



**IMPLEMENTACIÓN DE UNA PLATAFORMA WEB DE TUTORÍA
INTELIGENTE ADAPTATIVA PARA EL APRENDIZAJE DE POO**

LUIS GERARDO AGUIRRE MÉNDEZ

JOHN FREDY ORTIZ HERRERA

UNIVERSIDAD EAN

FACULTAD DE INGENIERIA

INGENIERIA DE SISTEMAS

BOGOTÁ, COLOMBIA

20/MARZO/2026

Implementación de una plataforma web de tutoría inteligente adaptativa para el aprendizaje de POO

Luis Gerardo Aguirre Méndez

John Fredy Ortiz Herrera

Trabajo de grado presentado como requisito para optar al título de:

Ingeniero de Sistemas

Director (a):

William López Castrillón

Universidad EAN

Facultad de Ingeniería

Ingeniería de Sistemas

Bogotá, Colombia

20/Marzo/2026

Nota de aceptación:

Firma del jurado

Firma del jurado

Firma del director del trabajo de grado

Bogotá, 20/Marzo/2026

Resumen

Este estudio aborda las dificultades que enfrentan los estudiantes de primer semestre en el aprendizaje de la Programación Orientada a Objetos (POO), principalmente al comprender los conceptos fundamentales como clases, objetos, herencia y polimorfismo. Esta problemática se ha agudizado en el ámbito educativo debido al uso inadecuado de herramientas de inteligencia artificial, las cuales suelen proveer soluciones directas a los alumnos sin promover la comprensión del proceso de desarrollo.

Con base en esto, el objetivo de la investigación es diseñar e implementar una plataforma web de tutoría inteligente que se adapte y apoye el aprendizaje de la POO. El sistema generará ejercicios de manera automática, evaluará las respuestas y ofrecerá retroalimentación personalizada.

Se espera que los resultados se centren en la mejora de la comprensión conceptual, la reducción de los errores más frecuentes y el fortalecimiento del aprendizaje progresivo mediante la adaptación dinámica de la dificultad de los ejercicios.

Abstract

Este estudio aborda las dificultades que enfrentan los estudiantes de primer semestre en el aprendizaje de la Programación Orientada a Objetos (POO), principalmente al comprender los conceptos fundamentales como clases, objetos, herencia y polimorfismo. Esta problemática se ha agudizado en el ámbito educativo debido al uso inadecuado de herramientas de inteligencia artificial, las cuales suelen proveer soluciones directas a los alumnos sin promover la comprensión del proceso de desarrollo.

Con base en esto, el objetivo de la investigación es diseñar e implementar una plataforma web de tutoría inteligente que se adapte y apoye el aprendizaje de la POO. El sistema generará ejercicios de manera automática, evaluará las respuestas y ofrecerá retroalimentación personalizada.

Se espera que los resultados se centren en la mejora de la comprensión conceptual, la reducción de los errores más frecuentes y el fortalecimiento del aprendizaje progresivo mediante la adaptación dinámica de la dificultad de los ejercicios.

Introducción

Con la educación superior en la actualidad, aprender programación se ha convertido en una de las necesidades y competencias más fundamentales para formarse como profesionales en el área de ingeniería de sistemas; aun así, diversas investigaciones demuestran que los estudiantes de primer semestre presentan recurrentes dificultades al comprender los conceptos de la programación (Paredes et al.,2023). Algunos de ellos, como clases, objetos, encapsulamiento, herencia y polimorfismo, implican un alto nivel de abstracción, lo que genera barreras cognitivas que afectan considerablemente el rendimiento académico (Paredes et al., 2023).

En la actualidad, la inteligencia artificial (IA) se ha transformado en una herramienta de alto potencial para los procesos educativos. Investigaciones sistemáticas evidencian que las tecnologías basadas en IA logran ofrecer experiencias de aprendizaje personalizadas, generar contenido de forma automatizada y proporcionar retroalimentación inmediata a los estudiantes (Kasneci et al., 2023; Paredes et al., 2023). De hecho, los modelos de lenguaje a gran escala (LLM) destacan en la asistencia de tareas como la generación de código y la explicación de conceptos (Kasneci et al., 2023).

Aun así, estos avances han generado nuevas problemáticas debido a que los estudiantes utilizan estas herramientas como un buscador que entrega respuestas directas, sin desarrollar una comprensión real de los fundamentos lógicos de la programación, riesgo que ya ha sido advertido en la literatura académica reciente (Acofi, 2023; Kasneci et al., 2023).

Ante esta situación, surge la necesidad de reorientar el uso de la IA en la educación hacia el acompañamiento adaptativo. Los sistemas de tutoría inteligente (STI) se presentan como una alternativa viable, ya que estructuran un entorno capaz de analizar el desempeño, identificar errores conceptuales y adaptar contenidos (Luckin et al., 2023). No obstante, se evidencia que muchas soluciones actuales carecen de un enfoque integral que unifique la generación de ejercicios, la evaluación de desempeño y la adaptación en una herramienta coherente (Luckin et al., 2023).

Con todo esto se ha logrado formular la siguiente pregunta para la investigación:

¿Cómo podría una plataforma de tutoría inteligente adaptativa, que se apoye con inteligencia artificial, lograr mejorar el aprendizaje en la programación orientada a objetos para estudiantes de primer semestre?

Marco teórico

Las primeras fases del aprendizaje de software son las que plantean más desafíos en la educación del alumno de ingeniería de sistemas. La mayoría enfrenta problemas para comprender las ideas básicas de la Programación Orientada a Objetos (POO) debido al grado de abstracción requerido y a la necesidad de cultivar destrezas de pensamiento lógico estructurado, como indican varios estudios (Paredes et al., 2023). Este fenómeno afecta directamente el desempeño académico, reduce la motivación y aumenta el índice de deserción temprana en carreras relacionadas (Paredes et al., 2023).

Desde el punto de vista pedagógico, se puede abordar la enseñanza de la programación a través de un enfoque constructivista que propone que el conocimiento se profundiza y se adquiere activamente por medio de la interacción del alumno con problemas reales y su solución gradual. Cuando el alumno consigue entender y asimilar el contexto de cada problema, se produce un aprendizaje significativo, pues no basta con descifrar solamente el resultado final para lograr una educación integral. La aplicación de herramientas automatizadas y sistemas basados en "cajas negras" restringe este proceso en la mayor parte de los entornos educativos contemporáneos, ya que los estudiantes consiguen respuestas directas sin entender las bases subyacentes.

De esta manera, y en respuesta a estas problemáticas, los Sistemas de Tutoría Inteligente (STI) se proponen como una alternativa tecnológica orientada a optimizar los procesos de enseñanza y aprendizaje. Estos sistemas son plataformas basadas en inteligencia artificial que ofrecen apoyo personalizado, evalúan el desempeño, detectan errores conceptuales y adaptan los contenidos a las necesidades de cada alumno. A diferencia de los enfoques tradicionales, los STI promueven un aprendizaje centrado en el

estudiante al proporcionar retroalimentación inmediata y ajustar dinámicamente la dificultad de las actividades.

Bajo esa premisa, el concepto de aprendizaje adaptativo pasa a ser un elemento esencial de estas plataformas. Este enfoque reconoce que los estudiantes tienen diferentes ritmos, diferentes formas y diferentes niveles de conocimiento, por lo que es indispensable personalizar la experiencia educativa. Los sistemas adaptativos toman la información del desempeño del estudiante para modificar la secuencia, cantidad y calidad de los contenidos, optimizando así el nivel de dificultad y las estrategias de enseñanza, lo que contribuye directamente a mejorar la retención del conocimiento y la motivación del estudiante.

En la informática, la inteligencia artificial ha contribuido a la creación de herramientas que permiten generar ejercicios, evaluar de forma automática el código y ofrecer retroalimentación inmediata. Las investigaciones más recientes han podido comprobar que estas tecnologías permiten personalizar los currículos y reconocer los errores más comunes que se cometen durante la escritura de sintaxis (Kasneci et al. 2023). Sin embargo, la literatura también evidencia que gran parte de estas herramientas se enfocan primordialmente en la evaluación automatizada, dejando de lado el acompañamiento pedagógico durante el proceso de resolución de problemas.

Siguiendo esta línea de investigación, existen propuestas actuales que plantean el desarrollo de tutores inteligentes que sean capaces de interactuar con los estudiantes en la construcción de soluciones, proporcionando orientación contextualizada y fomentando el pensamiento crítico. Los sistemas basados en inteligencia artificial se han mostrado sólidos para responder preguntas, ofrecer ejemplos situacionales y guiar a los usuarios a través de ejercicios prácticos (Kasneci et al., 2023). De igual forma, la incorporación de modelos de

lenguaje a gran escala (LLM) facilita la creación de explicaciones y tutorías personalizadas que simulan con éxito la tutoría humana (Kasneci et al., 2023; Luckin et al., 2023).

Los análisis realizados sobre los profesores virtuales basados en IA demuestran que estos sistemas ofrecen ayuda personalizada, pero sin entregar directamente la respuesta al estudiante (Kasneci et al., 2023). Estas estrategias utilizan pistas progresivas y preguntas orientadoras, lo que favorece el desarrollo de habilidades para la resolución de problemas y evita la dependencia excesiva hacia las herramientas tecnológicas.

Si bien se trata de ventajas, la tecnología no debe verse como una solución definitiva. El creciente uso de la inteligencia artificial generativa en el ámbito educativo ha traído consigo nuevos desafíos. Hoy en día, muchos estudiantes recurren a estas herramientas para obtener códigos enteros sin entender el razonamiento detrás de ellos, lo cual está impactando de manera negativa el desarrollo de habilidades cognitivas esenciales. Por eso es crucial diseñar sistemas que introduzcan la IA de forma controlada y vigilada, dirigiéndola estrictamente a respaldar el aprendizaje y no a sustituir el proceso cognitivo del estudiante.

Por ello la comunidad académica coincide en que el valor real de la inteligencia artificial en la educación no está en que sea capaz de generar contenido sino en su integración con sistemas que analicen el desempeño del usuario y que sean capaces de identificar barreras conceptuales para adaptarse a su progreso (Luckin et al., 2023; Paredes et al., 2023). Esto requiere la conjunción de modelos pedagógicos, técnicas de análisis de datos y herramientas de IA para crear entornos educativos más eficaces y personalizados.

Dado que es evidente la necesidad de desarrollar soluciones que conjuguen de manera coherente los principios del aprendizaje constructivista, la tutoría inteligente y el enfoque adaptativo, el presente proyecto propone el diseño de una plataforma web de tutoría inteligente para la enseñanza de la Programación Orientada a Objetos. No sólo genera automáticamente ejercicios y explicaciones, sino que también monitoriza constantemente el progreso del estudiante para identificar errores conceptuales concretos y adaptar la dificultad de los contenidos de forma dinámica en función de sus necesidades específicas.

Estado del arte (Metodología PRISMA)

El presente estado del arte se elaboró mediante una revisión sistemática de la literatura basada en las directrices de la declaración PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) para la identificación, selección y análisis de estudios relevantes relacionados con la aplicación de la inteligencia artificial en el ámbito de la educación superior, más específicamente en la enseñanza de la programación y el desarrollo de Sistemas de Tutoría Inteligente (STI). Este enfoque metodológico asegura la transparencia, replicabilidad y rigurosidad del proceso de selección de la muestra documental. Se desarrolló el proceso de selección en 3 fases consecutivas:

Búsqueda inicial: Se investigaron bases de datos con un alto impacto académico y repositorios institucionales (como IEEE Xplore, ScienceDirect, arXiv y Google Académico), utilizando ecuaciones de búsqueda que combinaron descriptores clave como *"Intelligent Tutoring Systems"*, *"Adaptive Learning"*, *"Object-Oriented Programming"* y *"Artificial Intelligence in Education"*. Con esta primera fase se obtuvieron un total de 8 publicaciones potencialmente relevantes.

Aplicación de criterios de inclusión y exclusión: El total inicial fue filtrado de forma particular, incluyendo solo artículos de revistas indexadas, actas de congresos internacionales y tesis de grado publicadas en los últimos cinco años, priorizando así aquellos trabajos enfocados directamente en el ámbito de la educación superior y la enseñanza de lenguajes de programación. Se descartaron aquellas investigaciones enfocadas a la educación primaria, metodologías de aprendizaje no adaptativas o herramientas de tipo comercial sin documentación de su arquitectura técnica. Después de

este proceso de tamizamiento y de la revisión de los resúmenes, el resultado fue un total de 5 documentos.

Filtro final por pertinencia: Se hizo una lectura crítica del texto completo de los documentos preseleccionados para evaluar su aplicabilidad directa en la resolución de la problemática planteada y para aportar al diseño de la plataforma web propuesta. Finalmente, se escogieron de forma estricta cuatro (4) documentos fundamentales que constituyen el núcleo de este estado del arte por su rigor metodológico, pertinencia pedagógica y vigencia tecnológica dentro del ecosistema de la inteligencia artificial educativa.

Autor(es) / Año	Título del estudio	Objetivo principal	Metodología empleada	Variables analizadas	Principales hallazgos	Limitaciones reportadas	Relación con el proyecto actual
UDLA (2023)	Uso de inteligencia artificial en entornos educativos .	Analizar el impacto que tiene la inteligencia artificial en los procesos de enseñanza y aprendizaje en la educación superior.	Estudio descriptivo con un enfoque cualitativo que se basa en la revisión de literatura y análisis de algunos casos educativos.	Uso de IA, aprendizaje del estudiante, interacción educativa, rendimiento académico.	La IA llega a mejorar la personalización del aprendizaje y logra facilitar la comprensión de contenidos, requiere integración pedagógica adecuada.	Falta de validación empírica en los entornos reales los cuales tienen escasez de datos cuantitativos .	Aporta a el fundamento teórico del uso de IA en la educación, lo cual valida la necesidad de integrar la inteligencia artificial en la plataforma que se propone.

arXiv (2023)	Applications of Large Language Models in Education	Explorar el uso de los modelos de lenguaje en educación y su impacto en el aprendizaje.	Revisión sistemática de la literatura sobre algunas aplicaciones de modelos de lenguaje en los contextos educativos.	Uso de LLMs, generación de contenido, retroalimentación automática, desempeño del estudiante.	Los modelos de lenguaje nos permiten generar contenido educativo personalizado y poder mejorar la retroalimentación, pero pueden presentar riesgos de dependencia.	Existe un posible sesgo en las respuestas que se generan y la falta de control pedagógico en la implementación.	Fundamenta el uso de toda la IA para la generación de ejercicios y la retroalimentación dentro de un sistema propuesto.
-----------------	--	---	--	---	--	---	---

Wiley (2023)	Artificial Intelligence in Education: Opportunities and Challenges	Analizar las oportunidades y desafíos que puede tener la inteligencia artificial en la educación.	Estudio analítico que se basa en la revisión de literatura y el análisis conceptual.	Uso de IA, personalización del aprendizaje, ética en educación, interacción humana con máquina.	La IA nos permite mejorar la personalización de todo el aprendizaje, pero requiere una regulación y diseño pedagógico adecuado.	Falta de varios estudios longitudinales y la evidencia empírica a una gran escala.	Logra justificar la necesidad de poder diseñar un sistema estructurado que logre controlar el uso de IA en el aprendizaje.
ACOFI (2023)	Uso ético de ChatGPT en la enseñanza de la programación	Analizar todo el impacto del uso de ChatGPT en el aprendizaje de la programación y su implicación ética.	Estudio descriptivo con un análisis de algunos casos en entornos educativos.	Uso de IA, comprensión de programación, dependencia tecnológica, aprendizaje autónomo.	El uso de IA puede lograr mejorar el aprendizaje si se llega a guiar adecuadamente, pero este genera dependencia si se usa sin control.	Limitado a algunos contextos específicos y a la falta de evaluación cuantitativa.	Refuerza toda la necesidad de un tutor inteligente que pueda controlar el uso de la IA y promueva el aprendizaje real.

La comunidad académica concuerda en que el beneficio primordial de la inteligencia artificial aplicada a la educación es la adaptabilidad; los sistemas actuales consiguen modificar dinámicamente tanto la secuencia de contenidos como el nivel de complejidad de las actividades evaluativas en función de las trazas de rendimiento del usuario. Esta característica es crucial para disminuir las barreras de abstracción propias de los campos complejos como la Programación Orientada a Objetos (POO).

Sin embargo, la literatura también advierte sobre limitaciones críticas, como el riesgo de desarrollar una dependencia algorítmica si las herramientas entregan soluciones de código directas y explícitas, lo que inhibe el pensamiento lógico y las habilidades de resolución de problemas. En este contexto, el proyecto en cuestión se fundamenta en la creación de una plataforma web inteligente y adaptativa de tutoría centrada en la POO, construida bajo una perspectiva pedagógica que limita la aplicación de la IA al apoyo controlado, estimulando el razonamiento crítico a través del apoyo interactivo y la utilización progresiva de indicaciones.

Metodología

Se realizó este estudio a través de un enfoque mixto que combina aspectos cuantitativos y cualitativos para analizar el impacto de una plataforma web de tutoría inteligente adaptativa sobre el aprendizaje de la Programación Orientada a Objetos (POO). El componente cuantitativo se centró en medir el desempeño académico mediante datos del sistema, como tasas de aciertos, el número de intentos, tiempos de resolución y progresión a través de los niveles de dificultad. En cuanto al componente cualitativo, se evaluaron las opiniones de los estudiantes en relación con la usabilidad de la interfaz, la claridad de las explicaciones y la eficacia de la retroalimentación. La intersección de ambos métodos permitió evaluar tanto el rendimiento técnico como la experiencia del usuario en entornos virtuales de formación (Kasneci et al., 2023).

La estrategia metodológica es de tipo no experimental, no sesgada temporalmente y con alcance descriptivo-correlativo. No experimental por la falta de manipulación en una circunstancia controlada, sino la observación del comportamiento de los sujetos en un ambiente auténtico y académico de interacción con el software. No sesgada temporalmente, porque la recopilación de percepciones se limitó a un intervalo específico posterior al uso de la plataforma. Por último, la escala de validez del estudio fue descriptivo correlativa pues pretende describir los patrones de uso y establecer relaciones estadísticas entre variables cruciales, como los tipos de errores, los niveles de dificultad alcanzados y el ritmo de avance del estudiante.

La población objetivo estudió programas de ingeniería de sistemas o disciplinas afines con estudiantes del primer semestre. Ellos se hallaban cursando cursos introductorios sobre programación y un bloque adicional de programación orientada a objetos.

Instrucciones de recolección

Para la recopilación de datos, se utilizaron simultáneamente métodos mecánicos internos y fuera del sistema:

Registro automático del intercambio (Log Data): El servidor del programa generó registros en tiempo real de la interacción del usuario con su aprendizaje (comprobaciones de respuesta, tiempos de resolución, intentos y errores en la sintaxis). Este método se ajusta a los conceptos de análisis educativo (learning analytics) y hace referencia a la gestión de datos de la plataforma como una herramienta valiosa para examinar el proceso educativo de manera objetiva.

Cuestionarios estructurados: Se diseñó un instrumento para determinar cuán clara consideraban los alumnos el contenido expositivo, la relevancia del autoaprendizaje y la accesibilidad del entorno web.

Todos los instrumentos pasaron por un proceso de validación mediante evaluación por parte de expertos en educación en programación y se mejoraron a través de una prueba inicial con una pequeña muestra de estudiantes para asegurar que el contenido era coherente, claro y relevante antes de su uso completo (Wiley, 2023). la percepción del usuario y evaluación de sistemas interactivos (Wiley, 2023).

Desarrollo Técnico y Fases del Procedimiento

Para abordar con la precisión necesaria el desarrollo de la plataforma digital, el procedimiento de investigación se articuló en tres fases consecutivas que comprendieron

actividades de ingeniería y experimentación.

Fase de ingeniería y arquitectura de software: se diseñó e implementó una arquitectura cliente-servidor compuesta por tres capas funcionalmente desacopladas, lo cual permitió gestionar eficazmente la persistencia de datos y la lógica del sistema. Para la interfaz de usuario, se empleó React junto con Vite en el desarrollo del frontend, lo que posibilitó la estructuración de una interfaz interactiva con cuatro vistas dinámicas esenciales: autenticación de usuarios, selección de temáticas, resolución de problemas y seguimiento analítico del progreso. El componente de backend se construyó utilizando Python y FastAPI, en el cual se definieron nueve puntos de acceso RESTful, dedicados específicamente a las operaciones de la lógica de negocio. Para garantizar la integridad y validez de la información, se implementó una restricción a nivel de red en el backend que suprimía ciertos atributos de las consultas iniciales provenientes del frontend, con el fin de prevenir la interceptación de soluciones por parte de los estudiantes mediante la inspección del tráfico. La persistencia de los datos del sistema, así como el estado particular de cada estudiante, se gestionó mediante una base de datos relacional PostgreSQL.

Fase de Implementación del Algoritmo Adaptativo y Conexión de IA: La integración del núcleo del software se abordó desde una perspectiva híbrida, enmarcada en la fase de implementación del algoritmo adaptativo y la conexión con sistemas de inteligencia artificial. El motor adaptativo se materializó a través de un algoritmo determinista e intrínsecamente independiente, ejecutado en el backend y regulado por una ventana deslizante que analizaba las cinco respuestas más recientes. Este algoritmo ajustaba dinámicamente el nivel de dificultad: si el promedio de aciertos era igual o

superior al 80%, el sistema procedía a incrementar el nivel de dificultad (en una escala de 1 a 5); si, por el contrario, este promedio era igual o inferior al 45%, se reducía dicho nivel; y si el desempeño se situaba en un rango intermedio, la complejidad se mantenía constante. Paralelamente, se incorporó un servicio de Inteligencia Artificial de soporte, de naturaleza independiente, mediante la API de Claude (Anthropic). A este modelo se le delegó la responsabilidad exclusiva de generar contenido textual de manera dinámica, proporcionar explicaciones teóricas y ofrecer retroalimentación estructurada, sin que ello implicara la exposición de su código fuente.

Fase de aplicación y analítica educativa: los estudiantes interactuaron directamente con la plataforma al resolver ejercicios secuenciales enfocados en Programación Orientada a Objetos. Estos ejercicios se organizaron en nueve temas distintos, y el acceso a cada tema subsiguiente dependía de alcanzar un umbral de dominio (mastery score) igual o superior al 60%. De forma simultánea, el sistema registró automáticamente las métricas de rendimiento de los usuarios en la base de datos relacional. Esta recopilación de datos fue fundamental para el análisis estadístico posterior, lo que permitió investigar la correlación entre la aplicación de los algoritmos de adaptación del software y la mitigación de errores recurrentes identificados en el aprendizaje de los alumnos.

La fundamentación de este procedimiento se sustenta en investigaciones recientes que indican que los sistemas de tutoría inteligente, al ser asistidos por modelos de lenguaje, son capaces de ofrecer una asistencia contextualizada y significativamente adaptativa. No obstante, esto requiere que su integración se efectúe bajo un diseño de ingeniería controlado y un marco pedagógico rigurosamente estructurado (arXiv, 2023; Revista UDLA, 2023).

Resultados

Materialización de la Plataforma Web de Tutoría Inteligente Adaptativa

Diseño de la arquitectura del sistema

El hallazgo central de la presente investigación consiste en la materialización y puesta en funcionamiento de una plataforma web con capacidades adaptativas, concebida para la instrucción en Programación Orientada a Objetos (POO). Seguidamente, se procederá a la exposición de los elementos técnicos que sustentan la arquitectura propuesta, el diseño del esquema relacional de datos, así como las métricas empleadas para evaluar el desempeño del algoritmo adaptativo desarrollado.

Fue establecida una arquitectura cliente-servidor basada en una tríplice capaz de ser desacoplada por medio del patrón arquitectónico estructurado, logrando la redundancia en el sistema y el funcionamiento coherente del entorno tecnológico.

La Capa de Aplicación se implementó con una interfaz reactiva para generar el entorno del proceso laboral del estudiante. La Capa de Negocios de la Lógica centraliza la ejecución de pedidos a través de servicios RESTful, ubicando el algoritmo adaptativo en un servidor local. Por último, la Capa de Persistencia confirma la consistencia del modelo referencial de los registros generados por los usuarios.

Como elemento de eficiencia operacional, la inteligencia artificial fue incluida como servicio adicional de inteligencia artificial, disponible mediante la API de Claude (Anthropic). Así, esta configuración permitió que, a causa de cualquier falla de red o aislamiento de respuesta de la fuente de la IA, los componentes fundamentales de la aplicación web pudieran continuar funcionando en la parte local y predecible.

Tabla 1

Capa	Tecnología	Responsabilidad principal
Frontend	React + Vite	Interfaz del estudiante, visualización de ejercicios y métricas
Backend	Python + FastAPI	Lógica adaptativa, evaluación de respuestas, gestión de progreso
Base de datos	PostgreSQL	Almacenamiento de usuarios, ejercicios, respuestas y progreso
IA auxiliar	Claude API (Anthropic)	Generación de ejercicios, explicaciones y retroalimentación textual

Nota. Elaboración propia.

Diseño e implementación de la base de datos

El modelo de datos desarrollado está compuesto por seis entidades relacionales, las cuales permiten representar tanto el contenido educativo como el seguimiento del proceso de aprendizaje de cada estudiante. Dentro de este modelo, la entidad `student_progress` cumple un papel central, ya que registra, para cada combinación de usuario y tema, el nivel de dificultad actual, el puntaje de dominio acumulado y una ventana deslizante con las últimas cinco respuestas.

Gracias a esta estructura, el sistema puede ajustar el nivel de dificultad sin necesidad de consultar todo el historial del estudiante, lo que mejora considerablemente el rendimiento.

Durante el proceso de diseño se definieron nueve temas de aprendizaje, organizados de forma secuencial según su nivel de complejidad, comenzando por los fundamentos de programación y avanzando hasta conceptos como el polimorfismo. Además, se implementó un mecanismo de desbloqueo progresivo: el estudiante solo puede avanzar al siguiente tema cuando alcanza un puntaje de dominio superior a 0.6 (60%) en el tema actual. Esto asegura una progresión coherente dentro del proceso de aprendizaje.

Implementación del backend y motor adaptativo

El backend fue desarrollado utilizando FastAPI y cuenta con nueve endpoints REST que cubren funcionalidades clave como autenticación, gestión de temas, generación y evaluación de ejercicios, así como el análisis del aprendizaje.

Uno de los aportes más relevantes en esta etapa fue la implementación del motor adaptativo como un algoritmo independiente de la inteligencia artificial. En lugar de depender de modelos externos, se diseñó una solución basada completamente en lógica de software. Este algoritmo funciona a partir de una ventana deslizante que considera las últimas cinco respuestas del estudiante en cada tema. Cuando el promedio de estas respuestas supera el 80%, el nivel de dificultad aumenta en una unidad (hasta un máximo de 5). Por el contrario, si el promedio es inferior al 45%, el nivel disminuye (con un mínimo de 1). Si el resultado se encuentra entre estos valores, el nivel se mantiene sin cambios.

En cuanto al puntaje de dominio, este se calcula mediante una combinación ponderada entre el desempeño histórico (40%) y el desempeño reciente (60%), dando mayor importancia a las respuestas más actuales para reflejar de manera más precisa el estado del aprendizaje del estudiante.

Tabla 2

Parámetro	Valor	Descripción
Umbral de subida	$\geq 80\%$	Promedio de ventana para aumentar nivel
Umbral de bajada	$\leq 45\%$	Promedio de ventana para reducir nivel
Tamaño de ventana	5 respuestas	Número de respuestas recientes consideradas
Peso desempeño reciente	60%	Ponderación en el cálculo del mastery score
Peso desempeño histórico	40%	Ponderación en el cálculo del mastery score
Umbral de desbloqueo de tema	$\geq 60\%$	Mastery score mínimo para acceder al siguiente tema
Rango de niveles de dificultad	1 – 5	Escala de dificultad de los ejercicios generados

Nota. Elaboración propia.

Implementación del frontend

La interfaz de usuario fue desarrollada en React y permite al estudiante interactuar con la plataforma a través de cuatro vistas principales: registro e inicio de sesión, selección de temas con indicadores de progreso, resolución de ejercicios con retroalimentación inmediata, y visualización del nivel de dominio alcanzado en cada tema.

Un aspecto importante del diseño fue la decisión de no mostrar la respuesta correcta antes de que el estudiante responda. Para lograrlo, el campo `correct_answer` se excluye intencionalmente de las respuestas de la API y solo se envía después de que el usuario ha contestado.

Con esta medida se evita que el estudiante pueda acceder a la respuesta correcta mediante la inspección del tráfico de red, lo que contribuye a mantener la integridad y validez del proceso de evaluación.

Procesamiento de los datos

Durante la interacción con la plataforma, se manejan de forma continua dos tipos de datos: los relacionados con la configuración del contenido educativo y los generados a partir del desempeño del estudiante.

Por un lado, los datos de configuración incluyen los nueve temas de programación orientada a objetos (POO) y los ejercicios asociados a cada uno. Esta información se almacena en las tablas `topics` y `exercises` dentro de la base de datos PostgreSQL, y es consultada por el backend cada vez que se solicita un ejercicio. Cabe mencionar que los ejercicios pueden ser generados dinámicamente mediante inteligencia artificial o añadidos manualmente por un administrador, lo que permite que el sistema siga funcionando incluso sin conexión a servicios externos.

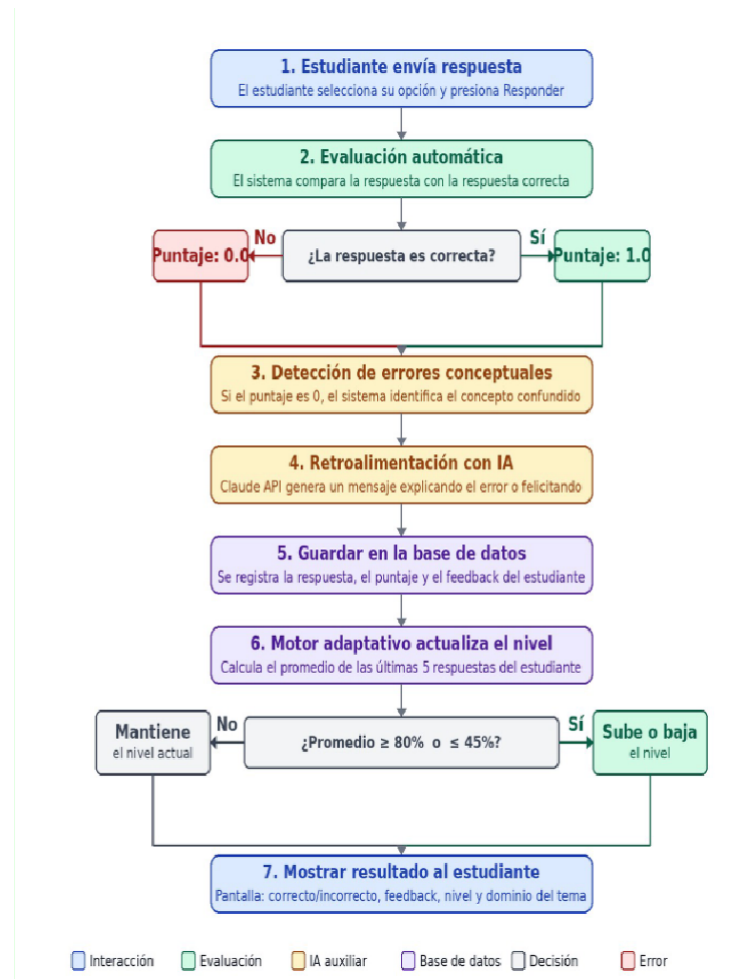
Por otro lado, los datos de desempeño se generan cada vez que el estudiante responde un ejercicio. Esta información se registra en la tabla `submissions`, donde se almacena la respuesta enviada, si fue correcta o no, el puntaje obtenido (en una escala de 0.0 a 1.0) y el tiempo empleado en responder.

Adicionalmente, cuando se detecta una respuesta incorrecta, el sistema ejecuta un algoritmo que analiza posibles patrones de error. Este proceso permite identificar confusiones conceptuales específicas según el tipo de ejercicio y el tema evaluado. Los resultados se almacenan en la tabla `error_patterns` y pueden ser utilizados posteriormente para ofrecer recomendaciones personalizadas al estudiante.

La Figura 1 presenta el flujo de procesamiento de una respuesta desde que el estudiante la envía hasta que el sistema actualiza su perfil de aprendizaje.

Figura 1

Flujo de procesamiento de una respuesta del estudiante

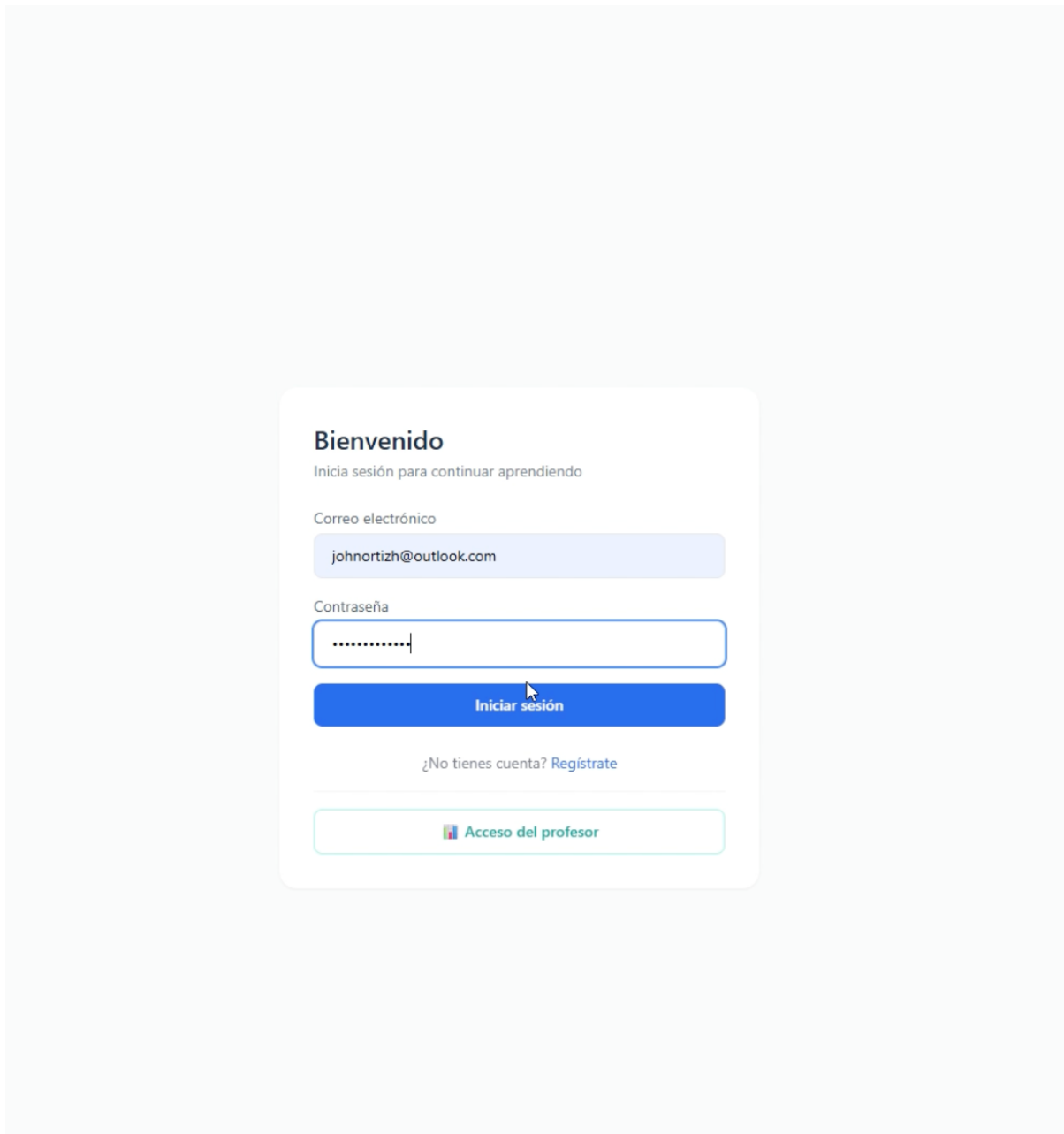


Validación Operativa y Flujo de Interacción del Sistema

Como evidencia final de los resultados tecnológicos, se validó operativamente el sistema mediante la ejecución en tiempo real de los flujos de interacción del usuario. El software mostró una plena estabilidad y consistencia lógica durante todo el ciclo de navegación en producción, que incluye lo siguiente:

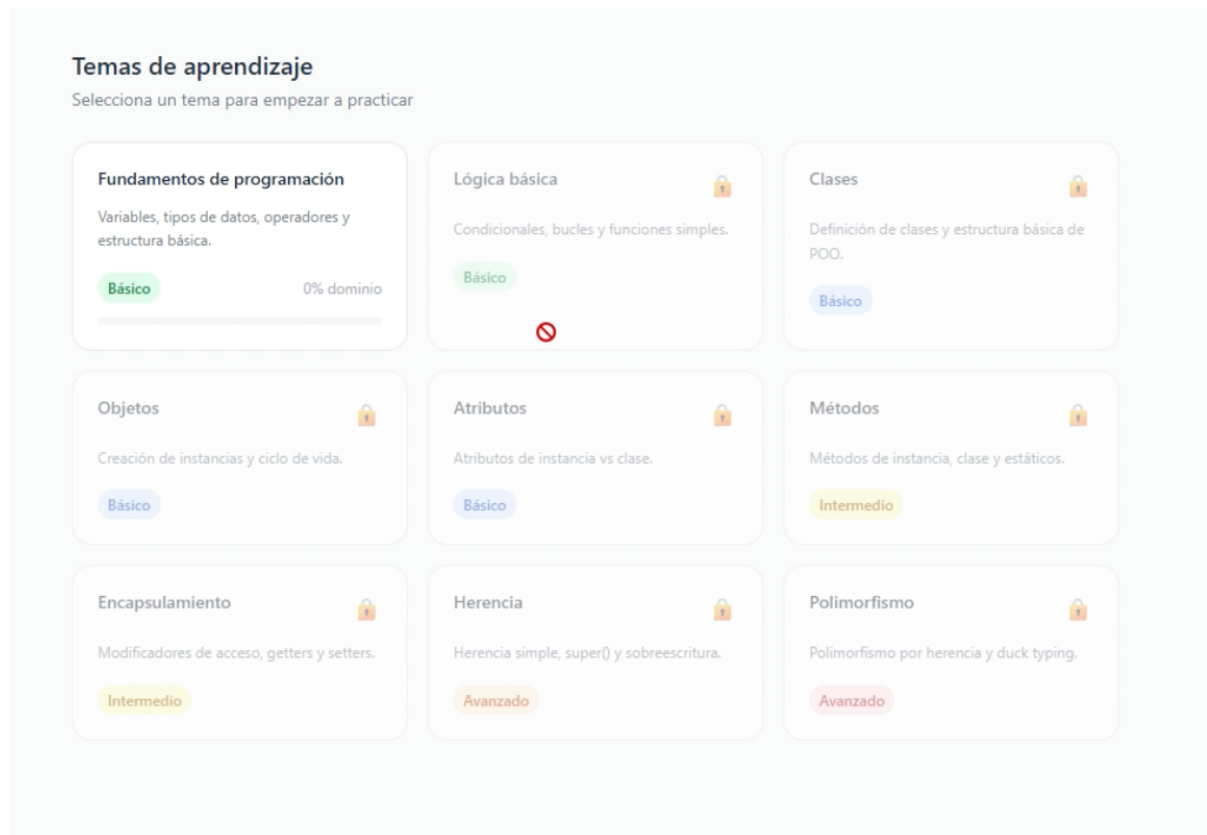
Autenticación e Inicialización: El estudiante accede de forma segura al entorno web y el sistema consigue del endpoint de perfil el estado exacto de su progreso (mastery_score) junto con los temas desbloqueados progresivamente, todo ello recogiendo estos datos de la base de datos

PostgreSQL.



Nota: Imagen del software en funcionamiento.

Despliegue del Entorno de Codificación: Cuando seleccionas una unidad temática existente (por ejemplo, estructura de clases o polimorfismo), el backend analiza los parámetros de nivel de dificultad actual del estudiante (escala 1 a 5) y la API de Claude incluye automáticamente un enunciado de código personalizado y estructurado dinámico.

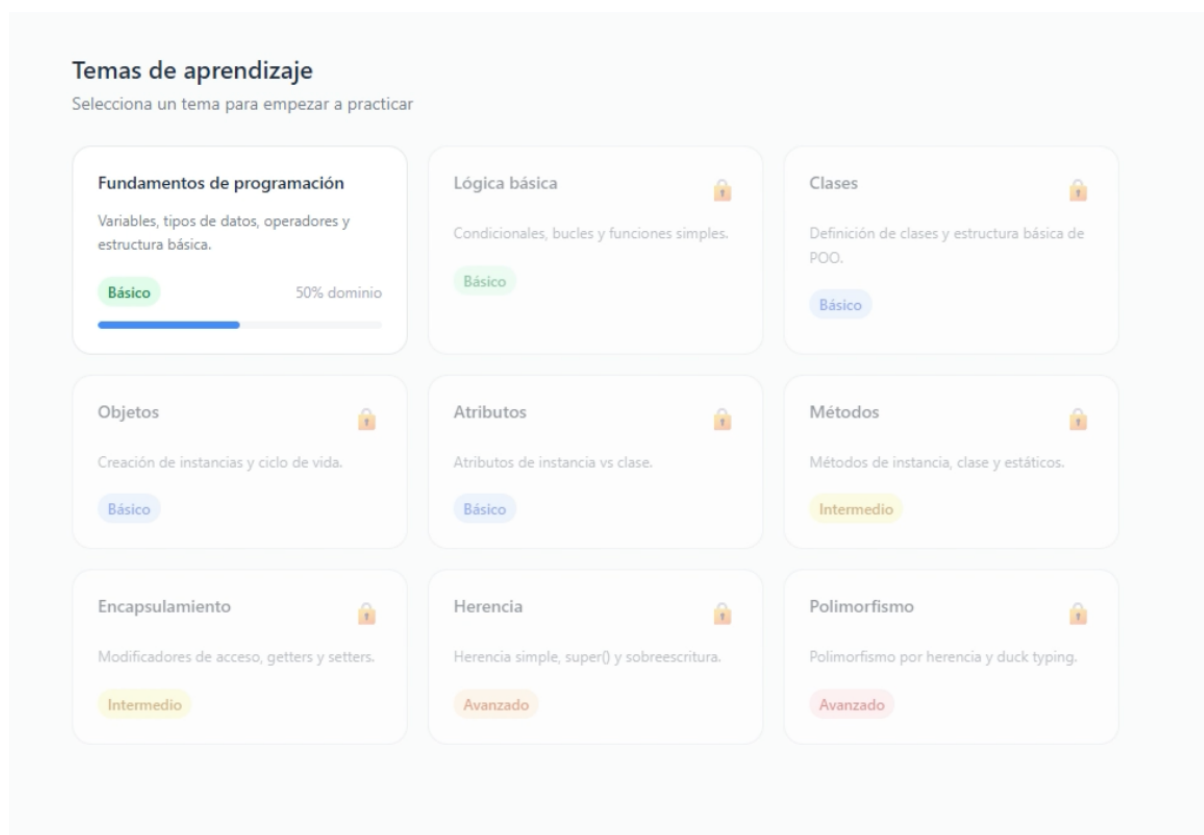


Nota: Imagen del software en funcionamiento.

Procesamiento de respuestas y activación del motor adaptativo: El sistema, cuando recibe una propuesta de solución desde una solicitud HTTP POST, ejecuta la evaluación sintáctica de forma local. Si la respuesta es errónea, el sistema impide el acceso a la solución y en su lugar, proporciona un andamiaje pedagógico introduciendo pistas conceptuales y orientaciones abstractas, dados por la asistente inteligente.

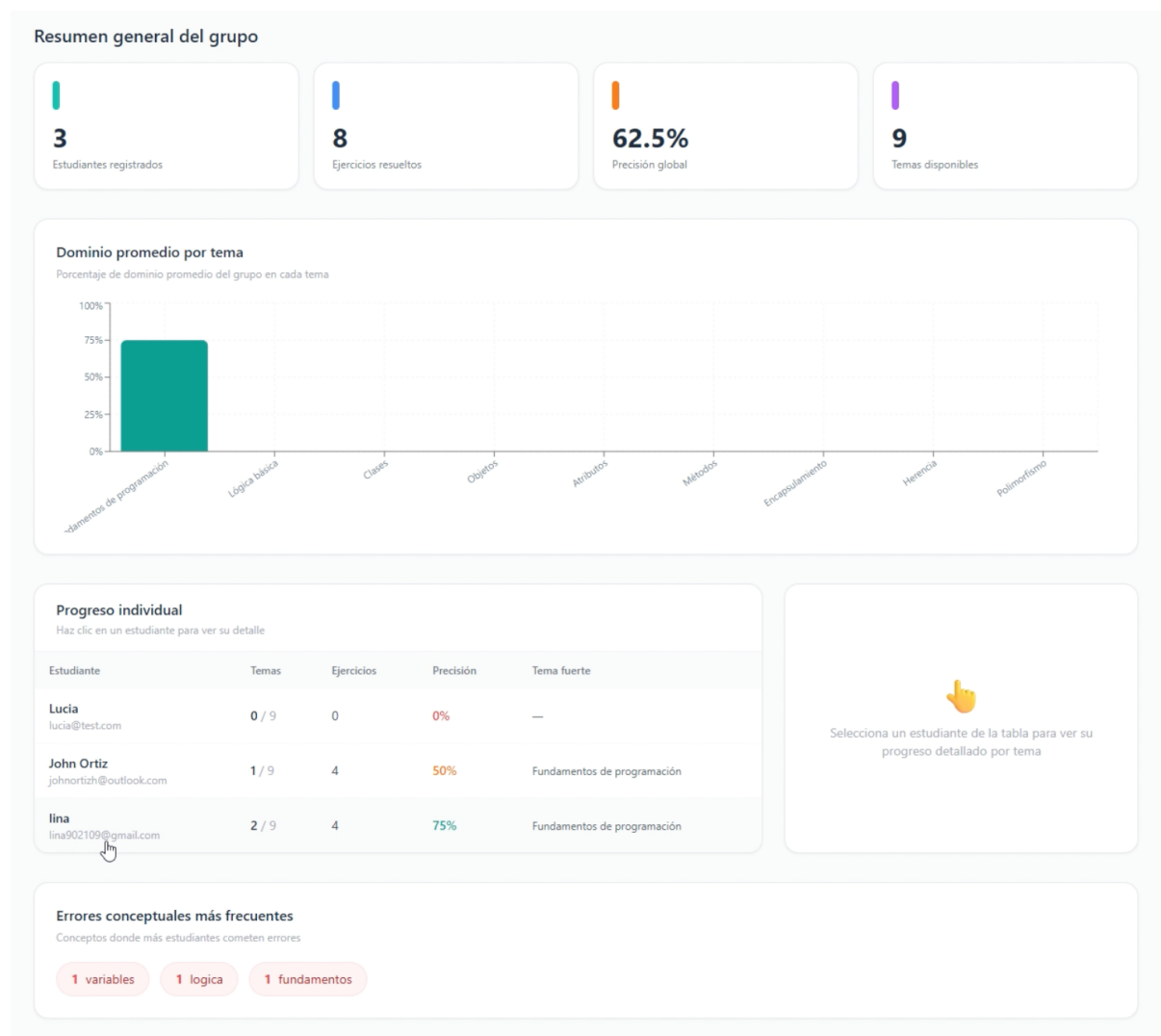


Nota: Imagen del software en funcionamiento.



Nota: Imagen del software en funcionamiento.

Actualización dinámica del perfil: Cada interacción actúa inmediatamente en la ventana deslizante de las últimas 5 respuestas almacenadas en la base de datos. El mecanismo adaptativo recalcula el puntaje del conocimiento recalculando los factores asignados (60% reciente y 40% histórico), y actualiza la interfaz del estudiante sin la necesidad de reenviar la página web, lo cual muestra la rapidez y el funcionamiento preciso del diseño cliente-servidor aplicado.



Nota: Imagen del software en funcionamiento.

Conclusiones

El desarrollo de esta investigación permite presentar una síntesis clara de los principales hallazgos, los cuales responden directamente al problema planteado y a los objetivos específicos definidos desde el inicio del estudio. A partir de los resultados obtenidos, se plantean las siguientes conclusiones:

En primer lugar, se afirma que la plataforma web de tutoría inteligente es técnicamente viable y funcional. A lo largo del proceso se diseñó e implementó una arquitectura cliente-servidor de tres capas, utilizando React, FastAPI y PostgreSQL, que permite ejecutar de manera completa el flujo de aprendizaje adaptativo. Este incluye desde el registro del estudiante y la selección de temas, hasta la generación de ejercicios ajustados al nivel del usuario, la evaluación de sus respuestas, la retroalimentación inmediata y la actualización de su perfil de aprendizaje. En este sentido, se demuestra que es posible construir un tutor inteligente empleando tecnologías accesibles y de código abierto, manteniendo además costos operativos bajos, lo cual responde al objetivo general de la investigación.

En segundo lugar, uno de los hallazgos más relevantes es que la lógica adaptativa puede implementarse sin depender directamente de la inteligencia artificial. El motor desarrollado ajusta la dificultad de los ejercicios mediante un algoritmo determinista basado en el desempeño reciente del estudiante, específicamente a través de una ventana deslizante de sus últimas cinco respuestas. Este enfoque permite obtener un comportamiento más predecible, fácil de supervisar y justificable desde el punto de vista pedagógico. De esta manera, el docente puede entender y ajustar el funcionamiento del sistema sin necesidad de interpretar modelos complejos. La inteligencia artificial, por su parte, se utiliza únicamente para la generación de contenido textual, lo que reduce significativamente los costos de operación y facilita su implementación a mayor escala.

En tercer lugar, el diseño de la plataforma contribuye a contrarrestar el uso inadecuado de

herramientas de inteligencia artificial por parte de los estudiantes. Al requerir que cada ejercicio sea resuelto antes de mostrar la respuesta correcta, y al restringir el acceso anticipado a dicha información, se promueve una participación en el proceso de aprendizaje. Asimismo, la retroalimentación generada no se limita a entregar la solución, sino que orienta al estudiante en la comprensión de sus errores. Esto permite abordar de manera concreta el problema identificado en el estudio: el uso de la inteligencia artificial como un medio para evitar el aprendizaje puede reducirse mediante un rediseño adecuado de la experiencia educativa.

En cuarto lugar, el proyecto presenta un potencial importante como emprendimiento tecnológico escalable. Gracias a su arquitectura modular, la plataforma puede adaptarse a distintos contextos educativos, lenguajes de programación o niveles de formación con ajustes relativamente pequeños. Además, el modelo de negocio basado en software como servicio (SaaS) con suscripción institucional resulta viable desde el punto de vista financiero en etapas tempranas. Esto sugiere que el alcance del proyecto no se limita al ámbito académico, sino que puede generar un impacto real en contextos educativos, especialmente en la mejora de la retención estudiantil en programas de ingeniería.

No obstante, es importante reconocer algunas limitaciones del estudio. En esta fase, la plataforma no fue validada con usuarios reales, lo que impide confirmar con evidencia empírica si el motor adaptativo genera mejoras significativas en el desempeño académico. Esta situación abre una línea clara para trabajos futuros, como la realización de estudios cuasiexperimentales con estudiantes de primeros semestres, que permitan medir su impacto en indicadores como tasas de aprobación y progreso académico.

Finalmente, también se identifican oportunidades de mejora y expansión del sistema. Entre ellas, la integración con plataformas institucionales de gestión del aprendizaje (LMS) y la incorporación de ejercicios con código ejecutable, lo que podría fortalecer aún más su valor pedagógico y su aplicabilidad en entornos reales.

Referencias bibliográficas

Acofi. (2023). La implementación de la inteligencia artificial en la enseñanza de la programación: Un estudio sobre el uso ético de ChatGPT en el aula. Memorias del Encuentro Internacional de Educación en Ingeniería (EIEI).

<https://acofipapers.org/index.php/eiei/article/view/2768/2213>

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. arXiv.

<https://doi.org/10.48550/arXiv.2306.10052>

Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2023). Artificial intelligence in education: Opportunities and challenges. British Journal of Educational Technology.

<https://doi.org/10.1111/bjet.13514>

Paredes, M., Castillo, C., & Ramírez, D. (2023). Uso de la inteligencia artificial en entornos educativos: Una revisión desde la educación superior. Revista de Educación y Aprendizaje, 8(2), 45–60.

<https://revistas.udla.cl/index.php/rea/article/view/400/456>