

Using intelligent tutors based on Generative AI to strengthen programming skills in STEM and Non-STEM profiles

Kevin Mejía-Parra[✉]; Brayan Briones-Oleas[✉]; Luis Borja-Zevallos[✉];
Cristopher Villón-Loor[✉]; Rodrigo Saraguro-Bravo[✉]

ESPOL Polytechnic University, Guayaquil, Ecuador

kesameji@espol.edu.ec, bsbrione@espol.edu.ec, laborja@espol.edu.ec, chrovill@espol.edu.ec, rasarag@espol.edu.ec

Abstract– *The integration of Large Language Models into conversational tutoring systems can lower entry barriers in introductory programming courses with heterogeneous student profiles. This study evaluates the impact of such systems on performance and interaction, comparing students from STEM and non-STEM backgrounds. A quasi-experimental design (N=30) was implemented, comprising a control group (standard documentation) and an experimental group (intelligent tutor). Performance was assessed through five Python activities validated by unit tests, using a normalized score (0–1). The group utilizing TutorIA demonstrated an improvement, with average scores increasing from 0.43 to 0.78. Although STEM students maintained higher overall grades, non-STEM students achieved higher prompt quality scores (6.33/8.0 vs. 5.98/8.0), suggesting superior semantic contextualization. These findings indicate that TutorIA facilitated a leveling of the learning curve; however, both cohorts require distinct support mechanisms: logical reinforcement for non-STEM students and enhanced interaction strategies with AI assistants for STEM students.*

Keywords– *Education, Programming, STEM, Intelligent Tutor, Prompt Engineering*

Uso de tutores inteligentes basados en IA Generativa para el fortalecimiento de competencias de programación en perfiles STEM y No-STEM

Kevin Mejía-Parra[✉]; Brayan Briones-Oleas[✉]; Luis Borja-Zevallos[✉];
Cristopher Villón-Loor[✉]; Rodrigo Saraguro-Bravo[✉]

Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador

kesameji@espol.edu.ec, bsbrione@espol.edu.ec, laborja@espol.edu.ec, chrovill@espol.edu.ec, rasarag@espol.edu.ec

Resumen– La incorporación de Modelos de Lenguaje de Gran Escala para la creación de tutores conversacionales puede reducir barreras en cursos introductorios de programación con perfiles heterogéneos. Este estudio evalúa su impacto en el desempeño y la interacción, comparando estudiantes de carreras STEM y No-STEM. Se aplicó un diseño cuasiexperimental con N=30, con grupo control (documentación) y grupo experimental (tutor inteligente). El rendimiento se midió mediante cinco actividades en Python validados por pruebas unitarias (puntaje normalizado 0–1); reflejando que el grupo que utilizó TutorIA incrementó el promedio de 0.43 a 0.78; aunque el grupo STEM mantuvo mejores calificaciones globales, los estudiantes No-STEM obtuvieron mayor calidad de prompts (6.33/8.0 vs. 5.98/8.0), sugiriendo mejor contextualización semántica. Los hallazgos indican que utilizando TutorIA fue posible nivelar el aprendizaje, pero ambos grupos requieren apoyos diferentes: refuerzo lógico para No-STEM y comunicación con los asistentes para STEM.

Palabras clave– Educación, Programación, STEM, No-STEM, Tutor Inteligente, Prompts

I. INTRODUCCIÓN

La evolución de los Modelos de Lenguaje de Gran Escala (LLM) está redefiniendo el panorama educativo, abriendo vías inéditas para personalizar la enseñanza y automatizar la tutoría académica [1], [2]. En la actualidad, las herramientas sustentadas en Inteligencia Artificial Generativa (IAG), tales como ChatGPT o Gemini, operan como asistentes virtuales competentes para brindar retroalimentación instantánea, orientar en la resolución de actividades y ajustar sus respuestas al nivel de competencia del alumno [3], [4]. Dicha tecnología facilita la simulación del seguimiento propio de un tutor humano, fomentando así ecosistemas de aprendizaje más interactivos y autónomos, lo cual resulta crítico en asignaturas de fundamentos de programación [5], [6].

Si bien la programación es una disciplina angular en la formación universitaria, esta suele imponer barreras cognitivas diferenciadas entre estudiantes de ramas tecnológicas y aquellos de carreras no afines [7], [8]. Tal disparidad impulsa la necesidad de implementar estrategias pedagógicas que optimicen la comprensión y disminuyan la fricción inicial en cursos introductorios para audiencias heterogéneas. Por consiguiente, la implementación de tutores inteligentes basados

en LLM surge como una solución viable para potenciar la autoeficacia, el compromiso y el desempeño académico del estudiantado [9], [10].

La evidencia reciente sugiere que estos tutores proporcionan un acompañamiento adaptable y situado, integrándose fluidamente en el ámbito universitario y promoviendo una educación más equitativa y autodirigida [11], [12]. No obstante, la literatura también advierte sobre ciertas limitaciones, tales como la exactitud de la información generada, el riesgo de dependencia tecnológica y la urgencia de validar su efectividad a través de distintos perfiles académicos [13], [14].

Estas carencias fundamentan la presente investigación, la cual se orienta a evaluar el impacto de los tutores de programación basados en IAG comparando estudiantes de carreras STEM y No-STEM. A través de este análisis comparativo, el estudio busca generar datos empíricos que orienten su adopción responsable y contrastar el nivel de progreso derivado de la mediación tecnológica en distintas disciplinas. Se pretende esclarecer cómo la frecuencia y la modalidad de interacción con el asistente virtual inciden en la eficiencia, la calidad del código desarrollado y la percepción de autoeficacia [15], [16]. Basándose en dicho análisis, se formulan recomendaciones metodológicas y protocolos de uso ético que coadyuven a maximizar el valor pedagógico de estas herramientas en escenarios universitarios diversos.

La contribución central de este manuscrito reside en aportar evidencia empírica sobre la eficacia relativa de los tutores de IAG en diferentes perfiles estudiantiles, además de proponer una guía práctica de implementación ajustable a contextos con distintos niveles de madurez tecnológica [17], [18]. De este modo, se profundiza en el entendimiento sobre la utilidad de los LLM como soporte educativo y se establecen criterios para su integración ética en la didáctica de la programación.

II. REVISIÓN DE LITERATURA

Múltiples investigaciones recientes indican un consenso en que los sistemas basados en LLM ofrecen retroalimentación al instante, explicaciones a medida y guía contextual, elementos

determinantes para fomentar un aprendizaje significativo y autónomo [1], [5]. Sin embargo, se subraya simultáneamente la importancia de un diseño pedagógico riguroso que regule la carga cognitiva y estimule la capacidad de autorregulación del alumno [2], [6].

A. Efectos sobre el aprendizaje y la autoeficacia

La adopción de herramientas de IAG se vincula positivamente con una mejor comprensión conceptual, mayor motivación y un incremento en la autoeficacia percibida. Yilmaz y Karaoglan-Yilmaz indican que los estudiantes asistidos por tutores de IA exhiben niveles superiores de confianza y persistencia ante problemas de programación [9]. La disponibilidad constante de estas herramientas mitiga la frustración y facilita una práctica guiada, aspecto crucial en las fases tempranas del aprendizaje [10], [12]. Asimismo, estudios longitudinales confirman que las ganancias en autoeficacia perduran en el tiempo, beneficiando especialmente a aquellos con escasa experiencia previa [8].

B. Diseño instruccional y retroalimentación adaptativa

La configuración de tutores basados en LLM demanda estrategias instruccionales que incorporen andamiaje, progresión en la dificultad y mecanismos de validación. Reicher et al. [5] enfatizan que un tutor eficaz debe suministrar explicaciones graduales e incentivar la reflexión sobre los fallos, evitando proveer la solución final de inmediato. En esta línea, Phung et al. [6] sugieren sistemas de retroalimentación automatizada que emulen la tutoría humana a través de pistas escalonadas y evaluación mediante pruebas unitarias. Por su parte, la ingeniería de *prompts* se perfila como un factor decisivo para maximizar la calidad de las interacciones [15], mientras que García y Martínez [17] resaltan la relevancia del andamiaje conversacional para edificar el conocimiento progresivamente.

C. Integridad académica y uso ético

Un desafío recurrente en la literatura es la preservación de la integridad académica. Kasneci et al. [1] alertan sobre los peligros de una dependencia desmedida y el empleo no autorizado de la IAG para la resolución de tareas. De igual manera, Anderson y Kim [13] examinan la perspectiva docente frente a la IA educativa, subrayando inquietudes relativas al plagio, la equidad y la justicia evaluativa. En consecuencia, se aconseja la instauración de directrices éticas y modelos de evaluación enfocados en el proceso de aprendizaje más que en el producto final [3]. La transparencia y la trazabilidad de las consultas se posicionan como estrategias fundamentales para garantizar un uso responsable [19].

D. Personalización y equidad entre perfiles estudiantiles

La capacidad adaptativa de los LLM facilita la personalización de la enseñanza según las necesidades individuales. Kovalchuk et al. [11] destacan que estos sistemas contribuyen a reducir la brecha entre estudiantes de ramas

STEM y No-STEM al ajustar el acompañamiento al nivel de conocimientos previos. Rodríguez et al. [7] evidencian, en un estudio con 200 participantes, que los alumnos de perfiles no técnicos logran mejoras del 40% en sus calificaciones al usar tutores adaptativos, frente a un 25% en estudiantes técnicos. Complementariamente, Wuy Zhang [18] proponen sistemas de recomendación basados en LLM que diseñan rutas de aprendizaje a medida del progreso del usuario.

E. Evaluación de la calidad del código y habilidades de depuración

La aptitud de los LLM para examinar código y ofrecer retroalimentación específica constituye una ventaja notable en la didáctica de la programación. Thompson y Silva [14] analizan cómo estos modelos reconfiguran el proceso de depuración de código, capacitando a los estudiantes para detectar y rectificar errores con mayor eficiencia. Por otro lado, Johnson y Brown [16] establecen métricas objetivas para valorar la calidad del código generado con asistencia de IAG, hallando mejoras sustanciales en legibilidad y eficiencia algorítmica. Tales resultados sugieren que los tutores inteligentes no solo asisten en la resolución puntual de problemas, sino que fomentan la adopción de buenas prácticas de desarrollo.

E. Propuesta de investigación

Pese a los progresos, la literatura muestra lagunas significativas. En la actualidad los estudios comparativos entre perfiles estudiantiles son dispares y falta estandarización en las métricas para medir la calidad del código y la eficiencia del aprendizaje [4]. Adicionalmente, la correlación entre la frecuencia de uso del tutor y la evolución de la autoeficacia sigue siendo un área poco explorada [3]. Estas limitaciones validan la pertinencia de esta investigación, que incorpora un análisis empírico, seguimiento de interacciones y evaluación de resultados en un entorno universitario real y en un contexto latinoamericano.

III. METODOLOGÍA

La estrategia metodológica se organiza en fases secuenciales con el fin de asegurar la replicabilidad del estudio sobre la implementación de tutores inteligentes en la enseñanza de programación para perfiles STEM y No-STEM. En concordancia con las directrices de investigaciones previas sobre experimentación con IA educativa [1], [2], cada bloque detalla los procedimientos requeridos antes de transitar a la siguiente etapa, garantizando la trazabilidad completa del proceso.

A. Selección de participantes y recolección inicial de datos

La muestra se conforma mediante la selección de estudiantes matriculados en 2 cursos de la asignatura Fundamentos de Programación en la Escuela Superior Politécnica del Litoral de Ecuador, se aplicaron criterios de inclusión estándar en la literatura de tutores inteligentes [5], [9].

Se reclutaron participantes de carreras de ingeniería (STEM) y licenciatura (No-STEM) para facilitar el análisis comparativo entre ambos perfiles [11].

Para la evaluación del Pre-test se administró una evaluación diagnóstica orientada a cuantificar los conocimientos previos, el nivel de familiaridad con los tutores inteligentes y la autopercepción de competencias técnicas, adoptando instrumentos validados en estudios de educación asistida por IA [3], [20]. Esta evaluación genera el primer conjunto de datos (D1). Adicionalmente, este instrumento permite clasificar y segmentar a la población estudiantil en los grupos STEM y No-STEM, dado que los cursos de origen presentan una matrícula heterogénea (año de inicio de estudios).

Para procesar el conjunto de datos inicial (D1) se realizaron procesos de limpieza y normalización de la información, siguiendo protocolos estándar para el manejo de volúmenes de datos educativos [6]. Las acciones incluyen:

- Depuración de registros duplicados y respuestas inconclusas.
- Estandarización de texto a formatos estructurados.
- Codificación de usuarios para evitar trabajar con datos personales.

B. División experimental y preparación del entorno

Para la asignación a grupos experimentales se utilizaron metodologías de intervención comparativa en perfiles académicos con LLM [4], se procede a la asignación aleatoria de los participantes en cuatro subgrupos de estudiantes:

- Grupo 1: STEM con asistencia del tutor.
- Grupo 2: No-STEM con asistencia del tutor.
- Grupo 3: STEM sin tutor (Grupo de Control).
- Grupo 4: No-STEM sin tutor (Grupo de Control).

Para la ejecución del experimento se preparó un entorno de desarrollo y varios ejercicios con una temática del silabo de la asignatura. Para los grupos experimentales (1 y 2) que utilizaron un LLM, se diseñó una guía para configurar un tutor de aprendizaje denominado TutorIA (Figura 1); esta guía facilitó la escritura de *prompts* para la generación de explicaciones conceptuales y ejercicios según el nivel de programación del estudiante y su carrera de estudio, evitar la entrega directa de código funcional y generar recomendaciones para optimizar respuestas y aplicar buenas prácticas de programación en su respuesta de la actividad.

Finalmente, para garantizar la recolección de métricas objetivas, se desplegaron *scripts* de monitoreo encargados de capturar las trazas de actividad, una técnica habitual en estudios de depuración asistida [6]. El conjunto de datos resultante (D2) integra:

- Tiempo total invertido por actividad.
- Tasa de éxito en pruebas unitarias.
- Secuencia de errores identificados y subsanados.
- Historial completo de *prompts* (exclusivo para grupos 1 y 2), permitiendo el análisis de la interacción humano-IA [4].



Fig. 1 TutorIA: Guía de diseño para configurar un tutor de programación

C. Ejecución del experimento

Los estudiantes realizaron una serie de actividades diseñadas para evaluar su razonamiento computacional, capacidades de depuración y resolución de problemas, competencias frecuentemente analizadas en la literatura sobre IA educativa [3], [9]. Las actividades propuestas en el experimento se centran en el uso de la librería pandas de Python, herramienta estándar en el análisis de datos y estadística. Es importante destacar que los grupos de control (3 y 4) resolvieron las actividades consultando exclusivamente la documentación oficial de pandas [21], sin asistencia del tutor inteligente.

La implementación técnica se lleva a cabo en la plataforma Google Colab, lo que facilitó la distribución de los cuadernos de trabajo y la ejecución de pruebas unitarias automatizadas para validar las soluciones propuestas por cada participante. Para cada grupo se tuvo en consideración lo siguiente:

- Interacciones con el tutor (grupos 1 y 2): El análisis de las interacciones se centraron en evaluar la calidad de los *prompts*, considerando dimensiones como claridad, estructura, intencionalidad y especificidad, variables críticas para el aprendizaje cognitivo mediado por IA [1], [2]. Asimismo, se cuantificaron métricas de eficiencia como la longitud de las entradas textuales y el número de iteraciones requeridas por el estudiante para alcanzar la solución de la actividad.
- Procesamiento del conjunto de datos experimental (D2): En la depuración del conjunto de datos experimental se eliminaron registros incompletos y se aplicó la normalización de las variables temporales, procedimientos recomendados para el análisis robusto del aprendizaje asistido por IA [6].

D. Pos-test y percepción subjetiva

Después de la fase práctica, se administraron dos versiones del pos-test diseñadas para medir la carga cognitiva, la claridad conceptual, la utilidad percibida, la satisfacción y la retención de información, basándose en instrumentos validados previamente [9], [20].

Se diferencian los instrumentos según el grupo experimental: una versión se dirige a los estudiantes que

interactuaron con el tutor inteligente, mientras que la otra se adapta para el grupo de control que utilizaron la documentación oficial de pandas.

Las respuestas cualitativas y cuantitativas fueron codificadas y normalizadas para su integración con los conjuntos de datos anteriores, siguiendo una estrategia de análisis multimodal típica en estudios mixtos sobre tecnologías educativas [3].

E. Integración de datos, rúbrica de evaluación y análisis

Para la construcción del conjunto de datos final (D4) se realizó la integración de los datos D1, D2 y D3, que permitieron generar métricas claves para el estudio:

- Ganancia de aprendizaje.
- Índice de eficiencia.
- Índice de claridad de *prompts*.
- Índice de depuración.

Estas variables compuestas reflejan las mejores prácticas en la evaluación de sistemas de tutoría inteligente [5], [6].

TABLA I
RÚBRICA PARA EVALUAR LA CALIDAD DE PROMPTS

Nro	Criterio (4 pts c/u)	Descriptores por nivel (4–0 pts)
1	Intención de aprendizaje	4 (Excelente): Declara objetivo de aprender y prohíbe respuesta directa. 3 (Bueno): Pide guía, límite contra “respuesta completa” implícito. 2 (Básico): Mezcla aprendizaje con petición de solución. 1–0 (Débil): Pide solución completa / “dame el código”.
2	Contexto y precisión técnica	4: Incluye librería, columnas, datos y duda específica. 3: Falta 1 elemento (ej. datos). 2: Vago (“no me sale el groupby”). 1–0: Sin contexto (“haz el ejercicio”).
3	Andamiaje: ayuda por pasos	4: Solicita pasos graduados y “mínima ayuda necesaria”. 3: Pide pasos sin control de granularidad. 2: Pide explicación general sin estructura. 1–0: Pide todo de una vez.
4	Auto-explicación y razonamiento	4: Pide “por qué”, alternativas o interpretación. 3: Pide explicación superficial. 2: Solo pide comandos. 1–0: Copiar/pegar sin explicación.
5	Participación activa (ICAP)	4: Incluye intento propio, error, salida esperada. 3: Incluye intento pero no el error. 2: No muestra intento. 1–0: Copia la consigna.
6	Verificación	4: Pide comprobaciones (head(), tipos, nulos). 3: Pide alguna verificación. 2: Menciona “verificar” sin detalle. 1–0: No contempla verificación.

Para evaluar la claridad de *prompts* se definió una rúbrica de evaluación (Tabla I) que sería aplicada a los grupos de estudiantes (STEM o No-STEM) que utilizaron la asistencia del tutor inteligente.

Por otra parte, el análisis de datos se realizó mediante las siguientes técnicas:

- ANOVA 2×2 para examinar los efectos principales y las interacciones entre variables.
- Análisis de correlación para vincular la claridad de la instrucción con el desempeño académico [2].
- Pruebas no paramétricas para aquellas variables que no cumplieron con supuestos de normalidad [3].

Con el objetivo de asegurar la transparencia y reproducibilidad científica, se documentaron exhaustivamente todos los códigos para análisis, las configuraciones del tutor inteligente y las estructuras de datos empleadas, en cumplimiento con los estándares de investigación en IAG [1].

IV. RESULTADOS

El estudio experimental se llevó a cabo en dos cursos distintos, impartidos por el mismo docente universitario y se han codificado como P1 y P2. La distribución demográfica de ambos cursos se detalla en la Tabla II.

TABLA II
DISTRIBUCIÓN GENERAL DE ESTUDIANTES STEM Y No-STEM

Paralelo Nro	STEM	No-STEM	Total
P1 (Sin LLM)	22	11	33
P2 (Con LLM)	26	9	35

La muestra final efectiva para la investigación (N=30) está compuesta por 18 estudiantes de carreras STEM y 12 de carreras No-STEM. En cuanto a la intervención, 9 participantes interactuaron con el tutor basado en LLM, mientras que el grupo de control, conformado por 21 estudiantes, resolvió las actividades utilizando únicamente la documentación oficial de pandas (Tabla III).

TABLA III
DISTRIBUCIÓN FINAL DE LA MUESTRA

Grupo	Con Tutor	Sin Tutor IA	Total
STEM	6	12	18
No-STEM	3	9	12

Para presentar los resultados del experimento se elaboró la Tabla IV que desglosa el desempeño individual por actividad (A1, A2, A3, A4, A5), así como la pertenencia a los grupos experimentales.

El análisis de los datos permitió desglosar el rendimiento según el perfil académico. Como se evidencia en la Tabla V los estudiantes de carreras STEM obtuvieron, en promedio, calificaciones superiores 0.51 en comparación con sus pares de carreras No-STEM 0.40. Asimismo, se contrastan los resultados entre el grupo experimental y el grupo de control; donde se observa una notable diferencia en el desempeño: el grupo asistido por el tutor de IAG superó al grupo que utilizó la documentación tradicional en todas las actividades evaluadas.

La Tabla V presenta el promedio de desempeño global usando el asistente inteligente y se aprecia una brecha considerable: el grupo experimental alcanzó un promedio cercano a 0.8, mientras que el grupo de control se situó alrededor de 0.33. Este hallazgo sugiere que, en términos generales, la interacción con el tutor inteligente facilitó la

resolución correcta de una mayor proporción de los ejercicios evaluados, evidenciando un rendimiento superior en el taller. Incluso situándose con mejor rendimiento que todos los subgrupos.

TABLA IV
RESULTADOS POR ESTUDIANTE, DESEMPEÑO Y DISTRIBUCIÓN

A1	A2	A3	A4	A5	Prom	STEM	IA
1	1	1	0	0	0.6	Sí	No
0	0	0	0	0	0.0	No	No
1	0	1	0	1	0.6	Sí	No
0	0	0	0	0	0.0	No	No
1	0	0	0	0	0.2	Sí	No
1	0	0	0	0	0.2	Sí	No
1	0	1	0	1	0.6	Sí	No
1	1	1	0	0	0.6	No	No
1	0	0	0	0	0.2	No	No
1	0	0	0	0	0.2	Sí	No
1	0	0	0	0	0.2	No	No
1	0	0	0	0.5	0.3	Sí	No
1	0	0	0	0	0.2	Sí	No
1	0	0	0	1	0.4	Sí	No
1	0.5	0	0	0	0.3	Sí	No
1	0	0	0	0	0.2	No	No
1	1	1	1	1	1.0	Sí	No
0	0	0	0	0	0.0	Sí	No
1	0	0	0	0	0.2	No	No
1	0	1	0	0	0.4	Sí	No
1	1	1	0	0	0.6	Sí	No
1	1	1	1	0	0.8	Sí	Sí
1	0	1	1	1	0.8	Sí	Sí
1	1	1	1	1	1.0	No	Sí
1	0	1	0	1	0.6	Sí	Sí
1	1	1	0	1	0.8	No	Sí
1	1	1	0	1	0.8	No	Sí
1	0	1	0	0	0.4	Sí	Sí
1	1	1	1	1	1.0	Sí	Sí
1	1	1	1	1	1.0	Sí	Sí

TABLA V
RESULTADOS POR ESTUDIANTE, DESEMPEÑO Y DISTRIBUCIÓN

Grupo	E1	E2	E3	E4	E5	Prom. Final
STEM	0.95	0.33	0.60	0.25	0.43	0.51
No-STEM	0.80	0.40	0.40	0.10	0.30	0.40
Con IA	1.00	0.67	1.00	0.56	0.78	0.80
Sin IA	0.86	0.21	0.33	0.05	0.21	0.33

La Tabla VI reporta la varianza del desempeño por actividad, métrica que indica la dispersión de los resultados entre los estudiantes. Los valores oscilan entre 0.09 y 0.26, sugiriendo una heterogeneidad moderada-alta en el rendimiento del taller. Específicamente, la Actividad 3 presentó la mayor varianza, identificándose como la actividad que generó mayores diferencias de desempeño. En contraste, la actividad 1

mostró la menor varianza, implicando resultados más homogéneos.

TABLA VI
DISPERSIÓN DE RESULTADOS POR ACTIVIDAD

Actividad	Varianza
1	0.09
2	0.23
3	0.26
4	0.17
5	0.24

A) *Tiempos de entrega de ejercicios*

Las Figura 2 y 3 comparan los tiempos de entrega entre estudiantes STEM y No-STEM, diferenciando entre quienes utilizaron IAG y quienes no.



Fig. 2 Tiempo de entrega – Estudiantes que usaron tutor inteligente

En el grupo experimental, la mediana del tiempo de entrega fue ligeramente inferior en los estudiantes STEM. Sin embargo, destaca la dispersión de los datos: el grupo STEM presenta un rango intercuartílico más amplio, indicando una mayor variabilidad en los tiempos de resolución (coexistencia de entregas rápidas y prolongadas). Por el contrario, en el grupo No-STEM se observa una mayor consistencia, con tiempos de entrega agrupados en un rango más estrecho.

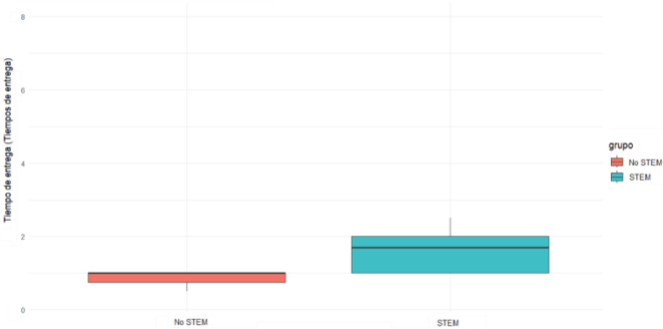


Fig. 3 Tiempo de entrega – Estudiantes que No usaron tutor inteligente

En el grupo de control, la diferencia fue más pronunciada ya que los estudiantes No-STEM registraron una mediana de tiempo más baja y una menor dispersión, lo que sugiere entregas rápidas y uniformes. En cambio, el grupo STEM mostró una mediana superior y una variabilidad considerable. En resumen, sin asistencia de IAG, los estudiantes No-STEM

tienden a entregar más rápido y con mayor uniformidad, mientras que los estudiantes STEM requieren más tiempo y muestran mayor heterogeneidad. La introducción de la IA tiende a homogeneizar las medianas, aunque la variabilidad en el grupo STEM persiste.

B) Evaluación de los prompts

La evaluación de la interacción con el tutor, detallada en la Tabla VII, revela diferencias en la formulación de *prompts*. Los datos indican que los estudiantes de carreras No-STEM obtuvieron un puntaje promedio superior (6.33) en comparación con el grupo STEM (5.98). Esto sugiere que los estudiantes de perfil No-STEM formularon *prompts* más estructuradas, completas y contextualmente ricas para guiar al tutor LLM.

TABLA VII

CALIDAD DE PROMPTS GRUPO ACADÉMICO QUE USÓ IA

Grupo	Promedio calidad de prompts
STEM	5.98
No-STEM	6.33

C) Percepciones iniciales de la IAG

Se documentaron las disposiciones y expectativas de los estudiantes frente a la inteligencia artificial antes de la interacción. Los resultados se estructuran en función de las dimensiones de actitud y conocimiento previo. La Figura 4 muestra que, en el grupo de Carreras No-STEM, los puntajes de percepción se concentran entre 3.2 y 4.1. En contraste, el grupo de carreras STEM exhibe una distribución más amplia (3.0 a 4.7), con una tendencia hacia valoraciones más altas (3.6–4.2).

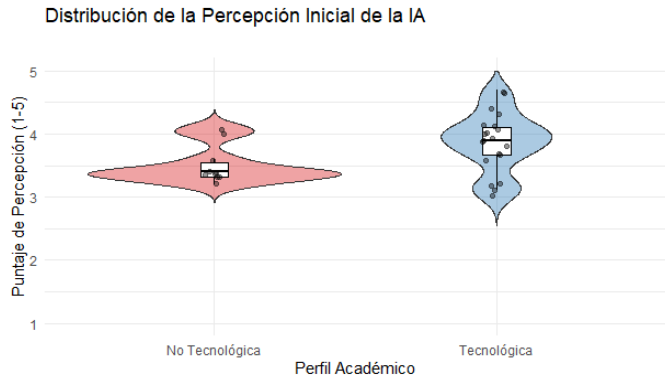


Fig. 4 Percepción inicial que tienen los estudiantes sobre el uso de IA

Se identifica un patrón consistente: los estudiantes con una percepción inicial alta obtuvieron mejores resultados en la totalidad de las actividades. Por ejemplo, en la Actividad 1, el grupo con percepción alta alcanzó 0.94 frente a 0.78 del grupo con percepción baja. La Figura 5 corrobora visualmente esta tendencia sistemática.

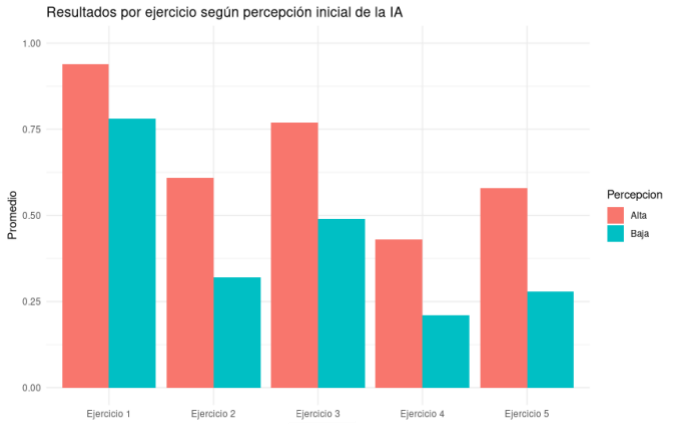


Fig. 5 Promedio de las actividades y la percepción inicial

D) Resultados del experimento vs. calidad de los prompts

La relación entre la calidad de las interacciones con el tutor y el resultado experimental se visualiza en las siguientes figuras. La Figura 6 ilustra la distribución de la calidad de los *prompts*. Si bien las medianas son similares entre ambos grupos, el grupo No-STEM presenta un rango intercuartílico más amplio, indicando una mayor heterogeneidad en la calidad de sus *prompts* (coexistencia de *prompts* excelentes y deficientes). Por el contrario, el grupo STEM muestra una caja más compacta, lo que denota una mayor consistencia en la calidad de sus interacciones.

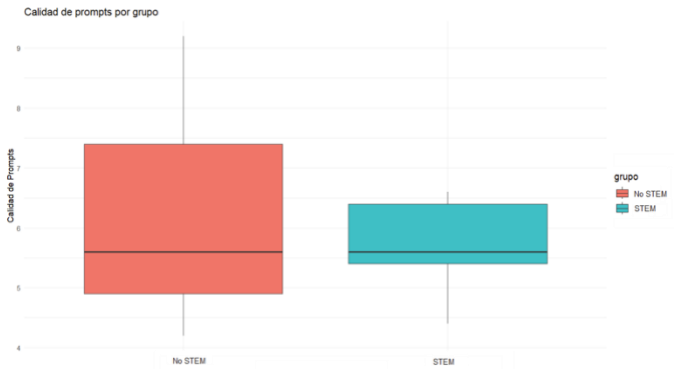


Fig. 6 Calidad de los prompts

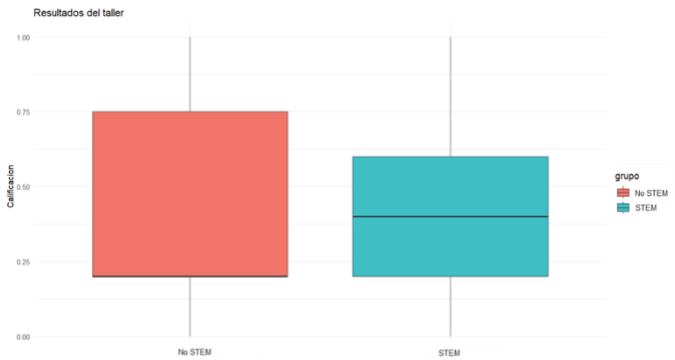


Fig. 7 Resultados del grupo con tutor inteligente

Al analizar las calificaciones del grupo experimental (Figura 7), se observa que la mediana del grupo STEM es superior, indicando un mejor desempeño general. El grupo No-STEM muestra una mayor dispersión, con resultados que fluctúan desde notas bajas hasta altas.

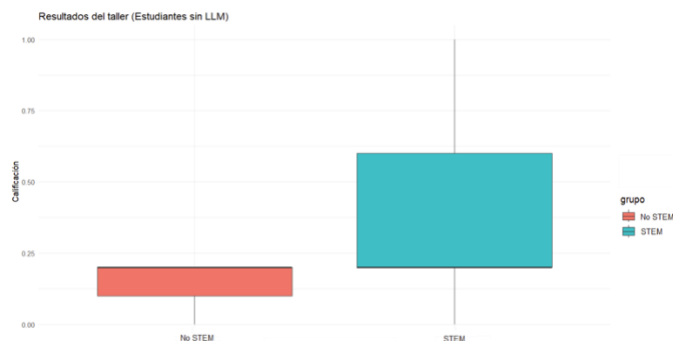


Fig. 8 Resultados del grupo sin tutor inteligente

Finalmente, se presentan los resultados del grupo de control sin tutor inteligente. La Figura 8 evidencia una diferencia clara en ausencia de asistencia por IA: la mediana del grupo STEM es considerablemente más alta que la del grupo No-STEM. Asimismo, la dispersión en STEM es amplia, reflejando diferencias individuales, mientras que en el grupo No-STEM los resultados se concentran en valores bajos, denotando una dificultad generalizada para resolver las actividades sin el apoyo del tutor inteligente.

V. DISCUSIÓN

Los hallazgos de esta investigación ratifican la hipótesis de que la implementación de tutores basados en Modelos de Lenguaje (LLM) ejerce un impacto positivo en el rendimiento académico dentro de cursos introductorios de programación. La comparación del promedio final entre el grupo experimental (0.78) y el grupo de control (0.43) (Tabla V) demuestra una ganancia de aprendizaje sustancial. Este resultado se alinea con la literatura contemporánea, que posiciona a la IAG como un catalizador clave para la eficiencia educativa y la personalización del aprendizaje [1], [9].

Un aspecto sobresaliente del análisis radica en la disparidad observada entre los perfiles académicos. Aunque los estudiantes de carreras STEM mantuvieron calificaciones superiores (Promedio Final: 0.59 frente a 0.47), la evaluación de la calidad de los *prompts* (Tabla VIII) reveló un patrón contraintuitivo: los estudiantes No-STEM formularon *prompts* de mayor calidad semántica (6.33) en comparación con sus pares STEM (5.98).

Esta evidencia sugiere que, mientras el grupo STEM tiende a apoyarse en su lógica técnica preexistente y no por una interacción semántica precisa, por su parte los estudiantes No-STEM compensan sus vacíos técnicos mediante una comunicación más detallada y contextualizada con el tutor. No obstante, la elevada varianza detectada en los *prompts* del grupo No-STEM (Fig. 6) advierte que esta competencia comunicativa

no es homogénea; existe una brecha interna donde un segmento del grupo logra interactuar eficazmente, mientras otra falla en la estructuración de sus consultas.

El examen de la métrica temporal arroja matices importantes sobre el comportamiento de los estudiantes. En el escenario sin asistencia de IA, el grupo No-STEM registró tiempos de entrega menores (Fig. 3), pero estos se correlacionaron con las calificaciones más bajas. Este fenómeno podría interpretarse no como eficiencia, sino como una “deserción prematura” o resolución superficial ante la ausencia de soporte. En contraste, la incorporación del tutor inteligente incrementó la variabilidad temporal en el grupo STEM (Fig. 3), lo que indica que la herramienta facilitó procesos de iteración y profundización, extendiendo el tiempo de dedicación productiva hacia soluciones más robustas.

Se valida estadísticamente la relación inversa entre la carga cognitiva y el desempeño académico. Tanto en las mediciones iniciales como finales, aquellos participantes que reportaron una carga “Alta” obtuvieron sistemáticamente los promedios más deficientes. Esto implica que el tutor inteligente, por sí solo, no logra mitigar las barreras cognitivas si el estudiante carece de una base conceptual mínima para procesar la ayuda. Paralelamente, se observó que una percepción inicial positiva hacia la tecnología se tradujo en mejores resultados, subrayando la importancia de la predisposición actitudinal en el aprendizaje asistido.

VI. CONCLUSIONES

El presente estudio logró evaluar y comparar la eficacia de los tutores inteligentes para reforzar el aprendizaje en programación entre estudiantes de perfiles STEM y No-STEM. Se confirmó que el uso del tutor inteligente incrementa el rendimiento académico de manera generalizada, superando al grupo de control en todas las actividades. Sin embargo, la utilidad es cualitativamente distinta: los estudiantes STEM emplean la herramienta para perfeccionar la calidad técnica y sintáctica del código, mientras que los estudiantes No-STEM se benefician primordialmente en conceptos conceptuales, logrando superar obstáculos iniciales de programación que, sin asistencia, resultaban insalvables.

Se identificó una dicotomía crítica en la interacción con el tutor inteligente y la calidad de *prompts*. Los estudiantes No-STEM demostraron una capacidad superior para la ingeniería de *prompts* (6.33 vs 5.98), utilizando el lenguaje natural para contextualizar sus problemas con mayor eficacia que el grupo STEM. No obstante, el grupo STEM exhibió mayor consistencia y menor varianza en sus tiempos y resultados, denotando un proceso de aprendizaje más estable y predecible.

Con respecto a la carga cognitiva, se evidenció que el grado de aprendizaje no depende exclusivamente de la corrección del código, sino de la capacidad de procesamiento del estudiante. Los datos de carga cognitiva evidencian que el tutor es altamente efectivo cuando el usuario experimenta una carga

moderada o baja; en escenarios de saturación cognitiva, la herramienta no logra compensar plenamente las deficiencias en la comprensión base.

Para una implementación adecuada de tutores inteligentes, es imperativo diversificar las estrategias pedagógicas. La estrategia de TutorIA facilitó la configuración de un asistente para perfiles No-STEM; sin embargo, se recomienda enfatizar el fortalecimiento de la lógica de programación para reducir la varianza en el desempeño. Por su parte para perfiles STEM, la instrucción debe enfocarse en desarrollar habilidades de comunicación efectiva con la IA (*prompting*), área en la que mostraron un desempeño inferior al potencial esperado.

En definitiva, si bien los tutores inteligentes actúan como un potente nivelador de oportunidades educativas, no eliminan por completo la brecha disciplinaria. El éxito de su integración curricular radica en reconocer que los estudiantes STEM y No-STEM interactúan con la inteligencia artificial desde paradigmas fundamentalmente distintos: los primeros desde la lógica algorítmica y los segundos desde la semántica del lenguaje natural.

VII. TRABAJO FUTURO

A partir de los hallazgos presentados, se plantean varias líneas de investigación para profundizar en el impacto de la IAG en la enseñanza de la programación como una habilidad transversal en universitarios:

Como paso inmediato, se planea expandir el estudio a una muestra significativamente mayor ($N > 100$), involucrando a estudiantes de diversas facultades y niveles académicos para robustecer la validez estadística de las comparaciones entre perfiles STEM y No-STEM, contemplando realizar un seguimiento a largo plazo para evaluar si la competencia en la "calidad del *prompt*" se traduce en una mejora sostenida de las habilidades de resolución de problemas y pensamiento computacional, más allá de la finalización del curso.

Se explorará la integración de métricas más avanzadas en las pruebas unitarias para medir no solo la corrección funcional del código, sino también su eficiencia y legibilidad, proporcionando una retroalimentación aún más matizada para el aprendizaje personalizado. Finalmente, evaluar cómo los diferentes LLM afectan el desempeño de los estudiantes según su perfil, con el fin de identificar herramientas óptimas para cada disciplina.

REFERENCIAS

- [1] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learn. Individ. Differ.*, vol. 103, p. 102274, abr. 2023, doi: 10.1016/j.lindif.2023.102274.
- [2] K. Zhang and A. B. Aslan, "Ai technologies for education: Recent research 'i&' future directions," 2021.
- [3] L. Labadze, M. Grigolia, and L. Machaidze, "Role of ai chatbots in education: systematic literature review," 2023.
- [4] V. Bhatt, Z. Yu, Y. Hou, and J. Jin, "Chatgpt as a programming tutor: Student perceptions, effectiveness, and challenges," in *IEEE Global Engineering Education Conference, EDUCON*, 2025.
- [5] H. Reicher, Y. Frenkel, M. J. Lavi, R. Nasser, Y. Ran-milo, R. Sheinin, M. Shtauf, and T. Milo, "A generative ai-empowered digital tutor for higher education courses," *Information (Switzerland)*, vol. 16, 4 2025.
- [6] T. Phung, V. A. P'adurean, A. Singh, C. Brooks, J. Cambronero, S. Gulwani, A. Singla, and G. Soares, "Automating human tutor-style programming feedback: Leveraging gpt-4 tutor model for hint generation and gpt-3.5 student model for hint validation," in *ACM International Conference Proceeding Series*. Association for Computing Machinery, 3 2024, pp. 12–23.
- [7] M. Rodriguez, J. Wilson, and S. Thompson, "Bridging the digital divide: Llm tutors for non-technical students in programming courses," in *Proceedings of the ACM Technical Symposium on Computer Science Education*, 2024, pp. 456–462.
- [8] S. Kumar and P. Patel, "Comparative analysis of ai tutoring systems in computer science education," *International Journal of Artificial Intelligence in Education*, vol. 33, no. 4, pp. 789–812, 2023.
- [9] R. Yilmaz and F.G.K. Yilmaz, "The effect of generative artificial intelligence (ai) – based tool use on students' computational thinking skills, programming self-efficacy and motivation," *Computers and Education: Artificial Intelligence*, vol. 4, 2023.
- [10] A. Patel and L. Wang, "Longitudinal effects of ai tutoring on programming self-efficacy across different student backgrounds," *Journal of Computer Assisted Learning*, vol. 40, no. 2, pp. 512–528, 2024.
- [11] V. Kovalchuk, S. Reva, I. Volch, S. Shcherbyna, H. Mykhailyshyn, and T. Lychova, "Artificial intelligence as an effective tool for personalized learning in modern education," in *Environment Technology Resources - Proceedings of the 16th International Scientific and Practical Conference*, vol. 3. RTU PRESS, 2025, pp. 187–194.
- [12] M. Lee and P. Ji-hoon, "Student engagement patterns with ai programming tutors: A mixed-methods study," *Computers in Human Behavior*, vol. 148, p. 107901, 2023.
- [13] R. Anderson and D. Kim, "Designing ethical ai tutors: Addressing bias and fairness in programming education," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–15.
- [14] R. Thompson and M. Silva, "Debugging with ai: How large language models transform programming error resolution," in *ACM Transactions on Computing Education*, vol. 23, no. 4, pp. 1–24, 2023.
- [15] X. Chen and H. Li, "The impact of prompt engineering on learning outcomes in ai-supported programming education," *Journal of Educational Computing Research*, vol. 62, no. 3, pp. 645–668, 2024.
- [16] E. Johnson and M. Brown, "Assessing code quality and learning outcomes in llm-supported programming courses," *IEEE Transactions on Education*, vol. 67, no. 2, pp. 156–169, 2024.
- [17] C. Garcia and E. Martinez, "Scaffolding programming concepts through conversational ai tutors," in *Proceedings of the International Conference on Learning Sciences*, 2025, pp. 234–245.
- [18] W. Wu and Z. Li, "Personalized learning paths in programming education using llm-based recommender systems," in *Proceedings of the IEEE Frontiers in Education Conference*, 2024, pp. 1–8.
- [19] R. Singh and A. Flores, "Faculty perspectives on implementing ai tutors in university programming courses," *Education and Information Technologies*, vol. 29, no. 3, pp. 3456–3478, 2024.
- [20] K. Doty and L. Lipien, "Online k-12 teachers' perceptions of students' ai utilization and teachers' outlook on the future of education in the context of artificial intelligence," *American Journal of Educational Research*, vol. 12, pp. 319–328, 9 2024.
- [21] The pandas development team, "Pandas documentation," 2026. [En línea]. Disponible en: <https://pandas.pydata.org/docs/>