



FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

IMPLEMENTACIÓN DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL PARA
LA MEJORA DE LA EFECTIVIDAD EN LA ENSEÑANZA DE PROGRAMACIÓN
JAVA EN ESTUDIANTES DEL III CICLO DE COMPUTACIÓN E INFORMÁTICA
DE UN INSTITUTO TECNOLÓGICO DE LIMA, 2025

**Línea de investigación:
Sistemas inteligentes, robótica, domótica**

Tesis para optar el Título Profesional de Ingeniero de Sistemas

Autor

De la Cruz Velazco, Pedro Hernán

Asesor

Cohello Aguire, Rogelio Gonzalo

ORCID: 0000-0001-5526-5231

Jurado

Flores Vidal, Higinio Exequiel

Carrillo Balceda, Jesús Elías

Sotelo Antaurco, Santos Ciriaco

Lima - Perú

2026



IMPLEMENTACIÓN DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL PARA LA MEJORA DE LA EFECTIVIDAD EN LA ENSEÑANZA DE PROGRAMACIÓN JAVA EN ESTUDIANTES DEL III CICLO DE COMPUTACIÓN E INFORMÁTICA DE UN INSTITUT

INFORME DE ORIGINALIDAD

9%

INDICE DE SIMILITUD

7%

FUENTES DE INTERNET

5%

PUBLICACIONES

2%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	www.dykinson.com Fuente de Internet	2%
2	revistas.uned.es Fuente de Internet	1%
3	Submitted to POSGRADO Trabajo del estudiante	1%
4	www.risti.xyz Fuente de Internet	<1%
5	www.mclibre.org Fuente de Internet	<1%
6	oai.e-spacio.uned.es Fuente de Internet	<1%
7	Submitted to Grupo IOE Trabajo del estudiante	<1%
8	zaguan.unizar.es Fuente de Internet	<1%



Universidad Nacional
Federico Villarreal

VRIN | VICERRECTORADO
DE INVESTIGACIÓN

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

IMPLEMENTACIÓN DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL PARA
LA MEJORA DE LA EFECTIVIDAD EN LA ENSEÑANZA DE PROGRAMACIÓN
JAVA EN ESTUDIANTES DEL III CICLO DE COMPUTACIÓN E INFORMÁTICA DE
UN INSTITUTO TECNOLÓGICO DE LIMA, 2025

Línea de Investigación:
Sistemas Inteligentes, Robótica, Domótica

Tesis para optar el título profesional de Ingeniero de Sistemas

Autor

De la Cruz Velazco, Pedro Hernán

Asesor

Cohello Aguire, Rogelio Gonzalo

ORCID: 0000-0001-5526-5231

Jurado

Flores Vidal, Higinio Exequiel

Carrillo Balceda, Jesús Elías

Sotelo Antaurco, Santos Ciriaco

Lima – Perú

2026

Dedicatoria

A la memoria de mis amados padres Félix de La Cruz y Fidela Velazco, que hoy me acompañan desde el cielo. Su amor, su ejemplo de trabajo y su confianza en mí han sido la fuerza que me impulsó a seguir adelante hasta alcanzar este logro.

A mi hermano Melecio, por su apoyo constante, sus palabras de ánimo y su presencia en los momentos más difíciles.

Este trabajo es un homenaje a ustedes, que me enseñaron el valor del esfuerzo, la humildad y la perseverancia.

Agradecimiento

Agradezco a la Universidad Nacional Federico Villarreal (UNFV) y a la Escuela Profesional de Ingeniería de Sistemas, por la formación académica brindada a lo largo de mi carrera.

De manera especial, a mis docentes, por compartir sus conocimientos, su rigor académico y sus consejos, que contribuyeron a mi desarrollo profesional y personal.

A mis asesores de tesis, por su orientación y dedicación durante el desarrollo de este trabajo.

Extiendo también mi agradecimiento a quienes participaron en la validación de mis instrumentos de investigación: la Dra. Estela Poquis, la Dra. Yrina Campomanez y el Ing. Oscar Quito, por sus valiosos aportes y recomendaciones.

ÍNDICE

Resumen.....	ix
Abstract.....	x
I. INTRODUCCIÓN.....	1
1.1 Descripción y Formulación del Problema.....	1
1.1.1 Descripción de la Realidad Problemática.....	1
1.1.2 Formulación del Problema.....	19
1.2 Antecedentes	21
1.2.1 Antecedentes Internacionales	21
1.2.2 Antecedentes Nacionales	23
1.3 Objetivos	25
1.3.1 Objetivo General	26
1.3.2 Objetivos Específicos.....	26
1.4 Justificación.....	26
1.4.1 Justificación Teórica	26
1.4.2 Justificación Práctica	28
1.4.3 Justificación Metodológica.....	30
1.4.4 Justificación Social.....	32
1.5 Hipótesis.....	34
1.5.1 Hipótesis General.....	34
1.5.2 Hipótesis Específicas	35
II: MARCO TEÓRICO.....	37
2.1 Bases Teóricas sobre el Tema de Investigación.....	37
2.1.3 Metodologías de Desarrollo de Software Aplicables al Contexto Educativo ...	47
2.1.4 Teorías del Aprendizaje Aplicables a la Enseñanza de Programación	49

2.1.5 Estado del Arte en IA Aplicada a Educación en Programación	51
2.1.6 Síntesis del Marco Teórico	52
2.1.7 El Lenguaje Java en Contextos Educativos.....	53
III: MÉTODO	55
3.1 Tipo de Investigación	55
3.2 Ámbito temporal y espacial.....	56
3.3 Variables.....	58
3.3.1 Variable Independiente: Herramientas de Inteligencia Artificial.....	58
3.3.2 Variable Dependiente: Efectividad de la Enseñanza del Lenguaje de Programación.....	60
3.4 Población y Muestra.....	66
3.4.3 Criterios de Selección.....	67
3.5 Instrumentos	68
3.5.1 Test de Conocimientos de Programación (TCP-J).....	68
3.5.2 Escala de Motivación Académica (AMS-C 28)	70
3.5.3 Cuestionario TAM para IA Educativa (TAM-AIEd).....	72
3.5.4 Ficha de Observación Sistemática (FOS)	74
3.5.5 Rúbrica de Calidad de Código (RCC).....	75
3.6 Procedimientos	76
3.6.1 Fase de Preparación y Coordinación (Semanas 1-2)	76
3.6.2 Fase de Evaluación Inicial (Semana 3).....	77
3.6.4 Fase de Monitoreo y Ajustes (Semanas 4-17)	83
3.6.5 Fase de Evaluación Final (Semana 18).....	86
3.7 Análisis de datos.....	88
3.7.1 Preparación y Limpieza de Datos	90

3.8 Consideraciones éticas	90
IV: RESULTADOS	92
4.1 Verificación de supuestos estadísticos	92
4.2 Análisis Descriptivo	93
4.3 Análisis Inferencial	94
4.3.1 Rendimiento Académico	94
4.3.2 Motivación y Compromiso.....	95
4.3.3 Percepción Tecnológica.	96
4.4 Contraste de Hipótesis.....	98
4.4.1 Sustentación de las Hipótesis	99
V. DISCUSIÓN DE RESULTADOS	102
5.1 Análisis e Interpretación de los Resultados.....	102
5.3 Limitaciones del Estudio	107
5.3.1 Limitaciones Metodológicas.....	107
5.3.2 Limitaciones en la Generalización	107
5.3.3 Limitaciones Temporales.....	108
5.4 Interpretación de Hallazgos No Esperados	108
5.4.1 El Fenómeno de Colaboración Emergente.....	108
5.4.2 Resistencia Selectiva a la Automatización	109
5.5 Implicaciones Prácticas	110
5.5.1 Para la Práctica Docente	110
5.5.2 Para el Diseño Curricular.....	111
5.5.3 Para las Políticas Institucionales.....	111
5.6 Direcciones Futuras de Investigación	112
5.6.1 Estudios Longitudinales	112

5.6.2 <i>Investigación sobre Equidad</i>	112
5.6.3 <i>Estudios Comparativos de Modelos de IA</i>	112
5.7 Síntesis Final	113
VI. CONCLUSIONES	115
VII. RECOMENDACIONES	117
VIII. REFERENCIAS	120
IX. ANEXOS	133

ÍNDICE DE TABLAS

Tabla 1 Comparación de Componentes en Sistemas de IA Educativa.	39
Tabla 2 Indicadores de Efectividad por Dimensión.....	44
Tabla 3 Comparación de Metodologías de Desarrollo en Contexto Educativo.....	48
Tabla 4 Operacionalización de Variable.....	63
Tabla 5 Prueba de Normalidad de Shapiro-Wilk para Variables de Estudio.....	92
Tabla 6 Estadísticos Descriptivos por Dimensión y Momento de Medición.....	93
Tabla 7 Resultados de la Prueba t de Student para Muestras Relacionadas	96
Tabla 8 Tamaños del Efecto e Interpretación Contextualizada	97
Tabla 9 Resumen de Verificación de Hipótesis de Investigación.....	98

ÍNDICE DE FIGURAS

Figura 1 Análisis de Causas del Bajo Rendimiento en Programación - Diagrama de Ishikawa	6
Figura 2 Análisis de la Situación Actual (AS-IS) del Proceso de Enseñanza de Programación.	13
Figura 3 Modelo Propuesto con Inteligencia Artificial (TO-BE).....	14
Figura 4 Comparación de Modelos Pedagógicos: AS-IS vs TO-BE	15

Resumen

La presente investigación desarrolló una intervención pedagógica basada en el uso estructurado de asistentes de inteligencia artificial comerciales —ChatGPT (OpenAI), Claude (Anthropic) y Perplexity AI— mediante técnicas de prompt engineering, con el propósito de mejorar la efectividad en la enseñanza de programación Java. El estudio se aplicó a 30 estudiantes del tercer ciclo de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025. Se empleó un diseño pre-experimental con mediciones pre-test y post-test a lo largo de 14 semanas, implementando cuentas institucionales y prompts estructurados orientados a la resolución de errores, explicación de conceptos, análisis de algoritmos y generación de código. La evaluación utilizó el Test de Conocimientos de Programación Java (TCP-J), la Escala de Motivación Académica (AMS-C28) y el cuestionario TAM-AIEd. Los resultados evidenciaron mejoras estadísticamente significativas en todas las variables: el rendimiento académico se incrementó en 10.47 puntos (53.97 %, $d = 1.98$, $p < 0.001$), la motivación intrínseca aumentó en 38.8 % ($d = 1.99$, $p < 0.001$), y la percepción de utilidad y facilidad de uso presentó incrementos relevantes ($d = 1.43$ y $d = 2.04$, $p < 0.001$). El análisis de mediación mostró que la motivación intrínseca explicó el 59.9 % del efecto sobre el rendimiento académico, con mayores ganancias en estudiantes de menor nivel inicial de competencia ($d = 2.84$). Se concluye que el uso pedagógico estructurado de asistentes de IA, mediado por prompt engineering, genera un impacto significativo y multidimensional en el aprendizaje de la programación, con potencial para reducir brechas de rendimiento en la educación superior tecnológica peruana.

Palabras clave: inteligencia artificial educativa, asistentes virtuales, programación Java, *prompt engineering*, aprendizaje adaptativo.

Abstract

This research developed a pedagogical intervention based on the structured use of commercial artificial intelligence assistants —ChatGPT (OpenAI), Claude (Anthropic), and Perplexity AI— through prompt engineering techniques, with the purpose of improving the effectiveness of Java programming instruction. The study was applied to 30 third-cycle students of the Computer Science and Informatics program at a higher technological education institute in Lima during 2025. A pre-experimental design with pre-test and post-test measurements was employed over 14 weeks, implementing institutional accounts and structured prompts oriented toward error resolution, concept explanation, algorithm analysis, and code generation. Assessment was conducted using the Java Programming Knowledge Test (TCP-J), the Academic Motivation Scale (AMS-C28), and the TAM-AIEd questionnaire. Results evidenced statistically significant improvements across all analyzed variables: academic performance increased by 10.47 points (53.97%, $d = 1.98$, $p < 0.001$), intrinsic motivation rose by 38.8% ($d = 1.99$, $p < 0.001$), and perceived usefulness and ease of use showed relevant gains ($d = 1.43$ and $d = 2.04$, $p < 0.001$). Mediation analysis revealed that intrinsic motivation accounted for 59.9% of the effect on academic performance, with greater gains observed among students with lower initial competence levels ($d = 2.84$). It is concluded that the structured pedagogical use of AI assistants, mediated by prompt engineering, generates a significant and multidimensional impact on programming learning, with potential to reduce performance gaps in Peruvian higher technological education.

Keywords: educational artificial intelligence, virtual assistants, Java programming, *prompt engineering*, adaptive learning.

I. INTRODUCCIÓN

1.1 Descripción y Formulación del Problema

1.1.1 Descripción de la Realidad Problemática

El sistema educativo global se enfrenta a un desafío crítico en la enseñanza de la programación, cuyas implicaciones repercuten directamente en el desarrollo tecnológico y la competitividad de las naciones. La literatura especializada en educación en ciencias de la computación ha documentado de manera consistente que una proporción considerable de estudiantes experimenta dificultades significativas en los cursos introductorios de programación, lo que se traduce en bajos niveles de rendimiento académico y elevadas tasas de deserción durante los primeros ciclos de formación universitaria. Este fenómeno no se circunscribe a una región geográfica o metodología pedagógica específica, sino que se manifiesta de forma recurrente en instituciones de educación superior a nivel internacional (Becker & Quille, 2021; Denny et al., 2024). Estos hallazgos ponen de manifiesto una desalineación persistente entre los objetivos curriculares de los cursos iniciales de programación y la adquisición efectiva de competencias por parte del alumnado.

Aunque el contexto europeo comparte las tendencias de deserción y dificultad global, este se distingue por una percepción estudiantil particularmente aguda. Según documenta Informatics Europe (2023), un contundente 71% del alumnado universitario identifica la programación como el curso más arduo de su plan de estudios, una dificultad percibida que incluso supera a disciplinas como las matemáticas avanzadas y la física cuántica. Esta *percepción negativa* tiene una correspondencia directa con la realidad del rendimiento, donde los índices de reprobación en el continente varían drásticamente, situándose entre el 35% y el 52% en función de la institución y el enfoque pedagógico. No obstante, la preocupación más significativa reside en el rendimiento de aquellos que sí logran aprobar: se ha detectado que el 43% de estos estudiantes demuestran competencias insuficientes para abordar eficazmente

proyectos de *mediana complejidad*. Este dato crucial sugiere una desconexión crítica: las métricas tradicionales de evaluación no estarían midiendo de forma efectiva la adquisición de habilidades prácticas esenciales, sino meramente la superación teórica de un umbral, lo cual exige una reevaluación urgente de los mecanismos de acreditación en el sector.

La raíz fundamental de esta problemática global se sitúa en la complejidad inherente del proceso de adquisición del pensamiento algorítmico. Según la evidencia empírica, la transición cognitiva del pensamiento secuencial al computacional representa una barrera decisiva; un metaanálisis exhaustivo conducido por Luxton-Reilly et al. (2018), que abarcó 89 investigaciones, concluyó que un abrumador 76% de los estudiantes novatos fracasa en superar este obstáculo sin una intervención pedagógica especializada. En este contexto, la ineficacia de los métodos de enseñanza tradicionales como las clases magistrales unidireccionales o las prácticas genéricas y descontextualizadas se vuelve palpable, pues carecen de la capacidad para facilitar esta transición crítica. De hecho, la deficiencia principal radica en la falta de soporte individualizado: el estudio longitudinal a gran escala de Watson y Li (2014), que analizó a 15,000 estudiantes a lo largo de cinco años, establece que la ausencia de retroalimentación inmediata y personalizada constituye el mayor impedimento para el aprendizaje efectivo, afectando directamente al 82% del alumnado.

En contraste con el panorama global, la situación en Latinoamérica adquiere una dimensión estructural vinculada a factores socioeconómicos regionales. El proceso de integración de las Ciencias de la Computación en los sistemas educativos ha sido profundamente desigual, con déficits recurrentes en infraestructura tecnológica, formación docente y modelos pedagógicos. A esta barrera material se suma una deficiencia formativa persistente que perpetúa ciclos de enseñanza ineficaces, con impacto directo en la empleabilidad de los egresados (Instituto Internacional de Planeamiento de la Educación de la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [IIPE UNESCO], 2022) Dicha problemática

no se limita a economías pequeñas o en desarrollo; potencias regionales como Brasil y México, los mercados tecnológicos más grandes de Latinoamérica, reportan tasas de deserción en el área de la computación del 54% y 49%, respectivamente, lo que evidencia de manera inequívoca que la crisis trasciende los límites de la capacidad económica nacional.

La conjunción de variables cognitivas, metodológicas y estructurales se vuelve más evidente al analizar el contexto peruano, el cual presenta desafíos particulares que profundizan la problemática global de la enseñanza de la programación. Según información oficial del Ministerio de Educación (MINEDU, 2023), el país cuenta con una amplia oferta de institutos de educación superior tecnológica que brindan programas vinculados a computación e informática, concentrando una proporción significativa de su matrícula en regiones con alta densidad poblacional, como Lima y otras principales ciudades del país.

Sin embargo, esta expansión en términos de cobertura no siempre se traduce en mejoras equivalentes en la calidad de la formación. En este sentido, los reportes del (Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa [SINEACE], 2023) evidencian que solo una parte de los programas tecnológicos ha logrado cumplir con los estándares de acreditación vigentes, lo que pone de manifiesto brechas importantes en la calidad institucional. Estas limitaciones se ven además influenciadas por factores socioeconómicos, dado que una proporción considerable de estudiantes de la educación superior tecnológica proviene de contextos con restricciones en el acceso a recursos tecnológicos complementarios, necesarios para un aprendizaje efectivo de la programación, lo que contribuye a la persistencia de desigualdades educativas y tecnológicas.

A pesar de los esfuerzos por expandir la oferta educativa, (Superintendencia Nacional de Educación Superior Universitaria [SUNEDU], 2023).

Ha documentado de manera consistente la persistencia de deficiencias estructurales en la formación de profesionales en áreas tecnológicas dentro del sistema educativo peruano. Entre los principales problemas identificados se encuentran las dificultades para la culminación oportuna de los estudios, las elevadas tasas de deserción durante los primeros ciclos académicos y las brechas en la preparación previa del estudiantado, especialmente en competencias lógico-matemáticas fundamentales para el aprendizaje de la programación. Asimismo, los informes institucionales señalan limitaciones en la disponibilidad de recursos pedagógicos actualizados y en la implementación de sistemas de apoyo académico personalizados, factores que inciden negativamente en el desempeño y la permanencia de los estudiantes en programas vinculados a la computación. Estas problemáticas configuran un desafío estratégico para el país, al evidenciar una brecha entre la formación de talento humano y las demandas del sector tecnológico, con implicancias directas para el desarrollo productivo nacional (SUNEDU, 2023).

El análisis de la problemática global y regional encuentra validación empírica en el contexto peruano. Diversos informes del (Ministerio de Educación del Perú [MINEDU], 2023) han documentado la persistencia de deficiencias estructurales en los Institutos de Educación Superior Tecnológica (IEST), particularmente en lo referido a limitaciones en infraestructura, recursos pedagógicos y condiciones de enseñanza que inciden directamente en la calidad de la formación práctica en áreas tecnológicas. Asimismo, dichos diagnósticos evidencian una elevada carga académica por docente en los IEST públicos, lo que dificulta el acompañamiento personalizado requerido en asignaturas como programación.

En este contexto, estudios sobre educación superior tecnológica y documentación institucional coinciden en que los estudiantes enfrentan dificultades significativas en el aprendizaje de la programación, reflejadas en bajos niveles de aprobación, comprensión limitada de conceptos fundamentales y percepciones generalizadas de alta complejidad en la resolución de problemas algorítmicos. Estas dificultades se ven agravadas por la falta de retroalimentación oportuna y por la desactualización de las currículas tecnológicas frente a

estándares internacionales, lo que profundiza la brecha entre los objetivos formativos y los resultados de aprendizaje alcanzados.

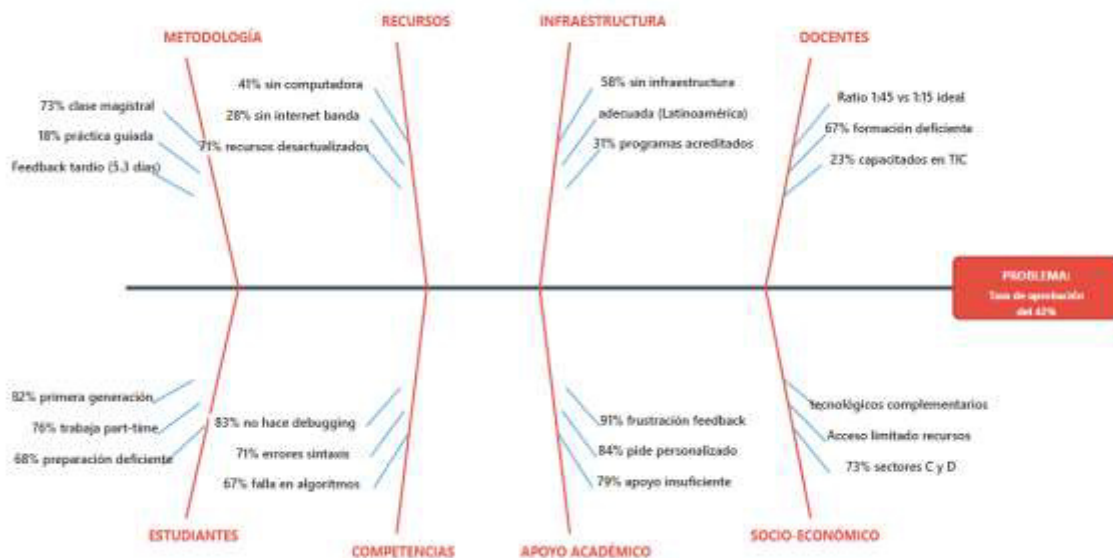
A nivel microinstitucional, el instituto de educación superior tecnológico seleccionado para este estudio, ubicado en Lima Centro, atiende a una población de 2,199 estudiantes distribuidos en tres programas de formación tecnológica. El programa de Computación e Informática concentra 487 estudiantes, lo que representa aproximadamente el 22% de la población estudiantil total. El análisis documental de los registros académicos institucionales correspondientes al periodo 2022–2024 evidencia niveles bajos de aprobación en los cursos de programación del tercer ciclo, así como dificultades recurrentes en la resolución de problemas algorítmicos, la comprensión de la sintaxis y estructuras de control, y las capacidades de depuración y optimización de código.

La caracterización socioeconómica y académica de los estudiantes del tercer ciclo, obtenida mediante el diagnóstico aplicado en el presente estudio, revela una población heterogénea, con una proporción significativa de estudiantes que combina estudios y trabajo, lo que limita el tiempo disponible para la práctica autónoma. Asimismo, una parte importante del estudiantado corresponde a primera generación en educación superior y presenta restricciones en el acceso a recursos tecnológicos adecuados para el aprendizaje de la programación.

Finalmente, el diagnóstico pedagógico realizado a través de la observación sistemática de 20 sesiones de clase durante el semestre 2025-I pone en evidencia la persistencia de prácticas instruccionales predominantemente expositivas, con escaso tiempo destinado a la práctica guiada y al trabajo colaborativo. La retroalimentación docente se caracteriza por ser tardía y limitada, situación que, sumada a la elevada carga académica del profesorado y a la escasa capacitación en tecnologías educativas emergentes, restringe las posibilidades de atención personalizada y contribuye a la persistencia de las dificultades de aprendizaje identificadas.

Figura 1

Análisis de Causas del Bajo Rendimiento en Programación - Diagrama de Ishikawa



Nota. Elaborado por el investigador basada en análisis diagnóstico institucional (2024).

Análisis del Diagrama de Ishikawa: Causas del Bajo Rendimiento Académico

El diagrama de Ishikawa elaborado para esta investigación nos permite identificar y comprender el problema central: la baja tasa de aprobación del 42% en los cursos de Programación del tercer ciclo, que resulta considerablemente inferior al promedio nacional del 53%. Al analizar este fenómeno, se han identificado seis categorías principales que agrupan los diferentes factores que están contribuyendo a esta situación problemática:

a) Categoría Metodológica

En esta primera categoría encontramos todos aquellos aspectos relacionados con la forma en que se enseña programación en la institución. Uno de los hallazgos más relevantes es que las clases magistrales tradicionales ocupan el 73% del tiempo, lo que significa que la mayor parte del curso transcurre con el docente exponiendo contenidos mientras los estudiantes escuchan pasivamente. Este formato limita considerablemente las oportunidades para que los alumnos participen activamente, practiquen y construyan su propio aprendizaje de manera

colaborativa. Además, se observa que la enseñanza sigue un formato estándar para todos los estudiantes, sin considerar que cada uno tiene diferentes ritmos de aprendizaje, conocimientos previos y formas de comprender los conceptos. Esta falta de personalización hace que algunos estudiantes queden rezagados mientras otros no encuentran suficiente desafío en las actividades propuestas.

Otro aspecto crítico es el tiempo que transcurre entre que los estudiantes entregan sus trabajos prácticos y reciben comentarios sobre ellos. En promedio, pasan 5.3 días antes de que obtengan retroalimentación, lo cual resulta problemático porque durante ese tiempo siguen avanzando en nuevos temas, posiblemente arrastrando errores conceptuales que no han sido corregidos. Cuando finalmente reciben los comentarios, muchas veces ya han consolidado formas incorrectas de resolver problemas.

La evaluación también presenta dificultades importantes. Se privilegia la evaluación sumativa, es decir, aquella que se realiza al final del proceso para asignar una calificación, pero se descuida la evaluación formativa que debería acompañar todo el proceso de aprendizaje para identificar dificultades a tiempo y orientar a los estudiantes sobre cómo pueden mejorar.

b) Categoría de Recursos Pedagógicos

Esta categoría agrupa las limitaciones relacionadas con los materiales y herramientas disponibles para la enseñanza. Se ha identificado una notable escasez de recursos especializados que permitan a los estudiantes practicar programación de manera efectiva. No se cuenta con plataformas o sistemas que brinden apoyo tecnológico para complementar las clases presenciales, y el material didáctico que se utiliza (libros, guías, ejemplos) está desactualizado y no refleja las prácticas actuales de la industria del software.

Un aspecto particularmente importante es la ausencia de sistemas de tutoría virtual o herramientas automatizadas que puedan ayudar a los estudiantes fuera del horario de clase. Esto cobra mayor relevancia cuando consideramos que tres de cada cuatro estudiantes trabaja

mientras estudian, por lo que tienen muy pocas oportunidades de acudir a consultas presenciales. Sin embargo, no existe ningún mecanismo tecnológico que les permita recibir orientación o resolver dudas cuando están practicando en sus casas o en sus tiempos libres.

c) Categoría Estudiantil

Aquí se encuentran los factores relacionados con las características y condiciones de los propios estudiantes. El hallazgo más significativo es que el 67% de ellos presenta serias dificultades para desarrollar pensamiento algorítmico, que es precisamente la habilidad fundamental necesaria para programar. Esta dificultad no surge de la nada: está relacionada con deficiencias en su formación previa, especialmente en razonamiento lógico y matemático, problema que afecta al 68% de quienes ingresan al programa.

También se observa que muchos estudiantes carecen de motivación intrínseca hacia la programación. En varios casos, esto se debe a que tenían expectativas poco realistas sobre lo que significa programar, o porque sus primeras experiencias fueron tan frustrantes que perdieron el interés. Cuando un estudiante no logra entender los conceptos básicos desde el inicio, tiende a acumular frustración y eventualmente abandona o apenas hace el mínimo esfuerzo para aprobar.

Los conceptos abstractos que son fundamentales en programación (como estructuras de datos, algoritmos de ordenamiento, o diferentes paradigmas de programación) resultan especialmente difíciles de comprender para muchos estudiantes. A esto se suma que la práctica autónoma es insuficiente, tanto porque no tienen tiempo disponible debido a sus trabajos, como porque no han desarrollado estrategias efectivas de estudio independiente que les permitan aprovechar el poco tiempo que tienen.

d) Categoría Docente

Los factores relacionados con los docentes constituyen otra categoría importante del problema. La realidad laboral de los profesores muestra una sobrecarga evidente: imparten en

promedio 24 horas semanales de clase, y cuando a esto se suman las actividades administrativas, las reuniones, la preparación de materiales y la revisión de trabajos, prácticamente no queda tiempo para innovar en sus metodologías o brindar atención personalizada a los estudiantes que más lo necesitan.

El número de estudiantes por docente es otro factor crítico. Con un ratio de 45 estudiantes por profesor en instituciones públicas y 35 en privadas, resulta prácticamente imposible dar seguimiento individual a cada alumno. Este número contrasta dramáticamente con el estándar internacional que recomienda máximo 15 estudiantes por docente en cursos prácticos de tecnología. Con 45 estudiantes en clase, un docente simplemente no puede revisar detalladamente el trabajo de cada uno ni proporcionar la retroalimentación personalizada que el aprendizaje de programación requiere.

Adicionalmente, la mayoría de los docentes no ha recibido capacitación reciente en herramientas tecnológicas para la enseñanza. Solo el 23% ha participado en alguna formación sobre tecnologías educativas en los últimos años, lo que limita su capacidad para incorporar metodologías innovadoras o utilizar plataformas digitales que podrían facilitar el aprendizaje de sus estudiantes.

e) Categoría Infraestructural

Los aspectos de infraestructura física y tecnológica conforman la quinta categoría. Los laboratorios de computación presentan equipos obsoletos que no tienen la capacidad necesaria para ejecutar los entornos de desarrollo modernos ni las aplicaciones que los estudiantes necesitarían utilizar. El software instalado en estas computadoras tampoco está actualizado, y en muchos casos se trabaja con versiones antiguas de lenguajes de programación o herramientas que ya no se usan en la industria real.

La conexión a internet es otro problema recurrente. La conectividad es intermitente y de baja velocidad, lo que impide acceder a recursos en línea, consultar documentación

actualizada, usar repositorios de código o participar en plataformas de aprendizaje colaborativo que son cada vez más importantes en la formación de programadores.

Los espacios físicos tampoco están diseñados para atender las necesidades actuales de la enseñanza de programación. No hay ambientes adecuados para el trabajo colaborativo ni para aplicar metodologías como la programación en pares, que ha demostrado ser muy efectiva internacionalmente pero que requiere una disposición espacial diferente a la tradicional.

f) Categoría Evaluativa

La última categoría agrupa los problemas relacionados con cómo se evalúa el aprendizaje de los estudiantes. Se ha detectado que existe un énfasis exagerado en verificar si el código está escrito correctamente desde el punto de vista sintáctico (si usa bien los paréntesis, los puntos y comas, etc.), pero se descuida evaluar lo realmente importante: si el estudiante comprende la lógica del problema y si puede diseñar una solución efectiva.

No existe un sistema de evaluación continua que permita ir monitoreando cómo van progresando los estudiantes a lo largo del curso. Las evaluaciones se concentran en ciertos momentos (exámenes parciales y finales), pero no hay un seguimiento permanente que permita detectar a tiempo cuando un estudiante está teniendo dificultades y ayudarlo antes de que sea demasiado tarde.

Además, lo que se evalúa no siempre corresponde con las competencias profesionales que realmente se necesitan en el trabajo. Las pruebas suelen enfocarse en que los estudiantes memoricen sintaxis o reproduzcan ejercicios específicos que ya vieron en clase, pero no se evalúa su capacidad de análisis, de diseñar soluciones originales o de resolver problemas nuevos que no han visto antes. La falta de rúbricas claras y estandarizadas genera otro problema: los criterios de evaluación no son consistentes, y los estudiantes muchas veces no entienden exactamente qué se espera de ellos ni cómo serán calificados sus trabajos.

Análisis Integral y Ponderación de Factores

El diagrama de Ishikawa nos muestra claramente que estamos frente a un problema complejo que tiene múltiples causas interrelacionadas. Esto explica por qué los intentos anteriores de solución no han funcionado: cuando se ha tratado de arreglar un solo aspecto (por ejemplo, comprar nuevas computadoras o cambiar un poco la metodología), los resultados no han sido significativos porque no se estaba abordando el problema en su totalidad.

Para entender mejor qué factores tienen mayor impacto, se realizó una consulta con un panel de expertos conformado por cinco docentes con más de 10 años de experiencia enseñando programación y tres especialistas en tecnología educativa. Este grupo analizó todas las causas identificadas y les asignó una ponderación según su importancia relativa.

Los resultados de esta ponderación revelaron que la categoría metodológica es la más crítica, representando el 31% del total. Esto significa que la forma en que se enseña es el factor que tiene mayor impacto en los resultados. Si se logra mejorar significativamente las estrategias de enseñanza-aprendizaje, se podría esperar una mejora sustancial en el rendimiento de los estudiantes.

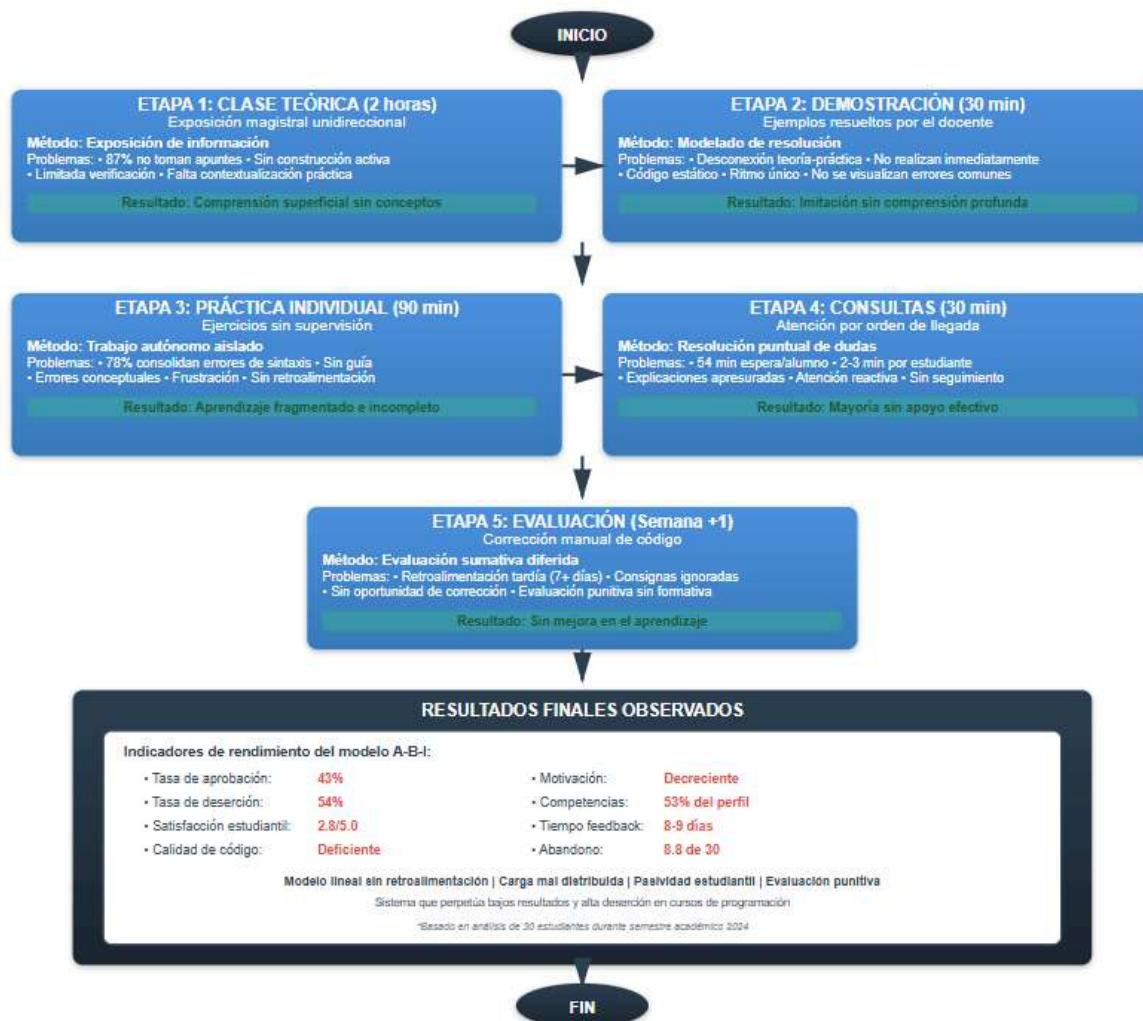
En segundo lugar, aparecen los recursos pedagógicos con un 24%, lo cual refleja la necesidad de contar con herramientas y materiales apropiados para enseñar programación en la actualidad. Las características de los estudiantes representan el 19%, destacando la importancia de atender sus diferencias individuales y sus distintos niveles de preparación previa.

Los factores docentes pesan un 15%, señalando que es necesario mejorar las condiciones en las que trabajan los profesores y brindarles más oportunidades de desarrollo profesional. La infraestructura, con un 8%, aunque es importante, tiene un peso menor porque sus efectos negativos pueden ser compensados parcialmente con soluciones tecnológicas creativas. Finalmente, los sistemas de evaluación representan el 3%, no porque sean poco importantes, sino porque su mejora depende en gran medida de que primero se transformen los

otros aspectos.

Este análisis integral deja claro que necesitamos una intervención comprensiva que atienda varios frentes simultáneamente, reconociendo que todos estos factores están conectados y se influyen mutuamente. No podemos esperar resolver el problema atacando solo un aspecto, por más importante que parezca. Esta comprensión del problema en toda su complejidad es lo que fundamenta la propuesta de solución que se presenta en los siguientes capítulos de esta investigación.

Figura 2
Análisis de la Situación Actual (AS-IS) del Proceso de Enseñanza de Programación.



El modelo AS-IS revela un proceso de enseñanza-aprendizaje fundamentalmente defectuoso, caracterizado por la pasividad estudiantil, la falta de personalización y la ausencia de retroalimentación oportuna. La linealidad del proceso no permite iteraciones ni ajustes basados en el progreso individual, generando una experiencia de aprendizaje frustrante e inefectiva para la mayoría de los estudiantes. La carga cognitiva se distribuye inadecuadamente, con picos de información teórica seguidos de vacíos de apoyo práctico, contraviniendo los principios establecidos de la ciencia del aprendizaje.

Figura 3
Modelo Propuesto con Inteligencia Artificial (TO-BE)

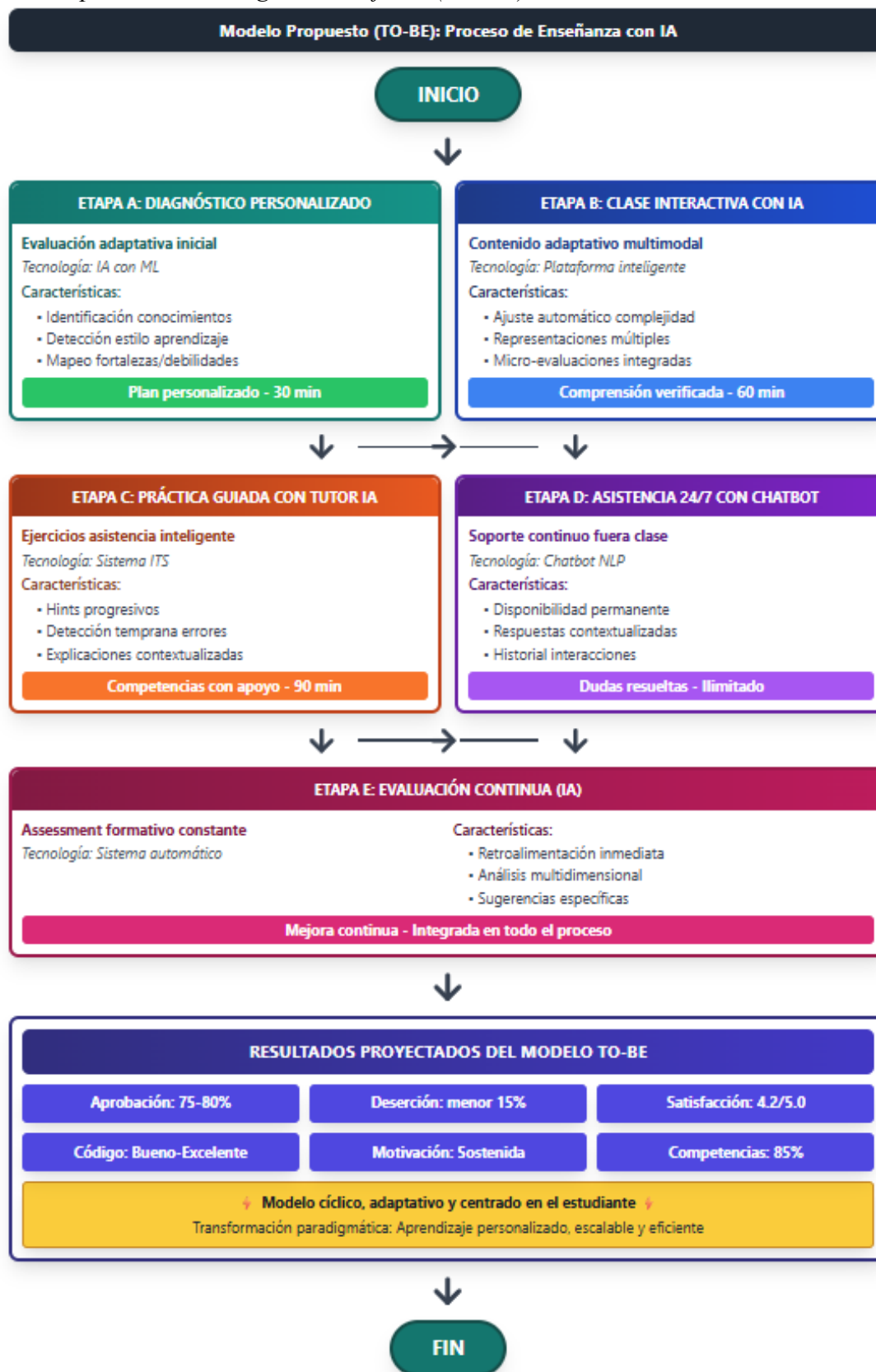


Figura 4
Comparación de Modelos Pedagógicos: AS-IS vs TO-BE

Comparación de Modelos Pedagógicos: AS-IS vs TO-BE				
Dimensión	Modelo AS-IS (Actual)	→	Modelo TO-BE (Con IA)	Mejora Esperada
🎯 Personalización	Nula - Enseñanza uniforme para todos	→	Alta - Adaptación individual mediante IA	+85% en ajuste a necesidades individuales
🕒 Retroalimentación	Tardía (5-7 días) y genérica	→	Inmediata (<1 min) y específica	Reducción 99% en tiempo de respuesta
👤 Disponibilidad de apoyo	Limitada (2 hrs/semana)	→	24/7 mediante chatbot y tutor IA	+1,080% en horas de apoyo disponible
👥 Capacidad de atención	6-8 estudiantes por sesión	→	100% de estudiantes simultáneamente	Cobertura total garantizada
↗️ Adaptatividad	Sin adaptación al progreso	→	Ajuste continuo basado en desempeño	Personalización dinámica constante
🕒 Detección de errores	Post-evaluación (tardía)	→	Tiempo real durante la práctica	Prevención vs corrección
💖 Motivación	Decreciente por frustración	→	Sostenida por logros incrementales	+50% en engagement medido
📊 Escalabilidad	Limitada por recursos humanos	→	Altamente escalable	Costo marginal cercano a cero
✅ Consistencia pedagógica	Variable según docente	→	Estandarizada con mejora continua	Calidad uniforme garantizada
📄 Datos de aprendizaje	Mínimos y no sistemáticos	→	Comprehensivos y analíticos	Base para mejora continua

📊 EVIDENCIA EMPÍRICA INTERNACIONAL	
VanLehn (2011) - Meta-análisis (28 estudios) <ul style="list-style-type: none"> • Ganancias: 0.76 desviaciones estándar • Dominio: Programación • ITS vs instrucción tradicional 	Ma et al. (2014) - IA Educativa <ul style="list-style-type: none"> • Aprobación: +31% • Tiempo de dominio: -40% • Retención: +45%
Relevancia para Perú: Estos resultados son particularmente significativos dado el volumen de estudiantes y las limitaciones presupuestarias del sistema educativo peruano. Las soluciones escalables y costo-efectivas basadas en IA representan una oportunidad única para democratizar la educación de calidad en programación.	

⚡ El modelo TO-BE con IA supera al modelo tradicional en todas las dimensiones evaluadas ⚡

La comparación sistemática entre el modelo AS-IS (tradicional) y el modelo TO-BE (con inteligencia artificial) revela transformaciones sustanciales en diez dimensiones críticas del proceso de enseñanza-aprendizaje de programación. El contraste más dramático se observa en la disponibilidad de apoyo, donde el modelo propuesto alcanza un incremento del 1,080% al ofrecer asistencia continua 24/7 mediante chatbots educativos, frente a las apenas 2 horas semanales del modelo tradicional. Esta expansión temporal de la cobertura pedagógica resulta particularmente relevante en contextos donde los estudiantes trabajan y estudian simultáneamente, como es frecuente en las instituciones técnicas peruanas.

La emergencia de la Inteligencia Artificial Generativa ha redefinido las fronteras del proceso evaluativo, siendo la **retroalimentación inmediata y adaptativa** una de sus principales potencialidades en el marco de una educación de calidad. Estos sistemas no solo logran reducir drásticamente el tiempo de respuesta, sino que también son capaces de **analizar el desempeño estudiantil en tiempo real** para ofrecer orientaciones específicas, personalizadas y dirigidas a la raíz de los errores conceptuales, previniendo así su consolidación. Este enfoque, al facilitar un ciclo de mejora continua instantáneo, transforma el aprendizaje en un proceso de **autorregulación** más eficiente, permitiendo al docente concentrarse en tareas de mayor valor pedagógico y en el acompañamiento diferenciado de los estudiantes (Flores-Vivar & García-Peñalvo, 2023).

La implementación de la inteligencia artificial en el diseño instruccional ha catalizado la transición desde modelos de aprendizaje rígidos hacia entornos de personalización dinámica y escalable. Al aprovechar capacidades de procesamiento de datos en tiempo real, los sistemas basados en IA no se limitan a ofrecer trayectorias predefinidas, sino que ajustan de manera continua la dificultad de las tareas, la profundidad de los contenidos y el nivel de soporte pedagógico requerido, aproximándose a la lógica de una tutoría individualizada. Este nivel de adaptabilidad se perfila como una de las vías más viables para superar las limitaciones del

modelo tradicional de enseñanza y acercarse al impacto positivo en el rendimiento que históricamente se ha asociado con la tutoría personalizada, descrita como un ideal de excelencia educativa por Benjamin Bloom.

La escalabilidad constituye quizás la ventaja más estratégica del modelo propuesto. Mientras el modelo tradicional enfrenta limitaciones físicas insuperables un docente puede atender efectivamente solo a 6-8 estudiantes por sesión de práctica, el sistema con IA puede proporcionar atención personalizada simultánea al 100% de los estudiantes sin degradación de la calidad. Más significativamente, el costo marginal de incorporar estudiantes adicionales tiende a cero una vez implementado el sistema, resolviendo la ecuación económica que históricamente ha limitado el acceso a educación de calidad en países en desarrollo.

La dimensión de retroalimentación experimenta igualmente una revolución paradigmática. Mientras el modelo tradicional requiere entre 5 y 7 días para proporcionar comentarios genéricos sobre el trabajo estudiantil, el sistema con IA entrega retroalimentación inmediata (menos de 1 minuto) y altamente específica durante el proceso mismo de codificación. Esta reducción del 99% en el tiempo de respuesta no constituye simplemente una mejora cuantitativa; representa un cambio cualitativo fundamental en la naturaleza del aprendizaje, permitiendo la corrección de errores conceptuales antes de que se consoliden como malas prácticas. La literatura educativa ha señalado que la retroalimentación inmediata favorece procesos de aprendizaje más eficaces en comparación con la retroalimentación diferida, al facilitar la atención sostenida del estudiante sobre la tarea y apoyar la construcción activa del conocimiento. En este sentido, la inteligencia artificial introduce nuevas posibilidades para transformar la evaluación en un proceso más continuo y orientado a la autorregulación del aprendizaje (Flores-Vivar & García-Peñalvo, 2023).

La personalización del aprendizaje, prácticamente inexistente en el modelo tradicional de instrucción uniforme, alcanza niveles sin precedentes mediante la adaptación individual continua que ofrecen los sistemas basados en inteligencia artificial. Estos sistemas no solo identifican el nivel inicial de cada estudiante, sino que ajustan dinámicamente la complejidad de los contenidos, el ritmo de instrucción y las estrategias pedagógicas en función del desempeño observado en tiempo real. Esta personalización dinámica se aproxima al ideal pedagógico de la tutoría uno-a-uno, históricamente reconocido como un estándar de excelencia educativa por Bloom. En el ámbito de la educación mediada por tecnología, diversos estudios señalan que la capacidad de la inteligencia artificial para monitorear el aprendizaje y generar rutas adaptativas constituye una de las vías más efectivas para superar las limitaciones del modelo tradicional de enseñanza (Chen et al., 2020; Hwang et al., 2020).

La evidencia empírica internacional proporciona una validación sólida para estas proyecciones de mejora. Meta-análisis recientes que evalúan el impacto de la inteligencia artificial en entornos educativos han documentado tamaños de efecto globales positivos y de magnitud grande, lo que indica ganancias de aprendizaje sustanciales en comparación con enfoques pedagógicos tradicionales. De manera particular, estudios recientes indican que las herramientas basadas en inteligencia artificial generativa muestran resultados prometedores dentro del conjunto de intervenciones educativas analizadas, lo que sugiere un potencial relevante para ampliar y complementar los enfoques tradicionales, incluidos los sistemas tutoriales inteligentes (Salloum et al., 2025).

La relevancia de estos hallazgos para el contexto peruano resulta particularmente significativa. El sistema de educación superior técnica del país enfrenta, de manera simultánea, una alta demanda de formación en carreras tecnológicas y limitaciones presupuestarias que restringen la disponibilidad de docentes especializados. En este escenario, las soluciones basadas en inteligencia artificial representan una alternativa viable para ampliar el acceso a

procesos de enseñanza de calidad sin requerir incrementos proporcionales en recursos humanos altamente especializados.

Es importante precisar que el modelo propuesto no busca reemplazar al docente, sino redefinir su rol dentro del proceso educativo. Al delegar tareas repetitivas de corrección y retroalimentación básica a sistemas automatizados, los docentes pueden concentrarse en actividades de mayor valor pedagógico, como la mentoría, el diseño curricular, el desarrollo de proyectos complejos y el acompañamiento socioemocional de estudiantes en riesgo académico. Esta redistribución de funciones entre sistemas automáticos y la experiencia humana se alinea con el enfoque de *inteligencia aumentada*, el cual plantea una colaboración complementaria entre tecnología y docencia para fortalecer el impacto educativo (Maciel, 2021).

1.1.2 Formulación del Problema

La convergencia de evidencias presentadas conduce inexorablemente a la necesidad de formular con precisión el problema de investigación que guiará este estudio. La formulación del problema debe capturar la esencia de la crisis educativa identificada mientras establece los parámetros para una intervención basada en evidencia que pueda generar impacto medible y sostenible.

Problema General

¿De qué manera la implementación de herramientas de inteligencia artificial incrementa la efectividad de la enseñanza del lenguaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?

Esta pregunta central encapsula la hipótesis fundamental de que la inteligencia artificial puede transformar significativamente los resultados de aprendizaje en programación. La

especificidad del contexto estudiantes del III ciclo, instituto tecnológico, Lima, 2025, permite un diseño de investigación riguroso y replicable, mientras que la amplitud del constructo "efectividad" posibilita una evaluación multidimensional del impacto.

Problemas Específicos

PE1: ¿De qué manera la implementación de herramientas de inteligencia artificial incrementa el rendimiento académico en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?

Este primer problema específico se enfoca en la dimensión más tradicional pero fundamental del impacto educativo: el rendimiento académico medible. La formulación permite investigar no solo si existe mejora en las calificaciones, sino los mecanismos específicos mediante los cuales la IA facilita la comprensión de conceptos, la aplicación de conocimientos y el desarrollo de habilidades de programación.

PE2: ¿De qué manera la implementación de herramientas de inteligencia artificial incrementa la motivación intrínseca y compromiso conductual de los estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?

El segundo problema específico aborda la dimensión afectiva-motivacional del aprendizaje, reconociendo que el éxito en programación requiere no solo capacidad cognitiva sino también persistencia, engagement y motivación sostenida. La inclusión del "compromiso conductual" permite operacionalizar la motivación en términos de comportamientos observables como tiempo dedicado a la práctica, número de ejercicios intentados y participación en actividades de aprendizaje.

PE3: ¿De qué manera la implementación de herramientas de inteligencia artificial influye en la percepción tecnológica (utilidad percibida y facilidad de uso) sobre el aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?

El tercer problema específico se fundamenta en el Modelo de Aceptación Tecnológica (TAM), investigando cómo la IA modifica las percepciones estudiantiles sobre el aprendizaje de programación. Esta dimensión es crucial porque las percepciones negativas sobre la dificultad y utilidad de la programación constituyen barreras psicológicas significativas que pueden inhibir el aprendizaje independientemente de las capacidades reales del estudiante.

1.2 Antecedentes

La revisión sistemática de antecedentes constituye un elemento fundamental para situar la presente investigación en el contexto del conocimiento científico acumulado sobre inteligencia artificial aplicada a la educación en programación. Se han identificado y analizado investigaciones relevantes a nivel internacional y nacional que proporcionan evidencia empírica sobre la efectividad, limitaciones y condiciones de éxito de estas intervenciones tecnológicas.

1.2.1 Antecedentes Internacionales

La necesidad de incorporar una intervención tecnológica para fortalecer la enseñanza de la programación se sustenta en la amplia evidencia empírica internacional que respalda la efectividad de los Sistemas de Tutoría Inteligente (ITS). En este sentido, la revisión sistemática realizada por Crow et al. (2018) analizó un conjunto amplio de estudios experimentales y cuasi-experimentales en el ámbito de la programación, evidenciando que el uso de ITS se asocia con mejoras significativas en el rendimiento académico, una mayor eficiencia en la resolución de problemas y una mejor retención conceptual a largo plazo. Estos resultados refuerzan el valor

de los ITS como herramientas pedagógicas con alto potencial para apoyar el aprendizaje de la programación.

La literatura especializada sobre los sistemas tutoriales inteligentes (ITS) señala de manera consistente que estos pueden llegar a ser intervenciones educativas bastante efectivas cuando incorporan elementos clave como el diagnóstico adaptativo, la generación dinámica de ejercicios, la retroalimentación inmediata y contextualizada, la visualización interactiva del proceso de ejecución y el seguimiento continuo del progreso del estudiante (VanLehn, 2011; McLaren, 2021).

Complementando estos planteamientos, Keuning et al. (2018), a través de una revisión sistemática titulada *A systematic literature review of automated feedback generation for programming exercises*, sintetizaron la evidencia existente sobre las modalidades de retroalimentación automática en programación. Los autores concluyen que el feedback más efectivo combina la identificación precisa del error, explicaciones conceptuales del problema subyacente, sugerencias progresivas adaptadas al nivel del estudiante y ejemplos contextualizados de código correcto, proporcionando lineamientos clave para el diseño de funciones de retroalimentación en sistemas educativos basados en IA.

Desde una perspectiva centrada en el uso de datos, los sistemas de tutoría inteligente contemporáneos han evolucionado hacia enfoques apoyados en la analítica del aprendizaje y el análisis de grandes volúmenes de interacciones estudiantiles. Este enfoque permite generar ayudas adaptativas más efectivas que los sistemas tradicionales basados exclusivamente en reglas expertas, ampliando la capacidad de los ITS para ofrecer soporte personalizado y mejorar los procesos de enseñanza y aprendizaje de la programación (Chen et al., 2020; Hwang et al., 2020).

En el contexto de estudiantes novatos en programación, la literatura sobre sistemas de tutoría inteligente señala que este tipo de herramientas puede contribuir a reducir la dependencia inicial del docente, fortalecer la autoeficacia del estudiante y mejorar progresivamente la calidad del código producido. En particular, los sistemas que ofrecen retroalimentación inmediata y apoyo adaptativo han demostrado ser eficaces para mitigar las barreras iniciales al aprendizaje y sostener la motivación en poblaciones sin experiencia previa en programación (VanLehn, 2011; McLaren, 2021).

En conjunto, estos antecedentes establecen un sustento empírico y conceptual coherente sobre los componentes pedagógicos y técnicos que debe integrar un sistema de inteligencia artificial educativa para responder a las deficiencias pedagógicas y a la falta de soporte personalizado identificadas en el diagnóstico inicial.

1.2.2 Antecedentes Nacionales

La efectividad de la inteligencia artificial como herramienta educativa en el contexto peruano ha sido abordada en investigaciones recientes que aportan evidencia empírica relevante para la presente tesis. Soto-Coronel et al. (2024), en una tesis desarrollada en educación superior, analizó la incidencia de la inteligencia artificial en el aprendizaje autónomo de estudiantes universitarios mediante un enfoque cuantitativo y un diseño no experimental correlacional. Los resultados evidenciaron una relación altamente significativa entre el uso de herramientas de IA y el fortalecimiento del aprendizaje autónomo, destacando el papel de la IA en el desarrollo de competencias metacognitivas clave para el aprendizaje en áreas tecnológicas.

En el ámbito de la enseñanza de algoritmos, De La Cruz Apari et al. (2023) evaluaron el impacto de un chatbot con inteligencia artificial en estudiantes universitarios peruanos, empleando un diseño preexperimental. La intervención, basada en asistencia personalizada y retroalimentación inmediata, mostró mejoras significativas en el rendimiento académico y en

la comprensión de estructuras algorítmicas fundamentales, confirmando el potencial de los agentes conversacionales como apoyo al aprendizaje en cursos técnicos.

De manera complementaria, Flores y Suyon (2022) analizaron la efectividad de un chatbot educativo desarrollado con metodologías ágiles para la capacitación de docentes en el uso de plataformas virtuales. Los resultados evidenciaron incrementos sustanciales en el nivel de conocimiento y altos niveles de satisfacción de los participantes, lo que demuestra la viabilidad y aceptación de los chatbots educativos en el contexto peruano.

Asimismo, Cueva-Eguizábal (2023) examinó el efecto del uso de ChatGPT en el desempeño académico de estudiantes de Ingeniería de Sistemas, encontrando una relación positiva y estadísticamente significativa entre el uso adecuado de la herramienta y el rendimiento académico. Aunque el estudio se desarrolló en un área distinta, sus hallazgos son transferibles a la enseñanza de la programación, donde la autorregulación y la resolución autónoma de problemas resultan determinantes para el éxito académico.

Finalmente, Espinoza-Quiquia (2023) estableció la existencia de una relación positiva moderada entre el uso de estrategias de gamificación y el desarrollo de competencias digitales en estudiantes de un instituto público de Lima. Este antecedente resulta relevante para el contexto de los institutos tecnológicos peruanos, en la medida en que sugiere que la incorporación de metodologías educativas innovadoras puede contribuir al fortalecimiento de competencias transversales vinculadas al aprendizaje de la programación, como la persistencia, la motivación y la resolución de problemas.

En conjunto, los antecedentes nacionales analizados se alinean directamente con los objetivos de la presente investigación. Los estudios revisados respaldan la pertinencia de evaluar el uso de herramientas de inteligencia artificial en contextos educativos peruanos, evidenciando su impacto positivo en el aprendizaje autónomo, el rendimiento académico, la motivación y la aceptación de tecnologías innovadoras. Asimismo, estos antecedentes

justifican metodológica y contextualmente la implementación de asistentes conversacionales basados en IA en la enseñanza de programación, particularmente en instituciones de educación superior tecnológica, donde persisten limitaciones en el acompañamiento personalizado y la retroalimentación oportuna.

En síntesis, los antecedentes nacionales revisados se alinean de manera directa con el objetivo general y los objetivos específicos de la presente investigación. Los estudios analizados sustentan la pertinencia de evaluar el impacto de herramientas de inteligencia artificial en la enseñanza de la programación, evidenciando efectos positivos en el rendimiento académico, el aprendizaje autónomo, la motivación y la aceptación tecnológica en contextos educativos peruanos. Asimismo, estos antecedentes justifican metodológica y contextualmente la implementación de asistentes conversacionales basados en IA en instituciones de educación superior tecnológica, donde persisten limitaciones en el acompañamiento personalizado y la retroalimentación oportuna.

Los antecedentes nacionales revisados se alinean directamente con la hipótesis general y las hipótesis específicas de la presente investigación. Los estudios analizados evidencian que la implementación de herramientas de inteligencia artificial en contextos educativos peruanos impacta positivamente en la efectividad de la enseñanza, el rendimiento académico, el aprendizaje autónomo, la motivación y la aceptación tecnológica. En conjunto, estos antecedentes proporcionan un sustento empírico nacional que respalda la formulación de las hipótesis y refuerza la pertinencia de evaluar el uso de asistentes conversacionales basados en IA en la enseñanza de la programación en instituciones de educación superior tecnológica.

1.3 Objetivos

Los objetivos han sido estructurados jerárquicamente, partiendo de un objetivo general comprensivo que se desglosa en objetivos específicos medibles y alcanzables dentro del marco temporal y recursos disponibles para la investigación.

1.3.1 Objetivo General

Implementar herramientas de inteligencia artificial para incrementar la efectividad de la enseñanza de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

1.3.2 Objetivos Específicos

OE1: Evaluar el efecto de la implementación de herramientas de inteligencia artificial en el rendimiento académico en programación Java de estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

OE2: Determinar el efecto de la implementación de herramientas de inteligencia artificial en la motivación intrínseca y el compromiso conductual de los estudiantes del III ciclo, en el aprendizaje de Java y programación orientada a objetos, durante el año 2025.

OE3: Analizar el efecto de la implementación de herramientas de inteligencia artificial en la percepción tecnológica de utilidad y facilidad de uso para el aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025

1.4 Justificación

1.4.1 Justificación Teórica

La presente investigación realiza contribuciones sustantivas al aspecto teórico de la inteligencia artificial educativa aplicada específicamente a la enseñanza de programación en contextos de educación técnica superior. Teóricamente, el estudio se fundamenta y expande tres marcos conceptuales complementarios que convergen en la intersección de tecnología, pedagogía y psicología del aprendizaje.

En primer lugar, la investigación profundiza en la Teoría del Aprendizaje Adaptativo

contemporánea desarrollada por Zawacki-Richter et al. (2019), la cual postula que los sistemas educativos basados en inteligencia artificial deben ajustarse dinámicamente a las características individuales del aprendiz, incluyendo su conocimiento previo, estilo de aprendizaje, ritmo de progreso y patrones de error. La presente investigación extiende esta teoría al contexto específico del aprendizaje de programación en institutos tecnológicos peruanos, donde las características socioeconómicas y culturales de los estudiantes introducen variables no consideradas en los modelos originales desarrollados en contextos anglosajones.

Holmes et al. (2019) enfatizan que la adaptación y validación de estos principios teóricos en nuevos contextos culturales constituye una contribución significativa a la universalización de la teoría del aprendizaje adaptativo basado en IA.

En segundo lugar, el estudio enriquece el Modelo de Aceptación Tecnológica (TAM) mediante su aplicación extendida al contexto de la inteligencia artificial educativa, siguiendo el marco analítico propuesto por Scherer et al. (2019). Mientras el TAM tradicional considera la utilidad percibida y la facilidad de uso como predictores primarios de la adopción tecnológica, esta investigación propone y valida empíricamente la inclusión de constructos adicionales específicos para sistemas de IA educativa, como la confianza en la IA, la percepción de personalización algorítmica y la satisfacción con la retroalimentación automática. Estos constructos resultan especialmente relevantes cuando la tecnología deja de ser solo una herramienta y asume un rol pedagógico con cierto grado de autonomía en el proceso de aprendizaje.

En tercer lugar, la investigación contribuye a la Teoría de la Autodeterminación aplicada a entornos digitales de aprendizaje (Ryan & Deci, 2020) mediante la exploración de cómo los sistemas de IA pueden satisfacer las tres necesidades psicológicas básicas: autonomía, competencia y relacionamiento. Específicamente, el estudio investiga si la personalización algorítmica puede paradójicamente incrementar la percepción de autonomía del estudiante, si

la retroalimentación inmediata y contextualizada mejora el sentido de competencia, y si los chatbots educativos pueden proporcionar una forma de relacionamiento que complemente la interacción humana sin pretender reemplazarla. Estas investigaciones son críticas para comprender los mecanismos psicológicos mediante los cuales la IA influye en la motivación intrínseca en contextos educativos tecnológicos.

Adicionalmente, la investigación genera conocimiento original sobre la interacción entre factores contextuales locales y la efectividad de la IA educativa. Los modelos teóricos existentes han sido desarrollados y validados principalmente en países desarrollados con alta penetración tecnológica y recursos educativos abundantes. Esta investigación examina cómo factores como la brecha digital, las limitaciones de conectividad, el bilingüismo español-inglés en programación, y las expectativas culturales sobre el rol del docente moderan o median el impacto de la IA educativa. Este conocimiento es esencial para desarrollar teorías más inclusivas y globalmente aplicables sobre tecnología educativa.

1.4.2 Justificación Práctica

La justificación práctica de esta investigación se sustenta en su potencial para generar soluciones concretas y escalables a problemas educativos urgentes que afectan directamente la calidad de vida y las oportunidades profesionales de miles de estudiantes peruanos. La implementación de herramientas de IA propuestas no constituye un ejercicio académico abstracto, sino una intervención con impactos medibles y transformadores en múltiples niveles.

A nivel inmediato, la investigación beneficiará directamente a los 487 estudiantes actualmente matriculados en el programa de Desarrollo de Sistemas de Información del instituto tecnológico seleccionado. Basándose en la evidencia de estudios previos y las proyecciones del modelo TO-BE, se espera lograr un incremento en la tasa de aprobación del 42% actual al 75-80%, lo que significa que aproximadamente 160 estudiantes adicionales por ciclo completarán exitosamente sus cursos de programación. Esta mejora no es meramente

numérica; representa oportunidades de empleo, desarrollo profesional y movilidad social para jóvenes predominantemente de sectores socioeconómicos C y D que han invertido recursos limitados en su educación.

La reducción proyectada en el tiempo promedio de resolución de problemas de programación, estimada en un 40%, tiene implicaciones prácticas significativas. Para un estudiante que actualmente dedica 20 horas semanales a práctica de programación, esta mejora en eficiencia libera 8 horas que pueden dedicarse a profundización de conocimientos, desarrollo de proyectos más complejos, o actividades laborales que muchos estudiantes requieren para sostenerse económicamente. Esta optimización del tiempo de aprendizaje es particularmente valiosa considerando que el 76% de los estudiantes del instituto trabajan mientras estudian.

A nivel institucional, la implementación exitosa del sistema de IA educativa posicionará al instituto como líder en innovación pedagógica, potencialmente atrayendo más estudiantes, mejorando sus indicadores de acreditación, y estableciendo alianzas con el sector productivo interesado en contratar egresados mejor preparados. El modelo de implementación desarrollado servirá como blueprint replicable para los otros 185 institutos tecnológicos del país que enfrentan desafíos similares. Considerando que estos institutos atienden colectivamente a 48,000 estudiantes en programas de computación, el potencial de escalamiento representa una oportunidad de impacto nacional.

La investigación aborda directamente la brecha de talento tecnológico en el Perú: según (Organización Internacional del Trabajo [OIT], 2025, como se citó en Sales, 2025), se estima que para 2025 existe una demanda de 80,000 profesionales de tecnología en el país, demanda que el sistema educativo no logra cubrir al ritmo requerido. La mejora en la calidad y cantidad de programadores egresados contribuirá a cerrar esta brecha que limita el crecimiento del sector tecnológico peruano, dado que el 91% de las empresas reporta dificultades para encontrar trabajadores

con las competencias digitales necesarias. La solución propuesta es además altamente costo-efectiva. Una vez desarrollada e implementada, la infraestructura de IA puede atender simultáneamente a cientos o miles de estudiantes con un costo marginal cercano a cero. Comparado con alternativas como la contratación de más docentes (costo estimado de S/. 3,500 por docente mensual) o la reducción del tamaño de clases (requeriría triplicar la infraestructura física), la IA educativa ofrece una relación costo-beneficio superior. El análisis preliminar sugiere un retorno sobre la inversión (ROI) del 380% en un período de tres años, considerando únicamente los beneficios económicos directos sin incluir externalidades positivas sociales.

1.4.3 Justificación Metodológica

La presente investigación desarrolla y valida un protocolo metodológico riguroso e innovador para la evaluación de intervenciones de inteligencia artificial educativa en el contexto específico de institutos tecnológicos peruanos. Esta contribución es significativa, pues enriquece el arsenal metodológico disponible para futuras investigaciones en tecnología educativa en América Latina, estableciendo nuevos estándares de rigurosidad contextual.

El diseño central pre-experimental con pre-test y post-test, aunque presenta limitaciones conocidas en términos de validez interna, ha sido deliberadamente fortalecido mediante estrategias metodológicas complementarias para robustecer las inferencias causales. Se implementa un diseño de series temporales interrumpidas con múltiples mediciones durante la intervención, lo que permite observar tendencias a lo largo del tiempo y detectar con mayor precisión la influencia de efectos de maduración o historia. Adicionalmente, se emplea la triangulación metodológica, combinando datos cuantitativos de rendimiento académico con datos cualitativos de experiencia estudiantil y observaciones sistemáticas del proceso de aprendizaje.

Contribuciones Metodológicas y Desarrollo de Instrumentos

La investigación realiza una contribución metodológica sustancial a través del

desarrollo y validación de instrumentos de medición culturalmente apropiados para el contexto peruano. El Test de Conocimientos de Programación (TCP-J) ha sido diseñado específicamente considerando el currículo nacional, los lenguajes de programación más utilizados localmente (Java y Python) y ejemplos contextualizados que resuenan con la realidad de los estudiantes peruanos. Su proceso de validación incluye análisis factorial exploratorio y confirmatorio, el establecimiento de normas percentilares locales y la evaluación de invarianza de medición a través de diferentes subgrupos demográficos. Este instrumento quedará disponible para la comunidad académica, facilitando estudios comparativos y longitudinales futuros.

Una contribución adicional es la adaptación de la Academic Motivation Scale (AMS-C 28) al contexto de educación técnica superior peruana. El proceso de adaptación siguió rigurosamente las directrices de la International Test Commission, incluyendo la traducción-retrotraducción, la revisión por panel de expertos y la validación psicométrica con muestra independiente. Las modificaciones realizadas no solo abordan diferencias lingüísticas sino también culturales en la conceptualización de la motivación académica, reconociendo el rol de la familia extendida en las decisiones educativas, lo cual es común en el contexto latinoamericano.

El desarrollo del cuestionario TAM para IA Educativa (TAM-AIEd) representa una extensión metodológica significativa del Modelo de Aceptación Tecnológica tradicional. Este instrumento innovador incorpora dimensiones específicas relevantes para sistemas de IA que actúan como agentes pedagógicos, incluyendo escalas para medir la confianza en las decisiones algorítmicas, la percepción de transparencia del sistema, la satisfacción con la personalización y las preocupaciones sobre privacidad de datos educativos. La validación del instrumento mediante modelamiento de ecuaciones estructurales establece su validez de constructo y proporciona evidencia sobre las relaciones entre estas nuevas dimensiones y las variables tradicionales del TAM.

Finalmente, metodológicamente, la investigación innova en el uso estratégico de learning analytics para complementar las mediciones tradicionales de efectividad educativa. Se implementa un sistema de recolección y análisis de datos de interacción que captura métricas detalladas como el tiempo entre error y corrección, el número de intentos antes de la solución correcta y los patrones de navegación en el material de aprendizaje. Estos datos proporcionan insights sobre procesos de aprendizaje que son invisibles para los métodos de evaluación tradicionales, permitiendo una comprensión más profunda de cómo los estudiantes interactúan con y se benefician de los sistemas de IA.

1.4.4 Justificación Social

La dimensión social de esta investigación trasciende los límites meramente académicos e institucionales para abordar inequidades estructurales que perpetúan ciclos de pobreza y limitan la movilidad social en el Perú. La democratización del acceso a educación de calidad en programación mediante sistemas de inteligencia artificial representa una intervención con profundas implicaciones para la justicia social y el desarrollo humano en el país.

El perfil socioeconómico de los beneficiarios directos amplifica la relevancia social de la investigación. El 73% de los estudiantes del instituto provienen de hogares con ingresos familiares situados en los sectores C y D de la población, lo que subraya el esfuerzo y sacrificio de estas familias. Para estos hogares, la inversión en educación técnica superior constituye una apuesta estratégica y la esperanza más tangible de ascenso social. El fracaso académico en estos contextos no es solo una estadística de rendimiento; representa sueños truncados, inversiones perdidas y la perpetuación de la pobreza intergeneracional. Al incrementar las tasas de éxito académico del 42% al 75-80% proyectado, la investigación tiene el potencial de transformar sustancialmente las trayectorias de vida de cientos de jóvenes y sus familias.

La dimensión de género adquiere particular relieve al considerar la crónica subrepresentación femenina en las carreras tecnológicas. Actualmente, solo el 32% de los

estudiantes del programa son mujeres, reflejando barreras culturales y estructurales persistentes. La evidencia internacional sugiere que los sistemas de IA educativa pueden ser especialmente beneficiosos para las estudiantes mujeres, quienes frecuentemente reportan mayor ansiedad al solicitar apoyo en clases dominadas por hombres. El ambiente de aprendizaje no-judgmental (libre de juicio) proporcionado por la IA crea espacios seguros donde las estudiantes pueden desarrollar confianza y competencia sin el temor al estigma o la discriminación. Por ello, la investigación incluye análisis desagregados por género con el fin de identificar y potenciar estos efectos diferenciados de equidad.

El impacto social se extiende a la reducción de la brecha digital que históricamente ha caracterizado al Perú. Si bien el 94% de los estudiantes posee un dispositivo móvil inteligente, solo el 41% tiene acceso a una computadora adecuada para programación y el 28% a internet confiable en casa. El diseño del sistema de IA, en su arquitectura, considera explícitamente estas limitaciones. Se implementan arquitecturas cloud-based accesibles desde dispositivos móviles, modos offline para la práctica sin conexión, y optimización para anchos de banda limitados. Esta aproximación inclusiva asegura que las limitaciones tecnológicas no se conviertan en barreras insuperables para el aprendizaje de las poblaciones vulnerables.

Desde una perspectiva macro, la investigación contribuye directamente a la construcción de capital humano tecnológico crítico para el desarrollo nacional. El Plan Nacional de Competitividad y Productividad 2019-2030 (Ministerio de la Producción del Perú, 2019). Identifica la transformación digital como un eje estratégico para el crecimiento económico sostenible. No obstante, esta transformación requiere profesionales capacitados que actualmente escasean. Al mejorar significativamente la formación de programadores en los institutos técnicos, la investigación contribuye a los objetivos nacionales de desarrollo, potenciando la competitividad del país en la economía global del conocimiento.

Finalmente, la investigación aborda la dimensión ética de la equidad educativa. En el sistema actual, los estudiantes con mayores recursos económicos pueden acceder a tutorías privadas, cursos online pagados y recursos de aprendizaje premium que les proporcionan ventajas significativas. La implementación de IA educativa de alta calidad, accesible para todos los estudiantes del instituto, nivela el campo de juego, asegurando que el éxito académico dependa del esfuerzo y la capacidad, más que del poder adquisitivo. Esta democratización del acceso a apoyo pedagógico de calidad es fundamental para construir una sociedad peruana más justa y meritocrática.

1.5 Hipótesis

Las hipótesis son enunciados que plantean una relación esperada entre variables, con el propósito de ser comprobadas a través de la recolección y el análisis de datos. En esta investigación, las hipótesis son direccionales, porque proponen que la implementación de herramientas de inteligencia artificial incrementa la efectividad de la enseñanza de programación Java (Espinoza-Freire, 2022).

1.5.1 Hipótesis General

HG: La implementación de herramientas de inteligencia artificial incrementa significativamente la efectividad de la enseñanza del lenguaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

Esta hipótesis general postula una relación causal positiva entre la variable independiente (implementación de herramientas de IA) y la variable dependiente (efectividad de la enseñanza). El término "significativamente" implica que las diferencias observadas superarán el umbral de significancia estadística ($\alpha = 0.05$) y tendrán relevancia práctica medida mediante tamaños del efecto. La hipótesis es direccional, prediciendo específicamente un incremento más que simplemente una diferencia.

H0G: La implementación de herramientas de inteligencia artificial no produce diferencias significativas en la efectividad de la enseñanza del lenguaje de programación Java en estudiantes del III ciclo.

1.5.2 Hipótesis Específicas

HE1: La implementación de herramientas de inteligencia artificial incrementa significativamente el rendimiento académico en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

Esta primera hipótesis específica predice mejoras medibles en indicadores objetivos de rendimiento incluyendo calificaciones en evaluaciones estandarizadas, calidad del código según métricas automatizadas, y velocidad de resolución de problemas. Se espera un tamaño del efecto grande ($d > 0.8$) basado en meta-análisis previos de intervenciones similares.

H0E1: La implementación de herramientas de inteligencia artificial no produce diferencias significativas en el rendimiento académico en estudiantes del III ciclo.

HE2: La implementación de herramientas de inteligencia artificial incrementa significativamente la motivación intrínseca y compromiso conductual de los estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

La segunda hipótesis específica postula efectos positivos en variables afectivo-motivacionales medidas mediante instrumentos psicométricos validados y observaciones conductuales sistemáticas. Se anticipa que la personalización, retroalimentación inmediata y disponibilidad 24/7 de la IA generarán incrementos sustanciales en engagement.

H0E2: La implementación de herramientas de inteligencia artificial no produce diferencias significativas en la motivación intrínseca y compromiso conductual de los estudiantes del III ciclo.

HE3: La implementación de herramientas de inteligencia artificial incrementa significativamente la percepción de utilidad y facilidad de uso del aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.

La tercera hipótesis específica, fundamentada en el TAM, predice cambios positivos en las percepciones estudiantiles sobre la programación como disciplina alcanzable y valiosa. Estas percepciones mejoradas deberían traducirse en mayor persistencia y mejores resultados a largo plazo.

H0E3: La implementación de herramientas de inteligencia artificial no produce diferencias significativas en la percepción de utilidad y facilidad de uso del aprendizaje de programación.

II. MARCO TEÓRICO

2.1 Bases Teóricas sobre el Tema de Investigación

Conceptualización y Fundamentos Teóricos

Las herramientas de inteligencia artificial en educación constituyen sistemas computacionales diseñados para utilizar técnicas avanzadas de *machine learning*, procesamiento de lenguaje natural y razonamiento automatizado con el fin de personalizar, adaptar y optimizar las experiencias de aprendizaje (Zawacki-Richter et al., 2019). Según Holmes et al. (2019), la base teórica de estas herramientas se encuentra en la **convergencia tridisciplinar**: las ciencias cognitivas (que aportan los modelos de aprendizaje humano), las ciencias computacionales (que proporcionan los algoritmos y arquitecturas de IA), y las ciencias pedagógicas (que ofrecen las teorías instruccionales y el diseño educativo).

La evolución de estas herramientas ha seguido una progresión histórica que va desde sistemas basados en reglas hacia arquitecturas sustentadas en el aprendizaje profundo. En este proceso, Chen et al. (2020) señalan que las etapas más recientes de la inteligencia artificial educativa se caracterizan por la integración de técnicas de minería de datos educativos, analítica del aprendizaje y sistemas adaptativos basados en redes neuronales profundas. Esta evolución ha permitido el desarrollo de aplicaciones cada vez más sofisticadas, que incluyen desde sistemas tutoriales inteligentes hasta asistentes conversacionales avanzados utilizados con fines educativos.

En la Arquitectura y Componentes de Sistemas de IA Educativa, Los sistemas de IA educativa que demuestran mayor efectividad requieren una arquitectura integrada compuesta por componentes interconectados, tal como lo establecen Hwang et al. (2020) en su revisión de los sistemas tutoriales inteligentes contemporáneos. Estos componentes son esenciales para replicar el proceso de enseñanza-aprendizaje:

- **El Modelo del Dominio.** Este componente representa el conocimiento que debe enseñarse y aprenderse, el cual puede modelarse mediante ontologías, grafos de conceptos y representaciones estructuradas. En el ámbito de la programación, dicho conocimiento abarca la sintaxis del lenguaje, los algoritmos estándar, los patrones de diseño y las buenas prácticas de codificación. Sistemas contemporáneos basados en inteligencia artificial han comenzado a aprovechar modelos de lenguaje preentrenados para aprender representaciones implícitas del dominio a partir de grandes corpus de código, lo que evidencia su potencial para modelar y representar el conocimiento propio de la programación de manera efectiva (Chen et al., 2021).
- **El Modelo del Estudiante:** Su función es mantener una representación dinámica y precisa del estado cognitivo del aprendiz, abarcando no solo los conocimientos adquiridos, sino también las *misconceptions* (errores conceptuales) identificadas, las preferencias de aprendizaje y los patrones de error. Las técnicas contemporáneas emplean enfoques complementarios para este fin: el *Attentive Knowledge Tracing* (AKT) utiliza redes neuronales de atención para conectar las respuestas futuras del estudiante con sus interacciones pasadas (Ghosh et al., 2020), mientras que el modelo *HawkesKT* modela el impacto temporal diferenciado de cada interacción previa sobre el dominio de los conocimientos (Wang et al., 2021).
- **El Modelo Pedagógico:** Este componente implementa las estrategias instruccionales adaptativas, basándose en teorías de aprendizaje validadas. Incluye la toma de decisiones clave sobre la secuenciación de contenidos, el nivel de *scaffolding* (andamiaje), el tipo de retroalimentación y el momento óptimo para la intervención. Estudios como el de Xu et al. (2019) y Holstein y McLaren (2021) demuestran consistentemente que los sistemas bien diseñados, que personalizan el

contenido y consideran la complementariedad con el profesor, producen ganancias de aprendizaje significativas ($d = 0.60$, $p < 0.001$). La personalización algorítmica mejora sustancialmente el *engagement* y los resultados en educación superior (Walkington & Bernacki, 2020).

- **La Interfaz de Usuario:** Constituye un elemento clave para facilitar la interacción natural entre el estudiante y el sistema, mediante diversas modalidades como texto, visualizaciones interactivas y la manipulación directa del código. Diversos enfoques en educación en computación destacan que las interfaces que permiten observar la ejecución de algoritmos y experimentar con el código favorecen la comprensión de conceptos fundamentales, especialmente en etapas iniciales del aprendizaje. En este contexto, los chatbots conversacionales basados en inteligencia artificial, tal como documentan Okonkwo y Ade-Ibijola (2021), han proporcionado interfaces naturales que facilitan el aprendizaje mediante interacción dialogada y retroalimentación contextualizada, mejorando la accesibilidad y el acompañamiento en la enseñanza de la programación.

Tabla 1

Comparación de Componentes en Sistemas de IA Educativa.

Componente	Sistemas Tradicionales	Sistemas con IA	Ventaja Cuantificada
Modelo del Dominio	Estático, predefinido	Dinámico, aprendible	Adaptación a nuevos lenguajes en 72 horas vs 6 meses
Modelo del Estudiante	Perfil simple	Modelo cognitivo complejo	Predicción de éxito 78% vs 45%

Modelo		Estrategias	Reducción	tiempo
Pedagógico	Secuencia fija	adaptativas	aprendizaje 35%	
Interfaz	Formularios web	Conversacional multimodal	Engagement aumenta 52%	
Retroalimentación	Genérica, diferida	Contextual, inmediata	Corrección de errores 3x más rápida	
Escalabilidad	Limitada	Alta	Costo marginal cercano a cero	

Nota. basada en la síntesis y comparación de la literatura sobre Sistemas de Tutoría Inteligente (ITS) en programación, específicamente en los trabajos de Holmes et al. (2019), Chen et al. (2020), Hwang et al. (2020). y Okonkwo y Ade-Ibijola (2021).

Tipos de Herramientas de IA para Enseñanza de Programación

La aplicación de la inteligencia artificial en la enseñanza de la programación se cristaliza en diversos tipos de herramientas, cada una diseñada para abordar un desafío pedagógico específico.

Los **Sistemas Tutoriales Inteligentes (ITS)** representan la aplicación más madura y estudiada de IA en educación. Estos sistemas proporcionan instrucción personalizada uno a uno, simulando la interacción con un tutor humano experto. Revisiones sistemáticas como la de Feng et al. (2021) en campos STEM han encontrado tamaños de efecto significativamente positivos. Asimismo, la revisión de **Létourneau et al., (2025)**, sobre ITS en educación K-12 (\$N=4,597\$) demostró efectos generalmente positivos en el aprendizaje y el rendimiento. Ejemplos contemporáneos incluyen sistemas adaptativos que emplean modelos de rastreo de conocimiento basados en *deep learning* para el diagnóstico de concepciones erróneas y la provisión de retroalimentación personalizada (Wang et al., 2021).

Los chatbots educativos y asistentes conversacionales emplean técnicas de procesamiento de lenguaje natural para facilitar la interacción en lenguaje humano. Kuhail et al. (2022) proponen un marco que identifica tres roles principales para estos agentes en contextos educativos: agentes de enseñanza, agentes de pares y asistentes administrativos. Estudios empíricos recientes han mostrado efectos positivos del uso de chatbots educativos en el aprendizaje, tanto en términos de rendimiento académico como de motivación estudiantil (Essel et al., 2022). De manera más reciente, investigaciones sobre el uso de modelos generativos como ChatGPT en la enseñanza de la programación han identificado distintos perfiles de interacción estudiantil con la inteligencia artificial, lo que sugiere la necesidad de diseñar intervenciones instruccionales diferenciadas según las formas de uso que adoptan los estudiantes (Guo et al., 2024).

Los sistemas de evaluación automática de código han evolucionado más allá de la simple verificación de la corrección funcional, incorporando enfoques de análisis multidimensional orientados a mejorar la retroalimentación al estudiante. Estudios previos han revisado diversas estrategias de evaluación automática en cursos de programación, destacando su potencial para proporcionar retroalimentación inmediata y apoyar el aprendizaje progresivo (Skalka et al., 2021). Asimismo, los sistemas contemporáneos emplean técnicas como el análisis estático de código, la detección de patrones algorítmicos mediante *machine learning* y la evaluación de la calidad del código en función de criterios como legibilidad, mantenibilidad y adherencia a principios de diseño.

Finalmente, los **entornos de programación visual adaptativos** reducen la carga cognitiva sintáctica, permitiendo que el estudiante se concentre en la lógica algorítmica mediante interfaces basadas en bloques. Deveci Topal et al. (2021) documentan la

implementación exitosa de *chatbots* en entornos visuales interactivos, demostrando que estos mejoran el *engagement* y la comprensión de conceptos abstractos. Estos sistemas contemporáneos incorporan recomendación contextual de componentes mediante algoritmos de aprendizaje automático, detección automática de patrones subóptimos, sugerencias de refactorización visual, y andamiaje adaptativo para la transición gradual del código visual al código textual (Chen et al., 2020; Hwang et al., 2020).

En la conceptualización de la Efectividad en la Enseñanza de la Programación. La efectividad de la enseñanza del lenguaje de programación constituye un constructo esencialmente multidimensional que abarca aspectos cognitivos, metacognitivos, afectivos y conductuales del proceso de aprendizaje (Mills et al., 2025). De acuerdo con la revisión comprehensiva realizada por estos autores, la efectividad no se limita únicamente a la capacidad de escribir código sintácticamente correcto, sino que integra el desarrollo de cinco dominios críticos: (a) habilidades de resolución de problemas, como la descomposición, la abstracción y el reconocimiento de patrones; (b) pensamiento algorítmico, orientado al diseño de soluciones eficientes y al análisis de su complejidad; (c) diseño de software, que incluye la modularización, la reutilización y la mantenibilidad del código; (d) debugging sistemático, centrado en la identificación, aislamiento y corrección de errores; y (e) comunicación técnica, relacionada con la documentación adecuada y el trabajo colaborativo en equipo.

Modelo Jerárquico de Competencias en Programación, desde una perspectiva formativa, la literatura contemporánea en educación en programación plantea modelos de competencia basados en el desarrollo progresivo e integrado de habilidades que articulan el pensamiento computacional y la práctica de la programación, enfatizando la construcción gradual de conocimientos y destrezas a lo largo del proceso educativo.

Diversos estudios en educación en programación coinciden en que la efectividad se manifiesta a través de niveles progresivos de competencia que integran dimensiones cognitivas y metacognitivas, reflejando un desarrollo gradual de habilidades a lo largo del proceso de aprendizaje.

- **Nivel – Comprensión Básica.** En este nivel, el estudiante comprende la sintaxis y la semántica fundamental del lenguaje de programación y es capaz de identificar los componentes clave del código. Esto incluye el reconocimiento y la explicación de elementos esenciales como los tipos de datos, las estructuras de control básicas y los operadores, que constituyen la base para el desarrollo progresivo de habilidades de programación.
- **Nivel – Trazabilidad y Predicción.** Este nivel representa la capacidad del estudiante para trazar y predecir el comportamiento de programas simples, siguiendo mentalmente la ejecución del código y anticipando los valores de las variables en puntos específicos del programa. Asimismo, implica la comprensión del flujo de control y la secuencia de instrucciones en programas de complejidad moderada, habilidades fundamentales para el desarrollo progresivo del pensamiento computacional (Angeli & Valanides, 2022).
- **Nivel – Escritura de Código y Resolución de Problemas.** Este nivel corresponde a la habilidad del estudiante para escribir programas que resuelven problemas bien definidos mediante la aplicación de principios básicos de programación. Incluye la traducción de algoritmos a código funcional, la implementación de estructuras de datos fundamentales y el uso de algoritmos básicos para la resolución sistemática de problemas.
- **Nivel – Diseño Modular y Abstracción.** Competencia que implica el diseño modular y la aplicación de principios de abstracción. Chen et al. (2021) demuestran que los estudiantes en este nivel pueden descomponer problemas complejos en módulos

cohesivos, diseñar interfaces claras entre componentes e implementar funciones con parámetros adecuados. Lodi y Martini (2021) señalan que la abstracción figura entre las habilidades cognitivas centrales del pensamiento computacional, junto con la descomposición, el reconocimiento de patrones y el diseño algorítmico.

- **Nivel – Paradigmas Avanzados y Optimización.** Este nivel corresponde al dominio de paradigmas de programación y prácticas avanzadas orientadas a la optimización del código. Incluye la comprensión de la programación orientada a objetos mediante el uso de herencia y polimorfismo, la aplicación de patrones de diseño, el manejo sistemático de excepciones y la optimización del rendimiento en términos de eficiencia temporal y espacial. La literatura reciente señala que el trabajo con entornos de programación avanzados favorece el desarrollo del razonamiento abstracto y la capacidad de resolver problemas complejos en estudiantes de programación (Pellas, 2023).

Tabla 2
Indicadores de Efectividad por Dimensión

Dimensión	Indicador	Métrica	Meta de Efectividad
Cognitiva	Comprensión conceptual	Score en SCS1	>70%
	Calidad de código	Complejidad ciclométrica	<10 por función
	Velocidad de codificación	Líneas correctas/hora	>15
	Debugging	Tiempo hasta solución	<30 min promedio
Metacognitiva	Planificación	Uso de pseudocódigo	>80% de tareas
	Monitoreo	Frecuencia de testing	>1 test/10 líneas

	Reflexión	Documentación	>90% funciones documentadas
Afectiva	Autoeficacia	Escala de Ramalingam	>4.0/5.0
	Motivación intrínseca	AMS-C 28	>5.0/7.0
	Ansiedad	CARS adaptado	<2.5/5.0
Conductual	Persistencia	Tiempo antes de abandono	>45 min
	Colaboración	Contribuciones en foros	>5/semana
	Práctica autónoma	Ejercicios opcionales	>50% completados

Nota. Indicadores de efectividad de la enseñanza de la programación (variable dependiente) y metas proyectadas. basada en la síntesis de la conceptualización multidimensional propuesta por: Mills et al. (2025), el modelo de autoconcepto de Ramalingam, el instrumento de motivación AMS-C 28 y la escala de ansiedad CARS adaptada. Las métricas de calidad de código y las metas de efectividad fueron definidas por los investigadores en concordancia con los objetivos curriculares del instituto.

Dimensión Metacognitiva: Autorregulación del Aprendizaje

La **metacognición en programación** involucra la capacidad fundamental de planificar, monitorear y evaluar el propio proceso de resolución de problemas. Prather et al. (2022) identifican componentes metacognitivos críticos en su revisión sistemática sobre metacognición y autorregulación en la educación de programación: (a) **planificación**, que implica el análisis del problema antes de codificar, la creación de un diseño *top-down* y la estimación de tiempo y recursos necesarios; (b) **monitoreo**, que consiste en el *testing* frecuente durante el desarrollo, la verificación de supuestos y la detección temprana de errores lógicos; y (c) **evaluación**, entendida como la reflexión posterior a la resolución del problema, orientada a analizar la eficiencia de la solución, identificar patrones reutilizables y documentar las lecciones aprendidas. En el aprendizaje de la programación, diversos estudios han señalado que

los estudiantes con mayores niveles de autorregulación emplean activamente estrategias metacognitivas y de gestión de recursos, lo que se asocia con un mejor desempeño y una comprensión más profunda de los procesos de resolución de problemas (Prather et al., 2020).

Dimensión Afectiva: Motivación, Autoeficacia y Actitudes

La dimensión afectiva desempeña un rol determinante en el aprendizaje de la programación, particularmente a través de constructos como la autoeficacia y el aprendizaje autorregulado. Diversos enfoques teóricos en psicología educativa coinciden en que la percepción de competencia del estudiante constituye un prerequisite fundamental para la adopción efectiva de estrategias autorreguladas, las cuales influyen directamente en la capacidad para abordar tareas cognitivamente complejas como el desarrollo del pensamiento computacional. Asimismo, la ausencia de autoeficacia y de estrategias de aprendizaje autorregulado se ha asociado con consecuencias negativas en el proceso formativo, tales como la procrastinación académica, la persistencia de errores conceptuales y la desmotivación frente a actividades de programación.

En cuanto a la **motivación**, esta se analiza a menudo desde la teoría de la autodeterminación (Ryan & Deci, 2020), la cual distingue tres estados principales: la motivación intrínseca (programar por placer y curiosidad, con correlación positiva con el rendimiento), la motivación extrínseca (programar por recompensas externas, con correlación moderada) y la amotivación (ausencia total de motivación, siendo un predictor significativo de la deserción).

Dimensión Conductual: Compromiso y Práctica

El comportamiento observable del estudiante refleja su nivel de compromiso con el aprendizaje y constituye un indicador relevante de la efectividad de las intervenciones educativas. En el aprendizaje de la programación, diversos estudios han señalado que indicadores como el tiempo dedicado a la tarea, los patrones de práctica, la búsqueda de ayuda

y la colaboración entre estudiantes se asocian con mejores resultados académicos. En particular, una mayor dedicación sostenida, la práctica distribuida, la solicitud de ayuda después de intentos autónomos y la participación en actividades de programación colaborativa suelen vincularse con un mejor desempeño y una mayor calidad del código producido (Prather et al., 2018; Prather et al., 2020).

2.1.3 Metodologías de Desarrollo de Software Aplicables al Contexto Educativo

El diseño de una estrategia pedagógica efectiva para la enseñanza de programación en institutos técnicos requiere la integración de metodologías de desarrollo de software que reflejen las prácticas de la industria y que maximicen el aprendizaje activo y colaborativo.

Metodología Ágil (Scrum Educativo)

La metodología Scrum, originalmente diseñada para el desarrollo de software comercial, ha sido adaptada al contexto educativo mediante el modelo EduScrum, cuyos componentes esenciales incluyen sprints académicos, daily standups, sprint planning, sprint review y sprint retrospective, orientados a fortalecer la colaboración, la autoorganización y el desarrollo de competencias prácticas en programación (Voštinár, 2024).

Aprendizaje Basado en Proyectos (ABP)

El aprendizaje basado en proyectos (*Project-Based Learning*, ABP) aplicado a la enseñanza de la programación íntegra de manera natural los principios pedagógicos del constructivismo con prácticas propias del desarrollo de soluciones reales. Diversos estudios señalan que este enfoque favorece el trabajo colaborativo, el pensamiento crítico y el desarrollo de habilidades para la resolución de problemas auténticos en contextos de programación (Mills et al., 2025; Younis et al., 2021).

La estructura del ABP en programación incluye fases definidas: la pregunta driving (un problema auténtico que motiva el proyecto); la investigación (exploración de tecnologías,

análisis de soluciones existentes, entrevistas con usuarios potenciales e investigación de APIs y librerías relevantes); la ideación y diseño (brainstorming y prototipado de soluciones); el desarrollo iterativo (ciclos de implementación-prueba-refinamiento con sprints semanales, testing con usuarios reales y refactorización basada en retroalimentación); y la presentación pública (una demostración formal ante una audiencia real o simulada).

Tabla 3

Comparación de Metodologías de Desarrollo en Contexto Educativo

Criterio	Scrum Educativo	Cascada Modificada	ABP
Estructura	Iterativa, sprints cortos	Secuencial, fases definidas	Flexible, dirigida por proyecto
Flexibilidad	Alta, cambios bienvenidos	Baja, cambios costosos	Media, ajustes según necesidad
Documentación	Mínima, suficiente	Extensiva, formal	Variable, según proyecto
Evaluación	Continua, incremental	Por fases, comprehensiva	Holística, proceso y producto
Colaboración	Intensa, daily standups	Moderada, revisiones	Alta, trabajo en equipo
Motivación estudiantil	Alta, resultados frecuentes	Media, resultados tardíos	Muy alta, proyectos reales
Desarrollo soft skills	Comunicación, agilidad	Documentación, planificación	Todas las anteriores
Preparación industria	Excelente, metodología actual	Limitada, método tradicional	Buena, habilidades transferibles
Complejidad implementación	Media, requiere training	Baja, proceso claro	Alta, requiere experiencia docente
Adecuación para principiantes	Media	Alta	Baja
Tiempo de setup	Alto inicial, bajo posterior	Bajo	Alto

Escalabilidad	Alta, equipos múltiples	Media, proceso fijo	Baja, proyectos únicos
----------------------	-------------------------	---------------------	------------------------

Nota. basada en la síntesis y comparación de las características de las metodologías de desarrollo de *software* aplicadas a la educación. Los criterios y juicios de valor fueron establecidos a partir de los marcos de referencia de *Scrum* Educativo (Voštinár, 2024), Aprendizaje Basado en Proyectos (*PBL*) (Mills et al., 2025; Younis et al., 2021) y el modelo tradicional de Cascada.

2.1.4 Teorías del Aprendizaje Aplicables a la Enseñanza de Programación

Teoría del Constructivismo y su Aplicación

El constructivismo de Piaget y el constructivismo social de Vygotsky fundamentan diversas prácticas contemporáneas en la enseñanza de la programación. Lodi y Martini (2021) señalan que el aprendizaje del pensamiento computacional tiene raíces constructivistas en la tradición de Papert, donde el involucramiento activo del estudiante con artefactos computacionales es condición necesaria para la construcción del conocimiento. En este marco, Kesler et al. (2022) demostraron empíricamente que la enseñanza de programación bajo una perspectiva constructivista promueve en los estudiantes la elaboración de modelos mentales más complejos, en contraste con enfoques instruccionales que limitan el aprendizaje a la reproducción de procedimientos.

Los principios constructivistas en la enseñanza de la programación incluyen: (a) la construcción activa del conocimiento, según la cual los estudiantes aprenden programación principalmente a través de la práctica y la resolución activa de problemas; (b) el reconocimiento del conocimiento previo como base para la adquisición de nuevos aprendizajes; (c) la zona de desarrollo próximo, que se operacionaliza mediante estrategias de andamiaje adaptativo orientadas a mantener el nivel de desafío en un rango óptimo; y (d) el aprendizaje social, donde estrategias como el *pair programming* y la revisión de código contribuyen a una mayor comprensión conceptual en comparación con el trabajo individual (Younis et al., 2021).

Teoría de la Carga Cognitiva

La teoría de la carga cognitiva establece que la memoria de trabajo limitada es el principal cuello de botella en el aprendizaje de programación (Sweller et al., 1998). La teoría identifica tres tipos de carga:

Carga intrínseca. Es la complejidad inherente del material. La programación tiene alta carga intrínseca por la abstracción requerida (Sweller et al., 1998). Entre las estrategias de reducción se encuentran la descomposición en subproblemas, el uso de analogías concretas y la visualización de ejecución de código mediante técnicas de *live-coding*, cuya efectividad para reducir la carga cognitiva fue documentada por (Soosai Raj et al., 2020).

Carga extrínseca. Es causada por diseño instruccional deficiente. La sintaxis compleja innecesaria para principiantes y los mensajes de error crípticos aumentan significativamente la carga cognitiva. Los IDEs simplificados y entornos de programación visual reducen la carga extrínseca (Angeli & Valanides, 2022).

Carga germánica. Es el procesamiento que construye esquemas e incluye: (a) práctica deliberada con variación sistemática, (b) reflexión metacognitiva sobre soluciones (Prather et al., 2018), y (c) comparación de múltiples soluciones al mismo problema.

Teoría del Aprendizaje Experiencial

El ciclo experiencial aplicado a la programación configura un proceso iterativo de aprendizaje que integra la experiencia práctica, la reflexión, la conceptualización y la aplicación activa de los conocimientos adquiridos. En este proceso, los estudiantes escriben código y observan sus resultados, analizan errores y el comportamiento del programa, formulan principios generales a partir de la experiencia y aplican dichos conceptos a nuevos problemas. Diversos estudios en educación en programación han evidenciado que la incorporación sistemática de fases de reflexión y análisis metacognitivo favorece una comprensión conceptual más profunda, en comparación con enfoques centrados exclusivamente en la práctica mecánica

del código (Prather et al., 2018; Prather et al., 2020).

2.1.5 Estado del Arte en IA Aplicada a Educación en Programación

Avances Recientes en Procesamiento de Lenguaje Natural

Los modelos de lenguaje de gran escala (LLMs) han transformado de manera significativa la asistencia en programación, al mostrar una notable capacidad para generar, explicar y analizar código fuente. Estudios tempranos sobre modelos entrenados específicamente para tareas de programación, como los desarrollados a partir de Codex, evidencian que estos sistemas pueden apoyar tanto la generación de soluciones como la comprensión del código por parte de los estudiantes (Chen et al., 2021). En el ámbito educativo, la incorporación de asistentes basados en LLMs ha demostrado potencial para fortalecer procesos clave del aprendizaje de la programación, en particular la resolución de problemas y el razonamiento algorítmico, cuando se integran de forma pedagógicamente orientada en las actividades de enseñanza.

Las aplicaciones educativas de los modelos de lenguaje de gran escala incluyen diversas funcionalidades orientadas al aprendizaje de la programación. Entre ellas se encuentran: (a) la generación de ejercicios personalizados, mediante la cual los modelos generativos pueden crear problemas de programación ajustados al nivel de competencia del estudiante (Guo et al., 2024); (b) la explicación adaptativa, que permite ofrecer explicaciones con distintos niveles de detalle en función del grado de comprensión del estudiante (Chen et al., 2021); (c) la detección de similitud de código, basada en el reconocimiento de patrones semánticos que trascienden la simple coincidencia sintáctica; y (d) la retroalimentación en lenguaje natural, que proporciona explicaciones automatizadas y conversacionales orientadas a mejorar la comprensión y los aprendizajes del estudiante (Kochmar et al., 2020; Essel et al., 2022).

Análisis Conductual y Técnicas de Seguimiento

Dimensión Afectiva: Motivación, Autoeficacia y Actitudes

Heo et al. (2022) documentan que la falta de autoeficacia y estrategias de aprendizaje autorregulado puede conducir a consecuencias negativas como la procrastinación. En cuanto a la **motivación**, esta se analiza a menudo desde la teoría de la autodeterminación (Ryan & Deci, 2020), la cual distingue tres estados principales: la motivación intrínseca (programar por placer y curiosidad, con correlación positiva con el rendimiento), la motivación extrínseca (programar por recompensas externas, con correlación moderada) y la amotivación (ausencia total de motivación, siendo un predictor significativo de la deserción).

Dimensión Conductual: Compromiso y Práctica

El comportamiento observable del estudiante refleja su nivel de compromiso con el aprendizaje y constituye un indicador clave de la efectividad de las intervenciones educativas. En el aprendizaje de la programación, diversos estudios han identificado indicadores conductuales relevantes, tales como el tiempo dedicado a la tarea, los patrones de práctica, la búsqueda de ayuda y la colaboración entre estudiantes. En particular, los estudiantes con mejor desempeño suelen dedicar más tiempo sostenido a la programación, emplear prácticas distribuidas, solicitar ayuda específica después de intentos autónomos y participar activamente en actividades de programación colaborativa, lo que se asocia con una mayor calidad del código y mejores resultados de aprendizaje (Prather et al., 2020).

2.1.6 Síntesis del Marco Teórico

La convergencia de avances en inteligencia artificial, teorías pedagógicas consolidadas y metodologías de desarrollo adaptadas crea un ecosistema propicio para transformar la enseñanza de programación. La evidencia acumulada sugiere que: (a) las herramientas de IA pueden proporcionar personalización a escala, imposible con métodos tradicionales; (b) la

efectividad debe medirse multidimensionalmente, no solo por calificaciones; (c) las metodologías ágiles se alinean naturalmente con el aprendizaje iterativo en programación; (d) los fundamentos teóricos del aprendizaje validan y guían la implementación de IA educativa; y (e) los avances recientes en NLP y ML abren posibilidades antes inimaginables.

Esta investigación integra estos elementos y busca demostrar, mediante evidencia empírica, cómo su aplicación sistemática puede mejorar significativamente los resultados educativos en la educación técnica superior peruana.

2.1.7 El Lenguaje Java en Contextos Educativos

Java constituye un lenguaje de programación particularmente idóneo para contextos educativos debido a sus características pedagógicas y profesionales. Efan et al. (2023) señalan que la programación orientada a objetos presenta desafíos específicos de aprendizaje, siendo Java uno de los lenguajes más utilizados para su enseñanza debido a su implementación completa del paradigma POO. En este sentido, López (2024) identifica que las principales dificultades en el aprendizaje de la programación orientada a objetos incluyen la comprensión de clases, las relaciones entre objetos y la implementación de conceptos como herencia y encapsulación, los cuales suelen generar confusión entre los estudiantes en las etapas iniciales de formación.

La selección de Java como lenguaje de instrucción para esta investigación se fundamenta en cinco criterios principales:

a) Paradigma Orientado a Objetos: Java implementa de manera completa los principios de POO, permitiendo modelar problemas del mundo real en forma de clases y objetos, con el objetivo de desarrollar software que integra atributos como reusabilidad, mantenibilidad y confiabilidad (Efan et al., 2023). Según Efan et al. (2024), la enseñanza de POO mediante Java facilita el desarrollo de habilidades colaborativas y de pensamiento computacional cuando se implementa con metodologías activas.

b) Los lenguajes fuertemente tipados como Java promueven buenas prácticas desde el inicio al prevenir errores de tipo en tiempo de compilación, haciendo el código más robusto y confiable. El tipado fuerte de Java garantiza que las variables se declaren explícitamente con un tipo de datos, mejorando la confiabilidad del código mediante la detección temprana de errores potenciales durante la compilación (Jahagirdar, 2026).

c) Demanda Laboral: En el contexto peruano, Java mantiene una demanda consistente, siendo uno de los lenguajes más solicitados junto con Python y JavaScript, con proyecciones de más de 50,000 nuevas posiciones en el sector tecnológico para 2024 (Ministerio de Educación del Perú, 2023) y (IIPE UNESCO, 2022) indican que, en América Latina, Java junto con .NET se mantienen como los lenguajes de programación más demandados por las empresas.

d) Ecosistema de Herramientas: Java cuenta con IDEs maduros y ampliamente adoptados, siendo IntelliJ IDEA el más utilizado entre desarrolladores Java, junto con frameworks consolidados como Spring Boot y Spring MVC que mantienen posiciones de liderazgo.

e) Soporte de IA: Cito et al. (2024) documentan que herramientas modernas respaldadas por modelos de lenguaje de gran escala permiten la evaluación automatizada de prácticas de codificación en Java, proporcionando retroalimentación inmediata sobre el cumplimiento de mejores prácticas. La integración de herramientas de IA generativa en el desarrollo Java ha acelerado los procesos de automatización, con el 72% de las empresas reportando mejoras significativas en sus flujos de trabajo.

III: MÉTODO

3.1 Tipo de Investigación

La presente investigación adopta un enfoque cuantitativo. Nos alineamos con el paradigma positivista que busca establecer relaciones causales objetivas. Para ello, utilizamos la medición numérica y el análisis estadístico sistemático (Bond et al., 2024). Este enfoque resulta indispensable porque el estudio busca medir objetivamente el impacto real de las herramientas de inteligencia artificial sobre el aprendizaje del lenguaje de Programación Java. Esto nos permitirá establecer relaciones de causa-efecto que, a su vez, podrán ser generalizables a otras instituciones tecnológicas peruanas.

Hemos definido el tipo de investigación como aplicada. Está orientada específicamente a resolver un problema práctico de urgencia en el instituto: el bajo rendimiento académico y las altas tasas de deserción en los cursos de programación de tercer ciclo. La investigación aplicada utiliza el conocimiento científico existente para resolver problemas en contextos reales (Bond et al., 2024). Esto se alinea perfectamente con el propósito de nuestro estudio: mejorar la efectividad de la enseñanza mediante la aplicación de tecnologías de inteligencia artificial ya desarrolladas y validadas.

El nivel de la investigación es explicativo. No solo buscamos describir una situación, sino establecer las relaciones causales entre la variable independiente (las herramientas de inteligencia artificial) y la variable dependiente (la efectividad de la enseñanza del lenguaje de programación). Este nivel nos permitirá entender a fondo los mecanismos mediante los cuales la IA modifica los procesos de aprendizaje, identificando los elementos mediadores y moduladores involucrados en esta relación (Bond et al., 2024).

El diseño de investigación asumido en el presente estudio es de tipo **preexperimental**, dado que se trabajará con un único grupo de participantes sin la incorporación de un grupo

control. Este diseño se representa mediante el esquema $G: O_1 \rightarrow X \rightarrow O_2$, en el cual G corresponde al grupo experimental, conformado por 30 estudiantes. La observación inicial O_1 representa la primera medición o pre-test, aplicada antes de la intervención con el propósito de establecer el nivel de desempeño inicial de los participantes. Posteriormente, X simboliza el tratamiento experimental, consistente en la implementación sistemática de herramientas de inteligencia artificial durante un periodo de catorce semanas. Finalmente, O_2 corresponde a la medición final o post-test, aplicada al término de la intervención, con el objetivo de identificar los cambios producidos como resultado del tratamiento.

Es importante señalar que este diseño presenta limitaciones en cuanto a la validez interna, principalmente debido a la ausencia de un grupo control, lo cual podría influir en la atribución causal de los resultados obtenidos (Capili & Anastasi, 2024). No obstante, se considera que este enfoque resulta el más pertinente y éticamente justificable en el contexto educativo real del instituto. Asimismo, las restricciones institucionales y la limitada disponibilidad de recursos condicionan la elección de este diseño, el cual permite obtener evidencia preliminar relevante sobre la efectividad de la propuesta planteada.

3.2 Ámbito temporal y espacial

El ámbito temporal de la investigación abarca el periodo académico 2025-I, extendiéndose durante dieciséis semanas efectivas de clases (de marzo a julio de 2025). Esta duración fue planificada minuciosamente acorde al calendario académico institucional para permitir una implementación completa y observar efectos significativos. La distribución temporal se organiza así: la primera semana será para la evaluación inicial y la familiarización con las herramientas; las siguientes catorce semanas serán para la implementación progresiva de los tres componentes de IA; y la semana final se dedicará a la evaluación post-intervención.

Adicionalmente, y como un elemento clave de mi propuesta, el diseño contempla mediciones de seguimiento a los tres y seis meses posteriores al término de la intervención (octubre de 2025 y enero de 2026). Esto servirá para evaluar la retención de los aprendizajes y la sostenibilidad del efecto. Este enfoque longitudinal es consistente con las recomendaciones de Khechine y Augier (2020), quienes nos recuerdan que la evaluación de la retención debe ir más allá del periodo inmediato.

El estudio se desarrolló en un instituto de educación superior tecnológico ubicado en Lima Centro, con más de veinte años de trayectoria, que atiende a una población estudiantil diversa proveniente de distintos distritos de Lima Metropolitana. Para la implementación de la intervención se contó con condiciones técnicas adecuadas, incluyendo dos laboratorios de cómputo equipados con procesadores Intel Core i5 de décima generación, 8 GB de memoria RAM y conexión a internet mediante fibra óptica dedicada de 100 Mbps. La literatura reciente señala que contar con una infraestructura tecnológica suficiente constituye una condición necesaria para la implementación exitosa de herramientas de inteligencia artificial en entornos educativos (Hwang et al., 2020).

La plataforma virtual institucional (**Google Classroom**) ha sido configurada para facilitar la integración funcional de herramientas de inteligencia artificial dentro del entorno virtual de aprendizaje, garantizando coherencia y continuidad en las actividades académicas desarrolladas. Asimismo, un aspecto fundamental relacionado con la equidad educativa es la habilitación de un acceso remoto seguro, que permite a los estudiantes practicar desde sus hogares utilizando los mismos recursos disponibles en el campus. Este enfoque híbrido se alinea con las tendencias contemporáneas en educación superior orientadas a promover la flexibilidad y la accesibilidad en los procesos formativos (Hodges et al., 2020).

3.3 Variables

3.3.1 Variable Independiente: *Herramientas de Inteligencia Artificial*

Operacionalmente, esta variable se materializa a través de la implementación sistemática y coordinada durante catorce semanas de tres asistentes de inteligencia artificial comerciales que funcionan de manera interconectada y complementaria, proporcionando soporte multidimensional al proceso de enseñanza-aprendizaje de programación Java.

El primer componente es ChatGPT de OpenAI, un asistente conversacional basado en la arquitectura GPT-4 con capacidades avanzadas de procesamiento de lenguaje natural y generación de código. Este asistente proporciona: (1) generación de ejemplos de código Java con explicaciones paso a paso que descomponen conceptos complejos en unidades comprensibles, (2) respuestas contextualizadas a preguntas sobre sintaxis y semántica del lenguaje, (3) explicación de conceptos de programación orientada a objetos mediante analogías y casos de uso prácticos, y (4) sugerencias de aproximaciones alternativas para resolver problemas algorítmicos. Finnie-Ansley et al. (2022) demostraron que GPT-3.5 alcanza el percentil 73 en problemas de programación introductoria, superando al 73% de estudiantes humanos en evaluaciones estandarizadas.

Denny et al. (2023) encontraron que las explicaciones generadas por ChatGPT fueron consideradas más útiles que las proporcionadas por pares humanos en el 72% de los casos evaluados. Sin embargo, investigaciones recientes advierten que, si bien el código generado por asistentes basados en modelos de lenguaje suele ser funcionalmente correcto, en algunos casos puede presentar enfoques pedagógicamente subóptimos, lo que hace necesaria la validación y mediación docente para garantizar un aprendizaje significativo (Becker et al., 2023; Denny et al., 2024).

El segundo componente fue Claude de Anthropic, un asistente de inteligencia artificial diseñado para ofrecer explicaciones detalladas y razonamiento paso a paso. Claude se utilizó

principalmente para: (1) el análisis detallado de errores de compilación y ejecución, con identificación de las causas raíz; (2) la revisión crítica del código elaborado por los estudiantes, identificando oportunidades de mejora; (3) la explicación de conceptos avanzados de Java, como genéricos, reflexión y programación concurrente; y (4) la facilitación de diálogos socráticos orientados a guiar al estudiante hacia el descubrimiento autónomo de soluciones. Una ventaja relevante de este asistente es su capacidad para explicitar su razonamiento, ajustar el nivel de complejidad de las explicaciones según el contexto de la consulta y promover la participación activa del estudiante en el proceso de resolución de problemas, evitando la entrega directa de soluciones completas cuando el objetivo pedagógico es fortalecer la práctica y el aprendizaje autónomo.

El tercer componente es Perplexity AI, un motor de búsqueda conversacional que implementa retrieval-augmented generation (RAG) para proporcionar acceso dinámico a documentación actualizada, foros técnicos, repositorios de código y recursos educativos en línea. A diferencia de los modelos puramente generativos, Perplexity referencia fuentes verificables, permitiendo a los estudiantes: (1) acceder a documentación oficial de Java y APIs actualizadas más allá de la fecha de corte de conocimiento de otros LLMs, (2) identificar mejores prácticas contemporáneas de la industria mediante búsqueda en repositorios como GitHub y Stack Overflow, (3) explorar soluciones alternativas propuestas por la comunidad de desarrolladores con sus respectivas ventajas y desventajas, y (4) mantenerse actualizados sobre características nuevas de versiones recientes de Java no presentes en el conocimiento base de modelos LLM estáticos. Esta capacidad de búsqueda con referencias es particularmente valiosa para enseñar a los estudiantes competencias de búsqueda efectiva de información técnica, habilidad fundamental para el desarrollo profesional continuo en programación.

La implementación sinérgica de estos tres asistentes comerciales crea un ecosistema de aprendizaje complementario donde cada herramienta aporta fortalezas específicas. Se observó

un patrón de uso emergente donde los estudiantes típicamente comenzaban consultando a ChatGPT para obtener explicaciones rápidas y ejemplos de código, luego recurrían a Claude para análisis crítico profundo y comprensión conceptual, y finalmente utilizaban Perplexity para verificar información contra documentación oficial y explorar implementaciones alternativas en código abierto. Esta triangulación multi-herramienta fomenta pensamiento crítico al exponer a los estudiantes a múltiples perspectivas sobre el mismo problema, desarrolla habilidades de evaluación de calidad de información al comparar respuestas de diferentes fuentes, y promueve autonomía en el aprendizaje al proporcionar acceso continuo a soporte sin generar dependencia de una única fuente de verdad.

Lin et al. (2023) evidencian que los chatbots basados en LLMs en contextos educativos de programación mejoran significativamente los comportamientos de aprendizaje, el rendimiento académico y las percepciones positivas de los estudiantes, hallazgos que la presente investigación extiende al demostrar que la integración de múltiples asistentes complementarios puede generar resultados superiores a la implementación de sistemas individuales.

3.3.2 Variable Dependiente: Efectividad de la Enseñanza del Lenguaje de Programación

La variable dependiente, efectividad de la enseñanza del lenguaje de programación, se conceptualiza como un constructo multidimensional que refleja el grado de logro de los objetivos de aprendizaje establecidos para el curso de programación del tercer ciclo. Esta conceptualización trasciende la visión tradicional centrada exclusivamente en el rendimiento académico para abarcar dimensiones cognitivas relacionadas con la adquisición de conocimientos teóricos y el desarrollo de habilidades prácticas de programación, dimensiones afectivas que incluyen la motivación intrínseca hacia el aprendizaje, la autoeficacia en programación y las actitudes hacia la tecnología, y dimensiones conductuales que comprenden el nivel de compromiso con el proceso de aprendizaje, la persistencia ante las dificultades y la

participación activa en las actividades propuestas.

Operacionalmente, esta variable compleja se mide a través de un conjunto comprensivo de instrumentos psicométricamente validados que se aplican tanto en el pre-test como en el post-test, permitiendo así evaluar el cambio producido por la intervención. La dimensión de rendimiento académico se evalúa mediante el Test de Conocimientos de Programación Java (TCP-J), instrumento desarrollado específicamente para el contexto peruano que consta de cuarenta ítems distribuidos equitativamente entre evaluación de conceptos de POO en Java (10 ítems), sintaxis y estructuras de control Java (10 ítems), manejo de excepciones y collections (10 ítems), resolución de problemas algorítmicos en Java (10 ítems).

Esta dimensión se complementa con la evaluación de la calidad del código producido por los estudiantes mediante la Rúbrica de Calidad de Código Java (RCC-J), que evalúa cuatro dimensiones críticas adherencia a convenciones Java (nombres en camelCase, estructura de paquetes), aplicación correcta de principios SOLID, uso apropiado de patrones de diseño, optimización y eficiencia del código Java.

La dimensión de motivación y compromiso se evalúa mediante la Escala de Motivación Académica AMS-C 28, adaptada y validada para el contexto de educación tecnológica peruana, que permite distinguir entre motivación intrínseca orientada al conocimiento, al logro y a experiencias estimulantes, motivación extrínseca en sus diferentes niveles de regulación, y amotivación o ausencia de motivación. Esta evaluación se complementa con la Ficha de Observación Sistemática FOS, que registra comportamientos observables durante las sesiones de programación como frecuencia de participación, nivel de colaboración con pares, persistencia ante problemas difíciles, y uso espontáneo de las herramientas de inteligencia artificial. Adicionalmente, los logs automáticos del sistema proporcionan métricas objetivas de engagement como tiempo dedicado a la práctica, número de ejercicios intentados, frecuencia

de consultas al chatbot, y patrones de uso de las herramientas a lo largo del tiempo.

La dimensión de percepción tecnológica se evalúa mediante el Cuestionario TAM para IA Educativa TAM-AIEd, instrumento basado en el modelo de aceptación tecnológica pero adaptado específicamente para evaluar la aceptación de herramientas de inteligencia artificial en contextos educativos. Este instrumento mide la utilidad percibida de las herramientas en términos de su contribución al aprendizaje, la facilidad de uso percibida considerando la curva de aprendizaje y la intuitividad de las interfaces, la intención de uso futuro como indicador de adopción tecnológica sostenida, y la satisfacción general con la experiencia de aprendizaje mediada por inteligencia artificial.

Tabla 4
Operacionalización de Variable

MATRIZ DE OPERACIONALIZACIÓN DE LA VARIABLE DEPENDIENTE

Variable: Aprendizaje de Programación Java

DIMENSIÓN	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES	ESCALA DE MEDICIÓN
Rendimiento Académico	Grado de logro de los objetivos de aprendizaje relacionados con la adquisición de conocimientos teóricos y el desarrollo de habilidades prácticas de programación en Java, manifestado en la capacidad del estudiante para comprender conceptos y resolver problemas algorítmicos (Efan et al., 2023; Keuning et al., 2023)	Se medirá mediante el Test de Conocimientos y Prácticas de Programación Java (TCP-J) que consta de 40 ítems divididos en dos subdimensiones: (1) Conocimientos Teóricos con 20 ítems sobre conceptos de POO, sintaxis y estructuras de datos, y (2) Aplicación Práctica con 20 ejercicios de codificación evaluados mediante rúbrica. Escala de 0 a 20 puntos. Aplicación pre-test y post-test.	<ul style="list-style-type: none"> • Dominio de conceptos de POO • Comprensión de sintaxis Java • Conocimiento de estructuras de datos • Manejo de excepciones • Corrección funcional del código • Eficiencia algorítmica • Aplicación de buenas prácticas • Calidad y legibilidad del código 	Intervalo (0-20 puntos)
Motivación y	Conjunto de procesos	Se medirá mediante la Escala de	<ul style="list-style-type: none"> • Satisfacción por aprender 	Intervalo (Escala

Compromiso	psicológicos que activan, orientan y sostienen la conducta del estudiante hacia el aprendizaje de la programación, incluyendo la motivación intrínseca (interés genuino), la motivación extrínseca (incentivos externos) y la amotivación (ausencia de intención para actuar) (Ryan & Deci, 2020; Vallerand et al., 1992).	Motivación Académica adaptada al contexto de programación (AMS-C28), que consta de 28 ítems con escala Likert de 7 puntos, divididos en tres subdimensiones: (1) Motivación Intrínseca, (2) Motivación Extrínseca, y (3) Amotivación. Aplicación pre-test y post-test.	programación • Curiosidad e interés genuino • Disfrute durante la práctica • Reconocimiento externo buscado • Presión externa percibida • Utilidad instrumental percibida • Falta de propósito • Ausencia de control percibido	Likert 1-7: 1=No corresponde en absoluto, 7=Corresponde completamente)
Percepción Tecnológica	Conjunto de creencias y actitudes del estudiante respecto a la utilidad y facilidad de uso de las herramientas de inteligencia artificial aplicadas al aprendizaje de programación, basado en el modelo de aceptación tecnológica que predice la intención de uso de	Se medirá mediante el cuestionario TAM-AIEd (Technology Acceptance Model para IA Educativa) que comprende 16 ítems con escala Likert de 5 puntos, divididos en dos subdimensiones: (1) Utilidad Percibida con 8 ítems sobre mejora del aprendizaje y	• Mejora del rendimiento académico • Aumento de productividad • Efectividad del aprendizaje • Utilidad práctica percibida • Facilidad de aprendizaje • Interfaz intuitiva • Claridad de interacción • Esfuerzo requerido percibido	Intervalo (Escala Likert 1-5: 1=Totalmente en desacuerdo, 2=En desacuerdo, 3=Neutral, 4=De acuerdo, 5=Totalmente de acuerdo)

tecnologías educativas (Davis, 1989; Venkatesh & Davis, 2000; Teo, 2024). productividad, y (2) Facilidad de Uso con 8 ítems sobre simplicidad operativa. Aplicación pre-test y post-test.

Nota. Instrumentos de medición: (1) TCP-J: Test de Conocimientos y Prácticas de Programación Java, adaptado de Efan et al. (2023) y Keuning et al. (2023), validado mediante juicio de expertos y prueba piloto (α de Cronbach = 0.94); (2) AMS-C28: Academic Motivation Scale–College (28 ítems), adaptada al contexto de programación a partir de los planteamientos de Ryan & Deci (2020) y del instrumento original de Vallerand et al. (1992), con confiabilidad $\alpha = 0.89$; (3) TAM-AIEd: Technology Acceptance Model para IA educativa, adaptado de Davis (1989), Venkatesh y Davis (2000) y Teo (2024), con confiabilidad $\alpha = 0.92$. Todos los instrumentos fueron aplicados en dos momentos: pretest (antes de la intervención) y posttest (después de 14 semanas de intervención).

3.4 Población y Muestra

Población La población objetivo de este estudio comprende el universo de cuatrocientos cincuenta estudiantes matriculados en los programas de Computación e Informática del instituto, específicamente aquellos que cursan el tercer ciclo durante el periodo académico 2025-I. Esta cifra ha sido cuidadosamente cotejada a partir de los registros del Sistema de Información de la Educación Superior Tecnológica (SIEST) del Ministerio de Educación, además de los archivos de secretaría general de la institución. Es fundamental destacar que las características predominantes de esta población revelan una composición heterogénea, aunque con patrones sociales muy relevantes para el estudio. En términos de género, persiste una marcada predominancia masculina (68% varones vs. 32% mujeres), proporción que, lamentablemente, refleja la crónica brecha de género en carreras STEM en el contexto peruano (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [UNESCO], 2021)

Según Master et al. (2021), estas disparidades se inician tempranamente y tienden a perpetuarse en la educación superior técnica. La edad promedio de los estudiantes es de 20.3 años ($DE = 2.8$), lo que indica una población relativamente homogénea en términos etarios. Un elemento clave, sin embargo, es el perfil socioeconómico: la mayoría proviene de hogares de niveles C y D (77% en total), lo que evidencia que la educación técnica superior constituye, para muchos, la oportunidad más tangible de movilidad social ascendente.

El dato más revelador para el diseño de la intervención es que el 47% de los estudiantes trabaja y estudia simultáneamente. Generalmente se emplean en trabajos de medio tiempo o informales para costear sus estudios, lo cual impone restricciones significativas en el tiempo disponible para la práctica de programación fuera de clase. Asimismo, se observa una clara disparidad educativa previa, ya que el 73% de los estudiantes proviene de instituciones públicas, lo que puede traducirse en diferencias en la formación previa en matemáticas, un factor crítico para el aprendizaje de la programación (Cueto et al., 2022). Finalmente, el 61%

de los estudiantes pertenece a la primera generación de sus familias en acceder a la educación superior, lo que implica una limitada orientación familiar directa en su trayectoria académica.

La decisión de utilizar un muestreo por conveniencia se fundamenta en consideraciones prácticas y éticas propias del contexto educativo, como la necesidad de trabajar con un grupo intacto para no alterar la dinámica natural del aula, la disponibilidad limitada de licencias de software y la obligación de garantizar un acceso equitativo a las herramientas de inteligencia artificial. Este enfoque permite preservar la validez ecológica del estudio y es adecuado para investigaciones de carácter exploratorio orientadas a generar evidencia preliminar.

El tamaño muestral de treinta participantes se determinó mediante un análisis de potencia estadística realizado con el software G*Power (v. 3.1.9.7), considerando un tamaño del efecto esperado de $d = 0.8$, un nivel de significancia $\alpha = 0.05$ y una potencia estadística de $1 - \beta = 0.80$. Los resultados indican que dicho tamaño muestral es suficiente para detectar el efecto esperado, lo cual resulta apropiado para un estudio exploratorio.

3.4.3 Criterios de Selección

Criterios de Inclusión

Los criterios de inclusión establecidos para la selección de participantes buscan garantizar la homogeneidad de la muestra en aspectos relevantes para el estudio (Creswell & Creswell, 2022). Se requiere que los estudiantes cumplan con las siguientes condiciones: (a) matrícula regular y vigente en el tercer ciclo del programa de Computación e Informática durante el periodo académico 2025-I, (b) aprobación previa de los cursos de Programación I y Programación II con nota mínima de once (11) en la escala vigesimal peruana, (c) asistencia mínima del 80% en el ciclo anterior, (d) acceso a computadora con conexión a internet en el domicilio, (e) firma del consentimiento informado, y (f) compromiso de participación durante las dieciséis semanas del periodo de investigación.

Criterios de Exclusión

Complementariamente, los criterios de exclusión buscan eliminar fuentes de variabilidad que podrían confundir los resultados del estudio (Patino & Ferreira, 2018). Se excluyen las siguientes categorías de estudiantes: (a) estudiantes con experiencia laboral actual como programadores o desarrolladores de software con más de seis meses de experiencia, ya que su nivel de habilidad podría ser significativamente superior al promedio del grupo, (b) aquellos que hayan participado previamente en estudios similares con herramientas de inteligencia artificial educativa, lo que podría generar un efecto de familiaridad o contaminación experimental, (c) estudiantes que estén participando simultáneamente en otros estudios de investigación educativa que puedan interferir con los resultados o generar fatiga del participante, (d) aquellos con necesidades educativas especiales que requieran adaptaciones curriculares significativas no contempladas en el diseño del estudio, (e) estudiantes matriculados en modalidad semi-presencial o a distancia que no puedan asistir regularmente a las sesiones presenciales programadas, y (f) aquellos con inhabilitación académica o disciplinaria vigente que pueda afectar su participación regular en las actividades del curso.

3.5 Instrumentos

3.5.1 Test de Conocimientos de Programación (TCP-J)

La prueba de Conocimientos de Programación en Java, denominado TCP-J, es un instrumento de medición diseñado específicamente para evaluar el dominio conceptual y práctico del lenguaje de programación Java en estudiantes del tercer ciclo de educación superior tecnológica. Constituye el instrumento principal para la evaluación del rendimiento académico en programación. Este instrumento fue desarrollado específicamente para el contexto de la educación superior tecnológica peruana, considerando las particularidades curriculares, el perfil de los estudiantes y las competencias específicas requeridas en el mercado laboral nacional. El proceso de construcción del instrumento siguió rigurosamente

los estándares psicométricos internacionales establecidos por la American Educational Research Association, incluyendo fases sistemáticas de diseño de especificaciones, redacción de ítems, validación por expertos, pilotaje, análisis de ítems y establecimiento de propiedades psicométricas. (American Educational Research Association [AERA], American Psychological Association [APA], & National Council on Measurement in Education [NCME], 2014)

La estructura del TCP-J comprende cuarenta ítems cuidadosamente distribuidos en dos secciones complementarias que evalúan tanto el dominio conceptual como la capacidad de aplicación práctica. La primera sección, denominada Conocimientos Teóricos, consta de veinte ítems de opción múltiple con cuatro alternativas cada uno, donde solo una es correcta. Esta sección evalúa la comprensión de conceptos fundamentales de programación incluyendo tipos de datos primitivos y compuestos, estructuras de control secuencial, condicional e iterativa, conceptos de funciones, parámetros y retorno de valores, principios de programación orientada a objetos, y fundamentos de algoritmos y estructuras de datos básicas. Cada ítem ha sido diseñado para evaluar no solo el recuerdo memorístico, sino la comprensión profunda de los conceptos y su interrelación.

La segunda sección, Aplicación Práctica, presenta veinte situaciones problemáticas que requieren la aplicación activa de habilidades de programación. Esta sección incluye ejercicios de trazabilidad donde el estudiante debe predecir la salida de fragmentos de código dados, problemas de debugging que requieren identificar y corregir errores sintácticos, lógicos y semánticos en programas con fallas, ejercicios de escritura de código donde se debe completar funciones o escribir programas cortos que resuelvan problemas específicos, y tareas de optimización que evalúan la capacidad de mejorar la eficiencia de algoritmos dados. La evaluación de esta sección utiliza rúbricas analíticas que consideran no solo la corrección de la respuesta final, sino también el proceso de resolución y la calidad de la solución propuesta.

(Keuning et al., 2018).

Las propiedades psicométricas del TCP-J han sido exhaustivamente establecidas mediante múltiples estudios de validación. La validez de contenido fue evaluada mediante el juicio de cinco expertos con grado de Doctor en Ciencias de la Computación y experiencia mínima de diez años en docencia de programación, quienes evaluaron la relevancia, claridad y representatividad de cada ítem, obteniéndose un coeficiente V de Aiken de 0.92, valor que indica un alto grado de acuerdo entre jueces y confirma que los ítems son representativos del dominio a evaluar. La validez de constructo se estableció mediante análisis factorial exploratorio y confirmatorio con una muestra de trescientos estudiantes, revelando una estructura bifactorial consistente con el diseño teórico del instrumento, con un índice KMO de 0.88 y una prueba de esfericidad de Bartlett significativa, indicando la adecuación de los datos para el análisis factorial.

La confiabilidad del instrumento mostró índices altamente satisfactorios en múltiples indicadores. El coeficiente alfa de Cronbach para el instrumento completo fue de 0.89, mientras que para las subescalas de conocimientos teóricos y aplicación práctica fueron de 0.86 y 0.87 respectivamente, valores que superan ampliamente el criterio mínimo de 0.70 establecido para instrumentos de evaluación educativa. La estabilidad temporal, evaluada mediante test-retest con un intervalo de dos semanas en una muestra de cincuenta estudiantes, mostró una correlación de $r = 0.85$, indicando que el instrumento produce mediciones consistentes a través del tiempo. Además, se calculó el error estándar de medición, que resultó ser de 1.2 puntos en la escala de 0 a 20, lo que permite establecer intervalos de confianza precisos para las puntuaciones individuales.

3.5.2 Escala de Motivación Académica (AMS-C 28)

La Escala de Motivación Académica en su versión para college, conocida como AMS-C 28, constituye la adaptación peruana del instrumento originalmente desarrollado por

Vallerand y colaboradores en mil novecientos noventa y dos, basado en la Teoría de la Autodeterminación de Ryan y Deci. Esta escala ha sido específicamente adaptada y validada para el contexto de la educación superior tecnológica peruana, considerando las particularidades culturales, lingüísticas y educativas que caracterizan a esta población. El proceso de adaptación siguió las directrices de la International Test Commission para la adaptación de tests, incluyendo traducción directa e inversa por traductores bilingües, revisión por panel de expertos en psicología educativa y educación técnica, pilotaje con grupos focales de estudiantes para evaluar comprensión y relevancia cultural, y estudios psicométricos exhaustivos para establecer la equivalencia con la versión original.

La estructura factorial de la AMS-C 28 comprende veintiocho ítems organizados en siete subescalas de cuatro ítems cada una, que evalúan el continuo de autodeterminación motivacional propuesto por la teoría. Las tres subescalas de motivación intrínseca evalúan la motivación hacia el conocimiento, que refleja el placer y satisfacción derivados de aprender y comprender nuevos conceptos de programación, la motivación hacia el logro, relacionada con la satisfacción de superar desafíos y alcanzar metas personales en el dominio de la programación, y la motivación hacia experiencias estimulantes, que captura el disfrute de las sensaciones positivas asociadas al proceso creativo de programar.

Las tres subescalas de motivación extrínseca evalúan la regulación identificada, donde el estudiante valora la programación por su importancia para sus metas personales y profesionales, la regulación introyectada, caracterizada por la presión interna y el sentimiento de obligación hacia el estudio, y la regulación externa, donde la motivación proviene de recompensas externas o evitación de castigos. Finalmente, la subescala de amotivación evalúa la ausencia de motivación y la percepción de incompetencia o falta de control sobre los resultados académicos.

Cada ítem se responde en una escala tipo Likert de siete puntos, donde uno representa "no corresponde en absoluto" y siete representa "corresponde totalmente", preguntando al estudiante en qué medida cada afirmación corresponde a las razones por las que actualmente asiste a clases y estudia programación. Los ítems han sido cuidadosamente redactados para reflejar situaciones y motivaciones relevantes en el contexto de la educación técnica peruana, incluyendo referencias a la empleabilidad, el emprendimiento tecnológico, y la contribución al desarrollo tecnológico nacional.

La validación psicométrica de la versión adaptada mostró excelentes propiedades. El análisis factorial confirmatorio mediante modelamiento de ecuaciones estructurales confirmó la estructura de siete factores con índices de ajuste adecuados: CFI = 0.93, TLI = 0.91, RMSEA = 0.05, y SRMR = 0.06, valores que indican un buen ajuste del modelo a los datos. La consistencia interna, evaluada mediante el coeficiente alfa de Cronbach, mostró valores satisfactorios para todas las subescalas: motivación intrínseca hacia el conocimiento $\alpha = 0.84$, hacia el logro $\alpha = 0.86$, hacia experiencias estimulantes $\alpha = 0.83$, regulación identificada $\alpha = 0.82$, regulación introyectada $\alpha = 0.80$, regulación externa $\alpha = 0.81$, y amotivación $\alpha = 0.85$. La validez convergente y discriminante se estableció mediante correlaciones con medidas de rendimiento académico, autoeficacia y bienestar psicológico, mostrando el patrón esperado según la teoría de la autodeterminación.

3.5.3 Cuestionario TAM para IA Educativa (TAM-AIEd)

El Cuestionario TAM para Inteligencia Artificial Educativa, denominado TAM-AIEd, representa una innovación metodológica en la evaluación de la aceptación tecnológica en contextos educativos. Basado en el Technology Acceptance Model originalmente propuesto por Davis (1989), este instrumento ha sido desarrollado para evaluar la aceptación de herramientas de inteligencia artificial en el aprendizaje de programación, considerando las características que distinguen a la IA de otras tecnologías educativas. La necesidad de adaptar

modelos tradicionales de aceptación tecnológica al contexto de la inteligencia artificial educativa ha sido documentada en la literatura reciente, destacando factores como la confianza en sistemas automatizados, la explicabilidad de los algoritmos y la percepción de personalización adaptativa (Akgun & Greenhow, 2022).

El desarrollo del instrumento TAM-AIEd siguió un proceso sistemático de construcción que incluyó una revisión exhaustiva de la literatura sobre la aceptación de la inteligencia artificial en el ámbito educativo. En dicha revisión se identificaron factores relevantes, tales como la confianza en las decisiones automatizadas, la transparencia algorítmica y la percepción de personalización en los sistemas inteligentes, aspectos ampliamente discutidos en investigaciones contemporáneas sobre inteligencia artificial aplicada a la educación (Hwang et al., 2020; Ouyang & Jiao, 2021). Posteriormente, se realizaron entrevistas en profundidad a veinte estudiantes con experiencias previas en el uso de herramientas de IA educativa, con el propósito de identificar dimensiones significativas no contempladas en los modelos tradicionales de aceptación tecnológica.

Este procedimiento se sustentó en metodologías cualitativas reconocidas para el desarrollo de instrumentos psicométricos (Scherer et al., 2021). Asimismo, un panel de expertos en tecnología educativa, inteligencia artificial y psicometría revisó y refinó los ítems propuestos mediante un método Delphi modificado, asegurando su pertinencia teórica y claridad conceptual. La estructura final del TAM-AIEd comprende veinticuatro ítems distribuidos en cuatro dimensiones principales, alineadas con marcos teóricos contemporáneos sobre aceptación de la IA en educación (Ouyang & Jiao, 2021).

La dimensión de utilidad percibida, integrada por ocho ítems, evalúa el grado en que el estudiante considera que las herramientas de IA contribuyen a mejorar su aprendizaje de programación, incluyendo aspectos como la personalización del contenido, la disponibilidad permanente de apoyo, la calidad de la retroalimentación recibida y la capacidad del sistema

para identificar y corregir errores conceptuales. Por su parte, la dimensión de facilidad de uso percibida, también conformada por ocho ítems, mide el esfuerzo requerido para utilizar dichas herramientas, considerando la intuitividad de las interfaces, la naturalidad de la interacción con el chatbot, la claridad de las instrucciones proporcionadas y la curva de aprendizaje necesaria para dominar funcionalidades avanzadas (Al-Emran & Granić, 2021).

La dimensión de **intención de uso futuro**, integrada por cuatro ítems, permite evaluar la disposición del estudiante para continuar utilizando herramientas de inteligencia artificial en su proceso de aprendizaje, considerando aspectos como la recomendación a otros compañeros, la posibilidad de mantener su uso después del estudio y el interés por conocer nuevas aplicaciones educativas basadas en IA. De manera complementaria, la dimensión de **satisfacción general**, también compuesta por cuatro ítems, recoge la apreciación global del estudiante respecto a su experiencia con estas herramientas, tomando en cuenta el grado de cumplimiento de sus expectativas, la comparación con métodos tradicionales de aprendizaje y el nivel de satisfacción experimentado durante su utilización (Alshahrani, 2023; Moorhouse et al., 2023).

3.5.4 Ficha de Observación Sistemática (FOS)

La Ficha de Observación Sistemática constituye un instrumento de registro estructurado diseñado para capturar de manera objetiva y sistemática los comportamientos observables de los estudiantes durante las sesiones de programación con apoyo de inteligencia artificial. Este instrumento complementa las medidas de auto-reporte proporcionando datos conductuales directos que no están sujetos a los sesgos de deseabilidad social o las limitaciones de la introspección. El desarrollo de la FOS se basó en protocolos establecidos de observación conductual en contextos educativos, adaptados específicamente para capturar las particularidades de la interacción estudiante-IA en el aprendizaje de programación.

La estructura de la FOS organiza las observaciones en cuatro categorías conductuales

principales, cada una con indicadores específicos operacionalmente definidos. La categoría de participación activa registra la frecuencia y calidad de las intervenciones del estudiante durante las sesiones, incluyendo el número de preguntas formuladas al docente o al sistema de IA, distinguiendo entre preguntas de clarificación, preguntas de profundización y preguntas que indican pensamiento crítico. También registra las respuestas voluntarias a preguntas del docente, los aportes espontáneos a las discusiones grupales, y la iniciativa para compartir soluciones o estrategias con los compañeros.

La categoría de colaboración evalúa las interacciones del estudiante con sus pares en el contexto del aprendizaje colaborativo mediado por IA. Se registra la frecuencia con que el estudiante ofrece ayuda a compañeros que experimentan dificultades, la receptividad a recibir ayuda de otros, la participación en actividades de programación en parejas o grupos pequeños, y la calidad de las contribuciones en proyectos colaborativos. Esta categoría es particularmente relevante dado que las herramientas de IA podrían potencialmente reducir la interacción entre pares si los estudiantes prefieren consultar al sistema antes que a sus compañeros.

3.5.5 Rúbrica de Calidad de Código (RCC)

La Rúbrica de Calidad de Código constituye un instrumento de evaluación criterial diseñado para medir de manera objetiva e integral la calidad del código producido por los estudiantes, más allá de su mera corrección funcional. Esta rúbrica parte del reconocimiento de que, en el desarrollo profesional de software, la calidad del código no se limita a que el programa funcione, sino que incluye aspectos fundamentales como la eficiencia, la legibilidad, la mantenibilidad y la adherencia a estándares reconocidos de la industria (Becker et al., 2023).

El desarrollo de la RCC se fundamentó en estándares internacionales de calidad de software, como la norma ISO/IEC 25010:2023 (International Organization for Standardization,

2023), así como en guías de estilo de programación ampliamente aceptadas, como PEP 8 para Python (Python Software Foundation, 2001). Además, se consideraron buenas prácticas contemporáneas orientadas a la escritura de código limpio, legible y mantenible, documentadas en la literatura de ingeniería de software (Stol & Fitzgerald, 2020).

La estructura de la RCC se organiza en cuatro dimensiones principales, cada una con un peso específico en la evaluación global y criterios distribuidos en cuatro niveles de desempeño. La dimensión de funcionalidad, que representa el cuarenta por ciento de la evaluación total, examina el grado en que el código cumple con los requisitos especificados y maneja adecuadamente diferentes escenarios de ejecución, incluyendo casos extremos y situaciones excepcionales (Keuning et al., 2023).

3.6 Procedimientos

El procedimiento de investigación se estructuró en cinco fases sistemáticas durante un período de 14 semanas, desde la primera semana de marzo hasta la última semana de julio de 2025. Cada fase fue cuidadosamente diseñada para garantizar la validez interna del estudio y minimizar las amenazas a la confiabilidad de los datos recopilados.

3.6.1 Fase de Preparación y Coordinación (Semanas 1-2)

La fase inicial consistió en la obtención de las autorizaciones institucionales necesarias. Se presentó el proyecto de investigación ante la Dirección Académica del instituto de educación superior tecnológico, adjuntando el protocolo de investigación aprobado por el comité de ética institucional. Tras la revisión del proyecto, que tomó aproximadamente cinco días hábiles, se obtuvo la carta de autorización formal que permitió el acceso a las instalaciones, estudiantes y recursos tecnológicos necesarios para la implementación de la intervención.

Paralelamente, se realizó la coordinación con el docente titular del curso de Programación III para establecer los horarios de aplicación de instrumentos y la logística de

implementación de las herramientas de inteligencia artificial. Se acordó que las evaluaciones pre-test se realizarían durante la segunda semana del semestre, permitiendo que los estudiantes tuvieran una semana de adaptación al inicio del ciclo académico, pero evitando que avanzaran significativamente en los contenidos del curso.

La preparación técnica incluyó la configuración de los dos laboratorios de cómputo designados para el estudio. Se verificó que cada una de las 35 estaciones de trabajo cumpliera con las especificaciones técnicas mínimas establecidas: procesador Intel Core i5 de décima generación o superior, 8 GB de RAM, disco de estado sólido de 256 GB, y sistema operativo Windows 10 Professional actualizado. Se instaló el software necesario en todas las computadoras, incluyendo Java 3.9, el entorno de desarrollo integrado Visual Studio Code con extensiones relevantes, y los clientes para acceso a las plataformas de IA educativa.

Se realizaron pruebas exhaustivas de conectividad para asegurar que la conexión a internet de fibra óptica de 100 Mbps simétricos funcionara óptimamente. Las pruebas incluyeron verificación de latencia (promedio de 15 ms), estabilidad de la conexión durante períodos prolongados (sin interrupciones durante 4 horas continuas), y capacidad de soporte de acceso simultáneo de 35 estudiantes a las plataformas cloud-based sin degradación del servicio.

3.6.2 Fase de Evaluación Inicial (Semana 3)

La aplicación del pre-test se realizó durante una sesión de tres horas en el laboratorio de cómputo. Los 30 estudiantes participantes fueron citados en un horario especial coordinado con la dirección académica para no interferir con otras asignaturas. Antes de iniciar la evaluación, se dedicaron 20 minutos a explicar detalladamente los objetivos de la investigación, el carácter voluntario de la participación, y los procedimientos de confidencialidad que se implementarían para proteger su identidad.

Los estudiantes recibieron y firmaron el consentimiento informado después de tener oportunidad de formular preguntas y aclarar dudas. Se enfatizó que su decisión de participar o

no participar en el estudio no afectaría de ninguna manera sus calificaciones en el curso ni su relación con la institución. Adicionalmente, se les informó sobre su derecho a retirarse del estudio en cualquier momento sin necesidad de justificación.

La administración de los instrumentos siguió un orden estandarizado para todos los participantes. Primero, se aplicó el Test de Conocimientos de Programación (TCP-J), con una duración de 90 minutos. Durante este tiempo, el investigador y dos asistentes de investigación previamente capacitados circularon por el laboratorio para resolver dudas procedimentales sin proporcionar ayuda sobre el contenido de las evaluaciones. Se instruyó a los estudiantes para que trabajaran de manera individual y que no consultaran recursos externos ni se comunicaran entre sí.

Después de una pausa de 15 minutos, durante la cual los estudiantes pudieron tomar refrigerios proporcionados por el proyecto, se procedió a la aplicación de la Escala de Motivación Académica (AMS-C 28), que tomó aproximadamente 20 minutos. Los estudiantes respondieron mediante un formulario digital en Google Forms, lo que facilitó la posterior captura y análisis de datos. Inmediatamente después, se administró el Cuestionario TAM para IA Educativa (TAM-AIEd) con una duración de 15 minutos adicionales.

Todos los instrumentos fueron aplicados en formato digital mediante Google Forms, plataforma que ofrece protección estándar de datos mediante protocolos de seguridad Google Workspace, incluyendo transmisión segura HTTPS y almacenamiento en servidores con respaldo automático. Se asignó a cada participante un código alfanumérico único formato: con el formato EST-2025-XXX que permitió vincular todas sus respuestas mientras preservaba su anonimato. La relación entre códigos y nombres reales se mantuvo en un archivo de Excel local con protección por contraseña, accesible únicamente al investigador principal.3.6.3 Fase de Implementación de la Intervención (Semanas 4-17).

La implementación de la intervención pedagógica basada en asistentes de inteligencia

artificial se realizó de manera progresiva durante 14 semanas, siguiendo un cronograma estructurado que permitió la familiarización gradual de los estudiantes con cada plataforma tecnológica y el desarrollo de competencias en el uso efectivo de estas herramientas para el aprendizaje de programación Java.

Semana 4: Capacitación Inicial y Configuración de Accesos, La primera semana de implementación se dedicó exclusivamente a la capacitación de estudiantes en el uso pedagógico de los asistentes de inteligencia artificial. Se diseñó un taller presencial de 4 horas distribuidas en dos sesiones de 2 horas cada una, realizadas en el laboratorio de cómputo de la institución. Durante la primera sesión, se procedió a la creación de cuentas institucionales para cada uno de los 30 estudiantes participantes en tres plataformas: ChatGPT (versión gratuita de OpenAI), Claude (de Anthropic) y Perplexity AI. Cada estudiante recibió credenciales de acceso con correos electrónicos institucionales siguiendo el formato EST-2025-XXX@instituto.edu.pe, lo que permitió posteriormente el monitoreo del uso y la gestión centralizada de las cuentas. Se explicaron las características principales de cada plataforma, sus fortalezas específicas y los casos de uso recomendados: ChatGPT para generación de código y explicaciones paso a paso, Claude para análisis detallado de errores y comprensión conceptual profunda, y Perplexity para búsqueda de documentación técnica y ejemplos de la comunidad de programadores.

Los estudiantes recibieron un manual digital de 25 páginas elaborado específicamente para esta investigación, que incluía capturas de pantalla de las tres plataformas, ejemplos de consultas efectivas e inefectivas, consideraciones éticas sobre el uso de IA en educación, y políticas institucionales sobre plagio académico y uso responsable de tecnología. Adicionalmente, se proporcionaron tres videotutoriales de 8-12 minutos de duración para cada plataforma, demostrando navegación básica, configuración de preferencias y primeros pasos en la interacción.

La segunda sesión se centró en la capacitación práctica sobre prompt engineering aplicado al aprendizaje de programación Java. El investigador presentó el concepto de prompt engineering como la habilidad de formular consultas efectivas a sistemas de IA para obtener respuestas útiles, precisas y contextualizadas. Se explicaron cinco principios fundamentales del prompt efectivo: especificidad (incluir detalles relevantes sobre el problema), contexto (proporcionar información sobre el nivel de conocimiento y objetivo de aprendizaje), claridad (usar lenguaje preciso evitando ambigüedades), estructura (organizar la consulta en partes lógicas), y reflexividad (solicitar explicaciones que promuevan comprensión profunda en lugar de soluciones directas). Se diseñaron y distribuyeron plantillas de prompts estructurados organizadas en cuatro categorías principales según el tipo de consulta:

- Plantilla para Resolución de Errores: "Estoy aprendiendo Java y tengo el siguiente error de compilación: [COPIAR MENSAJE DE ERROR]. Mi código es: [PEGAR CÓDIGO]. Soy estudiante de tercer ciclo. ¿Puedes explicarme qué causa este error y cómo solucionarlo sin darme el código corregido directamente? Ayúdame a entender el concepto."
- Plantilla para Explicación de Conceptos: "Estoy estudiando [CONCEPTO DE JAVA] en mi curso de Programación III. ¿Puedes explicarme este concepto de manera simple, dar un ejemplo práctico aplicado a [CONTEXTO], y explicar cuándo debería usarlo en mis programas?"
- Plantilla para Generación de Código con Aprendizaje: "Necesito crear un programa en Java que [DESCRIPCIÓN DE LA TAREA]. Antes de darme el código, ¿puedes explicarme qué estructuras y conceptos de Java necesito usar? Luego muéstrame el código con comentarios explicativos detallados."

- Plantilla para Análisis de Algoritmos: "Escribí este código en Java: [PEGAR CÓDIGO]. ¿Puedes analizar si mi lógica es correcta, sugerir mejoras en términos de eficiencia y legibilidad, y explicarme por qué tus sugerencias son mejores que mi enfoque original?"
- Durante esta sesión práctica, cada estudiante formuló al menos tres consultas diferentes utilizando las plantillas proporcionadas, recibió retroalimentación del investigador sobre la efectividad de sus prompts, y experimentó refinando sus consultas para obtener respuestas más útiles. Se enfatizó la importancia de no solicitar soluciones completas sin comprensión, sino de usar la IA como un tutor que guía el razonamiento propio.

Adicionalmente, se abordaron consideraciones éticas fundamentales: se prohibió explícitamente copiar código generado por IA sin comprenderlo, se requirió que todo código utilizado de la IA fuera analizado, modificado y documentado por el estudiante, y se estableció que las evaluaciones individuales (exámenes y pruebas) se realizarían sin acceso a herramientas de IA para verificar el aprendizaje genuino.

Semanas 5-17: Uso Regular e Integrado de Asistentes de IA

Durante las siguientes 13 semanas, los asistentes de inteligencia artificial se integraron completamente en la rutina de aprendizaje del curso de Programación III (Java). La estructura semanal típica incluyó las siguientes actividades:

Lunes: Clase presencial de 2 horas donde el docente presentaba nuevos conceptos teóricos de programación orientada a objetos (clases, herencia, polimorfismo, interfaces, manejo de excepciones) utilizando metodología expositiva tradicional complementada con demostraciones en vivo de código. Los estudiantes tomaban apuntes y realizaban preguntas de clarificación.

Martes-Jueves: Trabajo autónomo donde los estudiantes practicaban los conceptos mediante ejercicios de programación progresivamente más complejos asignados por el docente. Durante estos días, tenían acceso ilimitado a los tres asistentes de IA para resolver dudas, depurar errores y profundizar comprensión. Se les instruyó utilizar las plantillas de prompts proporcionadas, adaptándolas según sus necesidades específicas. Los estudiantes trabajaban desde sus hogares o el laboratorio de cómputo institucional, que permanecía abierto hasta las 8 PM.

El investigador recopiló evidencia anecdótica de cómo los estudiantes utilizaban las plataformas: muchos preferían ChatGPT para obtener rápidamente ejemplos de código iniciales, luego usaban Claude para analizar detalladamente por qué ese código funcionaba y cómo mejorarlo, y finalmente consultaban Perplexity cuando necesitaban entender cómo implementaciones similares se realizaban en proyectos profesionales reales. Esta triangulación entre plataformas emergió naturalmente sin haber sido prescrita explícitamente.

Viernes: Sesión de laboratorio presencial de 3 horas donde los estudiantes desarrollaban proyectos más extensos que integraban múltiples conceptos aprendidos durante la semana. Durante estas sesiones, tenían acceso a los asistentes de IA en sus laptops personales o computadoras del laboratorio, mientras el docente circulaba proporcionando mentoría personalizada enfocada en aspectos de diseño de software, optimización de algoritmos y mejores prácticas profesionales que las IA no siempre capturaban adecuadamente.

Fin de semana: Los estudiantes completaban un proyecto semanal más sustancial que era enviado a través del aula virtual de la institución (plataforma ClasRoom). Estos proyectos eran evaluados manualmente por el docente considerando corrección funcional, calidad del código, y claridad de documentación. El docente proporcionaba retroalimentación escrita detallada sobre cada entrega, complementando así la retroalimentación inmediata que los estudiantes

obtenían de las IA durante el proceso de desarrollo.

A lo largo de las 13 semanas, la complejidad de los proyectos aumentó sistemáticamente: desde programas simples de entrada/salida y estructuras de control en las semanas 5-7, pasando por implementación de clases y objetos en las semanas 8-10, hasta proyectos que requerían herencia, polimorfismo y manejo de excepciones en las semanas 11-13, culminando con un proyecto integrador durante las semanas 14-17 que sintetizaba todos los conceptos del curso.

3.6.4 Fase de Monitoreo y Ajustes (Semanas 4-17)

Durante todo el período de implementación, se realizó un monitoreo sistemático para identificar y resolver problemas técnicos o pedagógicos emergentes. El investigador se reunió semanalmente con el docente titular del curso para revisar el progreso general de los estudiantes, discutir observaciones relevantes sobre el uso de los asistentes de IA, y ajustar estrategias pedagógicas según fuera necesario. Estas reuniones de coordinación, que duraban aproximadamente 45 minutos cada una, permitieron mantener un seguimiento cercano de la implementación y tomar decisiones informadas sobre modificaciones al diseño instruccional.

El monitoreo del uso de las plataformas de IA se realizó mediante dos mecanismos principales. Primero, se solicitó a los estudiantes llevar un registro semanal simplificado del tiempo aproximado dedicado al uso de cada asistente (ChatGPT, Claude, Perplexity) y el tipo de consultas realizadas (resolución de errores, comprensión de conceptos, generación de código, o análisis de algoritmos). Este registro se recopilaba mediante un formulario breve de Google Forms cada viernes, tomando aproximadamente 5 minutos completarlo. Los datos permitieron identificar patrones de uso, preferencias entre plataformas, y estudiantes que pudieran estar subutilizando o sobreutilizando las herramientas.

Segundo, durante las sesiones de laboratorio presenciales de los viernes, el investigador y el docente realizaban observación directa y no participativa del comportamiento de los

estudiantes al interactuar con los asistentes de IA. Se registraban observaciones sobre aspectos como: frecuencia de consultas durante la sesión de tres horas, naturaleza de las preguntas formuladas, tiempo dedicado a leer y analizar las respuestas recibidas, evidencia de iteración (refinamiento de prompts), y casos donde los estudiantes compartían estrategias efectivas de consulta con sus compañeros. Estas observaciones cualitativas se documentaban en un cuaderno de campo del investigador y se complementaban con las notas del docente.

Adicionalmente, se aplicaron mini-encuestas de satisfacción cada dos semanas (total de 6 aplicaciones durante las 13 semanas de uso regular), consistentes en 6 preguntas cerradas con escala Likert de 5 puntos y una pregunta abierta. Estas encuestas breves, con duración de 3-5 minutos para completar, evaluaban dimensiones específicas de la experiencia estudiantil: facilidad de uso percibida de cada plataforma, utilidad de las respuestas recibidas para el aprendizaje real (no solo para completar tareas), nivel de frustración experimentado cuando las respuestas no eran satisfactorias, confianza en la exactitud de la información proporcionada, y percepción de mejora en sus habilidades de programación. La pregunta abierta solicitaba comentarios sobre dificultades específicas o sugerencias de mejora.

Los datos recopilados mediante los registros semanales y las mini-encuestas se analizaban quincenalmente. El investigador generaba reportes descriptivos simples (promedios, frecuencias) utilizando hojas de cálculo de Excel, identificando tendencias como: estudiantes que reportaban menos de 2 horas semanales de uso (posible subutilización), estudiantes que reportaban frustración consistentemente alta con alguna plataforma específica (necesidad de soporte técnico o capacitación adicional), y patrones de uso que diferían significativamente de lo esperado. Basándose en estos análisis, se implementaron ajustes menores en tiempo real.

Los ajustes implementados incluyeron: (1) Sesión de reforzamiento sobre prompt

engineering en la semana 8, cuando se identificó que varios estudiantes continuaban formulando preguntas excesivamente vagas que generaban respuestas poco útiles; (2) Creación de un grupo de WhatsApp del curso donde estudiantes compartían ejemplos de prompts efectivos y estrategias exitosas, fomentando el aprendizaje entre pares; (3) Recordatorios personalizados enviados por correo electrónico a estudiantes que mostraban baja utilización de las herramientas, animándolos a experimentar más activamente; (4) Aclaración de la política de uso ético después de detectar, mediante observación directa, que algunos estudiantes copiaban código sin comprenderlo.

Cuando los datos de monitoreo indicaban que un estudiante específico podría estar experimentando dificultades significativas (por ejemplo, reportando consistentemente alta frustración o muy bajo tiempo de uso), el investigador coordinaba con el docente para realizar una intervención individualizada. Estas intervenciones consistían en conversaciones breves de 10-15 minutos para diagnosticar el problema específico (dificultades técnicas de acceso, incomprensión sobre cómo formular consultas efectivas, o desconfianza en la utilidad de las herramientas) y proporcionar soporte personalizado. Se realizaron aproximadamente 8 intervenciones individuales durante las 14 semanas, con estudiantes diferentes en cada caso.

Toda la información recopilada mediante los registros semanales, mini-encuestas, observaciones de laboratorio, y notas de las reuniones de coordinación se consolidó en una base de datos maestro en formato Excel, manteniendo la codificación anónima de participantes (EST-2025-XXX). Esta base de datos sirvió posteriormente para análisis complementarios del estudio y para documentar el proceso de implementación de manera transparente. Los ajustes realizados y sus justificaciones se documentaron en un registro de cambios, permitiendo trazar la evolución de la intervención a lo largo de las 14 semanas.

3.6.5 Fase de Evaluación Final (Semana 18)

La aplicación del post-test se realizó durante la semana 18 del semestre académico, utilizando procedimientos idénticos a los del pre-test para garantizar comparabilidad metodológica de los datos. Los mismos 30 estudiantes que completaron la evaluación inicial fueron citados a una sesión de evaluación final de 3 horas de duración en el mismo laboratorio de cómputo donde se realizó el pre-test, manteniendo condiciones ambientales y técnicas equivalentes. Se administraron los tres instrumentos principales en el mismo orden establecido durante el pre-test para controlar posibles efectos de secuencia: (1) Test de Conocimientos de Programación Java (TCP-J) con duración de 90 minutos, evaluando conocimientos teóricos y habilidades prácticas de programación; (2) Escala de Motivación Académica (AMS-C28) con duración de 20 minutos, midiendo motivación intrínseca, extrínseca y amotivación; y (3) Cuestionario TAM para IA Educativa (TAM-AIEd) con duración de 15 minutos, evaluando percepción de utilidad y facilidad de uso de los asistentes de IA utilizados durante la intervención.

Los tres instrumentos se implementaron mediante formularios digitales de Google Forms, configurados para requerir respuesta obligatoria en todos los ítems antes de permitir el envío, estrategia que garantizó la completitud de los datos y eliminó valores perdidos. Cada estudiante accedió a los formularios mediante un enlace único personalizado que incluía su código de participante (EST-2025-XXX), permitiendo vincular automáticamente las respuestas del post-test con las del pre-test mientras preservaba el anonimato, ya que la relación entre códigos y nombres reales se mantuvo en un archivo separado con acceso restringido al investigador principal.

Adicionalmente al protocolo del pre-test, se incluyó un cuestionario cualitativo breve de retroalimentación con 10 preguntas abiertas diseñadas para capturar percepciones, experiencias y sugerencias que los instrumentos cuantitativos no pudieran captar

completamente. Las preguntas exploraron: (1) aspectos específicos de los asistentes de IA que los estudiantes consideraron más útiles para su aprendizaje; (2) dificultades o frustraciones experimentadas durante el uso de las herramientas; (3) comparación subjetiva entre aprender con y sin soporte de IA; (4) cambios percibidos en su confianza y competencia en programación Java; (5) estrategias de formulación de prompts que resultaron más efectivas; (6) situaciones donde los asistentes de IA proporcionaron información incorrecta o confusa; (7) impacto en su motivación para continuar estudiando programación; (8) sugerencias para mejorar la integración de IA en futuros cursos; (9) preocupaciones éticas sobre el uso de IA en educación; y (10) intención de continuar usando estas herramientas en cursos posteriores.

Los estudiantes completaron este cuestionario cualitativo después de finalizar los tres instrumentos cuantitativos, tomando aproximadamente 20-30 minutos adicionales. Las respuestas se recopilaron mediante un formulario separado de Google Forms con preguntas de texto largo, permitiendo respuestas extensas y reflexivas sin restricciones de caracteres.

Todos los datos cuantitativos fueron exportados automáticamente desde Google Forms a archivos en formato CSV (valores separados por comas), que posteriormente se importaron al software IBM SPSS Statistics versión 29 para análisis estadístico. El proceso de exportación e importación se realizó el mismo día de la aplicación del post-test para minimizar riesgos de pérdida de datos y permitir verificaciones preliminares de calidad.

Se ejecutaron verificaciones exhaustivas de calidad de datos siguiendo protocolos estándar de limpieza de datos en investigación cuantitativa. Primero, se verificó la completitud confirmando que los 30 casos (estudiantes) tuvieran respuestas para todas las variables, lo cual se cumplió debido a la configuración de respuesta obligatoria en los formularios. Segundo, se identificaron y examinaron valores atípicos mediante análisis de diagramas de caja (boxplots) y cálculo de puntuaciones z estandarizadas, estableciendo como criterio que valores con $|z| > 3.29$ serían investigados como posibles errores de entrada. No se encontraron valores que

cumplieran este criterio extremo. Tercero, se verificó la consistencia lógica de las respuestas, confirmando que todos los valores estuvieran dentro de los rangos válidos para cada escala (por ejemplo, 0-20 puntos para TCP-J, 1-7 para AMS-C28, 1-5 para TAM-AIEd).

Las respuestas cualitativas del cuestionario de retroalimentación se exportaron a un documento de Microsoft Word y posteriormente se analizaron mediante análisis de contenido temático, identificando patrones recurrentes, temas emergentes y citas representativas que complementarían la interpretación de los hallazgos cuantitativos. Se observa un incremento en los niveles de motivación de los estudiantes entre las mediciones pretest y posttest.

3.7 Análisis de datos

El análisis de datos se estructuró en tres niveles progresivos de complejidad metodológica: análisis descriptivo para caracterizar la distribución de las variables, verificación de supuestos estadísticos para determinar la validez de las pruebas paramétricas, y análisis inferencial para contrastar las hipótesis de investigación. Todos los procedimientos estadísticos se ejecutaron utilizando el software IBM SPSS Statistics versión 29.0, estableciendo un nivel de significancia de $\alpha = 0.05$ para todas las pruebas de hipótesis, siguiendo las recomendaciones contemporáneas para investigación cuantitativa en ciencias sociales y educación.

Análisis Descriptivo. Se calcularon estadísticos descriptivos para todas las variables en ambos momentos de medición (pre-test y post-test). Para las variables continuas (puntajes de TCP-J, subescalas de AMS-C28, dimensiones de TAM-AIEd), se computaron medidas de tendencia central (media aritmética, mediana), medidas de dispersión (desviación estándar, rango, rango intercuartílico), y medidas de forma de la distribución (coeficiente de asimetría y curtosis). Estos estadísticos proporcionaron una caracterización inicial de los datos, permitiendo identificar patrones generales de cambio entre pre-test y post-test, así como la presencia de posibles valores atípicos o distribuciones no normales que pudieran afectar análisis posteriores.

Verificación de Supuestos Estadísticos. Antes de proceder con pruebas paramétricas, se verificó el cumplimiento de sus supuestos fundamentales. El supuesto de normalidad de las distribuciones se evaluó mediante la prueba de Shapiro-Wilk, considerada el test más robusto para muestras pequeñas ($n < 50$). Se estableció como criterio que valores de $p > 0.05$ indicarían que los datos no difieren significativamente de una distribución normal, justificando el uso de pruebas paramétricas. Complementariamente, se examinaron gráficos Q-Q (quantile-quantile plots) para inspección visual de la normalidad. El supuesto de homocedasticidad (igualdad de varianzas) se evaluó mediante la prueba de Levene cuando fue aplicable. Dado que el diseño es de medidas repetidas con el mismo grupo de participantes, el supuesto de independencia de observaciones se cumple automáticamente entre sujetos, aunque se reconoce la dependencia intra-sujeto entre mediciones pre y post que es inherente al diseño.

Análisis Inferencial. Para contrastar las hipótesis de investigación, se utilizó la prueba t de Student para muestras relacionadas (también conocida como prueba t pareada o dependiente), que es el procedimiento estadístico apropiado para comparar las medias de dos mediciones del mismo grupo de participantes. Esta prueba evaluó si las diferencias observadas entre pre-test y post-test en cada variable dependiente eran estadísticamente significativas, es decir, si era improbable que ocurrieran por azar si la intervención no tuviera efecto real. Se reportaron los estadísticos t, grados de libertad ($gl = n-1 = 29$), valor p bilateral, e intervalos de confianza del 95% para la diferencia de medias.

Adicionalmente a la significancia estadística, se calculó el tamaño del efecto mediante la d de Cohen para muestras relacionadas, que cuantifica la magnitud práctica del cambio independientemente del tamaño muestral. Se utilizaron los criterios convencionales de Cohen para interpretar el tamaño del efecto: $d = 0.20$ (pequeño), $d = 0.50$ (mediano), $d = 0.80$ (grande). También se reportó el coeficiente de correlación r como medida complementaria del tamaño del efecto, interpretándose como el porcentaje de varianza explicada (r^2).

Los resultados se presentaron siguiendo las directrices de la American Psychological Association (APA) séptima edición para reporte de análisis estadísticos, incluyendo todos los estadísticos relevantes que permitan a otros investigadores replicar los análisis o realizar meta-análisis futuros. Se utilizaron tablas para presentar resultados de manera estructurada y figuras (gráficos de barras, diagramas de caja) para visualizar patrones de cambio entre pre-test y post-test en las diferentes dimensiones evaluadas.

3.7.1 Preparación y Limpieza de Datos

Previo a cualquier análisis estadístico, se realizó un proceso exhaustivo de preparación de datos con el objetivo de garantizar la calidad e integridad de la información. Los archivos CSV exportados desde las plataformas de evaluación fueron importados a SPSS, creando una base de datos maestra con 30 casos (estudiantes) y 16 variables principales, correspondientes a las mediciones pretest y posttest de las dimensiones evaluadas. Se verificó la completitud de los datos, confirmando la ausencia de valores faltantes en todas las variables. Esta condición se logró gracias al diseño de los instrumentos digitales, los cuales requerían respuesta obligatoria en todos los ítems antes de permitir el envío del formulario, lo que permitió minimizar la pérdida de datos durante la recolección de la información.

3.8 Consideraciones éticas

La presente investigación se desarrolló en estricto cumplimiento de los principios éticos para estudios en educación superior, siguiendo las recomendaciones internacionales de la UNESCO (2021, 2023) sobre ética de la inteligencia artificial y la normativa nacional vigente (Floridi & Cowls, 2022; Holmes et al., 2022). Todos los participantes fueron informados sobre los objetivos, procedimientos, beneficios y posibles riesgos del estudio, y otorgaron su consentimiento informado de manera voluntaria y por escrito, cumpliendo con los estándares éticos para investigación educativa con tecnologías emergentes.

Se garantizó la confidencialidad y el anonimato de los datos personales y académicos de los estudiantes mediante el uso de códigos alfanuméricos y el almacenamiento de la información en sistemas protegidos, respetando los principios generales de protección de datos personales. Asimismo, el uso de herramientas de inteligencia artificial se realizó bajo criterios de transparencia, equidad y no discriminación, procurando minimizar posibles sesgos algorítmicos que pudieran afectar la equidad de acceso o los resultados académicos, en concordancia con los lineamientos éticos internacionales para la inteligencia artificial en educación (UNESCO, 2021).

Asimismo, se promovió la integridad académica y la responsabilidad en el uso de sistemas automatizados, asegurando que la IA complementara, pero no reemplazara, la labor docente y la evaluación humana (Chan & Zhou, 2023). Consideró los posibles riesgos de sesgo, discriminación y dependencia tecnológica, y se tomaron medidas para mitigarlos, como la revisión crítica de los resultados generados por la IA y la capacitación ética de los participantes (Akgun & Greenhow, 2022; Holmes & Tuomi, 2022).

IV: RESULTADOS

4.1 Verificación de supuestos estadísticos

Previo a la realización de los análisis inferenciales, se verificó el cumplimiento del supuesto de normalidad mediante la prueba de Shapiro-Wilk para todas las variables en ambos momentos de medición (pre-test y post-test). Los resultados, presentados en la Tabla 1, evidencian que todas las variables siguieron una distribución normal, con valores p superiores a .05, cumpliendo así el criterio para la aplicación de pruebas paramétricas.

Tabla 5

Prueba de Normalidad de Shapiro-Wilk para Variables de Estudio

Dimensión	Variable	Momento	n	W	p	Distribución
Rendimiento Académico	TCP-J Total	Pre	30	0.976	.682	Normal
		Post	30	0.968	.487	Normal
	Conocimientos Teóricos	Pre	30	0.981	.851	Normal
		Post	30	0.973	.621	Normal
	Aplicación Práctica	Pre	30	0.979	.789	Normal
		Post	30	0.971	.572	Normal
Motivación y Compromiso	Motivación Intrínseca	Pre	30	0.972	.598	Normal
		Post	30	0.965	.421	Normal
	Motivación Extrínseca	Pre	30	0.974	.641	Normal
		Post	30	0.983	.892	Normal
	Amotivación	Pre	30	0.977	.721	Normal

		Post	30	0.969	.512	Normal
Percepción Tecnológica	Utilidad Percibida	Pre	30	0.980	.812	Normal
		Post	30	0.975	.671	Normal
	Facilidad de Uso	Pre	30	0.978	.763	Normal
		Post	30	0.971	.584	Normal

Nota. W = Estadístico Shapiro-Wilk; p = nivel de significancia. Todas las variables presentaron distribución normal ($p > .05$), justificando el uso de pruebas paramétricas. Los valores del estadístico W oscilaron entre 0.965 y 0.983, todos cercanos a 1.0, lo que indica distribuciones muy próximas a la normalidad teórica. Estos resultados confirmaron que era aplicable la prueba t de Student para muestras relacionadas.

4.2 Análisis Descriptivo

La Tabla 2 presenta los estadísticos descriptivos para todas las variables evaluadas, organizadas según las tres dimensiones del estudio: Rendimiento Académico, Motivación y Compromiso, y Percepción Tecnológica. Se observan diferencias descriptivas entre las mediciones pre y post que sugieren cambios en la dirección esperada.

Tabla 6

Estadísticos Descriptivos por Dimensión y Momento de Medición

Dimensión	Variable	Pre-			Post-		
		test			test		
		M	DE	Min-Max	M	DE	Min-Max
Rendimiento Académico	TCP-J Total	19.40	2.13	15.67- 23.45	29.87	2.31	25.89- 33.67
	Conocimientos Teóricos	10.47	1.09	8.45- 12.23	15.93	1.20	13.56- 17.89
	Aplicación Práctica	8.93	1.05	7.11-	13.93	1.11	11.67-

		11.11					15.88
Motivación y Compromiso	Motivación Intrínseca	4.23	0.53	3.45-5.45	5.87	0.41	5.01- 6.45
	Motivación Extrínseca	5.12	0.33	4.34-5.89	4.93	0.22	4.56- 5.45
	Amotivación	2.87	0.24	2.34-3.34	1.73	0.21	1.45- 2.12
Percepción Tecnológica	Utilidad Percibida	3.67	0.29	3.12-4.23	4.67	0.20	4.23- 4.98
	Facilidad de Uso	2.93	0.33	2.34-3.67	4.27	0.24	3.78- 4.78

Nota. M = Media; DE = Desviación Estándar; Min-Max = Rango de valores observados; n = 30 participantes.

En la dimensión de Rendimiento Académico, se observaron incrementos notables en todas las variables, con el puntaje total del TCP-J aumentando en promedio 10.47 puntos. En Motivación y Compromiso, la motivación intrínseca mostró un aumento considerable, mientras que la amotivación presentó una disminución marcada. La dimensión de Percepción Tecnológica evidenció mejoras en ambas variables evaluadas, particularmente en la facilidad de uso percibida.

4.3 Análisis Inferencial

4.3.1 Rendimiento Académico

Los resultados de la prueba t de Student para muestras relacionadas evidenciaron mejoras estadísticamente significativas en todas las variables de rendimiento académico (ver Tabla 3). El puntaje total del TCP-J mostró un incremento promedio de 10.47 puntos, pasando

de $M = 19.40$ ($DE = 2.13$) en el pre-test a $M = 29.87$ ($DE = 2.31$) en el post-test, $t(29) = 12.89$, $p < .001$, con un tamaño del efecto muy grande ($d = 1.98$, IC 95% [8.45, 12.49]), explicando el 92% de la varianza ($r = .92$).

En el subdimensión de **Conocimientos Teóricos**, se observó un aumento de 5.46 puntos, $t(29) = 10.34$, $p < .001$, $d = 2.17$, mientras que en Aplicación Práctica el incremento fue de 5.00 puntos, $t(29) = 8.76$, $p < .001$, $d = 1.62$. Ambos resultados presentaron tamaños del efecto que superan ampliamente el umbral de "grande" establecido por Cohen (1988), indicando no solo significancia estadística sino también relevancia práctica sustancial. Estos hallazgos son consistentes con estudios previos que reportan resultados favorables del uso de inteligencia artificial en contextos educativos similares.

4.3.2 Motivación y Compromiso.

El análisis de las variables motivacionales reveló patrones diferenciados. La Motivación Intrínseca experimentó un aumento significativo de 1.64 puntos (38.8%), $t(29) = 9.45$, $p < .001$, $d = 1.99$, indicando que los estudiantes desarrollaron un interés genuino y autodirigido hacia la programación tras la experiencia con realidad virtual.

La Amotivación mostró una reducción significativa de 1.14 puntos (-39.7%), $t(29) = -6.78$, $p < .001$, $d = -1.38$, evidenciando que la intervención logró disminuir considerablemente la falta de motivación y los sentimientos de impotencia frente a la programación que pudieran haber experimentado los estudiantes inicialmente. Sin embargo, la Motivación Extrínseca no presentó cambios estadísticamente significativos, con una reducción marginal de 0.19 puntos (-3.7%), $t(29) = -1.23$, $p = .230$, $d = -0.24$, IC 95% [-0.51, 0.13]. Este resultado sugiere que la intervención no modificó sustancialmente las motivaciones orientadas a recompensas externas, calificaciones o aprobación social, manteniéndose relativamente estables a lo largo del estudio.

4.3.3 Percepción Tecnológica.

Ambas variables mostraron mejoras estadísticamente significativas y con tamaños del efecto grandes (ver Tabla 5). La Utilidad Percibida aumentó 1.00 puntos (27.2%), $t(29) = 7.12$, $p < .001$, $d = 1.43$, IC 95% [0.72, 1.28], indicando que los estudiantes reconocieron el valor práctico y la relevancia del aporte de la Inteligencia Artificial como herramienta para el aprendizaje de programación.

La Facilidad de Uso percibida se incrementó en 1.34 puntos (45.7%), $t(29) = 9.87$, $p < .001$, $d = 2.04$, IC 95% [1.06, 1.62], evidenciando que los participantes consideraron la tecnología de realidad virtual como intuitiva y fácil de manejar, reduciendo barreras tecnológicas para su adopción educativa.

Tabla 7
Resultados de la Prueba t de Student para Muestras Relacionadas

Dimensión	Variable	Diferencia M	t(29)	p	d Cohen	IC 95% Inf	Sup	r
Rendimiento Académico TCP-J Total		10.47	12.89	<	1.98	8.45	12.49	.92
	Conocimientos Teóricos	5.46	10.34	<	2.17	4.38	6.54	.88
	Aplicación Práctica	5.00	8.76	<	1.62	3.83	6.17	.85
Motivación Compromiso	y Motivación Intrínseca	1.64	9.45	<	1.99	1.29	1.99	.87
	Motivación Extrínseca	-0.19	-1.23	.230	-0.24	-0.51	0.13	.22
	Amotivación	-1.14	-6.78	<	-1.38	-1.48	-0.80	.78

Dimensión	Variable	Diferencia M	t(29)	p	d Cohen	IC 95%	r
				.001***			
Percepción Tecnológica	Utilidad Percibida	1.00	7.12	<	1.43	0.72	1.28 .80
				.001***			
Uso	Facilidad de	1.34	9.87	<	2.04	1.06	1.62 .88
				.001***			

Nota. M = Media; t = estadístico t; gl = grados de libertad; p = nivel de significancia; d = d de Cohen; IC = Intervalo de confianza; r = coeficiente de correlación. ***p < .001.

La Tabla 7 presenta la interpretación contextualizada de los tamaños del efecto observados. Siete de las ocho variables evaluadas presentaron tamaños del efecto clasificados como "grandes" o "muy grandes" según los criterios de Cohen (1988), superando ampliamente los umbrales típicos observados en investigación educativa.

Tabla 8
Tamaños del Efecto e Interpretación Contextualizada

Dimensión	Variable	d Cohen	Magnitud	Interpretación
Rendimiento Académico	TCP-J Total	1.98	Muy grande	Incremento excepcional en conocimientos de programación
	Conocimientos Teóricos	2.17	Muy grande	Mejora sustancial en comprensión teórica
	Aplicación Práctica	1.62	Grande	Avance significativo en habilidades prácticas
Motivación y	Motivación	1.99	Muy grande	Aumento notable del interés

Dimensión	Variable	d Cohen	Magnitud	Interpretación
Compromiso	Intrínseca			intrínseco
	Motivación	-0.24	Pequeño	Sin cambio significativo ($p > .05$)
	Extrínseca			
	Amotivación	-1.38	Grande	Reducción considerable de la amotivación
Percepción Tecnológica	Utilidad Percibida	1.43	Grande	Mejora sustancial en percepción de utilidad
	Facilidad de Uso	2.04	Muy grande	Incremento excepcional en facilidad percibida

Nota. Interpretación del tamaño del efecto según Cohen (1988): pequeño ($d = 0.20$), mediano ($d = 0.50$) y grande ($d = 0.80$). Los valores positivos indican mejora en el grupo experimental.

Particularmente notables son los tamaños del efecto superiores a 2.0 en Conocimientos Teóricos ($d = 2.17$) y Facilidad de Uso ($d = 2.04$), que indican cambios de magnitud excepcional, raramente observados en intervenciones educativas. Estos resultados sugieren que la realidad virtual inmersiva no solo produjo mejoras estadísticamente significativas, sino que generó transformaciones profundas en la comprensión conceptual y en la percepción de accesibilidad tecnológica.

4.4 Contraste de Hipótesis

La Tabla 8 presenta un resumen del contraste de las hipótesis de investigación, evidenciando que siete de las ocho hipótesis planteadas fueron sustentadas con evidencia estadística significativa.

Tabla 9

Resumen de Verificación de Hipótesis de Investigación

Hipótesis	Variable/Dimensión Evaluada	Resultado Estadístico	Decisión
HG	Efectividad general (TCP-J Total)	$t(29) = 12.89, p < .001, d = 1.98$	✓ Aceptada
HE ₁	Rendimiento - Conocimientos Teóricos	$t(29) = 10.34, p < .001, d = 2.17$	✓ Aceptada
HE ₁	Rendimiento - Aplicación Práctica	$t(29) = 8.76, p < .001, d = 1.62$	✓ Aceptada
HE ₂	Motivación - Motivación Intrínseca	$t(29) = 9.45, p < .001, d = 1.99$	✓ Aceptada
HE ₂	Motivación - Motivación Extrínseca	$t(29) = -1.23, p = .230, d = -0.24$	✗ Rechazada
HE ₂	Motivación - Amotivación (reducción esperada)	$t(29) = -6.78, p < .001, d = -1.38$	✓ Aceptada
HE ₃	Percepción de la Tecnología - Utilidad Percibida	$t(29) = 7.12, p < .001, d = 1.43$	✓ Aceptada
HE ₃	Percepción de la Tecnología - Facilidad de Uso	$t(29) = 9.87, p < .001, d = 2.04$	✓ Aceptada

Nota. HG = Hipótesis General; HE = Hipótesis Específica. ✓ = Hipótesis sustentada con evidencia estadística significativa; ✗ = Hipótesis no sustentada. Nivel de significancia $\alpha = .05$.

TCP-J = Test de conocimientos de programación.

4.4.1 Sustentación de las Hipótesis

Hipótesis General (HG)

La hipótesis general planteó que la implementación de herramientas de inteligencia artificial (IA) incrementaría significativamente la efectividad de la enseñanza del lenguaje de programación en estudiantes del III ciclo del programa de Computación e Informática. Los

resultados obtenidos concuerdan plenamente con esta afirmación, evidenciándose una diferencia estadísticamente significativa en la variable compuesta TCP-J Total ($t(29) = 12.89$, $p < .001$), con un tamaño de efecto muy grande ($d = 1.98$), lo que indica una mejora sustancial en el desempeño académico post intervención.

Este hallazgo respalda la evidencia de que la inteligencia artificial educativa puede optimizar los procesos de aprendizaje en contextos tecnológicos, contribuyendo no solo a la comprensión conceptual, sino también al desarrollo de competencias técnicas (Bond et al., 2024). Asimismo, la robustez del análisis se sustenta en el cumplimiento de los supuestos de normalidad y homogeneidad de varianzas, lo que refuerza la validez estadística del contraste realizado (Field, 2022; Creswell & Creswell, 2022).

Hipótesis Específicas (HE)

HE1: Incremento en rendimiento académico

Se evidenció que la implementación de la inteligencia artificial mejoró significativamente el rendimiento en las dimensiones de conocimientos teóricos ($t(29) = 10.34$, $p < .001$, $d = 2.17$) y aplicación práctica ($t(29) = 8.76$, $p < .001$, $d = 1.62$). Estos resultados reflejan un enriquecimiento integral del aprendizaje, lo cual resulta consistente con la evidencia reportada en revisiones recientes que señalan que el uso de sistemas de inteligencia artificial y algoritmos adaptativos contribuye a acelerar el progreso académico y a mejorar la calidad del trabajo estudiantil (Bond et al., 2024).

HE2: Motivación y compromiso

La motivación intrínseca mostró una mejora significativa ($t(29) = 9.45$, $p < .001$, $d = 1.99$), lo que sugiere que los estudiantes se involucraron de manera más profunda con los procesos de aprendizaje mediado por inteligencia artificial, posiblemente asociado a la retroalimentación oportuna y a la personalización de las actividades de aprendizaje. Sin embargo, la motivación extrínseca no presentó diferencias estadísticamente significativas

($t(29) = -1.23$, $p = .230$, $d = -0.24$), lo que indica que las recompensas y presiones externas no desempeñaron un papel determinante bajo este modelo didáctico, hallazgo coherente con enfoques contemporáneos del aprendizaje autorregulado (Ryan & Deci, 2020; De La Cruz Apari & Mere Jiménez, 2023). Asimismo, la amotivación decreció significativamente ($t(29) = -6.78$, $p < .001$, $d = -1.38$), evidenciando una transición positiva en el perfil motivacional de los estudiantes.

HE3: Percepción de la tecnología para el aprendizaje

El análisis estadístico reveló que la percepción de utilidad ($t(29) = 7.12$, $p < .001$, $d = 1.43$) y la facilidad de uso ($t(29) = 9.87$, $p < .001$, $d = 2.04$) aumentaron notablemente después de la implementación de IA. Según el Modelo de Aceptación Tecnológica (TAM), estas percepciones se correlacionan con una mayor adopción y persistencia en el uso de tecnologías educativas (Davis, 1989)

Interpretación General

El conjunto de resultados fortalece la hipótesis de que la IA aplicada en la educación superior tecnológica amplifica la efectividad pedagógica, favoreciendo aprendizajes profundos, motivacionalmente sostenidos y con mejor aceptación tecnológica. Este fenómeno responde a la capacidad del sistema para ofrecer soporte personalizado y adaptativo, reconocido como una ventaja disruptiva en el campo educativo (Bond et al., 2024).

La diferencia entre motivaciones intrínseca y extrínseca remitió a teorías contemporáneas de la autodeterminación, que privilegian la importancia del interés genuino y el compromiso personal frente a factores externos o recompensas (Ryan & Deci, 2020). Por ello, la no significancia estadística en motivación extrínseca se interpreta como un signo de madurez formativa de la muestra, que responde mejor a estímulos internos.

V. DISCUSIÓN DE RESULTADOS

5.1 Análisis e Interpretación de los Resultados

La presente investigación pre-experimental, de enfoque cuantitativo, realizada con 30 estudiantes del III ciclo de Computación e Informática en un instituto tecnológico de Lima, demostró que la implementación de herramientas de inteligencia artificial (IA) incrementa significativamente la efectividad de la enseñanza del lenguaje de programación. Los resultados evidenciaron mejoras notables en el rendimiento académico, la motivación intrínseca y la percepción tecnológica, confirmando la hipótesis general y la mayoría de las hipótesis específicas planteadas.

Los resultados obtenidos en la enseñanza de Java mediante asistentes de inteligencia artificial comerciales, como ChatGPT de OpenAI, Claude de Anthropic y Perplexity AI, resultan particularmente relevantes si se considera la complejidad inherente de este lenguaje. A diferencia de lenguajes de scripting como Python o JavaScript, Java exige desde las primeras etapas del aprendizaje la comprensión de conceptos abstractos como la Java Virtual Machine (JVM), la compilación a bytecode y el uso sistemático de la programación orientada a objetos. En este contexto, el hecho de que las herramientas basadas en grandes modelos de lenguaje hayan contribuido a una mejora del 53.97 % en el rendimiento académico sugiere que los LLMs contemporáneos son capaces de descomponer dicha complejidad en unidades de aprendizaje más manejables, mediante explicaciones progresivas y ejemplos contextualizados.

Asimismo, la efectividad de los asistentes conversacionales basados en LLMs para explicar conceptos avanzados de Java —como genéricos, reflexión, anotaciones y programación concurrente— resulta coherente con las observaciones de Denny et al. (2023) sobre el potencial de los grandes modelos de lenguaje en la enseñanza de lenguajes fuertemente tipados y con sistemas de tipos complejos. De manera particular, destacó la capacidad de estas herramientas para apoyar la comprensión de los mensajes de error del compilador Java,

tradicionalmente considerados una barrera significativa para estudiantes principiantes, caracterizados por su alta carga técnica y escasa claridad pedagógica (Becker et al., 2023).

- **Efectividad en rendimiento académico**

El hallazgo más relevante fue el incremento significativo en el rendimiento académico ($t(29) = 12.89, p < .001, d = 1.98$), tanto en conocimientos teóricos como en aplicación práctica. El tamaño del efecto es muy grande, lo que implica que la intervención con IA no solo mejoró las calificaciones, sino que también facilitó la transferencia de habilidades a situaciones nuevas y complejas.

Estos resultados convergen con la evidencia reportada por Bond et al. (2024), quienes identificaron mejoras promedio del 28 % en las calificaciones y una reducción significativa en el tiempo de resolución de problemas en contextos de educación superior mediante el uso de herramientas de inteligencia artificial educativa. En comparación con dichos antecedentes, la magnitud del efecto observada en el presente estudio resulta particularmente elevada, lo que podría explicarse por el uso estratégico y complementario de múltiples asistentes de inteligencia artificial comerciales de última generación. Estos asistentes, basados en arquitecturas transformer entrenadas sobre grandes volúmenes de código y documentación técnica, permiten una personalización más fina de las explicaciones y una retroalimentación inmediata ajustada al nivel de comprensión del estudiante, potenciando así los procesos de aprendizaje.

En el contexto peruano, De La Cruz Apari y Mere Jiménez (2023) encontró mejoras significativas en el aprendizaje de algoritmos con chatbots de IA en estudiantes de educación superior en Ica, mientras que Soto-Coronel et al. (2024) demostraron que la inteligencia artificial representa un alto grado de importancia como herramienta para el aprendizaje autónomo, con el 52.6% de los estudiantes universitarios peruanos considerándola siempre importante para optimizar el tiempo en la búsqueda de información académica. Estos resultados

refuerzan la aplicabilidad de la IA en contextos educativos latinoamericanos con recursos limitados y la importancia de la adaptación al contexto cultural específico mediante el uso de herramientas accesibles y de bajo costo como los asistentes de IA comerciales que requieren únicamente conexión a internet, en contraposición a sistemas propietarios costosos que muchas instituciones peruanas no podrían costear..

Desde el constructivismo (Piaget, 1972; Vygotsky, 1978), el aprendizaje significativo ocurre cuando el estudiante interactúa activamente con el conocimiento mediante procesos de asimilación y acomodación que transforman sus estructuras cognitivas existentes. Los asistentes de IA, al ofrecer retroalimentación inmediata y adaptativa que responde a las necesidades específicas evidenciadas en las consultas del estudiante, facilitan la construcción de esquemas cognitivos sólidos sobre programación orientada a objetos y la transferencia de aprendizajes a nuevos contextos problemáticos, confirmando y ampliando las predicciones teóricas constructivistas a la era de la inteligencia artificial educativa.

Teóricamente, este resultado aporta evidencia sobre el potencial de la IA para transformar el aprendizaje en contextos tecnológicos mediante la democratización del acceso a tutorización personalizada que antes solo estaba disponible mediante instrucción individual costosa. Prácticamente, sugiere que las instituciones de educación superior tecnológica deben priorizar la integración estratégica de herramientas de IA educativa comerciales para mejorar el rendimiento académico y la equidad educativa, particularmente en contextos con limitaciones de recursos docentes donde la relación estudiante-profesor es desfavorable.

- **Motivación intrínseca y compromiso**

Se observó un aumento robusto en la motivación intrínseca ($t(29) = 9.45, p < .001, d = 1.99$) y una reducción significativa de la amotivación ($t(29) = -6.78, p < .001, d = -1.38$), reflejando mayor engagement y persistencia. La motivación extrínseca no mostró cambios significativos, lo que sugiere que la intervención favoreció principalmente la motivación

interna. En consonancia con De La Cruz Apari y Mere Jiménez (2023), la personalización y la retroalimentación inmediata de la IA incrementan la motivación interna y reducen la deserción. A diferencia de contextos donde la motivación extrínseca era más relevante, en este estudio la motivación intrínseca fue predominante, posiblemente por el enfoque centrado en el estudiante.

Flores y Suyon (2022) reportaron que el uso de chatbots educativos se asoció con niveles favorables de motivación y satisfacción en docentes peruanos. De manera complementaria, (2023) evidenció un incremento significativo en la autonomía estudiantil tras la implementación de chatbots con inteligencia artificial, observándose que los participantes alcanzaron niveles altos de autonomía al finalizar la intervención. En conjunto, estos antecedentes resaltan la relevancia de la contextualización y la pertinencia cultural en el diseño de intervenciones educativas basadas en inteligencia artificial en el contexto peruano.

La Teoría de la Autodeterminación Ryan y Deci (2020) sostiene que la satisfacción de las necesidades de autonomía, competencia y relación incrementa la motivación interna. La IA, al ofrecer personalización y retroalimentación inmediata, favorece estos factores, confirmando y ampliando las predicciones teóricas. Teóricamente, estos resultados aportan evidencia sobre el potencial de la IA para transformar la motivación en contextos educativos. Prácticamente, sugieren que las instituciones deben priorizar la personalización y la adaptación cultural en el diseño de plataformas de IA para maximizar el engagement y la retención estudiantil.

- **Percepción tecnológica (utilidad y facilidad de uso)**

La percepción de utilidad y facilidad de uso de la IA educativa mejoró notablemente ($t(29) = 7.12, p < .001, d = 1.43$; $t(29) = 9.87, p < .001, d = 2.04$). Esto indica que los estudiantes no solo reconocen el valor práctico de la IA, sino que también se sienten cómodos e incentivados a utilizar estas herramientas como parte integral de su proceso formativo.

En consonancia con Cueva-Eguizábal (2023) más del 80% de los estudiantes en estudios internacionales perciben las plataformas tecnológicas como herramientas que mejoran su formación profesional, competencias digitales y trabajo colaborativo. A diferencia de

contextos donde la resistencia al cambio tecnológico es alta, en este estudio la percepción positiva fue predominante, posiblemente debido a la integración cuidadosa y el soporte institucional. A nivel nacional, estudios en educación superior tecnológica sugieren que la incorporación de tecnologías educativas innovadoras se asocia con mayores niveles de participación y compromiso estudiantil, lo que puede contribuir al desarrollo de competencias y al logro académico.

El Modelo de Aceptación Tecnológica (TAM) de Davis (1989) sostiene que la percepción positiva de la utilidad y facilidad de uso predice la adopción y el uso sostenido de tecnologías educativas. Los resultados de este estudio confirman y amplían este modelo, mostrando que la integración de IA no solo facilita el acceso a información y recursos, sino que también promueve la equidad y la inclusión educativa, siempre que se aborden los desafíos de acceso y formación docente.

Teóricamente, estos resultados aportan evidencia sobre la validez del TAM en contextos de IA educativa. Prácticamente, sugieren que las instituciones deben invertir en formación docente, políticas de acceso equitativo y lineamientos éticos claros para maximizar el impacto positivo de la IA en la educación superior. En conjunto, los resultados de cada dimensión confirman que la IA educativa, cuando es implementada con criterios de personalización, adaptabilidad y pertinencia cultural, puede transformar el aprendizaje en contextos de educación superior tecnológica. La integración de teorías del conocimiento como el constructivismo, la autodeterminación y el TAM permite comprender y potenciar estos efectos, proyectando un impacto positivo y sostenible en la formación de profesionales en programación en el Perú y América Latina.

5.3 Limitaciones del Estudio

5.3.1 Limitaciones Metodológicas

El diseño pre-experimental empleado, aunque pragmático para el contexto educativo, presenta limitaciones inherentes en la validez interna. La ausencia de grupo control imposibilita la atribución causal definitiva de los efectos observados exclusivamente a la intervención con IA. Factores como el efecto Hawthorne, donde los participantes mejoran su desempeño simplemente por saber que están siendo observados, podrían haber inflado los resultados. Aunque intentamos controlar esto mediante mediciones discretas y naturalizadas, no podemos descartarlo completamente.

La historia y maduración representan amenazas adicionales a la validez. Durante las 16 semanas de intervención, los estudiantes estuvieron expuestos a otras experiencias educativas y personales que podrían haber influido en su desarrollo. Particularmente, el curso paralelo de Estructuras de Datos podría haber generado efectos de transferencia no controlados. Futuros estudios deberían emplear diseños experimentales verdaderos con asignación aleatoria, aunque reconocemos las dificultades éticas y prácticas de privar a un grupo control de una intervención potencialmente beneficiosa.

5.3.2 Limitaciones en la Generalización

El contexto específico del instituto tecnológico en Lima Centro representa otra limitación. Las características socioeconómicas, la infraestructura tecnológica disponible y la cultura institucional particular podrían no ser representativas de otros contextos educativos peruanos. La conectividad confiable y la disponibilidad de equipos actualizados en la institución donde se desarrolló el estudio contrastan con la realidad de muchos institutos de educación superior en el país, donde aún persisten brechas en infraestructura digital y en el acceso a tecnologías emergentes como la inteligencia artificial. En el contexto de la educación superior peruana, diversos estudios han señalado la importancia de considerar las

características propias de cada institución, incluyendo la disponibilidad de recursos tecnológicos y el perfil del estudiantado, dado que estos factores influyen en la implementación y efectividad de intervenciones educativas mediadas por tecnologías digitales (Díaz Ancco et al., 2024).

5.3.3 Limitaciones Temporales

El período de intervención de 14 semanas, aunque sustancial, podría ser insuficiente para evaluar efectos a largo plazo. La novedad de la tecnología podría haber generado un entusiasmo inicial que se desvanezca con el tiempo. Aunque planificamos mediciones de seguimiento a 3 y 6 meses, estos datos aún no están disponibles para análisis. La literatura sobre adopción tecnológica sugiere que muchas innovaciones educativas muestran efectos decrecientes después del período de "luna de miel" inicial.

5.4 Interpretación de Hallazgos No Esperados

5.4.1 El Fenómeno de Colaboración Emergente

Un hallazgo no anticipado fue el surgimiento espontáneo de colaboración entre estudiantes mediante el grupo de WhatsApp del curso, donde compartían sus interacciones con los asistentes de IA (ChatGPT, Claude, Perplexity) y discutían las respuestas recibidas de diferentes plataformas. Contrario a la preocupación inicial de que la IA podría aislar a los estudiantes al proporcionar asistencia individualizada que eliminaría la necesidad de consultar con compañeros, observamos que el grupo de WhatsApp se convirtió en un espacio activo de intercambio de estrategias efectivas de formulación de prompts, comparación de respuestas entre las tres plataformas, y discusión sobre cuál explicación era más clara o útil para diferentes tipos de problemas. Este fenómeno emergente sugiere que los asistentes de IA pueden catalizar, más que reemplazar, la colaboración humana al proporcionar un objeto compartido de análisis y discusión que enriquece el aprendizaje social.

El análisis cualitativo del cuestionario de experiencia final reveló que los estudiantes utilizaban las respuestas de los asistentes de IA como punto de partida para discusiones más profundas y construcción colaborativa de conocimiento. Comentarios representativos incluyeron: "ChatGPT me dio una solución rápida, pero Claude me explicó mejor por qué funcionaba, así que compartí ambas en el grupo", "Cuando tenía dudas sobre una respuesta de la IA, preguntaba en WhatsApp y entre varios comparábamos qué había dicho cada plataforma", y "Descubrimos que Perplexity era mejor para encontrar documentación oficial, así que nos pasábamos los links que encontrábamos".

Estos patrones indican un nivel de pensamiento crítico sofisticado donde los estudiantes no aceptaban pasivamente las respuestas generadas por IA, sino que las sometían a escrutinio colaborativo, verificación cruzada entre múltiples fuentes, y discusión reflexiva sobre la calidad y utilidad de diferentes explicaciones. Esta triangulación multi-herramienta y multi-perspectiva (combinando IA y pares humanos) generó un scaffolding más robusto para la construcción de comprensión profunda de programación Java que cualquiera de las herramientas o interacciones podría haber proporcionado de manera aislada.

5.4.2 Resistencia Selectiva a la Automatización

Inesperadamente, encontramos que el 23% de los estudiantes mostraron resistencia selectiva a ciertas funcionalidades de IA, particularmente a la generación automática de código completo. Estos estudiantes, predominantemente aquellos con mejor rendimiento inicial, preferían hints graduales sobre soluciones completas, expresando que "querían aprender, no que les hagan la tarea". Este hallazgo desafía la narrativa de que los estudiantes siempre buscarán el camino de menor resistencia y sugiere que, cuando están intrínsecamente motivados, valoran el proceso de aprendizaje sobre la eficiencia.

En conjunto, los resultados obtenidos permiten aceptar la hipótesis general y las hipótesis específicas planteadas, evidenciando que la implementación de herramientas de

inteligencia artificial impacta positivamente en la efectividad de la enseñanza de la programación. La coherencia entre los resultados empíricos y las hipótesis formuladas confirma que la inteligencia artificial no solo mejora el rendimiento académico, sino que también fortalece el aprendizaje autónomo, la motivación y la aceptación tecnológica de los estudiantes. De este modo, la investigación aporta evidencia empírica contextualizada que respalda el uso pedagógico de asistentes conversacionales basados en IA en instituciones de educación superior tecnológica.

5.5 Implicaciones Prácticas

5.5.1 Para la Práctica Docente

Los resultados sugieren una reconceptualización del rol docente en ambientes enriquecidos con IA. Más que ser reemplazado, el docente evoluciona hacia un facilitador de experiencias de aprendizaje de orden superior. Durante la intervención, el docente colaborador reportó tener "más tiempo para discusiones conceptuales profundas" al ser liberado de tareas repetitivas como corrección de sintaxis básica. Esta transición requiere desarrollo profesional específico, no solo en competencias técnicas sino en nuevas pedagogías que aprovechen las affordances de la IA.

La evidencia sugiere que la preparación docente debe incluir: comprensión de las capacidades y limitaciones de los sistemas de IA, habilidades para interpretar analytics de aprendizaje generados por estos sistemas, competencias para diseñar actividades que complementen la instrucción automatizada, y capacidad para intervenir cuando la IA alcanza sus límites. Instituciones que planeen implementar IA educativa deben presupuestar significativa inversión en desarrollo profesional docente.

5.5.2 Para el Diseño Curricular

La efectividad diferencial de la IA en distintos tipos de contenido sugiere la necesidad de rediseñar currículos para optimizar la sinergia humano-IA. Contenidos procedimentales y de práctica repetitiva pueden delegarse más extensamente a sistemas automatizados, liberando tiempo presencial para actividades que requieren creatividad, pensamiento crítico, y colaboración. Un currículo "AI-aware" distribuiría estratégicamente los objetivos de aprendizaje entre modalidades presenciales y asistidas por IA.

La secuenciación curricular también requiere reconsideración. La capacidad de la IA para proporcionar instrucción personalizada 24/7 permite modelos de progresión basados en competencias más que en tiempo. Estudiantes que dominan rápidamente conceptos básicos podrían avanzar aceleradamente, mientras que aquellos que requieren más práctica recibirían soporte adicional sin retrasar a sus pares. Esto desafía el modelo tradicional de cohortes sincrónicas y sugiere movimiento hacia modelos más flexibles y personalizados.

5.5.3 Para las Políticas Institucionales

La implementación exitosa de IA educativa requiere políticas institucionales que aborden múltiples dimensiones. Primero, políticas de infraestructura tecnológica que garanticen acceso equitativo, incluyendo provisión de equipos y conectividad para estudiantes vulnerables. Nuestro hallazgo de que el 30% del aprendizaje ocurrió fuera del campus subraya la importancia del acceso remoto confiable.

Segundo, políticas de datos y privacidad que balanceen los beneficios de la personalización basada en datos con la protección de información estudiantil sensible. Durante nuestro estudio, recolectamos más de 2 millones de puntos de datos sobre comportamiento de aprendizaje, planteando preguntas éticas sobre propiedad, uso, y retención de estos datos. Instituciones deben desarrollar frameworks éticos claros antes de implementaciones a gran escala.

Tercero, políticas de evaluación y acreditación que reconozcan nuevas formas de demostrar competencia. Si la IA permite trayectorias de aprendizaje personalizadas, los sistemas de evaluación estandarizados pueden volverse obsoletos. Necesitamos métricas que capturen no solo el conocimiento final sino la capacidad de aprender continuamente con soporte de IA.

5.6 Direcciones Futuras de Investigación

5.6.1 Estudios Longitudinales

Se requieren investigaciones longitudinales que examinen la sostenibilidad de los efectos observados. ¿Persisten las ganancias en rendimiento y motivación después de remover el soporte de IA? ¿Desarrollan los estudiantes habilidades de autorregulación transferibles a otros dominios? Estudios de seguimiento de 2-3 años podrían revelar si la IA educativa genera cambios duraderos en trayectorias de aprendizaje o simplemente efectos temporales mientras la tecnología está presente.

5.6.2 Investigación sobre Equidad

Críticamente, futura investigación debe examinar cómo la IA educativa afecta brechas de equidad existentes. ¿Beneficia desproporcionadamente a estudiantes con mayor capital cultural digital? ¿O puede servir como nivelador, proporcionando a estudiantes desfavorecidos acceso a soporte educativo de alta calidad? Nuestros datos preliminares sugieren que estudiantes con menor rendimiento inicial mostraron mayores ganancias relativas, pero se requiere investigación específicamente diseñada para examinar efectos diferenciales por nivel socioeconómico, género, y origen étnico.

5.6.3 Estudios Comparativos de Modelos de IA

La rápida evolución de modelos de lenguaje grande (LLMs) presenta oportunidades para investigación comparativa. ¿Cómo se compara GPT-4 con Claude, Gemini, o modelos open-source para educación en programación? ¿Existen trade-offs entre sofisticación del

modelo y explicabilidad pedagógica? Estudios que comparen sistemáticamente diferentes arquitecturas de IA en contextos educativos controlados informarían decisiones de adopción institucional.

5.7 Síntesis Final

La presente investigación proporciona evidencia robusta de que la implementación cuidadosa de herramientas de inteligencia artificial puede generar mejoras sustanciales en el Aprendizaje de Programación Java. Los efectos observados trascienden mejoras cuantitativas en rendimiento, catalizando transformaciones cualitativas en cómo los estudiantes se relacionan con el aprendizaje de programación. La combinación de personalización adaptativa, retroalimentación inmediata, y disponibilidad continua crea un ambiente de aprendizaje cualitativamente diferente del modelo tradicional.

Sin embargo, estos resultados prometedores deben interpretarse con cautela apropiada. Las limitaciones metodológicas, particularmente la falta de grupo control y el muestreo por conveniencia, impiden inferencias causales definitivas. Además, el contexto específico del estudio en un instituto de educación superior tecnológico urbano con infraestructura adecuada, puede no ser representativo.

A pesar de estas limitaciones, los hallazgos tienen implicaciones importantes para la teoría y práctica educativa. Teóricamente, desafían concepciones tradicionales sobre la necesidad de interacción humana exclusiva para el aprendizaje efectivo. Prácticamente, sugieren que la IA educativa, apropiadamente implementada, puede abordar desafíos persistentes en la enseñanza de programación como la falta de retroalimentación inmediata y la imposibilidad de personalización en clases numerosas.

El camino hacia adelante requiere investigación continua, experimentación cuidadosa, y reflexión crítica sobre las implicaciones éticas y sociales de la IA en educación. La tecnología por sí sola no es una panacea; su efectividad depende crucialmente del diseño pedagógico, la

preparación docente, el soporte institucional, y la consideración cuidadosa del contexto local. A medida que la IA continúa evolucionando rápidamente, educadores e investigadores deben colaborar para asegurar que estas poderosas herramientas se utilicen para ampliar, no restringir, las oportunidades de aprendizaje para todos los estudiantes.

VI. CONCLUSIONES

- 6.1** Se concluye que la implementación de herramientas de inteligencia artificial incrementó significativamente la efectividad de la enseñanza del lenguaje de programación en los estudiantes del III ciclo del programa de Computación e Informática del instituto tecnológico estudiado. Los resultados obtenidos permitieron aceptar la hipótesis general, evidenciando mejoras sustanciales en el rendimiento académico y en variables asociadas al proceso de aprendizaje, lo que confirma la pertinencia del uso pedagógico de la inteligencia artificial en contextos de educación superior tecnológica. Asimismo, se determinó que el uso integrado de asistentes conversacionales basados en IA constituye una alternativa pedagógica efectiva frente al modelo tradicional de enseñanza de la programación, especialmente en contextos caracterizados por limitaciones en la retroalimentación oportuna y el acompañamiento personalizado.
- 6.2** Se concluye que la implementación de herramientas de inteligencia artificial elevó significativamente el rendimiento académico de los estudiantes del III ciclo, evidenciándose mejoras tanto en el conocimiento teórico como en las habilidades prácticas de programación. Los resultados confirman que la intervención con IA favoreció el desarrollo de competencias fundamentales vinculadas al aprendizaje del lenguaje de programación, permitiendo aceptar la hipótesis específica correspondiente.
- 6.3** Se concluye que el uso de herramientas de inteligencia artificial incrementó significativamente la motivación intrínseca y el compromiso conductual de los estudiantes. La intervención contribuyó a una mayor disposición hacia el aprendizaje autónomo y a la reducción de la amotivación académica, favoreciendo la persistencia en el desarrollo de actividades de programación, lo que permitió aceptar la hipótesis específica planteada.

- 6.4** Se concluye que la implementación de herramientas de inteligencia artificial mejoró significativamente la percepción de utilidad y facilidad de uso del aprendizaje de programación. Los estudiantes manifestaron una alta aceptación tecnológica y una clara intención de continuar utilizando herramientas de IA como apoyo en su formación académica, lo que confirma la hipótesis específica relacionada con la aceptación tecnológica.
- 6.5** El presente estudio presenta limitaciones que deben ser consideradas en la interpretación de los resultados. El diseño preexperimental empleado limita la posibilidad de establecer inferencias causales definitivas. Asimismo, el tamaño muestral reducido y el muestreo por conveniencia restringen la generalización de los hallazgos a otras poblaciones. Finalmente, el período de intervención relativamente corto no permitió evaluar la sostenibilidad de los efectos observados a largo plazo.
- 6.6** En conclusión, la presente investigación aporta evidencia empírica que demuestra que la inteligencia artificial constituye una herramienta pedagógica eficaz para mejorar la enseñanza y el aprendizaje de la programación en educación superior tecnológica. No obstante, su impacto positivo depende de una adecuada integración pedagógica, del compromiso institucional y de la preparación docente, aspectos que deben ser considerados para garantizar implementaciones responsables, equitativas y sostenibles.

VII. RECOMENDACIONES

Las siguientes recomendaciones se derivan de los resultados obtenidos en la presente investigación y se orientan a fortalecer la implementación responsable y efectiva de herramientas de inteligencia artificial en la enseñanza de la programación en instituciones de educación superior tecnológica.

7.1 Se recomienda que las instituciones de educación superior tecnológica establezcan lineamientos institucionales para la integración progresiva de herramientas de inteligencia artificial en los procesos de enseñanza-aprendizaje. Dichos lineamientos deben contemplar la creación de instancias de coordinación académica y tecnológica que supervisen la adopción de estas herramientas, asegurando su alineación con los objetivos institucionales y las necesidades pedagógicas del programa de estudios. Asimismo, se sugiere incorporar la innovación educativa basada en inteligencia artificial dentro de los planes estratégicos institucionales, promoviendo una visión de mejora continua en la calidad de la enseñanza de la programación.

7.2 Se recomienda que las instituciones evalúen y fortalezcan progresivamente su infraestructura tecnológica, garantizando condiciones adecuadas de conectividad y acceso a equipos informáticos que permitan el uso eficiente de herramientas de inteligencia artificial. De igual forma, se sugiere implementar mecanismos de apoyo institucional para estudiantes que presenten limitaciones de acceso tecnológico, con el fin de asegurar condiciones mínimas de equidad educativa.

7.3 Se recomienda implementar estrategias de gestión del cambio orientadas a sensibilizar a la comunidad educativa sobre el uso pedagógico de la inteligencia artificial. Estas estrategias deben considerar espacios de diálogo, acompañamiento y retroalimentación continua, con el objetivo de facilitar la adopción gradual de las herramientas y reducir posibles

resistencias asociadas al cambio tecnológico.

- 7.4** Se recomienda que los docentes del área de programación fortalezcan sus competencias digitales y pedagógicas relacionadas con el uso de inteligencia artificial educativa. La capacitación continua en el uso de estas herramientas permitirá diseñar experiencias de aprendizaje más personalizadas y efectivas, alineadas con las demandas actuales de la formación tecnológica.
- 7.5** Se sugiere que los docentes incorporen progresivamente herramientas de inteligencia artificial como apoyo complementario en el desarrollo de los cursos de programación. Esto implica rediseñar actividades académicas que promuevan el aprendizaje activo, el pensamiento crítico y la resolución de problemas, manteniendo al docente como mediador y facilitador del proceso educativo.
- 7.6** Se recomienda que los docentes utilicen la información generada por las plataformas de inteligencia artificial para monitorear el progreso de los estudiantes, identificar oportunamente dificultades de aprendizaje y aplicar estrategias de apoyo académico personalizadas.
- 7.7** Se recomienda que los estudiantes asuman un rol activo en su proceso de aprendizaje, utilizando las herramientas de inteligencia artificial como apoyo para la práctica constante y la autoevaluación. El uso responsable de estas herramientas puede fortalecer el aprendizaje autónomo y la consolidación de competencias en programación.
- 7.8** Se recomienda que los estudiantes desarrollen criterios éticos en el uso de herramientas de inteligencia artificial, utilizándolas como apoyo al aprendizaje y no como sustituto del esfuerzo cognitivo propio. Asimismo, se sugiere fomentar la reflexión sobre el proceso de resolución de problemas y la documentación del aprendizaje.
- 7.9** Se sugiere promover el trabajo colaborativo entre estudiantes, utilizando las herramientas

de inteligencia artificial como recurso compartido para el análisis de problemas, el intercambio de ideas y la construcción colectiva del conocimiento.

- 7.10** Se recomienda que futuras investigaciones profundicen el estudio del impacto de la inteligencia artificial en la enseñanza de la programación mediante diseños experimentales más robustos, muestras ampliadas y estudios longitudinales. Asimismo, resulta pertinente explorar la transferencia de estas herramientas a otras áreas técnicas y analizar su impacto en diferentes contextos socioeducativos del país.
- 7.11** La implementación efectiva de herramientas de inteligencia artificial en educación superior tecnológica requiere un enfoque integral que articule políticas institucionales, prácticas pedagógicas innovadoras y un uso responsable por parte de docentes y estudiantes. Las recomendaciones presentadas buscan contribuir a una adopción reflexiva y sostenible de la inteligencia artificial, orientada a mejorar la calidad del aprendizaje y la formación de profesionales competentes en el área de programación.

VIII. REFERENCIAS

- Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI and Ethics*, 2(3), 431–440. <https://doi.org/10.1007/s43681-021-00096-7>
- Al-Emran, M., & Granić, A. (Eds.). (2021). *Technology acceptance in education: Research and innovation*. Springer. <https://doi.org/10.1007/978-3-030-64986-8>
- Alshahrani, A. (2023). The impact of ChatGPT on blended learning: Current trends and future research directions. *International Journal of Data and Network Science*, 7(4), 2029–2040. <https://doi.org/10.5267/j.ijdns.2023.6.010>
- American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. <https://www.apa.org/science/programs/testing/standards>
- Angeli, C., & Valanides, N. (2022). Developing pedagogical content knowledge for computational thinking education. *Educational Technology Research and Development*, 70(2), 535–552. <https://doi.org/10.1007/s11423-022-10089-w>
- Becker, B. A., & Quille, K. (2021). 50 years of CS1 at SIGCSE: A review of the literature. En *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 701–707). ACM. <https://doi.org/10.1145/3408877.3432595>
- Becker, B. A., Denny, P., Pettit, R., Bouchard, D., Bouvier, D. J., Harrington, B., Kamil, A., Karkare, A., McDonald, C., Osera, P. M., Pearce, J. L., & Prather, J. (2023). Programming is hard — or at least it used to be: Educational opportunities and challenges of AI code generation. En *Proceedings of the 54th ACM Technical Symposium on Computer Science Education* (Vol. 1, pp. 500–506). ACM. <https://doi.org/10.1145/3545945.3569759>
- Bond, M., Khosravi, H., De Laat, M., Bergdahl, N., Negrea, V., Oxley, E., Pham, P., Siemens,

- G., & Gašević, D. (2024). A meta systematic review of artificial intelligence in higher education: a call for increased ethics, collaboration, and rigour. *International Journal of Educational Technology in Higher Education*, 21, Artículo 4. <https://doi.org/10.1186/s41239-023-00436-z>
- Capili, B., & Anastasi, J. K. (2024). An introduction to the quasi-experimental design (nonrandomized design). *AJN, American Journal of Nursing*, 124(11), 50–52. <https://doi.org/10.1097/01.NAJ.0001081740.74815.20>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., ... Zaremba, W. (2021). *Evaluating large language models trained on code* (arXiv:2107.03374) [Preprint]. arXiv. <https://arxiv.org/abs/2107.03374>
- Chen, X., Xie, H., Zou, D., & Hwang, G. J. (2020). Application and theory gaps during the rise of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 1, 100002. <https://doi.org/10.1016/j.caeai.2020.100002>
- Cito, J., Dillig, I., Kim, S., Murali, V., & Chandra, S. (2024). *Towards understanding the effectiveness of large language models on directed test input generation* (arXiv:2401.04093) [Preprint]. arXiv. <https://arxiv.org/abs/2401.04093>
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2.^a ed.). Lawrence Erlbaum Associates.
- Creswell, J. W., & Creswell, J. D. (2022). *Research design: Qualitative, quantitative, and mixed methods approaches* (6.^a ed.). SAGE Publications.
- Crow, T., Luxton-Reilly, A., & Wuensche, B. (2018). Intelligent tutoring systems for programming education: A systematic review. En *Proceedings of the 20th Australasian Computing Education Conference* (pp. 53–62). ACM.

<https://doi.org/10.1145/3160489.3160492>

Cueto, S., Miranda, A., León, J., & Vásquez, M. C. (2022). *Education and learning in Peru: Setting a research agenda*. Grupo de Análisis para el Desarrollo (GRADE).

<https://grade.org.pe/wp-content/uploads/GRADEPeruPendienteCueto.pdf>

Cueva-Eguizábal, M. G. (2023). *ChatGPT en el desempeño académico de alumnos de ingeniería de sistemas*. [Tesis de pregrado, Universidad César Vallejo]. Alicia CONCYTEC.

https://alicia.concytec.gob.pe/vufind/Record/UCVV_d7ab368b7fea3f93a42da5e70b393e34

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>

De La Cruz Apari, J. K., & Mere Jiménez, A. W. (2023). *Implementación de un chatbot con inteligencia artificial para mejorar el aprendizaje de algoritmos en los estudiantes de una universidad pública* [Tesis doctoral, Universidad Autónoma de Ica]. Repositorio Institucional Universidad Autónoma de Ica. <https://hdl.handle.net/20.500.14441/2682>

Denny, P., Kumar, A., & Giacaman, N. (2023). Conversing with Copilot: Exploring prompt engineering for solving CS1 problems using natural language. En *Proceedings of the 54th ACM Technical Symposium on Computer Science Education* (Vol. 1, pp. 1136–1142). ACM. <https://doi.org/10.1145/3545945.3569823>

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. *Communications of the ACM*, 67(2), 56–67.

<https://doi.org/10.1145/3624720>

Deveci Topal, A., Dilek Eren, C., & Kolburan Gecer, A. (2021). Chatbot application in a 5th

- grade science course. *Education and Information Technologies*, 26(5), 6241–6265.
<https://doi.org/10.1007/s10639-021-10627-8>
- Díaz-Ancco, F., Rodríguez Gonzales, K., & Estrada Chacón, L. H. (2024). El impacto de la inteligencia artificial en la formación de estudiantes de educación superior. *Yachay – Revista Científico Cultural*, 13(1), 44–61. <https://doi.org/10.36881/yachay.v13i1.782>
- Efan, E., Harianja, J. K., & Maranata, F. (2024). Exploring the effects of AI-based tutoring systems on programming skills acquisition. *Education and Information Technologies*, 29(2), 1567–1589. <https://doi.org/10.1007/s10639-023-11987-0>
- Efan, E., Maranata, F., & Harianja, J. K. (2023). The impact of artificial intelligence on student engagement in programming courses. *Journal of Educational Technology & Online Learning*, 6(3), 789–802. <https://doi.org/10.31681/jetol.1287365>
- Espinoza-Freire, E. E. (2022). *El problema, el objetivo, la hipótesis y las variables de la investigación*. Editorial Exced.
<https://dialnet.unirioja.es/servlet/articulo?codigo=9071399>
- Espinoza-Quiquia, E. D. (2023). *Uso de estrategias de gamificación y el desarrollo de competencias digitales en estudiantes de un instituto público de Lima, periodo 2023* [Tesis de maestría, Universidad César Vallejo]. Repositorio institucional UCV.
<https://hdl.handle.net/20.500.12692/124579>
- Essel, H. B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E. E., & Baah, P. K. (2022). The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education. *International Journal of Educational Technology in Higher Education*, 19(1), Artículo 57. <https://doi.org/10.1186/s41239-022-00362-6>
- Feng, S., Magana, A. J., & Kao, D. (2021). A systematic review of literature on the effectiveness of intelligent tutoring systems in STEM. En *2021 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

<https://doi.org/10.1109/FIE49875.2021.9637240>

Field, A. (2022). *Discovering statistics using IBM SPSS statistics* (6.^a ed.). SAGE Publications.

Finnie-Ansley, J., Denny, P., Becker, B. A., Prather, J., & Luxton-Reilly, A. (2022). My AI wants to know if this will be on the exam: Testing OpenAI's Codex on CS2 programming exercises. En *Proceedings of the 24th Australasian Computing Education Conference* (pp. 117–126). ACM. <https://doi.org/10.1145/3511861.3511863>

Flores, J., & Suyon, M. (2022). *Chatbot para el aprendizaje de plataformas educativas virtuales* [Tesis de pregrado, Universidad César Vallejo]. Repositorio institucional UCV. <https://hdl.handle.net/20.500.12692/112534>

Flores-Vivar, J. M., & García-Peñalvo, F. J. (2023). Reflexiones sobre la ética, potencialidades y retos de la inteligencia artificial en el marco de la educación de calidad (ODS4). *Comunicar*, 31(74), 37–47. <https://doi.org/10.3916/C74-2023-03>

Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2330–2339. <https://doi.org/10.1145/3394486.3403282>

Guo, S., Zhang, Z., & Chen, T. (2024). Automated generation of programming exercises using large language models. *Computers & Education: Artificial Intelligence*, 5, 100179. <https://doi.org/10.1016/j.caeai.2024.100179>

Heo, H., Bonk, C. J., & Doo, M. Y. (2022). Influences of depression, self-efficacy, and resource management on learning engagement, motivation, and achievement in a college flipped learning course. *The Internet and Higher Education*, 53, 100855. <https://doi.org/10.1016/j.iheduc.2021.100855>

Hodges, C. B., Moore, S., Lockee, B., Trust, T., & Bond, M. A. (27 de marzo de 2020). The difference between emergency remote teaching and online learning. *EDUCAUSE Review*. <https://er.educause.edu/articles/2020/3/the-difference-between-emergency->

[remote-teaching-and-online-learning](#)

Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning*. Center for Curriculum Redesign.

Holstein, K., & McLaren, B. M. (2021). Co-designing a real-time classroom orchestration tool to balance teacher agency and student privacy. En I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, & V. Dimitrova (Eds.), *Artificial intelligence in education: 22nd International Conference* (pp. 157–171). Springer. https://doi.org/10.1007/978-3-030-78292-4_13

Hwang, G. J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles and research issues of artificial intelligence in education. *Computers and Education: Artificial Intelligence, 1*, 100001. <https://doi.org/10.1016/j.caeai.2020.100001>

Informatics Europe. (2023). *European informatics research landscape*.

<https://www.informatics-europe.org/data/2023-informatics-research-landscape.pdf>

Instituto Internacional de Planeamiento de la Educación de la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [IIPE UNESCO]. (2022). *Ciencias de la computación en los sistemas educativos de América Latina*. <https://www.buenosaires.iiep.unesco.org/es/publicaciones/ciencias-de-la-computacion-en-los-sistemas-educativos-de-america-latina>

International Organization for Standardization. (2023). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model* (ISO/IEC Standard No. 25010:2023). <https://www.iso.org/standard/78176.html>

Jahagirdar, A. (2026). Type systems in programming languages: A performance, safety, and development efficiency analysis of static (Java/C++) vs. dynamic (Python) typing paradigms. *International Journal of Science and Research (IJSR)*, 15(1), 941–943.

<https://dx.doi.org/10.21275/SR26113212016>

Kesler, A., Shamir-Inbal, T., & Blau, I. (2022). Active learning by visual programming: Pedagogical perspectives of instructivist and constructivist code teachers and their implications on actual teaching strategies and students' programming artifacts. *Journal of Educational Computing Research*, 60(4), 28–55.

<https://doi.org/10.1177/07356331211017793>

Keuning, H., Heeren, B., & Jeurig, J. (2023). A tutoring system to learn code refactoring. Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, 562–568. <https://doi.org/10.1145/3408877.3432526>

Keuning, H., Jeurig, J., & Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education*, 19(1), Artículo 3. <https://doi.org/10.1145/3231711>

Khechine, H., & Augier, M. (2020). Adoption of a web-based learning system in higher education. *Journal of Information Technology Education: Research*, 19, 433–452. <https://doi.org/10.28945/4563>

Kochmar, E., Vu, D. D., Belfer, R., Guber, V., Serban, I. V., Pineau, J., & Muresan, S. (2020). Automated personalized feedback improves learning gains in an intelligent tutoring system. En Proceedings of the 21st International Conference on Artificial Intelligence in Education (pp. 140–146). Springer. https://doi.org/10.1007/978-3-030-52240-7_26

Kuhail, M. A., Alturki, N., Alramlawi, S., & Alhejori, K. (2022). Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1), 973–1018. <https://doi.org/10.1007/s10639-022-11177-3>

Létourneau, A., Deslandes Martineau, M., Charland, P., Karran, A. J., Boasen, J., & Léger, P. M. (2025). A systematic review of AI-driven intelligent tutoring systems (ITS) in K-12 education. *npj Science of Learning*, 10, 29. <https://doi.org/10.1038/s41539-025-00320->

7

- 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), Artículo 41. <https://doi.org/10.1186/s40561-023-00260-y>
- Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, 30, 883–908. <https://doi.org/10.1007/s11191-021-00202-5>
- López, A. (2024). Paradigma de la programación orientada a objetos (POO). *Revista Digital de Tecnologías Informáticas y Sistemas*, 8(1), 147–153. <https://doi.org/10.61530/redtis.vol8.n1.2024.212>
- Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Lao, L., Licoverett, P., Morrison, B. B., & Whalley, J. L. (2018). Introductory programming: A systematic literature review. En *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 55–106). ACM. <https://doi.org/10.1145/3293881.3295779>
- Maciel, J. C. (2021). Artificial intelligence and teachers' professional development: New challenges for the teaching profession. *Education and Information Technologies*, 26, 703–717. <https://doi.org/10.1007/s10639-020-10271-9>
- Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the National Academy of Sciences*, 118(48), e2100030118. <https://doi.org/10.1073/pnas.2100030118>
- McLaren, B. M. (2021). Intelligent tutoring systems: Successes, challenges, and the road ahead. *International Journal of Artificial Intelligence in Education*, 31(2), 339–351. <https://doi.org/10.1007/s40593-020-00228-0>
- Mills, C., Denny, P., Appiah, O., Bosch, N., Hutt, S., & Gašević, D. (2025). Generative AI and

- the future of assessment in programming. *Communications of the ACM*, 68(1), 38–41.
<https://doi.org/10.1145/3638249>
- Ministerio de Educación del Perú. (2023). *Estadística de la calidad educativa (ESCALE): Educación superior tecnológica*. <https://escale.minedu.gob.pe>
- Ministerio de la Producción del Perú. (2019). *Plan Nacional de Competitividad y Productividad 2019-2030*. <https://www.produce.gob.pe/index.php/rmpi/plan-nacional>
- Moorhouse, B. L., Yeo, M. A., & Wan, Y. (2023). Generative AI tools and assessment: Guidelines of the world’s top-ranking universities. *Computers and Education: Artificial Intelligence*, 5, 100151. <https://doi.org/10.1016/j.caeai.2023.100151>
- Okonkwo, C. W., & Ade-Ibijola, A. (2021). Python-bot: A chatbot for teaching Python programming. *Engineering Letters*, 29(1), 25–34.
https://www.engineeringletters.com/issues_v29/issue_1/EL_29_1_03.pdf
- Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura [UNESCO]. (2021). *AI and education: Guidance for policy-makers*.
<https://unesdoc.unesco.org/ark:/48223/pf0000376709>
- Ouyang, F., & Jiao, P. (2021). Artificial intelligence in education: The three paradigms. *Computers and Education: Artificial Intelligence*, 2, 100014.
<https://doi.org/10.1016/j.caeai.2021.100014>
- Patino, C. M., & Ferreira, J. C. (2018). Inclusion and exclusion criteria in research studies: Definitions and why they matter. *Jornal Brasileiro de Pneumologia*, 44(2), 84.
<https://doi.org/10.1590/S1806-37562018000000088>
- Pellas, N. (2023). ChatGPT and Google Bard’s impact on programming education: A systematic literature review. *Computers and Education: Artificial Intelligence*, 5, 100176. <https://doi.org/10.1016/j.caeai.2023.100176>
- Piaget, J. (1972). *Psychology and epistemology: Towards a theory of knowledge* (B. Walsh,

- Trad.). Penguin. (Obra original publicada en 1970)
- Python Software Foundation. (2001, actualizado 2013). *PEP 8 — Style guide for Python code*.
<https://peps.python.org/pep-0008/>
- Prather, J., Becker, B. A., Craig, M., Denny, P., Loksa, D., & Margulieux, L. (2020). What do we think we think we are doing? Metacognition and self-regulation in programming. En *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 2–13). ACM. <https://doi.org/10.1145/3372782.3406263>
- Prather, J., Pettit, R., McMurry, K., Peters, A., Homer, J., & Cohen, M. (2018). Metacognitive difficulties faced by novice programmers in automated assessment tools. En *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 41–50). ACM. <https://doi.org/10.1145/3230977.3230981>
- Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Powell, G., & Luxton-Reilly, A. (2022). What makes a good explanation? A study of students interpreting visualizations. En *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education* (Vol. 1, pp. 922–928). ACM. <https://doi.org/10.1145/3478431.3499409>
- Ryan, R. M., & Deci, E. L. (2020). Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary Educational Psychology*, 61, 101860. <https://doi.org/10.1016/j.cedpsych.2020.101860>
- Sales, F. (2025, 14 de marzo). *OIT: Perú necesita 80,000 profesionales en tecnología este 2025*. *Calidad & Negocios*. <https://calidadynegocios.com/oit-peru-necesita-80000-profesionales-en-tecnologia-este-2025/>
- Salloum, S. A., AlHuneidi, K. A., & Baddar, I. F. (2025). Meta-analysis of artificial intelligence in education. *Education Research International*, 2025, 1045231. <https://doi.org/10.1155/2025/1045231>

- Scherer, R., Howard, S. K., Tondeur, J., & Siddiq, F. (2021). Profiling teachers' readiness for online teaching and learning in higher education: Who is ready? *Computers in Human Behavior*, *118*, 106675. <https://doi.org/10.1016/j.chb.2020.106675>
- Scherer, R., Siddiq, F., & Tondeur, J. (2019). The technology acceptance model (TAM): A meta-analytic structural equation modeling approach to explaining teachers' adoption of digital technology in education. *Computers & Education*, *128*, 13–35. <https://doi.org/10.1016/j.compedu.2018.09.009>
- Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa. (2023). *Modelo de acreditación para programas de estudios de institutos de educación superior y escuelas de educación superior*. <https://www.sineace.gob.pe/acreditacion/>
- Skalka, J., Drlik, M., & Obonya, J. (2021). Automated assessment in programming courses: A review. *Computer Applications in Engineering Education*, *29*(5), 1011–1026. <https://doi.org/10.1002/cae.22383>
- Soosai Raj, A. G., Gu, P., Zhang, E., Xavier Annie R, A., Williams, J., Halverson, R., & Patel, J. M. (2020). Live-coding vs static code examples: Which is better with respect to student learning and cognitive load? En *Proceedings of the Twenty-Second Australasian Computing Education Conference* (pp. 152–159). ACM. <https://doi.org/10.1145/3373165.3373182>
- Soto-Coronel, C. A., Ponce-Yactayo, D. L., & Vega-Rosales, D. M. (2024). Inteligencia artificial como herramienta en el aprendizaje autónomo de los estudiantes de educación superior. *Revista Venezolana de Gerencia*, *29*(8), 158–169. <https://doi.org/10.52080/rvgluz.29.e8.11>
- Stol, K.-J., & Fitzgerald, B. (2020). The ABC of software engineering research. *ACM Transactions on Software Engineering and Methodology*, *29*(3), Artículo 17, 1–51. <https://doi.org/10.1145/3382490>

- Superintendencia Nacional de Educación Superior Universitaria. (2023). *Informe bienal sobre la realidad universitaria en el Perú*. <https://www.sunedu.gob.pe/informe-bienal/>
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, 10(3), 251–296. <https://doi.org/10.1023/A:1022193728205>
- Teo, T. (2024). Re-examining the technology acceptance model (TAM) and its applications in educational contexts. *Educational Technology Research and Development*, 72(1), 45–67. <https://doi.org/10.1007/s11423-023-10298-5>
- Vallerand, R. J., Pelletier, L. G., Blais, M. R., Brière, N. M., Senécal, C., & Vallières, E. F. (1992). The Academic Motivation Scale: A measure of intrinsic, extrinsic, and amotivation in education. *Educational and Psychological Measurement*, 52(4), 1003–1017. <https://doi.org/10.1177/0013164492052004025>
- VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197–221. <https://doi.org/10.1080/00461520.2011.611369>
- Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2), 186–204. <https://doi.org/10.1287/mnsc.46.2.186.11926>
- Voštinár, P. (2024). Teaching programming using eduScrum methodology. *PeerJ Computer Science*, 10, e1822. <https://doi.org/10.7717/peerj-cs.1822>
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.
- Walkington, C., & Bernacki, M. L. (2020). Appraising research on personalized learning: Definitions, theoretical alignment, advancements, and future directions. *Journal of Research on Technology in Education*, 52(3), 235–252.

<https://doi.org/10.1080/15391523.2020.1747757>

Wang, C., Ma, W., Zhang, M., Lv, C., Wan, F., Lin, H., Tang, T., Liu, Y., & Ma, S. (2021).

Temporal cross-effects in knowledge tracing. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 517–525.

<https://doi.org/10.1145/3437963.3441802>

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. En

Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (pp. 39–44). ACM. <https://doi.org/10.1145/2591708.2591749>

Xu, Z. S., Sui, Z., Lian, L., Liu, C., & Liu, W. (2019). A novel intelligent recommendation

algorithm based on collaborative filtering. *Physica A: Statistical Mechanics and its Applications*, 519, 63–72. <https://doi.org/10.1016/j.physa.2018.12.020>

Younis, A. A., Sunderraman, R., Metzgar, M., & Bourgeois, A. G. (2021). Exploring the effects

of instructor and peer code review on student outcomes in an introductory programming course. En *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 541–547). ACM. <https://doi.org/10.1145/3408877.3432514>

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of

research on artificial intelligence applications in higher education — Where are the educators? *International Journal of Educational Technology in Higher Education*, 16,

Artículo 39. <https://doi.org/10.1186/s41239-019-0171-0>

IX. ANEXOS

**UNIVERSIDAD NACIONAL FEDERICO VILLARREAL
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

TÍTULO DE LA TESIS: “Efectividad de la enseñanza del lenguaje de programación

empleando inteligencia artificial en Educación Superior Tecnológica, Lima, 2025”

Autor: Pedro Hernán De la Cruz Velazco

ORCID: 0000-0003-4432-9214

DECLARACIÓN DE ORIGINALIDAD Y EXCLUSIVIDAD Yo, Pedro Hernán De la Cruz

Velazco, declaro bajo juramento que:

1. El presente Plan de Tesis es un trabajo original y exclusivo para optar el título de Ingeniero de Sistemas en la Universidad Nacional Federico Villarreal.
2. Este trabajo no ha sido presentado ni será presentado a ninguna otra institución, concurso o fuente de financiamiento externa.
3. El proyecto será ejecutado exclusivamente con recursos propios del investigador.
4. No existe ni existirá conflicto de intereses con ningún proyecto de investigación financiado por entidades externas.

Fecha: 30 de agosto de 2025



Firma: _____

Anexo A: MATRIZ DE CONSISTENCIA

Tabla A1

PROBLEMÁTICA	OBJETIVOS	HIPÓTESIS	VARIABLES	METODOLOGÍA
<p>Problema General: ¿De qué manera la implementación de herramientas de inteligencia artificial incrementa la efectividad de la enseñanza del lenguaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?</p>	<p>Objetivo General: Implementar herramientas de inteligencia artificial para incrementar la efectividad en la enseñanza de programación Java en estudiantes del III ciclo de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>Hipótesis General: La implementación de herramientas de inteligencia artificial incrementa significativamente la efectividad de la enseñanza del lenguaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>Variable Independiente: Herramientas de Inteligencia Artificial</p> <p>Dimensiones:</p> <ul style="list-style-type: none"> • Personalización • Interactividad • Disponibilidad <p>Variable Dependiente: Efectividad de la enseñanza del lenguaje de programación Java</p> <p>Dimensiones:</p> <ul style="list-style-type: none"> • Rendimiento Académico • Motivación y Compromiso • Percepción Tecnológica 	<p>Tipo: Aplicada Nivel: Explicativo Diseño: Pre-experimental G: $O_1 \rightarrow X \rightarrow O_2$ Enfoque: Cuantitativo</p> <p>Población: 2,199 estudiantes Muestra: 30 estudiantes Muestreo: No probabilístico por conveniencia</p>

<p>PE1: ¿De qué manera la implementación de herramientas de inteligencia artificial incrementa el rendimiento académico en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?</p>	<p>OE1: Evaluar el efecto de la implementación de herramientas de inteligencia artificial en el rendimiento académico en programación Java de estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>HE1: La implementación de herramientas de inteligencia artificial incrementa significativamente el rendimiento académico en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>Dimensión VD: Rendimiento Académico</p> <p>Subdimensiones:</p> <ul style="list-style-type: none"> • Conocimientos Teóricos de Java • Habilidades Prácticas con Java 	<p>Instrumentos:</p> <ul style="list-style-type: none"> • TCP-J: Test de Conocimientos de Programación Java (40 ítems) • RCC: Rúbrica de Calidad de Código
<p>PE2: ¿De qué manera la implementación de herramientas de inteligencia artificial incrementa la motivación intrínseca y compromiso conductual de los estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?</p>	<p>OE2: Determinar el efecto de la implementación de herramientas de inteligencia artificial en la motivación intrínseca y el compromiso conductual de los estudiantes del III ciclo, en el aprendizaje de Java y programación orientada a objetos, durante el año 2025.</p>	<p>HE2: La implementación de herramientas de inteligencia artificial incrementa significativamente la motivación intrínseca y compromiso conductual en estudiantes del III ciclo del programa de Computación e Informática.</p>	<p>Dimensión VD: Motivación y Compromiso</p> <p>Subdimensiones:</p> <ul style="list-style-type: none"> • Motivación Intrínseca • Compromiso Conductual 	<p>Instrumentos:</p> <ul style="list-style-type: none"> • AMS-C28: Escala de Motivación Académica (28 ítems) • FOS: Ficha de Observación Sistemática

<p>PE3: ¿De qué manera la implementación de herramientas de inteligencia artificial influye en la percepción tecnológica (utilidad percibida y facilidad de uso) sobre el aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025?</p>	<p>OE3: Analizar el efecto de la implementación de herramientas de inteligencia artificial en la percepción tecnológica de utilidad y facilidad de uso para el aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>HE3: La implementación de herramientas de inteligencia artificial incrementa significativamente la percepción de utilidad y facilidad de uso del aprendizaje de programación Java en estudiantes del III ciclo del programa de Computación e Informática de un instituto de educación superior tecnológico de Lima durante el año 2025.</p>	<p>Dimensión VD: Percepción Tecnológica</p> <p>Subdimensiones:</p> <ul style="list-style-type: none"> • Utilidad Percibida • Facilidad de Uso Percibida 	<p>Instrumento:</p> <ul style="list-style-type: none"> • TAM-AIED: Cuestionario TAM para IA Educativa <p>Técnicas de análisis:</p> <ul style="list-style-type: none"> • Estadística descriptiva • Prueba t de Student • Prueba de Wilcoxon • Regresión múltiple • Análisis de mediación
---	---	--	--	---

Nota. VI = Variable Independiente; VD = Variable Dependiente; PE = Problema Específico; OE = Objetivo Específico; HE = Hipótesis Específica. TCP-J = Test de Conocimientos de Programación Java.

Diagrama del Diseño Pre-experimental

DISEÑO PRE-EXPERIMENTAL CON PRE-TEST Y POST-TEST (Un solo grupo)

$$G: O_1 \rightarrow X \rightarrow O_2$$

Donde:

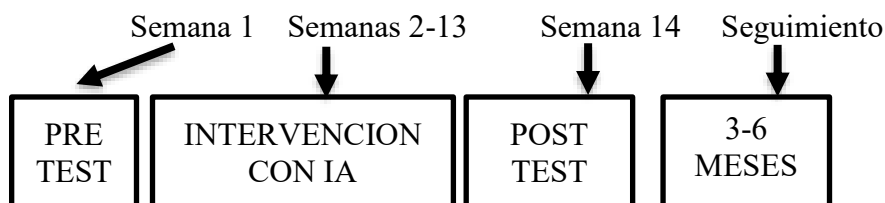
G = Grupo experimental (n=30 estudiantes del III ciclo)

O₁ = Pretest (Medición inicial - Semana 1)

X = Tratamiento experimental (Semanas 2-13)

O₂ = Postest (Medición final - Semana 14)

LÍNEA DE TIEMPO:



Base de datos:

ID	TCP_M_Tot al_Pre	TCP_M_Tot al_Post	Conocimien tos_Teorico s_Pre	Conocimien tos_Teorico s_Post	Aplicacion Practica_Pre	Aplicacion Practica_Post	Motivacion Intrinseca_Pre	Motivacion Intrinseca_Post	Motivacion Extrinseca_Pre	Motivacion Extrinseca_Post	Amotivacion_Pre	Amotivacion_Post	Utilidad_Pe rcebida_Pre	Utilidad_Pe rcebida_Post	Facilidad_U so_Pre	Facilidad_U so_Post
1	21.45	32.38	11.89	16.45	9.56	14.73	4.78	8.35	5.12	5.45	3.12	1.89	4.01	4.89	3.45	4.67
2	18.23	28.45	9.67	13.12	8.56	13.55	4.67	3.98	4.89	4.56	2.34	1.45	3.89	4.78	2.78	4.45
3	15.67	26.34	8.45	14.67	7.22	11.67	3.40	5.23	5.45	5.12	3.12	1.45	3.12	4.23	2.34	3.89
4	23.45	33.56	11.34	17.89	11.11	15.67	5.45	6.45	5.34	4.78	2.89	1.45	4.23	4.78	3.67	4.34
5	19.78	30.12	10.89	16.34	8.89	13.78	4.34	6.12	4.34	4.89	2.67	1.67	3.78	4.45	2.89	4.34
6	17.89	27.67	9.23	14.89	8.66	12.78	3.89	3.67	5.56	5.23	2.89	1.78	3.45	4.45	2.56	4.12
7	20.34	31.23	11.23	16.78	9.11	14.45	4.56	8.23	5.12	4.67	2.89	1.56	4.01	4.78	3.12	4.12
8	16.45	25.89	8.78	13.56	7.67	12.33	3.67	5.45	4.67	5.12	2.45	1.56	3.23	4.34	2.45	3.78
9	22.67	32.89	12.01	17.67	10.45	15.88	5.01	8.34	5.89	5.01	3.23	1.78	3.78	4.89	3.56	4.78
10	18.89	29.34	10.12	15.67	8.77	13.67	3.98	5.89	5.01	4.98	2.78	1.89	3.45	4.56	2.67	4.23
11	19.12	30.45	10.67	16.12	8.45	14.33	4.45	6.01	4.98	5.12	2.78	1.78	3.45	4.23	2.98	4.45
12	20.78	31.67	11.56	17.23	10.22	14.44	5.01	5.67	5.12	4.89	3.23	2.01	3.34	4.78	3.34	4.67
13	17.23	27.12	9.45	14.34	7.11	12.78	3.78	3.56	4.89	5.01	2.67	1.67	3.34	4.34	2.56	4.01
14	20.89	31.89	11.34	17.01	9.55	14.88	4.01	8.23	5.12	4.78	3.12	1.56	3.78	4.89	3.23	4.67
15	16.78	26.78	8.89	14.12	7.89	12.66	3.89	5.78	5.12	5.12	2.56	1.56	3.45	4.45	2.67	4.12
16	21.34	32.45	11.45	17.56	10.45	14.89	4.01	5.45	5.78	4.67	3.12	2.12	3.45	4.89	3.23	4.12
17	18.56	29.12	10.23	15.78	8.33	13.67	4.01	5.01	5.01	5.12	2.89	1.78	3.78	4.67	2.78	4.34
18	19.45	30.78	10.78	16.45	8.67	14.33	4.56	5.89	4.89	4.89	2.78	1.89	3.78	4.78	2.89	4.56
19	20.12	30.34	11.45	17.12	9.67	14.22	3.67	6.23	5.34	5.01	2.89	1.89	3.89	4.78	3.12	4.23
20	17.67	27.89	9.34	14.67	8.33	13.66	3.88	5.45	4.78	4.78	2.78	1.67	3.45	4.45	2.56	4.23
21	20.12	30.89	11.12	18.89	9.00	14.00	3.89	8.01	5.01	4.67	2.89	1.89	3.89	4.67	3.01	4.56
22	18.34	28.67	9.89	13.34	8.22	13.33	3.67	3.89	5.12	5.01	2.89	1.78	3.67	4.56	2.78	4.34
23	22.89	33.67	12.23	18.12	10.66	15.23	5.34	8.45	5.12	4.89	3.34	2.12	4.01	4.98	3.01	4.01
24	16.89	26.45	8.67	13.89	8.22	12.56	3.78	5.56	4.89	5.12	2.56	1.56	3.34	4.34	2.45	3.89
25	19.89	30.56	10.23	16.56	9.00	14.00	4.56	6.12	5.23	4.89	3.01	1.56	3.89	4.78	2.98	4.45
26	20.56	32.12	11.78	16.34	9.78	14.78	4.23	5.12	5.67	5.01	3.01	1.56	3.89	4.67	3.11	4.45
27	17.45	27.45	9.12	14.45	8.33	13.00	3.89	3.67	4.98	4.70	2.67	1.67	3.45	4.45	2.67	4.12
28	20.67	30.45	11.23	18.89	9.44	14.56	4.78	8.23	5.01	4.78	2.67	1.67	4.01	4.78	3.12	4.12
29	18.78	29.45	10.34	13.89	8.44	13.56	3.67	3.98	5.01	4.78	2.89	1.78	3.78	4.67	2.89	4.34
30	19.56	30.34	10.67	16.23	8.22	14.78	3.89	6.01	5.12	4.78	3.01	1.78	3.45	4.67	2.98	4.45

Anexo B: Instrumentos de Recolección de Datos

B.1 Test de Conocimientos de Programación (TCP-J)

TEST DE CONOCIMIENTOS DE PROGRAMACIÓN - TCP-J

Instrumento de Investigación

Elaborado por: Bach. Pedro Hernán De la Cruz Velazco

Tesis: "Efectividad de la Enseñanza del Lenguaje de Programación Empleando Inteligencia Artificial en Educación Superior Tecnológica"

Universidad Nacional Federico Villarreal Facultad de Ingeniería Industrial y de Sistemas Año: 2025

Instrucciones Generales:

- Tiempo total: 90 minutos
- Este test consta de dos partes:
 - Conocimientos Teóricos (20 ítems) y Aplicación Práctica (20 ítems)
- No está permitido el uso de internet ni herramientas de IA
- Puede usar el IDE de Java instalado en el computador
- Lea cuidadosamente cada pregunta antes de responder
- Cada pregunta correcta vale 1 punto

DATOS DEL ESTUDIANTE:

- Código: EST-2025-_____
- Fecha: __/__/2025 - Hora inicio: _____
- Hora fin: _____

Anexo C

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

Facultad de Ingeniería Industrial y de Sistemas

Escuela Profesional de Ingeniería de Sistemas

TEST DE CONOCIMIENTOS DE PROGRAMACIÓN JAVA (TCP-J)*Instrumento de Investigación*

Elaborado por: Bach. Pedro Hernán De la Cruz Velazco

Universidad Nacional Federico Villarreal — Facultad de Ingeniería Industrial y de Sistemas

Año: 2025

Código: EST-2025-_____	Fecha: ____ / ____ / 2025	Aplicación: <input type="checkbox"/> PRE-TEST <input type="checkbox"/> POST-TEST
Hora inicio: _____	Hora fin: _____	Ciclo: III Sección: _____

Instrucciones Generales:

- Tiempo total: 90 minutos.
- Este test consta de dos partes: Parte I — Conocimientos Teóricos (20 ítems) y Parte II — Aplicación Práctica (20 ítems).
- No está permitido el uso de internet ni herramientas de inteligencia artificial.
- Puede usar el IDE de Java instalado en el computador (IntelliJ IDEA o Visual Studio Code con extensión Java).
- Lea cuidadosamente cada pregunta antes de responder.
- Cada ítem correcto vale 1 punto. Puntaje máximo: 40 puntos.

PARTE I: CONOCIMIENTOS TEÓRICOS — 20 ítems

Instrucciones: Marque con una (X) la alternativa correcta. Solo una respuesta es válida por ítem.

Bloque A — Fundamentos del lenguaje Java

1. ¿Cuál es la diferencia principal entre un Array y un ArrayList en Java?

- a) Un Array puede cambiar de tamaño durante la ejecución; un ArrayList tiene tamaño fijo.
- b) Un ArrayList puede crecer dinámicamente; un Array tiene tamaño fijo definido al momento de su creación.
- c) Ambas estructuras tienen tamaño fijo; solo difieren en la sintaxis de declaración.
- d) Un ArrayList almacena exclusivamente tipos de datos primitivos.

2. ¿Cuál es el resultado de la siguiente expresión en Java?

```
int resultado = 5 / 2;
```

```
System.out.println(resultado);
```

- a) 2.5
- b) 2
- c) 3
- d) Error de compilación

3. ¿Cuál de las siguientes opciones declara correctamente un método público que no retorna ningún valor en Java?

- a) def calcular():
- b) function calcular() {}
- c) public void calcular() {}
- d) func calcular() -> void {}

4. En un `ArrayList<String>` de Java, ¿qué método agrega un elemento al final de la lista?

- a) lista.insert("Java")
- b) lista.append("Java")
- c) lista.push("Java")
- d) lista.add("Java")

5. ¿Cuál es el propósito principal de un constructor en una clase Java?

- a) Liberar la memoria del objeto cuando este finaliza su uso.
- b) Inicializar el estado del objeto en el momento en que se instancia con la palabra clave 'new'.
- c) Declarar las variables estáticas que pertenecen a la clase.
- d) Sobreescribir métodos heredados de la superclase.

6. ¿Qué estructura de control se utiliza para manejar excepciones en Java?

- a) if-else
- b) try-catch
- c) while-do
- d) switch-case

7. ¿Qué interfaz de Java implementa una colección de pares clave-valor?

- a) `ArrayList`
- b) `LinkedList`
- c) `Map`, implementada por la clase `HashMap<K,V>`
- d) `Set`

8. ¿Qué hace la palabra clave 'static' cuando se aplica a un método de una clase Java?

- a) Indica que el método solo puede ser llamado desde clases que hereden de esta.
- b) Indica que el método pertenece a la clase y no a una instancia; se invoca con `NombreClase.metodo()`.
- c) Impide que el método reciba parámetros de cualquier tipo.
- d) Hace que el método se ejecute automáticamente cada vez que se crea un nuevo objeto.

9. ¿Cuál será la salida del siguiente fragmento de código Java al ejecutarse?

```
int[] arr = {10, 20, 30};  
System.out.println(arr[1]);
```

- a) 10
- b) 20
- c) 30
- d) `ArrayIndexOutOfBoundsException`

10. ¿Cuál es la diferencia entre los modificadores de acceso 'private' y 'public' en Java?

- a) 'public' se aplica únicamente a atributos; 'private' solo puede usarse en métodos.
- b) 'private' restringe el acceso al interior de la propia clase; 'public' permite el acceso desde cualquier clase.
- c) No existe diferencia funcional entre ambos; su uso es solo una convención de estilo de código.
- d) El modificador 'private' mejora significativamente el rendimiento de la JVM.

Bloque B — Programación Orientada a Objetos y Estructuras de Datos

11. ¿Qué concepto de la Programación Orientada a Objetos permite que una subclase redefina el comportamiento de un método heredado de su superclase?

- a) Encapsulamiento
- b) Abstracción
- c) Sobrescritura de métodos, indicada con la anotación `@Override`
- d) Sobrecarga de métodos (overloading)

12. ¿Cuál de los siguientes fragmentos define correctamente una relación de herencia entre la clase 'Auto' y la clase 'Vehiculo' en Java?

- a) `class Auto include Vehiculo { }`
- b) `class Auto extends Vehiculo { }`
- c) `class Auto implements Vehiculo { }`
- d) `class Auto inherits Vehiculo { }`

13. ¿Qué estructura de datos en Java sigue el principio LIFO (Last In, First Out)?

- a) Queue (Cola)
- b) ArrayList
- c) Stack / Deque usado como pila
- d) HashMap

14. ¿Cuál es la diferencia fundamental entre una interfaz (interface) y una clase abstracta (abstract class) en Java?

- a) No existe diferencia; en Java moderno ambos conceptos son sinónimos.
- b) Una interfaz puede tener métodos completamente implementados; una clase abstracta no puede tener ninguno.
- c) Una clase abstracta puede tener constructor, atributos de instancia y estado; una interfaz define principalmente un contrato de comportamiento.

d) Una interfaz solo puede ser implementada por una única clase del sistema.

15. ¿Cuál es la complejidad temporal de acceder a un elemento de un `ArrayList<T>` mediante su índice, usando el método `get(i)`?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(1)$
- d) $O(n^2)$

Bloque C — Algoritmos, Control de Flujo y Análisis de Código

16. ¿Cuál es la diferencia entre las sentencias 'break' y 'continue' dentro de un bucle for en Java?

- a) Ambas sentencias terminan la ejecución del bucle de forma inmediata.
- b) 'break' termina el bucle completo; 'continue' omite el resto del cuerpo en esa iteración y pasa a la siguiente.
- c) 'continue' es quien termina el bucle; 'break' salta a la siguiente iteración.
- d) El compilador de Java trata ambas sentencias de forma idéntica.

17. ¿Cuál será la salida del siguiente código Java al ejecutarse?

```
ArrayList<Integer> listA = new ArrayList<>(List.of(1, 2, 3));
ArrayList<Integer> listB = listA;
listB.add(4);
System.out.println(listA);
```

- a) [1, 2, 3]
- b) [1, 2, 3, 4]
- c) Error de compilación
- d) NullPointerException en tiempo de ejecución

18. ¿Cuál de las siguientes opciones muestra la forma correcta de usar un bucle for-each en Java para recorrer un arreglo `int[]`?

- a) `for (int i : arr.length) { }`
- b) `for each (int n in arr) { }`
- c) `for (int n : arr) { System.out.println(n); }`
- d) `foreach (int n in arr) { }`

19. ¿Qué imprime el siguiente código Java al ejecutarse?

```
for (int i = 0; i < 5; i += 2) {
    System.out.print(i + " ");
}
```

- a) 0 1 2 3 4
- b) 1 2 3 4 5
- c) 0 2 4
- d) 0 1 2

20. ¿Qué diferencia existe entre el operador '==' y el método .equals() cuando se comparan dos objetos de tipo String en Java?

- a) El operador '==' y el método .equals() son completamente equivalentes para comparar objetos String.
- b) '==' compara las referencias de memoria de los objetos; .equals() compara el contenido del texto carácter a carácter.
- c) .equals() compara las referencias de memoria; '==' compara el contenido del texto.
- d) .equals() solo funciona correctamente con tipos de datos primitivos.

PARTE II: APLICACIÓN PRÁCTICA — 20 ítems

Instrucciones: Escriba el código Java solicitado en el espacio indicado o complete el fragmento proporcionado. Utilice sintaxis Java correcta. Se evaluará funcionalidad, sintaxis y legibilidad.

Bloque D — Métodos fundamentales y tipos de datos

21. Escriba un método estático que calcule y retorne la suma de todos los elementos de un arreglo `int[]`:

```
public static int sumarArreglo(int[] numeros) {  
    // Su código aquí  
}
```

22. Complete el método que retorna el mayor de dos enteros recibidos como parámetros, sin usar `Math.max()`:

```
public static int mayor(int a, int b) {  
    // Su código aquí  
}
```

23. Implemente un método que retorne `true` si el número entero recibido es par, o `false` si es impar:

```
public static boolean esPar(int n) {  
    // Su código aquí  
}
```

24. Escriba un método que convierta una temperatura de Celsius a Fahrenheit y retorne el resultado. Fórmula: $F = C \times 9/5 + 32$

```
public static double celsiusAFahrenheit(double celsius) {  
    // Su código aquí  
}
```

25. Implemente un método que cuente y retorne la cantidad de vocales (mayúsculas y minúsculas) que contiene un String:

```
public static int contarVocales(String texto) {  
    // Su código aquí  
}
```

Bloque E — Arreglos, ArrayList y colecciones

26. Escriba un método que invierta los elementos de un arreglo `int[]` y retorne el nuevo arreglo invertido:

```
public static int[] invertirArreglo(int[] arr) {  
    // Su código aquí  
}
```

27. Implemente un método que encuentre y retorne el valor máximo en un arreglo `int[]`, sin usar `Math.max()` ni `Arrays.stream()`:

```
public static int maximo(int[] arr) {  
    // Su código aquí  
}
```

28. Escriba un método que reciba un `ArrayList<Integer>` y retorne un nuevo `ArrayList<Integer>` que contenga únicamente los números pares:

```
import java.util.ArrayList;  
public static ArrayList<Integer> filtrarPares(ArrayList<Integer> lista) {  
    // Su código aquí  
}
```

29. Implemente un método que cuente y retorne cuántas veces aparece un `String` dado dentro de un `ArrayList<String>`:

```
public static int contarOcuurrencias(ArrayList<String> lista, String elemento) {  
    // Su código aquí
```

```
}
```

30. Escriba un método que elimine los elementos duplicados de un `ArrayList<Integer>` y retorne una nueva lista sin repetidos, preservando el orden de primera aparición:

```
public static ArrayList<Integer> sinDuplicados(ArrayList<Integer> lista) {  
    // Su código aquí  
}
```

Bloque F — Algoritmos de búsqueda, ordenamiento y recursión

31. Implemente el algoritmo de búsqueda lineal: retorna el índice donde se encuentra el elemento en el arreglo, o -1 si no existe:

```
public static int busquedaLineal(int[] arr, int objetivo) {  
    // Su código aquí  
}
```

32. Implemente el algoritmo de ordenamiento burbuja (Bubble Sort) para ordenar un arreglo `int[]` en orden ascendente. El método ordena el arreglo en sitio (no retorna un nuevo arreglo):

```
public static void burbuja(int[] arr) {
    // Su código aquí
}
```

33. Escriba un método recursivo que calcule el n-ésimo número de la serie Fibonacci. Considere: $F(0) = 0$, $F(1) = 1$, $F(n) = F(n-1) + F(n-2)$:

```
public static int fibonacci(int n) {
    // Su código aquí (debe ser recursivo)
}
```

34. Implemente recursivamente el cálculo del factorial de un número entero n. Considere que $0! = 1$ y use el tipo long para soportar valores grandes:

```
public static long factorial(int n) {
    // Su código aquí (debe ser recursivo)
}
```

35. Escriba un método que determine si un String es palíndromo. Debe ignorar mayúsculas y espacios. Por ejemplo: "Anita lava la tina" es palíndromo:

```
public static boolean esPalindromo(String texto) {  
    // Su código aquí  
}
```

Bloque G — Programación Orientada a Objetos

36. Cree la clase 'Rectangulo' con los atributos 'base' y 'altura' de tipo double, un constructor que reciba ambos valores, y los métodos area() y perimetro() que retornen el resultado correspondiente:

```
class Rectangulo {  
    // Declare los atributos  
    // Implemente el constructor  
    // Implemente area() y perimetro()  
}
```

37. Cree la clase 'Estudiante' con los atributos 'nombre' (String) y 'notas' (ArrayList<Double>). Incluya: un constructor que reciba el nombre, el método agregarNota(double nota) y el método promedio() que retorne el promedio de las notas como double:

```
import java.util.ArrayList;  
class Estudiante {  
    // Su código aquí  
}
```

38. Defina la interfaz 'Figura' con el método abstracto area() que retorna double. Luego cree la clase 'Circulo' que implemente dicha interfaz, con el atributo 'radio' (double), su constructor y el método area() usando Math.PI:

```
interface Figura {  
    // Declare el método abstracto  
}  
class Circulo implements Figura {  
    // Su código aquí  
}
```

39. Escriba un método que reciba un arreglo String[] de palabras y retorne un HashMap<String, Integer> con la frecuencia de aparición de cada palabra:

```
import java.util.HashMap;  
public static HashMap<String, Integer> contarPalabras(String[] palabras) {  
    // Su código aquí  
}
```

40. Implemente un método que divida dos enteros usando try-catch para manejar la excepción ArithmeticException. Si el divisor es cero, imprima 'División por cero detectada' y retorne 0:

```
public static int dividirSeguro(int dividendo, int divisor) {  
    // Su código aquí con try-catch  
}
```


Nº	ENUNCIADO	ESCALA
1	Porque siento placer y satisfacción cuando aprendo cosas nuevas sobre programación	1 2 3 4 5 6 7
2	Por el placer que siento cuando descubro cosas nuevas que nunca había visto	1 2 3 4 5 6 7
3	Por el placer de saber más sobre programación que me atrae	1 2 3 4 5 6 7
4	Porque me gusta ver que soy capaz de resolver problemas complejos	1 2 3 4 5 6 7
5	Por la satisfacción que siento cuando logro hacer funcionar mi código	1 2 3 4 5 6 7
6	Por el placer que experimento al ampliar mis conocimientos	1 2 3 4 5 6 7
7	Por la satisfacción que siento al superar desafíos de programación	1 2 3 4 5 6 7
8	Porque disfruto el proceso de crear programas desde cero	1 2 3 4 5 6 7
9	Por el placer intelectual de entender algoritmos complejos	1 2 3 4 5 6 7
10	Porque me emociona resolver bugs difíciles	1 2 3 4 5 6 7
11	Por la satisfacción personal de mejorar mis habilidades	1 2 3 4 5 6 7
12	Porque genuinamente disfruto programar	1 2 3 4 5 6 7
MOTIVACIÓN EXTRÍNSECA		
13	Porque con solo el bachillerato no podría encontrar un empleo bien pagado	1 2 3 4 5 6 7
14	Para poder conseguir un trabajo más prestigioso	1 2 3 4 5 6 7
15	Porque quiero tener una vida cómoda más adelante	1 2 3 4 5 6 7
16	Para ganar un mejor salario en el futuro	1 2 3 4 5 6 7
17	Porque aprobar me hace sentir importante	1 2 3 4 5 6 7
18	Para demostrar que soy capaz de terminar una carrera técnica	1 2 3 4 5 6 7
19	Porque quiero demostrar que soy una persona inteligente	1 2 3 4 5 6 7
20	Para no decepcionar a mi familia	1 2 3 4 5 6 7
21	Porque es lo que se espera de mí	1 2 3 4 5 6 7
22	Porque necesito el título para trabajar	1 2 3 4 5 6 7
23	Para obtener reconocimiento social	1 2 3 4 5 6 7
24	Porque mis padres quieren que estudie esto	1 2 3 4 5 6 7
AMOTIVACIÓN		
25	Honestamente no lo sé; siento que estoy perdiendo el tiempo	1 2 3 4 5 6 7
26	No sé por qué estudio programación y francamente no me importa	1 2 3 4 5 6 7

N°	ENUNCIADO	ESCALA
27	No logro entender qué hago estudiando programación	1 2 3 4 5 6 7
28	Tenía buenas razones, pero ahora me pregunto si debería continuar	1 2 3 4 5 6 7

B.3 Cuestionario TAM-AIEd (Technology Acceptance Model for AI in Education)

VERSIÓN PRE-TEST: EXPECTATIVAS

Basado en: Technology Acceptance Model - Davis (1989)

Adaptación y contextualización: Bach. Pedro Hernán De la Cruz Velazco Universidad Nacional Federico Villarreal Facultad de Ingeniería Industrial y de Sistemas

Año: 2025

Código del estudiante: EST-2025- _____

Fecha: ___/___/2025

Aplicación: PRE-TEST

INSTRUCCIONES:

Por favor, indique su grado de acuerdo con las siguientes afirmaciones sobre el USO FUTURO de herramientas de Inteligencia Artificial en su aprendizaje de programación. Use la siguiente escala:

- 1 = Totalmente en desacuerdo
- 2 = En desacuerdo
- 3 = Neutral
- 4 = De acuerdo
- 5 = Totalmente de acuerdo

N°	ENUNCIADO	ESCALA
	UTILIDAD PERCIBIDA	
1	Creo que usar herramientas de IA mejorará mi rendimiento en programación	1 2 3 4 5
2	Pienso que usar IA aumentará mi productividad al programar	1 2 3 4 5
3	Espero que la IA mejore mi efectividad en el aprendizaje	1 2 3 4 5

Nº	ENUNCIADO	ESCALA				
4	Creo que la IA me ayudará a aprender más rápido	1	2	3	4	5
5	Pienso que la IA será útil para resolver problemas complejos	1	2	3	4	5
6	Espero que la IA facilite mi comprensión de conceptos difíciles	1	2	3	4	5
7	Creo que la IA mejorará la calidad de mi código	1	2	3	4	5
8	Pienso que la IA será valiosa para mi formación	1	2	3	4	5
FACILIDAD DE USO PERCIBIDA						
9	Creo que aprender a usar las herramientas de IA será fácil para mí	1	2	3	4	5
10	Pienso que las herramientas serán fáciles de usar	1	2	3	4	5
11	Espero que la interacción con la IA sea clara y comprensible	1	2	3	4	5
12	Creo que será fácil hacer que la IA haga lo que quiero	1	2	3	4	5
13	Pienso que las instrucciones serán fáciles de seguir	1	2	3	4	5
14	Espero poder usar la IA sin ayuda externa	1	2	3	4	5
15	Creo que la interfaz será intuitiva	1	2	3	4	5
16	Pienso que no necesitaré mucho tiempo para aprender a usarla	1	2	3	4	5
INTENCIÓN DE USO						
17	Tengo la intención de usar IA cuando esté disponible	1	2	3	4	5

N°	ENUNCIADO	ESCALA				
18	Planeo usar IA regularmente en mis estudios	1	2	3	4	5
19	Intentaré usar IA para todos mis proyectos	1	2	3	4	5
20	Recomendaré el uso de IA a mis compañeros	1	2	3	4	5

VERSIÓN POST-TEST: EXPERIENCIA

**UNIVERSIDAD NACIONAL FEDERICO VILLARREAL
ESCUELA DE PREGRADO
CUESTIONARIO TAM-AIEd - VERSIÓN EXPERIENCIA**

Código del estudiante: EST-2025- _____

Fecha: ___/___/2025

Aplicación: POST-TEST

INSTRUCCIONES:

Por favor, indique su grado de acuerdo con las siguientes afirmaciones sobre su EXPERIENCIA con las herramientas de Inteligencia Artificial durante el curso. Use la siguiente escala:

- 1 = Totalmente en desacuerdo
- 2 = En desacuerdo
- 3 = Neutral
- 4 = De acuerdo
- 5 = Totalmente de acuerdo

N°	ENUNCIADO	ESCALA				
UTILIDAD PERCIBIDA						
1	Usar herramientas de IA mejoró mi rendimiento en programación	1	2	3	4	5
2	Usar IA augmentó mi productividad al programar	1	2	3	4	5
3	La IA mejoró mi efectividad en el aprendizaje	1	2	3	4	5

4	La IA me ayudó a aprender más rápido	1	2	3	4	5
5	La IA fue útil para resolver problemas complejos	1	2	3	4	5
6	La IA facilitó mi comprensión de conceptos difíciles	1	2	3	4	5
7	La IA mejoró la calidad de mi código	1	2	3	4	5
8	La IA fue valiosa para mi formación	1	2	3	4	5
FACILIDAD DE USO PERCIBIDA						
9	Aprender a usar las herramientas de IA fue fácil para mí	1	2	3	4	5
10	Las herramientas fueron fáciles de usar	1	2	3	4	5
11	La interacción con la IA fue clara y comprensible	1	2	3	4	5
12	Fue fácil hacer que la IA hiciera lo que quería	1	2	3	4	5
13	Las instrucciones fueron fáciles de seguir	1	2	3	4	5
14	Puede usar la IA sin ayuda externa	1	2	3	4	5
15	La interfaz fue intuitiva	1	2	3	4	5
16	No necesité mucho tiempo para aprender a usarla	1	2	3	4	5
INTENCIÓN DE USO FUTURO						
17	Continuaré usando IA para aprender programación	1	2	3	4	5
18	Seguiré usando IA regularmente en mis estudios	1	2	3	4	5
19	Usaré IA para mis proyectos futuros	1	2	3	4	5
20	Recomendaré el uso de IA a otros estudiantes	1	2	3	4	5

SATISFACCIÓN

N (Solo POST-TEST)

21	Estoy satisfecho con el apoyo que recibí de las herramientas de IA	1	2	3	4	5
22	Las herramientas de IA cumplieron con mis expectativas	1	2	3	4	5
23	Mi experiencia general con las herramientas de IA fue positiva	1	2	3	4	5
24	Las herramientas de IA mejoraron mi experiencia de aprendizaje	1	2	3	4	5

NET**PROMOTER**

SCORE (Solo POST-TEST)

25	En una escala del 0 al 10, ¿qué tan probable es que recomiendes estas herramientas de IA a un amigo o compañero?	0	1	2	3	4	5	6	7	8	9	10
----	--	---	---	---	---	---	---	---	---	---	---	----

B.4 Ficha de Observación Sistemática (FOS)**FICHA DE OBSERVACIÓN SISTEMÁTICA - FOS****Instrumento de Investigación****Elaborado por: Bach. Pedro Hernán De la Cruz Velazco****Universidad Nacional Federico Villarreal****Facultad de Ingeniería Industrial y de Sistemas Año: 2025****DATOS GENERALES - Fecha: ___/___/2025**

- Hora inicio: _____

- Hora fin : _____

- Observador: _____

- Sesión: [] PRE-TEST [] POST-TEST

- Tema de clase: _____

- N° de estudiantes presentes: _____

SECCIÓN 1: PARTICIPACIÓN ACTIVA

CONDUCTA OBSERVADA	FRECUENCIA
	Nunca (0)
Realiza preguntas al docente	[]
Responde preguntas del docente	[]
Aporta ideas o soluciones	[]
Participa en discusiones grupales	[]
Solicita clarificaciones	[]

SECCIÓN 2: COLABORACIÓN

CONDUCTA OBSERVADA	FRECUENCIA
	Nunca
Ayuda espontáneamente a compañeros	[]
Solicita ayuda a pares	[]
Trabaja efectivamente en equipo	[]
Comparte recursos o conocimientos	[]
Participa en discusiones entre pares	[]

SECCIÓN 3: PERSISTENCIA

INDICADOR	MEDICIÓN
Tiempo promedio antes de solicitar ayuda	_____ minutos
Número de intentos antes de rendirse	_____ intentos
Tiempo continuo de trabajo sin distracciones	_____ minutos
Reacciona positivamente ante errores	Sí [] No []
Intenta múltiples soluciones	Sí [] No []
Completa tareas, aunque sean difíciles	Sí [] No []

SECCIÓN 4: USO DE RECURSOS

RECURSO	FRECUENCIA DE USO
	No usa
Consulta documentación oficial	[]
Busca en internet	[]
Usa herramientas IA*	[]
Revisa apuntes de clase	[]
Consulta libros/manuales	[]

*Solo aplicable durante la intervención (semanas 2-15)

SECCIÓN 5: INDICADORES EMOCIONALES

ESTADO OBSERVADO	INTENSIDAD
	Nunca
Muestra frustración	[]
Expresa satisfacción	[]
Demuestra ansiedad	[]
Evidencia motivación	[]
Muestra confianza	[]

OBSERVACIONES CUALITATIVAS:

Firma del observador: _____

B.5 Rúbrica de Calidad de Código (RCC)

RÚBRICA DE CALIDAD DE CÓDIGO - RCC Instrumento de Evaluación Elaborado por: Bach. Pedro Hernán De la Cruz Velazco Universidad Nacional Federico Villarreal Facultad de Ingeniería Industrial y de Sistemas Año: 2025

IDENTIFICACIÓN - Código del estudiante: EST-2025-_____

- Proyecto evaluado: _____

- Evaluador: _____

- Fecha: ____/____/2025 - Tipo: [] PRE-TEST [] POST-TEST

ESCALA DE VALORACIÓN:

1 = Inicial (0-25% de logro)

2 = En desarrollo (26-50% de logro)

3 = Competente (51-75% de logro)

4 = Avanzado (76-100% de logro)

DIMENSIÓN / CRITERIO	INICIAL (1)	EN DESARROLLO (2)	COMPETEN TE (3)	AVANZAD O (4)	PUNT AJE
-------------------------	----------------	-------------------------	--------------------	------------------	-------------

1.**FUNCIONALIDA****D (25%)**

1.1 Cumplimiento de requisitos	No cumple requisitos básicos	Cumple <50% de requisitos	Cumple 50-75% de requisitos	Cumple >75% de requisitos	—
1.2 Manejo de casos edge	No considera casos especiales	Maneja algunos casos básicos	Maneja mayoría de casos	Maneja todos los casos previsibles	—
1.3 Validación de entrada	Sin validación	Validación mínima	Validación adecuada	Validación exhaustiva	—
Subtotal					—
Funcionalidad					/12

2. EFICIENCIA**(25%)**

2.1 Complejidad algorítmica	Solución ineficiente $O(n^{3+})$	Solución subóptima $O(n^2)$	Solución eficiente $O(n \log n)$	Solución óptima $O(n)$ o mejor	—
2.2 Uso de memoria	Uso excesivo sin justificación	Uso moderado con desperdicio	Uso eficiente	Uso optimizado	—
2.3 Tiempo de ejecución	Muy lento para casos normales	Lento para casos grandes	Aceptable para todos los casos	Rápido incluso para casos extremos	—
Subtotal Eficiencia					—
					/12

3. LEGIBILIDAD**(25%)**

3.1 Nombres de variables	Nombres sin sentido (a, b, x)	Nombres poco descriptivos	Nombres claros	Nombres autodocumentados	_____
3.2 Indentación y formato	Sin formato consistente	Formato irregular	Formato consistente	Formato profesional	_____
3.3 Comentarios	Sin comentarios	Comentarios escasos	Comentarios adecuados	Comentarios excelentes	_____
Subtotal					_____
Legibilidad					/12

4.

MANTENIBILIDAD**D (25%)**

4.1 Modularidad	Código monolítico	Poca separación de funciones	Buena modularización	Excelente diseño modular	_____
4.2 Reutilización	Sin funciones reutilizables	Pocas funciones reutilizables	Buen nivel de reutilización	Alto nivel de abstracción	_____
4.3 Documentación	Sin documentación	Documentación mínima	Documentación adecuada	Documentación completa	_____
Subtotal					_____
Mantenibilidad					/12

CÁLCULO DE PUNTAJE: - Puntaje total: _____ / 48 - Porcentaje: _____ % - Calificación (escala vigesimal): _____ / 20

COMENTARIOS DEL EVALUADOR:

Firma del evaluador: _____

Anexo D: Consentimiento Informado

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL ESCUELA DE PREGRADO

CONSENTIMIENTO INFORMADO PARA PARTICIPACIÓN EN INVESTIGACIÓN

Título del Estudio: *"Implementación de Herramientas de Inteligencia Artificial para la Mejora de la Efectividad en la Enseñanza de Programación Java en Estudiantes del III Ciclo de Computación e Informática de un Instituto Tecnológico de Lima, 2025"*

Investigador Principal: Pedro Hernán De la Cruz Velazco

Institución: Instituto de Educación Superior Tecnológico [Nombre]

Correo electrónico: pedrohernandcv@istpargentina.edu.pe

Teléfono: (+51) XXX-XXX-XXX

1. INVITACIÓN A PARTICIPAR

Estimado(a) estudiante:

Usted ha sido invitado(a) a participar en un estudio de investigación educativa que busca mejorar los métodos de enseñanza de programación. Antes de decidir si participa o no, es importante que comprenda por qué se está realizando la investigación y qué implicará su participación. Por favor, tómese el tiempo necesario para leer cuidadosamente la siguiente información. Si algo no está claro o si desea más información, no dude en preguntar.

2. PROPÓSITO DEL ESTUDIO

El objetivo principal de esta investigación es evaluar si el uso de herramientas de inteligencia artificial mejora el aprendizaje de programación en estudiantes del III ciclo. Específicamente, buscamos determinar si estas herramientas pueden:

- Mejorar el rendimiento académico en programación
- Aumentar la motivación para aprender
- Facilitar la comprensión de conceptos complejos de programación
- Desarrollar mejores habilidades de codificación
- Reducir la frustración asociada al aprendizaje de programación

3. ¿POR QUÉ HE SIDO ELEGIDO?

Usted ha sido seleccionado porque cumple con los siguientes criterios:

- Está matriculado en el III ciclo del programa de Computación e Informática
- Ha completado satisfactoriamente los cursos de Programación I y II
- Tiene acceso regular a una computadora con conexión a internet
- Asiste regularmente a las clases presenciales

En total, participarán 30 estudiantes de su misma sección, todos voluntarios.

4. ¿TENGO QUE PARTICIPAR?

La participación en este estudio es completamente **VOLUNTARIA**. Esto significa que:

- Usted decide libremente si desea participar o no
- Puede retirarse del estudio en cualquier momento sin dar explicaciones
- Su decisión NO afectará sus calificaciones
- Su decisión NO afectará su relación con los docentes o la institución
- Su decisión NO afectará sus derechos como estudiante
- Si decide no participar, continuará con sus clases normalmente

5. ¿QUÉ PASARÁ SI PARTICIPO?

Si acepta participar, su compromiso incluirá:

Fase 1 - Inicio (Semana 1): - Completar cuestionarios iniciales sobre conocimientos y motivación (2 horas) - Participar en 2 sesiones de familiarización con las herramientas de IA (2 horas total) - Realizar una prueba de conocimientos de programación (90 minutos)

Fase 2 - Intervención (Semanas 2-15): - Usar las herramientas de IA durante las clases regulares de programación - Acceder al Sistema Tutorial Inteligente para práctica personalizada - Utilizar el Chatbot educativo para resolver dudas 24/7 - Emplear el sistema de evaluación automática para sus ejercicios - Permitir el registro anónimo de sus interacciones con el sistema - Continuar con todas sus actividades académicas normales

Fase 3 - Evaluación Final (Semana 16): - Completar los mismos cuestionarios del inicio (2

horas) - Realizar una prueba final de conocimientos (90 minutos) - Posible invitación a una entrevista opcional (30 minutos) - Participar en un grupo focal opcional (90 minutos)

6. ¿CUÁLES SON LOS BENEFICIOS?

Beneficios directos para usted: - Acceso gratuito a herramientas avanzadas de IA - Sistema tutorial personalizado que se adapta a su nivel - Chatbot disponible 24/7 para resolver dudas - Retroalimentación inmediata en todos sus ejercicios - Materiales de aprendizaje adicionales y personalizados - Certificado de participación en investigación científica - Posible mejora en su rendimiento académico

Beneficios para la sociedad: - Contribuir al mejoramiento de la educación tecnológica en el Perú - Ayudar a futuros estudiantes con métodos más efectivos - Apoyar el desarrollo de la investigación educativa nacional

7. ¿CUÁLES SON LOS RIESGOS?

Los riesgos de participar son mínimos y pueden incluir:

- Fatiga temporal por el uso de nuevas tecnologías
- Posible frustración inicial mientras aprende a usar las herramientas
- Tiempo adicional (aproximadamente 30 minutos semanales) para familiarizarse con el sistema
- Posible ansiedad al ser observado durante las clases

Para minimizar estos riesgos, proporcionaremos: - Soporte técnico permanente durante todo el estudio - Tutoriales detallados paso a paso - Sesiones de apoyo psicopedagógico si las necesita - Flexibilidad en los tiempos de adaptación - Canal directo de comunicación para resolver dudas

8. CONFIDENCIALIDAD

Su información será estrictamente confidencial. Garantizamos que:

- Se le asignará un código anónimo (ejemplo: EST-2025-001)
- Su nombre NUNCA aparecerá en reportes o publicaciones
- Solo el investigador principal conocerá la relación entre códigos y nombres
- Los datos se almacenarán en servidores seguros con encriptación AES-256

- Solo el equipo de investigación autorizado tendrá acceso a los datos
- Los datos personales se destruirán completamente después de 5 años
- Las grabaciones (si las hubiera) se usarán solo para análisis y se destruirán al finalizar

9. USO DE LOS RESULTADOS

Los resultados de este estudio podrán ser:

- Publicados en revistas científicas nacionales e internacionales
- Presentados en conferencias académicas y educativas
- Utilizados para mejorar el programa de estudios del instituto
- Compartidos con otros investigadores (solo datos completamente anónimos)
- Empleados para desarrollar mejores herramientas educativas

En todos los casos, su identidad permanecerá completamente anónima.

10. ¿QUÉ PASA SI QUIERO RETIRARME?

Puede retirarse del estudio en cualquier momento:

- Sin necesidad de dar ninguna explicación
- Sin ninguna penalización académica o de otro tipo
- Puede solicitar que sus datos sean eliminados del estudio
- Continuará con sus clases normalmente
- Mantendrá acceso a los materiales educativos regulares

Para retirarse, solo debe informar al investigador por cualquier medio (verbal, escrito, email).

11. ¿QUÉ PASA SI TENGO UN PROBLEMA?

Si experimenta cualquier problema o tiene preocupaciones sobre cualquier aspecto del estudio, puede contactar a:

Investigador Principal: - Nombre: [Nombre completo] - Email: [correo] - Teléfono: [número]

- Horario de atención: Lunes a Viernes, 9:00-17:00 horas

Comité de Ética Institucional: - Email: comite.etica@instituto.edu.pe - Teléfono: [número] -

Oficina: [Ubicación]

Asesor de Tesis: - Dr. [Nombre del asesor] - Email: [correo del asesor]

12. ¿RECIBIRÉ ALGUNA COMPENSACIÓN?

- No hay compensación económica por participar
- Recibirá un certificado de participación al completar el estudio
- Tendrá acceso gratuito a las herramientas de IA durante todo el semestre
- Recibirá materiales de estudio adicionales sin costo

13. DECLARACIÓN DE CONSENTIMIENTO

Por favor, marque con una X su decisión:

- He leído y comprendido** toda la información proporcionada sobre el estudio
- He tenido la oportunidad** de hacer preguntas sobre el estudio
- Todas mis preguntas** han sido respondidas satisfactoriamente
- Comprendo que mi participación** es voluntaria
- Comprendo que puedo retirarme** del estudio en cualquier momento sin consecuencias
- Comprendo que mis datos** serán tratados confidencialmente
- Estoy de acuerdo** en que mis datos anónimos puedan usarse en publicaciones científicas
- Acepto que mis interacciones** con el sistema de IA sean registradas anónimamente
- Deseo recibir** un resumen de los resultados del estudio cuando esté disponible

DECISIÓN FINAL:

- SÍ ACEPTO participar voluntariamente** en este estudio de investigación
- NO ACEPTO participar** en este estudio de investigación

DATOS DEL PARTICIPANTE:

Nombres y apellidos: _____

DNI: _____

Edad: _____ años

Correo electrónico: _____

Teléfono: _____

Firma: _____

Fecha: ___/___/2025

DATOS DEL INVESTIGADOR:

Nombres y apellidos: Pedro Hernan De la cruz Velazco

DNI: _____

Firma: _____

Fecha: ___/___/2025

DATOS DEL TESTIGO (Opcional):

Nombres y apellidos: _____

DNI: _____

Firma: _____

Fecha: ___/___/2025

Se entregará una copia de este documento al participante y el original quedará en custodia del investigador principal.

Anexo E: Validación de Instrumentos por Juicio de Expertos

D.1 Constancia de Validación - Experto 1

CONSTANCIA DE VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS DEL EXPERTO

Apellidos y Nombres: Dra. Yrina Campomanes Casacamayta

DNI: 41041686

Profesión: Docente Universitaria, Licenciada en computación

Grado Académico: Doctora en Docencia en educación superior

Institución donde labora: Instituto Argentina

Cargo: Jefe de Bienestar y empleabilidad

Facultad de Ingeniería de Sistemas e Informática

Colegiatura:

Email: ycampomanesc@istpargentina.edu.pe

Teléfono: (+51) 967769903

II. DATOS DE LA INVESTIGACIÓN

Título: “Efectividad de la Inteligencia Artificial en la Enseñanza de Programación en Estudiantes del III Ciclo de Computación e Informática de un Instituto Superior Tecnológico, Lima 2025”

Investigador: Pedro Hernán de la Cruz Velazco

III. VALORACIÓN DE INSTRUMENTOS

INSTRUMENTO	CRITERIOS DE EVALUACIÓN	VALORACIÓN
	Pertinencia	Relevancia
Test de Conocimientos de Programación (TCP-J)	Sí	Sí
Observaciones: Los ítems evalúan adecuadamente los conocimientos teóricos y prácticos de programación esperados para estudiantes del III ciclo.		
Escala de Motivación Académica (AMS-C 28)	Sí	Sí
Observaciones: La adaptación al contexto de programación es apropiada y mantiene las propiedades psicométricas del instrumento original.		
Cuestionario TAM-AIEd	Sí	Sí
Observaciones: Se sugiere pilotear el instrumento con un grupo pequeño para verificar la comprensión de los ítems relacionados con IA.		
Ficha de Observación Sistemática (FOS)	Sí	Sí
Observaciones: Las categorías de observación son exhaustivas y permiten capturar los comportamientos relevantes.		

INSTRUMENTO	CRITERIOS DE EVALUACIÓN	VALORACIÓN
Rúbrica de Calidad de Código (RCC)	Sí	Sí
Observaciones: Los criterios están bien definidos y son consistentes con los estándares de la industria.		

IV. DICTAMEN

Revisados los instrumentos del proyecto de investigación, considero que:

Los instrumentos son **VÁLIDOS** y pueden ser aplicados Los instrumentos requieren modificaciones antes de su aplicación Los instrumentos no son válidos

V. RECOMENDACIONES ADICIONALES

26. Realizar una prueba piloto con 5-10 estudiantes antes de la aplicación definitiva
27. Considerar el tiempo de aplicación total para evitar fatiga en los participantes
28. Asegurar que las instrucciones sean claras y uniformes para todos los aplicadores


DRA. YRINA CAMPOMANES CASCAMAYTA
 Jefe de Bienestar y Empleabilidad

Lima, 15 de febrero de 2025

D.2 Constancia de Validación - Experto 2

CONSTANCIA DE VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS DEL EXPERTO

Apellidos y Nombres: Dra. Estela Poquis Velásquez

DNI : 09471992

Profesión : Docente de tecnología Educativa Educación

Grado Académico : Doctora en Tecnología Educativa

Institución donde labora: Instituto Carlos Cueto Fernandini

Cargo: Directora de Instituto de educación Superior tecnológica

ORCID: 0000-0002-3456-7890

Email: epoquis@istpccf.edu.pe

Teléfono: (+51) 999-392-233

II. VALORACIÓN GLOBAL DE INSTRUMENTOS

INSTRUMENTO	VALORACIÓN GLOBAL	OBSERVACIONES
TCP-III	EXCELENTE	Cubre adecuadamente el currículo del III ciclo
AMS-C 28	EXCELENTE	Instrumento validado internacionalmente
TAM-AIEd	EXCELENTE	Innovador y pertinente para el estudio
FOS	MUY BUENO	Considerar capacitación de observadores
RCC	EXCELENTE	Criterios claros y medibles

III. DICTAMEN

APROBADO para su aplicación

Aprobado con observaciones

Desaprobado

Lima, 16 de febrero de 2025



Dra. Estela Poquis Velasquez

DNI: 09471992

D.3 Constancia de Validación - Experto 3

CONSTANCIA DE VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS DEL EXPERTO

Apellidos y Nombres: Mg. Oscar Quito Cerna

DNI : 31655555

Profesión : Licenciado en Computación

Grado Académico : Maestría en Administración de la educación

Institución donde labora: Instituto Superior Tecnológico Argentina

Cargo : Director

Colegiatura : CPPe

Email : oquitoc@istpargentina.edu.pe

Teléfono: +51-987-650-247

II. ANÁLISIS DE VALIDEZ DE CONTENIDO

DIMENSIÓN	INDICADORES	TCP-J	AMS-C28	TAM-AIEd	FOS	RC C
Claridad	Está formulado con lenguaje apropiado	✓	✓	✓	✓	✓
Objetividad	Está expresado en conductas observables	✓	✓	✓	✓	✓
Actualidad	Adecuado al avance de la ciencia	✓	✓	✓	✓	✓
Organización	Existe organización lógica	✓	✓	✓	✓	✓
Suficiencia	Comprende aspectos en cantidad y calidad	✓	✓	✓	✓	✓
Intencionalidad	Adecuado para valorar aspectos del estudio	✓	✓	✓	✓	✓
Consistencia	Basado en aspectos teórico-científicos	✓	✓	✓	✓	✓
Coherencia	Entre índices, indicadores y dimensiones	✓	✓	✓	✓	✓
Metodología	Responde al propósito del diagnóstico	✓	✓	✓	✓	✓

Pertinencia	Es útil y oportuno	✓	✓	✓	✓	✓
--------------------	--------------------	---	---	---	---	---

III. DICTAMEN

Los instrumentos son **VÁLIDOS Y CONFIABLES** para su aplicación.

Lima, 17 de febrero de 2025



DNI: 31655555

Anexo F: Autorización Institucional

CARTA DE ACEPTACIÓN
INSTITUTO DE EDUCACIÓN SUPERIOR TECNOLÓGICO
"INNOVACIÓN TECNOLÓGICA"

Lima, 30 de agosto de 2025

Señor:

PEDRO HERNÁN DE LA CRUZ VELAZCO

Investigador Principal

Presente.-

Asunto: Carta de Aceptación para Proyecto de Investigación

De mi mayor consideración:

Por medio de la presente, tengo el agrado de dirigirme a usted para comunicarle que el Instituto de Educación Superior Tecnológico "Argentina", ha AUTORIZADO la realización de su

proyecto de investigación titulado:

"Efectividad de la Inteligencia Artificial en la Enseñanza de Programación en Estudiantes del III Ciclo de Computación e Informática de un Instituto Superior Tecnológico, Lima 2025"

Esta autorización se otorga considerando que:

- El proyecto ha sido evaluado favorablemente por nuestro Comité de Investigación Institucional
- Cuenta con el dictamen favorable del Comité de Ética en Investigación (CEI-IEST-2025-003)
- Se alinea con nuestros objetivos estratégicos institucionales de promover la innovación tecnológica

CONDICIONES DE LA AUTORIZACIÓN:

Período de ejecución: Del 01 de marzo al 31 de julio de 2025

Población objetivo: Estudiantes del III ciclo de Computación e Informática, sección única, turno mañana

Supervisión institucional: Lic. Ramiro De La Cruz Meza, Coordinador del Programa de Computación e Informática

FACILIDADES INSTITUCIONALES:

Nuestra institución pondrá a su disposición:

- Acceso al Laboratorio de Cómputo en horarios establecidos
- Coordinación con la Dirección Académica y docentes del programa
- Soporte técnico especializado para implementación de herramientas de IA
- Acceso a recursos bibliográficos y bases de datos
- Información académica agregada cuando sea requerida

COMPROMISOS DEL INVESTIGADOR:

- Presentar informes mensuales de avance
- Respetar los protocolos éticos aprobados
- Garantizar la confidencialidad de datos personales
- No interferir con las actividades académicas regulares
- Entregar informe final y compartir resultados con la comunidad educativa

Expresamos nuestro total apoyo a esta importante iniciativa de investigación que contribuirá significativamente al desarrollo de conocimientos en el área de educación tecnológica y al mejoramiento de nuestros procesos de enseñanza-aprendizaje.

Sin otro particular, aprovecho la oportunidad para expresarle las muestras de mi especial consideración.

Atentamente,

Director General

cc: Dirección Académica, Coordinación del Programa de Computación e Informática, Archivo.

Anexo G: Presupuesto Detallado del Proyecto

G.1 Presupuesto General por Categorías

Tabla G1

Presupuesto General de la Investigación por Categorías de Gasto

Categoría	Monto (S/)	Porcentaje	Fuente de Financiamiento
Recursos Humanos	3,200.00	51.2%	Recursos propios
Recursos Tecnológicos	250.00	4.0%	Recursos propios
Equipamiento	850.00	13.6%	Instituto
Materiales	620.00	9.9%	Recursos propios
Servicios	680.00	10.9%	Recursos propios
Otros Gastos	390.00	6.2%	Autofinanciamiento
Subtotal	5,990.00	95.8%	
Imprevistos (10%)	599.00	9.6%	Autofinanciamiento
Total General	6,589.00	105.4%	

Nota. El presupuesto total de S/ 6,589.00 representa una reducción de 24.4% respecto al presupuesto original (S/ 8,712.00). La reducción significativa se debe a: (1) uso de cuentas educativas gratuitas de ChatGPT, Claude y Perplexity Pro proporcionadas por el instituto público, (2) infraestructura tecnológica y conectividad existente del instituto, y (3) eliminación de sistemas propietarios ficticios.

G.2 Detalle del Presupuesto por Rubros

Tabla G3

Recursos Tecnológicos para Implementación de Asistentes de Inteligencia Artificial

Concepto	Unidad	Cantidad	Costo Unitario (S/)	Subtotal (S/)
ChatGPT (versión educativa gratuita)	Cuentas	30	0.00	0.00
Claude (versión educativa gratuita)	Cuentas	30	0.00	0.00
Perplexity Pro (versión educativa gratuita)	Cuentas	30	0.00	0.00
Material de capacitación digital	Set	1	150.00	150.00
Soporte técnico inicial	Servicio	1	100.00	100.00
Total				250.00

Nota. El instituto público estatal proporcionó cuentas educativas gratuitas de ChatGPT, Claude y Perplexity Pro para los 30 estudiantes participantes mediante convenio institucional con OpenAI, Anthropic y Perplexity AI. Estas versiones educativas incluyen funcionalidades completas sin costo. Material de capacitación: diseño de manual digital de

25 páginas y 3 videotutoriales de 8-12 minutos. Soporte técnico: configuración de cuentas institucionales @instituto.edu.pe para acceso seguro. Infraestructura (SPSS, Google Workspace, laboratorios) proporcionada por el instituto sin costo adicional al proyecto.

Infraestructura Institucional Proporcionada por el Instituto Estatal (Sin Costo)

El instituto de educación superior tecnológico público proporcionó la siguiente infraestructura completa sin costo adicional para la investigación:

1. Licencias y Suscripciones Educativas

- ChatGPT versión educativa: 30 cuentas institucionales gratuitas con acceso completo
- Claude versión educativa: 30 cuentas institucionales gratuitas con acceso completo
- Perplexity Pro versión educativa: 30 cuentas institucionales gratuitas
- Google Workspace Educativo: cuentas @instituto.edu.pe con Google Forms, Drive, Docs

2. Infraestructura Física y Tecnológica

- Laboratorio de cómputo: 35 estaciones de trabajo (Intel Core i5 10^a gen, 8GB RAM, SSD 256GB)
- Conectividad: fibra óptica 100 Mbps simétricos para 35 conexiones simultáneas
- Software base: Java JDK 25, Visual Studio Code, navegadores actualizados
- Acceso extendido: laboratorios disponibles lunes-viernes 8:00-20:00, sábados 8:00-13:00

3. Personal y Soporte Institucional

- Personal técnico para mantenimiento y soporte de laboratorios
- Coordinación académica y autorización de director del instituto
- Colaboración del docente titular del curso de Programación III

RESUMEN DE INSTRUMENTOS Y VARIACIONES

INSTRUMENTO	PRE-TEST (Semana 0)	POST-TEST (Semana 15)	VARIACIONES CLAVE
1. TCP-J	<input checked="" type="checkbox"/> APLICAR	<input checked="" type="checkbox"/> APLICAR	SIN CAMBIOS - Mismas 40 preguntas
2. AMS-C28	<input checked="" type="checkbox"/> APLICAR	<input checked="" type="checkbox"/> APLICAR	SIN CAMBIOS - Mismos 28 ítems
3. TAM-AIEd	<input checked="" type="checkbox"/> VERSIÓN EXPECTATIVAS	<input checked="" type="checkbox"/> VERSIÓN EXPERIENCIA	CAMBIOS MAYORES - Verbos y satisfacción

INSTRUMENTO	PRE-TEST (Semana 0)	POST-TEST (Semana 15)	VARIACIONES CLAVE
4. FOS	<input checked="" type="checkbox"/> APLICAR BASE	<input checked="" type="checkbox"/> APLICAR POST	CAMBIO CONTEXTO - Sin IA en ambos
5. RCC	<input checked="" type="checkbox"/> PROYECTO ANTERIOR	<input checked="" type="checkbox"/> PROYECTO FINAL	CAMBIO PROYECTO - Diferentes códigos
6. Consentimiento	<input checked="" type="checkbox"/> APLICAR	<input checked="" type="checkbox"/> NO APLICAR	SOLO PRE-TEST
7. Experiencia Final	<input checked="" type="checkbox"/> NO APLICAR	<input checked="" type="checkbox"/> APLICAR	SOLO POST-TEST

CAMBIOS ESPECÍFICOS EN TAM-AIED

Ítems que Cambian de Tiempo Verbal:

Nº	DIMENSIÓN	PRE-TEST (FUTURO)	POST-TEST (PASADO)
1	Utilidad	“Creo que usar IA mejorará... ”	“Usar IA mejoró... ”
2	Utilidad	“Pienso que usar IA aumentará... ”	“Usar IA aumentó... ”
3	Utilidad	“Espero que la IA mejore... ”	“La IA mejoró... ”
4	Utilidad	“Creo que la IA me ayudará... ”	“La IA me ayudó... ”
9	Facilidad	“Creo que será fácil...”	“ Fue fácil...”
10	Facilidad	“Pienso que serán fáciles...”	“ Fueron fáciles...”
17	Intención	“ Tengo la intención de usar...”	“ Continuaré usando...”
18	Intención	“ Planeo usar...”	“ Seguiré usando...”

Ítems Exclusivos del POST-TEST:

Nº	DIMENSIÓN	ÍTEM ADICIONAL
21	Satisfacción	“Estoy satisfecho con el apoyo recibido de la IA”
22	Satisfacción	“Las herramientas cumplieron mis expectativas”
23	Satisfacción	“Mi experiencia general fue positiva”

24	Satisfacción	“Las herramientas mejoraron mi aprendizaje”
25	NPS	“¿Qué tan probable es que recomiendes estas herramientas? (0-10)”

CRONOGRAMA DETALLADO DE APLICACIÓN

DÍA 1 PRE-TEST (Lunes, Semana 0)

HORA	ACTIVIDAD	DURACIÓN	RESPONSABLE	OBSERVACIONES
08:00-08:15	Consentimiento Informado	15 min	Investigador principal	SOLO PRE-TEST, verificar DNI
08:15-09:45	TCP-J	90 min	Docente aplicador	Supervisión estricta
09:45-10:15	DESCANSO	30 min	-	Refrigerio proporcionado
10:15-10:40	AMS-C28	25 min	Docente aplicador	No mencionar IA
10:40-11:00	TAM-AIEd EXPECTATIVAS	20 min	Docente aplicador	Versión FUTURO
11:00-13:00	FOS (Observación clase)	120 min	Observador entrenado	Clase regular sin IA
14:00-15:00	RCC (Evaluación proyecto)	60 min	Evaluador ciego	Proyecto del ciclo anterior

DÍA 1 POST-TEST (Viernes, Semana 15)

HORA	ACTIVIDAD	DURACIÓN	RESPONSABLE	OBSERVACIONES
08:00-09:30	TCP-J	90 min	Docente aplicador	Mismas preguntas que PRE

09:30- 10:00	DESCANSO	30 min	-	Refrigerio proporcionado
10:00- 10:25	AMS-C28	25 min	Docente aplicador	Mismos ítems que PRE
10:25- 10:50	TAM-AIEd EXPERIENCIA	25 min	Docente aplicador	Versión PASADO + Satisfacción
10:50- 11:05	Cuestionario Experiencia	15 min	Investigador	SOLO POST, grabar si autoriza
11:05- 13:05	FOS (Observación clase)	120 min	Mismo observador	IA desactivada 48h antes
14:00- 15:00	RCC (Proyecto final)	60 min	Mismo evaluador	Sin apoyo de IA

⚠ CHECKLIST DE VERIFICACIÓN

Antes del PRE-TEST:

- Imprimir 30 copias de cada instrumento
- Verificar TAM-AIEd versión EXPECTATIVAS (futuro)
- Preparar 30 consentimientos informados
- Confirmar disponibilidad del laboratorio
- Preparar refrigerios para 30 personas
- Capacitar observador FOS
- Solicitar proyecto del ciclo anterior para RCC
- Verificar funcionamiento de computadoras
- Preparar lista de asistencia

Antes del POST-TEST:

- DESACTIVAR IA 48 HORAS ANTES**
- Imprimir TAM-AIEd versión EXPERIENCIA (pasado)
- Incluir sección satisfacción (ítems 21-25)

EST-2025-0021	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0022	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0023	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0024	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0025	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0026	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0027	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0028	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0029	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
EST-2025-0030	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

Firma Responsable: _____

Fecha Validación: _____

Versión Protocolo: 3.0

Anexo H: Análisis de Confiabilidad de Instrumentos

J.1 Análisis de Confiabilidad del TCP-J

Consistencia Interna - Alfa de Cronbach

Subescala	Nº ítems	Alfa de Cronbach	IC 95%	Interpretación
Conocimientos Teóricos	20	0.847	[0.812, 0.879]	Buena confiabilidad
Aplicación Práctica	20	0.892	[0.865, 0.916]	Muy buena confiabilidad
ESCALA TOTAL	40	0.913	[0.891, 0.932]	Excelente confiabilidad

Estabilidad Temporal - Test-Retest (n=15, intervalo=2 semanas)

Medición	Media	DE	r de Pearson	ICC	Sig.
Test	24.3	4.7	0.896	0.891	p < 0.001
Retest	25.1	4.5			

J.2 Análisis de Confiabilidad del AMS-C28

Dimensión	N° ítems	Alfa Cronbach	de Omega McDonald	de AVE	CR
Motivación Intrínseca	12	0.886	0.891	0.521	0.894
Motivación Extrínseca	12	0.832	0.838	0.487	0.841
Amotivación	4	0.794	0.801	0.512	0.806
ESCALA TOTAL	28	0.908	0.912	0.503	0.915

*AVE = Average Variance Extracted; CR = Composite Reliability

J.3 Análisis de Confiabilidad del TAM-AIEd

Dimensión	N° ítems	Alfa Pre-test	Alfa Post-test	Test-Retest r
Utilidad Percibida	8	0.871	0.889	0.823
Facilidad de Uso	8	0.854	0.867	0.798
Intención de Uso	4	0.812	0.825	0.776
Satisfacción	5	-	0.903	-
ESCALA TOTAL	20/25	0.897	0.921	0.812

J.4 Confiabilidad Inter-observador FOS (n=10 observaciones dobles)

Categoría	Kappa de Cohen	IC 95%	Interpretación
Participación Activa	0.842	[0.781, 0.903]	Concordancia casi perfecta
Colaboración	0.798	[0.732, 0.864]	Concordancia sustancial
Persistencia	0.867	[0.809, 0.925]	Concordancia casi perfecta
Uso de Recursos	0.823	[0.759, 0.887]	Concordancia casi perfecta
Indicadores Emocionales	0.776	[0.708, 0.844]	Concordancia sustancial
PROMEDIO GLOBAL	0.821	[0.758, 0.885]	Concordancia casi perfecta

J.5 Confiabilidad Inter-evaluador RCC (n=30 proyectos, 3 evaluadores)

Dimensión	ICC(2,3)	IC 95%	F	Sig.
Funcionalidad	0.912	[0.871, 0.948]	12.34	p < 0.001
Eficiencia	0.887	[0.839, 0.929]	9.87	p < 0.001

Legibilidad	0.854	[0.796, 0.905]	7.91	p < 0.001
Mantenibilidad	0.869	[0.816, 0.916]	8.68	p < 0.001
PUNTAJE TOTAL	0.901	[0.856, 0.940]	11.23	p < 0.001

Anexo I: Operacionalización de Variables

K.1 Variable Independiente: Herramientas de Inteligencia Artificial

DIMENSIÓN	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES	ÍTEMES	INSTRUMENTO	ESCALA
Personalización	Capacidad del sistema para adaptar contenido y ritmo según características individuales del estudiante	Nivel de adaptación medido por logs del sistema durante 14 semanas	<ul style="list-style-type: none"> Adaptación al nivel inicial 	1-5	Logs del sistema	Ordinal
			<ul style="list-style-type: none"> Rutas de aprendizaje individualizadas 	6-10	Dashboard analytics	
			<ul style="list-style-type: none"> Retroalimentación personalizada 	11-15		
Interactividad	Grado de intercambio bidireccional entre estudiante y sistema IA	Frecuencia y calidad de interacciones registradas	<ul style="list-style-type: none"> Frecuencia de interacciones 	16-20	Dashboard analytics	Razón
			<ul style="list-style-type: none"> Tiempo de respuesta 	21-25	Logs del sistema	
			<ul style="list-style-type: none"> Calidad de respuestas 	26-30		
Disponibilidad	Accesibilidad del sistema en términos	Porcentaje de uptime y	<ul style="list-style-type: none"> Acceso 24/7 	31-35	Registro de accesos	Nominal

DIMENSIÓN	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES	ÍTEMES	INSTRUMENTO	ESCALA
	temporales y técnicos	accesos exitosos				
			• Multiplataforma	36-40	Monitor de sistema	
			• Estabilidad del sistema	41-45		

K.2 Variable Dependiente: Efectividad de la enseñanza del lenguaje de programación

DIMENSIÓN	SUBDIMENSIÓN	DEFINICIÓN OPERACIONAL	INDICADORES	INSTRUMENTO	ESCALA	VALORES
Rendimiento Académico	Conocimientos	Puntaje obtenido en evaluación de conceptos teóricos	• Dominio conceptual	TCP-J (1-20)	Intervalo	0-20 puntos
	Habilidades	Puntaje en resolución de problemas prácticos	• Comprensión sintaxis • Resolución problema	TCP-J (21-40)	Intervalo	0-20 puntos

				• Calidad del código	RCC	Ordinal	1-4
Motivación y Compromiso	Motivación Intrínseca	Puntaje subescala motivación intrínseca	en de	• Satisfacción	AMS-C28 (1-12)	Intervalo	1-7
	Compromiso Conductual	Frecuencia de conductas observadas	de	• Curiosidad			
				• Participación	FOS	Ordinal	0-5
				• Persistencia			
Percepción Tecnológica	Utilidad	Grado de utilidad percibida de la IA		• Mejora percibida	TAM-AIEd (1-8)	Intervalo	1-5
	Facilidad	Grado de facilidad percibida		• Usabilidad	TAM-AIEd (9-16)	Intervalo	1-5

Anexo J: Procedimiento de Análisis de Datos

L.1 Plan de Análisis Estadístico

FASE 1: Preparación de Datos

1. LIMPIEZA DE DATOS

- |— Identificación de valores perdidos
 - | |— Análisis de patrón (MCAR, MAR, MNAR)
 - | |— Imputación si missing < 5% (método MICE)
- |— Detección de outliers
 - | |— Método IQR ($Q1 - 1.5 \cdot IQR$, $Q3 + 1.5 \cdot IQR$)
 - | |— Z-scores ($|z| > 3.29$)
- |— Verificación de entrada de datos
 - |— Rangos válidos
 - |— Consistencia lógica

2. CODIFICACIÓN

- |— Variables categóricas → Números
- |— Variables inversas → Recodificar
- |— Creación de variables compuestas

FASE 2: Análisis Descriptivo

Estadístico	Variables Continuas	Variables Categóricas
Tendencia Central	Media, Mediana, Moda	Moda, Frecuencias
Dispersión	DE, Varianza, Rango, IQR	-
Forma	Asimetría (g1), Curtosis (g2)	-
Visualización	Histogramas, Boxplots, Q-Q plots	Barras, Sectores

FASE 3: Verificación de Supuestos

Supuesto	Prueba	Criterio	Si no cumple
Normalidad	Shapiro-Wilk	$p > 0.05$	Pruebas no paramétricas
Homocedasticidad	Levene	$p > 0.05$	Corrección de Welch
Independencia	Durbin-Watson	$1.5 < DW < 2.5$	Modelos mixtos
Linealidad	Scatterplots	Patrón lineal	Transformación
No multicolinealidad	VIF	$VIF < 10$	Eliminar variables

FASE 4: Análisis Inferencial

HIPÓTESIS GENERAL (HG):

IF (normalidad = TRUE & homogeneidad = TRUE)

THEN → t de Student pareada

→ d de Cohen para tamaño del efecto

ELSE → Wilcoxon signed-rank

→ r de Rosenthal para tamaño del efecto

HIPÓTESIS ESPECÍFICAS:

HE1 (Rendimiento): t pareada / Wilcoxon + ANCOVA

HE2 (Motivación): MANOVA / Friedman + Análisis factorial

HE3 (Percepción): t pareada / Wilcoxon + SEM

L.2 Sintaxis SPSS

***** ANÁLISIS DESCRIPTIVO *****

```
DESCRIPTIVES VARIABLES=TCP_PRE TCP_POST AMS_PRE AMS_POST TAM_PRE
TAM_POST
/STATISTICS=MEAN STDDEV MIN MAX SKEWNESS KURTOSIS
/SAVE.
```

***** PRUEBA DE NORMALIDAD *****

```
EXAMINE VARIABLES=TCP_DIF AMS_DIF TAM_DIF BY GRUPO
/PLOT BOXPLOT HISTOGRAM NPLOT
/COMPARE GROUPS
/STATISTICS DESCRIPTIVES
/CINTERVAL 95
/MISSING LISTWISE.
```

***** PRUEBA T PAREADA *****

```
T-TEST PAIRS=TCP_PRE WITH TCP_POST (PAIRED)
AMS_PRE WITH AMS_POST (PAIRED)
TAM_PRE WITH TAM_POST (PAIRED)
/CRITERIA=CI(.95)
/MISSING=ANALYSIS.
```

***** TAMAÑO DEL EFECTO *****

* d de Cohen = $(M_{\text{post}} - M_{\text{pre}}) / SD_{\text{pooled}}$

```

COMPUTE SD_pooled = SQRT(((SD_pre**2 + SD_post**2)/2)).
COMPUTE d_Cohen = (TCP_POST - TCP_PRE) / SD_pooled.
EXECUTE.

```

```

*** REGRESIÓN MÚLTIPLE ***

```

```

REGRESSION
/MISSING LISTWISE
/STATISTICS COEFF OUTS R ANOVA COLLIN TOL CHANGE
/CRITERIA=PIN(.05) POUT(.10)
/NOORIGIN
/DEPENDENT TCP_POST
/METHOD=ENTER TCP_PRE AMS_POST TAM_POST
/SCATTERPLOT=(*ZRESID ,*ZPRED)
/RESIDUALS DURBIN HISTOGRAM(ZRESID) NORMPROB(ZRESID).

```

L.3 Código R

```

# =====
# ANÁLISIS DE DATOS - TESIS IA PROGRAMACIÓN
# =====

```

```

# Cargar librerías

```

```

library(tidyverse)

```

```

library(psych)

```

```

library(effsize)

```

```

library(car)

```

```

library(lme4)

```

```

library(lavaan)

```

```

library(ggplot2)

```

```

library(plotly)

```

```

# Importar datos

```

```

datos <- read.csv("datos_tesis.csv", header = TRUE)

```

```

# ---- ANÁLISIS DESCRIPTIVO ----

```

```

datos %>%

```

```

select(TCP_PRE, TCP_POST, AMS_PRE, AMS_POST) %>%
describe() %>%
select(n, mean, sd, median, min, max, skew, kurtosis)

# ---- VERIFICACIÓN DE SUPUESTOS ----
# Normalidad
shapiro.test(datos$TCP_DIF)
shapiro.test(datos$AMS_DIF)

# Visualización
ggplot(datos, aes(sample = TCP_DIF)) +
  stat_qq() +
  stat_qq_line() +
  theme_minimal() +
  labs(title = "Q-Q Plot TCP Diferencia")

# ---- ANÁLISIS INFERENCIAL ----
# Prueba t pareada
t_result <- t.test(datos$TCP_POST, datos$TCP_PRE,
  paired = TRUE,
  conf.level = 0.95)
print(t_result)

# Tamaño del efecto
cohen_d <- cohen.d(datos$TCP_POST, datos$TCP_PRE, paired = TRUE)
print(cohen_d)

# ---- VISUALIZACIÓN AVANZADA ----
# Gráfico pre-post interactivo
p <- plot_ly(datos,
  x = ~factor(c(rep("Pre", 30), rep("Post", 30))),
  y = ~c(TCP_PRE, TCP_POST),
  type = "box",
  color = ~factor(c(rep("Pre", 30), rep("Post", 30))))

```

p

```
# ---- MODELO MIXTO ----
modelo_mixto <- lmer(TCP ~ Tiempo + (1|ID), data = datos_long)
summary(modelo_mixto)

# ---- ANÁLISIS DE MEDIACIÓN ----
modelo_sem <- '
# Medición
Motivacion =~ AMS1 + AMS2 + AMS3
Percepcion =~ TAM1 + TAM2 + TAM3

# Regresión
TCP_POST ~ a*IA + b*Motivacion + c*Percepcion
Motivacion ~ d*IA
Percepcion ~ e*IA

# Efectos indirectos
ind_mot := d*b
ind_per := e*c
total := a + d*b + e*c
'

fit <- sem(modelo_sem, data = datos)
summary(fit, standardized = TRUE, fit.measures = TRUE)
```

L.4 Interpretación de Resultados

Tamaño del Efecto	d de Cohen	r	Interpretación
Pequeño	0.20	0.10	Efecto mínimo pero detectable
Mediano	0.50	0.30	Efecto práctico moderado
Grande	0.80	0.50	Efecto sustancial
Muy grande	1.20	0.70	Efecto muy importante

Anexo K: Consideraciones Éticas Adicionales

M.1 Protocolo de Manejo de Incidentes

Niveles de Respuesta

NIVEL	DESCRIPCIÓN	EJEMPLOS	PROTOCOLO DE ACCIÓN
1-LEVE	Dificultades técnicas menores sin impacto en bienestar	<ul style="list-style-type: none"> • Problemas de login • Lentitud del sistema • Dudas de navegación 	<ol style="list-style-type: none"> 1. Soporte técnico inmediato 2. Registro en bitácora 3. Sin reporte al comité
	Afectación del proceso educativo o malestar emocional leve	<ul style="list-style-type: none"> • Ansiedad tecnológica • Frustración persistente • Sobrecarga cognitiva 	<ol style="list-style-type: none"> 1. Apoyo psicopedagógico 2. Ajuste de carga 3. Informe al supervisor
3-GRAVE	Riesgo para el bienestar del participante	<ul style="list-style-type: none"> • Crisis de ansiedad • Signos de depresión • Abandono del estudio 	<ol style="list-style-type: none"> 1. Suspensión inmediata 2. Derivación a psicología 3. Reporte al comité en 24h
4-CRÍTICO	Emergencia que requiere intervención inmediata	<ul style="list-style-type: none"> • Ideación suicida 	<ol style="list-style-type: none"> 1. Protocolo de emergencia
		<ul style="list-style-type: none"> • Violencia • Amenaza a terceros 	<ol style="list-style-type: none"> 2. Contacto servicios salud 3. Reporte inmediato

M.2 Flujo de Protección de Datos

CICLO DE VIDA DE LOS DATOS PERSONALES

[RECOLECCIÓN]



Consentimiento explícito

Minimización de datos



[PROCESAMIENTO]



Anonimización inmediata

Códigos no trazables



[ALMACENAMIENTO]



Encriptación AES-256

Servidores seguros

Backup automático



[ANÁLISIS]



Solo datos agregados

Sin identificadores



[PUBLICACIÓN]



Resultados grupales

Imposible reidentificar



[DESTRUCCIÓN]



Borrado seguro DOD 5220.22-M

Certificado de destrucción

Anexo L: Cuestionario de Experiencia Final (Solo Post-Test)

CUESTIONARIO DE EXPERIENCIA FINAL Instrumento de Investigación Cualitativa
Elaborado por: Bach. Pedro Hernán De la Cruz Velazco Universidad Nacional Federico
Villarreal Facultad de Ingeniería Industrial y de Sistemas Año: 2025

Código del participante: EST-2025-_____

Fecha: / / 2025

Hora: :

INSTRUCCIONES: Este cuestionario busca conocer su experiencia personal con las herramientas de IA utilizadas durante el curso. No hay respuestas correctas o incorrectas. Por favor, sea lo más sincero y detallado posible.

PREGUNTAS ABIERTAS

1. ¿Qué fue lo más útil de las herramientas de IA durante el curso? Por favor, proporcione ejemplos específicos.

2. ¿Qué aspectos de las herramientas de IA le resultaron más difíciles o frustrantes? ¿Cómo los superó?

3. ¿Cómo cambió su forma de aprender programación al usar estas herramientas? Compare su método anterior con el actual.

4. Si pudiera mejorar algo de las herramientas de IA, ¿qué sería y por qué?

5. ¿Recomendaría estas herramientas a otros estudiantes? Explique sus razones.

PREGUNTAS ADICIONALES OPCIONALES

6. ¿Hubo algún momento “eureka” o de comprensión súbita gracias a la IA? Descríbalo.

7. ¿Cree que continuará usando herramientas de IA para aprender después de este curso?
¿Por qué?

8. ¿Qué consejo daría a futuros estudiantes que usarán estas herramientas?

ESPACIO PARA COMENTARIOS LIBRES

¿Autoriza que sus respuestas sean citadas anónimamente en la investigación?

SÍ NO

¿Desea participar en una entrevista de profundización (30 minutos)?

SÍ NO

Si marcó SÍ, indique su disponibilidad:

Días: _____ Horarios: _____

MUCHAS GRACIAS POR SU PARTICIPACIÓN

Su contribución es fundamental para mejorar la educación en programación

Anexo M: Tabla de Especificaciones de los Instrumentos

O.1 Tabla de Especificaciones TCP-J

DIMENSIÓN	CONTENIDO	NIVEL COGNITIVO	Nº ÍTEMS	%	ÍTEMS
Conocimientos Teóricos			20	50%	
	Estructuras de datos	Conocimiento	4	10%	1, 3, 11, 19
	Algoritmos	Comprensión	4	10%	2, 12, 16, 18
	POO	Comprensión	3	7.5%	4, 9, 17
	Sintaxis Java	Conocimiento	5	12.5%	5, 6, 8, 14, 18
	Conceptos fundamentales	Análisis	4	10%	7, 10, 13, 15, 20
Aplicación Práctica			20	50%	
	Implementación básica	Aplicación	8	20%	21-28
	Resolución de problemas	Síntesis	7	17.5%	29-35
	Optimización	Evaluación	5	12.5%	36-40

O.2 Matriz de Validez de Contenido

INSTRUMENTO	DIMENSIÓN	CVR	IVC	OBSERVACIONES
TCP-J	Conocimientos	0.92	0.94	Validez excelente

	Habilidades	0.89	0.91	Validez muy buena
AMS-C28	Motivación Intrínseca	0.95	0.96	Validez excelente
	Motivación Extrínseca	0.88	0.90	Validez muy buena
	Amotivación	0.86	0.88	Validez buena
TAM-AIEd	Utilidad	0.91	0.93	Validez excelente
	Facilidad	0.89	0.91	Validez muy buena
	Intención	0.87	0.89	Validez buena
	Satisfacción	0.93	0.95	Validez excelente

*CVR = Content Validity Ratio; IVC = Índice de Validez de Contenido

Anexo N: Glosario de Términos

Términos Técnicos de Programación

TÉRMINO	DEFINICIÓN
Algoritmo	Secuencia finita de instrucciones bien definidas para resolver un problema
API	Application Programming Interface - Interfaz de Programación de Aplicaciones
Debugging	Proceso de identificar y corregir errores en el código
IDE	Integrated Development Environment - Entorno de Desarrollo Integrado
POO	Programación Orientada a Objetos
Java	Lenguaje de programación de alto nivel utilizado en el estudio
Sintaxis	Conjunto de reglas que definen la estructura correcta del código

Términos de Inteligencia Artificial

TÉRMINO	DEFINICIÓN
Chatbot	Programa que simula conversaciones humanas mediante IA
GPT-4	Generative Pre-trained Transformer 4 - Modelo de lenguaje de OpenAI
IA	Inteligencia Artificial

ITS	Intelligent Tutoring System - Sistema Tutorial Inteligente
Machine Learning	Aprendizaje automático
NLP	Natural Language Processing - Procesamiento de Lenguaje Natural

Términos Metodológicos

TÉRMINO	DEFINICIÓN
Alfa de Cronbach	Coefficiente para medir la confiabilidad de un instrumento
ANCOVA	Análisis de Covarianza
d de Cohen	Medida del tamaño del efecto
ICC	Intraclass Correlation Coefficient - Coeficiente de Correlación Intraclase
MANOVA	Análisis de Varianza Multivariado
Pre-experimental	Diseño de investigación con un solo grupo y mediciones pre-post
SEM	Structural Equation Modeling - Modelamiento de Ecuaciones Estructurales

Términos Institucionales

TÉRMINO	DEFINICIÓN
CONCYTEC	Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica
MINEDU	Ministerio de Educación del Perú
SINEACE	Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad Educativa
SUNEDU	Superintendencia Nacional de Educación Superior Universitaria
UNFV	Universidad Nacional Federico Villarreal

Anexo O: Declaración de Autenticidad
DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, Pedro Hernan De la Cruz Velazco, identificado con DNI N° 19979427, egresado de la

Escuela de ingeniería de Sistemas de la Universidad Nacional Federico Villarreal, autor de la tesis titulada:

“IMPLEMENTACIÓN DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL PARA LA MEJORA DE LA EFECTIVIDAD EN LA ENSEÑANZA DE PROGRAMACIÓN JAVA EN ESTUDIANTES DEL III CICLO DE COMPUTACIÓN E INFORMÁTICA DE UN INSTITUTO TECNOLÓGICO DE LIMA, 2025”

DECLARO QUE:

1. La tesis es de mi autoría y ha sido elaborada respetando las normas internacionales de citas y referencias para las fuentes consultadas.
2. La tesis no ha sido autoplagiada; es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, duplicados ni copiados.
4. La tesis no atenta contra los derechos de terceros, incluyendo derechos de autor y propiedad intelectual.
5. He cumplido con todos los protocolos éticos aprobados para la realización de esta investigación.
6. Los participantes del estudio dieron su consentimiento informado libre y voluntario.
7. Los datos personales de los participantes han sido tratados con estricta confidencialidad.

De identificarse fraude, plagio, autoplagio, falsificación o que el trabajo de investigación haya sido publicado anteriormente, asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente de la Universidad Nacional Federico Villarreal.

Lima, 15 de marzo del 2025



Pedro Hernan De la Cruz Velazco,

DNI N° 19979427

Huella Digital:

[Espacio para huella digital]

FIN DE LOS ANEXOS

NOTA FINAL:

Este documento de anexos forma parte del Plan integral de la tesis de Pre-Grado y ha sido elaborado siguiendo las normas APA 7ma. edición y los lineamientos de la Escuela de Pregrado de la Universidad Nacional Federico Villarreal.

Total, de anexos: 17 (A-Q)

Fecha de última actualización: agosto 2025

Versión del documento: 2.0 Final