2.2	Población y muestra	26
2.2.3	Metodología de desarrollo de software	28
2.2.4	Recolección de información	32
2.2.5	Procesamiento y análisis de datos	45
CAPÍTULO III. RESULTADOS Y DISCUSIÓN		49
3.1	Desarrollo de la propuesta	49
3.1.1	Requisitos funcionales y no funcionales	50
3.1.2	Casos de uso	53
3.1.3	Selección de tecnologías	63
3.1.4	Arquitectura del sistema	71
3.1.5	Diseño de la base de datos	73
3.1.6	Diagramas de clases por paquetes	77
3.1.7	Diseño de la API REST	85
3.1.8	Flujos	88
3.1.9	Diseño de prompts	93
3.1.10	Diseño de la interfaz de usuario	98
3.1.11	Seguridad, protección de datos y auditoría	108
3.2	Validación de la propuesta	111

3.2.1	Evaluación de usabilidad con estudiantes mediante el SUS	111
3.2.2	Validación cualitativa con docentes	116
3.2.3	Discusión integrada de resultados	121
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES		124
4.1	Conclusiones	124
4.2	Recomendaciones	126
BIBLIOGRAFÍA		130
ANEXOS		134

ÍNDICE DE TABLAS

1.1	Sistemas conversacionales en educación superior	4
1.2	Sistemas de retorno de información al docente	6
1.3	Sistemas de retroalimentación personalizada al estudiante	8
1.4	Implementaciones de producción asistida con IA generativa	10
1.5	Riesgos identificados y mecanismos de supervisión propuestos	13
1.6	Principales riesgos y consideraciones éticas en el uso de IA generativa en educación	20
2.7	Formato de ficha de contenido documental	23
2.8	Comparación de marcos ágiles considerados para el desarrollo del sis- tema	28
2.9	Roles del equipo de desarrollo	29
2.10	Resultados de la Pregunta 1 a Docentes	33
2.11	Resultados de la Pregunta 2 a Docentes	34
2.12	Resultados de la Pregunta 3 a Docentes	35
2.13	Resultados de la Pregunta 4 a Docentes	36
2.14	Resultados de la Pregunta 5 a Docentes	37
2.15	Resultados de la Pregunta 6 a Docentes	38
2.16	Ficha de Contenido N° 1	39
2.17	Ficha de Contenido N° 2	40
2.18	Ficha de Contenido N° 3	41
3.19	Requisitos funcionales del subsistema de tutoría con IA.	50
3.20	Requisitos no funcionales del subsistema de tutoría con IA.	51
3.21	Casos de uso de la tutoría con IA.	54
3.22	Especificación de CU-01.	57
3.23	Especificación de CU-02.	58
3.24	Especificación de CU-03.	59
3.25	Especificación de CU-04.	59

3.26	Especificación de CU-05.	59
3.27	Especificación de CU-06.	60
3.28	Especificación de CU-07.	60
3.29	Especificación de CU-08.	61
3.30	Especificación de CU-09.	62
3.31	Especificación de CU-10.	62
3.32	Especificación de CU-11.	63
3.33	Comparación de frameworks de backend.	63
3.34	Comparación de frameworks de frontend.	64
3.35	Comparación de gestores de base de datos relacional.	65
3.36	Comparación de ORM para el acceso a datos.	66
3.37	Comparación de soluciones de almacenamiento vectorial.	67
3.38	Comparación de mecanismos de procesamiento asíncrono.	68
3.39	Comparación de proveedores de modelos de IA.	69
3.40	Comparación de enfoques de validación de salidas de IA.	70
3.41	Resumen del stack tecnológico del subsistema de tutoría con IA. . . .	71
3.42	Entidades de la base de datos correspondientes a la tutoría con IA. . .	75
3.43	Endpoints del tutor inteligente.	86
3.44	Endpoints de interacción con el objeto de aprendizaje.	87
3.45	Endpoints de consulta de retroalimentación al docente.	87
3.46	Endpoints de consulta de retroalimentación al estudiante.	88
3.47	Catálogo de prompts del sistema.	95
3.48	Estadística descriptiva del puntaje SUS global obtenido sobre los estu- diantes participantes	112
3.49	Promedio y dispersión por ítem del cuestionario SUS	114
3.50	Subpuntajes por dimensión (Lewis y Sauro, 2009), escala 0–100 . . .	115
4.51	Datos de contexto de los docentes participantes	147
4.52	Síntesis de respuestas — Utilidad pedagógica	148
4.53	Síntesis de respuestas — Claridad del contenido	148
4.54	Síntesis de respuestas — Confianza en la información	149
4.55	Síntesis de respuestas — Frecuencia y oportunidad	149

4.56 Síntesis de respuestas — Facilidad de acceso y consulta	150
4.57 Síntesis de respuestas — Disposición a utilizar el sistema	150
4.58 Síntesis de comentarios finales	151

ÍNDICE DE FIGURAS

2.1	Tablero de Trello utilizado para la gestión del backlog y del flujo de trabajo del proyecto.	31
3.2	Diagrama de casos de uso del subsistema de tutoría con IA.	56
3.3	Arquitectura del sistema en tres capas.	72
3.4	Modelo entidad-relación de la tutoría con IA.	77
3.5	Diagrama de clases del paquete providers/ai.	79
3.6	Diagrama de clases del paquete features/chat.	80
3.7	Diagrama de clases del paquete features/teacher-feedback. . . .	82
3.8	Diagrama de clases del paquete features/student-ai-feedback. .	83
3.9	Diagrama de dependencias entre paquetes de la tutoría con IA.	85
3.10	Diagrama de secuencia del flujo del tutor inteligente.	90
3.11	Diagrama de secuencia del flujo de generación de retroalimentación al docente.	91
3.12	Diagrama de secuencia del flujo de generación de retroalimentación al estudiante.	92
3.13	Actividades, foro de preguntas y registro de feedback en la vista del objeto de aprendizaje.	100
3.14	Vista de chat del tutor inteligente.	101
3.15	Vista del listado de retroalimentaciones al docente en un módulo de instrucción.	102
3.16	Vista del detalle de una retroalimentación al docente.	103
3.17	Vista del listado de retroalimentaciones recibidas por el estudiante en un módulo de instrucción.	104
3.18	Vista del detalle de una retroalimentación recibida por el estudiante. .	105
3.19	Plantilla del correo de retroalimentación al docente (Parte 1).	106
3.20	Plantilla del correo de retroalimentación al docente (Parte 2).	106
3.21	Plantilla del correo de retroalimentación al estudiante (Parte 1). . . .	107
3.22	Plantilla del correo de retroalimentación al estudiante (Parte 2). . . .	107

3.23	Distribución de los puntajes SUS individuales obtenidos sobre los es-	
	tudiantes participantes.	113
3.24	Puntaje promedio por ítem del cuestionario SUS.	114
4.25	Diagrama global del esquema de base de datos.	135

ÍNDICE DE ANEXOS

Anexo A	Esquema global de base de datos	134
Anexo B	Catálogo de prompts del sistema	136
Anexo C	Respuestas del cuestionario docente	147

RESUMEN EJECUTIVO

En la educación superior, los estudiantes suelen recibir retroalimentación tardía o intermitente, y los docentes rara vez disponen de información oportuna sobre el aprendizaje de sus grupos. Este trabajo desarrolla un sistema de tutoría docente basado en Inteligencia Artificial Generativa supervisada, que ofrece al estudiante acompañamiento alineado con el contenido de cada lección y devuelve al docente una visión consolidada del avance de su grupo. El sistema expone una API REST mediante la cual el tutor inteligente y las retroalimentaciones se integraron a NousAI, que reúne tres propuestas complementarias.

La propuesta articula tres componentes: un tutor inteligente contextualizado por objeto de aprendizaje, retroalimentación semanal al docente por objeto de aprendizaje y por módulo, y retroalimentación semanal al estudiante por módulo. La supervisión recae en el docente, quien decide qué objetos de aprendizaje se publican y accede a retroalimentación que considera las acciones de los estudiantes.

La validación aplicó el cuestionario System Usability Scale a una muestra de estudiantes y un cuestionario cualitativo a cinco docentes. La usabilidad percibida superó el valor de referencia adoptado, con una media de 71,09 sobre 100 a partir de 154 respuestas válidas y un intervalo de confianza al 95 % entre 68,34 y 73,84. La facilidad de uso resultó sólida (72,79) frente a una aprendibilidad más baja (64,29), que concentra el mayor margen de mejora. Los docentes manifestaron disposición unánime a seguir usándolo, con la confianza en la información generada por el modelo como principal reserva.

Palabras clave: NousAI, Inteligencia artificial generativa, Tutoría inteligente, Retroalimentación docente, Aprendizaje autorregulado.

ABSTRACT

In higher education, students often receive late or intermittent feedback, and teachers rarely have timely information about their groups' learning. This work develops a teacher-supervised generative AI tutoring system, which provides students with support aligned to the content of each lesson and gives teachers a consolidated view of their group's progress. The system exposes a REST API through which the intelligent tutor and the feedback features were integrated into NousAI, which brings together three complementary proposals.

The proposal articulates three components: an intelligent tutor contextualized by learning object, weekly feedback for the teacher at the learning-object and module levels, and weekly feedback for the student by module. Supervision rests with the teacher, who decides which learning objects are published and accesses feedback that takes into account students' actions.

Validation applied the System Usability Scale questionnaire to a sample of students and a qualitative questionnaire to five teachers. Perceived usability exceeded the adopted reference value, with a mean of 71.09 out of 100 from 154 valid responses and a 95 % confidence interval between 68.34 and 73.84. Usability proved solid (72.79) against a lower learnability (64.29), which concentrates the greatest room for improvement. Teachers expressed unanimous willingness to keep using it, with confidence in the information generated by the model as the main reservation.

Keywords: NousAI, Generative artificial intelligence, Intelligent tutoring, Teacher feedback, Self-regulated learning.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

Sistema de tutoría docente basado en IA generativa supervisada.

1.1.1 Planteamiento del problema

Uno de los pilares del proceso de enseñanza-aprendizaje en la educación superior es que el estudiante regule su propio aprendizaje. Para hacerlo necesita retroalimentación oportuna, recibida mientras todavía hay tiempo de corregir el rumbo. Sostener esa retroalimentación de forma continua, sin embargo, resulta difícil en cursos con matrícula amplia y tiempo de tutoría acotado: el seguimiento individual del avance compete con las demás responsabilidades docentes, y los vacíos conceptuales tienden a hacerse visibles tarde.

Desde la liberación pública de ChatGPT en 2022, la investigación sobre Inteligencia Artificial en educación superior ha crecido de manera sostenida. Una revisión sistemática de 88 publicaciones del periodo 2013-2023 muestra que la mayor parte de la producción se concentra entre 2021 y 2023 [1]. Los Grandes Modelos de Lenguaje (LLMs) generan texto coherente, procesan instrucciones complejas y producen respuestas contextualizadas [2], lo que abre la posibilidad de que asuman tareas que antes dependían exclusivamente del docente.

Calificar y retroalimentar el trabajo de los estudiantes representa una parte importante de la jornada docente, sobre todo en cursos numerosos. La evidencia experimental muestra que los LLMs pueden absorber parte de esa carga sin sacrificar calidad: en un curso de Física en Estados Unidos, el 70 % del feedback generado por GPT-3.5 con cuatro ejemplos few-shot requirió mínima o ninguna modificación antes de entregarlo, y los propios estudiantes lo calificaron como más útil que el del instructor humano [3]. El uso libre de estas herramientas, sin acompañamiento docente, también produce daño: un experimento con estudiantes de programación en Eslovenia encontró correlaciones negativas entre el uso intensivo de LLMs para tareas que exigen pensamiento crítico y la calificación obtenida ($\rho = -0,347$, $p = 0,026$ para uso promedio) [4], mientras que no se halló impacto cuando los estudiantes recurrieron a la IA solo

para obtener explicaciones. La diferencia importa: la IA ayuda cuando complementa el esfuerzo cognitivo del estudiante y perjudica cuando lo sustituye.

En el contexto ecuatoriano, los antecedentes consultados muestran experiencias de IA aplicada a la generación de sílabos y a la evaluación automatizada [5], pero no de tutoría conversacional alineada con el contenido específico de cada curso. Esa diferencia de enfoque motiva el desarrollo y validación local de un sistema con esas características.

En la carrera de Software de la FISEI-UTA, los paralelos tienen una matrícula que normalmente excede la capacidad del docente para brindar atención individualizada, y las tutorías académicas se reducen a sesiones grupales de aproximadamente una hora semanal, sin espacios formales para acompañamiento individual sostenido. La facultad dispone de una herramienta de IA para generación de lecciones, pero su alcance se limita a la producción de material; no constituye un canal de tutoría conversacional ni un mecanismo para que los docentes accedan a información agregada sobre cómo sus estudiantes interactúan con el contenido del curso. La literatura recoge limitaciones recurrentes en este tipo de herramientas, asociadas frecuentemente al uso de datos de entrenamiento que no están alineados con el material académico. Sistemas con capacidades analíticas, en cambio, han mostrado que esa visibilidad sí se traduce en intervención efectiva: en un estudio en Australia, el grupo en riesgo intervenido alcanzó una tasa de aprobación del 45.5 %, frente al 10.9 % del grupo no intervenido [6].

La situación descrita responde a tres factores que se refuerzan entre sí. Primero, la relación estudiante-docente actual no permite la atención individual sostenida que la retroalimentación formativa requiere. Segundo, las herramientas de IA disponibles en la facultad no han alcanzado el nivel de ajuste necesario para responder de forma específica al contenido de cada asignatura. Tercero, los docentes no cuentan con un mecanismo que les devuelva información agregada sobre cómo están interactuando sus estudiantes con esas herramientas, de modo que la retroalimentación sigue dependiendo de lo que cada docente alcance a observar directamente. Si esta situación se mantiene, los vacíos conceptuales seguirán detectándose en las evaluaciones sumativas, cuando el margen para corregir es mínimo, y el rol del docente seguirá siendo reactivo en lugar de formativo.

A partir de lo anterior, este trabajo aborda la ausencia, en la FISEI-UTA, de un

sistema de tutoría docente basado en IA generativa supervisada que, sobre material académico revisado y aprobado por el docente, ofrezca a los estudiantes asistencia conversacional alineada con cada curso, entregue al docente una síntesis periódica del desempeño de sus estudiantes para apoyar las decisiones pedagógicas, y devuelva a cada estudiante un resumen personalizado sobre su propio proceso, de modo que las brechas de conocimiento puedan reconocerse mientras aún hay margen para corregir el rumbo.

1.2 Antecedentes investigativos

La incorporación de la Inteligencia Artificial Generativa en la educación superior ha pasado en los últimos años de las propuestas conceptuales a las implementaciones evaluadas en aulas reales. La revisión sistemática de Hernández-León y Rodríguez-Conde [1], sobre 88 publicaciones del periodo 2013-2023, muestra que la mayor parte de la producción se concentra en los últimos años, y la revisión de alcance de Kamalov et al. [7], sobre 44 estudios entre 2019 y mediados de 2023, agrupa esa producción en cuatro grandes aplicaciones: aprendizaje personalizado, sistemas de tutoría inteligente, automatización de la evaluación y colaboración profesor-estudiante.

Dentro de ese campo, los trabajos relevantes para este trabajo se distribuyen en cinco líneas, ordenadas según su cercanía al sistema propuesto: la mediación conversacional con el estudiante, el retorno de información al docente a partir del análisis de las interacciones, la retroalimentación personalizada al estudiante, la producción asistida de material didáctico, y los riesgos del uso no supervisado de los modelos.

1.2.1 Mediación conversacional con el estudiante

Los sistemas conversacionales que dialogan directamente con el estudiante para resolver dudas, repasar contenido o acompañar el estudio han sido objeto de un volumen creciente de investigación. La literatura ha avanzado tanto en demostrar viabilidad técnica como en identificar las condiciones pedagógicas bajo las cuales resultan útiles.

Tabla 1.1. Sistemas conversacionales en educación superior

Estudio	Tecnología	Contexto	Muestra	Hallazgos	Limitaciones
Lee et al. [8]	T-BERT (chatbot)	Salud pública, Taiwán	38 estudiantes	Mejoras en rendimiento ($\eta^2 = 0,18$), autoeficacia ($\eta^2 = 0,21$) y motivación ($\eta^2 = 0,13$); diferencias significativas en actitud ($p < 0,01$).	Intervención de dos semanas; posible efecto novedad.
Kuhail et al. [9]	Revisión sistemática	Multidisciplinario	36 chatbots	69.44 % en plataformas web; 88.88 % con flujos controlados por el bot; datos de entrenamiento inadecuados como limitación más reportada.	Cubre publicaciones hasta 2021.
Rudolph et al. [10]	GPT-4, Vicuna-13b, Mixtral-8x7b	Consejería, Alemania	64 diálogos, 41 estudiantes	GPT-4 con mejor desempeño (objetivo 3.54; emocional 2.87); fine-tuning de Vicuna acortó la brecha.	Muestra pequeña; GPT-4 es código cerrado.

Tabla 1.1. Sistemas conversacionales en educación superior (continuación)

Estudio	Tecnología	Contexto	Muestra	Hallazgos	Limitaciones
Jin et al. [11]	Maquetas conceptuales	Multidisciplinario, Corea	16 estudiantes	Preferencia por dashboards de analítica (10/16) y agentes de IA (5/16); 0/16 valoraron compañeros virtuales motivacionales como útiles.	Estudio cualitativo exploratorio.

Lee et al. [8] implementaron un chatbot de revisión post-clase y reportaron mejoras en rendimiento, autoeficacia y motivación, con diferencias estadísticamente significativas también en la actitud hacia el aprendizaje. Los estudiantes señalaron que la interfaz de chat redujo la timidez para preguntar y aumentó la disposición a investigar por cuenta propia. La corta duración del estudio aconseja interpretar los resultados con cautela frente al posible efecto novedad, observación que los propios autores hacen.

La revisión sistemática de Kuhail et al. [9], sobre 36 chatbots educativos publicados entre 2011 y 2021, ofrece la fotografía más amplia del campo en ese periodo. Tres hallazgos resultan relevantes. Primero, el 69.44 % de los chatbots se implementó en plataformas web. Segundo, la limitación reportada con mayor frecuencia fue el uso de datos de entrenamiento inadecuados, asociada a respuestas incorrectas y a frustración del usuario. Tercero, el 88.88 % operaba con conversaciones controladas por el bot, restricción que los sistemas basados en IA generativa pueden superar gracias a su capacidad de mantener diálogos abiertos. La revisión cubre publicaciones anteriores a la consolidación de los LLMs, por lo que sus resultados describen un punto de partida más que un estado actual.

Rudolph et al. [10] compararon modelos comerciales (GPT-3.5, GPT-4) con modelos de código abierto (Vicuna-13b, Mixtral-8x7b) en la formación de consejeros en Alemania. GPT-4 obtuvo el mejor desempeño en criterios objetivos (3.54) y emocionales (2.87), pero el fine-tuning de Vicuna acortó la brecha. El resultado tiene impli-

caciones operativas para una institución universitaria: existen alternativas de código abierto razonablemente competitivas, lo que abre opciones de costo y de soberanía de datos frente al uso exclusivo de modelos comerciales.

Jin et al. [11] exploraron qué tipo de interfaz preferirían los estudiantes para acompañar el estudio post-clase. La preferencia por dashboards de analítica y por agentes de IA, frente al rechazo de compañeros virtuales motivacionales, sugiere que los estudiantes valoran herramientas que les devuelvan información útil sobre su proceso, no figuras de acompañamiento emocional. El estudio es exploratorio y cualitativo, pero el patrón es consistente con lo que reporta el resto de la literatura.

1.2.2 Retorno de información al docente

Un segundo grupo de sistemas recoge información de las interacciones del estudiante con el material o con la herramienta de IA, y la procesa para entregarla al docente como insumo de decisión pedagógica. Esta vertiente se asocia en la literatura al campo de la Analítica del Aprendizaje (Learning Analytics).

Tabla 1.2. Sistemas de retorno de información al docente

Estudio	Tecnología	Contexto	Hallazgos	Limitaciones
Alalawi et al. [6]	Framework SP-PA con modelos predictivos de ML	Universitario, Australia	Grupo en riesgo intervenido alcanzó 45.5 % de aprobación, frente al 10.9 % del grupo no intervenido ($p < 0,0001$).	Generalización limitada; depende de la calidad de la predicción y de la acción docente.
Ouyang et al. [12]	Modelos predictivos + analítica del aprendizaje + retroalimentación al docente	Ingeniería en línea, China	El grupo experimental obtuvo mejor calificación final (89.33 vs 87.97; $F = 4,79$; $p = 0,008$) y mayor grado de interacción (9.52 vs 5.46; $p < 0,001$).	Estudio en un curso específico; resultados sensibles al diseño de la intervención.

Tabla 1.2. Sistemas de retorno de información al docente (continuación)

Estudio	Tecnología	Contexto	Hallazgos	Limitaciones
Mousavinasab et al. [13]	Revisión sistemática de Sistemas de Tutoría Inteligente	Multidisciplinario	Arquitectura clásica de 4 módulos; 54.71 % con interfaz web; uso extendido de reglas de condición-acción para dominios estructurados.	Cobertura previa a la consolidación de la IA generativa.

Alalawi et al. [6] presentan el caso más contundente de esta vertiente: cuando la predicción de desempeño se traduce en intervención docente concreta sobre estudiantes en riesgo, la tasa de aprobación cuadriplica la del grupo no intervenido (45.5 % frente a 10.9 %, $p < 0,0001$). El framework SPPA combina modelos de aprendizaje automático entrenados con las calificaciones históricas del propio curso con dashboards docentes que permiten enviar correos personalizados con retroalimentación a los estudiantes identificados como vulnerables. El estudio no aísla el efecto de la herramienta del efecto del docente, lo que es coherente con la postura general del campo: la analítica funciona cuando hay un docente que actúa sobre ella.

Ouyang et al. [12] avanzan sobre esa misma idea con un argumento de método: el valor de los modelos predictivos en educación no radica en la predicción aislada sino en su integración con la práctica docente. En un curso de ingeniería en línea con 62 estudiantes, la integración de un modelo predictivo con analítica de las interacciones y entrega de retroalimentación semanal produjo mejores calificaciones finales (89.33 vs 87.97) y un grado de interacción notablemente superior (9.52 vs 5.46 conexiones por estudiante). La conclusión confirma el principio de supervisión humana, pero también señala la variabilidad: docentes distintos interpretan el mismo dato de maneras distintas, lo que obliga a diseñar el retorno de información de modo que sea interpretable sin entrenamiento estadístico previo.

Mousavinasab et al. [13], en su revisión sistemática sobre 53 artículos de Sistemas de Tutoría Inteligente (ITS) publicados entre 2007 y 2017, confirman una arquitectura clásica organizada en cuatro módulos y un predominio de las plataformas web (54.71 %). Su estudio documenta que los ITS contemporáneos integran cada vez más

componentes que informan al docente sobre el grupo, no solo al estudiante, reflejando el reconocimiento de que la tutoría inteligente opera articulada con la decisión pedagógica humana.

1.2.3 Retroalimentación personalizada al estudiante

Más recientemente, ha cobrado interés un tipo de sistema que devuelve al propio estudiante información periódica y personalizada sobre su proceso: resúmenes de su desempeño, identificación de los conceptos donde tropieza y sugerencias concretas para reforzar el estudio. La premisa pedagógica conecta con el Aprendizaje Autorregulado, según el cual el estudiante puede actuar sobre su proceso en la medida en que dispone de información comprensible sobre dónde está y qué le falta.

Tabla 1.3. Sistemas de retroalimentación personalizada al estudiante

Estudio	Tecnología	Contexto	Hallazgos	Limitaciones
Su y Yang [14]	Marco IDEE para integración de IA generativa	Educación general	Marco teórico de cuatro pasos (Identificar, Determinar, Asegurar, Evaluar) para guiar la integración pedagógica de IA generativa.	Trabajo conceptual; no incluye validación empírica.
Weidlich et al. [15]	Retroalimentación Altamente Informativa (HIF) con analítica del aprendizaje	Universitario, Alemania, N=196	Estudiantes con HIF reportaron 4.30 veces más probabilidad de percibir el feedback como útil y 4.19 veces más probabilidad de regular su comportamiento de aprendizaje ($p < 0,001$).	Población acotada; intervención de una semana.

Tabla 1.3. Sistemas de retroalimentación personalizada al estudiante (continuación)

Estudio	Tecnología	Contexto	Hallazgos	Limitaciones
Chang et al. [16]	Principios de diseño para chatbots de IA basados en Aprendizaje Autorregulado (SRL) y Juicio de Aprendizaje (JOL)	Educación superior	Marco pedagógico que articula reverse prompting, establecimiento de objetivos y personalización para transformar el chatbot de proveedor de respuestas a facilitador del aprendizaje.	Propuesta conceptual basada en revisión teórica.

Su y Yang [14] proponen el marco IDEE como guía para integrar la IA generativa en educación. El marco articula cuatro pasos: identificar los resultados de aprendizaje deseados, determinar el nivel apropiado de automatización, asegurar las consideraciones éticas y evaluar la efectividad. El segundo paso es particularmente relevante para la retroalimentación dirigida al estudiante: los autores sostienen que el uso de ChatGPT no debe reemplazar el papel del docente, sino apoyar al estudiante en su proceso bajo parámetros pedagógicos definidos.

Weidlich et al. [15] llevaron esta idea al terreno experimental con un sistema de Retroalimentación Altamente Informativa (HIF) basado en analítica del aprendizaje. Cuando el feedback se construye con datos individuales del estudiante, las probabilidades de que lo perciba como útil y de que lo use para regular su comportamiento aumentan aproximadamente cuatro veces respecto a un feedback simple. Un hallazgo adicional importante es que la alfabetización en feedback de los estudiantes modera el efecto: quienes tienen mejor actitud hacia la retroalimentación aprovechan más estas herramientas, aunque incluso los estudiantes con baja alfabetización reportaron que el feedback detallado les aportó información valiosa para autorregularse.

Chang et al. [16] aportan los principios de diseño que sostienen pedagógicamente este tipo de sistemas. El reverse prompting, en el que el chatbot devuelve preguntas guía en lugar de respuestas directas, se alinea con la fase de autorreflexión del Aprendizaje Autorregulado y con el Juicio de Aprendizaje (Judgement of Learning), concepto que designa la evaluación que el estudiante hace de su propia comprensión. Los au-

tores documentan también que el feedback altamente informativo alcanza un tamaño de efecto reportado de $d = 0,99$, frente al $d = 0,75$ atribuido al feedback general en estudios clásicos.

1.2.4 Producción asistida de material didáctico

Una de las aplicaciones más documentadas de los LLMs en educación superior es la producción de material que el docente revisa antes de entregarlo al estudiante. La revisión es parte explícita del flujo, no un control posterior opcional, y es sobre esa idea que la literatura ha empezado a articular el concepto de IA supervisada.

Tabla 1.4. Implementaciones de producción asistida con IA generativa

Estudio	Tecnología	Contexto	Tipo de contenido	Hallazgos	Limitaciones
Wan y Chen [3]	GPT-3.5 con prompt engineering y few-shot (4 ejemplos)	Física conceptual, EE.UU.	Retroalimentación a respuestas escritas	70 % del feedback con mínima o ninguna modificación según 4 instructores; los estudiantes identificaron correctamente el origen humano o de IA en un rango de 13 % a 63 %.	Una sola pregunta conceptual; riesgo de alucinación.
Fuller et al. [17]	ChatGPT	Farmacia, EE.UU.	Análisis y síntesis de comentarios de evaluación	Tiempo de análisis reducido de 27.50 min a 10.50–12.50 min; concordancia interevaluador 0.71–0.82 en temas del curso.	Solo 4 cursos; comprensión contextual limitada.

Tabla 1.4. Implementaciones de producción asistida con IA generativa (continuación)

Estudio	Tecnología	Contexto	Tipo de contenido	Hallazgos	Limitaciones
Delgado Kloos et al. [18]	Llama 2 con fine-tuning sobre material del curso	Java, España	Borradores de retroalimentación y preguntas de entrevista	Fine-tuning sobre material del curso viable y de bajo costo; flujo borrador-aprobar formalizado.	Estudio de caso, sin medición de impacto.
Villegas-Ch et al. [5]	API de OpenAI y modelos BERT con similitud del coseno	Sílabos y rúbricas, Ecuador	Sílabos adaptativos y evaluación automatizada	Precisión de evaluación 78.5 % $\rightarrow$ 89.7 %; usabilidad 3.8 $\rightarrow$ 4.6/5; calificación media de los estudiantes 70.4 $\rightarrow$ 75.2 ($p < 0,001$; $d = 0,51$).	Dependencia crítica de la calidad de las rúbricas proporcionadas.

Wan y Chen [3] mostraron en un curso de Física que GPT-3.5, alimentado con cuatro ejemplos mediante few-shot learning, produjo retroalimentación que en el 70 % de los casos requirió mínima o ninguna modificación antes de ser entregada al estudiante, según el criterio de cuatro instructores. Los autores recomendaron explícitamente que el docente revise cada mensaje antes de entregarlo, en línea con el esquema supervisado que recorre toda la literatura de esta área.

Fuller et al. [17] aplicaron ChatGPT al análisis de comentarios de evaluación de cursos en educación farmacéutica. El tiempo medio de análisis pasó de 27.50 minutos (procesamiento humano) a entre 10.50 y 12.50 minutos con asistencia de IA, con una concordancia interevaluador en el rango 0.71 a 0.82 frente al análisis humano. Aunque el dominio es distinto, el patrón es transferible: la IA absorbe un volumen de texto que el docente no podría procesar manualmente, y la lectura interpretativa de los resultados queda del lado del docente.

Delgado Kloos et al. [18] presentaron un caso de implementación en un curso de Java en España utilizando Llama 2 con fine-tuning sobre el material del curso. El estudio aporta dos elementos. Primero, formalizó un esquema en el que la IA produce borradores que el docente revisa y aprueba, lo que coincide con la lógica supervisada que la literatura recomienda. Segundo, generó preguntas basadas en el código entregado por el estudiante, como mecanismo para verificar autoría en un escenario donde la generación automática de código es accesible.

En el ámbito ecuatoriano, Villegas-Ch et al. [5] desarrollaron una plataforma basada en la API de OpenAI y modelos BERT que generó sílabos adaptativos y evaluaciones automatizadas a partir de criterios extraídos de las rúbricas docentes mediante similitud del coseno. Tras seis meses de iteración guiada por los usuarios, la precisión de evaluación pasó de 78.5 % a 89.7 %, la usabilidad percibida de 3.8 a 4.6 sobre 5, y la calificación media de los estudiantes intervenidos subió de 70.4 a 75.2 ($p < 0,001$; $d = 0,51$). El aporte central del estudio para esta línea es de método: muestra que es posible alimentar al LLM con criterios extraídos del material del propio docente, en lugar de operar sobre un modelo genérico.

Los cuatro trabajos comparten una arquitectura común: el modelo genera, el docente revisa y aprueba, y solo entonces el contenido se incorpora al curso. Lo que el estudiante encuentra en estos sistemas es material que pasó por filtro docente, lo que convierte ese contenido aprobado en la base sobre la que pueden operar después otros componentes (tutoría conversacional, retroalimentación, analítica) sin que cada interacción individual exija una revisión docente adicional.

1.2.5 Riesgos del uso no supervisado y supervisión humana

Los trabajos revisados hasta aquí documentan condiciones bajo las cuales la IA aporta valor al proceso educativo. Un cuerpo paralelo de investigación examina lo que ocurre cuando el acompañamiento docente no existe, y describe los mecanismos que la literatura ha propuesto para mantenerlo presente cuando la IA opera a escala.

Tabla 1.5. Riesgos identificados y mecanismos de supervisión propuestos

Riesgo	Descripción	Mecanismo de mitigación reportado
Sustitución del es- fuerzo cognitivo	El estudiante delega en la IA tareas que exigen pensamiento crítico, lo que se asocia a peor desempeño.	Diseño que privilegie la explicación frente a la respuesta resuelta [4, 2].
Alucinación	Generación de información incorrecta presentada con alta confianza.	Esquema human-in-the-loop con revisión docente; validación estructural de las respuestas del modelo [3, 18].
Privacidad de datos	Exposición de información estudiantil sensible al transitar por servidores externos.	Acceso institucional; uso de modelos open-source cuando sea posible [2].
Equidad de acceso	Brecha digital que limita los beneficios de la IA a contextos con mejor infraestructura.	Acceso institucional; uso de modelos de código abierto y servicios con tier gratuito [2, 18].

Jost et al. [4], en el experimento ya citado con estudiantes de programación en Eslovenia, encontraron correlaciones negativas entre el uso intensivo de LLMs para tareas que exigen pensamiento crítico y la calificación obtenida ($\rho = -0,347$, $p = 0,026$ para uso promedio). La distinción que hacen los autores es operativa: no se halló impacto negativo cuando los estudiantes recurrieron a la IA solo para obtener explicaciones; el efecto adverso aparece cuando la IA entrega la solución y el estudiante la incorpora sin elaboración propia.

Kasneci et al. [2] sostienen, sobre la base de una revisión amplia del campo, que los LLMs no deben reemplazar al docente sino funcionar como herramientas de asistencia bajo supervisión humana continua. La supervisión, en su análisis, no es un control posterior sino un componente integrado al flujo: el docente revisa, ajusta y aprueba antes de que la salida del modelo llegue al estudiante. El concepto que articula esta postura es el de human-in-the-loop, un esquema en el que cada salida relevante del modelo pasa por revisión humana antes de incorporarse al flujo educativo.

Wan y Chen [3] y Delgado Kloos et al. [18] aportan, desde sus respectivos estudios, los dos mecanismos prácticos más recurrentes en la literatura para sostener esa supervisión: la revisión docente como paso obligatorio del flujo (en el caso del feedback

generado por GPT-3.5 a respuestas conceptuales) y el esquema “borrador, aprobación” como patrón formalizado (en el caso del fine-tuning de Llama 2 sobre material del curso). En ambos casos, la persistencia del contenido en el sistema educativo solo ocurre después de la aprobación del docente.

La revisión caso a caso, sin embargo, no es igualmente aplicable a todas las funciones que un sistema educativo basado en IA puede ofrecer. Aprobar un borrador de lección antes de su publicación es un acto que admite tiempos de revisión; intervenir cada respuesta de un tutor inteligente o cada resumen automático entregado al estudiante no lo es. Frente a esta diferencia, la literatura distingue de manera implícita entre la supervisión que opera sobre el contenido del curso (donde la aprobación docente puede ser previa y caso a caso) y la que opera sobre componentes derivados que actúan en tiempo real o de forma periódica sobre ese contenido ya aprobado. En estos últimos, la supervisión se sostiene mediante mecanismos estructurales: restricciones codificadas en el diseño del componente, validación automática de las salidas del modelo, registros de trazabilidad que permiten auditoría posterior, y la garantía de que toda la interacción se enmarca en material que el docente sí revisó previamente.

Estos mecanismos asumen además una condición previa: la validación estructural de las respuestas del modelo. Antes de que el docente pueda revisar una salida, esa salida debe ser técnicamente válida (formato esperado, campos completos, sin contenido espurio). La literatura coincide en que la combinación de validación automática y revisión humana define el estándar operativo para la integración de IA generativa en entornos educativos.

1.2.6 Lecturas del estado del arte

La revisión muestra que las cinco líneas tratadas en los apartados anteriores —mediación conversacional con el estudiante, retorno de información al docente, retroalimentación personalizada al estudiante, producción asistida de material didáctico y riesgos del uso no supervisado— se solapan con frecuencia. Los estudios más recientes tienden a combinar generación de contenido, diálogo con el estudiante y devolución de información al docente o al propio estudiante dentro de un mismo sistema. Esa convergencia refleja un campo que se ha movido desde funciones aisladas hacia propuestas que cubren distintos momentos del ciclo formativo.

Tres lecturas transversales emergen del conjunto. La primera es que la viabilidad técnica de los LLMs en educación superior cuenta con respaldo empírico suficiente [3, 16, 5], y el debate ha pasado de la capacidad de los modelos a las condiciones pedagógicas y arquitectónicas bajo las cuales su uso resulta responsable [2, 15]. La segunda es que la supervisión humana no aparece como un control añadido al final, sino como una característica de diseño que atraviesa toda la literatura [2, 1]; los estudios que reportan resultados positivos comparten la lógica de mantener al docente en el bucle [3, 18, 6], mientras que los que reportan daños lo hacen en escenarios de uso libre sin acompañamiento [4]. La tercera es que la utilidad de la IA depende fuertemente del ajuste de los datos sobre los que opera [18, 5]: los sistemas alimentados con el material del propio curso superan en relevancia a los sistemas genéricos, y los componentes de retroalimentación contruidos con datos individuales del estudiante superan a los que ofrecen mensajes uniformes [14, 11].

Estas tres lecturas configuran el espacio en el que se inscribe el sistema desarrollado en este trabajo: un sistema de tutoría que opera sobre material aprobado por el docente y se apoya en la supervisión humana de ese material, complementada con mecanismos estructurales de control sobre los componentes que interactúan directamente con el estudiante.

1.3 Fundamentación teórica

Los conceptos que sustentan el desarrollo del sistema se organizan en cinco bloques. Primero los fundamentos técnicos de la Inteligencia Artificial Generativa; luego los Sistemas de Tutoría Inteligente como tradición en la que se inscribe la propuesta; a continuación las bases pedagógicas del Aprendizaje Autorregulado y la retroalimentación educativa; después la Analítica del Aprendizaje como herramienta para el seguimiento del estudiante; y al cierre, las consideraciones éticas que enmarcan la integración de estas tecnologías en la educación superior.

1.3.1 Inteligencia Artificial Generativa y Grandes Modelos de Lenguaje

La Inteligencia Artificial es una rama de las Ciencias de la Computación que se ocupa del desarrollo de sistemas capaces de realizar tareas que normalmente requerirían inteligencia humana, como el razonamiento, la percepción y la toma de decisiones

[1]. Dentro de la IA, el Aprendizaje Automático (Machine Learning) reúne las técnicas mediante las cuales un sistema aprende patrones a partir de datos en lugar de seguir reglas explícitas [7], y el Aprendizaje Profundo (Deep Learning) es la rama del Machine Learning que utiliza redes neuronales con múltiples capas para aprender representaciones cada vez más abstractas de los datos [7]. La arquitectura Transformer, base de los modelos generativos actuales, pertenece a esta última familia. Su aporte central es el mecanismo de auto-atención (self-attention), que permite al modelo determinar la relevancia relativa de cada palabra de la entrada al generar la salida, sin las limitaciones secuenciales de arquitecturas anteriores [2].

La Inteligencia Artificial Generativa designa a los sistemas capaces de producir contenido nuevo, texto, código o imágenes, en lugar de limitarse a clasificar o predecir sobre datos existentes [2]. Los Grandes Modelos de Lenguaje (LLMs) son la materialización más visible de este enfoque: sistemas entrenados sobre grandes volúmenes de texto que generan oraciones verosímiles en respuesta a una instrucción o prompt [18]. ChatGPT, basado en la familia GPT (Generative Pre-trained Transformer), es el ejemplo de uso más extendido. La evolución de los LLMs ha llevado el uso de estos modelos en educación desde aplicaciones puntuales hacia integraciones más amplias dentro de plataformas de aprendizaje [7].

Para adaptar un LLM a un contexto específico sin reentrenarlo desde cero existen varias técnicas. El prompt engineering consiste en diseñar la instrucción que se envía al modelo de modo que la respuesta sea la deseada; la calidad de la salida depende fuertemente de la formulación del prompt [3]. El few-shot learning lleva esta idea más allá al incluir en el propio prompt un pequeño número de ejemplos resueltos, lo que permite al modelo inferir el patrón esperado sin reentrenamiento [2, 3]. El fine-tuning ajusta los pesos del modelo sobre un conjunto de datos específico del dominio, lo que produce respuestas más alineadas con un material concreto pero a un costo computacional mayor [18]. Finalmente, los embeddings son representaciones vectoriales del texto que capturan relaciones semánticas: dos textos con significado similar producen vectores cercanos en el espacio vectorial. Esta propiedad habilita la técnica de Retrieval-Augmented Generation (RAG), en la que el sistema recupera fragmentos de contenido semánticamente relevantes y los inyecta como contexto en el prompt, alineando la respuesta del LLM con un material específico sin modificar el modelo.

Su y Yang [14] proponen el marco IDEE (Identificar, Determinar, Asegurar y Evaluar) como guía para integrar la IA generativa en educación: identificar los resultados de aprendizaje deseados, determinar el nivel apropiado de automatización, asegurar las consideraciones éticas, y evaluar tanto los aprendizajes como el funcionamiento del sistema. El marco recoge la idea, hoy ampliamente compartida en la literatura, de que la integración de los LLMs en aulas universitarias se beneficia de procesos deliberados de diseño antes que del uso espontáneo de las herramientas.

1.3.2 Sistemas de Tutoría Inteligente

Los Sistemas de Tutoría Inteligente (ITS) son entornos de instrucción adaptativa que combinan técnicas de IA con métodos pedagógicos para ofrecer enseñanza personalizada [13]. Su propósito es aproximarse a la experiencia de aprender con un tutor humano, ajustando contenido y estrategia a las características del estudiante. La revisión sistemática de Mousavinasab et al. [13] identifica una arquitectura clásica organizada en cuatro módulos.

El Módulo Experto contiene el conocimiento del dominio que se va a enseñar. En los ITS clásicos este conocimiento se codificaba mediante reglas explícitas o bases de conocimiento estructuradas; en los ITS basados en IA generativa, el conocimiento puede proporcionarse al LLM mediante prompt engineering, fine-tuning o RAG.

El Módulo del Estudiante modela las características relevantes del aprendiz: nivel de conocimiento, actividades realizadas, respuestas, comportamiento observado, brechas detectadas. La forma en que se construye este modelo determina en buena medida la capacidad del sistema para personalizar la experiencia.

El Módulo Pedagógico decide qué hacer con la información del modelo del estudiante: qué retroalimentación entregar, qué pista ofrecer, qué tema reforzar. En los ITS clásicos, esta lógica se implementaba mediante reglas de condición-acción; en los basados en LLMs, las decisiones pedagógicas se canalizan a través de las instrucciones de sistema (system prompts) que guían al modelo para producir respuestas apropiadas.

El Módulo de Interfaz facilita la interacción entre el estudiante y el sistema. En los ITS conversacionales, este módulo adopta la forma de una interfaz de chat que permite la interacción en lenguaje natural.

La irrupción de los LLMs ha modificado lo que cada módulo puede hacer. La ge-

neración de contenido y de retroalimentación, que antes requería plantillas o reglas explícitas, puede ahora delegarse a un modelo generativo [7]. Esto reduce el costo de desarrollo de un ITS pero introduce a la vez la necesidad de mecanismos de control sobre la salida del modelo, que la fundamentación pedagógica de las siguientes subsecciones ayuda a definir.

1.3.3 Aprendizaje Autorregulado y retroalimentación educativa

El Aprendizaje Autorregulado (Self-Regulated Learning, SRL) es el proceso mediante el cual los estudiantes activan y sostienen la cognición, los comportamientos y los afectos orientados al logro de sus metas de aprendizaje [11]. La capacidad de autorregularse es uno de los factores asociados al alto rendimiento en contextos universitarios [11].

El modelo de Zimmerman conceptualiza el SRL como un ciclo de tres fases [11]. En la fase de previsión, el estudiante establece metas y planifica cómo alcanzarlas. En la fase de desempeño, se autocontrola y autoobserva durante la actividad. En la fase de autorreflexión, juzga su desempeño y reacciona ante los resultados, lo que retroalimenta la siguiente iteración del ciclo. El SRL abarca cuatro dimensiones de regulación: cognitiva, metacognitiva, conductual y motivacional. La evidencia disponible indica que las aplicaciones de IA pueden apoyar las tres primeras dimensiones, pero no la motivacional [11], lo que sugiere que la dimensión emocional del aprendizaje sigue requiriendo la intervención del docente. Un concepto asociado es el Juicio de Aprendizaje (Judgement of Learning, JOL), que se refiere a la evaluación que el propio estudiante hace de su comprensión o retención de un contenido [16].

La retroalimentación es uno de los mecanismos pedagógicos más potentes para apoyar el SRL. La literatura le atribuye un tamaño de efecto reportado de $d = 0,75$, valor que la ubica entre las intervenciones de mayor impacto sobre el rendimiento académico [16]. Para que sea efectiva, la retroalimentación debe ser detallada y procesable, formativa, accionable, alineada con los principios pedagógicos del curso, y escalable a las condiciones reales del aula [15].

El concepto de Retroalimentación Altamente Informativa (Highly Informative Feedback, HIF) responde a la dificultad de cumplir esos criterios a escala: combina retroalimentación detallada y accionable con escalabilidad apoyándose en analítica del apren-

dizaje y en IA [15]. La evidencia experimental muestra que un sistema HIF puede multiplicar por más de cuatro la probabilidad de que el estudiante perciba el feedback como útil y lo use para regular su comportamiento de aprendizaje, en comparación con una retroalimentación simple [15]. Estudios previos sintetizados por Chang et al. [16] reportan, además, que el feedback altamente informativo puede alcanzar tamaños de efecto cercanos a $d = 0,99$, frente al $d = 0,75$ atribuido al feedback general.

Una técnica pedagógica relevante para sistemas conversacionales es el reverse prompting, en el que el chatbot, en lugar de responder directamente, formula preguntas guía que llevan al estudiante a evaluar su propio razonamiento [16]. La técnica se alinea con la fase de autorreflexión del modelo de Zimmerman: un sistema que devuelve respuestas inmediatas tiende a sustituir el esfuerzo cognitivo del estudiante; uno que devuelve preguntas lo refuerza.

Por último, la alfabetización en feedback (feedback literacy) abarca las habilidades necesarias para interpretar y usar productivamente la retroalimentación recibida [15]. La evidencia muestra que esta variable modera el efecto de los sistemas de retroalimentación basados en IA, lo que indica que ningún sistema de tutoría sustituye por sí solo la capacidad del estudiante de leer y aplicar lo que se le devuelve.

1.3.4 Analítica del Aprendizaje

La Analítica del Aprendizaje (Learning Analytics, LA) es la medición, recolección, análisis y reporte de datos sobre los estudiantes y sus contextos, con el propósito de comprender y optimizar el aprendizaje [6]. Las Intervenciones de Analítica del Aprendizaje se definen como el uso de esos datos para identificar tempranamente a los estudiantes en riesgo y habilitar acciones formativas eficaces [6].

Los trace data son datos de rastreo capturados durante la actividad del estudiante en un entorno digital de aprendizaje. En un Learning Management System tradicional incluyen tiempo de lectura, número de clics o frecuencia de acceso. En sistemas de tutoría conversacional, los trace data se construyen a partir del registro de las interacciones entre el estudiante y el tutor: temas consultados, preguntas formuladas, respuestas a actividades, mensajes intercambiados [16, 12]. Su análisis permite construir indicadores que alimentan tanto la retroalimentación al estudiante como el reporte que se entrega al docente.

Los Sistemas de Alerta Temprana (Early Warning Systems) usan IA para detectar señales de riesgo, como caída en la actividad, repetición de errores o aumento de consultas sobre un mismo tema, y comunicarlas al docente antes de que el problema se traduzca en bajo rendimiento sumativo [6]. Su efectividad depende menos de la precisión del modelo predictivo, que en la literatura ya alcanza valores altos, y más de la capacidad de transformar esa predicción en acción pedagógica concreta [12].

1.3.5 Consideraciones éticas y diseño responsable

La integración de IA generativa en educación plantea consideraciones éticas que conviene abordar desde el diseño del sistema, no como agregado posterior. La Tabla 1.6 sintetiza los principales riesgos identificados en la literatura y las estrategias de mitigación propuestas.

Tabla 1.6. Principales riesgos y consideraciones éticas en el uso de IA generativa en educación

Dimensión	Riesgos identificados	Estrategias de mitigación
Sesgo algorítmico	Replicación de sesgos presentes en los datos de entrenamiento; discriminación en la retroalimentación generada.	Validación por expertos; auditorías periódicas; transparencia sobre las limitaciones del modelo [2].
Privacidad de datos	Exposición de información estudiantil sensible; almacenamiento inadecuado de interacciones en servidores externos.	Anonimización de datos; consentimiento informado; preferencia por modelos locales cuando sea posible [2, 5].
Integridad académica	Uso de la IA para fraude académico; generación de trabajos sin aprendizaje real.	Diseño de evaluaciones que requieran pensamiento crítico; preguntas de validación personalizadas [18, 4].
Dependencia excesiva	Reducción de habilidades de pensamiento crítico; atrofia de capacidades de resolución de problemas.	Uso supervisado; reverse prompting para promover reflexión; rol suplementario, no sustitutivo, de la IA [11, 16, 4].
Alucinación	Generación de información incorrecta presentada con alta confianza.	Esquema human-in-the-loop con revisión docente; validación estructural de las respuestas del modelo [3, 18].
Equidad de acceso	Brecha digital que limita los beneficios de la IA a contextos con mejor infraestructura.	Acceso institucional; uso de modelos open-source y servicios con tier gratuito [2, 18].

La alfabetización en IA (AI literacy) es una competencia que tanto docentes como estudiantes necesitan desarrollar para integrar estas herramientas de manera informada. La familiaridad declarada de los estudiantes universitarios con herramientas como ChatGPT es alta, pero la barrera relevante no es el desconocimiento sino la integración pedagógicamente responsable de la herramienta en el proceso formativo.

Sobre el lugar del docente frente a estos sistemas, dos posiciones convergentes orientan el diseño. Kasneci et al. [2] sostienen que los LLMs no deben reemplazar al docente sino funcionar como herramientas de asistencia bajo supervisión humana continua. Hernández-León y Rodríguez-Conde [1] concluyen, tras revisar literatura del periodo 2013-2023, que la IA no operará de manera autónoma en educación: la automatización analítica es válida, pero el docente conserva un papel insustituible en las dimensiones empática, ética y adaptativa de la enseñanza. Estas posiciones definen lo que en la literatura se denomina enfoque human-in-the-loop: un esquema en el que cada salida relevante del modelo pasa por revisión humana antes de incorporarse al flujo educativo.

En sistemas que combinan componentes con tiempos de operación distintos, este enfoque se materializa en arreglos diferenciados. Los componentes donde la salida del modelo puede esperar una decisión humana antes de su entrega, como la generación de material, la evaluación o la planificación, admiten revisión docente previa caso a caso. Los componentes que responden en tiempo real al estudiante o que operan de manera periódica sin intervención del docente requieren controles equivalentes pero de naturaleza estructural: límites codificados en el diseño del componente, validación automática de las salidas y registros que hagan posible una auditoría posterior. Ambos arreglos cumplen la función de mantener al docente como referente último de lo que el sistema entrega.

1.4 Objetivos

1.4.1 Objetivo general

Automatizar la tutoría docente en cursos universitarios utilizando la IA generativa supervisada.

1.4.2 Objetivos específicos

- Identificar las tecnologías a utilizar en la automatización de la tutoría docente en cursos universitarios utilizando la IA generativa.
- Implementar un sistema de tutoría docente basado en IA generativa supervisada.
- Medir el nivel de aceptación de la solución propuesta por parte de estudiantes y docentes de la carrera de Ingeniería de Software, de la Universidad Técnica de Ambato.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para el desarrollo del presente proyecto se emplearon cuatro instrumentos de recolección de información, organizados en función de las dos fases del trabajo: el diagnóstico que justifica la propuesta y la validación de la solución desarrollada.

La entrevista de diagnóstico se aplicó a docentes de la carrera de Software de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Este instrumento tuvo como propósito caracterizar el proceso vigente de retroalimentación académica, reconocer las limitaciones que los docentes enfrentan para identificar brechas de conocimiento en sus estudiantes y recoger su percepción sobre la integración de herramientas de Inteligencia Artificial en la práctica docente. La entrevista se estructuró en formato semiestructurado con preguntas abiertas, lo que permitió que los docentes profundizaran en aspectos relevantes desde su experiencia, sin restringir sus respuestas a categorías predefinidas.

Las fichas documentales se fundamentan en los antecedentes investigativos desarrollados en la sección 1.2, de los cuales se seleccionaron los estudios más relevantes para esta investigación. La obtención de información mediante esta técnica se realizó a través de la revisión de artículos científicos indexados en bases de datos académicas, principalmente Web of Science. Las fichas permitieron organizar de forma estructurada los hallazgos, conceptos clave, conclusiones y aportes de cada estudio revisado, lo que facilitó la identificación de patrones comunes en la literatura y de las decisiones de diseño que orientan el desarrollo del sistema propuesto. La Tabla 2.7 presenta la estructura empleada para la sistematización de la información.

Tabla 2.7. Formato de ficha de contenido documental

Campo	Descripción
Tema	Título o temática del estudio revisado
Año de publicación	Año en que fue publicado el estudio
Referencia bibliográfica	Referencia completa del artículo en formato IEEE
Ideas centrales	Hallazgos y planteamientos principales del estudio

Tabla 2.7. Formato de ficha de contenido documental (continuación)

Campo	Descripción
Conceptos clave	Términos y conceptos fundamentales abordados
Conclusiones	Conclusiones principales de los autores
Aporte al tema	Contribución del estudio a la presente investigación
Observaciones	Notas adicionales sobre alcances o limitaciones

El cuestionario System Usability Scale (SUS) se utilizó para evaluar la usabilidad percibida del sistema una vez transcurrido el periodo de prueba con los usuarios. El SUS es un instrumento estandarizado de diez ítems con escala Likert de cinco puntos, propuesto por Brooke [19] y consolidado como uno de los instrumentos más utilizados para la medición de usabilidad de productos de software. Para su aplicación en este trabajo se utilizó la versión adaptada al español por Sevilla-Gonzalez et al. [20], validada en población latinoamericana para la evaluación de herramientas electrónicas.

El cuestionario abierto a docentes se aplicó como instrumento de validación complementario al SUS, orientado a recoger la valoración cualitativa de los docentes sobre la propuesta. El cuestionario se administró tras una sesión individual de presentación del sistema, en la que cada docente conoció los tres componentes —tutor inteligente, retroalimentación al docente y retroalimentación al estudiante— y revisó ejemplos reales de retroalimentaciones generadas durante el periodo de prueba. El instrumento se estructuró en formato semiestructurado con preguntas abiertas, organizadas en torno a seis dimensiones de la experiencia docente sobre el sistema: utilidad pedagógica, claridad del contenido, confianza en la información, frecuencia y oportunidad, facilidad de acceso y consulta, y disposición a utilizar el sistema.

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación adopta un enfoque mixto, ya que combina datos cualitativos y cuantitativos en distintos momentos del trabajo. En la fase de diagnóstico se utiliza un componente cualitativo, basado en entrevistas a docentes y revisión documental, que permite comprender las condiciones del contexto educativo en el que se va a implementar la solución. La fase de validación combina ambos componentes: el cuestionario SUS aplicado a estudiantes entrega una medida cuantitativa de la usabilidad percibida sobre el sistema implementado, y el cuestionario abierto aplicado a docentes recoge una valoración cualitativa de la propuesta tras la presentación del sistema. Cada fase responde a una pregunta distinta: el diagnóstico busca caracterizar el problema desde la experiencia de los docentes y la validación busca medir el nivel de aceptación de la solución por parte de los dos actores principales del proceso educativo.

En cuanto a la modalidad, la investigación es aplicada, de campo y documental.

Es aplicada porque se orienta al diseño, desarrollo e implementación de una solución tecnológica concreta que responde a un problema educativo identificado en la FISEI-UTA: la ausencia de un sistema de tutoría docente que ofrezca retroalimentación continua y ajustada al contenido de cada asignatura. Este carácter aplicado se materializa en la construcción de un sistema funcional que integra técnicas de Inteligencia Artificial Generativa con principios pedagógicos, evaluado posteriormente con usuarios reales de la carrera.

Es de campo porque los datos se obtuvieron directamente de los actores del proceso educativo y no únicamente de fuentes secundarias. Las entrevistas con los docentes permitieron recoger información de primera mano sobre las limitaciones que enfrentan para ofrecer acompañamiento individualizado, los mecanismos de retroalimentación que utilizan actualmente y las dificultades para detectar brechas de conocimiento de forma temprana. La aplicación del SUS a estudiantes que utilizaron el sistema durante el periodo de prueba recoge directamente la valoración de los usuarios sobre la solución implementada.

Es documental porque se sustenta en una revisión de literatura científica en bases de datos académicas, principalmente Web of Science, sistematizada mediante fichas

documentales que permitieron identificar el estado del arte en sistemas educativos basados en IA generativa, las técnicas de implementación disponibles, las metodologías de evaluación empleadas en estudios previos y los principios pedagógicos aplicables.

El alcance de la investigación combina elementos descriptivos y exploratorios. Es descriptiva porque busca caracterizar las necesidades de retroalimentación académica, las prácticas docentes vigentes y las percepciones de los docentes respecto a la integración de IA en la enseñanza. Es exploratoria porque la integración de Grandes Modelos de Lenguaje en sistemas de tutoría aplicados al contenido específico de cada curso es un campo de investigación reciente, sobre el cual los antecedentes consultados durante este trabajo no incluyeron experiencias documentadas en facultades de ingeniería del país. Por esta razón, los hallazgos del presente trabajo aportan evidencia local sobre la viabilidad y los desafíos de implementar este tipo de soluciones en una institución pública de educación superior.

2.2.2 Población y muestra

Población. La población del estudio se delimitó en función de las dos fases del trabajo: el diagnóstico que justifica la propuesta y la validación de la solución implementada.

Para la fase de diagnóstico, la población está conformada por los 25 docentes de la carrera de Software de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato que imparten asignaturas durante el período académico febrero–julio 2026. Se consideró a los docentes como fuente primaria de información porque son quienes experimentan de forma directa las limitaciones del proceso de retroalimentación académica: conocen el volumen de estudiantes que atienden, las herramientas de las que disponen y las dificultades para dar seguimiento individualizado.

Para la fase de validación, la población se delimitó a los 282 estudiantes de la carrera de Software matriculados durante el mismo período académico en semestres intermedios, esto es, del tercero al octavo. Esta delimitación responde a un criterio de pertinencia respecto al uso del sistema. Los estudiantes de primero y segundo semestre se encuentran en la fase inicial de la carrera, cursando asignaturas de formación básica, y todavía no enfrentan de forma directa el problema de retroalimentación que la

propuesta atiende. Los de noveno semestre, por su parte, están en la etapa terminal de la carrera, dedicados principalmente a prácticas preprofesionales y al trabajo de titulación, con una carga de asignaturas regulares reducida. En ambos casos, su exposición a los cursos sobre los que el sistema opera durante el periodo de prueba es marginal, por lo que su inclusión no aportaría una valoración basada en uso real. La población queda así conformada por los estudiantes que se encuentran efectivamente en condiciones de utilizar el sistema como apoyo a su proceso de aprendizaje.

Muestra. La fase de diagnóstico se planteó como una indagación exploratoria orientada a identificar las dimensiones del problema desde la experiencia docente y fundamentar el diseño de la solución propuesta. Bajo esa lógica, se trabajó con una muestra de 5 docentes seleccionados mediante muestreo intencional, con tres criterios de selección: (a) docentes activos durante el período académico vigente, (b) cobertura de distintas áreas curriculares de la carrera, de modo que las respuestas no quedarán limitadas a la perspectiva de un solo campo de conocimiento, y (c) experiencia en asignaturas con componentes teóricos y con componentes prácticos, dado que las necesidades de retroalimentación difieren entre ambas.

Para la fase de validación de los resultados obtenidos se aplicó el cuestionario SUS a una muestra de estudiantes calculada mediante la fórmula propuesta por López [21] para poblaciones finitas:

$$n = \frac{N}{(N - 1) \cdot K^2 + 1} \quad (2.1)$$

donde $N = 282$ es el tamaño de la población estudiantil y K el margen de error aceptado. Se adoptó $K = 0,05$, valor que corresponde al límite inferior del rango habitual en investigación social aplicada y que ofrece un nivel de precisión adecuado para una validación de usabilidad. Aplicando la fórmula:

$$n = \frac{282}{(281)(0,05)^2 + 1} = \frac{282}{2,175} \approx 166 \quad (2.2)$$

El tamaño de la muestra calculado dio como resultado 166 encuestados. La selección se realizó entre los estudiantes de los cursos que participaron del periodo de prueba, durante el cual el sistema estuvo disponible varias semanas. Antes de aplicar el cuestionario se presentó la herramienta a los participantes, de modo que todos contaran

con una referencia común sobre sus funcionalidades al momento de responder.

Para la fase de validación con docentes se trabajó con los cinco docentes que participaron del diagnóstico. La consistencia entre las dos fases del trabajo es deliberada: los docentes que aportaron la caracterización del problema en la fase de diagnóstico son los mismos que valoran la solución desarrollada, lo que permite una lectura de la propuesta desde la formulación hasta su validación.

2.2.3 Metodología de desarrollo de software

El desarrollo del sistema requirió un marco metodológico que ordenara el trabajo colaborativo de un equipo multidisciplinario y permitiera entregar incrementos funcionales en plazos cortos, dado que la fase de validación con docentes y estudiantes requería contar con una versión utilizable del sistema dentro del periodo académico. Antes de adoptar un marco específico, se contrastaron las opciones más extendidas en desarrollo ágil contra los requisitos del proyecto.

a) Marcos ágiles candidatos

Tabla 2.8. Comparación de marcos ágiles considerados para el desarrollo del sistema

Marco	Características principales	Adecuación al proyecto
Scrum	Sprints de duración fija, roles definidos (Product Owner, Scrum Master, Development Team), reuniones regulares (planning, daily, review, retrospective), backlog priorizado.	Aporta cadencia y disciplina de iteración. Asume un Development Team sin jerarquía interna, lo que no se ajusta del todo a un equipo con líderes técnicos por área.
Kanban	Tablero visual con columnas de flujo (por hacer, en progreso, hecho), límite de trabajo en progreso (WIP), entrega continua sin sprints fijos.	Aporta visibilidad del estado del trabajo, pero la ausencia de cadencia fija dificulta sincronizar la entrega de incrementos con los plazos académicos.
Extreme Programming (XP)	Énfasis en prácticas técnicas: programación en parejas, integración continua, desarrollo guiado por pruebas, refactorización constante.	Sus prácticas exigen un nivel de trabajo presencial conjunto y disponibilidad que no era viable con un equipo distribuido entre clases y otras responsabilidades.

Tabla 2.8. Comparación de marcos ágiles considerados para el desarrollo del sistema (continuación)

Marco	Características principales	Adecuación al proyecto
Scrumban	Combinación de Scrum y Kanban: conserva la cadencia y roles de Scrum, e incorpora el tablero visual y el flujo continuo de Kanban.	Mantiene la disciplina de iteraciones y reuniones de revisión, y al mismo tiempo permite gestionar el flujo de tareas con un tablero visual flexible, sin imponer reuniones diarias que el equipo no podía sostener.

b) Marco seleccionado

Se adoptó Scrumban como marco de trabajo. La decisión respondió a tres consideraciones. Primero, el equipo necesitaba la cadencia de sprints para sincronizar entregas con el calendario académico y reservar ventanas estables para validación. Segundo, la gestión del trabajo se realizaba sobre un tablero visual de tareas, dado que los miembros tenían distinta disponibilidad según sus responsabilidades y conviene que el estado del trabajo sea visible para todo el equipo. Tercero, las reuniones diarias propias de Scrum estricto no eran sostenibles para un equipo de estudiantes con distinta disponibilidad, mientras que las reuniones semanales sí lo eran. Scrumban ofrece exactamente ese punto medio: estructura suficiente para mantener la disciplina iterativa, sin la rigidez completa de Scrum.

c) Composición del equipo y roles

El equipo de desarrollo estuvo conformado por siete personas con roles diferenciados según su responsabilidad técnica y funcional.

Tabla 2.9. Roles del equipo de desarrollo

Rol	Responsabilidad
Product Owner	Priorización del backlog, definición de criterios de aceptación, validación funcional de las entregas.
Scrum Master	Facilitación de las reuniones, gestión de impedimentos, mantenimiento de la dinámica del equipo.

Tabla 2.9. Roles del equipo de desarrollo (continuación)

Rol	Responsabilidad
Líder de backend	Diseño e implementación del backend principal, integración con servicios de IA, definición de la API.
Líder de frontend	Diseño e implementación de la interfaz de usuario, integración con la API del backend.
Desarrollador de videos	Implementación del módulo de ingesta y procesamiento de videos como objetos de aprendizaje.
Desarrolladores de frontend (2)	Implementación de componentes y vistas asignadas en cada sprint.
Desarrollador de backend	Implementación de funcionalidades complementarias del backend asignadas en cada sprint.

La distribución por áreas responde a la naturaleza técnica del proyecto, que combina un backend con integración de servicios de IA, un frontend con un editor de contenido enriquecido, y un componente de procesamiento de video.

d) Cadencia y reuniones del equipo

El equipo trabajó con sprints semanales. Esta duración se eligió por dos razones: permite que un cambio en el alcance o una observación del Product Owner se incorpore al sprint siguiente sin grandes demoras, y se alinea con la disponibilidad académica del equipo, que organiza sus actividades por semana.

Al cierre de cada sprint se realizaron dos actividades, ambas dentro de una misma reunión:

- **Sprint review**, donde se presentaron los incrementos terminados al Product Owner. La revisión sirvió para validar funcionalmente lo entregado, recoger observaciones y ajustar criterios de aceptación cuando el resultado lo justificaba.
- **Sprint retrospective**, donde el equipo identificó lo que había funcionado bien durante el sprint y lo que convenía mejorar para el siguiente. La retrospectiva permitió ajustar progresivamente la forma de trabajo, incluyendo la planificación de cada sprint, la asignación de tareas y la coordinación entre las áreas de

backend y frontend.

La planificación del siguiente sprint también se realizaba en esta reunión, con la incorporación de los elementos priorizados del backlog y la estimación informal del esfuerzo por parte de quienes asumían cada tarea.

e) Gestión del backlog y del flujo de trabajo

El backlog del producto y el tablero de tareas se gestionaron mediante Trello, herramienta seleccionada por su simplicidad, su disponibilidad gratuita para equipos pequeños y su soporte para los flujos visuales propios de Kanban.

El tablero se organizó en columnas que reflejaban el flujo del trabajo desde su definición hasta su entrega: una columna de elementos pendientes, una de tareas comprometidas para el sprint en curso, una de trabajo en progreso, una de tareas en revisión y una de tareas terminadas. Cada tarjeta describió una unidad funcional con sus criterios de aceptación, y se etiquetó según el área responsable y la prioridad asignada por el Product Owner.

La Figura 2.1 muestra el tablero de Trello utilizado durante el desarrollo del proyecto.

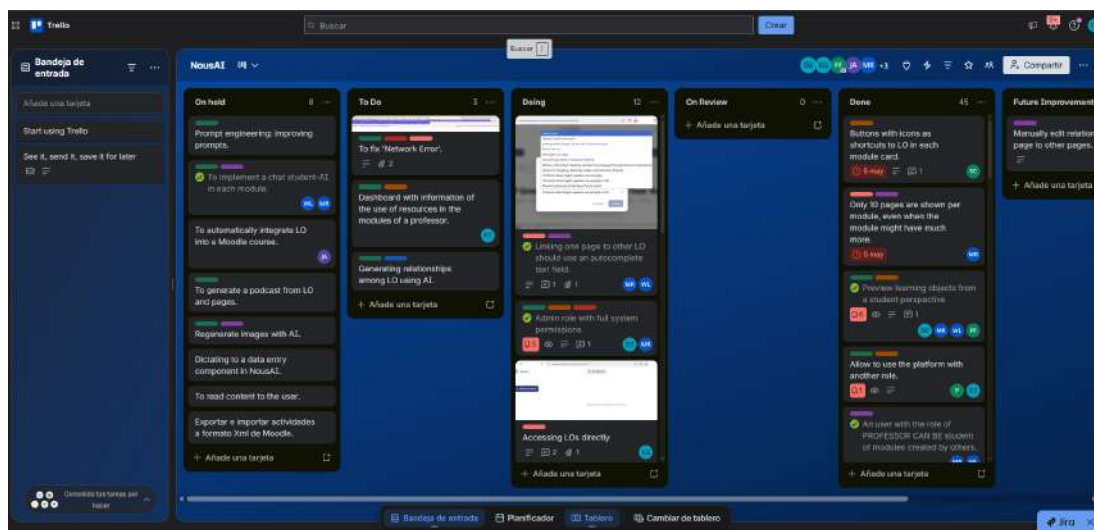


Figura 2.1. Tablero de Trello utilizado para la gestión del backlog y del flujo de trabajo del proyecto.

El uso del tablero permitió que cualquier integrante del equipo pudiera verificar en cualquier momento el estado del sprint en curso, identificar dependencias entre tareas y detectar bloqueos de manera temprana.

2.2.4 Recolección de información

La recolección de información se realizó mediante cuatro técnicas distribuidas entre las dos fases del trabajo. Las entrevistas de diagnóstico y las fichas documentales se aplicaron en la fase de diagnóstico, con el propósito de caracterizar el problema y fundamentar la pertinencia del desarrollo del sistema. El cuestionario SUS y el cuestionario abierto a docentes se aplican en la fase de validación, una vez transcurrido el periodo de prueba del sistema, para medir la usabilidad y la percepción por parte de los usuarios finales.

a) Entrevista de diagnóstico a docentes

Propósito: Comprender la experiencia de los docentes respecto a la retroalimentación académica que brindan y reciben, identificar las fuentes a las que recurren sus estudiantes para complementar el proceso formativo, y recoger su percepción sobre la integración de herramientas de Inteligencia Artificial Generativa en la práctica docente.

Formato: Entrevista individual presencial con grabación de audio previa autorización del participante.

Aplicación: Realizada a 5 docentes de la carrera de Software durante la segunda semana del período académico.

Estructura: 6 preguntas abiertas organizadas en dos bloques: situación actual de la retroalimentación en el aula (preguntas 1 a 3) y percepción sobre la IA generativa en la práctica docente (preguntas 4 a 6). Duración aproximada: 10 a 20 minutos por entrevista.

Validación del instrumento: La guía de entrevista fue revisada y validada por el tutor del trabajo antes de su aplicación, en su calidad de docente investigador con experiencia en el área. La revisión verificó la pertinencia y claridad de las preguntas respecto a los objetivos del diagnóstico, y los ajustes señalados se incorporaron en la versión final aplicada.

La síntesis de las respuestas obtenidas se presenta a continuación.

Pregunta 1: Dada su experiencia como docente, ¿qué instrumentos de retroalimentación con el estudiante ha utilizado?

Tabla 2.10. Resultados de la Pregunta 1 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Indica que no utiliza instrumentos formales de retroalimentación. Recurre a ejercicios de refuerzo y a bibliografía complementaria cuando observa que un tema no quedó claro. Menciona que ha empezado a apoyarse en IA generativa para producir ejemplos adicionales para sus estudiantes.
Docente 2	Describe que el principal mecanismo de retroalimentación es la consulta sobre temas previos al inicio de cada clase, aprovechando la secuencia lógica entre contenidos. Plantea que el problema actual es que los estudiantes consultan en internet o piden resúmenes a la IA generativa sin entender el contexto y sin capacidad de validar la respuesta.
Docente 3	Combina preguntas en clase, tutorías académicas y comunicación a través de herramientas institucionales de Microsoft, modalidad que adoptó durante la pandemia y que mantiene como canal de acompañamiento durante la semana. Menciona también los foros de la plataforma institucional, aunque señala que los estudiantes prácticamente no los utilizan.
Docente 4	Distingue dos instrumentos. El primero es la retroalimentación en el sitio, durante la realización de trabajos o exposiciones. El segundo son las observaciones registradas en la plataforma institucional tras la evaluación de trabajos.
Docente 5	Combina retroalimentación oral durante el desarrollo de actividades en clase con observaciones escritas en las entregas. Indica que ha intentado dar retroalimentación más sistemática, pero el tiempo disponible fuera del aula es la principal limitación.

Análisis: Las respuestas muestran que los instrumentos formales de retroalimentación son escasos en la práctica cotidiana. Solo dos de los cinco docentes describen instrumentos sistemáticos vinculados a la plataforma institucional o a herramientas digitales del aula. Los demás se apoyan en mecanismos informales: preguntas en clase, ejercicios de refuerzo, conversaciones puntuales. Ningún docente menciona un canal estructurado de retroalimentación disponible para el estudiante fuera del horario de clases. El tiempo y el volumen de estudiantes aparecen como limitaciones recurrentes que impiden sostener un acompañamiento individual continuo.

Pregunta 2: ¿Cómo recibe Ud. retroalimentación del proceso de enseñanza / aprendizaje?

Tabla 2.11. Resultados de la Pregunta 2 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Identifica dos vías. Por un lado, los propios estudiantes le señalan en qué temas necesitan refuerzo. Por otro, la información proviene de los talleres, laboratorios y pruebas, donde detecta qué contenidos no quedaron asentados. Reconoce que buena parte de esta lectura se apoya en su experiencia para identificar lo que hace falta profundizar.
Docente 2	Recibe retroalimentación principalmente a través de las preguntas que formula al inicio de cada clase sobre los contenidos vistos previamente, aprovechando la secuencia lógica entre temas. Las evaluaciones constituyen la segunda fuente.
Docente 3	Su respuesta remite a los instrumentos descritos en la pregunta anterior: las preguntas en clase, las tutorías y el contacto a través de las herramientas institucionales operan como canales de retroalimentación tanto hacia el estudiante como hacia él mismo.
Docente 4	Combina tres fuentes: preguntas durante la explicación, revisión de trabajos y evaluaciones puntuales sobre el tema visto en la sesión. Estas últimas le permiten verificar si el contenido fue comprendido en el momento, no solo al cierre del periodo.
Docente 5	Su principal fuente es la observación en clase y la corrección de actividades. Indica que entre evaluaciones tiene escasa visibilidad de qué estudiantes están comprendiendo y cuáles no, especialmente los que no participan activamente.

Análisis: Las cinco respuestas convergen en un mismo patrón. La retroalimentación que los docentes reciben proviene de tres canales: los resultados de evaluaciones, las preguntas formuladas en clase y la observación directa del comportamiento del grupo. Los tres comparten una limitación común: son tardíos o dependen de qué estudiantes participan activamente. La consecuencia recurrente es que los estudiantes que no se manifiestan quedan fuera del radar hasta que aparecen los resultados de una evaluación. La visibilidad pedagógica del docente sobre el avance individual de su grupo es, en la práctica, intermitente y reactiva.

Pregunta 3: ¿De qué otras fuentes considera Ud. que los estudiantes reciben retroalimentación?

Tabla 2.12. Resultados de la Pregunta 3 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Considera que la IA generativa se ha convertido en la principal fuente externa. Anteriormente los estudiantes recurrían a YouTube para la parte práctica y a portales web para la parte teórica. En la actualidad, observa que los estudiantes preguntan directamente a la IA y se quedan con la respuesta que reciben.
Docente 2	Identifica tres fuentes: los propios compañeros, otros profesores y la IA generativa. No menciona otras fuentes adicionales.
Docente 3	Menciona la información que los estudiantes consultan en internet y en buscadores como Google, así como el uso actual de IA generativa. Observa que la IA se utiliza sin control, en muchos casos para que resuelva los trabajos académicos por el estudiante.
Docente 4	Plantea que la revisión entre pares es una fuente importante de retroalimentación, especialmente en componentes técnicos. Considera que los propios compañeros funcionan como jueces que pueden señalar al estudiante qué está haciendo bien y qué debe mejorar.
Docente 5	Considera que la retroalimentación que el estudiante recibe fuera del aula es mayor en volumen que la que recibe dentro. La IA generativa, en particular, aparece como un canal que responde rápido a cualquier consulta, sin que el docente sepa cómo se está usando.

Análisis: Los docentes identifican tres tipos de fuentes externas: compañeros de estudio, recursos en línea (buscadores, videos, portales) y herramientas de IA generativa. La IA aparece de manera explícita en cuatro de los cinco relatos como una fuente que ha crecido rápidamente y cuya forma de uso escapa al control del docente. La observación recurrente no es que el uso de IA sea negativo en sí mismo, sino que ocurre sin orientación: sin verificación de la respuesta, sin alineación con el contenido del curso y sin que el docente tenga visibilidad de las dudas concretas que los estudiantes están resolviendo por esta vía.

Pregunta 4: ¿Qué opinión tiene del uso de la IA generativa en el ámbito de las asignaturas que Ud. ha impartido?

Tabla 2.13. Resultados de la Pregunta 4 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Considera que la IA generativa llegó para quedarse y que la postura adecuada es adaptarse. Plantea como punto central el uso ético de la herramienta, dado que la calidad de la respuesta depende del prompt formulado y que muchos modelos siguen produciendo alucinaciones cuando se les consulta de manera muy general. Reconoce que falta camino por recorrer para que el uso sea beneficioso tanto para estudiantes como para docentes.
Docente 2	Su valoración es favorable pero condicionada: la IA es útil si se usa con criterio. Le preocupa que las respuestas rápidas eliminen la capacidad de análisis y de comprensión de los estudiantes. Observa que cuando se les pide resolver ejercicios sin recurrir a la IA, muchos no logran hacerlo.
Docente 3	Considera que es algo inevitable y con lo que la docencia debe coexistir. Plantea como condición fundamental que el uso sea lo más ético posible y que la herramienta no reemplace la capacidad de razonamiento. Distingue entre usar la IA como apoyo informado y delegar en ella tareas que corresponden al proceso de aprendizaje.
Docente 4	Tiene una opinión dividida. Reconoce el potencial de la herramienta para facilitar el trabajo, pero observa que el uso como soporte total puede impedir el aprendizaje real. La describe como un arma de doble filo: ayuda cuando se acompaña con criterio, perjudica cuando reemplaza el esfuerzo de comprender.
Docente 5	Considera que la IA generativa es una herramienta que llegó para quedarse y que la pregunta relevante no es si usarla, sino cómo integrarla de forma responsable. Le interesa especialmente la posibilidad de que el uso ocurra en un entorno alineado con el contenido del curso, en lugar de en plataformas genéricas.

Análisis: Las opiniones de los cinco docentes son moderadamente favorables, todas condicionadas. Tres condiciones aparecen de manera recurrente: que el uso sea ético y acompañado de criterio, que la herramienta no reemplace el esfuerzo de comprensión del estudiante, y que el contenido al que accede esté alineado con el enfoque del curso. Ninguno de los docentes manifiesta rechazo a la IA generativa como tecnología; lo que cuestionan es el uso sin orientación. Esta lectura coincide con la postura predominante en la literatura revisada en el Capítulo I, que distingue entre el uso libre de los LLMs (asociado a riesgos) y el uso supervisado e integrado (asociado a

beneficios).

Pregunta 5: De utilizar una herramienta tutor con IA generativa de apoyo al proceso de enseñanza-aprendizaje en sus materias, ¿cuáles serían las funcionalidades más significativas que considera debería tener dicha herramienta para el docente y para el estudiante?

Tabla 2.14. Resultados de la Pregunta 5 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Considera fundamental que la herramienta ayude al estudiante a comprender la respuesta que la IA devuelve, en lugar de limitarse a copiarla. Plantea que el valor está en analizar cada respuesta para resolver mejor los problemas de cada asignatura, no en obtener una solución lista para usar.
Docente 2	Plantea que la IA debe operar como una guía, no como un recurso que entregue la respuesta resuelta. Considera que cuando el sistema da todo resuelto, el estudiante pierde la capacidad de hacerlo por sí mismo.
Docente 3	Plantea cuatro funcionalidades. Para el estudiante: que el sistema genere contenido textual revisado por el docente y que las respuestas se produzcan a partir de ese contenido, no de la base general del modelo. Para el docente: que la IA le retroalimente sobre los aspectos más consultados por los estudiantes, función especialmente relevante cuando se gestionan grupos numerosos donde no es posible hacer ese análisis manualmente. Considera también valioso que el estudiante reciba retroalimentación sobre lo que le hace falta a partir de su propia actividad en la plataforma.
Docente 4	Plantea funcionalidades orientadas a trabajos técnicos: revisión automática de código contra requisitos funcionales, generación de rúbricas a partir de una tarea, y comentarios directos sobre qué está fallando y qué hay que mejorar.
Docente 5	Considera que las funcionalidades clave son tres: para el estudiante, asistencia conversacional alineada con el curso y un resumen periódico sobre su propio desempeño; para el docente, un retorno consolidado de las interacciones del grupo que permita tomar decisiones pedagógicas basadas en datos.

Análisis: Las funcionalidades mencionadas espontáneamente por los docentes se agrupan en tres líneas. La primera se refiere a la asistencia conversacional al estudiante, mencionada por todos: respuestas alineadas con el contenido del curso, un comportamiento que privilegie la explicación y la guía sobre la respuesta resuelta, y la conserva-

ción de la capacidad de análisis del estudiante. La segunda se refiere a la información que retorna al docente, mencionada por los Docentes 3 y 5: temas más consultados, patrones de interacción del grupo, información que oriente las decisiones pedagógicas. La tercera, mencionada por los Docentes 3 y 5, se refiere a la entrega al estudiante de una retroalimentación periódica sobre su propio proceso de aprendizaje a partir de su actividad en la plataforma. La convergencia entre las funcionalidades sugeridas y los componentes que el sistema desarrollado incorpora aporta evidencia de que la propuesta responde a necesidades reconocidas por los propios usuarios.

Pregunta 6: De utilizar una herramienta tutor con IA generativa de apoyo al proceso de enseñanza-aprendizaje en sus materias, ¿qué preocupaciones o reservas tendría sobre su uso?

Tabla 2.15. Resultados de la Pregunta 6 a Docentes

Entrevistado	Síntesis de la respuesta
Docente 1	Su principal preocupación es que los estudiantes desarrollen una dependencia excesiva de la herramienta y dejen de ejercer el pensamiento crítico. Observa que el conformismo con la primera respuesta recibida puede llevar a perder la capacidad de debatir, discernir y decidir qué respuesta es la más adecuada.
Docente 2	Le preocupa que la herramienta facilite tanto las cosas que los estudiantes dejen de pensar y de analizar. Considera que si el sistema termina resolviendo todo, pierde su propósito formativo.
Docente 3	Identifica dos preocupaciones. La primera es que la IA se aleje del contenido que el docente está manejando o utilice términos distintos a los empleados en clase, lo que puede generar confusión. La segunda es que el sistema produzca retroalimentación poco útil, contenido por contenido, en lugar de devolver al estudiante o al docente información que aporte realmente al proceso.
Docente 4	Le preocupa que la herramienta presente sesgos o produzca alucinaciones. Particularmente le inquieta el caso de funcionalidades de calificación, donde una equivocación de la IA podría asignar al estudiante una nota que no corresponde a su desempeño.
Docente 5	Su mayor preocupación es la calidad y veracidad de las respuestas, y la heterogeneidad en el uso entre estudiantes: aquellos que aprovechen el sistema productivamente frente a otros que lo usen únicamente para resolver tareas. Considera que el sistema debería pasar por un proceso de validación docente antes de ponerse a disposición de los estudiantes.

Análisis: Las preocupaciones expresadas se agrupan en tres temas. El primero es

la dependencia excesiva del estudiante respecto al sistema y la consecuente pérdida de capacidad de análisis, mencionada por los Docentes 1, 2 y 5. El segundo es la calidad y veracidad de las respuestas, particularmente el riesgo de alucinaciones, mencionado por los Docentes 1, 4 y 5. El tercero es la alineación con el contenido del curso, planteado por el Docente 3 en dos dimensiones: que la IA no se aleje del material del docente y que la retroalimentación que produce sea efectivamente útil. Este conjunto de preocupaciones coincide con la literatura revisada en el Capítulo I, en particular con los hallazgos sobre uso no supervisado de LLMs [4] y con la posición sobre supervisión humana continua [2].

b. Fichas documentales

Las fichas documentales se fundamentan en los antecedentes investigativos desarrollados en la sección 1.2, de los cuales se seleccionaron los tres estudios más relevantes para esta investigación. La selección responde a tres criterios: pertinencia directa con las variables del estudio, disponibilidad de resultados cuantitativos que respaldan la viabilidad de la solución propuesta, y representatividad de las principales líneas de investigación identificadas en la revisión de literatura. Las fichas se presentan a continuación utilizando la estructura descrita en la sección 2.1.

Tabla 2.16. Ficha de Contenido N° 1

Campo	Contenido
Tema	Revisión sistemática de características, aplicaciones y métodos de evaluación de los Sistemas de Tutoría Inteligente.
Año de publicación	2021
Referencia bibliográfica	Mousavinasab, E., Zarifsanaiey, N., Niakan Kalhori, S. R., Rakhshan, M., Keikha, L. y Ghazi Saeedi, M. — “Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods.” <i>Interactive Learning Environments</i> , vol. 29, pp. 142–163.
Ideas centrales	Los ITS combinan técnicas de IA con metodologías educativas para crear entornos de aprendizaje adaptativos. La revisión de 53 artículos (2007–2017) identificó que las técnicas más frecuentes fueron el razonamiento basado en reglas (33.96 %), la minería de datos (22.64 %) y las redes bayesianas (20.75 %). El área de Ciencias de la Computación concentra el 37.73 % de las implementaciones, y el 54.71 % de los ITS utilizaron interfaces web.

Tabla 2.16. Ficha de Contenido N° 1 (continuación)

Campo	Contenido
Conceptos clave	Arquitectura clásica de 4 módulos del ITS (Módulo Experto, Módulo de Diagnóstico del Estudiante, Módulo Pedagógico, Módulo de Interfaz de Usuario). Modelo del estudiante construido con nivel de conocimiento (62.26 %), rendimiento de aprendizaje (52.83 %) y comportamiento del alumno como variables principales.
Conclusiones	La mayoría de los ITS utilizan combinaciones de características del estudiante para construir sus modelos, lo que mejora la personalización. El procesamiento de lenguaje natural representó solo el 11.32 % de las técnicas empleadas, evidenciando que los ITS tradicionales no aprovechaban plenamente las capacidades de generación de lenguaje.
Aporte al tema	Documenta la arquitectura modular consolidada en sistemas de tutoría previos a la IA generativa, lo que permite identificar la diferencia funcional con los sistemas conversacionales basados en LLMs y comprender cómo estos últimos modifican la forma en que se materializa cada componente. Justifica además la pertinencia del área de Ingeniería de Software como campo de aplicación, dado que es el dominio con mayor prevalencia entre los sistemas revisados.
Observaciones	La revisión abarca hasta 2017, previo a la irrupción de los LLMs. Los hallazgos son útiles como línea base para contrastar con las capacidades actuales de la IA generativa. Referencia IEEE [13].

Tabla 2.17. Ficha de Contenido N° 2

Campo	Contenido
Tema	Viabilidad de la IA generativa como asistente de retroalimentación educativa mediante prompt engineering y few-shot learning.
Año de publicación	2024
Referencia bibliográfica	Wan, T. y Chen, Z. — “Exploring generative AI assisted feedback writing for students’ written responses to a physics conceptual question with prompt engineering and few-shot learning.” Physical Review Physics Education Research, vol. 20.
Ideas centrales	Se utilizó GPT-3.5 como asistente de calificación en un curso introductorio de Física con alta matrícula. Mediante prompt engineering y few-shot learning con solo 4 ejemplos, el 70 % del feedback generado requirió modificación mínima o ninguna por parte de los instructores (calificación promedio 2.06/3). Los estudiantes percibieron el feedback de la IA como más útil que el elaborado manualmente por los docentes, debido a su mayor nivel de detalle y consistencia.

Tabla 2.17. Ficha de Contenido N° 2 (continuación)

Campo	Contenido
Conceptos clave	Prompt engineering: proceso de desarrollo y refinamiento de instrucciones para obtener la respuesta deseada del LLM. Few-shot learning: técnica que permite al modelo aprender patrones a partir de pocos ejemplos incorporados en el prompt. System prompt: instrucción de sistema que incluye el contexto del profesor, la pregunta, la respuesta experta ideal y las preconcepciones comunes a evitar.
Conclusiones	La IA generativa es viable como asistente de retroalimentación, pero los autores recomiendan que el docente siempre revise y apruebe los mensajes antes de entregarlos a los estudiantes. La categorización manual de respuestas para construir los ejemplos del prompt consume tiempo, lo que limita la escalabilidad sin automatización adicional.
Aporte al tema	Respalda el principio de revisión docente sobre el contenido generado por el LLM, principio que el sistema desarrollado adopta como decisión de diseño. Demuestra que con técnicas accesibles —prompt engineering y few-shot learning— se puede lograr retroalimentación de calidad utilizable sin necesidad de fine-tuning costoso.
Observaciones	El estudio se limitó a una sola pregunta conceptual y a evaluadores con conocimiento previo en IA. Se recomienda probar el enfoque en diálogos conversacionales (chatbots), que es precisamente lo que aborda esta investigación. Referencia IEEE [3].

Tabla 2.18. Ficha de Contenido N° 3

Campo	Contenido
Tema	Optimización de sistemas educativos automatizados mediante IA para la generación adaptativa de sílabos y evaluación precisa en Ecuador.
Año de publicación	2024
Referencia bibliográfica	Villegas-Ch, W., Govea, J., Albuja, R., Buenaño-Fernández, D. y Mera-Navarrete, A. — “Optimizing Automated Educational Systems: Adaptive Syllabus Generation and Precise Assessment Using Artificial Intelligence.” IEEE Access, vol. 12, pp. 173390–173409.

Tabla 2.18. Ficha de Contenido N° 3 (continuación)

Campo	Contenido
Ideas centrales	Se desarrolló una plataforma web adaptativa basada en la API de OpenAI (GPT-3) y modelos BERT para la generación de sílabos y la evaluación automatizada de trabajos académicos. La contextualización del LLM se logró mediante un algoritmo que extrajo criterios de las rúbricas docentes usando similitud del coseno. Tras un proceso iterativo de seis meses centrado en la retroalimentación de usuarios, la precisión de evaluación aumentó del 78.5 % al 89.7 %, las calificaciones medias subieron de 70.4 a 75.2 ($t = 5.12$, $p < 0.001$, $d = 0.51$) y la usabilidad mejoró un 21.1 %.
Conceptos clave	Similitud del coseno (Cosine Similarity): métrica para medir la alineación semántica entre el contenido evaluado y los criterios de la rúbrica. Procesamiento asíncrono: uso de Redis y Celery para gestionar colas de tareas de IA sin bloquear la interfaz. Arquitectura distribuida escalable desplegada en AWS con PostgreSQL.
Conclusiones	Un sistema educativo basado en IA, construido con un enfoque iterativo centrado en el usuario, no solo mejora técnicamente (precisión) sino que eleva el desempeño académico real de los estudiantes. La precisión se vio comprometida en casos que requieren alta interpretación contextual, lo que refuerza la necesidad de supervisión docente.
Aporte al tema	Constituye el único antecedente reportado en Ecuador de un sistema educativo basado en la API de OpenAI con resultados cuantitativos publicados. Valida la viabilidad técnica y pedagógica de implementar soluciones similares en el contexto universitario latinoamericano. Comparte decisiones tecnológicas con el presente proyecto (OpenAI API, PostgreSQL, procesamiento asíncrono).
Observaciones	El sistema se enfocó en generación de sílabos y evaluación, no en tutoría conversacional sobre el contenido específico de una lección. Referencia IEEE [5].

c) Cuestionario System Usability Scale (SUS)

Propósito: Medir la usabilidad percibida del sistema una vez transcurrido el periodo de prueba, recogiendo la valoración de los estudiantes de la carrera de Software que participaron de él.

Formato: Cuestionario estandarizado de 10 ítems con escala Likert de 5 puntos, donde 1 corresponde a “Totalmente en desacuerdo” y 5 a “Totalmente de acuerdo”. Los ítems alternan entre afirmaciones positivas (impares) y negativas (pares), lo que reduce

el sesgo de respuesta. Se utiliza la versión adaptada al español por Sevilla-Gonzalez et al. [20]. La aplicación se realiza en formato digital mediante un formulario en línea para facilitar la recolección y el procesamiento posterior.

Aplicación: El cuestionario se aplica una vez finalizado el periodo de prueba, una sola vez por participante. El sistema estuvo disponible para los estudiantes durante varias semanas previas; no obstante, dado que el grado de uso efectivo pudo variar de un participante a otro, antes de responder el cuestionario se les presentó la herramienta y sus funcionalidades, asegurando una base común de conocimiento del sistema sobre la cual emitir su valoración.

Estructura: Los 10 ítems del SUS evalúan distintas dimensiones de la usabilidad, entre ellas la facilidad de uso, la consistencia de la interfaz, la curva de aprendizaje, la confianza al usar el sistema y la disposición a seguir utilizándolo.

Justificación de la elección del instrumento: El SUS fue seleccionado por tres razones. Primero, es uno de los instrumentos de evaluación de usabilidad más utilizados en investigación [19], lo que permite comparar los resultados obtenidos con valores de referencia documentados en la literatura. Segundo, su brevedad facilita la aplicación a un grupo amplio de participantes sin generar fatiga, condición relevante considerando que se aplicará a una muestra de 166 estudiantes. Tercero, su carácter estandarizado reduce la subjetividad del análisis, ya que los puntajes se calculan mediante una fórmula establecida que entrega un valor único entre 0 y 100.

Procesamiento: El cálculo del puntaje SUS sigue el procedimiento estándar. Para los ítems impares (afirmaciones positivas), se resta 1 al valor seleccionado por el participante. Para los ítems pares (afirmaciones negativas), se resta el valor seleccionado a 5. La suma de los 10 valores ajustados se multiplica por 2.5, lo que entrega un puntaje final en una escala de 0 a 100. Los puntajes individuales se promedian para obtener la valoración global del grupo. La interpretación de los resultados se basa en los rangos establecidos en la literatura, donde puntajes superiores a 68 se consideran por encima del promedio de usabilidad. El detalle del análisis y los resultados obtenidos se presentan en el Capítulo III.

d) Cuestionario abierto a docentes

Propósito: Recoger la valoración cualitativa de los docentes sobre la propuesta, en

complemento al SUS aplicado a estudiantes, para responder al objetivo específico de medir el nivel de aceptación de la solución por parte de los dos actores principales del proceso educativo.

Formato: Cuestionario semiestructurado con preguntas abiertas, distribuido en formato digital mediante un formulario en línea. El cuestionario está organizado en tres secciones: datos de contexto del docente, preguntas abiertas por dimensión y un espacio final para comentarios libres.

Aplicación: El cuestionario se administra tras una sesión individual de presentación del sistema. En la sesión se presenta al docente los tres componentes desarrollados —tutor inteligente, retroalimentación al docente y retroalimentación al estudiante— y revisan en conjunto ejemplos reales de retroalimentaciones generadas durante el periodo de prueba.

Estructura: Las preguntas abiertas se organizan en torno a seis dimensiones de la experiencia docente sobre el sistema: (1) utilidad pedagógica de la información que el sistema entrega, (2) claridad de la forma en que se presenta el contenido, (3) confianza en la información generada por el modelo de lenguaje, (4) frecuencia y oportunidad de la entrega semanal, (5) facilidad de acceso y consulta de las retroalimentaciones, y (6) disposición a utilizar el sistema en periodos académicos posteriores.

Justificación de la elección del instrumento: La opción cualitativa se adoptó por dos razones. Primero, el formato abierto recoge matices que un instrumento estandarizado como el SUS no captura: los docentes pueden señalar qué aspectos del sistema consideran útiles, qué reservas tienen y qué cambios pedirían, con un nivel de detalle que orienta posteriormente las recomendaciones del trabajo. Segundo, el grupo docente es de tamaño reducido (cinco participantes), condición bajo la cual el procesamiento estadístico de un cuestionario cerrado no aporta evidencia robusta, mientras que el análisis temático de respuestas abiertas extrae información rica incluso con muestras pequeñas.

Procesamiento: Las respuestas se sistematizan mediante análisis temático por dimensión. Para cada una de las seis dimensiones se identifican las categorías recurrentes en las respuestas y se organizan los hallazgos en una síntesis que el Capítulo III presenta junto con los resultados del SUS.

Validación del instrumento: El cuestionario fue revisado y validado por el tutor

del trabajo antes de su aplicación. La revisión confirmó que las dimensiones y sus preguntas cubrían los aspectos relevantes de la experiencia docente con el sistema y que su redacción era clara y no inducía las respuestas; las observaciones resultantes se aplicaron en la versión definitiva.

2.2.5 Procesamiento y análisis de datos

El procesamiento y análisis de datos se organiza en dos partes, en correspondencia con las fases del trabajo. La primera presenta el análisis transversal de los datos recogidos durante la fase de diagnóstico, integrando las evidencias provenientes de las entrevistas a docentes y de las fichas documentales. La segunda describe el procedimiento que se seguirá para el análisis de los datos del cuestionario SUS en la fase de validación, cuyos resultados se presentan en el Capítulo III.

a) Análisis de los datos de diagnóstico

Los resultados obtenidos mediante los instrumentos de diagnóstico permiten fundamentar la pertinencia del desarrollo del sistema. La síntesis transversal de las entrevistas y de las fichas documentales puede organizarse en cinco lecturas.

Sobre la retroalimentación que el docente entrega. Las respuestas a la pregunta 1 muestran que los instrumentos formales de retroalimentación son escasos en la práctica cotidiana. Dos de los cinco docentes describen mecanismos sistemáticos vinculados a la plataforma institucional o a herramientas digitales del aula; los demás se apoyan en mecanismos informales como preguntas en clase, ejercicios de refuerzo y conversaciones puntuales. Ningún docente reporta un canal estructurado de retroalimentación disponible para el estudiante fuera del horario de clases. El tiempo y el volumen de estudiantes aparecen como las dos limitaciones recurrentes que impiden sostener un acompañamiento individual continuo.

Sobre la retroalimentación que el docente recibe. Las respuestas a la pregunta 2 muestran que la visibilidad pedagógica que el docente tiene sobre el avance individual de su grupo es, en la práctica, intermitente y reactiva. La información llega a través de tres canales: los resultados de evaluaciones (tardíos), las preguntas formuladas en clase (parciales, dependientes de qué estudiantes participan) y la observación directa del grupo (puntual y dependiente de quién se manifiesta). Los estudiantes que no par-

ticipan activamente quedan fuera del radar hasta que aparecen los resultados de una evaluación. Esta observación sostiene la pertinencia de un componente del sistema que devuelva al docente información agregada sobre las interacciones de sus estudiantes durante el curso, no solo al final.

Sobre las fuentes de retroalimentación a las que recurre el estudiante. Las respuestas a la pregunta 3 muestran que los estudiantes obtienen una parte importante de la retroalimentación fuera del aula, principalmente de compañeros, recursos en línea y herramientas de IA generativa. La IA aparece en cuatro de los cinco relatos como una fuente que ha crecido rápidamente y cuya forma de uso escapa al control del docente. La preocupación recurrente no es el uso de IA en sí mismo, sino su uso sin verificación, sin alineación con el contenido del curso y sin visibilidad pedagógica. Este escenario es el que el sistema desarrollado busca ordenar, ofreciendo un canal institucional alineado con el contenido del curso en lugar de dejar la práctica librada a herramientas externas sin contexto.

Sobre las condiciones de aceptación del sistema. Las respuestas a las preguntas 4 y 5 muestran apertura unánime de los docentes hacia la incorporación de una herramienta basada en IA generativa, condicionada a tres requisitos. El primero es que el contenido al que acceden los estudiantes esté alineado con el enfoque, la terminología y los objetivos del curso. El segundo es que la herramienta privilegie la explicación frente a la respuesta resuelta, para no sustituir el esfuerzo cognitivo del estudiante. El tercero es que el docente conserve la posibilidad de revisar y aprobar el material sobre el que opera el sistema, condición que conecta directamente con el principio de IA generativa supervisada que orienta este trabajo. Las funcionalidades mencionadas espontáneamente por los docentes coinciden con los componentes del sistema desarrollado: asistencia conversacional al estudiante, retorno de información al docente y resumen periódico al propio estudiante sobre su avance.

Sobre los riesgos a mitigar. Las respuestas a la pregunta 6 identifican tres líneas de preocupación: dependencia excesiva del estudiante respecto al sistema, calidad y veracidad de las respuestas, y alineación del contenido y la retroalimentación con el curso. Estas líneas coinciden con los riesgos documentados en la literatura revisada en la sección 1.2.5, lo que confirma que las preocupaciones locales no son particulares del contexto sino consistentes con las identificadas internacionalmente. Las estrategias

de mitigación propuestas en la literatura, en particular la supervisión humana sobre el material y los mecanismos estructurales sobre los componentes que interactúan en tiempo real con el estudiante, orientan las decisiones de diseño del sistema.

Las fichas documentales aportan respaldo cuantitativo y metodológico a estas cinco lecturas. La ficha sobre Mousavinasab et al. [13] documenta la arquitectura modular consolidada en los Sistemas de Tutoría Inteligente y la diferencia funcional que introducen los sistemas conversacionales basados en LLMs. La ficha sobre Wan y Chen [3] muestra que con técnicas accesibles, como prompt engineering y few-shot learning, es viable generar retroalimentación que el docente acepta con mínima modificación, lo que respalda la viabilidad técnica de la propuesta y refuerza la lógica de supervisión docente. La ficha sobre Villegas-Ch et al. [5] constituye el único antecedente reportado en Ecuador de un sistema educativo basado en la API de OpenAI con resultados cuantitativos publicados, lo que sitúa a esta investigación en continuidad con un esfuerzo local de integración de IA generativa en educación superior y respalda la viabilidad de su despliegue en el contexto universitario ecuatoriano.

b) Análisis de los datos de validación

El análisis de los datos del cuestionario SUS se realiza siguiendo el procedimiento estándar descrito en la sección 2.2.4. Los puntajes individuales obtenidos por cada participante se calculan aplicando la fórmula de transformación del SUS y se interpretan en función de los rangos establecidos en la literatura. Para la presentación de los resultados se reportan tres elementos: el puntaje promedio global de la muestra de estudiantes, la distribución de los puntajes individuales mediante medidas de tendencia central y dispersión, y la interpretación de los puntajes en relación con los rangos de referencia del SUS, donde un valor superior a 68 indica usabilidad por encima del promedio.

El análisis se complementa con dos lecturas adicionales. La primera es el examen de los puntajes promedio por ítem, que permite identificar qué aspectos concretos de la usabilidad obtienen mejor o peor valoración y orienta posibles mejoras en versiones posteriores del sistema. La segunda es la descomposición del puntaje en las dos subdimensiones propuestas por Lewis y Sauro [22]: usabilidad propiamente dicha, construida con ocho de los diez ítems, y aprendibilidad, construida con los dos ítems referidos

a la necesidad de apoyo y de conocimientos previos. Esta separación distingue la facilidad general de uso del sistema de la curva de adopción que implica para un usuario nuevo, y permite localizar con mayor precisión el origen de un puntaje global por debajo o por encima de lo esperado. Ambas lecturas, junto con el detalle de los resultados, se presentan en el Capítulo III.

Para la dimensión cualitativa de la validación, las respuestas del cuestionario abierto a docentes se procesan mediante análisis temático. Cada una de las seis dimensiones del instrumento se analiza por separado: se leen las cinco respuestas correspondientes, se identifican las categorías de contenido que emergen de forma recurrente, se extraen citas literales que ilustran cada categoría y se construye una síntesis que recoge tanto las coincidencias entre docentes como los matices individuales. La presentación de los resultados en el Capítulo III organiza la síntesis por dimensión.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Desarrollo de la propuesta

La propuesta es un sistema de tutoría docente basado en IA generativa supervisada, orientado a módulos de instrucción universitarios. En el ámbito educativo, la tutoría es el acompañamiento que recibe el estudiante durante su proceso de aprendizaje: la guía ante sus dudas, la práctica con retroalimentación sobre lo que resuelve y el seguimiento de su avance. Este trabajo lleva ese acompañamiento al entorno de cada objeto de aprendizaje, combinando la interacción directa entre estudiante y docente con la mediación de la IA generativa. El sistema se desarrolla como parte de NousAI, una plataforma educativa que integra tres propuestas de trabajo complementarias: la gestión de wiki basada en IA generativa, la generación de objetos de aprendizaje a partir de videos educativos y el sistema de tutoría docente desarrollado aquí. La primera produce los objetos de aprendizaje, los conceptos y las relaciones semánticas entre ellos; la segunda incorpora material audiovisual como objeto de aprendizaje. Sobre ese contenido, este trabajo aporta las funcionalidades que sostienen la tutoría: las actividades que el estudiante intenta, el foro de preguntas de cada objeto de aprendizaje donde interviene el docente, el feedback que el estudiante deja sobre el material, y los componentes de IA generativa que se nutren de toda esa interacción. La autenticación es transversal a las tres propuestas y, por esa razón, se documenta en la subsección de seguridad.

Sobre esa base, la mediación con IA generativa articula tres componentes. El primero es el tutor inteligente: un asistente conversacional basado en IA generativa, contextualizado por el objeto de aprendizaje que el estudiante está revisando, con el que mantiene una sesión de diálogo acotada al material de la lección. De aquí en adelante, este componente se denomina tutor inteligente para distinguirlo del sentido pedagógico general de la tutoría. El segundo es la generación de retroalimentación al docente, producida de forma semanal. El tercero es la generación de retroalimentación al estudiante, también semanal, elaborada a partir de su actividad reciente. Tanto la retroalimentación al docente como la del estudiante quedan disponibles para consulta dentro de la plataforma y, además, se entregan por correo electrónico con su contenido completo. El carácter supervisado radica en que el docente decide qué objetos de apren-

dizaje se publican y accede a una retroalimentación que tiene en cuenta las acciones de los estudiantes en la plataforma. Sobre esa supervisión humana, los componentes de IA se sostienen además mediante mecanismos estructurales de diseño, validación y trazabilidad que se detallan a lo largo del capítulo.

3.1.1 Requisitos funcionales y no funcionales

Los requisitos que delimitan el alcance de la tutoría con IA dentro de NousAI se derivan de los hallazgos de las entrevistas a docentes presentadas en el Capítulo II, de las funcionalidades anunciadas en el preámbulo y de las restricciones técnicas y pedagógicas del entorno universitario. Los requisitos funcionales describen qué debe hacer el sistema; los requisitos no funcionales describen las propiedades de calidad que su comportamiento debe satisfacer. Estos últimos se agrupan en seis categorías —adecuación funcional, eficiencia de desempeño, fiabilidad, seguridad, usabilidad y mantenibilidad— que ordenan la exposición y permiten contrastar cada propiedad con el aspecto del sistema al que se refiere.

Requisitos funcionales

La Tabla 3.19 presenta los requisitos funcionales de la tutoría con IA. Cada uno describe una capacidad observable del sistema: una acción que el estudiante o el docente realizan sobre la plataforma, o un proceso que el sistema ejecuta de forma automática. Los requisitos abarcan tanto la interacción directa entre estudiante y docente —actividades, foro y feedback sobre el objeto de aprendizaje— como los tres componentes de IA generativa.

Tabla 3.19. Requisitos funcionales del subsistema de tutoría con IA.

ID	Requisito
RF-01	El estudiante debe poder tener una sesión de tutoría conversacional con el tutor inteligente en un objeto de aprendizaje.
RF-02	El estudiante debe poder intentar las actividades de un objeto de aprendizaje.
RF-03	El estudiante debe poder plantear preguntas en el foro de un objeto de aprendizaje, y el docente debe poder responderlas.
RF-04	El estudiante debe poder registrar feedback sobre un objeto de aprendizaje, y el docente debe poder consultarlo.

Tabla 3.19. Requisitos funcionales del subsistema de tutoría con IA (continuación)

ID	Requisito
RF-05	El sistema debe generar retroalimentación pedagógica al docente por objeto de aprendizaje.
RF-06	El sistema debe generar retroalimentación pedagógica al docente por módulo de instrucción.
RF-07	El sistema debe generar retroalimentación pedagógica al estudiante por módulo de instrucción.
RF-08	El sistema debe mostrar a cada destinatario las retroalimentaciones generadas para él.
RF-09	El sistema debe enviar por correo electrónico a cada destinatario las retroalimentaciones generadas para él.

Requisitos no funcionales

La Tabla 3.20 presenta los requisitos no funcionales, organizados según las seis categorías de calidad descritas.

Tabla 3.20. Requisitos no funcionales del subsistema de tutoría con IA.

ID	Categoría de calidad	Requisito
RNF-01	Adecuación funcional	Las respuestas del tutor inteligente deben permanecer acotadas al contenido del objeto de aprendizaje activo en la sesión, sin introducir información ajena a ese material.
RNF-02	Adecuación funcional	Las retroalimentaciones generadas deben ajustarse a un esquema estructurado con campos definidos, validado de forma automática antes de su persistencia.
RNF-03	Eficiencia de desempeño	Los procesos de generación periódica de retroalimentaciones deben ejecutarse en un proceso de trabajo independiente del servicio que atiende las peticiones interactivas, para no degradar los tiempos de respuesta del usuario.
RNF-04	Eficiencia de desempeño	El envío de correos electrónicos de la plataforma debe respetar un límite diario configurable; los envíos que excedan el límite se diferirán al siguiente día.

Tabla 3.20. Requisitos no funcionales del subsistema de tutoría con IA (continuación)

ID	Categoría de calidad	Requisito
RNF-05	Eficiencia de desempeño	El tutor inteligente debe limitar la cantidad de mensajes que un estudiante puede enviar dentro de una ventana de tiempo configurable, deteniendo la aceptación de nuevos mensajes cuando se supere ese límite.
RNF-06	Fiabilidad	Los fallos transitorios en las llamadas al modelo de lenguaje deben gestionarse con un esquema de reintentos con retroceso exponencial, hasta un máximo de tres intentos por solicitud.
RNF-07	Fiabilidad	Una salida del modelo que no satisfaga la validación estructural no debe persistirse ni presentarse al usuario; el fallo debe quedar registrado para su revisión posterior.
RNF-08	Seguridad	Toda operación sobre los recursos del subsistema debe requerir autenticación previa y autorización conforme al rol del usuario solicitante.
RNF-09	Seguridad	El subsistema no debe persistir información identificable del estudiante en proveedores externos de modelos de lenguaje más allá del contenido estrictamente necesario para procesar la solicitud en curso.
RNF-10	Seguridad	Toda entrada proveniente de un cliente debe validarse en el servidor antes de ser procesada, con independencia de las validaciones realizadas en el cliente.
RNF-11	Usabilidad	La usabilidad de la interfaz de tutoría se evaluará mediante la escala SUS aplicada a los estudiantes del programa de software, tomando como referencia de interpretación el valor de 68 puntos correspondiente al promedio establecido en la literatura.
RNF-12	Usabilidad	Las decisiones de diseño de interfaz deben mantener consistencia visual en toda la plataforma NousAI.
RNF-13	Mantenibilidad	El subsistema debe estructurarse por componentes desacoplados que permitan modificar un componente sin alterar los demás.

Tabla 3.20. Requisitos no funcionales del subsistema de tutoría con IA (continuación)

ID	Categoría de calidad	Requisito
RNF-14	Mantenibilidad	El catálogo de prompts del sistema debe centralizarse y mantenerse separado del código que orquesta su uso, para permitir su revisión y ajuste sin tocar la lógica de negocio.
RNF-15	Mantenibilidad	Los modelos de lenguaje empleados por cada operación deben ser configurables en tiempo de ejecución a través de un parámetro del sistema, sin requerir un nuevo despliegue.

La Tabla 3.19 enuncia nueve requisitos funcionales, todos asociados a una capacidad observable por un actor; cada uno se corresponde con uno o más casos de uso del apartado siguiente, que detallan cómo se materializan. La validación estructural de la salida del modelo de lenguaje (RNF-07) y el reintento automático ante fallos transitorios (RNF-06) no se asocian a un flujo de uso específico: actúan sobre cualquier operación que invoque al proveedor de IA y se aplican como propiedad arquitectónica del servicio que comunica con él. Su tratamiento se documenta en los apartados 3.1.4 (arquitectura) y 3.1.11 (seguridad y validación).

3.1.2 Casos de uso

El inventario de casos de uso se organiza en tres categorías. Los casos principales cubren las capacidades que la tutoría con IA implementa y que un actor activa de forma directa. Los casos incluidos representan comportamientos que un caso principal ejecuta siempre como parte de su flujo. Los casos de soporte representan comportamientos transversales activados por otros casos y no por un actor. Entre ellos se establecen dos tipos de relación: la inclusión (`<<include>>`), que indica un comportamiento que el caso base ejecuta siempre como parte de su flujo, y la extensión (`<<extend>>`), que indica un comportamiento que ocurre solo bajo una condición determinada. Estas relaciones permiten descomponer una interacción en los comportamientos que la integran sin multiplicar los puntos de entrada del actor. Las dependencias hacia funcionalidades de las otras dos propuestas del proyecto NousAI —principalmente la publicación de objetos de aprendizaje y la producción de su contenido— no se modelan como casos de uso, porque no forman parte de los flujos que esta propuesta ejecuta; se documentan

como precondiciones en la especificación de cada caso que las requiere.

Actores

La tutoría con IA reconoce cuatro actores. El estudiante conversa con el tutor inteligente, intenta las actividades, participa en el foro, registra feedback sobre el material y consulta las retroalimentaciones que recibe. El docente responde las preguntas del foro, consulta el feedback de sus estudiantes y revisa las retroalimentaciones generadas sobre sus módulos de instrucción. El tutor inteligente es un actor no humano que representa al asistente basado en IA generativa que produce las respuestas dentro de una sesión de tutoría; se modela como actor para reflejar que la respuesta no proviene de otra persona, sino del modelo de lenguaje que el sistema invoca. El disparador semanal es un actor temporal, también no humano, que representa el evento de planificación periódica que activa los procesos automáticos de generación de retroalimentación; se modela con el estereotipo «time» para distinguirlo de los demás.

Inventario de casos de uso

La Tabla 3.21 presenta el inventario completo, organizado por categoría.

Tabla 3.21. Casos de uso de la tutoría con IA.

ID	Caso de uso	Actor principal	Relación
Casos de uso principales			
CU-01	Conversar con el tutor inteligente sobre el objeto de aprendizaje	Estudiante, Tutor inteligente	—
CU-02	Intentar una actividad del objeto de aprendizaje	Estudiante	—
CU-03	Plantear una pregunta en el foro del objeto de aprendizaje	Estudiante	—
CU-04	Responder una pregunta del foro	Docente	—
CU-05	Registrar feedback sobre el objeto de aprendizaje	Estudiante	—
CU-06	Consultar el feedback del objeto de aprendizaje	Docente	—
CU-07	Generar retroalimentación al docente por objeto de aprendizaje	Disparador semanal	—

Tabla 3.21. Casos de uso de la tutoría con IA (continuación)

ID	Caso de uso	Actor principal	Relación
CU-08	Generar retroalimentación al docente por módulo de instrucción	Disparador semanal	—
CU-09	Consultar retroalimentaciones del módulo de instrucción	Docente	—
CU-10	Generar retroalimentación al estudiante por módulo de instrucción	Disparador semanal	—
CU-11	Consultar retroalimentaciones recibidas	Estudiante	—
Casos de uso incluidos			
CU-I01	Mostrar el historial de la sesión de tutoría	—	«include» de CU-01
CU-I02	Procesar la interacción del estudiante en un objeto de aprendizaje	—	«include» de CU-10
Casos de uso de soporte			
CU-S01	Entregar retroalimentación por correo al docente	—	«extend» de CU-08
CU-S02	Entregar retroalimentación por correo al estudiante	—	«extend» de CU-10

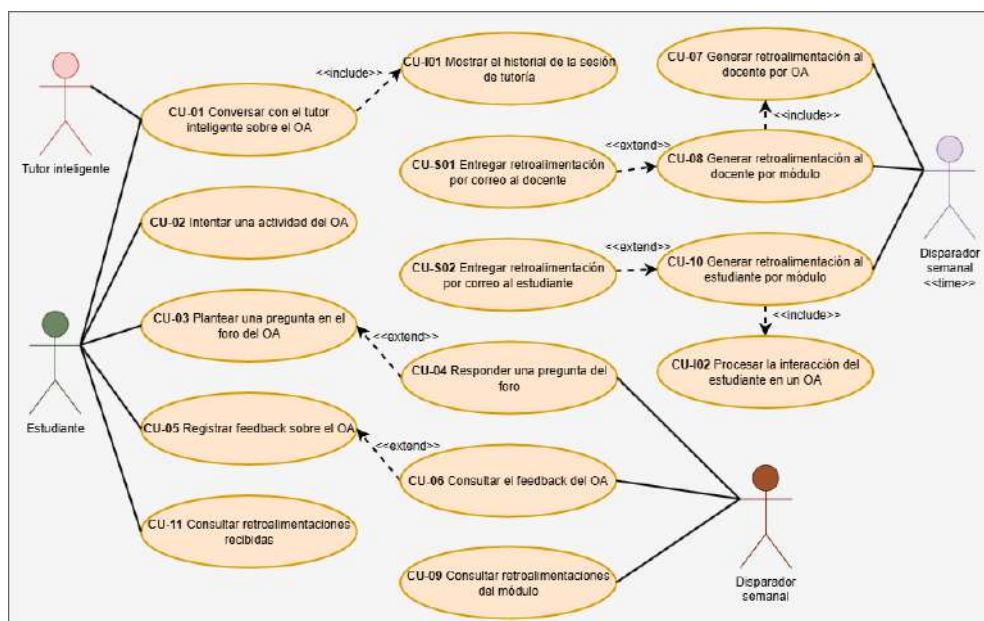


Figura 3.2. Diagrama de casos de uso del subsistema de tutoría con IA.

Relaciones entre casos de uso

La Figura 3.2 presenta el diagrama completo. Las relaciones se describen a continuación.

Inclusión. Dos casos principales incorporan comportamientos que se ejecutan siempre como parte de su flujo, mediante la relación «include». CU-01 incluye a CU-I01: cada vez que el estudiante abre o retoma una sesión, el sistema muestra el historial de mensajes de esa sesión antes de continuar la conversación. CU-10 incluye a CU-I02: antes de elaborar la retroalimentación al estudiante, el proceso reúne y procesa la interacción de ese estudiante con los objetos de aprendizaje del periodo —actividades resueltas, progreso, preguntas del foro y mensajes con el tutor inteligente—, sin persistir un resultado intermedio. Ninguno de los dos casos incluidos constituye un punto de entrada independiente.

Extensión. Tres relaciones «extend» incorporan comportamientos que ocurren solo bajo una condición. CU-03 es extendido por CU-04: el docente puede responder una pregunta del foro, pero esa respuesta depende de que exista una pregunta previa y no es un paso obligado del flujo del estudiante. CU-05 es extendido por CU-06 con el mismo criterio: el docente consulta el feedback que el estudiante registró, cuando ese feedback existe. La entrega por correo sigue el mismo patrón: CU-08 es extendido por CU-S01, que envía la retroalimentación de módulo al docente cuando se genera un

resultado nuevo y la difiere cuando el envío excede el límite diario de la plataforma; CU-10 es extendido por CU-S02 con el mismo criterio para la retroalimentación al estudiante. La retroalimentación por objeto de aprendizaje generada en CU-07 no tiene extensión asociada: se registra para consulta posterior y como insumo de la retroalimentación de módulo, sin envío de correo.

Precondiciones externas. CU-01 requiere la existencia de un objeto de aprendizaje publicado por el docente. CU-02, CU-03 y CU-05 requieren igualmente un objeto de aprendizaje publicado con el que el estudiante interactúa. CU-07 y CU-10 requieren, además, los registros de interacción asociados a ese contenido. La publicación de los objetos de aprendizaje y la producción de su contenido son responsabilidad de las otras dos propuestas del proyecto NousAI. Estas precondiciones figuran de manera explícita en la especificación de cada caso que las consume.

La validación de la salida del modelo de lenguaje, exigida por RNF-07, no se modela como caso de uso. Es una propiedad transversal del sistema, aplicada por el servicio del proveedor de IA en cada generación; su descripción corresponde al diseño de la arquitectura (apartado 3.1.4) y la seguridad (apartado 3.1.11).

Especificación de los casos de uso principales

Cada caso de uso principal se especifica con su nombre, los actores que intervienen, las precondiciones que deben cumplirse para activarlo, una descripción funcional y las postcondiciones que el sistema garantiza al finalizar.

Tabla 3.22. Especificación de CU-01.

Campo	Contenido
Nombre	Conversar con el tutor inteligente sobre el objeto de aprendizaje
Actores	Estudiante, Tutor inteligente
Precondiciones	El estudiante está autenticado y matriculado en el módulo al que pertenece el objeto de aprendizaje. El objeto de aprendizaje existe y ha sido publicado por el docente.

Tabla 3.22. Especificación de CU-01 (continuación)

Campo	Contenido
Descripción	El estudiante envía un mensaje sobre el objeto de aprendizaje que está consultando. Si no existe una sesión para ese objeto, el sistema la crea de forma implícita y la asocia al estudiante y al objeto. El sistema muestra el historial de la sesión (CU-I01), construye el contexto a partir del contenido del objeto y de los mensajes previos, y solicita al tutor inteligente una respuesta acotada a ese material, que se entrega al estudiante. La respuesta señala de forma explícita cuando la consulta queda fuera del contenido de la lección. El hilo de la conversación se conserva entre turnos sin que el estudiante deba reenviar el contexto. Si el estudiante supera el límite de mensajes definido para la ventana de tiempo, el sistema deja de aceptar nuevos mensajes hasta que la ventana se renueve.
Postcondiciones	El mensaje del estudiante y la respuesta del tutor inteligente quedan registrados en la sesión.

Tabla 3.23. Especificación de CU-02.

Campo	Contenido
Nombre	Intentar una actividad del objeto de aprendizaje
Actores	Estudiante
Precondiciones	El estudiante está autenticado y matriculado en el módulo al que pertenece el objeto de aprendizaje. El objeto de aprendizaje tiene al menos una actividad publicada por el docente.
Descripción	El estudiante responde una actividad asociada al objeto de aprendizaje. El sistema registra su respuesta como un intento, determina si es correcta y devuelve el resultado. El estudiante puede realizar nuevos intentos sobre la misma actividad.
Postcondiciones	El intento queda registrado con su resultado y su número de intento, disponible como insumo de la retroalimentación posterior.

Tabla 3.24. Especificación de CU-03.

Campo	Contenido
Nombre	Plantear una pregunta en el foro del objeto de aprendizaje
Actores	Estudiante
Precondiciones	El estudiante está autenticado y matriculado en el módulo al que pertenece el objeto de aprendizaje. El objeto de aprendizaje existe y está publicado.
Descripción	El estudiante publica una pregunta en el foro del objeto de aprendizaje. El sistema registra la pregunta y la asocia al estudiante y al objeto, y la deja visible en el foro de esa lección para que el docente y otros estudiantes puedan consultarla.
Postcondiciones	La pregunta queda registrada y visible en el foro del objeto de aprendizaje.

Tabla 3.25. Especificación de CU-04.

Campo	Contenido
Nombre	Responder una pregunta del foro
Actores	Docente
Precondiciones	El docente está autenticado y es responsable del módulo al que pertenece el objeto de aprendizaje. Existe al menos una pregunta planteada en el foro de ese objeto.
Descripción	El docente consulta las preguntas del foro de un objeto de aprendizaje y publica una respuesta a una de ellas. El sistema registra la respuesta, la asocia a la pregunta correspondiente y la marca como proveniente del docente. Este caso extiende a CU-03 y ocurre solo cuando existe una pregunta previa.
Postcondiciones	La respuesta queda registrada y asociada a la pregunta, visible en el foro del objeto de aprendizaje.

Tabla 3.26. Especificación de CU-05.

Campo	Contenido
Nombre	Registrar feedback sobre el objeto de aprendizaje

Tabla 3.26. Especificación de CU-05 (continuación)

Campo	Contenido
Actores	Estudiante
Precondiciones	El estudiante está autenticado y matriculado en el módulo al que pertenece el objeto de aprendizaje. El objeto de aprendizaje existe y está publicado.
Descripción	El estudiante registra un comentario de feedback sobre el objeto de aprendizaje que está revisando. El sistema lo guarda y lo asocia al estudiante y al objeto, de modo que quede disponible para la consulta del docente.
Postcondiciones	El feedback queda registrado y asociado al objeto de aprendizaje y al estudiante.

Tabla 3.27. Especificación de CU-06.

Campo	Contenido
Nombre	Consultar el feedback del objeto de aprendizaje
Actores	Docente
Precondiciones	El docente está autenticado y es responsable del módulo al que pertenece el objeto de aprendizaje. Existe al menos un feedback registrado sobre ese objeto.
Descripción	El docente consulta el feedback que los estudiantes registraron sobre un objeto de aprendizaje a su cargo. El sistema devuelve los comentarios asociados a ese objeto. Este caso extiende a CU-05 y ocurre solo cuando existe feedback registrado.
Postcondiciones	El sistema no modifica datos; la consulta es de solo lectura.

Tabla 3.28. Especificación de CU-07.

Campo	Contenido
Nombre	Generar retroalimentación al docente por objeto de aprendizaje
Actores	Disparador semanal

Tabla 3.28. Especificación de CU-07 (continuación)

Campo	Contenido
Precondiciones	El objeto de aprendizaje existe y está publicado. Dentro del periodo evaluado existen registros de interacción asociados a ese objeto —intentos de actividad, progreso, preguntas del foro, feedback o sesiones de tutoría—, y esos datos superan los umbrales mínimos definidos para ejecutar la generación.
Descripción	El sistema recopila los registros de interacción de los estudiantes con el objeto de aprendizaje durante el periodo, construye el contexto del prompt y solicita al modelo de lenguaje la retroalimentación al docente. La salida se valida contra el esquema esperado y, una vez aceptada, se persiste como insumo de la retroalimentación de módulo.
Postcondiciones	La retroalimentación queda registrada y disponible como insumo para la generación de la retroalimentación de módulo. No se envía correo en este caso.

Tabla 3.29. Especificación de CU-08.

Campo	Contenido
Nombre	Generar retroalimentación al docente por módulo de instrucción
Actores	Disparador semanal
Precondiciones	El módulo de instrucción tiene al menos un objeto de aprendizaje con datos suficientes para haber generado retroalimentación por objeto en el periodo.
Descripción	El sistema genera primero la retroalimentación de cada objeto de aprendizaje del módulo con datos suficientes (CU-07) y consolida sus salidas. Sobre ese insumo construye el contexto del prompt de módulo y solicita al modelo de lenguaje la retroalimentación consolidada al docente. La salida se valida contra el esquema esperado y, una vez aceptada, se persiste y se entrega al docente.
Postcondiciones	La retroalimentación de módulo queda registrada y asociada al docente y al módulo. La entrega por correo se activa como extensión (CU-S01).

Tabla 3.30. Especificación de CU-09.

Campo	Contenido
Nombre	Consultar retroalimentaciones del módulo de instrucción
Actores	Docente
Precondiciones	El docente está autenticado y es responsable del módulo de instrucción consultado.
Descripción	El docente solicita las retroalimentaciones generadas para su módulo. El sistema devuelve la lista con su fecha de generación y permite acceder al detalle, donde se muestran el resumen, las fortalezas, los puntos de mejora y las recomendaciones organizados en secciones.
Postcondiciones	El sistema no modifica datos; la consulta es de solo lectura.

Tabla 3.31. Especificación de CU-10.

Campo	Contenido
Nombre	Generar retroalimentación al estudiante por módulo de instrucción
Actores	Disparador semanal
Precondiciones	El estudiante mantiene matrícula activa en al menos un módulo de instrucción. Dentro del periodo evaluado existen registros de interacción asociados al estudiante, y esos datos superan los umbrales mínimos definidos para ejecutar la generación.
Descripción	El sistema procesa la interacción del estudiante con los objetos de aprendizaje del periodo (CU-I02): los objetos completados, los intentos de actividad y su tasa de acierto, los conceptos asociados a sus dificultades, las preguntas del foro y los mensajes con el tutor inteligente. Con esos datos construye el contexto del prompt y solicita al modelo de lenguaje la retroalimentación al estudiante. La salida se valida contra el esquema esperado y, una vez aceptada, se persiste y se entrega al estudiante.
Postcondiciones	La retroalimentación queda registrada y asociada al estudiante. La entrega por correo se activa como extensión (CU-S02).

Tabla 3.32. Especificación de CU-11.

Campo	Contenido
Nombre	Consultar retroalimentaciones recibidas
Actores	Estudiante
Precondiciones	El estudiante está autenticado y matriculado en el módulo a consultar.
Descripción	El estudiante solicita las retroalimentaciones generadas para él. El sistema devuelve la lista con su fecha y permite acceder al detalle, donde se muestran el resumen, las fortalezas, las dificultades y las recomendaciones organizados en secciones.
Postcondiciones	El sistema no modifica datos; la consulta es de solo lectura.

3.1.3 Selección de tecnologías

El conjunto de tecnologías sobre el que se construye la tutoría con IA fue definido por el equipo de NousAI para toda la plataforma, y este trabajo se suma a esas decisiones. Aun así, se documentan aquí porque condicionan de forma directa el diseño y la implementación de la tutoría, y porque varias de ellas —el almacenamiento vectorial, el procesamiento asíncrono y la validación de salidas de IA— se propusieron en el ámbito de este trabajo y resultan determinantes para las tres propuestas que integra la plataforma NousAI. Para cada decisión se presentan las alternativas consideradas y los criterios que llevaron a la elección.

Lenguaje y framework de backend

El backend concentra la lógica de la tutoría: la orquestación del tutor inteligente, la generación de retroalimentaciones, la integración con el proveedor de IA y la gestión de las interacciones del estudiante con cada objeto de aprendizaje. La elección debía favorecer un código organizado por módulos, tipado y con soporte para inyección de dependencias, dado el número de servicios que se integran.

Tabla 3.33. Comparación de frameworks de backend.

Criterio	Express	Fastify	NestJS (elegido)
Tipado estático	Opcional, configuración manual	Opcional	TypeScript de base

Tabla 3.33. Comparación de frameworks de backend (continuación)

Criterio	Express	Fastify	NestJS (elegido)
Arquitectura impuesta	Ninguna	Mínima	Modular, con inyección de dependencias
Soporte para procesos worker	Manual	Manual	Integración directa con colas
Documentación de API	Manual	Mediante plugin	Generación automática con decoradores
Curva de adopción en equipo	Baja	Media	Media

NestJS se eligió porque las tres responsabilidades del componente —orquestrar el tutor inteligente, generar las retroalimentaciones e integrar el proveedor de IA— exigen coordinar varios servicios externos sin que el código se vuelva difícil de seguir. Su sistema de inyección de dependencias resuelve esa coordinación de forma explícita, y su tipado de base reduce los errores en las fronteras entre componentes, que es donde se concentran las integraciones. La organización por módulos permite además mantener los componentes de tutoría en unidades separadas que se modifican de forma independiente, tal como exige RNF-13. Express y Fastify habrían requerido construir manualmente esa estructura y la integración con las colas de trabajo, sin aportar ventajas que el proyecto necesite.

Framework de frontend

La interfaz de la tutoría —vistas de chat y vistas de consulta de retroalimentaciones— se integra en la aplicación web de NousAI. La decisión de frontend debía garantizar consistencia visual y de interacción con el resto de la plataforma, requisito recogido en RNF-12.

Tabla 3.34. Comparación de frameworks de frontend.

Criterio	Angular	React	Vue (elegido)
Curva de aprendizaje	Alta	Media	Baja a media
Peso y rendimiento inicial	Mayor	Medio	Menor

Tabla 3.34. Comparación de frameworks de frontend (continuación)

Criterio	Angular	React	Vue (elegido)
Integración en una base de código existente	Estructura rígida, impone convenciones propias	Flexible, sin estructura impuesta	Flexible y ya adoptado en NousAI
Coherencia con el resto de la plataforma	Requeriría un segundo ecosistema	Requeriría un segundo ecosistema	Reutiliza componentes existentes

Vue se eligió porque es la base sobre la que ya está construido el frontend de NousAI. Las vistas de tutoría —el chat y las pantallas de consulta de retroalimentaciones— se integran en esa aplicación y reutilizan sus componentes, lo que satisface de forma directa la consistencia visual y de interacción que exige RNF-12. Adoptar Angular o React habría introducido un segundo modelo de componentes en una misma plataforma, con el costo de mantener dos ecosistemas y el riesgo de divergencia visual que precisamente RNF-12 busca evitar.

Base de datos y ORM

Los componentes de tutoría necesitan almacenar sesiones de chat, mensajes, retroalimentaciones generadas y los registros que sirven de insumo a esas retroalimentaciones. Además, requiere búsqueda por similitud semántica sobre los objetos de aprendizaje. La decisión de base de datos se tomó en conjunto con la de almacenamiento vectorial, que se trata por separado más adelante.

Tabla 3.35. Comparación de gestores de base de datos relacional.

Criterio	MySQL	PostgreSQL (elegido)
Tipos de dato estructurados (JSON)	Soporte básico, sin consulta nativa por campo	Soporte nativo y consultable por campo
Búsqueda vectorial	No disponible de forma nativa	Disponible mediante extensión
Operación junto al almacenamiento vectorial	Requeriría un motor externo separado	Vectores y datos relacionales en la misma instancia

Tabla 3.35. Comparación de gestores de base de datos relacional (continuación)

Criterio	MySQL	PostgreSQL (elegido)
Integridad referencial y transacciones	Completa	Completa

La decisión se inclinó por PostgreSQL por dos necesidades concretas del componente. La primera es el almacenamiento de las retroalimentaciones: tanto la dirigida al docente como la dirigida al estudiante se persisten como contenido estructurado en formato JSON con campos que después se consultan y filtran, y PostgreSQL ofrece soporte nativo para ese tipo de dato sin recurrir a serialización manual. La segunda es la búsqueda por similitud semántica sobre los objetos de aprendizaje, que requiere almacenar y comparar representaciones vectoriales; PostgreSQL admite esta capacidad mediante una extensión sobre la misma base, mientras que MySQL exigiría incorporar un motor externo dedicado. Mantener los vectores junto a los datos relacionales en un único gestor reduce la complejidad de despliegue en la infraestructura de la universidad y elimina una dependencia adicional. La integridad referencial y la adopción en el entorno académico son equivalentes en ambos gestores, por lo que no fueron el criterio decisivo.

Sobre PostgreSQL, el acceso a datos se realiza mediante un ORM. La Tabla 3.36 compara las alternativas consideradas.

Tabla 3.36. Comparación de ORM para el acceso a datos.

Criterio	Sequelize	TypeORM	Prisma (elegido)
Tipado de modelos y consultas	Parcial	Parcial	Completo, generado desde el esquema
Definición del esquema	En código	En código con decoradores	Esquema declarativo único
Gestión de migraciones	Manual o asistida	Asistida	Asistida desde el esquema
Legibilidad de las consultas	Media	Media	Alta

Prisma se eligió porque genera los tipos del acceso a datos a partir de un único

esquema declarativo. Como la base de datos es compartida por las tres propuestas de NousAI y el esquema evoluciona durante el desarrollo, mantener una sola fuente de la que derivan tanto el modelo como su tipado evita que el código y la estructura de datos se desincronicen, un riesgo que Sequelize y TypeORM solo cubren de forma parcial al definir el esquema en código. Su gestión de migraciones, derivada del mismo esquema, ordena además esa evolución sin pasos manuales propensos a error.

Almacenamiento y búsqueda vectorial

El tutor inteligente y la generación de retroalimentaciones se apoyan en la representación de los objetos de aprendizaje como vectores de características, lo que permite localizar contenido por similitud semántica y no solo por coincidencia de texto. La decisión consistía en elegir entre un motor vectorial dedicado y una extensión sobre la base de datos relacional ya adoptada.

Tabla 3.37. Comparación de soluciones de almacenamiento vectorial.

Criterio	Pinecone	Weaviate	pgvector (elegido)
Naturaleza del servicio	Servicio externo gestionado	Motor dedicado auto-alojado	Extensión de PostgreSQL
Infraestructura adicional	Requiere cuenta y servicio externo	Requiere despliegue separado	Ninguna; se integra en la base existente
Consistencia con los datos relacionales	Vectores en un sistema aparte	Vectores en un sistema aparte	Vectores y datos relacionales en la misma base
Dependencia de terceros	Alta	Baja	Baja
Ajuste al volumen del proyecto	Sobredimensionado	Sobredimensionado	Adecuado

pgvector se eligió porque integra la búsqueda vectorial dentro de PostgreSQL, sin sumar un servicio independiente al despliegue. Mantener los vectores junto a los datos relacionales que describen cada objeto de aprendizaje permite consultarlos en la misma operación y elimina la necesidad de sincronizar dos sistemas. Frente a un motor dedicado como Weaviate o un servicio externo como Pinecone, esta decisión simplifica la operación en los servidores de la universidad y evita una dependencia externa.

Procesamiento asíncrono

La generación de retroalimentaciones al docente y al estudiante son procesos periódicos que invocan al modelo de lenguaje, una operación de duración variable y susceptible de fallos transitorios. Ejecutarlos dentro del ciclo de atención de peticiones degradaría los tiempos de respuesta de la interfaz, lo que contraviene RNF-03. Se requería, por tanto, un mecanismo de colas de trabajo con un proceso de ejecución independiente.

Tabla 3.38. Comparación de mecanismos de procesamiento asíncrono.

Criterio	Cron del sistema operativo	RabbitMQ	BullMQ sobre Redis (elegido)
Reintentos automáticos	No	Configurable	Integrados, con retroceso configurable
Trabajos diferidos y programados	Limitado	Mediante extensiones	Soporte nativo
Infraestructura adicional	Ninguna	Servidor de mensajería dedicado	Redis, ya previsto para almacenamiento en memoria
Integración con el backend	Externa al proceso	Mediante cliente	Directa, mismo lenguaje y framework
Observabilidad de los trabajos	Escasa	Media	Estado de cada trabajo consultable

BullMQ sobre Redis se eligió porque ofrece dos capacidades que el componente usa de forma directa. La primera es el reintento automático con retroceso exponencial ante fallos transitorios del modelo de lenguaje, que es lo que sostiene RNF-06. La segunda es el soporte nativo de trabajos diferidos, que permite reprogramar al día siguiente los correos que exceden el límite diario, conforme a RNF-04. Si bien un servidor de mensajería dedicado como RabbitMQ era una alternativa, se prefirió BullMQ por su integración oficial y más fluida con el ecosistema de NestJS, lo que reduce la complejidad de configuración e implementación. La programación basada únicamente en cron se descartó por carecer de la gestión de reintentos y del control del estado de los trabajos.

Proveedor de modelos de IA

Los tres componentes de la propuesta dependen de un proveedor de modelos de lenguaje para la generación conversacional y de retroalimentaciones, y de modelos de representaciones vectoriales para la búsqueda semántica. La decisión se planteó entre desplegar modelos de forma local o recurrir a un proveedor de servicios de IA en la nube.

Tabla 3.39. Comparación de proveedores de modelos de IA.

Criterio	Modelos autoalojados	Google / Anthropic	OpenAI (elegido)
Cobertura de generación, vectores e imágenes	Parcial, con modelos separados	Parcial según proveedor	Las tres capacidades en un mismo proveedor
Gestión de estado conversacional	Manual	Manual o parcial	Soporte para conversación con estado en servidor
Requisitos de infraestructura	Altos, hardware especializado	Ninguno	Ninguno
Madurez de la interfaz de programación	Variable	Alta	Alta
Configurabilidad del modelo empleado	Alta	Media	Media a alta

El despliegue de modelos de IA generativa de forma local se descartó porque exige infraestructura de cómputo especializada de la que no se dispone en el entorno de la universidad, además del trabajo continuo de operación y actualización que conlleva. Entre los proveedores en la nube, OpenAI se eligió porque reúne en un mismo proveedor las tres capacidades que el componente necesita —generación de texto, representaciones vectoriales e imágenes— y porque ofrece estado conversacional gestionado en servidor, que es lo que permite encadenar los turnos del tutor sin reenviar el historial completo. Su interfaz de programación, estable y ampliamente adoptada en entornos de producción, reduce el riesgo técnico del proyecto. El modelo concreto empleado en cada operación no queda fijado en el código: se configura a través de un parámetro del

sistema, conforme a RNF-14, lo que permite sustituirlo sin un nuevo despliegue.

Validación de salidas de IA

Las respuestas del modelo de lenguaje son texto y no garantizan, por sí solas, una estructura predecible. Dado que las retroalimentaciones se persisten y se presentan al usuario con un formato definido, el sistema necesita verificar la estructura de cada salida antes de aceptarla, conforme a RNF-07. La validación de la entrada proveniente de los clientes ya se resuelve con las utilidades del framework de backend; la decisión aquí se refería específicamente a la validación de las salidas de la IA.

Tabla 3.40. Comparación de enfoques de validación de salidas de IA.

Criterio	Validación manual	JSON Schema	Zod (elegido)
Definición del esquema	Código a medida	Documento de esquema separado	Esquema en el mismo lenguaje del backend
Tipado derivado del esquema	No	Requiere herramientas adicionales	Inmediato, sin paso intermedio
Mensajes de error	A medida	Genéricos	Detallados, por campo
Mantenimiento ante cambios de formato	Costoso	Medio	Bajo

Zod se eligió porque permite declarar el esquema esperado en el mismo lenguaje del backend y obtener de él tanto la validación como el tipo de datos correspondiente, sin un paso intermedio. Como la estructura de las retroalimentaciones evoluciona durante el desarrollo, mantener la definición del formato junto al código que la consume reduce el costo de cada cambio, frente a un documento de esquema separado o a una validación escrita a medida. La validación se centraliza en el servicio que comunica con el proveedor de IA: la salida se contrasta con el esquema en el mismo punto en que se obtiene y, si no se ajusta, se solicita una nueva generación al modelo en lugar de propagar una respuesta malformada. Esta capacidad sostiene de forma directa RNF-07: ninguna salida que no se ajuste al esquema se persiste ni se presenta al usuario.

Síntesis

Las decisiones tecnológicas que sustentan el sistema cumplen propósitos específi-

cos dentro de la arquitectura. Las elecciones relativas al lenguaje, los frameworks, la base de datos y el ORM fueron establecidas de manera unificada desde el inicio del desarrollo de la plataforma NousAI, lo que dota a la tutoría con IA de una base tipada, modular y consistente con el resto de la aplicación. En contraste, las tecnologías de almacenamiento vectorial, procesamiento asíncrono, proveedor de IA y validación de salidas se seleccionaron por su capacidad para sostener de forma directa los componentes de la tutoría. La Tabla 3.41 resume el conjunto del stack y la función que cumple cada tecnología.

Tabla 3.41. Resumen del stack tecnológico del subsistema de tutoría con IA.

Capa	Tecnología	Función en el subsistema
Backend	NestJS (TypeScript)	Orquestación de los tres componentes e integración de servicios
Frontend	Vue	Vistas de chat y de consulta de retroalimentaciones
Base de datos	PostgreSQL	Almacenamiento de sesiones, mensajes y retroalimentaciones
Búsqueda vectorial	pgvector	Localización de contenido por similitud semántica
ORM	Prisma	Acceso a datos tipado y gestión de migraciones
Procesamiento asíncrono	BullMQ sobre Redis	Generación periódica de retroalimentaciones y envío diferido de correos
Modelos de IA	OpenAI	Generación conversacional, de retroalimentaciones y de representaciones vectoriales
Validación de salidas	Zod	Verificación estructural de las respuestas del modelo de lenguaje

3.1.4 Arquitectura del sistema

La arquitectura del sistema se describe en una vista de tres capas: presentación, lógica de negocio y datos. La Figura 3.3 muestra las tecnologías que componen cada capa y la forma en que se comunican entre sí.

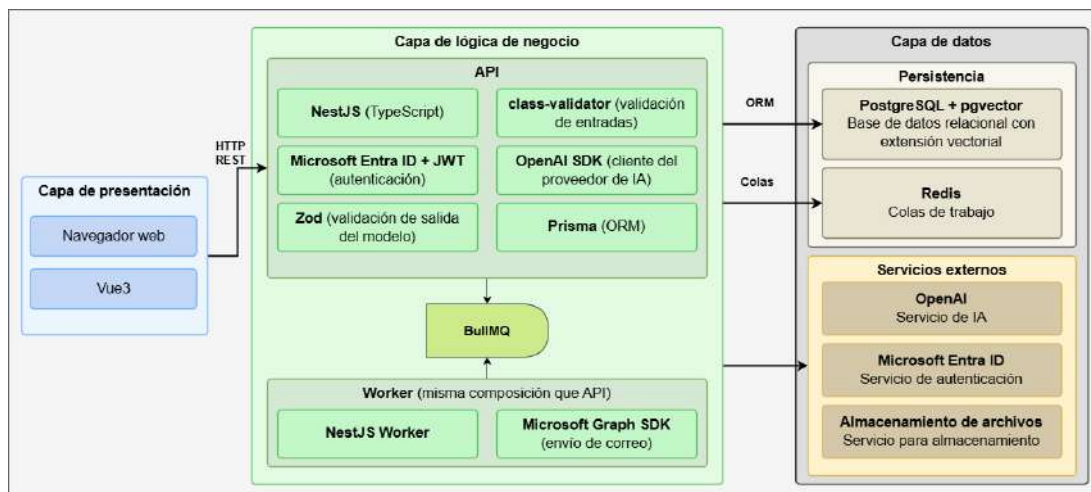


Figura 3.3. Arquitectura del sistema en tres capas.

Capa de presentación. La capa de presentación es la aplicación web construida con Vue, compartida por toda la plataforma NousAI. Se ejecuta en el navegador del usuario y se comunica con la capa de lógica de negocio a través de la API REST documentada en el apartado 3.1.7.

Capa de lógica de negocio. La capa de lógica de negocio es el backend construido con NestJS sobre TypeScript. Está compuesta por dos procesos que comparten el mismo código base pero se ejecutan de forma independiente. El proceso de API atiende las peticiones síncronas de la capa de presentación. El proceso worker ejecuta las tareas asíncronas y periódicas, en particular la generación de retroalimentaciones y la entrega diferida de correos.

El proceso de API integra class-validator para validar los datos que llegan en cada solicitud, Prisma como ORM para acceder a la base de datos, el SDK de OpenAI para comunicarse con el proveedor de modelos de lenguaje, y Zod para validar la estructura de las respuestas que ese proveedor devuelve. La autenticación se resuelve combinando Microsoft Entra ID, como proveedor de identidad institucional, con tokens JWT para sostener la sesión interna. El detalle de cada uno de estos mecanismos se desarrolla en el apartado 3.1.11.

El proceso worker integra BullMQ para consumir las colas de trabajo y el SDK de Microsoft Graph para el envío de los correos de retroalimentación a docentes y estudiantes. La generación de retroalimentaciones se delega a este proceso para que las operaciones prolongadas contra el modelo de lenguaje no afecten los tiempos de

respuesta del API.

Capa de datos. La capa de datos reúne los sistemas de persistencia y el sistema de mensajería sobre los que se apoya el backend. La base de datos PostgreSQL, con la extensión pgvector, almacena la información estructurada del sistema —sesiones de chat, mensajes, retroalimentaciones generadas— junto con las representaciones vectoriales de los objetos de aprendizaje. Redis sostiene las colas de trabajo que comunican el proceso de API con el worker, gestionadas a través de BullMQ. El almacenamiento local del servidor conserva los archivos asociados a los objetos de aprendizaje.

Como servicios externos, el sistema se integra con OpenAI para las operaciones de inferencia y embedding, con Microsoft Entra ID para la autenticación y con Microsoft Graph para el envío de correos. Estos servicios no forman parte de la arquitectura interna del sistema; el backend depende de ellos pero no los implementa.

Carácter supervisado y arquitectura. El carácter supervisado del sistema, en su componente técnico, queda concentrado en la capa de lógica de negocio. La capa de presentación no decide qué entradas son válidas ni qué salidas del modelo se aceptan, y la capa de datos solo persiste lo que la lógica de negocio ya validó. Esta concentración es deliberada: sitúa la validación de entradas (class-validator) y la verificación de salidas del modelo (Zod) en un único punto del sistema. Su funcionamiento se detalla en los apartados 3.1.6 y 3.1.11.

3.1.5 Diseño de la base de datos

NousAI usa una base de datos única, compartida por las tres propuestas, que reúne 27 entidades para la gestión de usuarios, módulos de instrucción, objetos de aprendizaje, generación de contenido, ingesta de video e interacciones de los estudiantes. El esquema global (Figura 4.25) se incluye en el Anexo A como referencia. Este apartado se concentra en las entidades que la tutoría crea y gestiona, junto con aquellas de las que depende como insumo; el resto del esquema corresponde a las otras dos propuestas del proyecto.

Conviene distinguir dos grupos de entidades. Las entidades propias son las que se crean y administran desde la tutoría: las sesiones y mensajes del tutor inteligente, las preguntas y respuestas del foro, las actividades y sus intentos, el feedback del estudiante sobre el material y las retroalimentaciones generadas para el docente y para el

estudiante. Las entidades de insumo son las que se leen pero no se modifican: los objetos de aprendizaje, la información de contexto de los módulos de instrucción y los registros de avance de los estudiantes.

Entidades propias

Session representa una sesión de tutoría conversacional entre un estudiante y un objeto de aprendizaje. La relación es de unicidad: un estudiante mantiene una sola sesión por objeto de aprendizaje, de modo que el diálogo sobre una misma lección se conserva como una conversación continua. Cada sesión registra su título y la fecha de inicio.

Message representa un mensaje dentro de una sesión. Almacena el contenido, el rol del emisor —estudiante o asistente— y un campo de metadatos donde se conserva el identificador de la respuesta del proveedor de IA. Ese identificador es el que permite encadenar los turnos de la conversación sin reenviar el historial completo en cada interacción. Una sesión agrupa muchos mensajes.

Activity representa una actividad evaluativa asociada a un objeto de aprendizaje, con su tipo, enunciado y opciones. ActivityAttempt representa cada intento de un estudiante sobre una actividad, con la respuesta entregada, si resultó correcta y el número de intento. Juntas registran la práctica del estudiante sobre el material.

StudentQuestion representa una pregunta que el estudiante plantea en el foro de un objeto de aprendizaje, y QuestionReply representa cada respuesta a esa pregunta, con un indicador de si proviene del docente. Ambas sostienen la interacción del foro entre estudiante y docente sobre una misma lección.

LearningObjectFeedback representa el feedback que el estudiante registra sobre un objeto de aprendizaje, asociado al estudiante y al objeto, y disponible para la consulta del docente.

TeacherAiFeedback representa una retroalimentación generada para el docente. Su campo de alcance distingue los dos niveles descritos en los requisitos: la retroalimentación de un objeto de aprendizaje concreto y la retroalimentación agregada de un módulo de instrucción completo. El contenido se almacena en un campo estructurado con formato JSON, que recoge el resumen, las fortalezas, los puntos de mejora y las recomendaciones. La entidad se asocia siempre a un módulo y, cuando el alcance es de objeto de aprendizaje, también al objeto correspondiente.

StudentAiFeedback representa una retroalimentación generada para el estudiante. Se asocia al estudiante y al módulo de instrucción sobre el que se elabora, y su contenido se almacena igualmente en un campo estructurado con formato JSON. Su alcance corresponde al resumen periódico de la actividad del estudiante.

Entidades de insumo

La tutoría lee varias entidades que pertenecen a la gestión de contenido. LearningObject es la entidad central de insumo: sobre ella se desarrolla la sesión de tutoría, se plantean las actividades y las preguntas del foro, y es la unidad sobre la que se genera la retroalimentación al docente. Module y AiConfiguration aportan, respectivamente, el contexto de pertenencia y la parametrización pedagógica que la tutoría con IA lee para condicionar el comportamiento del modelo en cada módulo de instrucción.

LearningObjectProgress registra el avance del estudiante sobre cada objeto de aprendizaje, y LearningObjectConcept guarda los conceptos cubiertos por el contenido de un objeto de aprendizaje. Enrollment vincula a cada estudiante con los módulos en los que está matriculado, lo que delimita el alcance de la retroalimentación que recibe. User se referencia como propietario de las sesiones y destinatario de las retroalimentaciones al estudiante, así como responsable de los módulos en el caso del docente.

La Tabla 3.42 resume las entidades de tutoría con IA y su clasificación.

Tabla 3.42. Entidades de la base de datos correspondientes a la tutoría con IA.

Entidad	Clasificación	Función
Session	Propia	Sesión de tutoría conversacional entre un estudiante y un objeto de aprendizaje
Message	Propia	Mensaje individual dentro de una sesión de tutoría
Activity	Propia	Actividad evaluativa asociada a un objeto de aprendizaje
ActivityAttempt	Propia	Intento de resolución de una actividad por parte de un estudiante
StudentQuestion	Propia	Pregunta planteada por un estudiante en el foro
QuestionReply	Propia	Respuesta a una pregunta del foro, del estudiante o del docente

Tabla 3.42. Entidades de la base de datos correspondientes a la tutoría con IA (continuación)

Entidad	Clasificación	Función
LearningObjectFeedback	Propia	Feedback de un estudiante sobre un objeto de aprendizaje
TeacherAiFeedback	Propia	Retroalimentación generada para el docente, con alcance de objeto de aprendizaje o de módulo
StudentAiFeedback	Propia	Retroalimentación periódica generada para el estudiante
LearningObject	Insumo	Objeto de aprendizaje sobre el que se conversa y se genera retroalimentación
Module	Insumo	Módulo de instrucción al que pertenecen los objetos de aprendizaje y las retroalimentaciones
AiConfiguration	Insumo	Parámetros pedagógicos de la IA por módulo de instrucción
Enrollment	Insumo	Matrícula que vincula a un estudiante con un módulo de instrucción
User	Insumo	Usuario del sistema; propietario de sesiones y destinatario de retroalimentaciones
LearningObjectProgress	Insumo	Registro de avance de un estudiante sobre un objeto de aprendizaje
LearningObjectConcept	Insumo	Concepto asociado a un objeto de aprendizaje

La Figura 3.4 presenta el modelo entidad-relación de la tutoría con IA, con las entidades propias y las de insumo diferenciadas, y las relaciones entre ellas.

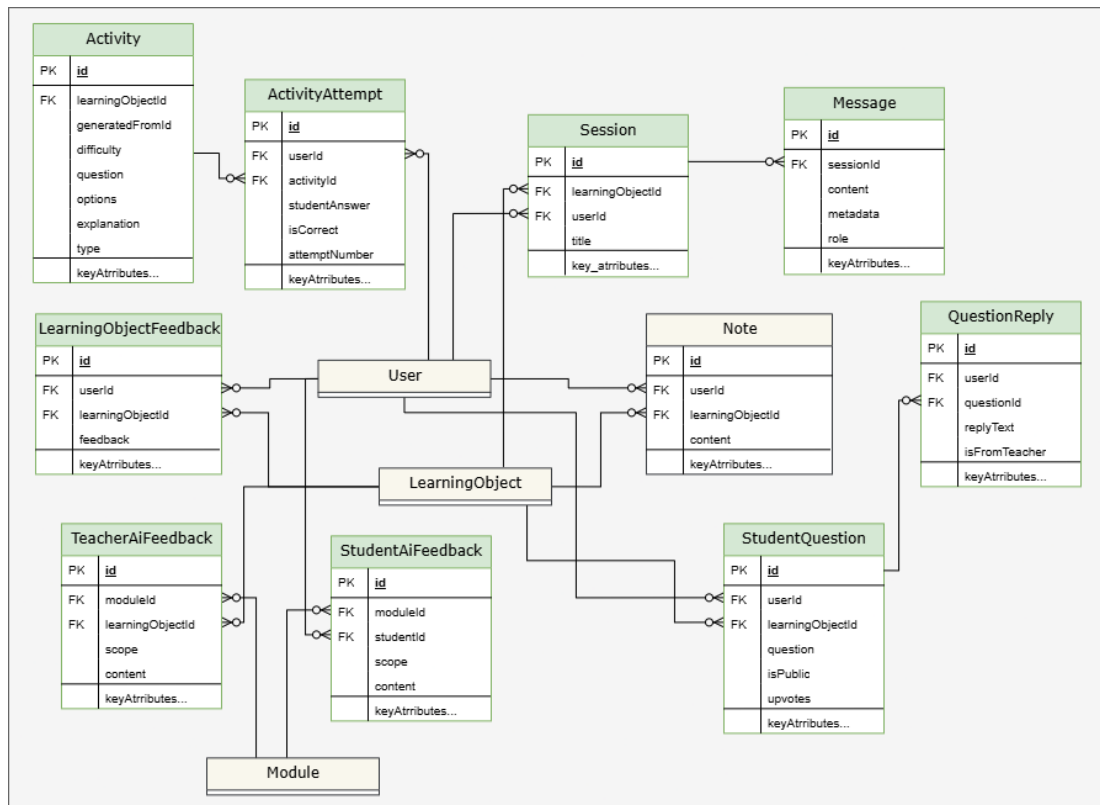


Figura 3.4. Modelo entidad-relación de la tutoría con IA.

3.1.6 Diagramas de clases por paquetes

La estructura de clases de la tutoría con IA se organiza por paquetes. Un paquete agrupa las clases que cumplen una responsabilidad común dentro del backend. La tutoría se reparte en dos grupos de paquetes. El primero concentra la lógica de mediación con IA generativa: `providers/ai`, que aísla la integración con el proveedor de modelos de lenguaje y es compartido por los demás; `features/chat`, `features/teacher-feedback` y `features/student-ai-feedback`, que implementan respectivamente el tutor inteligente, la retroalimentación al docente y la retroalimentación al estudiante. El segundo grupo, bajo `features/interactions`, reúne los paquetes que gestionan la interacción directa del estudiante con cada objeto de aprendizaje: las actividades y sus intentos, el foro de preguntas y respuestas, y el feedback del estudiante sobre el material. Los paquetes del primer grupo se presentan a continuación con su propio diagrama de clases, a nivel de controladores, servicios y procesos de trabajo, con sus métodos públicos principales, ya que su responsabilidad —orquestrar la generación con modelos de lenguaje, sostener el contexto conversacional y recolectar los datos de cada retroalimentación— concentra la lógica no trivial del backend. Los paquetes de fea-

tures/interactions resuelven operaciones de persistencia directa sin generación de IA, por lo que se documentan a nivel de inventario en esta subsección y sus puntos de acceso se detallan en el diseño de la API REST (apartado 3.1.7). Al final, un diagrama global muestra las dependencias entre los paquetes y el de envío de correo del que dependen dos de ellos. Las clases que pertenecen a otros paquetes de NousAI y que la tutoría utiliza como apoyo se identifican como externas.

Paquete providers/ai

El paquete providers/ai aísla la comunicación con el proveedor de modelos de lenguaje y ofrece al resto del backend una interfaz uniforme para las operaciones de IA. Es el punto único por el que pasan todas las llamadas a los modelos, lo que permite concentrar en un solo lugar los mecanismos de configuración.

OpenaiService es la clase central del paquete. Expone las operaciones de generación de respuestas estructuradas en formato JSON, de generación de texto libre en formato Markdown —empleada por el tutor inteligente—, de generación de representaciones vectoriales y de generación de imágenes. Internamente aplica el reintento con retroceso exponencial ante fallos transitorios y registra el consumo de tokens de cada solicitud.

AiConfigService resuelve qué modelo concreto se usa en cada operación. Los modelos no están fijados en el código: se leen de la configuración del sistema, lo que permite cambiarlos sin un nuevo despliegue. La gestión de esa configuración corresponde a la administración de la plataforma NousAI y queda fuera del alcance de esta propuesta.

La validación de las respuestas estructuradas no es una pieza separada, sino una responsabilidad del propio OpenaiService. Su método de generación de respuestas estructuradas recibe, como parámetro opcional, el esquema esperado de la salida; cuando se proporciona, el servicio valida la respuesta del modelo contra ese esquema antes de devolverla. Si la respuesta no se ajusta al esquema, el servicio solicita una nueva generación al modelo para obtener una salida corregida, de modo que ningún componente recibe una respuesta que no cumpla la estructura prevista. Esta integración es la que sostiene la verificación estructural exigida en los requisitos.

La Figura 3.5 presenta el diagrama de clases del paquete providers/ai.

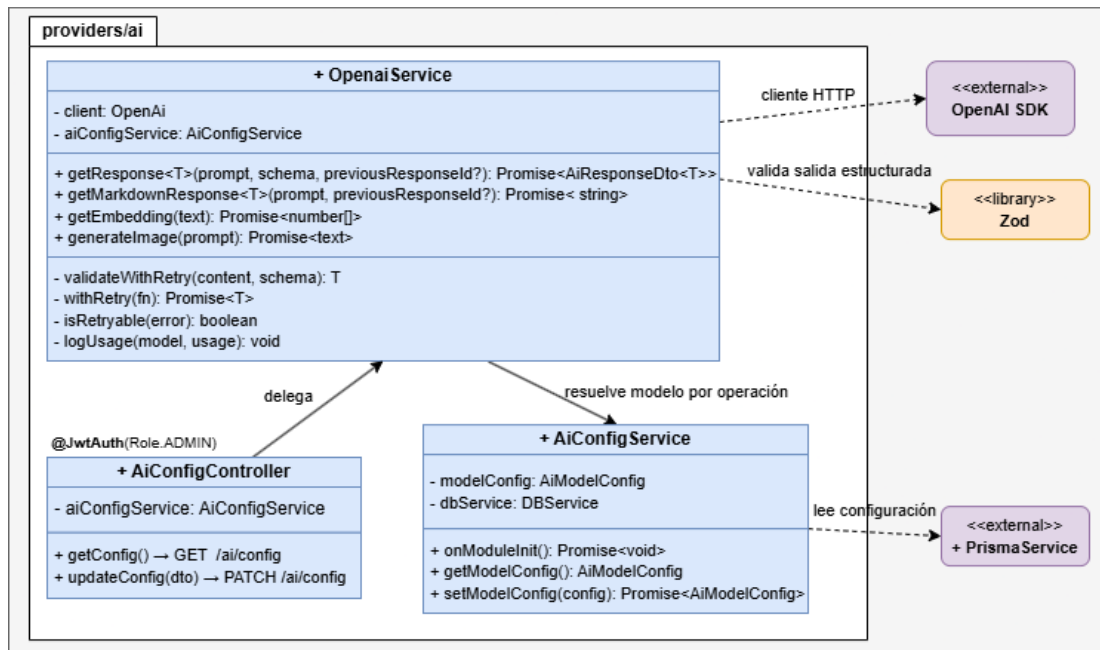


Figura 3.5. Diagrama de clases del paquete providers/ai.

Paquete features/chat

El paquete features/chat implementa el tutor inteligente. Una sesión de tutoría vincula a un estudiante con un objeto de aprendizaje y agrupa el intercambio de mensajes sobre esa lección.

La entrada al paquete se reparte en dos controladores. LoSessionsController atiende la creación u obtención de la sesión asociada a un objeto de aprendizaje. SessionMessagesController atiende el listado de mensajes de una sesión y el envío de nuevos mensajes. Esta separación responde a que las dos operaciones cuelgan de rutas distintas: una parte del objeto de aprendizaje y la otra de la sesión ya existente.

ChatService concentra la lógica del componente. El método de creación u obtención de sesión garantiza que exista una sola sesión por estudiante y objeto de aprendizaje. El método de envío de mensaje construye el contexto de la conversación, invoca al proveedor de IA y persiste tanto el mensaje del estudiante como la respuesta del asistente dentro de una misma transacción. Para sostener el hilo de la conversación, el servicio recupera el identificador de la respuesta anterior y lo reenvía al proveedor, de modo que solo en el primer mensaje se incluye el contenido completo de la lección como contexto.

El control del uso excesivo de una sesión no reside en el servicio, sino en ChatRa-

teLimitGuard, una clase de tipo guard que se interpone antes del envío de mensajes y rechaza la solicitud cuando se supera el límite configurado. Situar esta verificación en un guard la mantiene separada de la lógica de la conversación.

El paquete se apoya en dos clases externas: LoHelperService, del paquete de objetos de aprendizaje, que resuelve el acceso de lectura al objeto de aprendizaje y verifica los permisos del estudiante; y OpenaiService, del paquete providers/ai, que ejecuta la generación de la respuesta.

La Figura 3.6 presenta el diagrama de clases del paquete features/chat.

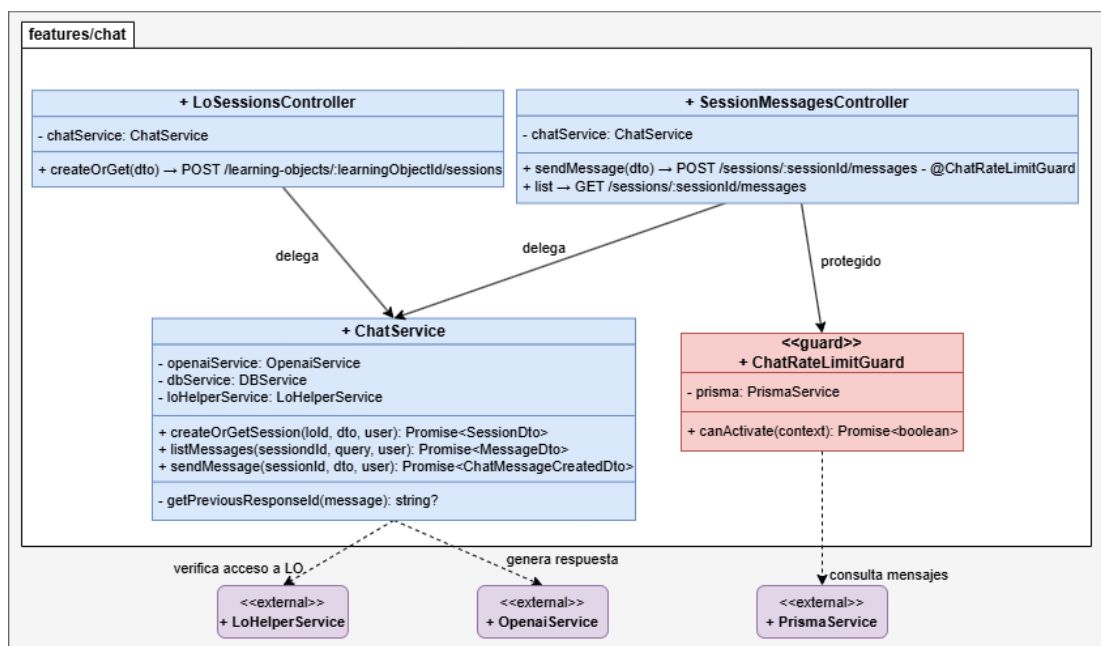


Figura 3.6. Diagrama de clases del paquete features/chat.

Paquete features/teacher-feedback

El paquete features/teacher-feedback implementa la generación de retroalimentación al docente y su consulta. La generación es un proceso periódico; la consulta es una operación que el docente realiza sobre la plataforma.

TeacherFeedbackController es la entrada del paquete para el docente. Expone el listado de retroalimentaciones de un módulo de instrucción y la consulta del detalle de una retroalimentación concreta.

TeacherFeedbackService concentra la lógica del componente. Sus métodos públicos cubren dos responsabilidades. La primera es la consulta: el listado por módulo y la obtención de una retroalimentación, ambos precedidos de la verificación de que

el módulo pertenece al docente que realiza la solicitud. La segunda es la generación: el método de generación para un módulo elabora la retroalimentación de cada objeto de aprendizaje publicado del módulo, la persiste, y con esas piezas construye la retroalimentación agregada del módulo; el método de generación para todos los módulos recorre los módulos activos y aplica sobre cada uno el proceso anterior. Cuando la retroalimentación de módulo se genera, el servicio solicita el envío del correo al docente, sujeto al mismo control de límite diario de envíos que rige para toda la plataforma.

FeedbackDataCollectorService es el servicio de apoyo que reúne los datos de interacción sobre los que se elabora la retroalimentación. Recolecta los mensajes del tutor inteligente, los resultados de las actividades, las valoraciones de los estudiantes y las preguntas del foro asociadas a un objeto de aprendizaje; agrega esos mismos datos a nivel de módulo; y determina si ha habido interacción nueva desde la última retroalimentación, lo que evita regenerar una retroalimentación cuando no hay datos que la justifiquen.

TeacherFeedbackWorker es el proceso de trabajo que ejecuta la generación de forma desatendida. Atiende dos tipos de tarea: la generación para todos los módulos y la generación para un módulo concreto, y en ambos casos delega en TeacherFeedbackService. TeacherFeedbackMapper convierte las entidades almacenadas en los objetos de respuesta que se entregan al docente.

El paquete se apoya en dos clases externas: OpenaiService, del paquete providers/ai, que genera el contenido de la retroalimentación; y EmailService, del paquete providers/email, que envía el correo al docente. EmailService centraliza el control del límite diario de envíos, común a toda la plataforma: antes de cada envío comprueba si se ha alcanzado el límite del día; si no se ha alcanzado, el correo se envía de inmediato; si se ha alcanzado, el correo no se descarta, sino que se reprograma para la mañana del día siguiente, con reintentos automáticos de retroceso exponencial hasta un máximo acotado. Este mecanismo es el mismo para todos los correos del sistema, con independencia del componente que los origine.

La Figura 3.7 presenta el diagrama de clases del paquete features/teacher-feedback.

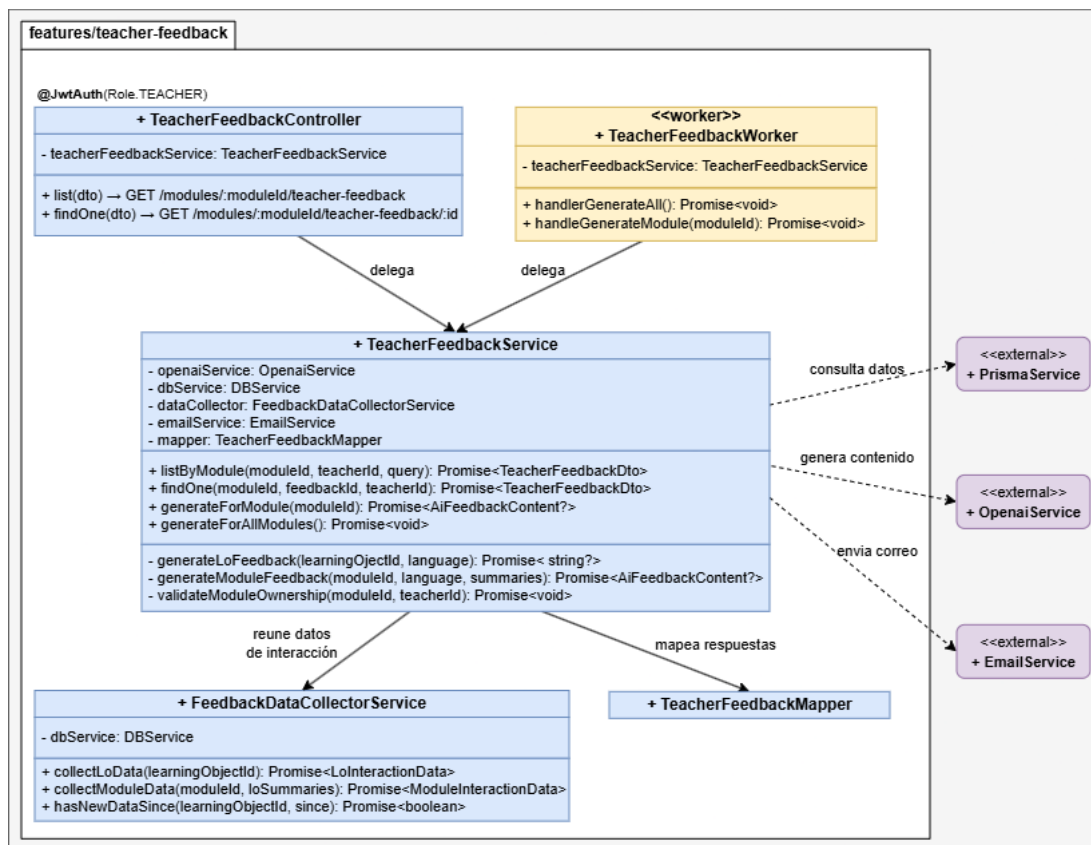


Figura 3.7. Diagrama de clases del paquete features/teacher-feedback.

Paquete features/student-ai-feedback

El paquete features/student-ai-feedback implementa la generación de retroalimentación al estudiante y su consulta. Su estructura es paralela a la del paquete de retroalimentación al docente.

StudentAIFeedbackController es la entrada del paquete para el estudiante. Expone el listado de las retroalimentaciones que ha recibido en un módulo de instrucción y la consulta del detalle de una retroalimentación.

StudentAIFeedbackService concentra la lógica del componente. Sus métodos públicos cubren la consulta —el listado por módulo y la obtención de una retroalimentación, precedidos de la verificación de que el estudiante está matriculado en el módulo— y la generación —para un estudiante en un módulo, y para todos los estudiantes matriculados—. La generación recolecta la actividad del estudiante en la última semana y, si no hay actividad, se omite. Cuando se genera la retroalimentación, el servicio solicita su envío por correo a EmailService.

StudentFeedbackDataCollectorService es el servicio de apoyo que reúne la

actividad del estudiante en la última semana: los objetos de aprendizaje completados, los resultados de las actividades resueltas con su tasa de acierto y las preguntas que planteó en el chat y en el foro.

StudentAIFeedbackWorker es el proceso de trabajo del componente. Atiende las tareas de: generación para todos los estudiantes y generación para un estudiante concreto. StudentFeedbackMapper convierte las entidades almacenadas en los objetos de respuesta que se entregan al estudiante.

El paquete se apoya en dos clases externas: OpenaiService, del paquete providers/ai, que genera el contenido de la retroalimentación; y EmailService, del paquete providers/email, que envía el correo al estudiante.

La Figura 3.8 presenta el diagrama de clases del paquete features/student-ai-feedback.

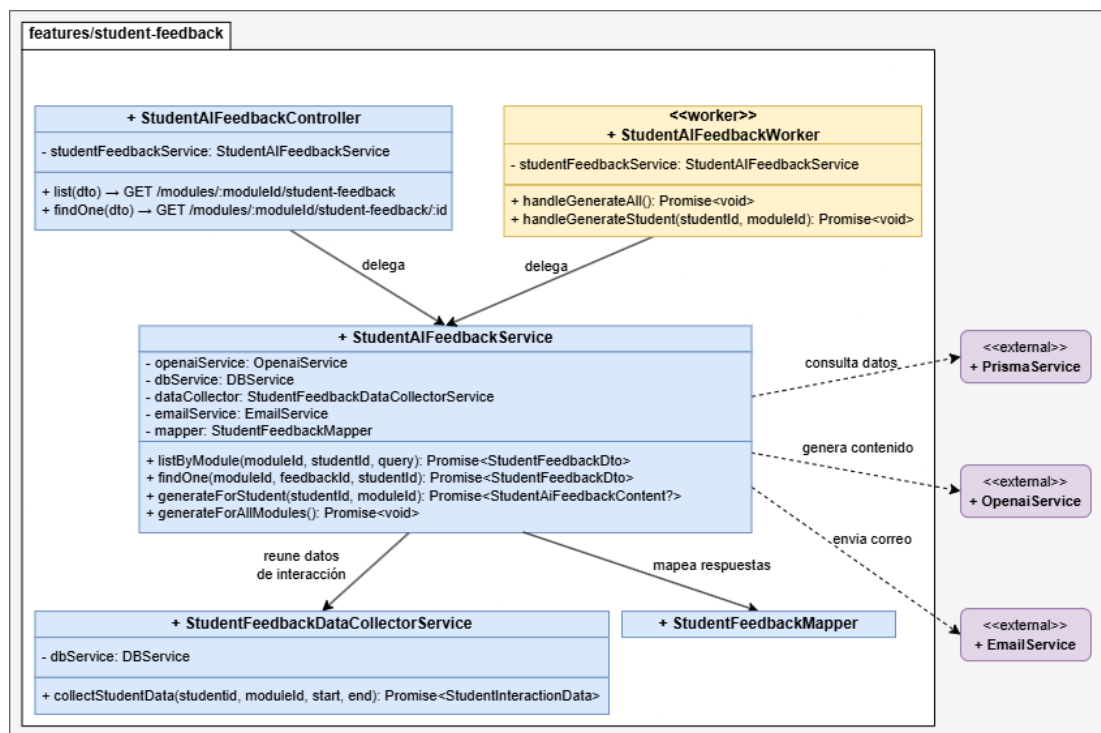


Figura 3.8. Diagrama de clases del paquete features/student-ai-feedback.

Paquetes de interacción del estudiante

Bajo features/interactions se agrupan los paquetes que registran la interacción directa del estudiante con cada objeto de aprendizaje. activities gestiona la resolución de actividades: recibe el intento del estudiante, determina si la respuesta es correcta y lo persiste como un nuevo registro. student-questions y question-

replies sostienen el foro de cada lección: el primero gestiona las preguntas que plantea el estudiante; el segundo, las respuestas, que pueden provenir tanto de otro estudiante como del docente. lo-feedbacks gestiona el feedback que el estudiante registra sobre un objeto de aprendizaje y que el docente consulta. A diferencia de los paquetes de mediación con IA, estos resuelven operaciones de creación, consulta y edición sobre sus propias entidades, sin invocar al modelo de lenguaje; los registros que producen son, además, parte del insumo que los paquetes de retroalimentación procesan más adelante.

Dependencias entre paquetes

Los cuatro paquetes no son independientes entre sí. La Figura 3.9 presenta el diagrama global de dependencias, que resume cómo se relacionan los paquetes de tutoría y de qué paquetes de apoyo dependen.

providers/ai ocupa la posición central: los tres paquetes de mediación con IA — features/chat, features/teacher-feedback y features/student-ai-feedback— dependen de él, ya que toda generación con modelos de lenguaje pasa por ese punto. providers/email es un paquete de apoyo del que dependen los dos paquetes de retroalimentación, porque ambos notifican por correo; features/chat no depende de él, dado que el tutor inteligente no genera notificaciones. Los paquetes de retroalimentación dependen además de los paquetes de features/interactions, de los que obtienen los registros de actividad, foro y feedback que procesan como insumo, y del paquete de objetos de aprendizaje de NousAI, que aporta el contenido de cada lección; esas dependencias se representan de forma agrupada para no recargar el diagrama.

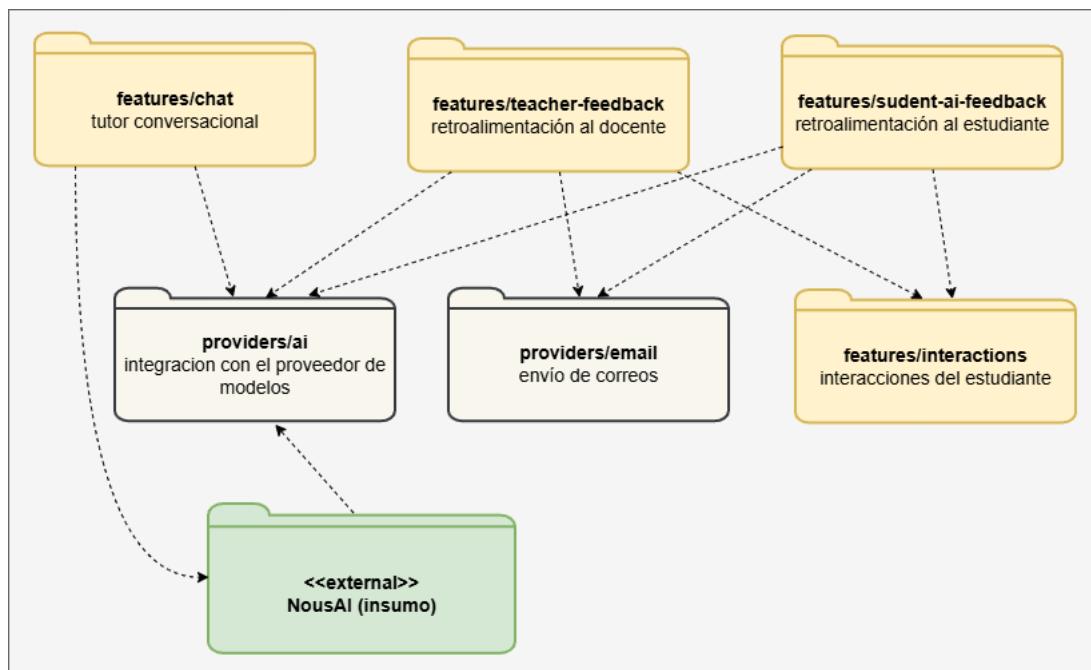


Figura 3.9. Diagrama de dependencias entre paquetes de la tutoría con IA.

3.1.7 Diseño de la API REST

La API REST es el punto por el que la capa de presentación accede a la funcionalidad del backend. Todos los endpoints comparten un prefijo común /api, devuelven sus respuestas con un formato uniforme y requieren autenticación.

Los endpoints se agrupan en cuatro conjuntos: los del tutor inteligente, los de la interacción directa del estudiante con el objeto de aprendizaje —actividades, foro y feedback—, los de consulta de retroalimentación al docente y los de consulta de retroalimentación al estudiante. La generación de retroalimentaciones no expone endpoints de cara al usuario, ya que es un proceso periódico ejecutado por el planificador; por esa razón no aparece en esta documentación.

Formato uniforme de respuesta

Todas las respuestas de la API se entregan dentro de un envoltorio común. Las respuestas de un único recurso usan la estructura ApiRes, que acompaña al dato con información de estado de la operación. Las respuestas de listados usan la estructura ApiPaginatedRes, que añade al conjunto de registros los datos de paginación: total de elementos, página actual, límite por página y número de páginas. Este envoltorio es transversal a todo el backend y se aplica de forma automática, de modo que la capa de

presentación recibe siempre una forma de respuesta predecible.

Los códigos de estado siguen la semántica estándar de HTTP. Una operación de consulta resuelta con éxito devuelve 200; la creación de un recurso devuelve 201. Una solicitud sin credenciales válidas devuelve 401. Una solicitud cuyo usuario no tiene permiso sobre el recurso devuelve 403. Una solicitud sobre un recurso inexistente devuelve 404. En el caso específico del envío de mensajes al tutor, una solicitud que supera el límite de uso de la sesión devuelve 429. Los errores se entregan con el mismo envoltorio uniforme, lo que permite a la capa de presentación tratarlos de forma consistente.

Endpoints del tutor inteligente

La Tabla 3.43 presenta los endpoints del tutor inteligente. La creación de la sesión cuelga de la ruta del objeto de aprendizaje, mientras que el listado y el envío de mensajes cuelgan de la ruta de la sesión ya existente. El envío de mensajes está protegido por una verificación de límite de uso, lo que explica que sea el único endpoint que puede responder con el código 429.

Tabla 3.43. Endpoints del tutor inteligente.

Método	Ruta	Descripción	Acceso	Entrada	Salida
POST	/api/learning-objects/:learningObjectId/sessions	Crea u obtiene la sesión de tutoría del estudiante para un objeto de aprendizaje	Autenticado	CreateOrGetSessionDto	SessionDto
GET	/api/sessions/:sessionId/messages	Lista de forma paginada los mensajes de una sesión	Autenticado	BaseParamsReqDto	ApiPaginatedRes<MessageDto>
POST	/api/sessions/:sessionId/messages	Envía un mensaje del estudiante y devuelve la respuesta del asistente	Autenticado, con límite de uso	SendMessageDto	ChatMessageCreatedDto

Endpoints de interacción con el objeto de aprendizaje

La Tabla 3.44 presenta los endpoints mediante los cuales el estudiante interactúa con el objeto de aprendizaje y el docente participa en esa interacción. Reúnen la resolución de actividades, el foro de preguntas y respuestas, y el feedback del estudiante

sobre el material. La tabla recoge la operación principal de cada flujo; cada recurso admite además las operaciones de actualización y eliminación sobre los registros propios del usuario que los creó.

Tabla 3.44. Endpoints de interacción con el objeto de aprendizaje.

Método	Ruta	Descripción	Acceso
GET	/api/learning-objects/:learningObjectId/activities	Lista las actividades de un objeto de aprendizaje	Autenticado
POST	/api/activities/:activityId/attempts	Registra un intento del estudiante sobre una actividad y devuelve su resultado	Estudiante
POST	/api/pages/student-questions	Crea una pregunta del estudiante en el foro de un objeto de aprendizaje	Autenticado
POST	/api/pages/question-replies	Crea una respuesta a una pregunta del foro, del estudiante o del docente	Autenticado
POST	/api/learning-object-feedbacks	Registra el feedback de un estudiante sobre un objeto de aprendizaje	Autenticado

Endpoints de retroalimentación al docente

La Tabla 3.45 presenta los endpoints de consulta de retroalimentación al docente. Ambos cuelgan de la ruta del módulo de instrucción y están restringidos al rol de docente. El acceso queda además condicionado a que el módulo consultado pertenezca al docente que realiza la solicitud, verificación que el servicio aplica antes de devolver cualquier dato.

Tabla 3.45. Endpoints de consulta de retroalimentación al docente.

Método	Ruta	Descripción	Acceso	Entrada	Salida
GET	/api/modules/:moduleId/teacher-feedback	Lista de forma paginada las retroalimentaciones de un módulo, con filtro opcional por alcance	Docente	ListTeacherFeedbackDto	ApiPaginatedRes<TeacherFeedbackDto>
GET	/api/modules/:moduleId/teacher-feedback/:id	Consulta el detalle de una retroalimentación concreta del módulo	Docente	—	TeacherFeedbackDto

Endpoints de retroalimentación al estudiante

La Tabla 3.46 presenta los endpoints de consulta de retroalimentación al estudiante. Cuelgan de la ruta del módulo de instrucción. El acceso queda condicionado a que el estudiante esté matriculado en el módulo consultado, verificación que el servicio aplica antes de devolver cualquier dato.

Tabla 3.46. Endpoints de consulta de retroalimentación al estudiante.

Método	Ruta	Descripción	Acceso	Entrada	Salida
GET	/api/modules/ :moduleId/ student-feedback	Lista de forma paginada las retroalimentaciones recibidas por el estudiante en un módulo	Estudiante	ListStudent FeedbackDto	ApiPaginated Res<Student FeedbackDto>
GET	/api/modules/ :moduleId/ student-feedback/ :id	Consulta el detalle de una retroalimentación recibida	Estudiante	—	Student FeedbackDto

3.1.8 Flujos

Los tres componentes de mediación con IA resuelven su operación principal mediante una secuencia de llamadas entre la capa de presentación, el backend y los servicios externos. Se documentan aquí con su diagrama de secuencia porque coordinan varias llamadas y dependen del proveedor de IA; la interacción directa del estudiante con el objeto de aprendizaje —intentar una actividad, plantear o responder una pregunta del foro, registrar feedback— se resuelve con operaciones de persistencia directa, sin esa coordinación, y queda descrita en los casos de uso y en la API REST. El flujo del tutor inteligente se activa por una acción del estudiante; los de generación de retroalimentación al docente y al estudiante son procesos periódicos que activa el planificador. Un elemento común a los tres es la llamada al proveedor de IA, que incorpora internamente mecanismos de reintento con retroceso exponencial ante fallos transitorios y ante respuestas que no cumplan el formato establecido, con un máximo de tres intentos en ambos casos.

Flujo del tutor inteligente

El flujo del tutor inteligente corresponde al envío de un mensaje por parte del estudiante dentro de una sesión de tutoría. Supone que la sesión ya existe, es decir, que el estudiante abrió previamente el objeto de aprendizaje y se creó u obtuvo su sesión asociada.

El estudiante envía su mensaje a la API. Antes de que la solicitud llegue a la lógica del componente, el guard de límite de uso verifica que la sesión no haya superado el umbral configurado; si lo ha superado, la solicitud se rechaza en ese punto y el estudiante recibe una respuesta de límite excedido. Si la verificación pasa, ChatService recupera la sesión y comprueba que pertenezca al estudiante y que este tenga acceso de lectura al objeto de aprendizaje.

A continuación, el servicio determina si es el primer mensaje de la conversación. Para ello recupera el identificador de la respuesta anterior, almacenado en los metadatos del último mensaje del asistente. Si no existe ese identificador, se trata del primer mensaje y el servicio incluye el contenido completo de la lección como contexto; si existe, el servicio solo envía el mensaje del estudiante, porque el proveedor de IA mantiene el hilo de la conversación a partir de ese identificador. Con esos elementos se construye el prompt y se solicita la respuesta a OpenaiService.

Una vez recibida la respuesta del asistente, el servicio persiste en una misma transacción el mensaje del estudiante y el mensaje del asistente; este último guarda en sus metadatos el nuevo identificador de respuesta, que servirá de enlace para el siguiente turno. Finalmente, la API devuelve al estudiante el mensaje del asistente.

La Figura 3.10 presenta el diagrama de secuencia del flujo del tutor inteligente.

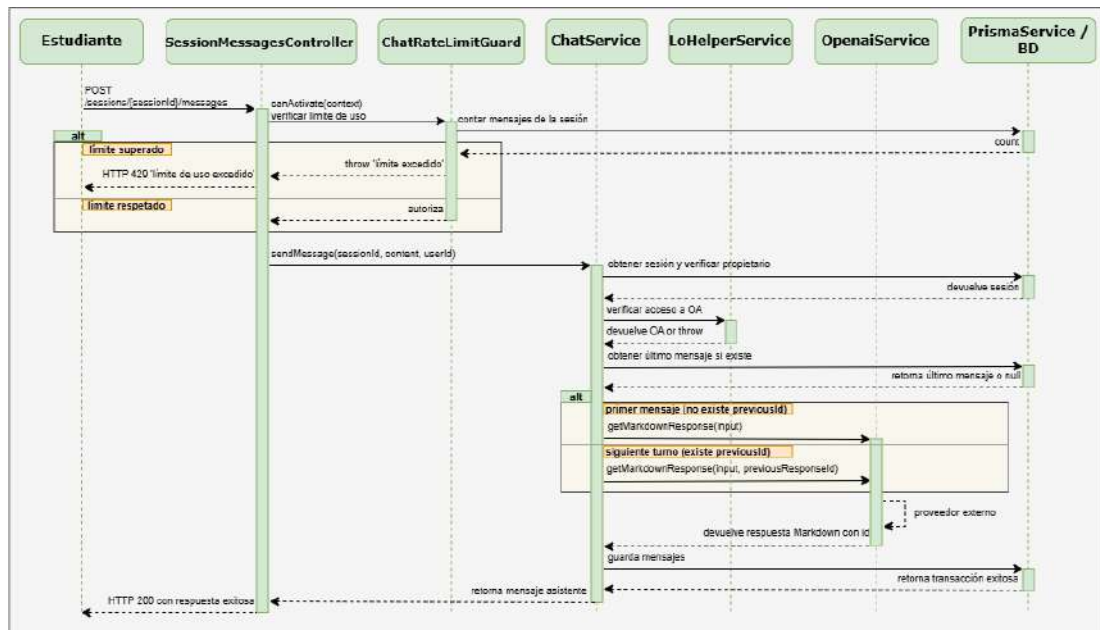


Figura 3.10. Diagrama de secuencia del flujo del tutor inteligente.

Flujo de la generación de retroalimentación al docente

La generación de retroalimentación al docente es un proceso periódico, sin intervención humana, que el planificador activa de forma semanal. Su resultado son las retroalimentaciones por objeto de aprendizaje y por módulo de instrucción, junto con la entrega de la retroalimentación de módulo al docente por correo.

El planificador encola la tarea de generación para todos los módulos, que el proceso de trabajo recoge y delega en TeacherFeedbackService. El servicio obtiene la lista de módulos de instrucción activos y los procesa uno a uno. Para cada módulo recupera sus objetos de aprendizaje publicados y, antes de generar nada, recurre a FeedbackDataCollectorService para decidir si cada objeto de aprendizaje tiene material suficiente.

Esa decisión combina dos comprobaciones. La primera verifica si ha habido interacción nueva desde la última retroalimentación de ese objeto de aprendizaje; si no la hay, el objeto se omite, lo que evita regenerar una retroalimentación que sería idéntica a la anterior. La segunda verifica que se alcancen los umbrales mínimos de participación: un número mínimo de estudiantes que han interactuado y un número mínimo de interacciones concretas. Solo los objetos de aprendizaje que superan ambas comprobaciones pasan a la generación.

Para cada objeto de aprendizaje que supera el filtro, el servicio reúne sus datos

de interacción —mensajes del tutor, resultados de actividades, valoraciones de los estudiantes y preguntas del foro—, construye el prompt correspondiente y solicita la retroalimentación a OpenaiService. La respuesta se persiste como retroalimentación de alcance de objeto de aprendizaje, y su resumen se conserva para el paso siguiente. Una vez procesados todos los objetos de aprendizaje del módulo, el servicio agrega los resúmenes obtenidos junto con las preguntas más relevantes del foro, construye el prompt de módulo y solicita a OpenaiService la retroalimentación agregada, que se persiste con alcance de módulo.

Si la retroalimentación de módulo se generó, el servicio solicita a EmailService el envío del correo al docente con el contenido de esa retroalimentación; el control del límite diario de envíos aplica de forma centralizada en EmailService. Si el módulo no tenía datos suficientes y no se generó ninguna retroalimentación de módulo, no se envía correo. El proceso continúa con el siguiente módulo hasta agotar la lista.

La Figura 3.11 presenta el diagrama de secuencia del flujo de generación de retroalimentación al docente.

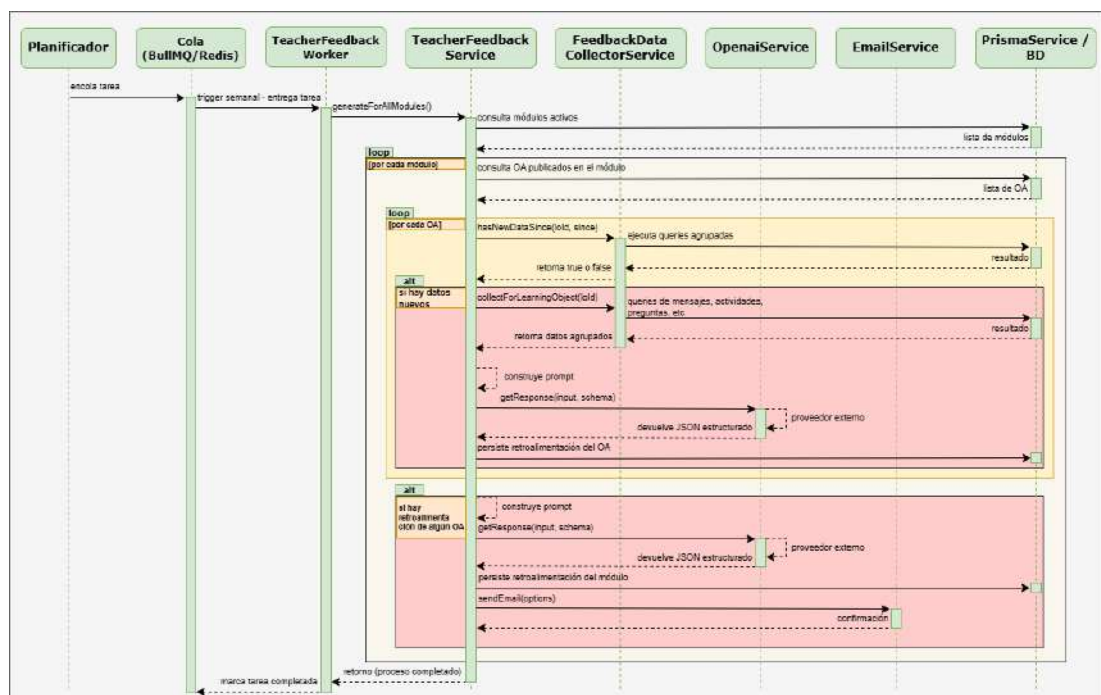


Figura 3.11. Diagrama de secuencia del flujo de generación de retroalimentación al docente.

Flujo de la generación de retroalimentación al estudiante

La generación de retroalimentación al estudiante es también un proceso periódico semanal activado por el planificador. Su resultado es una retroalimentación personalizada por estudiante y módulo de instrucción, que además se entrega por correo con su contenido completo.

El planificador encola la tarea de generación para todos los estudiantes, que el proceso de trabajo recoge y delega en StudentAIFeedbackService. El servicio obtiene la lista de matrículas activas y las procesa una a una. Para cada matrícula, delimita la última semana y recurre a StudentFeedbackDataCollectorService para reunir la actividad del estudiante en ese periodo: los objetos de aprendizaje completados, los resultados de las actividades resueltas con su tasa de acierto, los conceptos en los que cometió errores y las preguntas que planteó en el chat y en el foro.

Si el estudiante no registró actividad en la semana, la matrícula se omite y no se genera retroalimentación. Si hubo actividad, el servicio construye el prompt con los datos reunidos, solicita la retroalimentación a OpenaiService y persiste el resultado.

Tras persistir la retroalimentación, el servicio solicita su envío por correo a EmailService; el control del límite diario de envíos aplica de forma centralizada en EmailService. El proceso continúa con la siguiente matrícula hasta agotar la lista.

La Figura 3.12 presenta el diagrama de secuencia del flujo de generación de retroalimentación al estudiante.

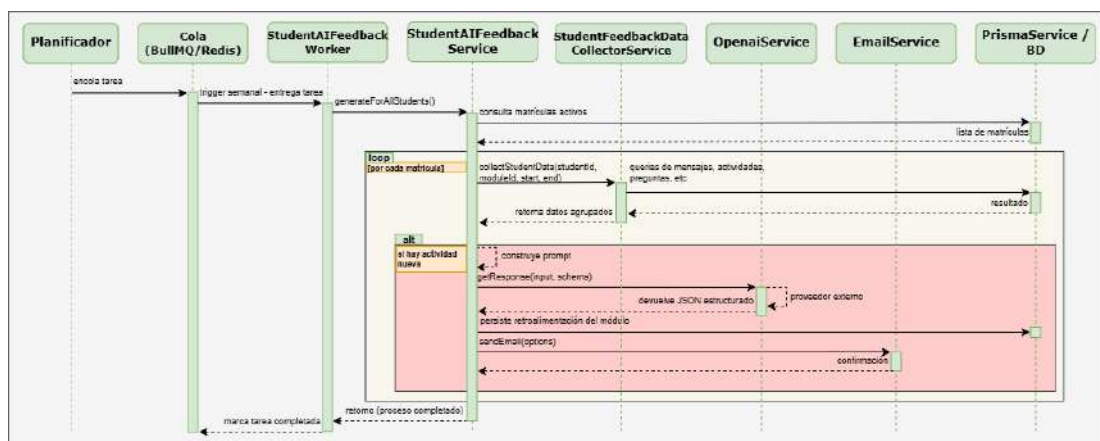


Figura 3.12. Diagrama de secuencia del flujo de generación de retroalimentación al estudiante.

3.1.9 Diseño de prompts

Un prompt es el conjunto de instrucciones que recibe el modelo de lenguaje antes de producir una respuesta. De su diseño dependen la pertinencia de la salida, su formato y los límites dentro de los que el modelo opera. En esta propuesta, el diseño de prompts es el mecanismo que sostiene el carácter supervisado de los componentes que no admiten revisión caso a caso: en lugar de revisar cada respuesta del modelo, el prompt restringe de antemano lo que puede producir y la salida se verifica después de forma automática.

El sistema emplea doce prompts repartidos en tres dominios: generación de contenido, tutoría conversacional y retroalimentación. Los cuatro prompts de los dominios de tutoría conversacional y de retroalimentación corresponden a los componentes de IA desarrollados en este trabajo; los ocho de generación de contenido pertenecen a la propuesta de gestión de wiki basada en IA generativa. Los doce, sin embargo, comparten un mismo conjunto de principios de diseño, que constituyen un aporte de este trabajo. A continuación se definen esos principios y luego se caracteriza el catálogo de prompts utilizado.

Principios de diseño de prompts

Separación entre instrucción y datos. Cada prompt se construye con dos mensajes de roles distintos. El mensaje de rol system define quién es el modelo, qué formato debe producir y qué reglas debe respetar; es la parte estable, que no cambia entre una ejecución y otra. El mensaje de rol user inyecta los datos concretos de cada caso: el título de una lección, la pregunta de un estudiante, los datos de interacción de un objeto de aprendizaje. Esta separación mantiene constante el comportamiento del modelo mientras varía únicamente el contexto, y evita que los datos de entrada puedan reinterpretarse como instrucciones.

Asignación de un rol experto. El mensaje system comienza siempre asignando al modelo un rol profesional concreto y acotado al dominio de la tarea. Según el prompt, el modelo actúa como creador de contenido educativo, como tutor educativo, como consultor educativo o como analista de aprendizaje. Fijar el rol orienta el registro, el nivel de detalle y el tipo de razonamiento de la respuesta antes de que el modelo reciba dato alguno.

Composición modular. Los prompts no se redactan como textos monolíticos, sino que se componen a partir de fragmentos reutilizables: la instrucción de responder solo en formato JSON, las reglas por tipo de bloque, las reglas de preservación de marcadores, los bloques de configuración pedagógica. Un mismo fragmento se emplea en varios prompts, de modo que una regla puede ajustarse en un único lugar y el cambio alcanza a todos los prompts que la incorporan. Este principio se aplica de forma más intensa en los prompts de generación de contenido, donde el número de reglas compartidas es mayor.

Tipado fuerte de la entrada. Cada prompt declara una interfaz que define con exactitud los datos que necesita para construirse. De este modo, los datos requeridos se comprueban en tiempo de compilación y no es posible invocar un prompt con información incompleta.

Salida estructurada y verificable. Todos los prompts que alimentan una funcionalidad —es decir, todos salvo el del tutor inteligente, que produce texto libre— exigen al modelo una respuesta en formato JSON con una estructura explícita, e incluyen en el mensaje `system` un ejemplo concreto de esa estructura. La respuesta se valida después contra un esquema definido con la biblioteca `Zod`, que la rechaza si no se ajusta a lo esperado. El prompt acota lo que el modelo debería producir; la validación posterior comprueba que efectivamente lo produjo.

Restricciones explícitas. Cada prompt enuncia de forma expresa lo que el modelo no debe hacer. El prompt del tutor inteligente restringe las respuestas al contenido de la lección; el de retroalimentación al estudiante prohíbe mencionar que el texto proviene de una IA; el de retroalimentación al docente prohíbe inventar datos y exige trabajar sobre patrones agregados y no sobre estudiantes individuales. Estas restricciones son el mecanismo por el que el diseño del prompt sustituye a la revisión caso a caso.

Configuración pedagógica parametrizada. El comportamiento de la IA se ajusta por módulo de instrucción sin modificar los prompts. Parámetros como el idioma, la audiencia, el nivel y el tono se inyectan en el mensaje correspondiente a partir de la configuración del módulo. Un mismo prompt produce así respuestas adecuadas a contextos pedagógicos distintos.

Secciones condicionales. Los prompts de retroalimentación construyen el mensaje `user` por composición de secciones según los datos disponibles para cada caso. Esta

construcción modular permite que un mismo prompt se adapte a la cantidad y el tipo de información de cada estudiante, objeto de aprendizaje o módulo de instrucción, sin redactar un prompt distinto para cada combinación.

Marcadores especiales. El contenido educativo admite marcadores con un formato propio que representan enlaces a conceptos y a otras páginas. Los prompts que manipulan ese contenido incluyen reglas explícitas para preservar los marcadores existentes y no alterarlos, de modo que la navegación semántica entre objetos de aprendizaje se mantenga intacta tras una regeneración.

Catálogo de prompts

La Tabla 3.47 resume los doce prompts del sistema, organizados por dominio. Para cada uno se indica su propósito, el rol experto que asigna al modelo y el tipo de salida que produce.

Tabla 3.47. Catálogo de prompts del sistema.

Dominio	Prompt	Propósito	Rol asignado	Salida
Generación de contenido	generate-page-content	Genera el contenido de una lección desde cero a partir de un título	Creador de contenido educativo	JSON estructurado
Generación de contenido	regenerate-page-content	Modifica el contenido completo de una lección según instrucciones del docente	Editor de contenido educativo	JSON estructurado
Generación de contenido	regenerate-block	Regenera un bloque individual con contexto de los bloques adyacentes	Editor de contenido educativo	JSON estructurado
Generación de contenido	expand-content	Genera bloques adicionales para ampliar una lección en una posición dada	Creador de contenido educativo	JSON estructurado
Generación de contenido	generate-activity	Genera una actividad evaluativa de uno de cuatro tipos	Creador de evaluaciones educativas	JSON estructurado
Generación de contenido	extract-lo-concepts	Extrae términos clave de una lección para mostrarlos como conceptos	Analizador de contenido educativo	JSON estructurado
Generación de contenido	generate-concept	Genera la definición de un término concreto seleccionado por el docente	Creador de contenido educativo	JSON estructurado

Tabla 3.47. Catálogo de prompts del sistema (continuación)

Dominio	Prompt	Propósito	Rol asignado	Salida
Generación de contenido	generate-relations	Identifica relaciones semánticas entre páginas para crear enlaces cruzados	Analista de contenido educativo	JSON estructurado
Tutoría conversacional	chat-session	Atiende las consultas del estudiante restringidas al contenido de la lección	Asistente educativo	Texto libre en Markdown
Retroalimentación	student-feedback	Genera la retroalimentación semanal personalizada para el estudiante	Consultor educativo	JSON estructurado
Retroalimentación	lo-feedback	Genera la retroalimentación al docente sobre un objeto de aprendizaje	Consultor educativo y analista de aprendizaje	JSON estructurado
Retroalimentación	module-feedback	Sintetiza la retroalimentación al docente a nivel de módulo de instrucción	Consultor educativo	JSON estructurado

Los ocho prompts del dominio de generación de contenido pertenecen a la propuesta paralela de generación de contenido y aplican los principios descritos, con un uso intensivo de la composición modular por la cantidad de reglas de formato que comparten. Los cuatro prompts de los dominios de tutoría conversacional y de retroalimentación corresponden a los componentes desarrollados en este trabajo y se describen a continuación.

Prompt del tutor inteligente (chat-session). Es el único prompt del sistema que produce texto libre en lugar de JSON, ya que su salida es la respuesta conversacional que lee el estudiante. El mensaje system asigna al modelo el rol de asistente educativo y fija la restricción central del componente: responder apoyándose exclusivamente en el contenido de la lección proporcionada y en conocimiento general básico, e indicar de forma explícita cuando una pregunta queda fuera de ese contenido. Las demás reglas determinan la forma de la respuesta: explicaciones claras y paso a paso cuando corresponde, formato Markdown, y un tono de apoyo. El mecanismo de contexto es particular: el contenido de la lección se inyecta únicamente en el primer mensaje de la sesión; en los mensajes siguientes, el hilo de la conversación mantiene el contexto, lo que evita reenviar la lección completa en cada turno. De los doce prompts, es el de

entrada más reducida, porque no necesita componer reglas de formato estructurado.

Prompt de retroalimentación al estudiante (student-feedback). El mensaje system asigna al modelo el rol de consultor educativo y define una salida JSON con cuatro campos: un resumen ejecutivo de la semana, una lista de fortalezas, una lista de dificultades y una lista de recomendaciones; cada elemento de las tres listas incluye un tema, un detalle y un nivel de prioridad. Entre sus reglas destacan dos restricciones propias del componente: el texto nunca debe mencionar que proviene de una IA, de un sistema o de un algoritmo, y debe centrarse en las dificultades específicas del estudiante y no en datos agregados de la clase. El tono se define de forma expresa como el de un acompañante semanal del aprendizaje, no como una evaluación. El mensaje user se construye con secciones que presentan la actividad del estudiante en la semana: los objetos de aprendizaje completados, las actividades resueltas con su tasa de acierto, los conceptos en los que cometió errores y las preguntas que planteó en el chat y en el foro.

Prompt de retroalimentación al docente por objeto de aprendizaje (lo-feedback). El mensaje system asigna al modelo el rol de consultor educativo y analista de aprendizaje, y define una salida JSON con cuatro campos: un resumen ejecutivo, las fortalezas, los puntos de mejora y las recomendaciones; cada elemento de las tres últimas listas lleva además un nivel de prioridad. Las reglas exigen trabajar sobre patrones y no sobre estudiantes individuales, tratar los datos de forma anónima, limitar el número de elementos por categoría y no inventar información que no esté en los datos. El mensaje user se compone con secciones condicionales que aportan el contenido del objeto de aprendizaje y los distintos registros de interacción: mensajes del tutor, resultados de actividades, valoraciones de los estudiantes, preguntas del foro y notas.

Prompt de retroalimentación al docente por módulo (module-feedback). El mensaje system asigna al modelo el rol de consultor educativo y produce una salida con la misma estructura de cuatro campos que el prompt anterior, lo que mantiene un formato uniforme entre los dos alcances de la retroalimentación al docente. Su particularidad está en la naturaleza de su entrada y en sus reglas: en lugar de datos de interacción en bruto, recibe los resúmenes de las retroalimentaciones ya generadas para cada objeto de aprendizaje del módulo, junto con las preguntas más relevantes del

foro, y su tarea es sintetizar los patrones recurrentes a través de todos ellos.

Prompts y supervisión estructural

Los principios descritos no operan de forma aislada: actúan en conjunto sobre cada respuesta del modelo. La separación entre instrucción y datos y la asignación de un rol experto fijan el comportamiento esperado antes de que el modelo reciba la entrada. Las restricciones explícitas, la configuración pedagógica parametrizada y las secciones condicionales acotan lo que el modelo puede producir y bajo qué condiciones. La salida estructurada y su validación posterior comprueban que el resultado tenga la forma prevista antes de que llegue a un usuario o se persista.

Esta combinación es la que permite sostener la lectura de “supervisada” en los componentes que no admiten revisión caso a caso. El tutor inteligente responde en tiempo real y la retroalimentación se genera de forma periódica y desatendida; en ninguno de los dos casos un docente revisa cada salida antes de su entrega. Lo que sustituye a esa revisión no es la confianza en el modelo, sino un diseño que restringe la generación por adelantado y la verifica después. En ese sentido, el diseño de prompts traslada la supervisión: de una acción humana caso a caso a una propiedad estructural verificada antes de que la salida llegue al usuario.

3.1.10 Diseño de la interfaz de usuario

Las decisiones de diseño de la interfaz de tutoría con IA se toman bajo tres principios transversales.

Consistencia con NousAI. Las vistas de tutoría se integran en la aplicación web existente y reutilizan sus componentes visuales: la paleta, la tipografía, los controles de entrada, los patrones de navegación y la estructura de páginas. Esta consistencia, recogida en el requisito de usabilidad RNF-12, evita que el estudiante o el docente perciban la tutoría como una herramienta separada y reduce la curva de aprendizaje para quienes ya emplean la plataforma para otras tareas.

Presentación legible del contenido generado. Las respuestas del tutor inteligente y las retroalimentaciones generadas se reciben con formato Markdown y se presentan al usuario interpretadas: encabezados, listas, código y énfasis se renderizan con el mismo estilo que el resto del contenido educativo de NousAI. Esto mantiene la legibilidad y refuerza la continuidad visual con el material que el usuario ya está acostumbrado a

leer en la plataforma.

Vistas de interacción con el objeto de aprendizaje

Dentro de la lectura de un objeto de aprendizaje, el estudiante dispone de tres formas de interacción directa que se integran en la propia página de la lección: la resolución de actividades, el foro de preguntas y el feedback sobre el material. Situarlas junto al contenido responde a la misma lógica que el resto de la tutoría: cada interacción tiene sentido en el contexto de la lección a la que pertenece, y no en una sección separada de la aplicación.

Las actividades se presentan en un apartado propio dentro del objeto de aprendizaje, donde el estudiante encuentra las que el docente ha publicado y puede resolverlas. Al enviar una respuesta, el sistema le indica de inmediato si es correcta y conserva el registro del intento, de modo que el estudiante puede volver a intentarlo y seguir su propio avance.

El foro y el feedback comparten la misma página de la lección y constituyen los dos canales de comunicación directa entre estudiante y docente sobre el material. En el foro, el estudiante plantea preguntas que quedan visibles en la lección y que el docente —u otros estudiantes— pueden responder; el feedback, en cambio, es un comentario que el estudiante dirige al docente sobre el objeto de aprendizaje y que el docente consulta al revisar esa lección. La Figura 3.13 reúne las tres formas de interacción tal como se presentan en la vista del objeto de aprendizaje. A la izquierda aparecen el feedback registrado por los estudiantes y, debajo, el foro de preguntas, donde se distingue la respuesta del docente del resto de aportes. A la derecha se muestra la resolución de una actividad de opción múltiple: en la parte superior, el momento del intento, y en la inferior, la confirmación del resultado una vez enviada la respuesta.

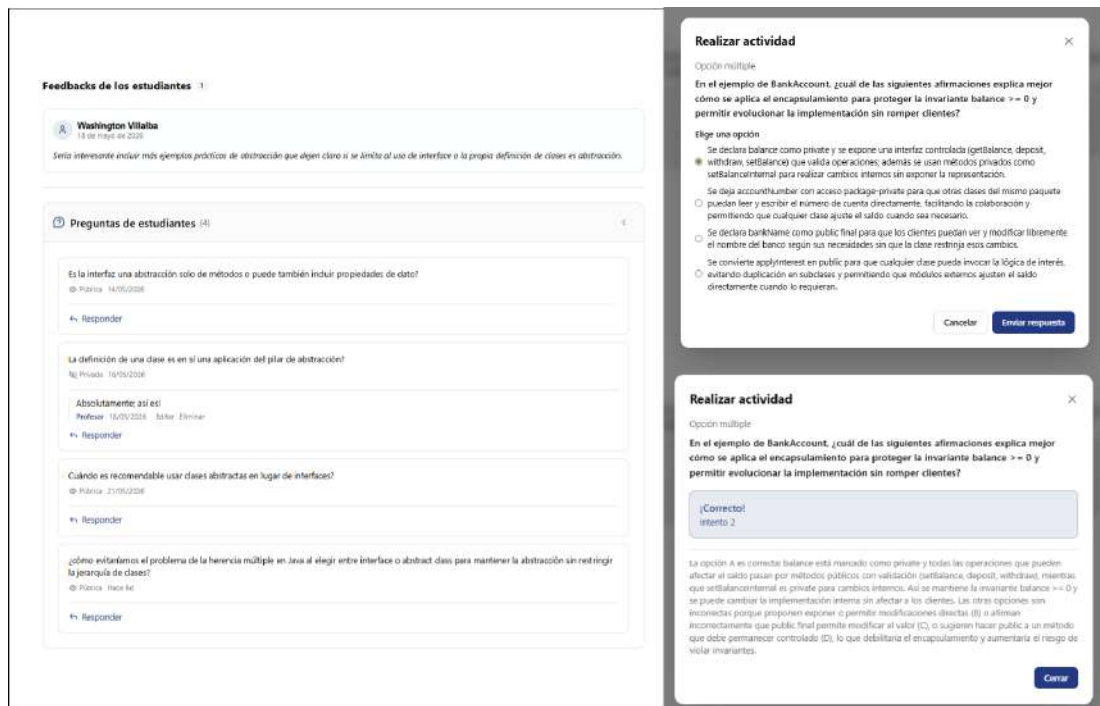


Figura 3.13. Actividades, foro de preguntas y registro de feedback en la vista del objeto de aprendizaje.

Vistas del tutor inteligente

El tutor inteligente se accede desde la lectura de un objeto de aprendizaje. La sesión se asocia al objeto de aprendizaje activo, por lo que la entrada al chat se sitúa en el contexto del propio material y no en una sección independiente de la aplicación. Esta decisión refleja una característica fundamental del componente: la conversación está acotada a la lección, y la interfaz lo expresa al integrar la entrada al chat en la página de la lección.

La vista de chat (Figura 3.14) presenta los elementos habituales de una conversación: la lista de mensajes intercambiados ordenados cronológicamente, con distinción visual entre los mensajes del estudiante y los del asistente, y un área de entrada en la parte inferior para escribir el siguiente mensaje. Los mensajes del asistente se renderizan desde Markdown, lo que permite que las explicaciones aprovechen estructura, listas y código sin perder formato.

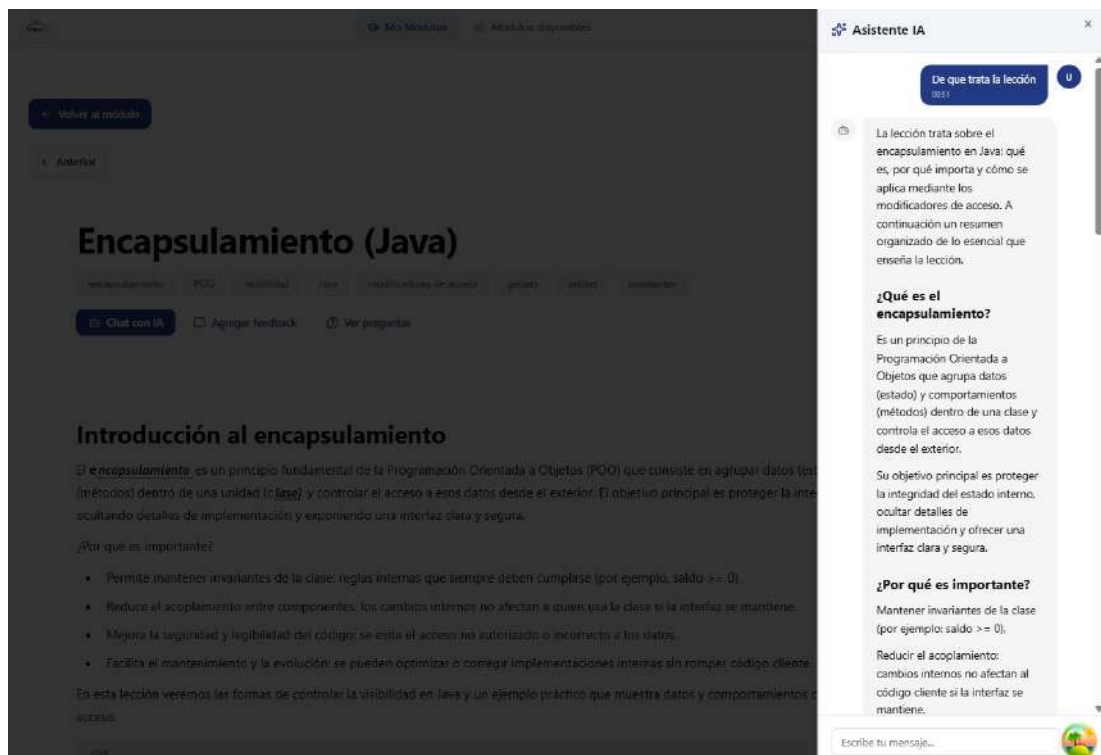


Figura 3.14. Vista de chat del tutor inteligente.

Vistas de retroalimentación al docente

El docente accede a las retroalimentaciones generadas desde la página de su módulo de instrucción. La vista de listado (Figura 3.15) presenta las retroalimentaciones del módulo ordenadas por fecha, con dos elementos de filtro visibles: la pertenencia al módulo, que es implícita por la página en la que se encuentra, y el alcance, que permite separar las retroalimentaciones de objeto de aprendizaje de las de módulo. Cada elemento del listado muestra la fecha de generación, el alcance, el objeto de aprendizaje asociado cuando corresponde y un fragmento del resumen, lo suficiente para que el docente identifique de un vistazo qué retroalimentación quiere consultar.

La decisión de mantener ambos alcances en un único listado responde a que el docente trabaja con el módulo como unidad: las retroalimentaciones de objetos de aprendizaje y la retroalimentación agregada del módulo se complementan, y separarlas en dos pantallas obligaría a una navegación adicional sin aportar claridad. El filtro por alcance resuelve los casos en los que el docente quiere mirar solo uno de los dos.

La vista de detalle (Figura 3.16) presenta el contenido completo de una retroalimentación, organizado en las cuatro secciones que produce el modelo: resumen, fortalezas, puntos de mejora y recomendaciones. Cada elemento de las tres últimas secciones lle-

va su nivel de prioridad como etiqueta visual, lo que permite al docente identificar rápidamente los aspectos más relevantes. El contenido se renderiza con la tipografía y los espaciados habituales de NousAI, sin estilos propios del componente, para que la lectura no rompa la continuidad con el resto de la plataforma.

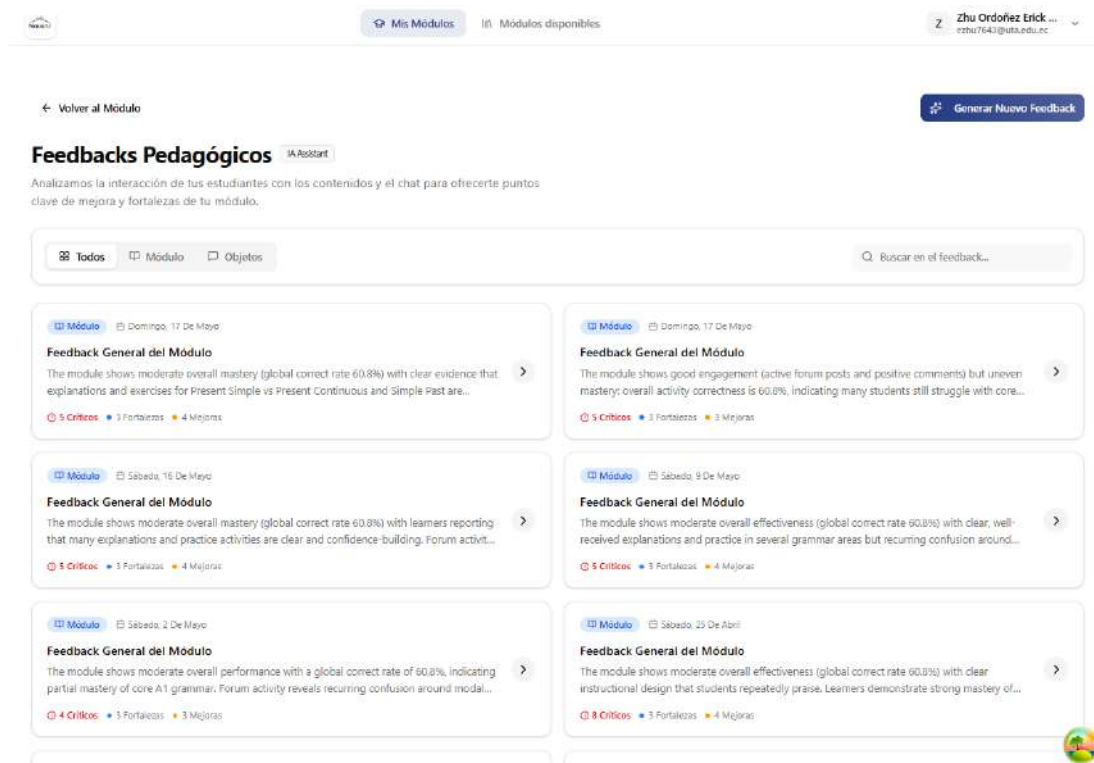


Figura 3.15. Vista del listado de retroalimentaciones al docente en un módulo de instrucción.

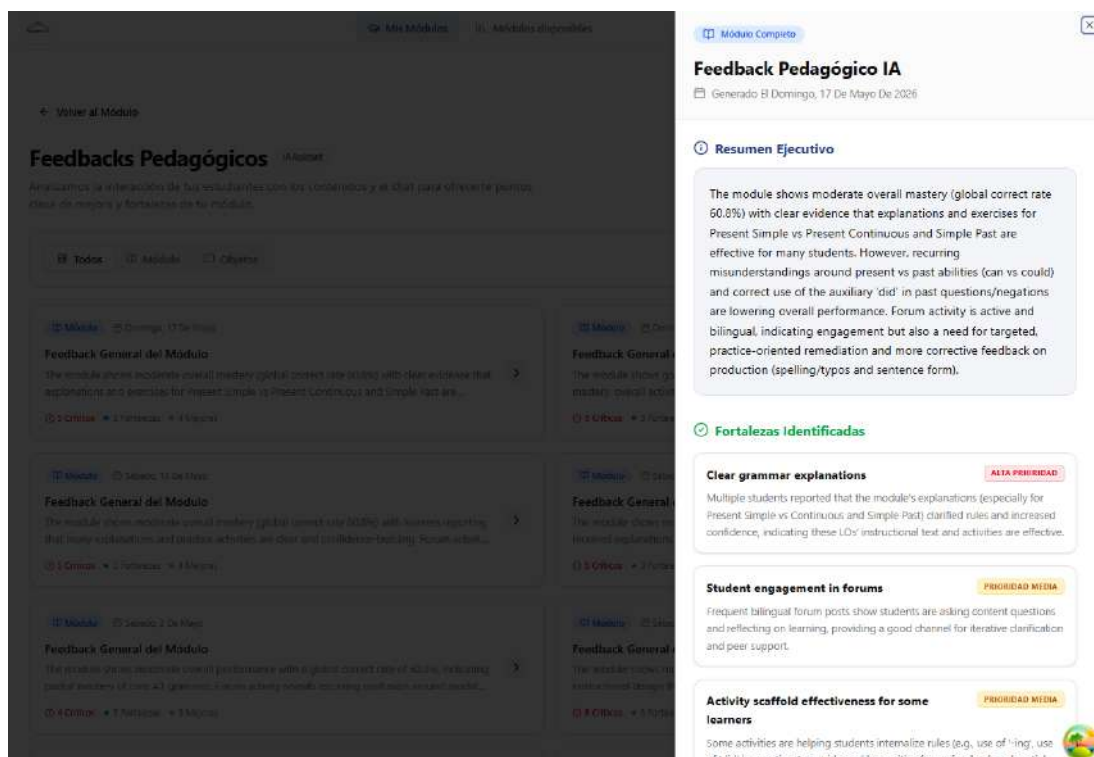


Figura 3.16. Vista del detalle de una retroalimentación al docente.

Vistas de retroalimentación al estudiante

El estudiante accede a las retroalimentaciones recibidas desde la página del módulo de instrucción en el que está matriculado. El acceso se ubica junto al material del módulo y no en una sección global de la aplicación, porque cada retroalimentación está asociada a un módulo concreto y carece de sentido fuera de ese contexto.

La vista de listado (Figura 3.17) presenta las retroalimentaciones recibidas en orden cronológico inverso, con la más reciente en primer lugar. Cada elemento del listado muestra la fecha de la retroalimentación y un fragmento del resumen semanal, lo suficiente para que el estudiante reconozca a qué semana corresponde.

La vista de detalle (Figura 3.18) presenta el contenido completo de una retroalimentación con la misma organización en cuatro secciones descrita para la vista del docente, salvo que el segundo y tercer apartado se titulan, para el estudiante, fortalezas y dificultades. Cada elemento de las tres últimas secciones lleva un nivel de prioridad como etiqueta visual, lo que permite al estudiante identificar los aspectos más relevantes. Las secciones tienen un encabezado propio y un estilo visual coherente con el resto del material del módulo.

El orden de las secciones no es accidental. Comienza por el resumen de la semana, que sitúa al estudiante; continúa con las fortalezas, que refuerzan lo que hizo bien; sigue con las dificultades y las recomendaciones, que son la parte accionable de la retroalimentación. Esta progresión refleja el tono que el prompt fija para el modelo: el de un acompañante semanal del aprendizaje, no el de una evaluación.

El orden de las secciones no es accidental. Comienza por el resumen de la semana, que sitúa al estudiante; continúa con las fortalezas, que refuerzan lo que hizo bien; sigue con las dificultades y las recomendaciones, que son la parte accionable de la retroalimentación. Esta progresión refleja el tono que el prompt fija para el modelo: el de un acompañante semanal del aprendizaje, no el de una evaluación.

Figura 3.17. Vista del listado de retroalimentaciones recibidas por el estudiante en un módulo de instrucción.

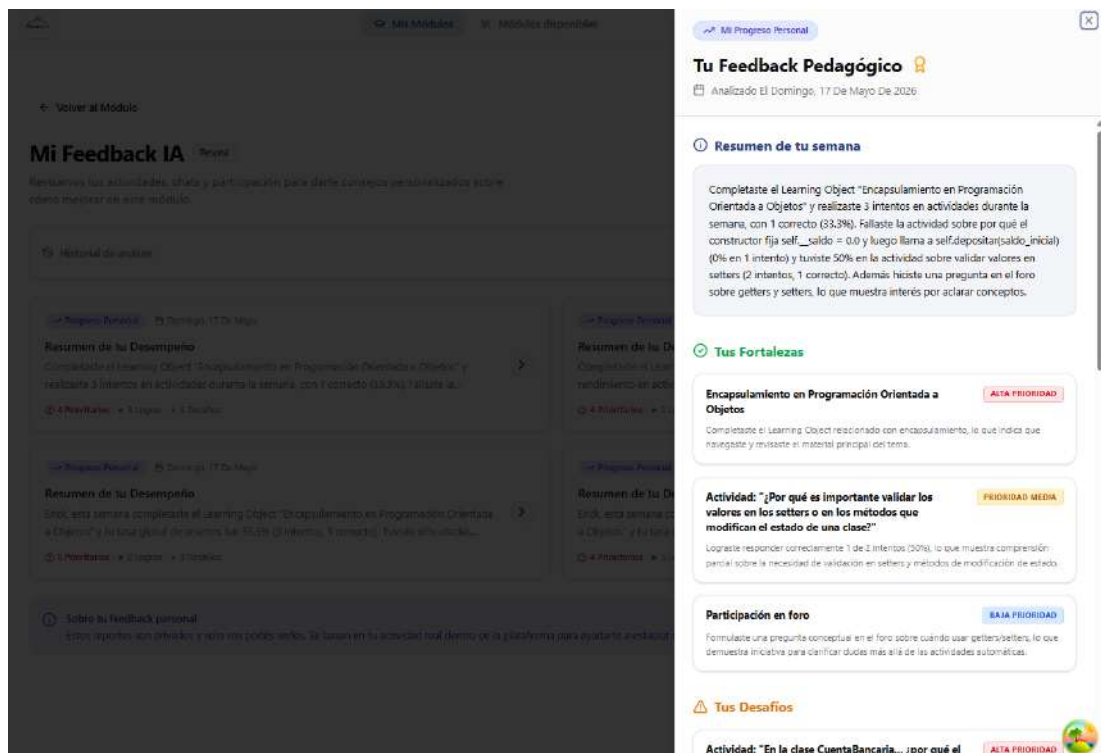


Figura 3.18. Vista del detalle de una retroalimentación recibida por el estudiante.

Entrega por correo electrónico

El correo electrónico es otro canal por el que se entrega la retroalimentación, junto con la consulta dentro de la plataforma. El cuerpo del correo contiene la retroalimentación completa, con las mismas secciones que la vista de detalle. Las dos plantillas comparten una estructura común: encabezado de marca, saludo personalizado, el contenido íntegro de la retroalimentación y un enlace de acceso directo a la vista de detalle dentro de la plataforma.

El correo al docente se envía con la generación de la retroalimentación de módulo de instrucción. Su cuerpo reproduce el título del módulo, la fecha y las mismas cuatro secciones de la vista de detalle. El envío se realiza una sola vez por módulo y no por cada retroalimentación de objeto de aprendizaje generada en el mismo proceso.

El correo al estudiante contiene su retroalimentación semanal completa, con las mismas cuatro secciones de la vista de detalle y en el mismo orden. El saludo incluye el nombre del estudiante y el módulo de instrucción. Cuando el envío se difiere por superar el límite diario de la plataforma, el correo llega de forma idéntica el día siguiente; lo único que varía es la fecha de recepción.



Figura 3.19. Plantilla del correo de retroalimentación al docente (Parte 1).

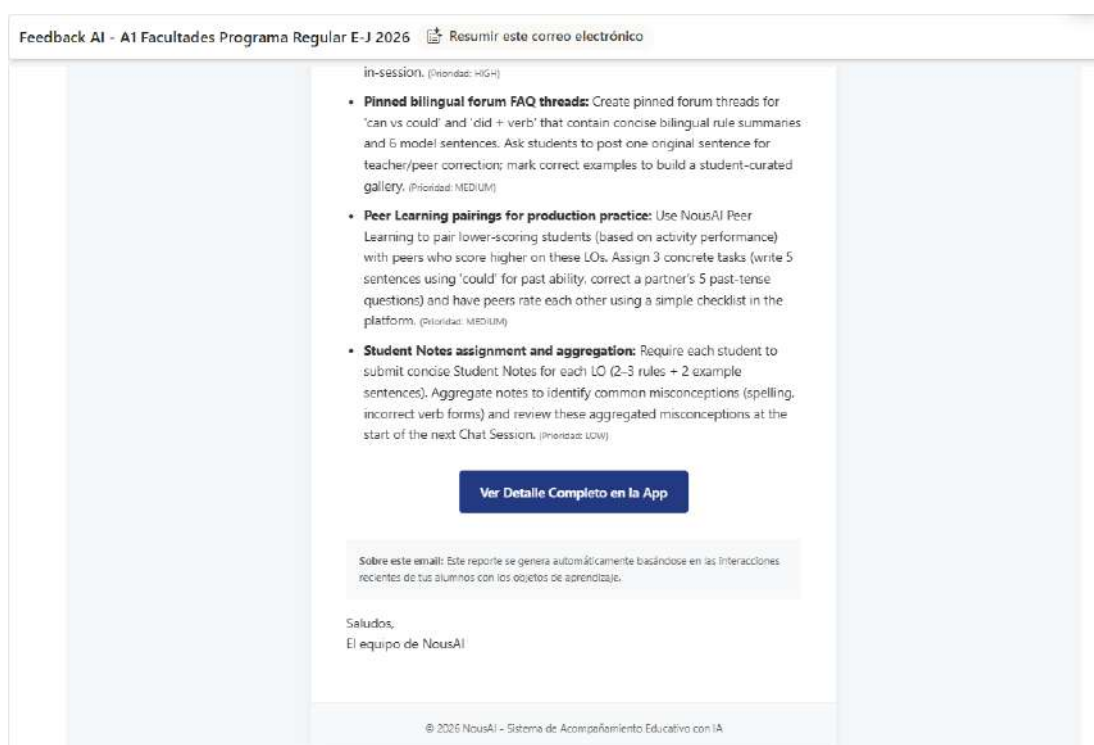


Figura 3.20. Plantilla del correo de retroalimentación al docente (Parte 2).

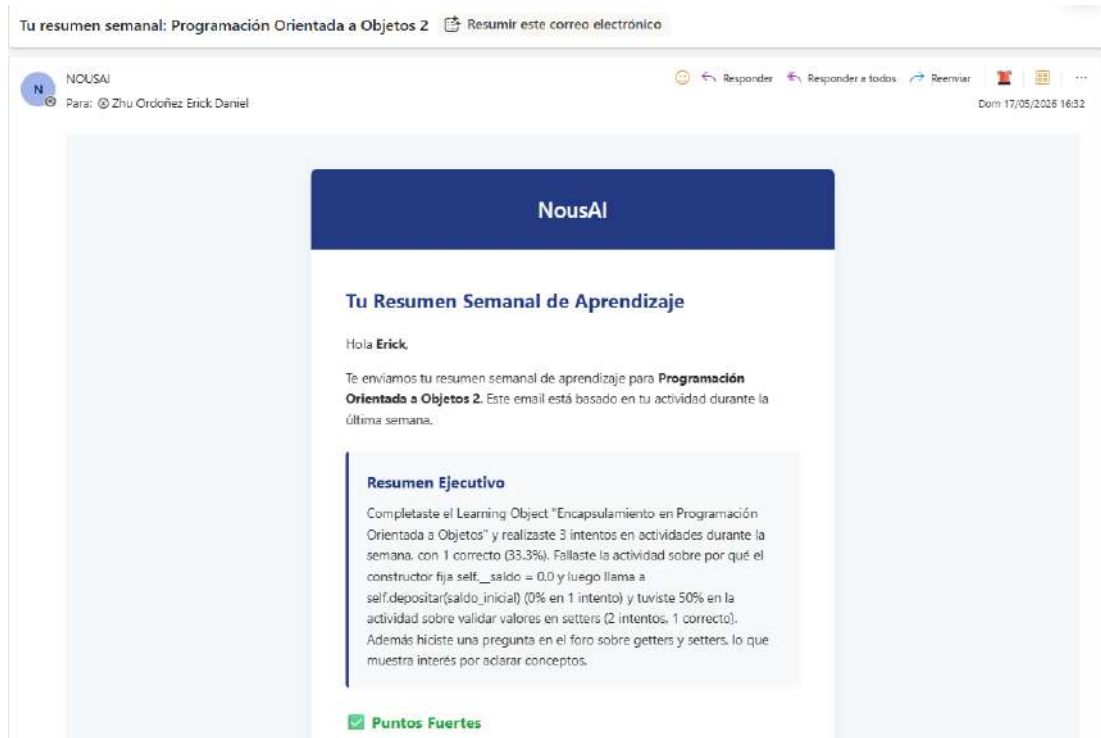


Figura 3.21. Plantilla del correo de retroalimentación al estudiante (Parte 1).

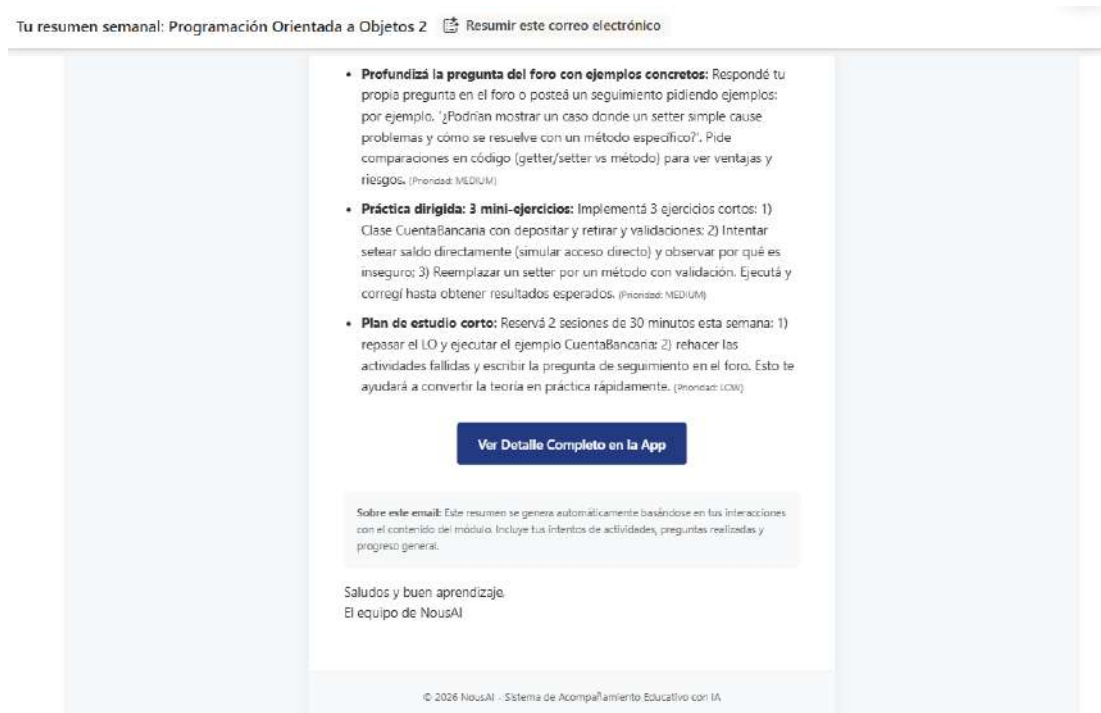


Figura 3.22. Plantilla del correo de retroalimentación al estudiante (Parte 2).

3.1.11 Seguridad, protección de datos y auditoría

La seguridad de la tutoría con IA se aborda en cinco frentes, cada uno con un mecanismo definido: autenticación, control de acceso, validación de entrada, validación de salida del modelo de lenguaje y trazabilidad de las operaciones.

Autenticación

La autenticación es transversal a las tres propuestas de NousAI y se resuelve mediante un esquema en dos capas. La capa externa delega la identificación del usuario en Microsoft Entra ID, el proveedor de identidad de la universidad, a través del protocolo OAuth 2.0. El usuario inicia sesión con sus credenciales institucionales y autoriza a la aplicación a recibir su identidad; NousAI no almacena contraseñas ni gestiona el proceso de inicio de sesión por sí mismo, lo que reduce la superficie de ataque y traslada al proveedor institucional las responsabilidades de cifrado de credenciales, recuperación de cuenta y políticas de complejidad.

La capa interna se resuelve con un token JWT propio que NousAI emite tras la verificación inicial con Entra ID. Este token acompaña a cada solicitud HTTP que el cliente envía al backend, incorporado en la cabecera de autorización, y contiene la información mínima para identificar al usuario y su rol. El token tiene una vigencia acotada, transcurrida la cual el cliente debe renovarlo o iniciar sesión de nuevo. Esta separación permite que los componentes de tutoría operen sin contactar al proveedor de identidad en cada solicitud y, al mismo tiempo, conserva el control sobre la sesión dentro de la propia aplicación.

Control de acceso

La autenticación responde a quién es el usuario; el control de acceso responde a qué puede hacer. El sistema lo resuelve en dos niveles que actúan de forma complementaria.

El primer nivel es el control por rol. Cada usuario tiene asignado un rol —estudiante o docente— y los controladores se protegen con el decorador correspondiente, que rechaza cualquier solicitud cuyo rol no sea el esperado antes de que llegue a la lógica del componente.

El segundo nivel es el control por pertenencia. El rol no basta para autorizar el

acceso, porque un docente no puede consultar las retroalimentaciones de un módulo ajeno y un estudiante no puede acceder a las retroalimentaciones de un módulo en el que no está matriculado. Cada servicio aplica antes de devolver datos una verificación específica: las consultas de retroalimentación al docente exigen que el módulo pertenezca al docente solicitante, las consultas de retroalimentación al estudiante exigen una matrícula activa, y las operaciones sobre una sesión de tutoría verifican que la sesión pertenezca al estudiante y que este conserve acceso de lectura al objeto de aprendizaje. Una solicitud que pase el control por rol pero falle el control por pertenencia recibe una respuesta de prohibición.

Validación de entrada

Toda solicitud que llega al backend se valida antes de ejecutar cualquier operación. La capa de presentación define para cada endpoint un objeto de transferencia de datos con los campos esperados, sus tipos y sus restricciones —longitudes, rangos numéricos, formatos—. La biblioteca `class-validator`, integrada con el pipe global del framework, aplica esas restricciones de forma automática: si la solicitud no las cumple, el sistema responde con un código de error antes de invocar al servicio. Esta validación es independiente de cualquier verificación que se haga en el cliente, en línea con el principio de no confiar en la entrada externa.

Validación de salida del modelo de lenguaje

El modelo de lenguaje devuelve texto libre, de modo que una respuesta malformada podría llegar a persistirse en la base de datos o a presentarse al usuario sin una verificación posterior, con efectos que van desde un error de renderizado hasta la propagación de información incoherente. La biblioteca `Zod` resuelve este punto: para cada prompt que produce salida estructurada se define un esquema que describe la forma esperada de la respuesta, con sus campos obligatorios, sus tipos y sus restricciones. Tras recibir la respuesta del modelo, el servicio que comunica con el proveedor de IA la valida contra el esquema correspondiente; si la respuesta no se ajusta, no se persiste ni se entrega, y el servicio solicita una nueva generación para obtener una salida corregida. Esta capa es el complemento natural del diseño de prompts descrito en el apartado 3.1.9: el prompt acota lo que el modelo debería producir, y la validación contra el esquema comprueba que efectivamente lo produjo.

Protección de los datos del estudiante frente al proveedor de IA

La interacción con el proveedor externo de modelos de lenguaje requiere enviar información para que la respuesta sea contextualizada. Dos decisiones de diseño limitan el alcance de esa exposición.

La primera es la minimización de los datos enviados. El sistema envía al proveedor solo lo necesario para procesar cada solicitud y nada más. El tutor inteligente remite el contenido de la lección y el mensaje del estudiante, pero no el nombre del estudiante ni datos de su perfil. La retroalimentación al docente agrega los datos de interacción por objeto de aprendizaje y por módulo, pero opera siempre sobre patrones y no sobre estudiantes individualmente identificables, en línea con las reglas explícitas del prompt correspondiente.

La segunda es la no persistencia en el proveedor. La política contractual del proveedor empleado establece que la información enviada a través de la interfaz de programación no se utiliza para entrenar modelos ni se retiene más allá del procesamiento de la solicitud. Esto delimita el alcance temporal de la exposición: los datos enviados existen en el proveedor el tiempo que dura el procesamiento de la solicitud y no quedan disponibles para otros fines.

Trazabilidad y auditoría

La trazabilidad permite reconstruir qué ocurrió, cuándo y a instancia de quién. El sistema la sostiene en tres niveles.

A nivel de persistencia, todas las entidades propias de la tutoría —sesiones, mensajes, retroalimentaciones al docente y al estudiante— se almacenan con su marca de tiempo de creación. Las retroalimentaciones registran además el alcance, el módulo de instrucción y el objeto de aprendizaje cuando corresponde, lo que permite ubicar cada salida en su contexto pedagógico. Los mensajes del asistente conservan en su campo de metadatos el identificador de la respuesta del proveedor de IA que los produjo, lo que crea un vínculo entre la salida persistida y la solicitud que la generó en el proveedor.

A nivel de eventos, el sistema deja constancia en los registros de operación de hechos relevantes para la auditoría funcional: la generación de cada retroalimentación —con su resultado, su omisión por umbrales insuficientes o su fallo de validación—, los rechazos del guard de límite de uso en el tutor inteligente, el resultado de cada

envío de correo —enviado de inmediato o diferido por superar el límite diario— y el procesamiento posterior de los correos diferidos. Estos registros no constituyen un sistema de auditoría completo en el sentido normativo, pero proporcionan la información necesaria para investigar incidencias y verificar el comportamiento del sistema en operación.

3.2 Validación de la propuesta

La validación de la propuesta se realizó mediante dos instrumentos complementarios, en correspondencia con los dos actores principales del proceso educativo. A los estudiantes se les aplicó el cuestionario System Usability Scale (SUS) tras un periodo de uso del sistema durante el ciclo académico febrero–julio 2026; a los docentes se les aplicó un cuestionario abierto tras una sesión individual de presentación del sistema, con revisión de retroalimentaciones reales generadas durante ese mismo periodo. La combinación de ambos instrumentos responde al objetivo específico de medir la aceptación de la propuesta tanto desde la experiencia de uso del estudiante como desde la valoración pedagógica del docente.

Los resultados se presentan en tres apartados. El primero recoge los datos cuantitativos obtenidos con estudiantes. El segundo sistematiza las respuestas cualitativas de los docentes. El tercero ofrece una lectura conjunta de ambas fuentes y sitúa la usabilidad medida frente al valor de referencia adoptado en el requisito no funcional RNF-11.

3.2.1 Evaluación de usabilidad con estudiantes mediante el SUS

El cuestionario SUS se aplicó mediante un formulario en línea presentado en los horarios de clase de los cursos que participaron del periodo de prueba. La aplicación se realizó de manera presencial, lo que permitió aclarar dudas de los participantes sobre el instrumento sin condicionar sus respuestas. Cada participante respondió una sola vez. El formulario incluyó un consentimiento informado, preguntas de contexto sobre carrera y semestre, los diez ítems del SUS en la versión española de Sevilla-Gonzalez et al. [20] en su orden estándar, y un espacio opcional de comentarios.

El cómputo del puntaje SUS individual siguió el procedimiento estándar establecido por Brooke [19]: para los ítems impares (afirmaciones positivas) se restó 1 al valor

seleccionado; para los ítems pares (afirmaciones negativas) se restó el valor seleccionado a 5; la suma de los diez valores ajustados se multiplicó por 2,5, lo que entrega un puntaje en una escala de 0 a 100.

Resultados globales.

Durante el periodo de prueba se recogieron 173 respuestas, cifra que supera la muestra prevista de 166 estudiantes. Antes del análisis se aplicó un control de calidad orientado a descartar las respuestas que no reflejan una lectura efectiva del instrumento. Dado que el SUS alterna ítems de polaridad positiva y negativa, una respuesta que asigna el mismo valor a los diez ítems resulta internamente contradictoria. Diecinueve respuestas presentaron este patrón y se excluyeron. El análisis se realizó sobre las 154 respuestas válidas restantes. Tras esta depuración, el número de respuestas analizadas queda ligeramente por debajo de la muestra calculada de 166; por ello, los estadísticos y el intervalo de confianza que se reportan a continuación se calcularon sobre las 154 respuestas efectivas, y el margen de error asociado se interpreta en función de ese tamaño. La Tabla 3.48 presenta los estadísticos descriptivos del puntaje SUS global.

Tabla 3.48. Estadística descriptiva del puntaje SUS global obtenido sobre los estudiantes participantes

Estadístico	Valor
Respuestas válidas (n)	154
Media	71,09
Desviación estándar	17,26
Mediana	70,00
Mínimo	20,00
Máximo	100,00
IC 95 % inferior	68,34
IC 95 % superior	73,84

El puntaje promedio del sistema fue de 71,09 sobre 100, con una desviación es-

tándar de 17,26 puntos. Los puntajes individuales abarcaron el rango completo de la escala, entre 20,00 y 100,00, con una mediana de 70,00. La media se sitúa por encima del valor de referencia de 68 puntos que la literatura asocia al promedio de usabilidad. El intervalo de confianza al 95 % para la media se extiende entre 68,34 y 73,84 puntos; al quedar su límite inferior por encima de la referencia, la usabilidad percibida del sistema supera el promedio documentado con un respaldo estadístico claro y no como producto de la variabilidad de la muestra. La desviación estándar refleja, no obstante, una experiencia heterogénea entre los participantes, lectura que se retoma en el apartado 3.2.3.

La Figura 3.23 presenta la distribución de los puntajes individuales mediante un histograma. La línea vertical en 68 puntos señala el valor de referencia adoptado en el RNF-11.

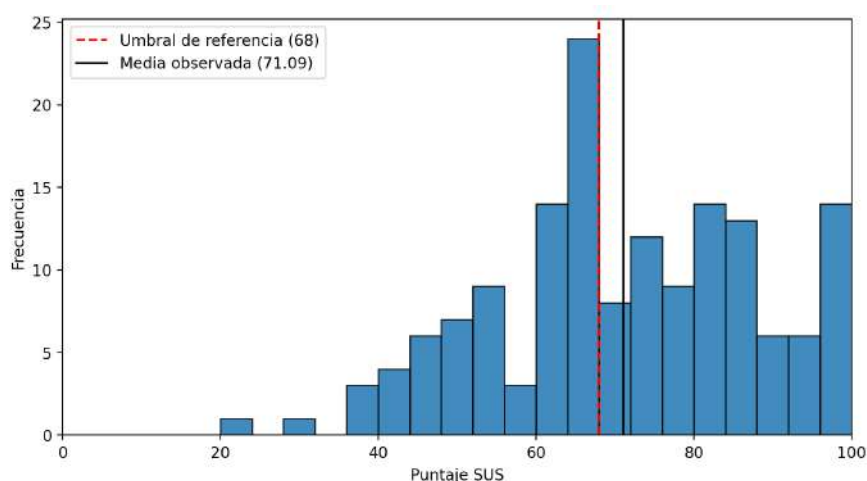


Figura 3.23. Distribución de los puntajes SUS individuales obtenidos sobre los estudiantes participantes.

Resultados por ítem.

Más allá del puntaje global, el análisis ítem por ítem permite identificar qué aspectos de la usabilidad obtuvieron mayor o menor valoración. La Tabla 3.49 presenta la media y la desviación estándar de cada uno de los diez ítems del SUS, expresados en la escala original de 1 a 5.

Tabla 3.49. Promedio y dispersión por ítem del cuestionario SUS

Ítem	Tipo	Texto	Media	DE
Q1	Positivo	Me gustaría usar esta herramienta frecuentemente	3,88	0,90
Q2	Negativo	Considero que esta herramienta es innecesariamente compleja	2,44	1,23
Q3	Positivo	Considero que la herramienta es fácil de usar	4,08	0,96
Q4	Negativo	Considero necesario el apoyo de personal experto para poder utilizar esta herramienta	2,55	1,24
Q5	Positivo	Considero que las funciones de la herramienta están bien integradas	4,05	0,92
Q6	Negativo	Considero que la herramienta presenta muchas contradicciones	2,22	1,16
Q7	Positivo	Imagino que la mayoría de las personas aprenderían a usar esta herramienta rápidamente	4,10	0,98
Q8	Negativo	Considero que el uso de esta herramienta es tedioso	2,12	1,14
Q9	Positivo	Me sentí muy confiado al usar la herramienta	3,97	0,98
Q10	Negativo	Necesité saber bastantes cosas antes de poder empezar a usar esta herramienta	2,31	1,11

En los ítems positivos un valor cercano a 5 indica buena valoración; en los negativos, un valor cercano a 1. La Figura 3.24 representa gráficamente los promedios por ítem, diferenciando por color los ítems positivos (1, 3, 5, 7, 9) de los negativos (2, 4, 6, 8, 10).

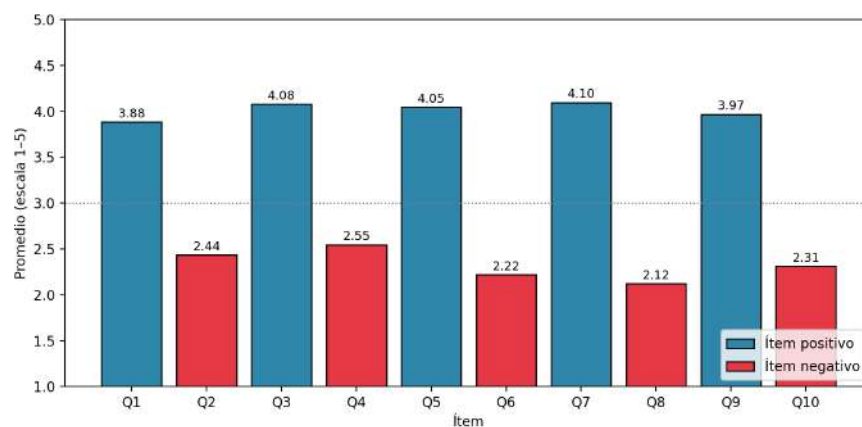


Figura 3.24. Puntaje promedio por ítem del cuestionario SUS.

Los ítems mejor valorados son Q7 (4,10) sobre la rapidez con que otras personas

aprenderían a usar la herramienta, Q3 (4,08) sobre la facilidad de uso y Q5 (4,05) sobre la integración de las funciones, lo que indica que los estudiantes perciben el sistema como abordable y bien estructurado. Los ítems negativos con menor promedio —Q8 (2,12) sobre carácter tedioso y Q6 (2,22) sobre contradicciones— refuerzan esa lectura: el sistema no se percibe como tedioso ni confuso. En el extremo opuesto, los ítems con valoración menos favorable son Q4 (2,55) sobre la necesidad de apoyo experto y Q10 (2,31) sobre los conocimientos previos requeridos para empezar, ambos referidos a la fase inicial de uso. Las observaciones convergen en una misma lectura: aun con una valoración global favorable, la curva de adopción del sistema es el aspecto con mayor margen de mejora, sobre todo en el componente de tutor inteligente, donde la formulación de las preguntas condiciona la calidad de las respuestas. La interpretación de estos resultados se profundiza en el apartado 3.2.3.

Análisis por dimensiones.

Los diez ítems del SUS admiten una descomposición en dos subdimensiones reconocidas por Lewis y Sauro [22]: usabilidad propiamente dicha (ítems 1, 2, 3, 5, 6, 7, 8 y 9; ocho ítems) y aprendibilidad (ítems 4 y 10; dos ítems). Esta separación distingue entre la facilidad general de uso del sistema y la curva de adopción que implica para un usuario nuevo. La Tabla 3.50 presenta los subpuntuajes de ambas dimensiones, escalados a 0–100 para que sean directamente comparables con el puntaje global.

Tabla 3.50. Subpuntuajes por dimensión (Lewis y Sauro, 2009), escala 0–100

Dimensión	Media	DE	IC 95 % inferior	IC 95 % superior
Usabilidad (8 ítems)	72,79	17,46	70,01	75,57
Aprendibilidad (2 ítems)	64,29	25,06	60,30	68,27

La separación por dimensiones muestra un patrón consistente con el análisis ítem por ítem. La subdimensión de Usabilidad obtiene 72,79 puntos, claramente por encima del valor de referencia, mientras que la subdimensión de Aprendibilidad obtiene 64,29 puntos, por debajo de ese valor. La diferencia entre ambas, cercana a nueve puntos, sostiene la lectura de que el sistema se percibe como manejable una vez aprendido, mientras que la curva inicial es la dimensión con mayor margen de mejora. Este contraste orienta las recomendaciones del trabajo: las acciones dirigidas a la adopción

—inducción, materiales de orientación, asistencia inicial al estudiante— son las que mayor impacto pueden tener sobre la usabilidad percibida del conjunto.

Conviene situar estos resultados en las condiciones reales en que se obtuvieron. El sistema se implementó durante el mismo semestre en que se aplicó la validación, de modo que los estudiantes respondieron el cuestionario tras un contacto inicial con la herramienta y no tras un uso sostenido a lo largo del tiempo. En consecuencia, el puntaje obtenido refleja en buena medida la valoración de una experiencia de uso inicial, propia de un primer contacto con la herramienta, y no la de un sistema ya consolidado en la práctica cotidiana. Para asegurar una base común al momento de responder, se presentó la herramienta a los participantes antes de aplicar el cuestionario, dado que el grado de uso efectivo durante el periodo de prueba pudo variar entre ellos. Esta condición es coherente con el menor puntaje de la dimensión de aprendibilidad, asociada precisamente a la fase de entrada al sistema. El alcance de esta limitación y su efecto sobre la interpretación de los resultados se retoman en el apartado 3.2.3, y orientan una de las recomendaciones del Capítulo IV.

3.2.2 Validación cualitativa con docentes

El cuestionario abierto se aplicó a los cinco docentes que participaron del diagnóstico, mediante un formulario en línea administrado tras una sesión individual de presentación del sistema. En cada sesión se presentaron los tres componentes desarrollados —el tutor inteligente, la retroalimentación al docente y la retroalimentación al estudiante— y se revisaron ejemplos reales de retroalimentaciones generadas durante el periodo de prueba sobre los módulos del propio docente cuando formaban parte de los cursos voluntarios, o ejemplos equivalentes en caso contrario. El formulario combinó tres preguntas de contexto —asignatura, número de estudiantes a cargo y familiaridad previa con herramientas de IA generativa— y seis preguntas abiertas correspondientes a las dimensiones de utilidad pedagógica, claridad del contenido, confianza en la información, frecuencia y oportunidad, facilidad de acceso y consulta, y disposición a utilizar el sistema en periodos académicos posteriores.

Las respuestas se procesaron mediante análisis temático. Para cada dimensión se identificaron las coincidencias y los contrastes entre los cinco docentes, y se construyó una síntesis que conserva tanto los puntos de acuerdo como los matices individuales. El

conjunto completo de respuestas se incluye en el Anexo C. La presentación mantiene las referencias anónimas (Docente 1 a Docente 5) y la correspondencia de identidades con el diagnóstico del Capítulo II: el docente que figuró como Docente 1 en aquel capítulo es el mismo Docente 1 en este apartado.

Caracterización del grupo.

Los cinco docentes imparten asignaturas en la carrera de Software durante el ciclo académico febrero–julio 2026, en áreas que abarcan desde la programación y las estructuras de datos hasta la gestión de software, los sistemas operativos y asignaturas de corte transversal. El número de estudiantes a cargo es heterogéneo, desde grupos cercanos a la treintena hasta cargas que superan el centenar, lo que resulta relevante para interpretar las observaciones sobre frecuencia y volumen de información. En cuanto a la familiaridad previa con IA generativa, dos docentes declaran haberla usado con fines docentes, dos haberla explorado de forma personal y uno reporta familiaridad nula o muy escasa. Esta diversidad permite contrastar la valoración del sistema entre perfiles con distinto grado de exposición previa a la tecnología.

Utilidad pedagógica. Los cinco docentes reconocen valor a la información que entrega el sistema, aunque desde énfasis distintos. La lectura más extendida es que la retroalimentación ofrece una visión consolidada del desempeño del grupo que normalmente no llega de forma sistemática, y que reduce el tiempo que el docente dedicaría a reconstruir esa información por sus propios medios. Varios subrayan que el valor depende de condiciones de uso: uno lo vincula a la calidad del contenido generado y, por extensión, a la formulación de las instrucciones que recibe el modelo; otro condiciona la utilidad a cuánto lo usen efectivamente los estudiantes y al esfuerzo de actualización que demande al docente. Uno de los docentes con grupos más numerosos valora especialmente que la información se presente de forma agrupada y transparente. Aparece también una observación que excede la utilidad inmediata y se orienta a la viabilidad: la conveniencia de incorporar métricas de auditoría, costos y consumo de recursos para evaluar la sostenibilidad del sistema a gran escala. Un docente aporta un matiz de fondo sobre el contexto en que esta utilidad cobra sentido: señala que la práctica habitual carece de un trabajo metodológico en equipo que retroalimente la ejecución de los currículos, de modo que una herramienta que devuelva información sobre el proceso

podría cubrir un vacío real.

Claridad del contenido. La forma en que el sistema organiza y presenta la información recibe una valoración mayoritariamente favorable: los docentes la describen como clara, comprensible y útil para interactuar con los estudiantes. Las observaciones de mejora apuntan a dos aspectos. El primero es la posibilidad de personalizar o configurar la presentación según las preferencias de cada docente. El segundo, más sustantivo, proviene del docente que advierte que la información a veces requiere interpretación y que, en ocasiones, las sugerencias del sistema entran en contradicción con el funcionamiento real de la plataforma; el ejemplo que ofrece —una recomendación de habilitar un canal de conversación entre docente y estudiante, cuando la conversación está implementada entre la IA y el estudiante— ilustra que el docente debe mantener una actitud crítica frente a lo que el sistema sugiere. Ese mismo docente considera, no obstante, que las secciones presentes son las mínimas necesarias y que su orden es adecuado, y plantea como posible añadido una sección de estadísticas. Otro docente observa que conviene aclarar términos que las respuestas generadas introducen y que con frecuencia se pasan por alto.

Confianza en la información. Es la dimensión donde se concentran las reservas, de manera consistente con la postura recogida en el diagnóstico. Ninguno de los docentes considera que la información del sistema deba tomarse como concluyente sin verificación. Las posiciones, sin embargo, no son homogéneas. Algunos la describen como un insumo de apoyo, útil para identificar tendencias generales, estudiantes en riesgo o necesidades de refuerzo, pero que requiere el juicio del profesor antes de fundamentar decisiones académicas, en particular porque parte de la valoración del aprendizaje ocurre de forma presencial. Otros señalan el riesgo de sesgos en la información generada. Una postura más matizada sostiene que el valor del sistema no reside necesariamente en la fiabilidad literal de lo que afirma, sino en su capacidad de interpelar al docente, dinamizar la enseñanza e incentivar la mejora continua. En el extremo más favorable, un docente considera la información confiable en la medida en que se apoya en un análisis detallado de los casos registrados en la plataforma. El conjunto dibuja una preocupación compartida —la veracidad de lo que produce el modelo— atravesada por lecturas distintas sobre qué se espera de esa información.

Frecuencia y oportunidad. La periodicidad semanal se valora de forma desigual, y de las respuestas emerge una demanda convergente: la posibilidad de parametrizar la entrega. Un docente la considera directamente adecuada para tomar decisiones académicas oportunas sobre el grupo. Los demás introducen condiciones. Uno propone que el envío se ajuste a horarios laborales o a un momento configurable, de modo que el docente lo reciba dentro de su jornada. Otro advierte que en cursos con gran número de estudiantes una entrega semanal fija puede generar sobrecarga de revisión y presión administrativa, por lo que debería poder ajustarse. Un tercero plantea que la utilidad de la frecuencia depende de que el sistema sea capaz de detectar cuándo no ha habido cambios y limitarse a avisar de ello, ya que de lo contrario el valor de la retroalimentación podría diluirse. También se señala que la retroalimentación, tal como se percibió, resulta general más que específica.

Facilidad de acceso y consulta. El acceso a la información, por correo y por plataforma, se describe como sencillo y fluido. Un docente destaca que la implementación le resultó cómoda y bien resuelta. Las observaciones se orientan a precisar el canal más conveniente: hay quien considera preferible consultar el detalle únicamente en la plataforma y recibir por correo solo un resumen, para evitar la dispersión de la información. En la misma línea, se menciona que, en contextos con gran cantidad de estudiantes, cursos y actividades, conviene prevenir una eventual sobrecarga de notificaciones que dificulte revisar la información de forma efectiva. Las dos lecturas coinciden en un punto: el acceso funciona, y el margen de mejora está en dosificar el volumen y elegir bien el canal de cada tipo de contenido.

Disposición a utilizar el sistema. Los cinco docentes manifiestan disposición a utilizar el sistema en periodos posteriores, en todos los casos como herramienta de apoyo a la práctica docente y no como sustituto del criterio profesional. Dos de ellos acompañan esa disposición con propuestas de extensión: la generación automática de evaluaciones y rúbricas a partir de lo trabajado en clase, y la producción de actividades o evaluaciones adaptativas según el avance de cada estudiante. Uno la vincula especialmente a asignaturas de niveles iniciales, donde los estudiantes carecen de conocimientos previos y requieren acompañamiento más frecuente. Otro condiciona su disposición a la cantidad de retroalimentaciones que llegue a recibir, y se abstiene de

formular recomendaciones mientras esa experiencia de uso sea limitada. La disposición favorable, en conjunto, aparece ligada a que el sistema continúe disponible más allá del periodo de prueba y a que incorpore las mejoras que los propios docentes señalan.

Hallazgos transversales.

Del conjunto de respuestas se desprenden tres lecturas que atraviesan las dimensiones.

La primera es que las reservas se concentran en la confianza en la información, no en la utilidad ni en la facilidad de uso. Los cinco docentes reconocen valor pedagógico al sistema y describen su acceso como comprensible, pero coinciden en que la información generada exige verificación y en que la decisión académica debe seguir recayendo en el docente. Esta concentración de la cautela en un único punto coincide con lo observado en el diagnóstico del Capítulo II, donde los riesgos identificados ya giraban en torno a la calidad y la veracidad de lo que produce el modelo. La consistencia entre ambas fases sugiere que la preocupación no es una reacción puntual, sino una postura estable frente a la incorporación de IA generativa en la enseñanza.

La segunda es que la valoración del sistema no depende del grado de familiaridad previa con la tecnología. Docentes con experiencia usando IA con fines docentes, otros con exploración personal y uno sin familiaridad apreciable reconocen un valor semejante y plantean reservas equivalentes. Lo que diferencia las posturas no es el dominio técnico, sino la posición pedagógica de cada docente respecto al papel que la IA debe ocupar en el aula, un debate que el Capítulo I situó como vigente en la literatura del área.

La tercera es que las condiciones que los docentes asocian a un uso sostenido se ubican en el plano de mejoras incrementales y no cuestionan la arquitectura ni el alcance de la propuesta. La parametrización de la frecuencia, la elección del canal por tipo de contenido, la trazabilidad entre lo que el sistema afirma y los registros que lo sostienen, las opciones de configuración y las extensiones hacia evaluaciones o rúbricas son todas adiciones sobre lo desarrollado. Un punto merece atención particular por su alcance: la observación sobre la seguridad, la privacidad y la gobernanza de los datos académicos, que apunta a un requisito institucional más amplio que el sistema deberá

considerar si se proyecta su uso sobre poblaciones grandes.

3.2.3 Discusión integrada de resultados

La validación combinó dos fuentes con propósitos distintos: el SUS midió la usabilidad percibida por los estudiantes que usaron el sistema, y el cuestionario abierto recogió la valoración de los docentes tras conocer la propuesta en detalle. Leerlas juntas permite contrastar lo que cada actor identifica como fortaleza y como reserva, y situar el puntaje obtenido frente al referente de usabilidad adoptado en el RNF-11.

El SUS arrojó una media de 71,09 puntos sobre las 154 respuestas válidas, con un intervalo de confianza al 95 % entre 68,34 y 73,84. Como el límite inferior del intervalo queda por encima del valor de referencia de 68, puede afirmarse que la usabilidad percibida supera el promedio documentado con respaldo estadístico. La descomposición por dimensiones precisa el resultado: la usabilidad propiamente dicha alcanza 72,79 puntos, mientras que la aprendibilidad se queda en 64,29. Esa diferencia, cercana a nueve puntos, señala con claridad dónde está el margen de mejora —en la entrada al sistema, no en su manejo una vez aprendido—, lectura que el análisis por ítem confirma: los valores más favorables corresponden a la facilidad de uso y a la rapidez con que otros aprenderían a usar la herramienta (Q3 y Q7), y los más cautos, a la necesidad de apoyo y de conocimientos previos para empezar (Q4 y Q10).

Conviene situar este puntaje en las condiciones reales de la medición. El sistema se implementó durante el mismo semestre en que se aplicó la validación, de modo que los estudiantes respondieron tras un contacto inicial con la herramienta y no tras un uso sostenido en el tiempo. El resultado refleja, por tanto, la valoración de una experiencia de uso inicial, y no la de un sistema ya asentado en la práctica. Esto es coherente con que la aprendibilidad sea la dimensión más baja, y matiza la lectura del SUS sin restarle validez: lo que se mide es una primera adopción, en la que la curva de entrada pesa de forma esperable.

Dos hallazgos surgen de cruzar ambas fuentes. El primero es que la curva de adopción inicial es el punto sobre el que estudiantes y docentes coinciden, desde lugares distintos. En los estudiantes aparece como la aprendibilidad rezagada; en los docentes, como la solicitud reiterada de materiales de orientación para ellos y para los estudiantes, y de un acompañamiento más frecuente en los niveles iniciales. Que la misma

necesidad emerja de actores que interactúan con componentes distintos del sistema indica que invertir en inducción y materiales de apoyo tendría un efecto directo sobre la usabilidad percibida del conjunto.

El segundo hallazgo es que la confianza en la información generada es la reserva central de los docentes, y que el SUS no la capta. Su único ítem sobre confianza (Q9) se refiere a la seguridad al usar la interfaz, no a la fiabilidad de los contenidos que el modelo produce. Los cinco docentes, en cambio, coinciden en que esa información exige verificación y en que la decisión académica debe seguir recayendo en el profesor; uno de ellos ilustró el punto al detectar que el sistema sugería habilitar un canal de conversación entre docente y estudiante cuando esa conversación está implementada entre la IA y el estudiante, una incongruencia que muestra por qué el criterio humano sigue siendo necesario. Esta brecha entre lo que el SUS mide y lo que los docentes priorizan no es una falla de la validación, sino la razón por la que se emplearon dos instrumentos: cada uno capta una dimensión de la aceptación que el otro no alcanza. Y conecta de manera directa con el carácter supervisado de la propuesta, en la que la última palabra sobre lo que se publica y sobre cómo se interpreta la retroalimentación corresponde al docente.

En conjunto, la aceptación de la propuesta puede describirse como favorable y condicionada. El objetivo específico tercero buscaba medir el nivel de aceptación de la solución por parte de estudiantes y docentes de la carrera de Software. El SUS, con una media que supera el referente de usabilidad, y el cuestionario docente, con disposición unánime a utilizar el sistema en periodos posteriores, sostienen el cumplimiento de ese objetivo. La aceptación no es incondicional: los docentes la vinculan a mejoras incrementales —parametrizar la frecuencia de las entregas, elegir el canal según el tipo de contenido, reforzar la trazabilidad entre lo que el sistema afirma y los registros que lo sostienen— y a una preocupación de mayor alcance sobre la seguridad y la gobernanza de los datos académicos cuando el uso se proyecte sobre poblaciones grandes. Ninguna de esas condiciones cuestiona la arquitectura ni el alcance de la propuesta; todas se ubican en el plano de la evolución posterior.

Esta validación tiene límites que conviene reconocer. El principal ya se señaló: la medición se realizó sobre una implementación reciente, por lo que captura una adopción temprana y no el uso consolidado que se obtendría evaluando tras un ciclo com-

pleto de funcionamiento; lo idóneo habría sido implementar en un semestre y validar en el siguiente. A ello se suma que la incorporación de cursos fue voluntaria, de modo que los estudiantes participantes provienen de los grupos cuyos docentes aceptaron usar el sistema, lo que aconseja leer los resultados como evidencia situada y no como medición sobre la totalidad de la población. El SUS mide la usabilidad percibida del sistema como producto, no su efecto sobre el aprendizaje ni sobre la práctica docente; determinar si el sistema mejora los resultados educativos requiere una evaluación posterior. A esto se añaden dos aspectos que el alcance de este trabajo no abordó de forma sistemática. Por un lado, la validación se centró en la usabilidad y en la percepción docente, pero no evaluó de manera formal la calidad intrínseca de las respuestas generadas por la IA —su exactitud, su pertinencia pedagógica, su alineación con el contenido fuente y la ausencia de afirmaciones no respaldadas—; esa valoración exigiría un instrumento específico, como una rúbrica aplicada a una muestra de respuestas. Por otro, la verificación del software se realizó de manera incremental durante el desarrollo, sin una batería formal de pruebas funcionales que relacione cada requisito y caso de uso con casos de prueba y resultados documentados. Ambos aspectos se retoman como líneas de trabajo futuro en el Capítulo IV

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

El desarrollo del presente trabajo permite extraer cuatro conclusiones que articulan los hallazgos obtenidos en sus distintas fases, en correspondencia con los objetivos específicos planteados en la sección 1.4.2 y con el objetivo general que orientó la propuesta. En lo que sigue, la automatización de la tutoría debe entenderse en su sentido preciso: el sistema asiste y media en el acompañamiento al estudiante y mantiene al docente como responsable de las decisiones académicas, sin sustituir su labor.

La identificación de tecnologías para automatizar la tutoría docente con IA generativa no se redujo a una elección de proveedor de modelos de lenguaje, sino que requirió decidir sobre tres bloques estructurales que condicionan el comportamiento del sistema: el almacenamiento de información estructurada y vectorial, el procesamiento asíncrono de tareas y la validación automática de las salidas del modelo. La revisión del estado del arte mostró que los sistemas educativos basados en IA generativa que reportan resultados favorables comparten estas tres decisiones técnicas, no por coincidencia sino porque son las que permiten ajustar las respuestas del modelo al contenido del curso, sostener tareas prolongadas sin afectar la interacción del usuario y verificar la forma de las salidas antes de presentarlas. La elección de PostgreSQL con la extensión pgvector, de BullMQ sobre Redis para la gestión de colas y de Zod para la validación estructural de las respuestas del modelo se sostiene sobre esa lectura del estado del arte y constituye, en sí misma, una contribución reutilizable para propuestas posteriores que aborden problemas similares en el contexto universitario ecuatoriano. Estas decisiones son las que sostienen la diferencia entre un sistema de tutoría supervisado y una integración directa con un modelo de lenguaje, distinción que la literatura del campo ha identificado como central para el uso responsable de la IA en educación.

El sistema implementado articula tres componentes complementarios —tutor inteligente, retroalimentación al docente y retroalimentación al estudiante— sobre una misma base de datos compartida con las otras propuestas del proyecto NousAI. La implementación demuestra que el carácter supervisado puede sostenerse a nivel arquitectónico, no solo declarativo: por un lado, mediante la decisión docente sobre qué

objetos de aprendizaje se publican y, por extensión, sobre qué material puede ser utilizado por el sistema como contexto de las respuestas; por otro, mediante mecanismos estructurales que aseguran que el sistema opere sobre material aprobado y produzca salidas verificables. La validación de entrada con class-validator, la verificación de las respuestas del modelo contra esquemas Zod, la persistencia de cada operación con identificadores trazables hacia el proveedor de IA y los controles de acceso por rol y por pertenencia conforman un conjunto coherente que sitúa al docente como referente último de lo que el sistema entrega. La integración con las otras dos propuestas del proyecto NousAI —la gestión de wiki basada en IA generativa y la producción de objetos de aprendizaje a partir de videos educativos— se resuelve sin acoplamiento técnico adicional, mediante el consumo del contenido y los registros que esas propuestas producen como insumos para los componentes desarrollados en este trabajo.

La aceptación de la propuesta se midió mediante dos instrumentos complementarios aplicados a los dos actores principales del proceso educativo. El cuestionario SUS, aplicado a los estudiantes que utilizaron el sistema durante el periodo de prueba, arrojó un puntaje promedio de 71,09 puntos sobre 100 a partir de 154 respuestas válidas, con un intervalo de confianza al 95 % entre 68,34 y 73,84. Al situarse el límite inferior de ese intervalo por encima del valor de referencia de 68, la usabilidad percibida supera el promedio documentado en la literatura con respaldo estadístico. La descomposición del puntaje en sus dos subdimensiones reveló un patrón consistente: la usabilidad propiamente dicha alcanzó 72,79 puntos y la aprendibilidad 64,29, lo que identifica la curva de adopción inicial como el aspecto con mayor margen de mejora. El cuestionario abierto aplicado a los cinco docentes tras una sesión individual de presentación del sistema mostró disposición unánime a continuar usándolo, junto con una reserva compartida que se concentra en la confianza sobre la información que el modelo genera y la necesidad de verificarla antes de fundamentar decisiones académicas. La lectura conjunta de ambas fuentes sostiene el cumplimiento del objetivo de medir la aceptación y, al mismo tiempo, orienta dónde concentrar las mejoras posteriores. Debe considerarse, no obstante, que la validación se realizó sobre una implementación reciente, por lo que el puntaje refleja una experiencia de uso inicial más que el uso consolidado que se obtendría tras un ciclo completo de funcionamiento.

La automatización supervisada de la tutoría docente en cursos universitarios me-

diente IA generativa es viable bajo condiciones específicas de diseño. Esta automatización es parcial y asistida: el sistema se encarga de tareas de apoyo —responder consultas acotadas a la lección, consolidar la actividad del grupo y generar retroalimentación—, mientras la validación del contenido y las decisiones pedagógicas permanecen en manos del docente. El trabajo muestra que la viabilidad no se sostiene solo en la capacidad técnica del modelo de lenguaje, sino, sobre todo, en la arquitectura que rodea su uso: alineación de las respuestas con el contenido aprobado por el docente, validación estructural de las salidas, trazabilidad de las operaciones y entrega de información agregada al docente como insumo de decisión. Esta lectura coincide con la postura predominante en la literatura del campo, que distingue entre el uso libre de los modelos de lenguaje —asociado a riesgos pedagógicos y de calidad— y el uso supervisado e integrado en el contexto del curso —asociado a beneficios observables sobre el proceso educativo—. La aceptación obtenida en la validación, cuantitativa por encima del referente de usabilidad y cualitativa con condiciones explícitas formuladas por los docentes, sitúa a la propuesta como una primera implementación válida sobre la que pueden construirse mejoras incrementales orientadas por la evidencia recogida. El trabajo entrega una herramienta funcional y en operación, validada con usuarios reales, que responde a una necesidad concreta del proceso de enseñanza-aprendizaje: ofrece al estudiante acompañamiento alineado con el contenido de cada lección y devuelve al docente una visión consolidada del avance de su grupo que hoy no recibe por otros medios. Su valor práctico se refuerza en que el sistema expone una API REST mediante la cual se integró a NousAI. Sobre esta base, la práctica docente dispone de un punto de partida sólido, y las mejoras incrementales que la evidencia sugiere pueden construirse sin rehacer lo desarrollado.

4.2 Recomendaciones

Las recomendaciones siguientes se derivan de los resultados de la validación, de las observaciones recogidas durante el cuestionario docente y la aplicación del SUS, y de los límites identificados en el alcance del trabajo. Cada una se dirige a una línea de acción específica.

1. Acompañar el despliegue con materiales de orientación e inducción inicial.

La curva de adopción inicial fue el aspecto más sensible según ambas fuentes: en

los estudiantes se reflejó en el subpuntaje de aprendibilidad (64,29, por debajo del referente de 68) y en los ítems sobre necesidad de apoyo y conocimientos previos para empezar; en los docentes, en la observación de que el sistema requiere interpretación al inicio y en la propuesta de un acompañamiento más frecuente para los estudiantes de niveles iniciales. Una guía breve para el estudiante, un documento de referencia para el docente y una inducción al comienzo de cada ciclo tienen un costo bajo frente al impacto que pueden tener sobre la usabilidad percibida, sobre todo considerando que la medición se realizó sobre un primer contacto con la herramienta.

2. **Exponer en cada retroalimentación los registros que la sustentan.** La reserva sobre la confianza en la información generada fue la preocupación transversal del cuerpo docente: los cinco coincidieron en que esa información exige verificación y en que la decisión final corresponde al profesor, y uno de ellos detectó una sugerencia del sistema que contradecía el funcionamiento real de la plataforma. Mostrar como respaldo navegable los registros concretos que respaldan cada afirmación —interacciones del tutor inteligente, actividades resueltas, consultas planteadas— responde directamente a esa reserva y refuerza el carácter supervisado de la propuesta. La mejora no exige rediseñar la arquitectura: los registros ya existen en la base de datos y se usan internamente para construir cada retroalimentación; se trata de exponerlos como respaldo consultable.
3. **Incorporar opciones de configuración sobre la entrega de las retroalimentaciones.** La periodicidad semanal recibió una valoración desigual: algunos docentes la consideraron adecuada, mientras que otros pidieron ajustarla al horario laboral, espaciarla cuando no haya cambios relevantes en el grupo o parametrizarla para evitar la sobrecarga en cursos numerosos. En cuanto al canal, se planteó la conveniencia de recibir por correo un resumen y reservar la plataforma para el detalle, de modo que el volumen de notificaciones no dificulte la revisión. Permitir que el docente configure cuándo y cómo recibe la información atiende estas observaciones sin alterar la lógica del sistema.
4. **Realizar una segunda etapa de validación tras un periodo de uso sostenido.** Idealmente, implementando el sistema en un ciclo académico y evaluándolo en

el siguiente. La presente validación capturó una adopción inicial, ya que la implementación coincidió con el semestre de evaluación; queda pendiente observar cómo evoluciona la valoración —de estudiantes y docentes— cuando el sistema se integra a la rutina a lo largo de varios meses. Esa evaluación posterior debería combinar la medición cuantitativa de usabilidad con preguntas cualitativas sobre el ajuste del sistema a la práctica docente, dos aspectos que solo un uso prolongado permite observar.

5. **Diseñar una línea de evaluación orientada al efecto sobre los resultados de aprendizaje.** Como complemento a la usabilidad medida en este trabajo, el SUS responde a si el sistema se percibe usable, pero no a si modifica el desempeño del estudiante. Un diseño que compare cursos que usan el sistema con cursos que no lo usan, a lo largo de un ciclo completo, permitiría aproximarse a esa pregunta mediante indicadores como el desempeño en evaluaciones, la tasa de aprobación o la cantidad de consultas que el estudiante formula sobre el contenido. Los trabajos de Wan y Chen, Alalawi et al. y Villegas-Ch et al., revisados en el Capítulo I, ofrecen referencias metodológicas adaptables a las condiciones de la carrera de Software en la FISEI-UTA.
6. **Atender los aspectos de seguridad, privacidad y gobernanza de los datos académicos de cara a un uso sobre poblaciones grandes.** Uno de los docentes señaló la necesidad de conocer cómo se almacenan los datos, qué mecanismos de protección existen y cómo se garantiza el cumplimiento de la normativa institucional, una preocupación que cobra peso al tratarse de información de un gran número de estudiantes. Este punto adquiere especial relevancia por el envío de datos de la actividad estudiantil a un proveedor externo de modelos de lenguaje, y debe enmarcarse en la Ley Orgánica de Protección de Datos Personales del Ecuador, así como en las políticas de tratamiento de datos del propio proveedor. Atender estas exigencias excede el desarrollo técnico del trabajo y requiere articularse con las políticas de datos de la institución —incluida la formalización de los acuerdos de tratamiento que correspondan—, pero conviene incorporarlo como condición para escalar el uso del sistema más allá del periodo de prueba.
7. **Evaluar de forma sistemática la calidad de las respuestas generadas por la**

IA. La validación de este trabajo se centró en la usabilidad y en la percepción docente estudiante, pero no midió de manera formal la calidad intrínseca de lo que el modelo produce. Un trabajo posterior debería diseñar una rúbrica que evalúe, sobre una muestra de respuestas del tutor inteligente y de las retroalimentaciones, dimensiones como la exactitud, la alineación con el contenido del objeto de aprendizaje, la ausencia de afirmaciones no respaldadas, la pertinencia pedagógica y la trazabilidad con la fuente. Esa evaluación complementaría la aceptación ya medida con una valoración directa de la fiabilidad del contenido, que es la principal reserva expresada por los docentes.

8. **Formalizar la verificación del software mediante una matriz de pruebas funcionales.** Durante el desarrollo, las funcionalidades se verificaron de manera incremental a medida que se implementaban, sin una batería de pruebas documentada de forma sistemática. Para reforzar la trazabilidad y facilitar el mantenimiento, conviene construir una matriz que relacione cada requisito funcional con su caso de uso, su endpoint, los casos de prueba asociados, el resultado esperado y el resultado obtenido. La estructura de requisitos, casos de uso y endpoints documentada en este trabajo ofrece la base sobre la que esa matriz puede elaborarse.

BIBLIOGRAFÍA

- [1] N. Hernandez-Leon and M.-J. Rodriguez-Conde, “Artificial intelligence applied to education and educational assessment at the university: introduction of intelligent tutoring systems, recognition systems, and other future trends.,” *RED-REVISTA DE EDUCACION A DISTANCIA*, vol. 24, 5 2024.
- [2] E. Kasneci, K. Sessler, S. Kuechemann, M. Bannert, D. Dementieva, F. Fischer, G. G. Urs, Groh, S. Guennemann, E. Huellermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, A. S. Michael, Schmidt, T. Seidel, M. Stadler, J. Weller, J. Kuhn, and G. Kasneci, “Chatgpt for good? on opportunities and challenges of large language models for education,” *LEARNING AND INDIVIDUAL DIFFERENCES*, vol. 103, 4 2023.
- [3] T. Wan and Z. Chen, “Exploring generative ai assisted feedback writing for students’ written responses to a physics conceptual question with prompt engineering and few-shot learning,” *PHYSICAL REVIEW PHYSICS EDUCATION RESEARCH*, vol. 20, 6 2024.
- [4] G. Jost, V. Taneski, and S. Karakatic, “The impact of large language models on programming education and student learning outcomes,” *APPLIED SCIENCES-BASEL*, vol. 14, 5 2024.
- [5] W. Villegas-Ch, J. Govea, R. Albuja, D. Buenano-Fernandez, and A. Mera-Navarrete, “Optimizing automated educational systems: Adaptive syllabus generation and precise assessment using artificial intelligence,” *IEEE ACCESS*, vol. 12, pp. 173390–173409, 2024.
- [6] K. Alalawi, R. Athauda, R. Chiong, and I. Renner, “Evaluating the student performance prediction and action framework through a learning analytics intervention study,” *EDUCATION AND INFORMATION TECHNOLOGIES*, vol. 30, pp. 2887–2916, 2 2025.

- [7] F. Kamalov, D. S. Calonge, and I. Gurrib, “New era of artificial intelligence in education: Towards a sustainable multifaceted revolution,” *SUSTAINABILITY*, vol. 15, 8 2023.
- [8] Y.-F. Lee, G.-J. Hwang, and P.-Y. Chen, “Impacts of an ai-based chabot on college students’ after-class review, academic performance, self-efficacy, learning attitude, and motivation,” *ETR&D-EDUCATIONAL TECHNOLOGY RESEARCH AND DEVELOPMENT*, vol. 70, pp. 1843–1865, 10 2022.
- [9] M. A. Kuhail, N. Alturki, S. Alramlawi, and K. Alhejori, “Interacting with educational chatbots: A systematic review,” *EDUCATION AND INFORMATION TECHNOLOGIES*, vol. 28, pp. 973–1018, 1 2023.
- [10] E. Rudolph, H. Seer, C. Mothes, and J. Albrecht, “Automated feedback generation in an intelligent tutoring system for counselor education,” in *2024 19TH CONFERENCE ON COMPUTER SCIENCE AND INTELLIGENCE SYSTEMS, FEDCSIS 2024* (M. Bolanowski, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Slezak, eds.), pp. 501–512, IEEE, 2024. 19th Conference on Computer Science and Intelligence Systems, Belgrade, SERBIA, SEP 08-11, 2024.
- [11] S.-H. Jin, K. Im, M. Yoo, I. Roll, and K. Seo, “Supporting students’ self-regulated learning in online learning using artificial intelligence applications,” *INTERNATIONAL JOURNAL OF EDUCATIONAL TECHNOLOGY IN HIGHER EDUCATION*, vol. 20, 6 2023.
- [12] F. Ouyang, M. Wu, L. Zheng, L. Zhang, and P. Jiao, “Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course,” *INTERNATIONAL JOURNAL OF EDUCATIONAL TECHNOLOGY IN HIGHER EDUCATION*, vol. 20, 1 2023.
- [13] E. Mousavinasab, N. Zarifsanaiey, S. R. N. Kalhori, M. Rakhshan, L. Keikha, and M. G. Saeedi, “Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods,” *INTERACTIVE LEARNING ENVIRONMENTS*, vol. 29, pp. 142–163, 1 2021.

- [14] J. Su and W. Yang, “Unlocking the power of chatgpt: A framework for applying generative ai in education,” *ECNU REVIEW OF EDUCATION*, vol. 6, pp. 355–366, 8 2023.
- [15] J. Weidlich, A. Fink, A. Frey, I. Jivet, S. Gombert, L. Menzel, T. Giorgashvili, J. Yau, and H. Drachsler, “Highly informative feedback using learning analytics: how feedback literacy moderates student perceptions of feedback,” *INTERNATIONAL JOURNAL OF EDUCATIONAL TECHNOLOGY IN HIGHER EDUCATION*, vol. 22, 7 2025.
- [16] D. H. Chang, M. P.-C. Lin, S. Hajian, and Q. Q. Wang, “Educational design principles of using ai chatbot that supports self-regulated learning in education: Goal setting, feedback, and personalization,” *SUSTAINABILITY*, vol. 15, 9 2023.
- [17] K. A. Fuller, K. A. Morbitzer, A. M. Z. J. M., Persky, A. C. Savage, and J. E. McLaughlin, “Exploring the use of chatgpt to analyze student course evaluation comments,” *BMC MEDICAL EDUCATION*, vol. 24, 4 2024.
- [18] C. D. Kloos, C. Alario-Hoyos, P. E.-A. Iria, Callejo-Pinardo, M. A. Hombrados-Herrera, P. J. Munoz-Merino, P. M. Moreno-Marcos, M. Munoz-Organero, and M. B. Ibanez, “How can generative ai support education?,” in *2024 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE, EDUCON 2024*, IEEE, 2024. 15th IEEE Global Engineering Education Conference (IEEE EDUCON), GREECE, MAY 08-11, 2024.
- [19] J. Brooke, “SUS: A quick and dirty usability scale,” *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4–7, 1996.
- [20] M. d. R. Sevilla-Gonzalez, L. Moreno Loaeza, L. S. Lazaro-Carrera, B. Bourguet Ramirez, A. Vázquez Rodríguez, M. L. Peralta-Pedrero, and P. Almeda-Valdes, “Spanish version of the system usability scale for the assessment of electronic tools: Development and validation,” *JMIR Human Factors*, vol. 7, no. 4, p. e21161, 2020.
- [21] P. L. López, “Población muestra y muestreo,” *Punto Cero*, vol. 9, no. 8, pp. 69–74, 2004.

- [22] J. R. Lewis and J. Sauro, “The factor structure of the system usability scale,” in *Human Centered Design* (M. Kurosu, ed.), vol. 5619 of *Lecture Notes in Computer Science*, (Berlin, Heidelberg), pp. 94–103, Springer, 2009.

ANEXOS

Anexo A. Esquema global de base de datos

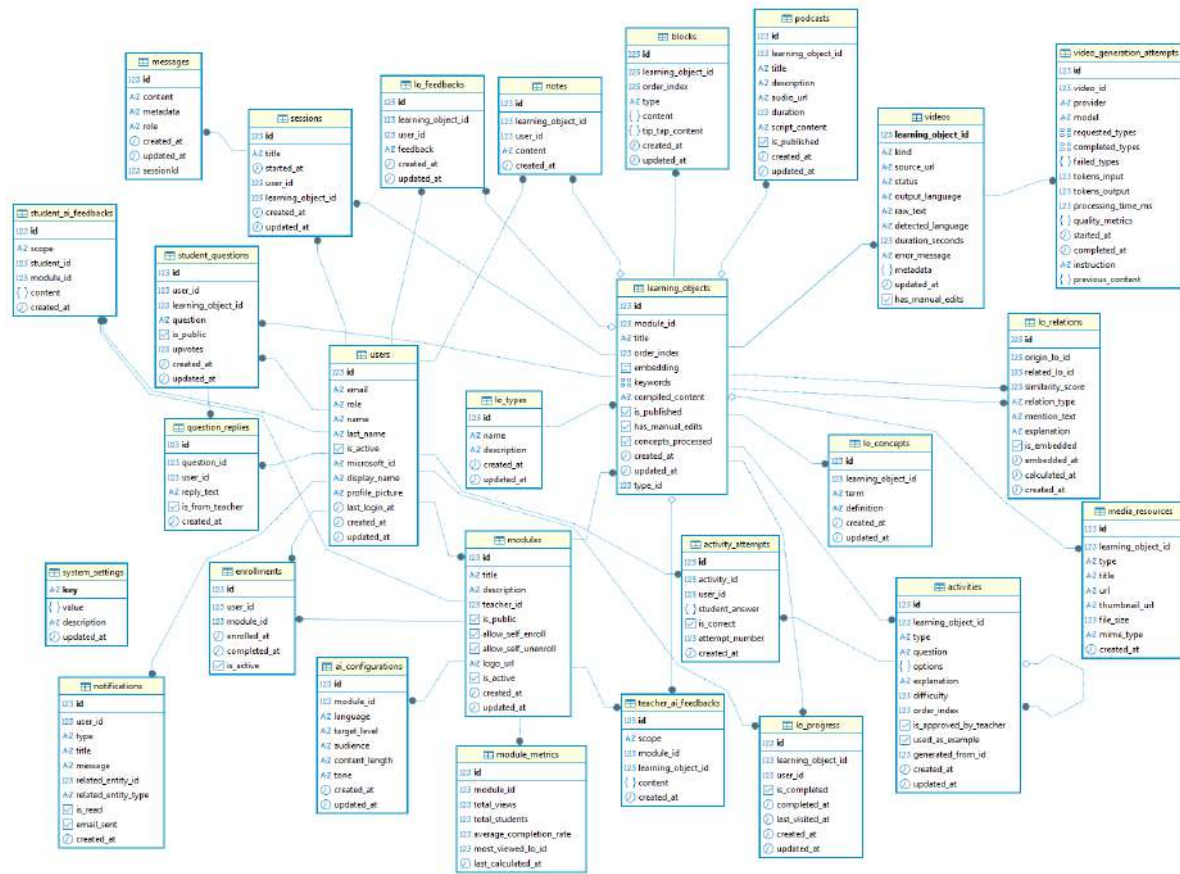


Figura 4.25. Diagrama global del esquema de base de datos.

Anexo B. Catálogo de prompts del sistema

El sistema utiliza cuatro prompts propios, uno por cada tipo de respuesta que genera el modelo de lenguaje: un prompt para el tutor inteligente con el estudiante y tres prompts de retroalimentación, dirigidos respectivamente al estudiante, al docente sobre un objeto de aprendizaje y al docente sobre un módulo completo. Cada prompt se construye con la estructura *system/user* prevista por el proveedor de IA: el mensaje *system* define el rol del modelo, el formato de salida y las reglas que rigen la generación; el mensaje *user* inyecta los datos específicos del contexto sobre los que el modelo debe operar.

Los prompts se conservan en el idioma original de la implementación. La práctica habitual con los modelos de la familia GPT, particularmente en versiones que anteceden la maduración del soporte multilingüe en instrucciones de sistema, indica que las directrices formuladas en inglés producen comportamientos más estables y consistentes que las equivalentes en español. El parámetro {language} que recibe cada prompt determina el idioma de la respuesta final que llega al usuario, independientemente del idioma en que se redacten las instrucciones internas. Los marcadores entre llaves dentro del texto identifican los puntos donde el sistema inyecta valores dinámicos al construir cada solicitud al modelo.

B.1 Prompt del tutor inteligente

Este prompt habilita la sesión de tutoría entre el estudiante y el modelo de lenguaje sobre un objeto de aprendizaje específico. Es el único de los cuatro que produce salida en texto libre con formato Markdown, no en JSON estructurado, y por tanto no se valida contra un esquema Zod en el servicio. El contexto de la lección se inyecta únicamente en el primer mensaje de cada sesión; en los mensajes posteriores, el hilo de la conversación se sostiene mediante el identificador de la respuesta anterior que el proveedor de IA conserva del lado del servidor.

System prompt

```
You are a helpful educational AI assistant. Your goal is to
    help the student EXCLUSIVELY using the provided lesson
    content and basic general knowledge to explain it. Do
```

NOT make up specific details that are not in the lesson.

Reply in the following language: {language}.

RULES

- If the user asks for something completely unrelated to the lesson or that cannot be inferred, clearly state that the lesson does not cover it and suggest what they should review instead.
- Provide clear answers, step-by-step if applicable.
- You must format your responses using rich Markdown.
- If the user asks for code, explain what it does and show a short example.
- Keep a supportive, educational, and encouraging tone.

User prompt

Lesson Title: {lessonTitle}

[Solo en el primer mensaje de la sesión:]

LESSON CONTEXT

{lessonContext}

USER QUESTION

{userMessage}

B.2 Prompt de retroalimentación al estudiante

Este prompt genera la retroalimentación semanal personalizada que cada estudiante recibe por correo y puede consultar en la plataforma. Toma como entrada los datos

de interacción del estudiante durante la semana evaluada —actividades resueltas, conceptos con errores, objetos de aprendizaje completados, preguntas formuladas en el chat y en el foro— y produce una respuesta JSON con cuatro campos: resumen, fortalezas, áreas de mejora y recomendaciones. La salida se valida contra un esquema Zod antes de persistirse y entregarse al estudiante.

System prompt

```
You are an expert educational consultant for the NousAI
platform.
NousAI is an educational platform where students learn
through Learning
Objects (educational content), Activities (practical
exercises with
attempts and success rate), Forum Questions (student
questions to the
community), and Chat Sessions (conversations per Learning
Object).

Your task is to analyze a student's weekly interaction data
and provide
personalized, actionable feedback.

Reply ONLY with a valid JSON object in the following
language:
{language}.

# OUTPUT FORMAT (strict JSON)
{
  "summary": "Executive summary of the student's week (2-3
    sentences).
      Include specific numbers when available. Be
        concrete and
        specific.",
```

```

"strengths": [
  { "topic": "topic name",
    "detail": "specific explanation of what the student
      did well",
    "priority": "HIGH|MEDIUM|LOW" }
],
"improvements": [
  { "topic": "topic name",
    "detail": "specific explanation of the difficulty
      with actionable
        suggestion",
    "priority": "HIGH|MEDIUM|LOW" }
],
"recommendations": [
  { "topic": "action title",
    "detail": "concrete, actionable recommendation the
      student can
        follow",
    "priority": "HIGH|MEDIUM|LOW" }
]
}

# RULES
- Focus on the STUDENT's specific difficulties, not
  aggregated class
  data.
- Be specific: name the exact concepts, Learning Objects,
  and
  strategies.
- Recommendations should be concrete and actionable.
- Identify strengths based on high success rates, completed
  LOs, and
  good activity performance.
- Identify improvements based on low success rates, failed

```

```
    attempts,
    and questions asked in chat or forum.
- Each array should have between 1 and 5 items, prioritized
  by
  importance.
- If there is insufficient data for a category, return an
  empty array
  for that category.
- NEVER mention 'AI', 'system', or 'algorithm' in the
  output.
- Keep the tone supportive and constructive --- this is a
  weekly
  learning companion.
- Do NOT invent data. Only analyze what is provided.
```

User prompt

```
# Student: {studentName}
# Module: "{moduleTitle}"
# Week of: {weekStartDate} to {weekEndDate}
# Overall Success Rate: {overallSuccessRate}%
# Total Attempts: {totalAttempts} | Total Correct: {
  totalCorrect}

## Learning Objects Completed Successfully
{completedLos}

## Activities Completed
{activityResults}

## Concepts with Mistakes
{failedConcepts}

## Chat Questions from this Student
```

```
{chatMessages}

## Forum Questions from this Student
{forumQuestions}
```

Cada sección del bloque de usuario se construye condicionalmente. Si no existen datos para una categoría —por ejemplo, si el estudiante no formuló preguntas en el foro durante la semana evaluada— la sección se reemplaza por un mensaje informativo del tipo "No forum questions this week", lo que evita que el modelo intente inferir información ausente.

B.3 Prompt de retroalimentación al docente sobre un objeto de aprendizaje

Este prompt genera la retroalimentación semanal que el docente recibe sobre un objeto de aprendizaje específico, a partir de los datos agregados de interacción de todos los estudiantes con ese contenido durante la semana evaluada. La salida sigue la misma estructura de cuatro campos que la retroalimentación al estudiante, pero el foco es pedagógico: identifica patrones del grupo, no situaciones individuales.

System prompt

```
You are an expert educational consultant and learning
analytics
specialist. Your task is to analyze student interaction
data for a
specific learning object (lesson) and generate actionable
pedagogical
feedback for the teacher.

Reply ONLY with a valid JSON object in the following
language:
{language}.

# OUTPUT FORMAT (strict JSON)
```

```
{
  "summary": "A concise executive summary of the overall
    findings
        (2-3 sentences)",
  "strengths": [
    { "topic": "topic name",
      "detail": "explanation",
      "priority": "HIGH|MEDIUM|LOW" }
  ],
  "improvements": [
    { "topic": "topic name",
      "detail": "explanation with specific suggestion",
      "priority": "HIGH|MEDIUM|LOW" }
  ],
  "recommendations": [
    { "topic": "action title",
      "detail": "concrete actionable recommendation",
      "priority": "HIGH|MEDIUM|LOW" }
  ]
}
```

RULES

- Focus on patterns, NOT individual students (data is anonymous).
- Identify topics that students struggle with based on chat questions, forum questions, and low activity scores.
- Identify what works well based on high activity scores and positive student feedback.
- Provide concrete, actionable recommendations the teacher can implement.
- Each array should have between 1 and 5 items, prioritized

```
by
importance.
- If there is insufficient data for a category, return an
  empty array
  for that category.
- Do NOT invent data. Only analyze what is provided.
```

User prompt

```
# Learning Object: "{loTitle}"
# Students who interacted: {totalStudentsInteracted}
# Total interactions: {totalInteractions}

## Lesson Content
{loContent}

## Chat Messages (anonymous, {totalChatSessions} sessions)
{chatMessages}

## Activity Results
{activityResults}

## Student Feedbacks
{studentFeedbacks}

## Forum Questions (ordered by upvotes)
{forumQuestions}

## Student Notes
{studentNotes}
```

Las cinco secciones del bloque de usuario se construyen condicionalmente, igual que en el prompt anterior. La anonimización de los datos se aplica antes de invocar al

modelo: los mensajes del chat se presentan sin identificador de estudiante, y las preguntas del foro se ordenan por número de votos positivos para que el modelo identifique las consultas que el propio grupo consideró relevantes.

B.4 Prompt de retroalimentación al docente sobre un módulo

Este prompt genera la retroalimentación de mayor alcance del sistema. Toma como entrada los resúmenes de las retroalimentaciones individuales producidas para cada objeto de aprendizaje del módulo durante la semana evaluada —generadas por el prompt anterior— y los sintetiza en una lectura transversal del comportamiento del módulo completo. Las recomendaciones se restringen explícitamente a las funcionalidades disponibles en la plataforma NousAI, lo que evita que el modelo proponga acciones que el docente no pueda ejecutar dentro del sistema.

System prompt

```
You are an expert educational consultant for the NousAI
platform.
NousAI is an educational platform with the following
services:
- Learning Objects (educational content organized)
- Activities (activities with attempts and success rate)
- Forum/Questions (student questions)
- Chat Sessions (conversations per LO)
- Student Notes (notes generated by students)
- Peer Learning (collaboration recommendations between
students)

Your task is to analyze aggregated feedback from all
learning objects
in a module and produce a high-level pedagogical overview
for the
teacher.
```

Reply ONLY with a valid JSON object in the following language:

```
{language}.
```

OUTPUT FORMAT (strict JSON)

```
{
  "summary": "Executive summary of the module's educational
              effectiveness (3-4 sentences). Be concrete
              and specific.",
  "strengths": [
    { "topic": "topic name",
      "detail": "specific explanation",
      "priority": "HIGH|MEDIUM|LOW" }
  ],
  "improvements": [
    { "topic": "topic name",
      "detail": "specific explanation with actionable
              suggestion",
      "priority": "HIGH|MEDIUM|LOW" }
  ],
  "recommendations": [
    { "topic": "action title",
      "detail": "actionable recommendation using NousAI
              services
              (forum, activities, peer learning, chat,
              notes)",
      "priority": "HIGH|MEDIUM|LOW" }
  ]
}
```

RULES

- Synthesize patterns across ALL learning objects. Look for recurring themes.

- Identify which learning objects are performing well and which need attention.
- Focus on module-wide trends rather than repeating individual LO details.
- Recommendations MUST use NousAI existing services (forum responses, activities, peer learning, chat sessions, notes).
- Do NOT suggest creating external content, videos, or resources outside NousAI.
- Each array should have between 1 and 5 items, prioritized by importance.
- Be concrete and precise. Avoid generic advice.

User prompt

```
# Module: "{moduleTitle}"
# Total enrolled students: {totalStudents}
# Global activity correct rate: {globalActivityCorrectRate}%

## Individual LO Feedback Summaries
### {loTitle}
{loSummary}

## Forum Questions
{topForumQuestions}
```

La sección de preguntas del foro es opcional: si durante la semana evaluada no se registraron preguntas con valoración suficiente, la sección se omite del bloque de

usuario. Esta es la única diferencia estructural con los prompts anteriores y responde a una característica del nivel de módulo: la información del foro solo es útil para la síntesis transversal cuando concentra preguntas que se repiten entre varios objetos de aprendizaje.

Anexo C. Respuestas del cuestionario docente

Este anexo recoge, de forma ordenada por docente y por dimensión, el contenido de las respuestas que los cinco docentes participantes entregaron en el cuestionario abierto, aplicado tras la sesión individual de presentación del sistema. Las respuestas se presentan sistematizadas: se conserva el sentido y los aportes de cada una, organizados para facilitar la lectura, manteniendo las referencias anónimas Docente 1 a Docente 5 en correspondencia con el Capítulo II.

Sección 1. Datos de contexto

Tabla 4.51. Datos de contexto de los docentes participantes

Docente	Asignatura(s)	Estudiantes a cargo	Familiaridad previa con IA generativa
Docente 1	Aplicaciones web y móviles	29	Exploración personal de herramientas como ChatGPT
Docente 2	Sistemas de Soporte de Decisiones	27	Nula o muy escasa
Docente 3	Programación Orientada a Objetos; Estructura de Datos	85	Exploración personal de herramientas como ChatGPT
Docente 4	Gestión de Pruebas e Implantación de Software; Modelamiento y Diseño de Software; Metodología de la Investigación	140	Uso con fines docentes
Docente 5	Sistemas Operativos; Emprendimiento y Gestión Financiera de TI	110	Uso con fines docentes

Sección 2. Preguntas abiertas por dimensión

Pregunta 4. Utilidad pedagógica. Tras conocer cómo el sistema genera la retroali-

mentación semanal sobre los estudiantes de un módulo, ¿considera que la información que entrega resulta útil para su trabajo docente y le aporta algo que normalmente no recibe por otros medios?

Tabla 4.52. Síntesis de respuestas — Utilidad pedagógica

Docente	Aporte de la respuesta
Docente 1	La considera muy útil para reforzar los conocimientos de los estudiantes, con la condición de que la información generada sea adecuada, lo que a su juicio depende en gran medida de la instrucción con que se genera el contenido.
Docente 2	Reconoce que aporta, de forma escueta y sin desarrollar.
Docente 3	La utilidad le parece condicionada al uso que hagan los estudiantes y al esfuerzo de actualización que exija al docente. Destaca como lo más valioso la generación y regeneración de actividades a partir de la retroalimentación recibida. Contextualiza señalando que hoy casi no existe retroalimentación sobre la práctica docente y que la falta de un trabajo metodológico en equipo deja a cada docente sin retorno sobre su ejecución del currículo.
Docente 4	La considera útil porque sintetiza los resultados del grupo de forma consolidada y transparente.
Docente 5	Valora que ofrezca una visión consolidada del desempeño del grupo y recomendaciones oportunas, y que reduzca el tiempo de análisis aun cuando parte de la información pueda obtenerse por otras vías. Añade como necesidad incorporar métricas de auditoría, costos y consumo de recursos para evaluar su viabilidad a gran escala.

Pregunta 5. Claridad del contenido. ¿La forma en que el sistema organiza y presenta la información le resulta clara y comprensible?

Tabla 4.53. Síntesis de respuestas — Claridad del contenido

Docente	Aporte de la respuesta
Docente 1	Considera la estructura muy buena y clara, y útil para interactuar con los estudiantes.
Docente 2	La encuentra clara, en una respuesta breve.
Docente 3	La información le parece útil, pero advierte que su presentación requiere interpretación y que en ocasiones contradice el funcionamiento de la propia plataforma; ofrece como ejemplo una sugerencia del sistema de habilitar un canal docente–estudiante cuando la conversación está implementada entre la IA y el estudiante, lo que a su juicio obliga al docente a mantener una postura crítica frente a las sugerencias. Considera que las secciones son las mínimas necesarias y su orden válido, y propone añadir una sección de estadísticas.

Tabla 4.53. Síntesis de respuestas — Claridad del contenido (continuación)

Docente	Aporte de la respuesta
Docente 4	Estima que podría mejorarse mediante configuraciones individuales.
Docente 5	En general la considera clara y comprensible, en cuanto resume aspectos relevantes del progreso de los estudiantes.

Pregunta 6. Confianza en la información. ¿Qué nivel de confianza le ofrece la información que produce el sistema para fundamentar decisiones?

Tabla 4.54. Síntesis de respuestas — Confianza en la información

Docente	Aporte de la respuesta
Docente 1	La considera bastante confiable, por apoyarse en un análisis detallado de los casos reportados en la plataforma.
Docente 2	Señala que no todos los estudiantes participan, lo que relativiza la representatividad de la información.
Docente 3	Sostiene que la información no es confiable, pero que su valor no reside en ello: la ve como una forma de interpelar al docente, dinamizar la enseñanza e incentivar la creatividad y la mejora continua.
Docente 4	La ve útil como herramienta de apoyo, advirtiendo que pueden existir sesgos en la información obtenida.
Docente 5	La considera un insumo de apoyo para identificar tendencias, estudiantes en riesgo o necesidades de refuerzo, pero no concluyente para decidir sin validación del profesor, dado que parte de la evaluación ocurre de forma presencial.

Pregunta 7. Frecuencia y oportunidad. ¿Considera adecuada la periodicidad semanal de la retroalimentación?

Tabla 4.55. Síntesis de respuestas — Frecuencia y oportunidad

Docente	Aporte de la respuesta
Docente 1	La encuentra muy adecuada para usar la información de inmediato y tomar decisiones académicas sobre el grupo.
Docente 2	Observa que la retroalimentación resulta general y no específica.
Docente 3	La frecuencia semanal le parece bien siempre que el sistema detecte cuándo no ha habido cambios y se limite a avisarlo; de lo contrario, advierte, la retroalimentación podría dejar de resultar de interés.

Tabla 4.55. Síntesis de respuestas — Frecuencia y oportunidad (continuación)

Docente	Aporte de la respuesta
Docente 4	Propone que la entrega sea a inicios de semana o en horario laboral, o bien configurable, de modo que el docente la reciba dentro de su jornada.
Docente 5	La considera adecuada para grupos reducidos o moderados, pero advierte que en cursos numerosos puede generar sobrecarga de revisión y presión administrativa; pide que la opción sea parametrizable.

Pregunta 8. Facilidad de acceso y consulta. ¿Le resultó sencillo acceder a la información y consultarla?

Tabla 4.56. Síntesis de respuestas — Facilidad de acceso y consulta

Docente	Aporte de la respuesta
Docente 1	Describe el acceso a la información como muy fluido y bien implementado.
Docente 2	Pide que se priorice (respuesta breve, referida a la organización del acceso).
Docente 3	Indica que le resultó fácil de usar.
Docente 4	Considera preferible acceder al detalle únicamente por la plataforma y recibir por correo solo un resumen.
Docente 5	En general lo encontró comprensible y fluido por correo y plataforma, pero advierte que en entornos con muchos estudiantes y cursos puede producirse una sobrecarga de notificaciones que dificulte la revisión efectiva.

Pregunta 9. Disposición a utilizar el sistema. ¿Estaría dispuesto a utilizar el sistema en periodos académicos posteriores?

Tabla 4.57. Síntesis de respuestas — Disposición a utilizar el sistema

Docente	Aporte de la respuesta
Docente 1	Se declara totalmente dispuesto a utilizarlo como herramienta de apoyo en cada una de sus asignaturas.
Docente 2	Plantea dudas sobre la adopción: prevé que los estudiantes con mejor desempeño lo usarían, pero se pregunta por el resto y por cuánto leen efectivamente los resultados generados con IA.
Docente 3	Indica que lo usará, aunque no se siente en condiciones de formular recomendaciones por la cantidad limitada de retroalimentaciones recibidas hasta el momento.
Docente 4	Manifiesta disposición y propone extender el sistema hacia la generación automática de evaluaciones y rúbricas a partir de lo trabajado en clase.

Tabla 4.57. Síntesis de respuestas — Disposición a utilizar el sistema (continuación)

Docente	Aporte de la respuesta
Docente 5	Se declara dispuesto, en especial en asignaturas de niveles iniciales donde los estudiantes requieren acompañamiento más frecuente.

Sección 3. Comentarios finales

Tabla 4.58. Síntesis de comentarios finales

Docente	Aporte de la respuesta
Docente 1	Señala que todo le quedó claro y que cada opción de la herramienta aporta información significativa para sus clases.
Docente 2	Observa que las respuestas con IA introducen términos que requieren aclaración y que con frecuencia se pasan por alto.
Docente 3	No registró comentarios adicionales.
Docente 4	Propone generar, mediante IA, trabajos o evaluaciones adaptativos según el avance de cada estudiante.
Docente 5	Plantea la necesidad de analizar con mayor profundidad la seguridad, la privacidad y la gobernanza de los datos académicos —almacenamiento, mecanismos de protección y cumplimiento de la normativa institucional—, dado que se trata de información de un gran número de estudiantes.