# 

**Tutor de lenguaje natural para mejorar la compresión de bases fundamentales de la programación en Algoritmos y Programación 1**

# Proyecto de Grado

**Sebastián Hidalgo**

**Juan Sebastián González Sarmiento**

**Santiago Arboleda Velasco**

**Tutor**

**Juan Felipe Aranguren Checa**

**Facultad Barberi de Ingeniería, Diseño y Ciencias Aplicadas**

**Ingeniería de Sistemas**

**Santiago De Cali**

Tabla de contenido

[Proyecto de Grado 1](#_Toc199197714)

[Resumen 4](#_Toc199197715)

[Abstract 5](#_Toc199197716)

[Lista de acrónimos 6](#_Toc199197717)

[Glosario de términos 7](#_Toc199197718)

[Índice de figuras 8](#_Toc199197719)

[Índice de tablas 9](#_Toc199197720)

[1. INTRODUCCIÓN 10](#_Toc199197721)

[1.1 Contexto 10](#_Toc199197722)

[1.2 Planteamiento del problema 11](#_Toc199197723)

[1.3 Impacto del proyecto 12](#_Toc199197724)

[1.4 Objetivo General 13](#_Toc199197725)

[1.5 Objetivos Específicos 14](#_Toc199197727)

[1.6 Organización del Documento 14](#_Toc199197728)

[2. ANTECEDENTES 15](#_Toc199197729)

[2.1. Marco Teórico 15](#_Toc199197730)

[2.2. Estado del arte/trabajos relacionados 18](#_Toc199197731)

[2.3 Estado de la práctica 23](#_Toc199197732)

[3. Metodología 23](#_Toc199197733)

[4. Presentación de la propuesta 23](#_Toc199197734)

[5. Validación y resultados obtenidos 23](#_Toc199197735)

[6. Conclusiones y futuro trabajo 23](#_Toc199197736)

[Anexos 24](#_Toc199197737)

[Referencias bibliográficas 25](#_Toc199197738)

[Herramientas para la revisión bibliográfica 26](#_Toc199197739)

[¿Gestores bibliográficos? 27](#_Toc199197740)

## Resumen

El presente proyecto aborda la problemática identificada en el curso de Algoritmos y Programación 1 de la Universidad Icesi, donde se evidencian dificultades significativas en la comprensión y aplicación de los conceptos fundamentales de la Programación Orientada a Objetos (POO) en etapas tempranas. Tales como funciones, ciclos y polimorfismo. Estas dificultades afectan negativamente el rendimiento académico y la continuidad de los estudiantes en su trayectoria profesional.

Con el fin de mitigar esta problemática, se propone el desarrollo de un tutor inteligente basado en técnicas de inteligencia artificial y procesamiento de lenguaje natural. Diseñado para brindar asistencia personalizada, revisar la sintaxis del código y promover buenas prácticas de programación. La solución se implementará mediante una aplicación web, con miras a integrarse en el ecosistema educativo de la universidad, facilitando su adopción por parte de estudiantes y docentes.

La metodología contempla el análisis de requerimientos, el desarrollo e integración de sistemas expertos y la evaluación de impacto a través de métricas de rendimiento académico. Los entregables incluyen un prototipo funcional del tutor inteligente, informes de pruebas y análisis de resultados, los cuales permitirán validar la efectividad de la herramienta.

Se espera que los resultados obtenidos contribuyan a la mejora en la comprensión de la Programación Orientada a Objetos, fomenten una mayor interacción y retroalimentación oportuna para los estudiantes. Todo esto con el fin de que fortalezcan el proceso de enseñanza aprendizaje en el curso de Algoritmos y Programación 1. En síntesis, este proyecto no solo atiende una necesidad educativa crítica, sino que también impulsa la innovación en las metodologías pedagógicas mediante el uso de tecnologías avanzadas.

**Palabras clave:** Tutor inteligente, Programación Orientada a Objetos, Inteligencia Artificial, Procesamiento de Lenguaje Natural, APO 1.

## Abstract

This project addresses the solution for an intelligent tutoring system, addressing the learning challenges faced by students in the Algorithms and Programming 1 course at Icesi University. Many students struggle to understand and apply fundamental concepts of Object-Oriented Programming (OOP), such as functions, loops, and polymorphism. This impacts their academic performance and retention. The proposed solution integrates artificial intelligence and natural language processing to provide personalized feedback, code review, and programming guidance through a web-based platform. Also includes system requirements analysis, expert system integration, and performance evaluation. The final product is a functional prototype that seeks to improve students' understanding of OOP and support instructors in improving the learning process.

**Keywords:** Intelligent tutoring system, Object Oriented Programming (OOP), Artificial intelligence, Code review, Personalized feedback, Programming education.

## Lista de acrónimos

POO Programación Orientada a Objetos

APO 1 Algoritmos y Programación 1

IA Inteligencia Artificial

PLN Procesamiento de Lenguaje Natural

LLM Large Language Models

RAG Retrieval Augmented Generation

## Glosario de términos

## Índice de figuras

**Página**

Figura 1: Árbol de Problemas 14

Figura 2: Árbol de Objetivos 16

# Índice de tablas

## INTRODUCCIÓN

## 1.1 Contexto

El proyecto se enmarca en el entorno académico de la Universidad Icesi, una institución reconocida por su compromiso con la innovación educativa y el desarrollo de competencias tecnológicas en sus estudiantes de Ingeniería de Sistemas, Ingeniería en Energías inteligentes y Diseño de Medios Interactivos. En este contexto, el curso de Algoritmos y Programación 1 representa la primera experiencia formal de los estudiantes con conceptos de programación, siendo el punto de partida para su formación en la disciplina.

La Universidad Icesi se ubica en una zona urbana de Cali, Colombia, en la que se destaca una creciente inversión en TIC queriendo incluso que la ciudad se convierta en un hub de Inteligencia artificial y transformación digital mediante la inversión gubernamental en centros de IA (MINTIC, 2024). La infraestructura tecnológica y el ecosistema educativo están en constante actualización para responder a las demandas de un mercado laboral cada vez más competitivo. Además, la diversidad cultural y el dinamismo de la comunidad universitaria propician la búsqueda de metodologías pedagógicas innovadoras.

Entre los actores principales se encuentran los estudiantes, quienes deben adaptarse a la enseñanza de conceptos técnicos; los docentes, responsables de transmitir estos conocimientos de manera efectiva; y la administración académica, que busca garantizar la calidad y pertinencia de la formación ofrecida. Estudios recientes en educación superior han mostrado que la adopción de herramientas digitales innovadoras puede mejorar significativamente los procesos de aprendizaje y retención de información. (Guaña-Moya et al., 2024)

Algoritmos y Programación 1 introduce a los estudiantes a la programación, dándoles herramientas básicas como lo son variables, ciclos y pensamiento lógico en el lenguaje de Programación Java. Al igual, les brinda conocimiento en el paradigma de la Programación Orientada a Objetos, lo que implica enfrentarse a conceptos abstractos como clases, objetos, encapsulación, herencia y polimorfismo. Estos conceptos son fundamentales para el desarrollo de competencias en áreas avanzadas de la ingeniería de software, sin embargo, su complejidad genera una curva de aprendizaje pronunciada.

La implementación de un sistema de tutoría digital responde a la imperante necesidad de incorporar herramientas tecnológicas que mejoren el proceso de enseñanza-aprendizaje. La capacidad de proporcionar retroalimentación inmediata, personalizada y adaptativa representa una ventaja competitiva en la formación de ingenieros y diseñadores capaces de enfrentar desafíos tecnológicos modernos. Al facilitar la comprensión de diversos conceptos de la programación, se espera que el tutor inteligente contribuya a mejorar el rendimiento académico de los estudiantes. La personalización del aprendizaje, mediante el análisis de interacciones y la identificación de áreas de dificultad, permitirá intervenciones pedagógicas más precisas y efectivas.

Este proyecto se alinea con las tendencias globales de integración de la IA en la educación, debido a que este tema se ha convertido en un punto de interés muy elevado para muchos (Guo et al., 2021) y puede servir como base para futuras investigaciones y desarrollos en sistemas de tutoría inteligente en otras áreas del conocimiento. La experiencia acumulada y los resultados obtenidos podrán generar nuevas propuestas de mejora en metodologías de enseñanza, en la generación de conocimiento en ingeniería de software o en otras áreas de la universidad. Debido a que, la mejora en la calidad educativa no solo repercute en el ámbito académico, sino que también tiene implicaciones económicas y sociales al formar profesionales mejor capacitados, capaces de responder a las demandas del sector tecnológico y contribuir al desarrollo de la industria.

## Planteamiento del problema

Los estudiantes del curso Algoritmos y Programación 1 de la Universidad Icesi presentan dificultades persistentes en la comprensión de conceptos clave de la Programación Orientada a Objetos. Esto afecta negativamente su rendimiento académico, disminuye su motivación y limita su proyección en áreas avanzadas de ingeniería de software. Resolver este problema mediante un tutor inteligente basado en inteligencia artificial permitirá brindar acompañamiento pedagógico personalizado, promoviendo un aprendizaje más efectivo y reduciendo la brecha que existen que limitan las capacidades actuales de los estudiantes.

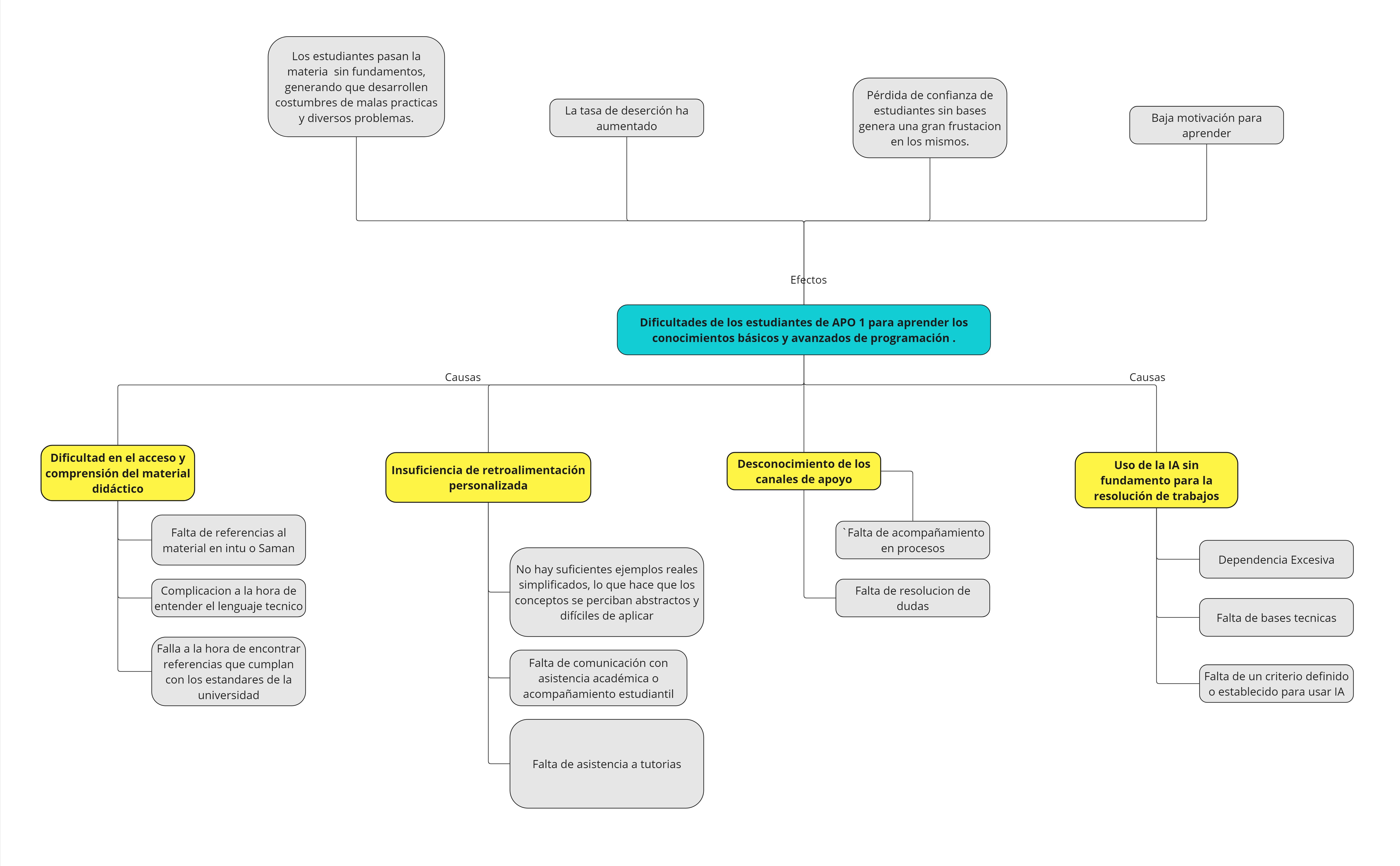


Figura 1: Árbol de Problemas.

## Impacto del proyecto

La implementación de un tutor inteligente mejora la calidad del proceso de enseñanza-aprendizaje, facilitando la comprensión de conceptos complejos y reduciendo la brecha de conocimiento. Esto se traduce en una mayor retención y éxito académico, contribuyendo a la formación de profesionales más competentes y preparados para enfrentar los retos del sector tecnológico. Además, el acceso a una retroalimentación personalizada promueve la equidad educativa, beneficiando a estudiantes con diferentes niveles de preparación previa. El proyecto fomenta la integración de tecnologías avanzadas en la educación, impulsando una transformación cultural en el ámbito académico. La adopción de soluciones basadas en inteligencia artificial incentiva el desarrollo de competencias digitales y promueve una mentalidad orientada a la innovación. Esto contribuye a modernizar los métodos pedagógicos y a generar una cultura de aprendizaje continuo y adaptativo. Debido a que, al mejorar los procesos de enseñanza mediante herramientas tecnológicas, el proyecto puede generar una reducción en los costos asociados a metodologías tradicionales y en la inversión en recursos didácticos. La mejora en la formación de los estudiantes también se traduce en un mercado laboral con profesionales mejor preparados, lo que impulsa el desarrollo económico y la competitividad del sector tecnológico a nivel regional.

Por último, el uso de un tutor inteligente que ofrece asistencia inmediata y adaptada a las necesidades individuales puede reducir los niveles de estrés y ansiedad en los estudiantes. Al proporcionar un entorno de aprendizaje más dinámico y centrado en el estudiante, se favorece un ambiente académico más saludable y se mejora la satisfacción general, impactando positivamente en el bienestar psicológico y social de la comunidad estudiantil.

## Objetivo General

Desarrollar un tutor inteligente basado en Inteligencia Artificial que asista en la comprensión y aplicación de los conceptos fundamentales de la programación en el curso de Algoritmos y Programación 1 de la Universidad Icesi, con el fin de mejorar el rendimiento académico de los estudiantes en un plazo aproximado de 4 meses.

## Diagrama El contenido generado por IA puede ser incorrecto.

Figura 2: Árbol de Objetivos.

## Objetivos Específicos

1. **Optimizar el acceso a material didáctico** proporcionando guías prácticas, glosarios técnicos y una base de conocimientos recomendada por los profesores a través del tutor, facilitando el aprendizaje de la Programación Orientada a Objetos (POO). Se garantizará que el 100% de los estudiantes tenga acceso a estos recursos antes del 30 de noviembre de 2024.
2. **Fortalecer la retroalimentación en el aprendizaje** mediante la implementación de ejemplos simplificados, asistencia con sintaxis y videos específicos, proporcionados por el tutor. Se evaluará la efectividad a través de encuestas de satisfacción y una mejora del 20% en el desempeño de los estudiantes en evaluaciones, con implementación completa antes del 15 de diciembre de 2024.
3. **Mejorar el acceso y difusión de los canales de apoyo** integrando toda la información relevante en la plataforma del tutor para un acceso rápido y eficiente. Se capacitará a los estudiantes en el uso de estos recursos, asegurando que al menos el 90% conozca y utilice los canales de apoyo antes del 1 de diciembre de 2024.
4. **Facilitar la recolección y análisis de datos sobre las necesidades de los estudiantes** mediante el registro de las búsquedas más frecuentes en la plataforma del tutor. Estos datos se presentarán en gráficos y reportes accesibles para los profesores, permitiéndoles identificar áreas de mejora y ajustar sus métodos de enseñanza. La implementación se completará antes del 20 de diciembre de 2024, asegurando que al menos el 80% de los profesores utilicen la información para mejorar sus clases.

## Organización del Documento

## ANTECEDENTES

Diversas investigaciones (Diaz-Leyva, T., & Chamorro-Atalaya, O., 2020) y análisis sectoriales han evidenciado que el bajo rendimiento en APO 1 se relaciona directamente con la dificultad de los estudiantes para internalizar los conceptos básicos de POO. Entre los antecedentes más relevantes se destacan:

* **Falta de experiencia previa:** Muchos estudiantes ingresan al curso sin haber tenido contacto previo con la programación, lo que limita su capacidad para asimilar conceptos abstractos y dificulta la transición hacia paradigmas más complejos.
* **Desafíos en la abstracción y comprensión:** Diversos estudiantes no logran entender estructuras básicas como condicionales y bucles. Por lo que el salto a conceptos como clases, objetos, encapsulación y la herencia resulta particularmente desafiante.
* **Limitaciones en los recursos didácticos y metodologías de enseñanza:** La carencia de material didáctico adaptado a los distintos estilos de aprendizaje y la utilización de metodologías de enseñanza tradicionales han contribuido a que la brecha de conocimiento persista, afectando el rendimiento y la motivación de los estudiantes.

Los antecedentes descritos revelan que el problema no es aislado, sino el resultado de un conjunto de factores interrelacionados que afectan la calidad del aprendizaje en APO 1, lo que justifica la búsqueda de soluciones innovadoras que permitan superar estas barreras. La problemática abordada se deriva directamente del contexto académico y tecnológico de la Universidad Icesi, en donde se ha observado que el curso de Algoritmos y Programación 1 presenta dificultades significativas en la asimilación de conceptos fundamentales de la programación. Tales como funciones, ciclos, y POO. Estas dificultades afectan el rendimiento académico, la preparación de los estudiantes para cursos avanzados y su vida laboral.

## Marco Teórico

El proceso de aprendizaje de la Programación Orientada a Objetos (POO) representa un reto en la educación superior, especialmente en cursos introductorios como Algoritmos y Programación 1 en la Universidad Icesi. Conceptos básicos como ciclos, variables y métodos. Hasta encapsulamiento, herencia, polimorfismo y abstracción suelen presentar un nivel alto de complejidad para los estudiantes, lo que genera dificultades tanto en el rendimiento académico como en la retención de estos en la carrera. Para abordar este problema, este proyecto propone el desarrollo de un tutor inteligente que brinde asistencia personalizada en temas de POO, haciendo uso de técnicas de Inteligencia Artificial (IA) y Procesamiento de Lenguaje Natural (PLN)

* **Inteligencia artificial en la educación**

La Inteligencia Artificial ha evolucionado significativamente en las últimas décadas, permitiendo automatizar procesos cognitivos mediante algoritmos que aprenden y toman decisiones. En el ámbito educativo, la IA ha demostrado ser efectiva para personalizar la experiencia de aprendizaje, identificar errores comunes en el razonamiento de los estudiantes y generar recomendaciones individualizadas. Diversos estudios han evidenciado la eficacia de la IA en la educación, demostrando que la personalización y la retroalimentación inmediata pueden aumentar la motivación y mejorar el rendimiento académico (Yaseen et al., 2025).

Aplicaciones como los sistemas expertos o los modelos de machine learning permiten modelar perfiles de estudiantes, predecir dificultades y adaptar los contenidos según el progreso del usuario. Este enfoque resulta clave para el primer objetivo específico del proyecto, que busca optimizar el acceso al material didáctico de forma personalizada y efectiva. Por ejemplo, se ha demostrado que algoritmos de aprendizaje automático pueden analizar los datos de un alumno para predecir su rendimiento o detectar tempranamente qué estudiantes tendrán dificultades, lo que permite a los docentes intervenir a tiempo y prevenir deserciones (Lin et al., 2023). En entornos virtuales de aprendizaje, técnicas de deep learning logran más de un 90% de acierto al predecir el desempeño estudiantil, identificando fortalezas y debilidades (Alnasyan et al., 2024). Con ello, el sistema puede personalizar las intervenciones y contenidos para apoyar las necesidades específicas de cada alumno.

La incorporación de tecnologías educativas en universidades permite no solo ampliar el acceso a los contenidos, sino también mejorar el monitoreo del proceso de aprendizaje. Estudios previos en educación tecnológica y publicaciones indexadas en revistas de IEEE y ACM proporcionan un marco teórico robusto que respalda la aplicación de estas tecnologías en contextos educativos similares al de la Universidad Icesi (Guo et al., 2021). Estos estudios destacan que plataformas interactivas mejoran la eficiencia de la enseñanza y fomentan el aprendizaje activo, siempre que estén alineadas con las necesidades curriculares y pedagógicas. En el caso del tutor inteligente propuesto, la integración con el entorno tecnológico de la universidad facilitará su adopción, permitiendo que los estudiantes accedan fácilmente a recursos clave, reciban ayuda en tiempo real y los docentes puedan monitorear los avances mediante reportes personalizados, como lo plantea el cuarto objetivo del proyecto.

* **Procesamiento de Lenguaje Natural (PLN)**

El PLN es una rama de la IA que permite a las máquinas interpretar y producir lenguaje humano, siendo fundamental en aplicaciones que requieren interacción con el usuario, como asistentes virtuales o tutores inteligentes. En el contexto educativo, el PLN puede analizar preguntas escritas por estudiantes, identificar intenciones, detectar errores semánticos o sintácticos en el código y ofrecer respuestas comprensibles. Esta capacidad es esencial para el segundo objetivo específico del proyecto, orientado a proporcionar retroalimentación contextualizada mediante la revisión automática de código y la asistencia en tiempo real.

Entre las técnicas más utilizadas en PLN están los modelos de clasificación de texto, análisis de dependencias gramaticales y embeddings de palabras como Word2Vec o BERT. Estos modelos han sido aplicados exitosamente en proyectos como CodeBERT o DeepCode, que integran PLN para la comprensión de código fuente. (Banjade et al., 2022)

* **Sistemas de Tutoría Inteligente (STI)**

Los STI combinan IA y principios pedagógicos para ofrecer acompañamiento individualizado a los estudiantes. Estos sistemas han sido implementados con éxito en múltiples disciplinas, proporcionando retroalimentación inmediata, recomendaciones personalizadas y análisis de desempeño. Investigaciones en sistemas de tutoría inteligente han documentado mejoras en la retención de conocimientos y en la disminución de la curva de aprendizaje en áreas complejas como la programación (Lai & Lin, 2025).

Estos sistemas suelen incorporar modelos del estudiante, del contenido y de la tutoría. Lo que permite adaptar la enseñanza al ritmo y estilo de aprendizaje de cada uno. Estos sistemas son fundamentales para el tercer y cuarto objetivo específico del proyecto, donde se busca mejorar el acceso a canales de apoyo y recoger datos que permitan retroalimentar a los docentes. (Murphy, 2019)

* **Fundamentos Pedagógicos: Constructivismo y Andamiaje**

El desarrollo del tutor inteligente también se apoya en teorías del aprendizaje ampliamente validadas. El constructivismo, promovido por autores como Piaget y Vygotsky, sostiene que los estudiantes construyen activamente su conocimiento a través de la experiencia y la interacción con su entorno. Este enfoque se ve potenciado por el uso de herramientas que permiten experimentar con código, recibir retroalimentación inmediata y corregir errores de forma autónoma.

Por otro lado, la teoría del andamiaje, desarrollada por Bruner, plantea que el aprendizaje mejora cuando se brinda apoyo progresivo, retirado gradualmente a medida que el estudiante gana competencia. Esta lógica es central en el diseño del tutor, el cual proporcionará ejemplos guiados, asistencia paso a paso y, eventualmente, desafíos más complejos que fomenten la autonomía.

## Estado del arte/trabajos relacionados

El auge de los modelos de lenguaje de gran escala (LLMs) ha impulsado el desarrollo de nuevas herramientas capaces de asistir a los estudiantes en diversos contextos, incluyendo el aprendizaje de programación. Soluciones como las Interfaces de Programación de Aplicaciones (APIs) de ChatGPT, Gemini y DeepSeek ofrecen capacidades avanzadas de comprensión del lenguaje natural, generación de texto y análisis de código, siendo consideradas por docentes universitarios como tecnologías prometedoras. Sin embargo, una revisión crítica de estas tecnologías es necesaria para establecer sus potencialidades y limitaciones frente al desarrollo de un tutor inteligente adaptado específicamente al curso Algoritmos y Programación 1 (APO 1) en la Universidad Icesi.

* **API de ChatGPT (OpenAI)**

Permite acceder a diversos modelos de OpenAI como GPT-3.5-Turbo y la familia GPT-4, entrenados sobre vastos volúmenes de texto, código y documentación técnica. Su uso facilita la construcción de asistentes conversacionales que entienden y generan lenguaje natural, resuelven dudas, explican conceptos e incluso pueden ayudar a corregir y optimizar código.

* **Ventajas:**
  + **Excelente comprensión del lenguaje natural y capacidad conversacional:** Ideal para asistir a estudiantes en la formulación de sus dudas de manera intuitiva.
  + **Habilidad para generar explicaciones detalladas y guías:** Puede adaptar sus explicaciones si se le provee el contexto adecuado (mediante *prompt engineering*), pudiendo orientar a los estudiantes sin ofrecer soluciones directas.
  + **Amplia documentación y comunidad activa:** Cuenta con una vasta cantidad de recursos, tutoriales y una comunidad de desarrolladores madura, lo que facilita su integración y la resolución de problemas.
  + **Flexibilidad y mejora continua:** Los modelos son actualizados y mejorados constantemente por OpenAI, ofreciendo capacidades crecientes.
* **Desventajas:**
  + **Costo y naturaleza propietaria:** Es un servicio comercial y su uso incurre en costos económicos basados en el consumo (tokens), lo cual puede ser una limitación para la sostenibilidad en proyectos educativos a largo plazo o de uso intensivo. No es de código abierto.
  + **Requiere alineación pedagógica explícita:** No está intrínsecamente alineada con teorías pedagógicas como el constructivismo o el andamiaje. Se necesita un diseño de *prompts* cuidadoso y, a menudo, complejo para que sus respuestas sirvan a objetivos educativos específicos.
  + **Retroalimentación potencialmente genérica:** Aunque puede ofrecer retroalimentación sobre el código, esta puede ser genérica o incluso errónea si no se le estructura un contexto detallado del estudiante (nivel, errores previos, progreso en el curso) y de los objetivos de aprendizaje específicos de APO 1.
  + **Posibles "alucinaciones" o información incorrecta:** Como todos los LLMs, puede generar información que parezca plausible pero sea incorrecta, lo que requiere validación en un contexto educativo.
* **API de Gemini (Google)**

La API de Gemini, desarrollada por Google AI, provee acceso a la familia de modelos multimodales Gemini (ej. Gemini Pro 2.0, Gemini Flash 2.0, y potencialmente versiones más avanzadas). Estos modelos están diseñados para comprender, razonar y generar texto, código, y procesar diversos tipos de información, destacando por sus capacidades de razonamiento complejo y multimodalidad.

**• Ventajas:**

* **Fuertes capacidades de razonamiento y multimodalidad:** Puede comprender contextos complejos, lo que es útil para explicar conceptos algorítmicos y estructuras de datos. Su capacidad para procesar imágenes o audio (aunque la API de texto sea la principal para este caso) podría abrir futuras posibilidades.
* **Buen rendimiento en tareas de generación y explicación de código:** Muestra competencia en diversos lenguajes de programación relevantes para APO 1.
* **Escalabilidad y ecosistema Google:** Se beneficia de la infraestructura de Google, asegurando buena escalabilidad y fiabilidad. Potencial de integración con otras herramientas del ecosistema Google.
* **Niveles de uso gratuitos o créditos:** Google suele ofrecer un nivel de uso gratuito o créditos iniciales que facilitan la experimentación y el desarrollo temprano de prototipos.

**• Desventajas:**

* **Costo en uso extensivo:** Al igual que ChatGPT, el uso más allá de las cuotas gratuitas es un servicio propietario con costos asociados que pueden escalar con el número de estudiantes y la frecuencia de interacción.
* **Alineación pedagógica dependiente del diseño:** Aunque potente, su alineación con estrategias pedagógicas específicas para la enseñanza de la programación requiere un diseño instruccional y de prompts elaborado por los desarrolladores del tutor.
* **Novedad relativa:** Aunque evoluciona rápidamente, la cantidad de recursos de la comunidad, ejemplos específicos y bibliotecas de terceros podría ser menor en comparación con la API de OpenAI, aunque esta brecha se cierra continuamente.
* **Personalización de la retroalimentación:** Sigue siendo un desafío que depende de la lógica externa implementada para que sea verdaderamente adaptativa al progreso individual del estudiante en APO 1.
* **API de DeepSeek (DeepSeek AI)**

La API de DeepSeek proporciona acceso a modelos de lenguaje especializados, particularmente aquellos de la familia DeepSeek Coder, que han sido entrenados con un enfoque intensivo en grandes volúmenes de código fuente (más de 2 billones de tokens de código) y material técnico relacionado con la programación y las matemáticas.

**Ventajas:**

* **Alta especialización en código:** Su entrenamiento focalizado en código lo hace particularmente apto para la comprensión, generación, explicación y depuración de programas, lo cual es directamente aplicable a los desafíos de aprendizaje en APO 1.
* **Potencial de mayor precisión en tareas de programación:** Podría ofrecer una mayor precisión y relevancia en tareas específicas de programación (por ejemplo, identificar errores sutiles, explicar algoritmos complejos, o traducir fragmentos de código) en comparación con modelos más generalistas.
* **Modelos base con componentes de código abierto:** Aunque la API es un servicio, el hecho de que algunos de sus modelos base (como DeepSeek Coder) sean de código abierto ofrece una mayor transparencia sobre su arquitectura y datos de entrenamiento.
* **Costos potencialmente competitivos:** Dado su enfoque especializado, podría ofrecer una estructura de costos más ventajosa para tareas intensivas en código.

**Desventajas:**

* **Capacidades conversacionales generales:** Al estar más especializado en código, su habilidad para la conversación general o la comprensión de matices en lenguaje natural no técnico podría ser menos robusta que la de modelos como GPT-4 o Gemini. Esto podría afectar la naturalidad de la interacción si el estudiante se desvía de temas puramente técnicos.
* **Comunidad y documentación más recientes:** Siendo un actor más nuevo en el mercado de APIs de LLMs en comparación con OpenAI o Google, la documentación, herramientas de soporte y comunidad de desarrolladores pueden ser más limitadas, aunque están en crecimiento.
* **Necesidad de integración pedagógica:** La especialización en código no exime de la necesidad de diseñar una interacción pedagógica. El modelo puede explicar código, pero no necesariamente enseñar de manera efectiva sin una guía instruccional y adaptativa.
* **Ollama**

Es una plataforma que facilita la ejecución local de modelos de lenguaje como LLaMA 2, Mistral y otros modelos de código abierto. Permite ejecutar modelos de lenguaje de gran escala en dispositivos personales, sin necesidad de conexión a servidores externos, ofreciendo privacidad y control.

* + **Ventajas:** 
    - Permite usar modelos potentes de manera local, lo cual es ideal para instituciones que requieren control de datos y privacidad.
    - Reduce costos operativos frente a soluciones basadas en nube.
    - Es compatible con múltiples modelos de lenguaje, siendo fácil de instalar.
  + **Desventajas:** 
    - Su enfoque está en la ejecución local del modelo, pero no incluye funciones pedagógicas por defecto.
    - Requiere configuración técnica y recursos de hardware considerables.

Es valioso como herramienta para ejecutar tutores de forma local y proteger la privacidad de los estudiantes. Su uso requeriría una arquitectura adicional que implemente los modelos de tutoría, interacción y contenido requeridos en APO 1.

Por estas razones, el desarrollo de un tutor inteligente propio, que aproveche estas tecnologías, pero construya sobre ellas un sistema pedagógico personalizado, se presenta como una necesidad válida y justificada dentro del estado del arte actual.

* **Comparación de Soluciones Existentes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Proyecto** | **Enfoque Principal** | **Tecnología** | **Aporte Clave** |
| Teaching Robots through Simulated Student Feedback | Entrenamiento de agentes docentes con feedback simulado de estudiantes. | Simulación de retroalimentación, aprendizaje reforzado, modelado. | Técnica para entrenar agentes educativos con retroalimentación sintética. |
| Exploring LLMs as Autonomous Tutors | Evaluación de LLMs como tutores autónomos en ambientes reales. | GPT-4, retroalimentación automática, razonamiento. | LLMs manejan sesiones completas de tutoría. |
| Personalized Feedback using NLP in ITS | Generación de retroalimentación personalizada con NLP. | Procesamiento de lenguaje natural (NLP) para análisis de respuestas. | Feedback adaptado al rendimiento y estilo del estudiante. |

## Estado de la práctica

## 3. Metodología

El desarrollo del proyecto se estructura bajo el enfoque ágil de Scrum, una metodología ampliamente utilizada en la industria del software que permite una gestión iterativa e incremental del desarrollo. Esta elección responde a la necesidad de entregar valor de forma continua, adaptarse a los cambios y mantener una comunicación constante entre los miembros del equipo. La herramienta seleccionada para la implementación de esta metodología es Jira Software, que facilita la planificación, ejecución y seguimiento de las tareas mediante tableros, historias de usuario y reportes de avance.

El proyecto se desarrollará en cinco sprints, cada uno con objetivos concretos que responden directamente al cumplimiento de los objetivos específicos planteados. Cada fase emplea técnicas específicas para garantizar calidad y eficiencia, tales como: diseño centrado en el usuario, integración continua, revisión por pares, prototipado en Figma, análisis de métricas y pruebas funcionales con usuarios representativos. A continuación, se describen las fases del proyecto, su duración estimada y las técnicas empleadas:

**Fase 1: Diseño preliminar y configuración inicial**

**Duración:** 27 de mayo – 12 de junio

* Implementar el sistema de login y registro de usuarios.
* Definir la estructura visual y navegación de la plataforma.
* Diseño de wireframes y mockups en Figma.
* Definición de paleta de colores, tipografía y jerarquía visual.
* Configuración del repositorio, entornos de desarrollo y primer modelo de base de datos.
* Implementación del sistema de autenticación en Django.

**Fase 2: Desarrollo funcional - núcleo del sistema**

**Duración:** 24 de junio – 29 de julio

* Iniciar el desarrollo de las funcionalidades pedagógicas del tutor inteligente.
* Implementación de las funciones del chatbot educativo.
* Integración inicial con la API de ChatGPT.
* Desarrollo del historial de conversaciones y visualización personalizada por rol.
* Avance en la arquitectura de microservicios y manejo de datos con PostgreSQL.

**Fase 3: Expansión funcional**

**Duración:** 5 de agosto – 2 de septiembre

* Continuar con la integración de funcionalidades del sistema de tutoría.
* Construcción de componentes de retroalimentación y reportes.
* Implementación de métricas visuales para docentes.
* Sistema de clasificación de errores de código y retroalimentación guiada.
* Integración de material didáctico y herramientas de apoyo.

**Fase 4: Ajustes finales y validaciones funcionales**

**Duración:** 9 de septiembre – 7 de octubre

* Completar módulos pendientes y comenzar validación de casos de uso.
* Ajustes sobre módulos incompletos.
* Pruebas funcionales con usuarios controlados.
* Recolección de métricas de aprendizaje y rendimiento.

**Fase 5: Pruebas, documentación y despliegue**

**Duración:** 14 de octubre – 11 de noviembre

* Finalizar el proyecto con pruebas formales, documentación técnica y despliegue en entorno productivo.
* Pruebas de usabilidad (técnica: test de usuario).
* Documentación de endpoints, arquitectura y proceso de desarrollo.
* Despliegue en la nube mediante Render y verificación de disponibilidad.
* Validación final de requerimientos funcionales y no funcionales.

## Presentación de la propuesta

## Validación y resultados obtenidos

## Conclusiones y futuro trabajo

## Anexos

## Referencias bibliográficas

* Guaña-Moya, J., Arteaga-Alcívar, Y., Criollo-C, S., & Cajamarca-Carrazco, D. (2024). Use of Interactive Technologies to Increase Motivation in University Online Courses. *Education Sciences*, *14*(12), 1406. <https://doi.org/10.3390/educsci14121406>
* Diaz-Leyva, T., & Chamorro-Atalaya, O. (2020). Analysis of Learning Difficulties in Object Oriented Programming in Systems Engineering Students at UNTELS. *Advances in Science, Technology and Engineering Systems Journal*, *5*(6), 1704–1709. <https://doi.org/10.25046/aj0506203>
* MINTIC (2024). *“Queremos que Cali se convierta en un hub de Inteligencia Artificial”: Ministro Lizcano*. [https://www.mintic.gov.co/portal/inicio/Sala-de-prensa/Noticias/334142:Queremos-que-Cali-se-convierta-en-un-hub-de-Inteligencia-Artificial-Ministro-Lizcano#:~:text=impulsando%20as%C3%AD%20el%20ecosistema%20de,millones%2C%20se%20instalar%C3%A1n%20antenas%20de](https://www.mintic.gov.co/portal/inicio/Sala-de-prensa/Noticias/334142:Queremos-que-Cali-se-convierta-en-un-hub-de-Inteligencia-Artificial-Ministro-Lizcano%23:~:text=impulsando%20as%C3%AD%20el%20ecosistema%20de,millones%2C%20se%20instalar%C3%A1n%20antenas%20de)
* Guo, L., Wang, D., Gu, F., Li, Y., Wang, Y., & Zhou, R. (2021). Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, *22*(3), 441–461. <https://doi.org/10.1007/s12564-021-09697-7>
* Yaseen, H., Mohammad, A. S., Ashal, N., Abusaimeh, H., Ali, A., & Sharabati, A.-A. A. (2025). The Impact of Adaptive Learning Technologies, Personalized Feedback, and Interactive AI Tools on Student Engagement: The Moderating Role of Digital Literacy. *Sustainability*, *17*(3), 1133. <https://doi.org/10.3390/su17031133>
* Lai, C.-H., & Lin, C.-Y. (2025). Analysis of Learning Behaviors and Outcomes for Students with Different Knowledge Levels: A Case Study of Intelligent Tutoring System for Coding and Learning (ITS-CAL). *Applied Sciences*, *15*(4), 1922. <https://doi.org/10.3390/app15041922>
* Lin, C.-C., Huang, A. Y. Q., & Lu, O. H. T. (2023). Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, *10*(1), 41. <https://doi.org/10.1186/s40561-023-00260-y>
* Alnasyan, B., Basheri, M., & Alassafi, M. (2024). The power of Deep Learning techniques for predicting student performance in Virtual Learning Environments: A systematic literature review. [https://doaj.org/article/2dd814986b084cb6bfd757ac54170564#:~:text=their%20success,failure%20and%20prevent%20their%20dropout](https://doaj.org/article/2dd814986b084cb6bfd757ac54170564%23:~:text=their%20success,failure%20and%20prevent%20their%20dropout)
* Banjade, R., Oli, P., Tamang, L. J., & Rus, V. (2022). Preliminary Experiments with Transformer based Approaches To Automatically Inferring Domain Models from Textbooks. *Zenodo (CERN European Organization For Nuclear Research)*. <https://doi.org/10.5281/zenodo.6853051>
* Murphy, R. F. (2019). *Perspective Expert insights on a timely policy issue Artificial Intelligence Applications to Support K-12 Teachers and Teaching A Review of Promising Applications, Opportunities, and Challenges*. [https://mvwsd.novusagenda.com/AgendaPublic/AttachmentViewer.ashx?AttachmentID=6267&ItemID=4542#:~:text=A%20typical%20ITS%20architecture%20comprises,domain%20can%20be%20reduced%20to](https://mvwsd.novusagenda.com/AgendaPublic/AttachmentViewer.ashx?AttachmentID=6267&ItemID=4542%23:~:text=A%20typical%20ITS%20architecture%20comprises,domain%20can%20be%20reduced%20to)

### Herramientas para la revisión bibliográfica

1. <https://parsif.al/>
2. ScientoPy:  <https://github.com/jpruiz84/ScientoPy>,
3. <https://www.youtube.com/watch?v=rwYO6XVY-fI>.
4. SciMAT:  <https://sci2s.ugr.es/scimat/>

## ¿Gestores bibliográficos?

Mendeley