Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), septiembre-octubre 2026,
Volumen 10, Número 5.

https://doi.org/10.37811/cl_rcm.v10i5

ASISTENTES VIRTUALES PARA LA ENSEÑANZA DE PROGRAMACIÓN EN ESTUDIANTES DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

VIRTUAL ASSISTANTS FOR PROGRAMMING EDUCATION IN SYSTEMS AND INFORMATICS ENGINEERING STUDENTS.

Carlos Eugenio Vega Moreno
Universidad Nacional del Santa

Hugo Esteban Caselli Gismondi
Universidad Nacional del Santa

Lizbeth Dora Briones Pereyra
Universidad Nacional del Santa

DOI: https://doi.org/10.37811/cl_rcm.v10i5.25869

Asistentes virtuales para la enseñanza de programación en estudiantes de Ingeniería de Sistemas e Informática

Carlos Eugenio Vega Moreno [1]
cvega@uns.edu.pe
https://orcid.org/0000-0003-2955-0674
Universidad Nacional del Santa
Perú

Hugo Esteban Caselli Gismondi
hcasellig@uns.edu.pe
https://orcid.org/0000-0002-2812-6727
Universidad Nacional del Santa
Perú

Lizbeth Dora Briones Pereyra
lbriones@uns.edu.pe
https://orcid.org/0000-0003-0626-7227
Universidad Nacional del Santa
Perú

RESUMEN

La enseñanza de la Programación Orientada a Objetos (POO) en ingeniería presenta desafíos pedagógicos debido a la alta complejidad de conceptos como herencia, encapsulamiento y polimorfismo. Esta investigación tuvo como objetivo evaluar el impacto de los asistentes virtuales generativos (GitHub Copilot) en el rendimiento académico, la calidad del código y la percepción estudiantil. Se aplicó un diseño cuasiexperimental de enfoque cuantitativo con una muestra de 54 estudiantes de la Escuela Profesional de Ingeniería de Sistemas e Informática (EPISI), divididos en un grupo control ($N = 27$) y un experimental ($N = 27$). Los instrumentos incluyeron pruebas de desempeño práctico y un cuestionario psicométrico de percepción validado ($\alpha = 0.85$). Los resultados evidencian una mejora estadísticamente significativa en el rendimiento del grupo experimental ($M = 16.30$) frente al control ($M = 13.22$) ($t = 9.45$, $p < 0.001$), así como una reducción en la dispersión de las calificaciones, operando la IA como un ecualizador de competencias. Aunque el 74% de los estudiantes reportó alta satisfacción, no se halló una correlación significativa ($r = 0.277$, $p > 0.05$) entre la destreza en Prompt Engineering y la calificación final. Se concluye que la IA generativa actúa como un andamiaje cognitivo eficaz bajo la zona de desarrollo próximo, aunque requiere una instrucción docente formal para evitar la dependencia sintáctica.

Palabras clave: Inteligencia Artificial Generativa; Programación Orientada a Objetos; Andamiaje Cognitivo; Calidad del Código; Ingeniería de Prompts

[1] Autor principal.
Correspondencia: cvega@uns.edu.pe

Virtual assistants for programming education in Systems and Informatics Engineering students.

ABSTRACT

Teaching Object-Oriented Programming (OOP) in engineering presents pedagogical challenges due to the high complexity of concepts such as inheritance, encapsulation, and polymorphism. This research aimed to evaluate the impact of generative virtual assistants (GitHub Copilot) on academic performance, code quality, and student perception. A quantitative quasi-experimental design was applied with a sample of 54 students from the Professional School of Systems and Informatics Engineering (EPISI), divided into a control group ($N = 27$) and an experimental group ($N = 27$). Instruments included practical performance tests and a validated psychometric perception questionnaire ($\alpha = 0.85$). Results show a statistically significant improvement in the performance of the experimental group ($M = 16.30$) compared to the control group ($M = 13.22$) ($t = 9.45$, $p < 0.001$), as well as a reduction in grade dispersion, with AI acting as a competency equalizer. Although 74% of students reported high satisfaction, no significant correlation ($r = 0.277$, $p > 0.05$) was found between Prompt Engineering proficiency and the final grade. It is concluded that generative AI acts as an effective cognitive scaffolding under the zone of proximal development, although it requires formal instructional guidance to prevent syntactic dependency.

***Keywords*:** Generative Artificial Intelligence; Object-Oriented Programming; Cognitive Scaffolding; Code Quality; Prompt Engineering

Artículo recibido 06 septiembre 2026
Aceptado para publicación: 07 octubre 2026

INTRODUCCIÓN

La enseñanza de la programación representa hoy uno de los pilares fundamentales en la formación de ingenieros; sin embargo, esta disciplina presenta desafíos significativos debido a la alta complejidad de sus conceptos abstractos y la constante necesidad de práctica técnica. En el contexto de la Escuela Profesional de Ingeniería de Sistemas e Informática (EPISI) de la Universidad Nacional del Santa, se ha observado que los estudiantes requieren con frecuencia de una asistencia personalizada para la resolución de errores y el desarrollo de lógica algorítmica, un apoyo que, en el modelo de aula tradicional, resulta insuficiente o de difícil acceso.

Frente a esta problemática, la Inteligencia Artificial (IA) generativa ha irrumpido como una oportunidad disruptiva. Estudios previos, como los de Rodríguez Esquíroz (2024), Pérez et al. (2020) y García-Peñalvo (2021), han demostrado que los asistentes virtuales son recursos capaces de proporcionar apoyo personalizado en tiempo real, optimizar la práctica docente y fomentar nuevas dinámicas de aprendizaje. No obstante, su aplicación específica en el proceso de enseñanza-aprendizaje de la programación aún requiere de una exploración rigurosa que valide su efectividad pedagógica y técnica.

En este contexto, si bien la literatura reciente ha documentado extensamente los beneficios generales de los Modelos de Lenguaje Grande (LLM) en la resolución de algoritmos secuenciales básicos y la depuración rápida (Becker et al., 2023), existe un notable vacío empírico respecto a la adopción de asistentes de codificación integrados (como GitHub Copilot) orientados a paradigmas de alta abstracción como la Programación Orientada a Objetos (POO). Más críticamente, los estudios actuales en la intersección de AIED (Inteligencia Artificial en Educación) aún no han resuelto la tensión entre la generación automática de código y el riesgo de la "ilusión de competencia" en estudiantes novatos (Chen et al., 2024). Investigaciones preliminares advierten sobre una excesiva dependencia tecnológica (Prather et al., 2023), donde la obtención de una sintaxis funcional autogenerada enmascara la falta de asimilación cognitiva de conceptos complejos como el encapsulamiento y el polimorfismo. Asimismo, aunque se reconoce el valor del andamiaje cognitivo automatizado (Kazemitabaar et al., 2023), se carece de evidencia concluyente sobre cómo la destreza del estudiante en la formulación de instrucciones (Prompt Engineering) actúa como variable moderadora para mitigar esta dependencia y promover un aprendizaje autorregulado en el diseño de software.

Esta investigación surge para dar respuesta a la siguiente pregunta: ¿Cómo influyen los asistentes virtuales en el aprendizaje del curso de Programación de los estudiantes de la EPISI de la Universidad Nacional del Santa? La importancia de este estudio radica en la creciente necesidad del mercado laboral por profesionales con competencias tecnológicas avanzadas. La investigación se justifica socialmente por su potencial para democratizar el acceso a recursos educativos de calidad; técnicamente, por su capacidad de ofrecer retroalimentación inmediata en la depuración de código; económicamente, por la optimización de recursos en la gestión educativa; y metodológicamente, al proponer un diseño cuantitativo cuasiexperimental que articula el análisis del rendimiento académico con la evaluación psicométrica de la percepción y adopción tecnológica del estudiante.

Para alcanzar el propósito central de esta investigación, se ha definido como objetivo general: Evaluar el impacto de los asistentes virtuales en el aprendizaje del curso de Programación de los estudiantes de la EPISI. Este objetivo se desglosa en los siguientes específicos: 1) Analizar la mejora en el rendimiento académico de los estudiantes que utilizan asistentes virtuales. 2) Evaluar la satisfacción y percepción de los estudiantes respecto al uso de estas herramientas. 3) Identificar las funcionalidades de los asistentes virtuales que resultan más útiles durante el proceso de aprendizaje. 4) Analizar la relación entre la calidad de la ingeniería de prompts (iteración con la IA) y el rendimiento académico en el grupo experimental. 5) Evaluar el impacto de los asistentes virtuales en la calidad específica del código desarrollado en Programación Orientada a Objetos (abstracción, encapsulamiento y polimorfismo).

Con el fin de cumplir con estos propósitos, se formula la hipótesis de investigación de que los estudiantes de la Escuela Profesional de Ingeniería de Sistemas e Informática (EPISI) de la Universidad Nacional del Santa que utilizan el asistente virtual GitHub Copilot en el curso de Programación muestran una mejora estadísticamente significativa en su rendimiento académico final, en comparación con el grupo control que sigue una metodología tradicional sin mediación de Inteligencia Artificial.

A nivel de marco teórico, la teoría del aprendizaje constructivista sostiene que los estudiantes construyen activamente su conocimiento a través de la interacción con el entorno y la resolución de problemas (Anderson, 2013). En entornos digitales, el diseño de estas actividades requiere estructurar experiencias de aprendizaje auténticas que estimulen la autonomía del estudiante (Zualkernan, 2006). Por otro lado, los sistemas de tutoría inteligente (ITS) utilizan algoritmos de IA para adaptar el contenido educativo a

las necesidades individuales de los estudiantes (Brusilovsky & Millán, 2007). El desarrollo de estos sistemas representa una oportunidad emergente para reducir las brechas educativas en regiones en desarrollo (Nye, 2015). Los asistentes virtuales pueden ofrecer tutorías personalizadas, ayudando a los estudiantes a superar obstáculos específicos (Chen & Duh, 2008), analizando su progreso y ajustando las lecciones en consecuencia (Dillenbourg et al., 2009). Asimismo, actúan como mediadores en el proceso de enseñanza-aprendizaje, simulando la interacción social que, según Vygotsky (1978), enriquece la adquisición del conocimiento (Cariaga, 2020).

A nivel del marco conceptual, la Ingeniería de Prompts (*Prompt Engineering*) se define como el proceso sistemático de diseño y refinamiento continuo de las instrucciones enviadas a un LLM para asegurar que las respuestas estén pedagógicamente alineadas con las metas del curso de POO (García-Peñalvo, 2021). El Aprendizaje Adaptativo representa la capacidad del entorno digital para ajustar el contenido y el soporte basándose en el desempeño, promoviendo el aprendizaje autorregulado del estudiante (Shute & Zapata-Rivera, 2012; Zimmerman, 2002). La Retroalimentación Adaptativa proporciona pistas lógicas y andamiajes intermedios en lugar de la solución final, evitando la dependencia pasiva (Koedinger & Corbett, 2006; Vygotsky, 1978; Zumbach & Reimann, 2002). Finalmente, la Aceptación Tecnológica se fundamenta en el modelo TAM mediante las variables de utilidad percibida y facilidad de uso percibida (Davis, 1989). El rendimiento académico se operacionaliza como el nivel de competencia técnica, conceptual y práctica alcanzado en la asignatura, medido cuantitativamente mediante rúbricas y exámenes prácticos (VanLehn, 2011).

METODOLOGÍA

El estudio es de tipo aplicado, con un enfoque cuantitativo y un diseño de investigación cuasiexperimental con grupos de control y experimental evaluados de manera controlada y simultánea.

Población y Muestra

- **Población:** Estudiantes matriculados en la Escuela Profesional de Ingeniería de Sistemas e Informática (EPISI) de la Universidad Nacional del Santa en el semestre 2025-II.
- **Muestra:** Estudiantes matriculados en el curso de Programación de la EPISI en el ciclo 2025-II.

Se aplicó un muestreo no probabilístico por conveniencia, conformando dos grupos con asignación intacta: un grupo experimental ($N = 27$) y un grupo control ($N = 27$) .

- **Criterios de inclusión:** Estudiantes matriculados oficialmente en la asignatura en el ciclo 2025-II, con acceso a dispositivos con internet, y que firmaron el consentimiento informado voluntario.
- **Criterios de exclusión:** Inasistencia regular o no compleción de las encuestas o evaluaciones académicas requeridas.

Variables de Estudio

- **Variable Independiente:** Uso de asistentes virtuales basados en IA generativa (GitHub Copilot) en la enseñanza de programación, operacionalizado en dimensiones de frecuencia de uso, utilidades operativas e interacción (ingeniería de prompts).
- **Variable Dependiente:** Rendimiento académico de los estudiantes, medido a través de evaluaciones teóricas (exámenes), evaluaciones prácticas (resolución de ejercicios, proyectos) y el promedio final en escala vigesimal (0 a 20).

Técnicas e Instrumentos de Recolección de Datos

Para la recopilación de la información se utilizaron dos técnicas estructuradas:

1. **Encuestas de satisfacción y percepción:** Aplicadas mediante un cuestionario estructurado de 11 ítems con escala Likert de 5 puntos.
2. **Pruebas de rendimiento académico:** Exámenes de codificación y registros prácticos evaluados mediante rúbrica vigesimal para cuantificar el impacto cognitivo y arquitectónico en POO.

Tabla 1
Técnicas e instrumentos de recolección de datos

Técnica	Instrumento	Sujetos	Objetivo principal
Encuestas de satisfacción y percepción	Cuestionarios estructurados (Likert 1 -5)	Estudiantes	Evaluar la experiencia de uso, la percepción de utilidad y la autopercepción de interacción (ingeniería de prompts) con la IA.
Pruebas de rendimiento académico	Exámenes, evaluaciones de código y registros de notas (0 – 20)	Estudiantes	Cuantificar el impacto del asistente virtual en las calificaciones, el promedio final y la calidad técnica del código de POO.

Nota. Adaptado de los registros metodológicos del proyecto

Validez y Confiabilidad

El cuestionario de percepción fue sometido a validez de contenido mediante juicio de tres expertos en ingeniería de sistemas y educación. La consistencia interna se evaluó mediante una prueba piloto que arrojó un Alfa de Cronbach de 0.85, confirmando una confiabilidad satisfactoria.

Materiales Tecnológicos y Procedimiento

La intervención tecnológica se estructuró empleando GitHub Copilot integrado como extensión nativa en el IDE Visual Studio Code. El grupo experimental interactuó de manera activa con el asistente para la depuración y generación asistida de código, mientras que el grupo de control desarrolló exactamente los mismos retos algorítmicos en Visual Studio Code pero con la extensión de Copilot deshabilitada. El experimento se desarrolló secuencialmente durante el semestre 2025-II en cinco etapas: preparación, capacitación ética y técnica, implementación guiada, recolección de datos y procesamiento estadístico mediante pruebas paramétricas (t de Student) y no paramétricas (correlaciones de Pearson y Spearman).

RESULTADOS Y DISCUSIÓN

Impacto en el Rendimiento Académico (Objetivo Específico 1)

El análisis descriptivo comparativo de las calificaciones finales vigesimales entre ambos grupos de estudio revela un impacto positivo sobresaliente a favor del aprendizaje mediado por Inteligencia Artificial.

Tabla 2

Estadísticos descriptivos del rendimiento académico final según grupo de estudio.

Grupo	Muestra (*N*)	Media (*M*)	Desviación Estándar (SD)	Mediana (Mdn)	Min	Max
Control	27	13.22	1.25	13	10	16
Experimental	27	16.30	1.14	16	15	19

Nota: Calificaciones evaluadas sobre una escala vigesimal (0 a 20).

Como se aprecia en la Tabla 2, el grupo experimental alcanzó un promedio final de $M = 16.30$ puntos frente a un $M = 13.22$ del grupo control, lo que denota una **ganancia absoluta de 3.08 puntos** en el rendimiento de los estudiantes asistidos por la IA. Metodológicamente, se observa además una disminución en la desviación estándar del grupo experimental ($SD = 1.14$) en relación con el grupo de control ($SD = 1.25$), reduciendo el rango de notas de 6 a 4 puntos. Esto corrobora el **efecto ecualizador de la IA generativa**, la cual eleva de manera uniforme el desempeño de la muestra, homogeneizando el

nivel de adquisición de competencias técnicas.

Para contrastar formalmente la hipótesis de investigación, se aplicó la prueba *t* de Student para muestras independientes bajo varianzas asumidas iguales.

Tabla 3

Resultados de la prueba t de Student para muestras independientes del rendimiento académico.

	Grupo Experimental	***Grupo de Control***
Media	16.30	13.22
Varianza	1.30	1.56
Observaciones	27	27
Diferencia hipotética de las medias	0	
Grados de libertad	52	
Estadístico t	9.45	
P(T<=t) una cola	3.54E-13	
Valor crítico de t (una cola)	1.68	
P(T<=t) dos colas	7.083E-13	
Valor crítico de t (dos colas)	2.007	

Nota. p < .001 (prueba bilateral o de dos colas) con varianzas asumidas iguales

El análisis inferencial de la Tabla 3 confirma una diferencia estadísticamente muy significativa entre ambos grupos ($t(52) = 9.45$, $p < 0.001$). Se rechaza la hipótesis nula y se demuestra empíricamente que la incorporación guiada de asistentes de codificación optimiza significativamente los promedios finales de la asignatura de programación.

Análisis de Percepción, Adopción y Utilidades Operativas (Objetivos Específicos 2 y 3)

La evaluación psicométrica mediante cuestionario Likert en el grupo experimental permitió categorizar el comportamiento y la valoración subjetiva de los participantes hacia las dimensiones clave de la IA

Tabla 4

Distribución de frecuencias y porcentajes sobre el uso frecuente y utilidades operativas .

Reactivos de Percepción (Escala Likert 1-5)	Totalmente en desacuerdo (1)	En desacuerdo (2)	Neutral (3)	De acuerdo (4)	Totalmente de acuerdo (5)	M Media	SD Desv.
Ítem 1: Uso semanal recurrente para tareas	0.00%	18.52%	25.93%	40.74%	14.82%	3.52	0.98
Ítem 2: Dedicación de tiempo diario	3.70%	7.41%	44.44%	37.04%	7.41%	3.37	0.88
Ítem 3: Uso enfocado a generar código	3.70%	7.41%	40.74%	29.63%	18.52%	3.52	1.01
Ítem 4: Uso enfocado a resolver dudas	0.00%	7.41%	3.70%	37.04%	51.85%	4.33	0.88
Ítem 5: Uso para comprender conceptos	0.00%	3.70%	14.81%	37.04%	44.44%	4.22	0.85

Nota. N = 27 (grupo experimental). Cada ítem refleja la proporción porcentual de frecuencias absolutas del cuestionario

Los datos consolidados de la Tabla 4 reflejan patrones de adopción altamente reflexivos. Se observa que el valor más apreciado por los estudiantes no reside en la simple generación sintáctica automatizada de código (Ítem 3, donde predomina la neutralidad con un 40.74%), sino en la retroalimentación inmediata y la capacidad del modelo para actuar como un tutor personalizado que desglosa dudas (Ítem 4, con **88.89% de acuerdo** masivo y $M = 4.33$) y facilita la asimilación conceptual de abstracciones complejas (Ítem 5, con **81.48% de acuerdo** y $M = 4.22$).

Evaluación de la Ingeniería de Prompts e Interacción Técnica (Objetivo Específico 4)

La interacción de los estudiantes con el asistente virtual se evaluó a través de su autopercepción respecto a la claridad, nivel de detalle e inclusión de ejemplos previos al formular prompts en los retos prácticos

de POO, medida a través de la dimensión de interacción (Ítems 6, 7 y 8) del cuestionario.

Tabla 5

Distribución de frecuencias sobre competencias de Ingeniería de Prompts

Estrategias de Prompting (Escala Likert 1-5)	**1 – TD**	**2 - D**	**3 - N**	**4 - A**	**5 - TA**	**M Media**	**SD Desv.**
Ítem 6: Formulación de preguntas claras	0.00%	7.41%	14.81%	51.85%	25.93%	3.96	0.89
Ítem 7: Provisión de contexto detallado	0.00%	7.41%	25.93%	59.26%	7.41%	3.67	0.78
Ítem 8: Inclusión de ejemplos previos	0.00%	7.41%	22.22%	62.96%	7.41%	3.70	0.72

Nota. N = 27. Categorías simplificadas: TD (Totalmente en desacuerdo), D (En desacuerdo), N (Neutral), A (De acuerdo), TA (Totalmente de acuerdo)

La Tabla 5 evidencia que la gran mayoría de los estudiantes aplican estrategias proactivas de interacción técnica, destacando que el 77.78% se esfuerza en estructurar preguntas claras (Ítem 6) y el 70.37% proporciona ejemplos de código previo para contextualizar la depuración del modelo (Ítem 8). No obstante, existe un rango de incertidumbre metodológica de un 22% a un 25% de estudiantes en nivel neutral, lo que devela una brecha en la instrucción formal de *Prompt Engineering* en la malla de ingeniería

La Relación entre Ingeniería de Prompts y el Promedio Académico

Para contrastar el nivel de asociación entre la calidad instruccional (prompts) y el logro de aprendizaje, se realizó un análisis de correlación bivariada paramétrico (Pearson) y no paramétrico (Spearman)

Tabla 6

Resultados del análisis de correlación entre calidad del prompt y promedio académico.

Variables en Análisis	Coeficiente	Valor Coeficiente	Significancia (*p*-value)	Decisión Estadística
Calidad del Prompt vs. Promedio Final	Pearson (*r*)	0.277	0.162	No significativo ($p > .05$)
Calidad del Prompt vs. Promedio Final	Spearman (ρ)	0.236	0.236	No significativo ($p > .05$)

Nota. $N = 27$ estudiantes pertenecientes al grupo experimental. La variable 'Promedio final' fue evaluada empleando la escala vigesimal de 0 a 20 puntos, mientras que la variable 'Calidad del prompt' representa la autopercepción de competencia del estudiante, calculada mediante el puntaje compuesto promedio de la Dimensión de Interacción (Ítems 6, 7 y 8) del cuestionario en escala Likert (1 a 5).

Los resultados de la Tabla 6 confirman una correlación positiva débil y **sin significancia estadística** ($p > 0.05$) entre la destreza del estudiante para estructurar prompts y su calificación final en la asignatura . Este hallazgo fundamenta de manera empírica la **"ilusión de competencia"** sugerida por Denny et al. (2023). El estudiante puede poseer el conocimiento procedimental para generar código funcional mediante instrucciones adecuadas, pero la comprensión lógica profunda del diseño de software y de los pilares de la POO sigue dependiendo de su capacidad analítica autónoma. La competencia lógica humana se posiciona así como una variable moderadora restrictiva e insustituible.

Análisis de la Calidad de Código en Programación Orientada a Objetos (Objetivo Específico 5)

Para evaluar el impacto arquitectónico real en la codificación, se midieron los entregables finales del grupo experimental mediante rúbricas analíticas en tres dimensiones de la POO

Tabla 7

Estadísticos descriptivos de las dimensiones de calidad de código en POO .

Dimensión de Calidad en POO	Media (*M*)	Mediana (*Mdn*)	Desviación Estándar (*SD*)	Mínimo (*Min*)	Máximo (*Max*)
Abstracción (Jerarquías y clases / Máx: 10)	8.65	9.00	0.83	7.00	10.00
Encapsulamiento (Ocultamiento y modificadores / Máx: 5)	4.96	5.00	0.19	4.00	5.00
Polimorfismo (Implementación lógica / Máx: 5)	4.80	5.00	0.40	4.00	5.00
Nota de Código Global (Evaluación integral / Máx: 20)	18.41	19.00	1.12	16.00	20.00
Promedio Final del Curso (Calificación final / Máx: 20)	16.30	16.00	1.14	15.00	19.00

Nota. $N = 27$ (grupo experimental). Evaluado mediante rúbrica analítica directa en el IDE de desarrollo

El desglose de la Tabla 7 refleja una calidad técnica sobresaliente, con una Nota de Código Global promedio de $M = 18.41$ sobre 20. Bajo el enfoque constructivista de Vygotsky, la asistencia del LLM libera carga cognitiva repetitiva (sintáctica), permitiendo que el alumno canalice su esfuerzo hacia el diseño de jerarquías de clases adecuadas (Abstracción, $M = 8.65$) y asegure el correcto ocultamiento de información (Encapsulamiento, $M = 4.96$, con una dispersión mínima de $SD = 0.19$) mediante modificadores de acceso precisos.

Evaluación de Satisfacción General (Modelo TAM)

La actitud de uso se consolida de manera favorable, confirmando los preceptos de aceptación tecnológica

Tabla 8

Distribución de frecuencias sobre satisfacción y percepción de aprendizaje.

Reactivos de Satisfacción (Likert 1-5)	1 - TD	2 - D	3 - N	4 - A	5 - TA	M Media	SD Desv.
Ítem 9: Satisfacción general con la herramienta	0.00%	11.11%	14.81%	51.85%	22.22%	3.85	0.95
Ítem 10: El asistente mejoró mi aprendizaje	0.00%	11.11%	18.52%	37.04%	33.33%	3.93	1.00
Ítem 11: Recomendaría el asistente virtual	3.70%	14.81%	14.81%	40.74%	25.93%	3.70	1.14

Nota. N = 27. Categorías simplificadas: TD (Totalmente en desacuerdo), D (En desacuerdo), N (Neutral), A (De acuerdo), TA (Totalmente de acuerdo).

La Tabla 8 consolida una alta satisfacción general del **74.07%** (Ítem 9) y una percepción de mejora cognitiva del **70.37%** (Ítem 10). No obstante, el 11.11% de desacuerdo advierte sobre la brecha de frustración técnica que pueden inducir las alucinaciones temporales del modelo, demandando una sólida inducción por parte del docente

DISCUSIÓN

Los hallazgos de este estudio demuestran que la integración de asistentes virtuales basados en Inteligencia Artificial Generativa (GitHub Copilot) actúa como un catalizador en el proceso de enseñanza-aprendizaje de la programación, logrando una diferencia neta y estadísticamente significativa de 3.08 puntos a favor del grupo experimental ($t(52) = 9.45$, $p < 0.001$) . Este comportamiento se alinea de manera consistente con lo reportado por Finnie-Ansley et al. (2022), quienes sustentan que los modelos de lenguaje reducen la sobrecarga cognitiva sintáctica en paradigmas de programación compleja. Un hallazgo de alto impacto reside en la reducción de la desviación estándar (de 1.25 a 1.14) en el grupo experimental, demostrando empíricamente un "efecto ecualizador" capaz de homogeneizar

positivamente las competencias de los estudiantes, mitigando las brechas de fracaso escolar tradicional en la EPISI.

No obstante, al profundizar en la Ingeniería de Prompts, la total ausencia de correlación lineal significativa entre la calidad del prompt y el promedio final del estudiante ($r = 0.277$, $p = 0.162$) aporta evidencia científica clave a la discusión de la "ilusión de competencia" descrita por Denny et al. (2023). Sostener una buena sintaxis de prompting permite obtener códigos funcionales rápidos, pero no asegura la asimilación conceptual de la lógica arquitectónica.. El pensamiento computacional humano e independiente se mantiene como una variable moderadora restrictiva e insustituible; un estudiante sin una sólida base de algoritmos lógicos carecerá del criterio metodológico para depurar, refactorizar e integrar las propuestas autogeneradas por la IA dentro de un diseño de software complejo orientado a objetos.

CONCLUSIONES

1. **La integración de asistentes virtuales de Inteligencia Artificial en la enseñanza de la Programación Orientada a Objetos eleva de manera estadísticamente muy significativa el rendimiento académico final de los estudiantes.** Este impacto se evidencia en una ganancia neta de 3.08 puntos en la calificación promedio del grupo experimental ($M = 16.30$, $SD = 1.14$) frente al control ($M = 13.22$, $SD = 1.25$), con una reducción del 8.8% en la desviación estándar, operando el asistente como un factor ecualizador que homogeneiza positivamente el aprendizaje y disminuye las brechas de fracaso en la EPISI.

2. **Existe un nivel de aceptación tecnológica sobresaliente y una percepción altamente favorable entre los estudiantes respecto a la utilidad pedagógica del asistente virtual.** El 74.07% del grupo experimental manifiesta una alta satisfacción general con el uso de la herramienta ($M = 3.85$, $SD = 0.95$) y un 70.37% percibe que su incorporación mejoró de manera tangible su aprendizaje de la programación ($M = 3.93$, $SD = 1.00$).

3. **Las utilidades de los asistentes virtuales más valoradas por los estudiantes son aquellas que actúan como andamiaje explicativo y soporte aclaratorio ante bloqueos de lógica, por encima de la generación automática de código.** Esto se demuestra al registrarse un contundente 88.89% de aprobación en el uso de la IA para la resolución de dudas ($M = 4.33$, $SD = 0.88$) y un 81.48% en su rol

para la comprensión de conceptos teóricos complejos de la POO ($M = 4.22$, $SD = 0.85$).

4. **La habilidad técnica para formular, detallar y estructurar instrucciones (Ingeniería de Prompts) no guarda una relación lineal directa ni estadísticamente significativa con la calificación final obtenida en el curso.** Los análisis bivariados confirman una correlación positiva débil y no significativa ($r = 0.277, p = 0.162$ y $\rho = 0.236, p = 0.236$), demostrando empíricamente que la competencia lógica y algorítmica autónoma del estudiante sigue siendo el factor moderador e insustituible para el éxito académico.

5. **El uso de asistentes de IA generativa propicia una mejora notable en la calidad técnica y el rigor arquitectónico del código desarrollado bajo el paradigma orientado a objetos.** El grupo experimental alcanzó medias sobresalientes en Abstracción ($M = 8.65$, $SD = 0.83$), Encapsulamiento ($M = 4.96$, $SD = 0.19$) y Funcionalidad/Polimorfismo ($M = 4.80$, $SD = 0.40$), consolidando una Nota de Código Global promedio de 18.41 sobre 20, lo cual confirma que el asistente libera carga cognitiva para permitir una mayor concentración en el diseño arquitectónico de software.

REFERENCIAS BIBLIOGRÁFICAS

Anderson, J. R. (2013). *Cognitive psychology and its implications*. Worth Publishers.

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. En J. A. Larusson & B. White (Eds.), *Learning analytics: From research to practice* (pp. 61–75). Springer

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming is hard — or at least it used to be: Educational opportunities and challenges of AI code generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 500–506. https://doi.org/10.1145/3545945.3569759

Brusilovsky, P., & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. En P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.), *The adaptive web* (pp. 3–53). Springer.

Cariaga, R. (2020). Un marco teórico para analizar el rol docente en entornos de aprendizaje virtualizados. *Revista de Educación a Distancia*, *50*, 1–14.

Chen, M., Li, Y., & O'Rourke, E. (2024). Understanding the reasoning behind students' self-assessments

of ability in introductory computer science courses. *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1*, 1–13. Association for Computing Machinery. https://doi.org/10.1145/3640205.3646215

Chen, C. M., & Duh, L. M. (2008). Personalized web-based tutoring system based on fuzzy item response theory. *Expert Systems with Applications*, *34*(4), 2298–2315.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340. https://doi.org/10.2307/249008

Definición.de. (2023). *Asistente virtual — Qué es, tipos, definición y concepto*. https://definicion.de/asistente-virtual/

Denny, P., Luxton-Reilly, A., & Prather, J. (2023). Evaluating the effectiveness of large language models on CS1 coding problems. *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER '23)*, *1*, 118–130. https://doi.org/10.1145/3568846.3587677

Dillenbourg, P., Järvelä, S., & Fischer, F. (2009). The evolution of research on computer-supported collaborative learning. En N. Balacheff, S. Ludvigsen, T. de Jong, A. Lazonder & S. Barnes (Eds.), *Technology-enhanced learning* (pp. 3–19). Springer.

Escuelas de Lenguaje. (2023). *¿Cuáles son los tipos de asistentes?*. https://escuelasdelenguaje.com/cuales-son-los-tipos-de-asistentes/

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. *Proceedings of the 24th Australasian Computing Education Conference (ACE '22)*, 10–19. https://doi.org/10.1145/3511861.3511863

García-Peñalvo, F. J. (2021). Beneficios y desafíos de los asistentes virtuales en el aprendizaje. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, *26*, 45–58.

Graesser, A. C., & McNamara, D. S. (2010). Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist*, *45*(4), 234–244.

Hattie, J. (2009). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. Routledge.

Johnson, W. L., & Lester, J. C. (2016). Face-to-face interaction with pedagogical agents, twenty years later. *International Journal of Artificial Intelligence in Education*, *26*(1), 25–36.

Kay, J., & Kummerfeld, B. (2012). Creating personalized systems that people can scrutinize and control: Drivers, principles and experience. *ACM Transactions on Interactive Intelligent Systems*, *2*(4), 1–42.

Kazemitabaar, M., Chow, J., Ma, C. K. A., Barbara, E., Weintrop, D., & Grossman, T. (2023). Studying the effect of AI Code Generators on supporting novice learners in introductory programming. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3544548.3580919

Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning sciences to the classroom. En R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61–78). Cambridge University Press.

Luckin, R. (2010). *Re-designing learning contexts: Technology-rich, learner-centred ecologies*. Routledge.

Mitrovic, A., Martin, B., & Mayo, M. (2002). Using evaluation to shape ITS design: Results and experiences with SQL-Tutor. *International Journal of Artificial Intelligence in Education*, *13*(2–4), 243–257.

Nye, B. D. (2015). Intelligent tutoring systems by and for the developing world: A review of trends and opportunities in the global south. *International Journal of Artificial Intelligence in Education*, *25*(2), 177–203.

Onrubia, J. (2016). Aprender y enseñar en entornos virtuales: Actividad conjunta, ayuda pedagógica y construcción del conocimiento. *Revista de Educación a Distancia*, *50*, 1–14.

Pardo, A., & Siemens, G. (2014). Ethical and privacy principles for learning analytics. *British Journal of Educational Technology*, *45*(3), 438–450.

Pérez, M. del S., Villalonga, C., Guillén, A., & Baños, O. (2020). Análisis del uso de asistentes virtuales en el aula como recurso complementario en la práctica docente. *Revista de Educación a Distancia*, *50*, 1–14.

Prather, J., Liu, B., Finnie-Ansley, J., Garza, A., & Becker, B. A. (2023). "It's weird that it knows what

I want to write": Investigating novice programmers' interactions with GitHub Copilot. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3544548.3580934

Rey Valzacchi, J. (2024). *Aprendizaje ilimitado: Potenciando la educación con ChatGPT y DALL-E: Una exploración pragmática de la IA en la educación*. Aprende Virtual — Instituto Latinoamericano de Desarrollo Profesional Docente.

Rodríguez Esquíroz, L. (2024). *La conveniencia de asistentes virtuales en educación* [Tesis de maestría, Universidad Pública de Navarra]. Repositorio de la Universidad Pública de Navarra.

Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. En P. J. Durlach & A. M. Lesgold (Eds.), *Adaptive technologies for training and education* (pp. 7–27). Cambridge University Press.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory into Practice*, *41*(2), 64–70.

Zualkernan, I. A. (2006). A framework and a methodology for developing authentic constructivist e-learning environments. *Educational Technology & Society*, *9*(2), 198–212.

Zumbach, J., & Reimann, P. (2002). Supporting distributed problem-based learning: The use of feedback mechanisms in online learning. En G. Stahl (Ed.), *Computer support for collaborative learning* (pp. 51–60). Lawrence Erlbaum Associates.